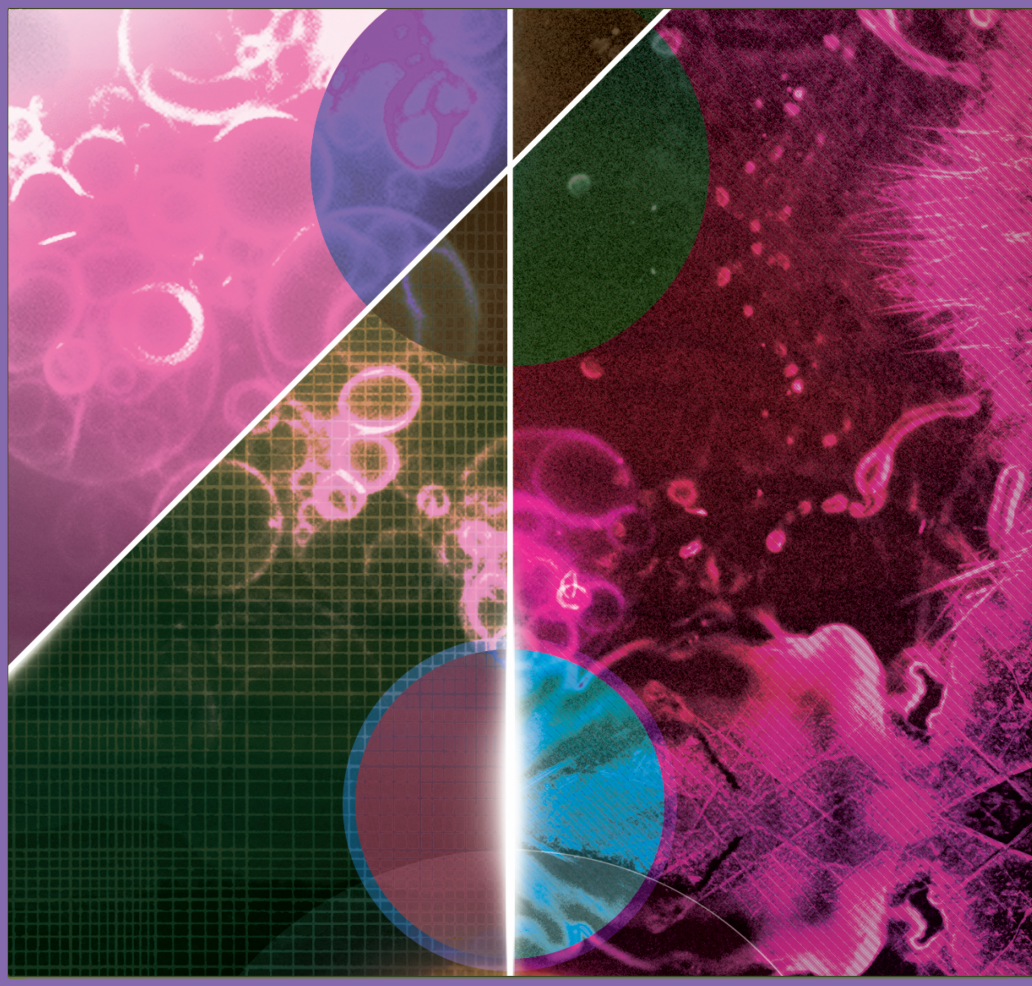
ARTIFICIAL LIFE XII



Proceedings of the Twelfth
International Conference on
the Synthesis and Simulation
of Living Systems

edited by

Harold Fellermann, Mark Dörr, Martin M. Hanczyc,
Lone Ladegaard Laursen, Sarah Maurer, Daniel Merkle,
Pierre-Alain Monnard, Kasper Støy, and Steen Rasmussen

Artificial Life XII

Artificial Life XII

Artificial Life XII Proceedings of the Twelfth International Conference on the Synthesis and Simulation of Living Systems

edited by

Harold Fellermann, Mark Dörr, Martin Hanczyc, Lone Ladegaard Laursen, Sarah Maurer, Daniel Merkle, Pierre-Alain Monnard, Kasper Støy, Steen Rasmussen

An ISAL Book

The MIT Press
Cambridge, Massachusetts
London, England

ISBN: 978-0-262-29075-3

Artificial Life XII:
The 12$^{\text{th}}$ International Conference on the Synthesis and Simulation of Living Systems

This is the proceeding for the Artificial Life XII Conference (`http://www.alife12.org/`), hosted by the Center for Fundamental Living Technology (FLinT) (`http://www.sdu.dk/flint/`) at University of Southern Denmark, Odense, August 19-23, 2010. Twenty three years ago in September 1987, the first Artificial Life Workshop was held at Los Alamos National Laboratory and the subsequent Alife workshops and conferences have been hosted in the US eight times (Los Alamos 1987, Santa Fe 1990 & 1992, MIT 1994, UCLA 1998, Reed 2000, Boston 2004), Japan once (Nara 1996), Australia once (Sydney 2002), England once (Southampton 2008) and now in Denmark (Odense 2010).

What is different about Alife XII?

You may have noticed that we have switched sequence of the concepts "Simulation" and "Synthesis" in the title of the conference to emphasize some changes within our community. First of all, the Alife XII submissions consist of a significantly higher fraction of wet Alife papers than at any earlier Alife conference. It is a pleasure to see how the communities from wet and soft Alife are increasingly engaging with each other. These submissions are also congruent with a clearer view in the broader scientific community on how we might create life either from scratch or through top-down design [1, 2, 3]. This trend is also reflected by a number of recent international collaborations across the top-down and the bottom-up communities, often sponsored under the title of synthetic biology.[1]

Living processes have been implemented and studied for many years in soft Alife systems (living processes implemented on computers), but the emergence of replicating programs from noisy computational environments remain an open issue. Significant progress has also been made for life-like robotics systems, for example through the development of polymorphic robots, where e.g. simple self-assembly, self-replication as well as complex collective behavior now have been obtained [4, 5].

In general, we see more integration between wet, hard, soft, and mixed living systems both within the Alife community and across the broader scientific and technological landscapes. This is in part captured by the definition of emerging living technology which comprises all technological applications of living and life-like processes at all levels [6].

As the Alife community inches closer to an understanding of life as a physical process by constructing living processes, we are also increasingly assessing the technological implications of the ability to engineer systems, whose power is based on the core features of life: robustness, adaptation, self-repair, self-assembly, and self-replication, centralized and distributed intelligence, and evolution [7].

In the coming years, we will likely see an accelerated movement towards more life-like, living, and intelligent processes as well as their integration across many technologies to form new biology-technology

[1] E.g., the European Science Foundation sponsored synthetics biology workshop on "Streamlined and synthetic genomes", November 16-17, 2009, Valencia, Spain. The Los Alamos National Laboratory sponsored synthetic biology workshop, June 28-29, 2010, Los Alamos NM, USA.

ecologies, that also include human institutions. If implemented appropriately, these new systems, technologies, and organizations could become more in tune with the needs of human society and the natural dynamics of the biosphere.

These developments are emerging from a knowledge convergence between a variety of sciences and technologies which we, within the Alife community, may group into (i) wet carbon-chemistry-based systems, (ii) computational and robotics based, ICT (information and communications technology) systems, and (iii) human organizations and institutions dominated by culture and human nature.

As part of the Alife XII program, we have scheduled a session "Looking backwards, looking forwards" to address the scientific questions related to these developments. Ten years have passed since the last Alife community status report [8, 9, 10], and we hope that this conference program can contribute to updating the critical open Alife questions. The day after the conclusion of the Alife XII conference, we have a one-day workshop for a similar discussion focused on the technological implications of Alife. Part of this discussion will be open to the public [11].

We should also emphasize that after 23 years, a hallmark for Alife community is still its scientific breath and inclusiveness. The Alife conferences clearly continue to act as a Big Tent, where scientists from many different disciplines and domains meet to present results and exchange ideas. This unique community feature has historically made the Alife community highly innovative, however it also makes peer review difficult as scientific methods vary dramatically across the many domains and disciplines. This breath also causes problems when papers need to be categorized into sessions as most papers in this volume could fit under several of the conference themes.

Background for Alife XII

For Alife XII 156 out of well over than 200 contributions (papers and abstracts) were accepted in the peer review process. These papers and abstracts represent authors from 34 countries and they consist of 152 (= 156 presentations - 4 plenary talks) contributed talks in four, and at times five, parallel sessions. All contributions have 15 minutes for their presentation and five minutes for discussion. The contributed plenary talks have 40 minutes. Alife XII also has a vibrant Poster Session, which is a crucial component of the Conference.

In addition to the peer reviewed presentations, Alife XII has six Satellite Events, which are proposed and organized by individuals and groups from the community. Traditionally, these workshops add an important dimension to the Alife meetings due to their free format and often more exploratory topic selection. Often, radically new ideas are presented in these workshops or tutorials on specific topics and explored in more details than regular peer reviewed presentations allow.

In order to assemble the Alife XII conference program, we have harvested as much domain and expert knowledge as reasonably possible. This process started well before the first call for papers with a call for contributed themes, where we consulted the invited Scientific Advisory Committee (SAC) for advice. The Organizing Committee (OC) solicited the SAC, which effort we are deeply indebted for. The Alife XII SAC consists of:

Chris Adami	Pascale Ehrenfreund	Andrés Moya
Martyn Amos	Takashi Ikegami	Ole Mouritsen
Wolfgang Banzhaf	Martin N. Jacobi	Peter Nielsen
Mark Bedau	David Krakauer	Norman Packard
Jim Boncella	Doron Lancet	Rolf Pfeifer
Liaohai Chen	Kristian Lindgren	Vitor Dos Santos
Greg Chirikjian	Jerzy Maselko	Andrew Shreve
David Deamer	John McCaskill	Ricard Solé
Peter Dittrich	Chris Melhuish	Richard Vaughan

The SAC together with the OC proposed a variety of conference themes and the SAC also took part in the multiple conference announcements.

Upon submission, authors were asked to attribute their submission to several of these conference themes. In response to these preliminary assignments, the original themes were slightly revised to more closely match the accepted contributions. A group of 18 track organizers were asked to vote for the contributions potentially pertaining to their themes, and to suggest coherent sessions based on these submissions. This voting was performed using online spreadsheets (Google documents). The Pareto front of the track organizer votes identified few areas of strong overlap – mainly in the area of wet artificial life. For these areas, the session assignment was done jointly by the responsible track organizers. At this stage 113 out of the 152 contributions could be assigned to the unique highest bidder. The remaining 39 submissions with conflicting votes were then assigned in a way that lead to the most consistent sessions. In only five cases, we overruled the bare votes in favor of coherent session themes. However, it should be noted that many contributions fit well within several of these themes due to the interdisciplinary character of the Alife community.

This collective intelligence process resulted in the following themes (with theme organizer names):

Chemical Self-Assembly and Complexity (Jerzy Maselko)
Origin of Life (Mark Dörr & Bruce Damer)
Bottom-up Synthetic Cells (Pierre-Alain Monnard)
Systems Biology (Luis Delaye)
Biological and Chemical Information Processing and Production (John McCaskill)
Artificial Chemistries (Wolfgang Banzhaf)
Minimal Cognition and Physical Intelligence (Martin Hanczyc)
Evolutionary Dynamics (Chris Adami)
Theoretical and Computational Frameworks (Peter Dittrich)
Complex Networks (Carlos Gershensen & Mikhail Prokopenko)
Ecology (Seth Bullock)
Collective Intelligence (Johan Bollen)
Emergent Engineering (Norman Packard)
Intelligence and Learning (Takashi Ikegami)
Robots (Kasper Støy)
Socio-Technical Systems (Kristian Lindgren)
Philosophy (Mark Bedau)

We have tried to organize the sequence of conference topics from lower to higher levels of organization with a variety of methods themes sandwiched in between.

Four keynote presentations – by Christian de Duve, Tetsuya Yomo, John McCaskill, and Serge Kernbach – provide overarching perspectives on the origins of life, artificial cells, the connection between biochemistry and computational hardware and software as well as robotics, covering the classical wet, soft, and hard artificial life research areas. In addition to the invited keynote presentations, Alife XII also features contributed plenary talks. Reviewers, theme organizers and the organizing committee jointly suggested candidates for these presentations. Four plenary, contributed presentations were picked by the organizers to ensure an overall balanced conference program. Unfortunately, many other papers deserving to be highlighted as plenary talks could not be accommodated.

The review process was conducted and coordinated utilizing the distributed online tool EasyChair (`http://www.easychair.org/`), which the organizers can recommend for reviewing many conference paper and abstract submissions. We should stress that the assembly of the conference program would have been impossible without the fantastic work of the 135 Alife XII submission reviewers. The OC is deeply indebted to all of them and they are separately acknowledged on the next pages.

It is our belief that the resulting review process and conference program – a true child of bottom-up collective intelligence – benefited significantly from the participation of the many domain experts. It would have

been very difficult to assemble a theme-based program using a traditional top-down approach. The bottom-up process ensures a program organization, that reflects the highly diverse current activities within the Alife community. The disadvantage of this collective intelligence based program assembly process is that more time and effort is spend by more people.

We, the Alife XII OC, sincerely hope you will find these proceedings both useful and inspirational and that you will enjoy the conference.

Harold Fellermann (Alife XII co-chair)
Mark Dörr
Martin Hanczyc
Lone Ladegaard Laursen (Alife XII administrative chair)
Sarah Maurer
Daniel Merkle (Alife XII EasyChair chair)
Pierre-Alain Monnard
Kasper Støy
Steen Rasmussen (Alife XII chair)

August 2010, Odense, Denmark.

References

[1] Rasmussen, S. et al. editors (2009). *Protocells: Transitions from nonliving to living matter*. MIT Press, Cambridge, MA.

[2] D. Gibson et al. (2010). Creation of a bacterial cell controlled by a chemically synthesized genome *Science*, 329(5987):52–56.

[3] Bedau, M. A. et al. (2010). Life after the synthetic cell *Nature*, 465:422–424.

[4] Zykov, V. et al. (2005). Self-reproducing machines *Nature*, 435(7038):163–164

[5] Ieropoulos, I. et al. (2005). EcoBot-II: An artificial agent with a natural metabolism. *Advanced Robotics Systems* 2(4):295–300.

[6] Bedau, M. A. et al. (2010). Living Technology: Exploiting Life's Principles in Technology. *Artificial Life* 16:89–97.

[7] Bedau, M. A. et al. editors (2010). *Living technology: 5 Questions*. Automatic Press, Copenhagen.

[8] Discussions: "Looking backwards, looking forwards" during the 7th Artificial Life Conference in Posrtland, Oregon, USA. See also Bedau, M. A. et al. editors (2000). *Artificial Life VII: Proceedings of the Seventh International Conference on the Simulation and Synthesis of Living Systems. MIT Press. Cambridge, MA*.

[9] Bedau, M. A. et al. (2000). Open problems in artificial life *Artificial Life* 6:363.

[10] Rasmussen, S. et al. (2003). Collective intelligence of the artificial life community on its own successes, failures, and future. *Artificial Life*, 9:207.

[11] `http://science-society-policy.org/`, see under events, August 24, 2010, "Putting people in the presence".

Program Committee

Andy Adamatzky, Chris Adami, Lee Altenberg, Martyn Amos, Claes Andersson, Takaya Arita, Jacob Axelsen, Carsten Baldauf, Wolfgang Banzhaf, Rafael Barrio, Mark Bedau, Randall Beer, Luis Bettencourt, Tanmoy Bhattacharya, Eleonora Bilotta, Jon Bird, Christian Blum, Johan Bollen, Josh Bongard, Seth Bullock, Pedro Castillo, Sung-Bae Cho, Dominique Chu, Jeff Clune, Stirling Colgate, David Cornforth, Bruce Damer, Thomas Dandekar, Kerstin Dautenhahn, Luis Delaye, Jordi Delgado, Peter Dittrich, Konrad Diwold, Marco Dorigo, Alan Dorin, Mark Dörr, Rene Doursat, Bruce Edmonds, Harold Fellermann, Pietro Speroni di Fenizio, Christoph Flamm, David Fogel, Rudolf Füchslin, Toni Gabaldón, Nic Geard, Philip Gerlee, Carlos Gershenson, Mario Giacobini, Martin Hanczyc, Inman Harvey, Paulien Hogeweg, Paul Humphreys, Phil Husbands, Tim Hutton, Hiroyuki Iizuka, Takashi Ikegami, Christian Jacob, Martin Jacobi, Yi Jiang, George Kampis, Niles Lehman, Tom Lenaerts, Lukas Lichtensteiger, Kristian Lidgren, Torbjörn Lundh, George Magoulas, Jerzy Maselko, Sarah E. Maurer, John McCaskill, Barry McMullin, Juan Merelo, Daniel Merkle, Andrés Moya, Pierre-Alain Monnard, Chrystopher L. Nehaniv, Peter Nielson, Jason Noble, Stefano Nolfi, Charles Ofria, Eckehard Olbrich, Naoki Ono, Ezequiel di Paolo, Simone Pigolotti, Erich Prem, Makhail Prokopenko, Steen Rasmussen, Tom Ray, Craig Reynolds, John Rieffel, Adam Rotaru-Varga, Kepa Ruiz-Mirazo, Vítor dos Santos, Francisco C. Santos, Josep Sardanyés, Hiroki Sayama, Roberto Serra, Ricard V. Solé, Susan Stepney, Kasper Støy, Hideaki Suzuki, Uwe Tangen, Charles Taylor, Tim Taylor, Adrian Thompson, Vito Trianni, Soichiro Tsuda, Elio Tuci, Kolbjorn Tunstrom, Jon Umerez, Edgar Vallejo, Richard Vaughan, Gwenn Volkert, Richard Watson, Niek Wijngaards, Janet Wiles, Jon Wilkins, Larry Yaeger, Lidia Yamamoto, Xin Yao, Tom Ziemke, Hans-Joachim Ziock

Additional Reviewers

Maribel G. Arenas, Joshua Auerbach, Alexandre Campo, Adam Davies, Pablo García-Sánchez, Onofrio Gigliotta, Alex Graudenzi, Arend Hintze, Ben Jones, Wenjian Luo, Thomas Meyer, Rehan O'Grady, Gustavo Romero, Marta Santos, Marco Villani

Table of Contents

Chemical Self-Assembly and Complexity.

Origins of Life.

Systems Biology.

Minimal Synthetic Cells.

Biological & Chemical Information Processing and Production.

Minimal Cognition and Physical Intelligence.

Artificial Chemistries.

Complex Networks.

Theoretical and Computational Frameworks.

Evolutionary Dynamics.

Ecology.

Emergent Engineering.

Intelligence and Learning.

Collective Intelligence.

Robotics.

Socio-Technical Systems.

Philosophy of Artificial Life.

Chemical Self-Assembly and Complexity

Self-Assembly and Self-Construction

Jerzy Maselko

Chemistry Department, University of Alaska Anchorage, 3211 Providence Drive Anchorage, AK 99508
USAafjm1@uaa.alaska.edu

Extended Abstract

The spontaneous increase of complexity in nature from the formation of elements, followed by the formation of compounds, both inorganic and organic, leading to the emergence of life -- from a single cell to multi-cellular organisms -- and the later formation of communities followed by the emergence of new technologies where complex structures are created by a humans is probably the most important property of matter.

One common property of self-construction is the formation of new entities. The formation of elements and chemical compounds are relatively well studied, so the next step is to study the transition from non-living to living matter. This requires formation of complex structures on a scale that begins with nanometers and increases. Most of this is done by "self-assembly," defined as a process that must be completed without external assistance and must include stochastic aggregation of pre-existing components. The formation of more complex structures inside cells and in multi-cellular systems requires a more complex mechanism. Here, the formation of structures requires a complex network of physical and chemical processes that are precisely organized in space and time -- the parts are constantly produced in hierarchy. The stochastic process of movement is replaced by the controlled movement of different parts (components) using different forces and different routes. This process can be seen in the formation of magnets in magnetic bacteria; functioning of xylem and phloem in biological plants; veins, arteries and the lymphatic system in animals; as well as tubes and pumps in industrial plants.

This complex spatio-temporal organization of chemical and physical processes that goes beyond the simple process of self-assembly can also be observed in chemical systems. The construction of complex forms is controlled by the complex network of chemical reactions. These chemical and physical processes may start in a defined place in space and time and be finished in another. This will be discussed in the case of precipitation pattern formation in simple, even two component inorganic systems like, Cu^{2+} - PO_4^{3-}, $Al^{3+,}$ silicate, Cu^{2+} - $C_2O_4^{2}$, Pb^{2+} - chlorite – thiourea, and Fe^{2+} - silicate.

Most of these structures are grown from a chemical seed that is immerged in a chemical solution. The initial study of this seed theory is based on studies of cellular automata and numerical studies of multi-cellular chemical systems development, which will also be presented.

The biological organism evolves forming structures of unbelievable complexity and precision in its construction process and in the functions of its controlling systems.

The emergence of man follows as the next important step in the self-construction of the universe. It has allowed the emergence of new construction technologies that have increased the number of constructed systems and their properties. As predicted by Leonardo da Vinci, we now have the capacity to create technology:

> ***"Where nature finishes producing its species, the man begins with natural things to make with the aid of this nature an infinite number of species."***
>
> ***-Leonardo da Vinci (1452-1519)***

A final important step for discussion regards the construction of computers, allowing for the mathematical modeling and, further, the construction of virtual universes.

Autocatalyses

Raphaël Plasson[1], Axel Brandenburg[1], Ludovic Jullien[2], Hugues Bersini[3]

[1]: NORDITA, Stockholm, Sweden
[2]: ENS, Paris, France
[3]: IRIDIA, ULB, Brussels, Belgium
rplasson@nordita.org

Abstract

The notion of autocatalysis actually covers a large variety of mechanistic realisations of chemical systems. From the most general definition of autocatalysis, that is a process in which a chemical compound is able to catalyze its own formation, several different systems can be described. We detail the different categories of autocatalyses, and compare them on the basis of their mechanistic, kinetic, and dynamic properties. It is proposed that the key signature of autocatalysis is its kinetic pattern expressed in a mathematical form. It will be shown how such a pattern can be generated by different systems of chemical reactions.

Introduction

The notion of "autocatalysis" was introduced by Ostwald in 1890 for describing reactions showing a rate acceleration as a function of time. It is for example the case of esters hydrolysis, that is at the same time acid catalyzed and producing an organic acid (Laidler, 1986). Defined as a chemical reaction that is catalyzed by its own products, it has quickly been described on the basis of a characteristic differential equation (Ostwald, 1902, 1912). Typically used to describe complex behaviors of chemical systems, like oscillatory patterns (Lotka, 1910), it has immediately appeared to be essential for the description of biological systems: growth of individual living beings (Robertson, 1908), population evolution (Lotka, 1920) or gene evolution (Muller, 1922).

Extending this concept from a chemical description to a more open context was initially carefully described as an analogy, sometime qualified by the more general notion of "autocatakinesis" (Lotka, 1925; Witzemann, 1933). However, this eventually leads to an overgeneralization of the term of autocatalysis, tending to be assimilated to the notion of "positive feedback", for example in economy (Malcai et al., 2002).

The notion of autocatalysis is now actively being used for describing self-organizing systems, namely in the field of emergence of life. Autocatalytic processes are the core of the mechanisms leading to the symmetry breaking of chemical compounds towards homochirality (Frank, 1953; Plasson et al., 2007), and could be identified in several experimental systems (Kondepudi et al., 1990; Soai et al., 1995). However, how such autocatalytic processes shall manifest is still under heavy debate (Plasson, 2008; Blackmond, 2009).

The purpose of this article is thus to clarify the meaning of chemical autocatalysis and this effort will be undertaken by covering these following points:

- What is autocatalysis for a chemical system? On the basis of the general description of autocatalysis as a process allowing a chemical compound to enhance the rate of its own formation, it is defined by a kinetic signature, expressed in a mathematical form.
- How can an autocatalytic process be realized? As many mechanisms can reduce to the same macroscopic kinetic laws exhibiting autocatalysis, the focus is put on several mechanistic realisations of autocatalytic processes, on the basis of simple models further illustrated by concrete chemical examples.
- How can autocatalysis be observed and characterized? The focus is put on the dynamic properties, showing that this observable is the direct consequence of the kinetic pattern, rather than the underlying mechanism.
- What is the role of autocatalysis? Embedded in non-equilibrium reaction network, the competition between autocatalytic processes allows the onset of chemical selection, that is the existence of bifurcation phenomena allowing the extinction of some compounds in favor of others.

Autocatalysis: a Practical Definition

A Kinetic Signature

From its origin, the notion of autocatalysis has focused on the kinetic pattern of the chemical evolution (Ostwald, 1902). The general definition of autocatalysis as a chemical process in which one of the products catalyzes its own formation can be mathematically generalized as:

$$\frac{dx_i}{dt} = k(\boldsymbol{X}) \cdot x_i^n + f(\boldsymbol{X}), \quad k > 0;\ n > 0;\ |k| \gg |f| \quad (1)$$

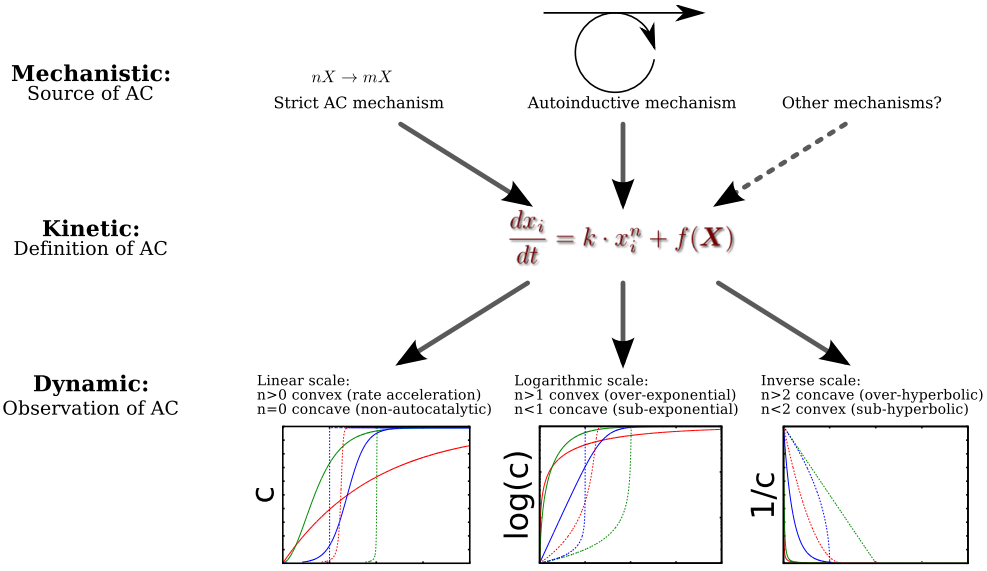


Figure 1: Classification of the concepts of autocatalysis (AC) depending on their descriptions (mechanistic, kinetic, and dynamic). The graphs represents the time evolution of a non-autocatalytic reaction (red), and of autocatalytic reaction of order $1/2$ (green), 1 (blue), $3/2$ (dotted red), 2 (dotted green), and 3 (dotted blue).

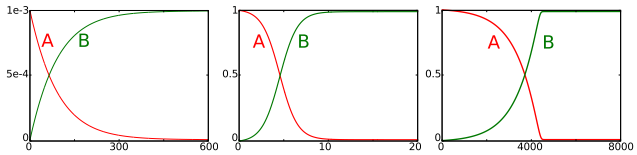
The term $k(\boldsymbol{X}) \cdot x_i^n$ describes the autocatalytic process itself, while $f(\boldsymbol{X})$ describes the sum of all other contributions coming from the rest of the chemical system.

We have an effective practical definition of the concept of autocatalysis, based on a precise mathematical formulation. The causes of this kinetic signature can be investigated, searching what mechanism is responsible for the autocatalytic term. This leads to the discovery of a series of different kinds of autocatalysis processes, and their respective effect, describing what observable behavior is generated by the autocatalytic term (see Fig. 1).

Potential vs Effective Autocatalysis

This kinetic definition is purely structural. As a matter of fact, a system may contain *potential* autocatalysis i.e. an autocatalytic core exists in the reaction network. However, in the absence of some specific conditions necessary for this autocatalysis to be *effective*, the potential autocatalysis may be hidden by other kinetic effects, thus turns out not to manifest its behavior in practice.

Possibly, in Eq. (1), the term $f(\boldsymbol{X})$ may simply overwhelm the autocatalytic process. This is typically the case when an autocatalysis is present together with the non-catalyzed version of the same reaction, that may not be negligible in all conditions. Imagine the simple example of a system simultaneously containing a direct autocatalysis $A + B \rightarrow 2B$, concurrent with the non autocatalytic reaction $A \rightarrow B$. The autocatalytic process follows a bimolecular kinetics, and will be more efficient in a concentrated than in a diluted solution. The dynamic profile of the reaction is thus sigmoidal for high initial concentration of A, but no more for low initial concentration (see Fig. 2(a-b)).



(a) Non-autocatalytic (b) Autocatalytic (c) Undamped AC

Figure 2: (a-b): First order autocatalytic process ($\Gamma_1 = 10^2$ M.s^{-1}) in presence of a non-autocatalytic reaction ($\Gamma_2 = 10^{-2}$ M.s^{-1}) of spontaneous transformation of A into B ($K_A = 1$ M, $K_B = 10^2$ M). (a) Diluted ($a_o = 10^{-3}$ M). (b) Concentrated ($a_0 = 1$ M). (c) Undamped autocatalysis (Indirect autocatalysis, described in Fig. 4(b), $\Gamma_4 = 0.1$ M.s^{-1})

It can also be seen that the term $k(\boldsymbol{X})$ may also vary during the reaction process. In a simple autocatalytic process as describe above, k is proportional to the concentration in A, and is thus more important at the beginning of the reaction (thus an initial exponential increase of the product B) that at the end (thus a damping of the autocatalysis) resulting in a global sigmoidal evolution. In systems were the influence of A on k is weaker, as detailed further, an undamped autocatalysis will be observed characterized by an exponential variation until the very end (see Fig. 2(c)).

Mechanistic Distinctions

How can this kinetic pattern be realized? Let us now detail several types of mechanisms. They can all be reduced,

in some conditions, to the autocatalysis kinetic pattern of Eq. (1). All of them will be equally defined in the paper as autocatalytic, while this status may have been disputed in the past on account of the distinct chemical realisations. In the following, we emphasize the major mechanistic pattern to eventually be reduced to an equivalent kinetic autocatalysis, and discuss where their difference comes from.

Template Autocatalysis

The simplest autocatalysis is obtained by the $X \rightarrow 2X$ pattern. It can be represented by:

$$A + B \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} B + B \tag{2}$$

The corresponding network is given in Fig. 3(a). It can further be decomposed through the introduction of an intermediate compound C:

$$A + B \overset{\Gamma_1}{\rightleftharpoons} C \tag{3}$$

$$C \overset{\Gamma_2}{\rightleftharpoons} B + B \tag{4}$$

The corresponding network is given in Fig. 3(b).

The first mechanism entails the following kinetic evolution:

$$\dot{b} = -\dot{a} = k_1 ab - k_{-1} b^2 \tag{5}$$

This can be expressed as a chemical flux φ, by relying on the Mikulecky formalism (Peusner et al., 1985; Mikulecky, 2001; Plasson and Bersini, 2009):

$$\varphi = \Gamma_1 (V_A V_B - V_B^2) = \Gamma_1 V_B (V_A - V_B) \tag{6}$$

$$V_A = \frac{a}{K_A} \tag{7}$$

$$V_B = \frac{b}{K_B} \tag{8}$$

$$\Gamma_1 = k_1 \cdot K_A K_B = k_{-1} \cdot K_B^2 \tag{9}$$

Formally there is a linear flux φ of transformation of A into B, coupled to a circular flux of same intensity from B back to B (see Fig. 3(a-b)). In presence of an intermediate compound, the equations becomes:

$$\varphi_1 = \Gamma_1 (V_A V_B - V_C) \tag{10}$$

$$\varphi_2 = \Gamma_2 (V_C - V_B^2) \tag{11}$$

Under the hypothesis that C is an unstable intermediate, (i.e. $K_C \ll K_B, K_A$), the variation of C can be neglected compared to the variations of A and B (quasi steady-state approximation, hereafter QSSA), so that:

$$\varphi_1 \simeq \varphi_2 \tag{12}$$

$$= \varphi \tag{13}$$

$$\Rightarrow \quad \varphi = \frac{\Gamma_1 \Gamma_2}{\Gamma_1 + \Gamma_2} (V_A V_B - V_B^2) \tag{14}$$

The system is strictly equivalent to the direct autocatalysis, with an apparent rate $\Gamma_1 \Gamma_2 / (\Gamma_1 + \Gamma_2)$. With these two systems, we are in presence of the perfect kinetic signature of an autocatalytic system i.e. following a sigmoidal evolution (see Fig. 4(a)). This equivalence is guaranteed as long as the compound C remains unstable. When it is not the case, the dimeric intermediate C hardly liberates the final compound B, which gives rise to an autocatalytic process of order $1/2$ rather than 1 (von Kiedrowski, 1993; Wills et al., 1998).

Template autocatalysis requires a direct association between the reactants and the products. This is typically the case of DNA replication, one double strand molecule giving birth to two identical double strand molecules, thanks to the very selective association of complementary nucleotides along each strand. More simple examples can be found in some biological mechanisms that requires autocatalytic processes, for example for the generation of chemical oscillation inducing circadian rhytmicity in cells. The system described by Mehra et al. (2006) is based on a non equilibrium system of association/dissociation of proteins forming a large chemical cycle $[C \rightarrow AC \rightarrow AC^* \rightarrow ABC^* \rightarrow BC^* \rightarrow C^* \rightarrow C]$, maintained by a flux of ATP consumption, one cycle consuming and freeing A and B. The oscillations are generated by coupling this chemical flux to an autocatalytic process of phosphorylation obeying to the reaction scheme: $A + C + AC^* \rightarrow 2AC^*$ (Wang and Wu, 2002).

Network Autocatalysis

The direct mechanism of template autocatalysis just seen is conceptually the simplest framework. It may actually not be the most representative class of autocatalysis, as a similar kinetic signature can appear as resulting from a complex reaction network.

Indirect Autocatalysis: The autocatalytic effect may be only indirect when reactant and products never directly interact:

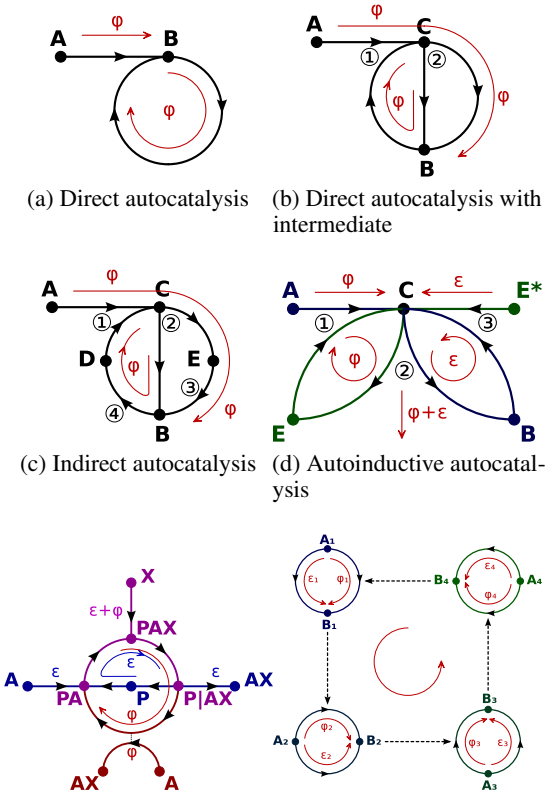
$$A + D \overset{\Gamma_1}{\rightleftharpoons} C \tag{15}$$

$$C \overset{\Gamma_2}{\rightleftharpoons} B + E \tag{16}$$

$$E \overset{\Gamma_3}{\rightleftharpoons} B \tag{17}$$

$$B \overset{\Gamma_4}{\rightleftharpoons} D \tag{18}$$

There is no direct A/B coupling, nor direct $2B$ formation, but the presence of a dimeric compound C. The network decomposition of this system (see Fig. 3(c)) implies once again a linear flux of transformation of A into B, linked to a large cycle of reaction transforming B back to B. Nevertheless, this system is still reducible to an $X \rightarrow 2X$ pattern.

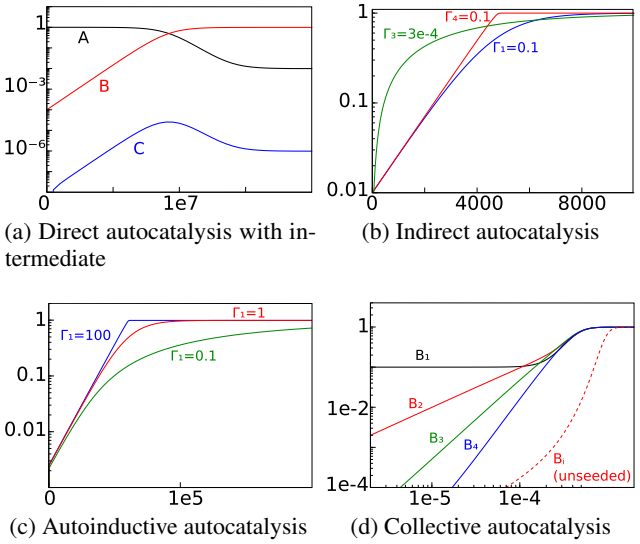


(a) Direct autocatalysis (b) Direct autocatalysis with intermediate

(c) Indirect autocatalysis (d) Autoinductive autocatalysis

(e) Iwamura et al. (2004) system (f) Collective autocatalysis

Figure 3: Reaction network of different autocatalytic processes of spontaneous transformation of A into B (a-d), of $A + X$ into AX (e), and of A_i into B_i (f). The indicated fluxes correspond to what is observed within the QSSA.



(a) Direct autocatalysis with intermediate (b) Indirect autocatalysis

(c) Autoinductive autocatalysis (d) Collective autocatalysis

Figure 4: Time evolution of compound concentrations for different autocatalytic processes of spontaneous transformation of A into B ($K_A = 1$ and $K_B = 100$) in a logarithmic scale for concentrations (a-c), or logarithmic scales for both time and concentrations (d). K and concentrations are in M, times in s, and Γ in M.s^{-1}. (a): Fig. 3(b), $\Gamma_1 = 1$, $\Gamma_2 = 10^{-4}$, $K_C = 0.01$; (b): Fig. 3(c), $\Gamma_1 = \Gamma_2 = \Gamma_3 = \Gamma_4 = 10$ (except the values indicated on the graph), $K_C = K_D = K_E = 0.01$; (c): Fig. 3(d), $\Gamma_2 = \Gamma_3 = 100$, $K_C = K_E = 1$, $K_{E^*} = 10$; (d): Fig. 3(f), $\Gamma_1 = 100$, $\Gamma_2 = 1$.

The QSSA for compounds C, D, E allows to express the reaction flux as:

$$\varphi = \frac{\Gamma_1\Gamma_4}{\Gamma_1 V_A + \Gamma_4} V_A V_B - \epsilon \qquad (19)$$

ϵ express the back-reactions fluxes, and can be neglected as long as Γ_3 is large enough. If it is not the case, the autocatalytic effect is destroyed.

When $\Gamma_1 \ll \Gamma_4$, the system can behave like a simple autocatalytic system, with $\varphi \propto a \cdot b$ before the reaction completion, implying a progressive damping of the exponential growth as long as A is consumed. When $\Gamma_1 \gg \Gamma_4$, the flux is $\varphi \propto b$: the profile remains exponential up to the reaction completion, with no damping due to A consumption (see Fig. 4(b)).

Network autocatalysis is probably the most common kind of mechanisms. A typical biochemical example is the presence of autocatalysis in glycolysis (Ashkenazi and Othmer, 1977; Nielsen et al., 1997). In this system, there is a net balance following the $X \to 2X$ pattern. ATP must be consumed to initiate the degradation of glucose, but much more molecules of ATP are produced during the whole process. While these systems are effectively autocatalytic, there is obviously no possible "templating" effect of one molecule of ATP to generate another one.

Collective Autocatalysis: More general systems, reminiscent of the Eigen's hypercycles (Eigen and Schuster, 1977), are responsible of even more indirect autocatalysis. No compound influence its own formation rate, but rather influences the formation of other compounds, which in turn influence other reactions, in such a way that the whole set of compounds collectively catalyzes its own formation.

A simple framework can be built from the association of several systems of transformation $A_i \to B_i$, each B_i catalyzing the next reaction (see Fig. 3(f)):

$$A_i + B_{i-1} \underset{}{\overset{\Gamma_i}{\rightleftharpoons}} B_i + B_{i-1} \qquad (20)$$
$$i = \{1, 2, 3, 4\}$$

with $B_5 = B_0$ to close the cycle of reactions. There are four independent systems, only connected by catalytic activities.

If the system is totally symmetric, then all b_i are equal, and

all a_i are equal, so that the rates become:

$$\varphi_i = \Gamma_i V_{B_{i-1}}(V_{A_i} - V_{B_i}) \quad (21)$$
$$\varphi = \Gamma V_B(V_A - V_B) \quad (22)$$

This leads to a *collective* autocatalysis with all compounds present. They mutually favor their formation, which results in an exponential growth of each compound (see Fig. 4(d) dotted curve).

With symmetrical initial conditions (i.e. identical for the four systems), the system strictly behaves autocatalytically. If the symmetry is broken, e.g. by seeding only one of the B_i, the system acts with delays. The evolution laws are sub-exponential, of increasing order: At the very beginning of the reaction, considering that A_i do not significantly change and that B_i are in low concentration, we obtain $\varphi_i \propto t^{i-1}$. If seeding with B_1, the compound 2 evolves in t^2. Its impact on compound 3 induces an evolution in t^3. In its turn, the impact of compound 3 on compound 4 induces an evolution in t^4. The compound 1 at first remains constant, and it is only following a given delay that it gets catalyzed by B_4 (see Fig. 4(d)).

This system is actually not characterized by a direct cyclic flux, but by a cycle of fluxes influencing each other and resulting in a cooperative collective effect:

$$\begin{array}{c}(A_1 + A_2 + A_3 + A_4) + (B_1 + B_2 + B_3 + B_4) \\ \longrightarrow \quad 2(B_1 + B_2 + B_3 + B_4)\end{array} \quad (23)$$

The simultaneous presence of all different compounds is needed to observe a first order autocatalytic effect. Given asymmetric initial conditions, a transitory evolution of lower order is first observed, until the formation of the full set of compounds.

A typical example of collective autocatalysis is observed for the replication of viroids (Flores et al., 2004). Each opposite strand of cyclic RNAs can catalyze the formation of the other one, leading to the global growth of the viroid RNA in the infected cell.

Template vs Network Autocatalysis: Nevertheless, all these systems can still be reduced to a $X \rightarrow 2X$ pattern. This is characterized by a linear flux coupled to a loop flux, i.e. for each molecule (or set of molecules) A transformed into B, one B is transformed and goes back to B, following a more or less complex pathways. They can be considered as mechanistically equivalent: a seemingly direct autocatalysis may really be an indirect autocatalysis once its precise mechanism is known, decomposing the global reaction into several elementary reactions.

Practically, autocatalysis will be considered to be direct (or template) when a dimeric complex of the product is formed (i.e. allowing the "imprint" of the product onto the reactant). If such template complex is never formed, we preferentially speak of network autocatalysis, in which the $X \rightarrow 2X$ pattern only results from the reaction balance.

Autoinductive Autocatalysis

Some reactions are not characterized by an $X \rightarrow 2X$ pattern, but still exhibit a mechanism for the enhancement of the reaction rate through the products. This is typically the case for systems where the products increase the reactivity of the reaction catalyst rather than directly influencing their reaction production itself. These systems still possess the kinetic signature of Eq. (1), but are sometime referred as "autoinductive" instead of "autocatalytic" (Blackmond, 2009).

Let us take a simple reaction network of a tranformation $A \rightarrow B$ catalyzed by a compound that can exist under two forms E/E^*, E^* being the more stable one. These two forms of the catalyst interact differently with the product B (see Fig. 3(d)):

$$A + E \underset{}{\overset{\Gamma_1}{\rightleftharpoons}} C \quad (24)$$
$$C \overset{\Gamma_2}{\rightleftharpoons} B + E \quad (25)$$
$$C \overset{\Gamma_3}{\rightleftharpoons} B + E^* \quad (26)$$

There is no dimeric compound in the system, even indirectly formed.

Provided the catalyst, present in C, E, E^*, is in low total concentration, the QSSA implies the presence of two fluxes: the transformation of A into B catalyzed by E of intensity φ, and the transformation of E^* into E catalyzed by B of intensity ε, with $\varphi \gg \varepsilon$. This decomposition gives:

$$\varphi = \frac{\alpha V_A V_B}{\beta V_B + \gamma} - \delta V_B \quad (27)$$

with $\alpha = \delta(\Gamma_1 + \Gamma_2)$, $\beta = \Gamma_2 - \Gamma_1 \frac{K_B}{K_A}$, $\gamma = \frac{\Gamma_1}{V_A^{\text{tot}}}$ and $\delta = \frac{\Gamma_1}{V_{E^*}^{\text{tot}}}$.

The autoinduction is kinetically equivalent to the indirect autocatalysis mechanism:

- When $\Gamma_2 \gg \Gamma_1 \frac{K_B}{K_A}$, the flux tends to $\varphi = \frac{\alpha}{\beta} V_A - \delta V_B$: the system is non-autocatalytic.
- When $\Gamma_2 \approx \Gamma_1 \frac{K_B}{K_A}$, the flux tends to $\varphi = \frac{\alpha}{\gamma} V_A V_B - \delta V_B$: the system is simply autocatalytic.
- When $\Gamma_2 \ll \Gamma_1 \frac{K_B}{K_A}$, the flux tends to $\varphi = \frac{\alpha}{\Gamma_1} V_B - \delta V_B$: the system presents an undamped autocatalysis.

Following the kinetic analysis, the behavior is similar to the time evolution of autocatalytic systems (See Fig. 4(c)). The behavioral equivalence of these two systems (kinetically equivalent but mechanistically very different) will be investigated in more details in the next section.

The mechanism of Iwamura et al. (2004) is an autoinductive autocatalysis, with a slightly more complex mechanism (see Fig. 3(e)). The core principle is a reaction $A+X \rightarrow AX$,

catalyzed by P, the product AX catalyzing the first catalytic step $P+A \rightarrow PA$. This chemical system can be decomposed into two different fluxes $A + X \rightarrow AX$, one coupled to a catalytic cycle $[P \rightarrow PA \rightarrow PAX \rightarrow P|AX \rightarrow P]$, and one coupled to a catalytic cycle $[PA \rightarrow PAX \rightarrow P|AX \rightarrow PA]$. The first one contains the slow reaction of A on P, and corresponds to a slow flux ε. The second one only contains fast reactions, and corresponds to a fast flux φ. These two fluxes can be shown to be related by:

$$\frac{\varphi}{\varepsilon} = \alpha V_A + \beta V_{AX} \quad (28)$$

α and β being constants depending on the kinetic parameters of the system. This implies an increase of the effective rate production φ as a function of the concentration in product.

Network vs Autoinductive Autocatalysis: Autoinductive autocatalysis is mechanistically different from network or template autocatalysis. The balance equation is rather of the form $A + \alpha B \rightarrow (1 + \alpha)B$, with $\alpha \ll 1$. The linear transformation $A \rightarrow B$ is only weakly coupled to the cycle of B back to itself, this latter one being subject to a much lower flux than the linear flux. However, autoinduction is kinetically and dynamically equivalent to network autocatalysis, leading to the same kind of differential equation, and thus of behavior. It can be noted that the undamped exponential profile due to a flux only proportional to the products and not to the reactant is not characteristic of autoinductive processes (Iwamura et al., 2004) but can also be explained by network autocatalytic mechanisms, when the consumption of the reactant is not limiting the kinetic of the network.

Embedded Autocatalyses

Autocatalysis is not so important *per se* but as a way of giving birth to rich non-linear behaviors like bifurcation, multistability or chemical oscillations. It becomes capital to study the interaction of autocatalytic mechanisms and their ability to generate such behaviors when embedded in a larger chemical network.

Dynamical Distinctions

Different behaviors depending on the order n of the autocatalysis can be observed in biochemical competitive systems. They are classically studied in population evolution (Szathmáry, 1991; Nowak, 2006) and described as "survival of the all" in the case of $0 < n < 1$ (characterized by the coexistence of all compounds), as "survival of the fittest" in the case of $n = 1$ (when the only stable solution retains the fittest compound or the most "reproductible") and as "survival of the first" in the case of $n > 1$ (when the final solution just retains the product initially present in the highest concentration).

The case $0 < n < 1$ is the least interesting, as it hardly leads to a clear selectionnist process. However, real mechanism that seems to possess a first order autocatalysis may actually present a lower autocatalytic order. This is typically the case for direct template autocatalysis, in which the order falls to $1/2$ on account of the high stability of the dimeric intermediate—which is actually a necessary condition for the selectivity of template replication (von Kiedrowski, 1986, 1993; Wills et al., 1998). This turns out to be a fundamental problem for understanding the emergence of the first replicative molecules (Szathmáry and Gladkih, 1989; Lifson and Lifson, 1999; Scheuring and Szathmáry, 2001).

More complex mechanisms may lead to higher orders, typically by the formation of dimeric autocatalysts (Wagner and Ashkenasy, 2009). This is the case of the Soai reaction whose high sensitivity to initial conditions may potentially be explained by the formation of trimeric (Gridnev et al., 2003) or even hexameric complexes (Schiaffino and Ercolani, 2008).

Comparative Efficiency of Direct and Autoinductive Autocatalyses

Bifurcations appear when installing two autocatalytic processes in competition, placing them in a non-equilibrium open-flow system, both being fed by the same incoming compound and with cross-inhibition between them:

$$\rightarrow A \quad \text{(incoming flux)} \quad (29)$$

$$A \rightleftharpoons B_1 \quad \text{(Direct AC)} \quad (30)$$

$$A \rightleftharpoons B_2 \quad \text{(Autoinduced AC)} \quad (31)$$

$$B_1 + B_2 \rightarrow (P) \quad \text{(cross inhibition)} \quad (32)$$

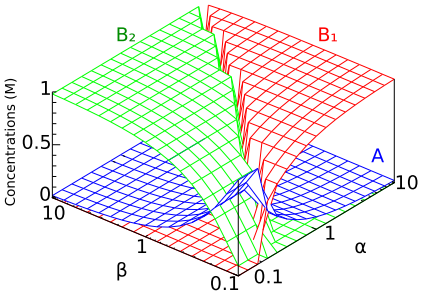
$$B_1 \rightarrow \quad \text{(outgoing flux)} \quad (33)$$

$$B_2 \rightarrow \quad \text{(outgoing flux)} \quad (34)$$

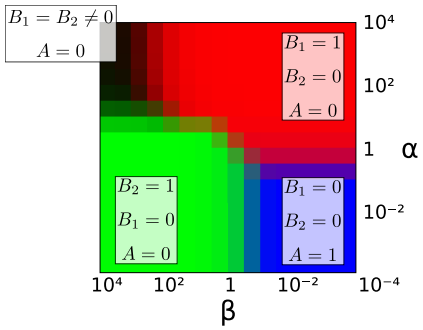
In the case of total symmetry between B_1 and B_2, with the same direct autocatalystic mechanism, this system would correspond to the classical Frank model for the emergence of homochirality (Frank, 1953), leading to a the same probability to end up with either B_1 or B_2.

The kinetic equivalence between template autocatalysis and autoinductive autocatalysis can be shown by making these two mechanisms to compete, replacing Eq. (30) and (31) by the corresponding mechanism. Kinetic parameters have first been normalized so that both reaction leads to the same kinetic behavior (sigmoidal evolution, half-reaction at 10^5 s), and then multiplied by respectively α and β parameters in order to tune the respective velocity of each mechanism. The result is actually quite symmetrical between the two processes and only the fastest product is maintained in the system: B_1 when $\alpha > \beta$, and B_2 when $\alpha < \beta$ (see Fig. 5(a)).

This selection is independent of the relative stability of B_1 and B_2, but is only possible for kinetics that are well adapted to the global influx of matter. For slow kinetics, there is a flush of the system, and no B_1 nor B_2 compound can be maintained. For fast kinetics, the system is close to equilibrium,



(a) Sharp bifurcation depending on the relative values of α and β for moderate reactivities.



(b) Different zones of behaviors: majority of A for $\alpha, \beta \ll 1$, majority of B_1 for $\alpha > \beta$, majority of B_2 for $\alpha < \beta$, and coexistence of B_1 and B_2 for $\alpha, \beta \gg 1$.

Figure 5: Competition between template and autoinductive autocatalysis, generating respectively B_1 and B_2 compounds from the same A compound. Incoming flux of A, and outgoing fluxes of B_1 and B_2, 10^{-5} M.s^{-1}. $K_A = 1$, $K_{B_1} = K_{B_2} = 100$. Direct autocatalysis: $\Gamma_{AC} = 10^{-2} \cdot \alpha$, $\Gamma_{NC} = 10^{-6} \cdot \alpha$. Autoinduction, according to Fig. 3(d): $\Gamma_1 = \beta$, $\Gamma_2 = \Gamma_3 = 100 \cdot \beta$, $K_C = K_E = 1$; $K_{E^*} = 10$.

the compounds B_1 and B_2 being both present in proportion to their respective stability (see Fig. 5(b)). Such result is well known for open flow Frank systems (Cruz et al., 2008).

From Autocatalytic Processes towards Autocatalytic Sets

These competitive systems are able to dynamically maintain a set of components, to the detriment of others. The notion of autocatalytic set (requiring the system to be materially closed and self-maintained by a crossing energetical flux) is rather popular in the artificial life literature and relies much more on the cooperation between autocatalytic mechanisms than on the competition that has just been detailed here. It implies a notion of closure of the system and of self maintenance of the whole network (Kauffman, 1986; Hordijk and Steel, 2004; Benkö et al., 2009). Confusion among these different phenomena can be pinpointed in the literature (Blackmond, 2009), when the failure of autoinductive sets to be maintained do not originate from a difference of behavior between autocatalytic and autoinductive mechanisms, but from a defect in the closure of the system.

Conclusion

Important distinctions need to be done between mechanistic and dynamic aspects of autocatalysis. The same mechanisms can produce different dynamics, while identical dynamics can originate from different mechanisms. But all these different autocatalytic processes are able to generate autocatalytic kinetics, that may constitute a pathways towards the onset of "self-sustaining autocatalytic sets", as a chemical attractor in non-equilibrium networks. However, the problem of the evolvability of such systems must be kept in mind (Vasas et al., 2010). If a system evolves towards a stable attractor, no evolution turns out to be possible. There is the necessity of "open-ended" evolution (Ruiz-Mirazo, 2007) i.e. the possibility of a dynamic set not only to maintain itself (i.e. a strictly autocatalytic system) but act as a "general autocatalytic set", redounding upon the concept originally introduced by Muller (1922) for the autocatalytic power linked to mutability of genes. Insights can be gained by a deeper and renewed study of the evolution of prions as a simple mechanism of mutable autocatalytic systems (Li et al., 2010).

Acknowledgements

This work was done within the scope of the European program COST "System Chemistry" CM0703. We additionally thank R. Pascal for useful discussions.

References

Ashkenazi, M. and Othmer, H. G. (1977). Spatial patterns in coupled biochemical oscillators. *J. Math. Biol.*, 5(4):305–350.

Benkö, G., Centler, F., Dittrich, P., Flamm, C., Stadler, B. M. R., and Stadler, P. F. (2009). A topological approach to chemical organizations. *Artificial Life*, 15(1):71–88.

Blackmond, D. (2009). An examination of the role of autocatalytic cycles in the chemistry of proposed primordial reactions. *Angew. Chem.*, 48(2).

Cruz, J. M., Parmananda, P., and Buhse, T. (2008). Noise-induced enantioselection in chiral autocatalysis. *J. Phys. Chem. A*, 112:1673–1676.

Eigen, M. and Schuster, P. (1977). The hypercycle, a principle of natural self-organization. *Naturwiss.*, 64:541–565.

Flores, R., Delgado, S., Gas, M.-E., Carbonell, A., Molina, D., Gago, S., and la Peña, M. D. (2004). Viroids: the minimal non-coding RNAs with autonomous replication. *FEBS Lett.*, 567(1):42 – 48.

Frank, F. C. (1953). Spontaneous asymmetric synthesis. *Biochem. Biophys. Acta*, 11:459–463.

Gridnev, I., Serafimov, J., Quiney, H., and Brown, J. (2003). Reflections on spontaneous asymmetric synthesis by amplifying autocatalysis. *Org. Biomol. Chem.*, 1(21):3811–3819.

Hordijk, M. and Steel, M. (2004). Detecting autocatalytic, self-sustaining sets in chemical reaction systems. *J. Theor. Biol.*, 227(4):451–461.

Iwamura, H., Wells, D. H., Mathew, S. P., Klussmann, M., Armstrong, A., and Blackmond, D. G. (2004). Probing the active catalyst in product-accelerated proline-mediated reactions. *J. Am. Chem. Soc.*, 126(50):16312–16313.

Kauffman, S. A. (1986). Autocatalytic sets of proteins. *J. Theor. Biol.*, 119:1–24.

Kondepudi, D. K., Kaufman, R. J., and Singh, N. (1990). Chiral symmetry breaking in sodium chlorate crystallization. *Science*, 250(4983):975–976.

Laidler, K. J. (1986). The development of theories of catalysis. *Arch. Hist. Exact Sci.*, 35(4):345–374.

Li, J., Browning, S., Mahal, S. P., Oelschlegel, A. M., and Weissmann, C. (2010). Darwinian evolution of prions in cell culture. *Science*, 327(5967):869–872.

Lifson, S. and Lifson, H. (1999). A model of prebiotic replication: Survival of the fittest versus extinction of the unfittest. *J. Theor. Biol.*, 199(4):425 – 433.

Lotka, A. J. (1910). Contribution to the theory of periodic reactions. *J. Phys. Chem.*, 14(3):271–274.

Lotka, A. J. (1920). Analytical note on certain rhythmic relations in organic systems. *Proc. Natl. Acad. Sci. USA*, 6(7):410.

Lotka, A. J. (1925). *Elements of physical biology*. Williams & Wilkins company.

Malcai, O., Biham, O., Richmond, P., and Solomon, S. (2002). Theoretical analysis and simulations of the generalized Lotka-Volterra model. *Phys. Rev. E*, 66(3):031102.

Mehra, A., Hong, C. I., Shi, M., Loros, J. J., Dunlap, J. C., and Ruoff, P. (2006). Circadian rhythmicity by autocatalysis. *PLoS Comput. Biol.*, 2(7):e96.

Mikulecky, D. C. (2001). Network thermodynamics and complexity: a transition to relational systems theory. *Comp. Chem.*, 25:369–391.

Muller, H. J. (1922). Variation due to change in the individual gene. *Amer. Naturalist*, 56(642):32–50.

Nielsen, K., Sørensen, P. G., and Hynne, F. (1997). Chaos in glycolysis. *J. Theor. Biol.*, 186(3):303 – 306.

Nowak, M. A. (2006). *Evolutionary Dynamics: exploring the equations of life*. The Belknap Press of Harvard University Press.

Ostwald, W. (1890). Über autokatalyse. *Ber. Verh. Kgl. Sächs. Ges. Wiss. Leipzig, Math.- Phys. Classe*, 42:189–191.

Ostwald, W. (1902). *Lehrbuch der Allgemeinen Chemie*, page 263. Engelmann, Leipsic, 2nd edition.

Ostwald, W. (1912). *Outlines of general chemistry (trad. Taylor, W.W.)*, chapter XI.1, page 301. Macmillan and co.

Peusner, L., Mikulecky, D. C., Bunow, B., and Caplan, S. R. (1985). A network thermodynamic approach to hill and king-altman reaction-diffusion kinetics. *J. Chem. Phys.*, 83(11):5559–5566.

Plasson, R. (2008). Comment on "re-examination of reversibility in reaction models for the spontaneous emergence of homochirality". *J. Phys. Chem. B*, 112(31):9550–9552.

Plasson, R. and Bersini, H. (2009). Energetic and entropic analysis of mirror symmetry breaking processes in a recycled microreversible chemical system. *J. Phys. Chem. B*, 113(11):3477–3490.

Plasson, R., Kondepudi, D. K., Bersini, H., Commeyras, A., and Asakura, K. (2007). Emergence of homochirality in far-from-equilibrium systems: mechanisms and role in prebiotic chemistry. *Chirality*, 19(8):589–600.

Robertson, T. (1908). Further remarks on the normal rate of growth of an individual, and its biochemical significance. *Devel. Genes Evol.*, 26(1):108–118.

Ruiz-Mirazo, K. (2007). Enabling conditions for 'open-ended evolution'. *Biol. Philos.*, 23(1):67.

Scheuring, I. and Szathmáry, E. (2001). Survival of replicators with parabolic growth tendency and exponential decay. *J. Theor. Biol.*, 212(1):99 – 105.

Schiaffino, L. and Ercolani, G. (2008). Unraveling the mechanism of the soai asymmetric autocatalytic reaction by first-principles calculations: Induction and amplification of chirality by self-assembly of hexamolecular complexes. *Angew. Chem.*, 120(36):6938–6941.

Soai, K., Shibata, T., Morioka, H., and Choji, K. (1995). Asymmetric autocatalysis and amplification of enantiomeric excess of a chiral molecule. *Nature*, 378:767–768.

Szathmáry, E. (1991). Simple growth laws and selection consequences. *Trends Ecol. Evol.*, 6(11):366 – 370.

Szathmáry, E. and Gladkih, I. (1989). Sub-exponential growth and coexistence of non-enzymatically replicating templates. *J. Theor. Biol.*, 138(1):55 – 58.

Vasas, V., Szathmáry, E., and Santos, M. (2010). Lack of evolvability in self-sustaining autocatalytic networks constraints metabolism-first scenarios for the origin of life. *Proc. Natl. Acad. Sci. USA*, 107(4):1470–1475.

von Kiedrowski, G. (1986). A self-replicating hexadeoxynucleotide. *Angew. Chem.*, 25(10):932–935.

von Kiedrowski, G. (1993). Minimal replicator theory I: Parabolic versus exponential growth. *Bioorg. Chem. Front.*, 3:115–146.

Wagner, N. and Ashkenasy, G. (2009). Symmetry and order in systems chemistry. *J. Chem. Phys.*, 130(16):164907.

Wang, Z.-X. and Wu, J.-W. (2002). Autophosphorylation kinetics of protein kinases. *Biochem. J.*, 368(3):947–952.

Wills, P. R., Kauffman, S. A., Stadler, B. M. R., and Stadler, P. F. (1998). Selection dynamics in autocatalytic systems: Templates replicating through binary ligation. *Bull. Math. Biol.*, 60(6):1073 – 1098.

Witzemann, E. J. (1933). Mutation and adaptation as component parts of a universal principle: II. the autocatalysis curve. *Amer. Naturalist*, 67(710):264–275.

Spontaneous Assembly of Cell-Like Structures from Likely Prebiotic Materials: Problems and Prospects

H. James Cleaves II[1]

[1] Geophysical Laboratory, the Carnegie Institution of Washington
hjcleaves@ciw.edu

Extended Abstract

It has been suggested by a number of theoreticians that cellularity is a precondition for a living system. Over the years many researchers have sought to synthesize structures morphologically resembling cells under prebiotic conditions. These structures may be vesicular or contain no lipid and are perhaps best termed "cell-like structures" than "proto-cells" or "cells". Conversely, likely prebiotic organic amphiphiles such as fatty acids only produce micelles or vesicles under select conditions: high ionic strength and divalent cations often inhibit the self-assembly of cell-like structures assembled from lipid amphiphiles such as vesicles.

Hydrogen cyanide (HCN) is a ubiquitous compound in young circumstellar disks (Carr & Najita, 2008) and cometary comae (Irvine *et al.*, 1997), and is readily produced in simulations of prebiotic atmospheric chemistry (Miller, 1957). During investigations of the chemistry of self-condensation of aqueous HCN in the presence of aldehydes we have discovered cell-like spherical and filamentous structures of extremely homogeneous size distribution which are produced robustly from these simple reactions (Figure 1). While there is some precedent for these structures (see for example Labadie *et al.*, 1968; Kenyon & Nissenbaum, 1976), the chemical and morphological structure of these and their interactions with amphiphilic species have been investigated in considerably more detail here. These are potentially important as scaffolds for cellular development on the primitive Earth, and may have implications for life-detection on other planets and in the geological record.

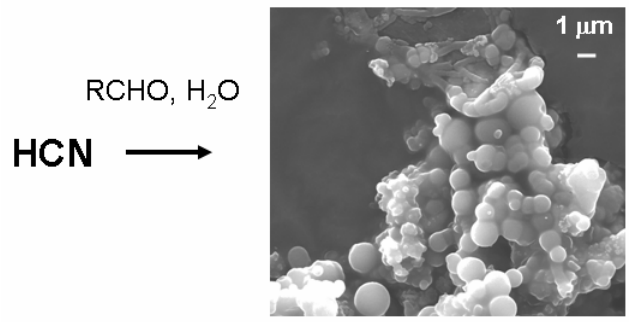


Figure 1. Spherical and filamentous structures formed from the reaction of aqueous HCN and aldehydes.

References

Carr, J.S. and Najita, J.R. (2008) Organic Molecules and Water in the Planet Formation Region of Young Circumstellar Disks. *Science*, 319:1504-1506.

Irvine, W.M., Dickens, J.E., Lovell, A.J., Schloerb, F.P., Senay, M., Bergin, E.A., Jewitt, D. and Matthews, H.E. (1997) The HNC/HCN Ratio in Comets. *Earth, Moon, and Planets*, 78:29-35.

Kenyon, D. H. and Nissenbaum, A. (1976) Melanoidin and Aldocyanoin Microspheres: Implications for Chemical Evolution and Early Precambrian Micropaleontology. *J. Mol .Evol.*, 7:245-251.

Labadie, M., Jensen, R. and Neuzil, E. (1968) Recherches sur l'Évolution Prébiologique III. Les Acides Azulmiques Noirs Formés à Partir du Cyanure d'Ammonium. *Biochim. Biophys. Acta,* 165:525-533.

Miller, S.L. (1957) The Mechanism of Synthesis of Amino Acids by Electric Discharges. *Biochim. Biophys. Acta*, 23:480-489.

Possible Role of Ice in the Synthesis of Polymeric Compounds

Mark Dörr[1], Philipp M.G. Löffler[1] and Pierre-Alain Monnard[1]

[1]Institute for Physics and Chemistry/FLinT; University of Southern Denmark, Odense, Denmark
markd@ifk.sdu.dk

Extended Abstract

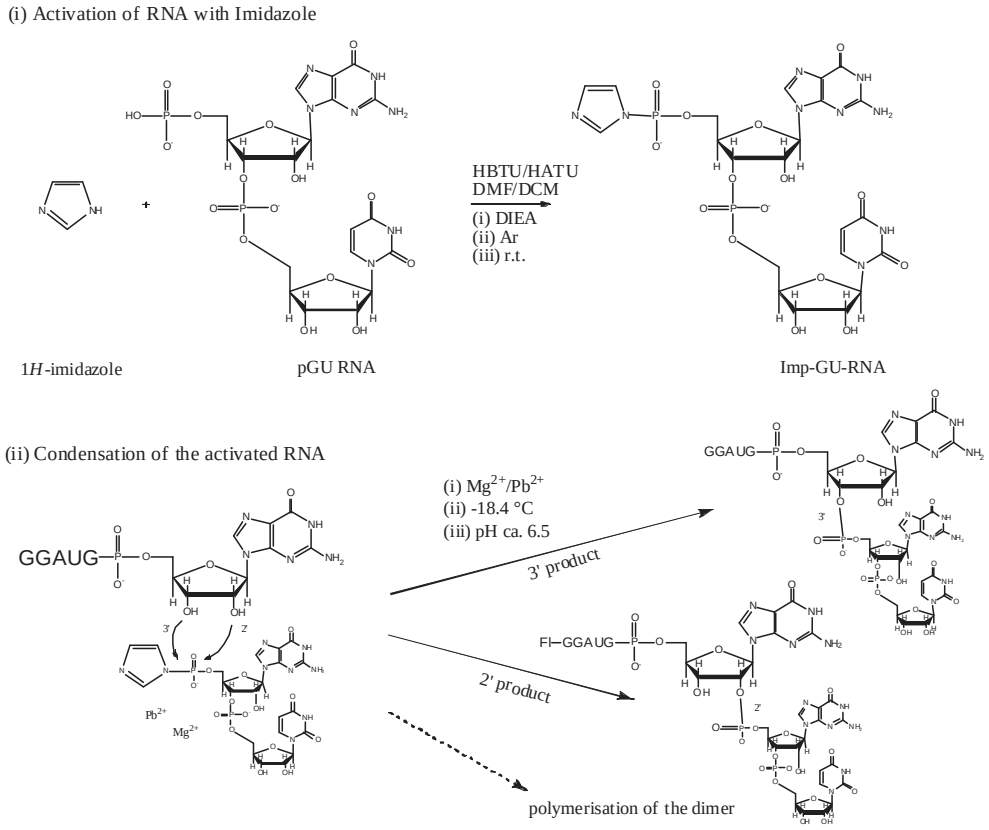


Figure 1: (i) Activation of RNA with imidazole. (ii) RNA condensation reactions.

Cellular life relies on a collection of linear polymers (among them DNA, RNA, proteins) to perform the functions necessary to its survival. It seems likely that catalytic and informational polymers played essential roles in the emergence of the first living entities, precursors of contemporary cells. Thus, their detection on other planetary bodies might hint at either emerging, or extant, or past life in these environments.

A non-enzymatic synthesis of such polymeric materials or their precursors likely had to rely on a supply of monomers dissolved at low concentrations in an aqueous medium. An aqueous environment represents a clear hurdle to the synthesis of long polymers as it tends to inhibit polymerization due to entropic effects and favors the reverse reaction (decomposition by hydrolysis). It was therefore proposed that polymerization could occur in a distinct micro- or nanostructured environment that would permit a local increase in the monomer concentration, reduce water activity and protect monomers and polymers from hydrolysis. Several types of micro- or nanostructured environments, among them mineral surfaces [1], lattices of organic molecules, such as amphiphile bilayer structures [2], and the eutectic phase in water-ice [3, 4, add 2008 JIB, 2008 Chem. Biodiv] have been proposed to promote RNA

and peptide formation. This last environment might be of particular interest since space exploration has established that water exists on Mars, Europa, Enceladus and comets, mostly as ice. Ice deposits may also have existed on the early Earth.
When an aqueous solution is cooled below its freezing point, but above the eutectic point, two aqueous phases co-exist and form the eutectic phase system: a solid (the ice crystals made of pure water) and a liquid phase containing most solutes. The role of water likely extends beyond that of a simple chemical liquid medium since the surfaces of ice crystals could act as a substrate on which other reactants can attach and/or become aligned.
The emergence of a polymer-based genetic or/and catalytic system, as it for example the "RNA World hypothesis" states, initially requires the synthesis of monomers followed by three non-enzymatic processes: polymerization of monomers; elongation of existing polymers with monomers or short oligomers; and replication of existing polymers in a template-directed fashion. Ideally, these processes should take place efficiently, using simple metal ions as catalysts. However, in a dilute solution, even when using activated monomers, these chemical processes occur very slowly, if at all.
We have been exploring the plausibility of chemical reactions, such as non-enzymatic nucleotide condensations forming RNA, under cold environmental conditions and found that the polymerization of RNA from imidazole-activated ribonucleotides (s. Fig. 1) can proceed efficiently in the eutectic phase in water-ice when metal ions are available as catalysts [4]. Starting from monomer mixtures, polymers up to 30 monomeric units in length can be readily formed [5]. Longer polymers can be obtained by adding freshly activated monomers or short oligomers to a solution over several freeze-thawing cycles. Depending on their sequences, oligomers can be elongated using monomers to obtain up to a 45-mer. Furthermore, the decomposition of the longer chains remained low. By using activated short oligomers, even longer polymers can be formed [6].
Studying template-directed RNA polymerization under these conditions, we discovered that the initial elongation rates depended on the complementarity of the monomers with the templating nucleobases. That means that the polymerization rates for all four nucleobases pairing with their corresponding Watson-Crick nucleobase were higher than in cases where hydrogen bond based pairing is not favoured [7]- this was even the found for low H-bridging uridine monomers [7, 8]. The presence of templates further allows the synthesis of long complementary strands [9]. Thus, template-directed elongation of RNA in the eutectic phase of the water-ice system seems possible.
Recently, Miller's group [10, 11] in San Diego further established that dilute solutions of ammonium cyanide maintained frozen at -78 ˚C could promote the synthesis of nucleobases, although with rather low yields. The catalytic activity of a ligase was also detected in the eutectic phase [12].
All the observations on the promotion of synthetic reactions in the eutectic phase in water-ice suggest that the cold conditions with transient thawing periods could have allowed the formation of RNA monomers on our Earth and possibly on other planets.

References

[1] Ferris, J. P. (2006). *Phil. Trans. R. Soc. B*, 361: 1777.
[2] Rajamani, S., Vlassov, A.; Coombs, A.; F., O.; Deamer, D. W. *Orig Life Evol Biosph*, accepted.
[3] Bada, J. L., Bigham, C., and Miller S. L. (1994). *Proc. Nat. Acad Sci USA*, 91: 1248.
[4] Kanavarioti, A., Monnard, P.-A., and Deamer, D. W. (2001). *Astrobiology*, 1: 271.
[5] Monnard, P.-A., Kanavarioti, A., and Deamer, D. W. (2003). *J. Am. Chem. Soc.*, 125: 13734.
[6] Dörr, M., and Monnard, P.-A. *in preparation*.
[7] Monnard, P.-A., and Szostak, J. W. (2008). *J. Inorg. Biochem.*, 112: 1104.
[8] Vogel, S. R., and Richert, C. (2007). *Chem Commun (Camb)*: 1896.
[9] Trinks, H., Schroder, W., and Biebricher, C. K. (2005). *Orig Life Evol Biosph*, 35: 429.
[10] Miyakawa, S., Cleaves, H. J., and Miller, S. L. (2002). *Orig. Life Evol Biosphere*, 32: 195.
[11] Miyakawa, S., Cleaves, H. J., and Miller, S. L. (2002). *Orig. Life Evol Biosphere*, 32: 209.
[12] Vlassov, A., Johnston, B. H., Landweber, L. F., and Kazakov, S. A. (2004). *Nucl. Acids. Res.*, 32: 2966

Light Induced Replication and Selection in Peptide Networks

Gonen Ashkenasy, Zehavit Dadon and Nathaniel Wagner

Department of Chemistry, Ben Gurion University of the Negev, Beer Sheva, Israel
gonenash@bgu.ac.il

Extended Abstract

Complex-systems research has received a lot of attention in mathematics, physics, and biology, but until not too long ago was significantly underdeveloped in chemistry. Recently, it has been realized that while cell biochemistry is a natural model for studying functional networks, rationally designed self-organized synthetic networks might also provide useful models for understanding and exploitation of complex systems' behavior [1]. Thus, several relatively complex networks were studied, and it was found possible to predict and analyze their connectivity and global topology [2]. Moreover, the networks could also be manipulated in various ways to show that just like the cellular networks, their rewiring following changes in the environmental conditions is substantial, and that they can carry out chemical transformations via various complex pathways, such as the Boolean logic operations [3,4].

An important family of the studied non-enzymatic systems uses template directed autocatalysis and cross catalysis as a means of wiring the network components and controlling their dynamics and replication. As such, these networks have also received considerable attention with respect to possible scenarios in the origins of life and early molecular evolution. Several approaches have been taken to manipulate the systems studied so far, based on chemical changes that can affect the replication efficiency. The ability to test and control the response of non-enzymatic networks to external signals might increase significantly their utility and applicability. Such triggering can be used to shift the self-organization states away from equilibrium and thus may provide temporal control over the progress of the chemical (replication) reactions and the entire network topology. To the best of our knowledge, this challenge has not yet been met. We will describe in this presentation the use of light as an external trigger for quantitative control of peptide tertiary structures and consequently as a tool for controlling peptide based self-replication, thereby affecting replication-dependent processes in small molecular networks and facilitating selective and programmable product formation via the AND Boolean function.

References

[1] Z. Dadon, N. Wagner, G. Ashkenasy "The Road to Non-enzymatic Molecular Networks", *Angew. Chem. Int. Ed.* **2008**, 47, 6128 – 6136.
[2] G. Ashkenasy, R. Jagasia, M. Yadav, M.R. Ghadiri “Design of a Directed Molecular Network” *Proc. Natl. Acad. Sci. U.S.A*. **2004**, 101, 10872-10877.
[3] G. Ashkenasy, M.R. Ghadiri “Boolean Logic Functions of a Synthetic Peptide Network”, *J. Am. Chem. Soc.* **2004**, 126, 11140-11141.
[4] N. Wagner, G. Ashkenasy "Systems Chemistry: Logic Gates, Arithmetic Units and Network Motifs in Small Networks" *Chem. Eur. J.* **2009**, 15, 1765 – 1775.

Devo Co-evolution of Shape and Metabolism for an Artificial Organ

Alessandro Fontana[1]

[1]IEEE
alessandro.fontana@ieee.org

Abstract

"Epigenetic Tracking" is an evo-devo method to generate arbitrary 2d or 3d shapes; as such, it belongs to the field of "artificial embryology". *In silico* experiments have proved the effectiveness of the method in devo-evolving any kind of shape, of any complexity (in terms of number of cells, number of colours, etc.); being shape complexity a metaphor for organismal complexity, such simulations established its potential to generate the complexity typical of biological systems. Furthermore, it has also been shown how the underlying model of development is able to produce the artificial version of key biological phenomena such as embryogenesis, the presence of "junk DNA", the phenomenon of ageing and the process of carcinogenesis. In this paper the model is enriched by adding computational capabilities to cells (besides spatial position and colour); the cells endowed with such properties constitute the nodes of an artificial "metabolic network", able to exchange signals and to process the equivalent of chemical substances. The potential of the extended model is evaluated in a computer simulation aimed at "devo co-evolving" shape and metabolism for an artificial organ.

Introduction

The previous work in the field of Artificial Embryology (see (Stanley and Miikkulainen, 2003) for a comprehensive review) can be divided into two broad categories: the grammatical approach and the cell chemistry approach. In the grammatical approach development is guided by sets of grammatical rewrite rules; context-free or context-sensitive grammars, instruction trees or directed graphs can be used; L-systems were first introduced by Lindenmayer (Lindenmayer, 1968) to describe the complex fractal patterns observed in the structure of trees. The cell chemistry approach draws inspiration from the early work of Turing (Turing, 1952), who introduced reaction and diffusion equations to explain the striped patterns observed in nature (e.g. shells and animals' fur); this approach attempts to simulating cell biology at a deeper level, going inside cells and reconstructing the dynamics of chemical reactions and the networks of chemical signals exchanged between cells. Notable examples of grammatical embryogenies are (Gruau et al., 1996), (De Garis, 1999) and (Hornby and Pollack, 2002); among cell chemistry embryogenies, we recall (Kauffman, 1969) and, more recently, (Miller and Banzhaf, 2003), (Joachimczak and Wrobel, 2008) and (Doursat, 2008).

"Epigenetic Tracking" the name of an embryogeny applied to morphogenesis, i.e. the task of generating arbitrary 2d or 3d shapes, described in (Fontana, 2008). From this initial work, two lines of research are possible. One tries to make use of the method as a general-purpose tool to solving real-world problems; the second line of research tries to bridge the gap between the model and real biology. This second line was pursued in (Fontana, 2009) (a work that explored the model's biological implications) and will be continued in this paper, whose aim is to enrich the model with metabolic-like capabilities, besides morphogenetic ones. The rest of this paper is organised as follows: section 2 highlights the main features of the model of development in its previous version and the relevant evo-devo method, section 3 describes the model extension, section 4 delves into the details of the simulation performed, section 5 discusses the biological correlates and section 6 draws the conclusions.

Epigenetic Tracking highlights

Shapes are composed of cells deployed on a grid; development starts with a cell (zygote) placed in the middle of the grid and unfolds in N age steps, counted by the variable "Age Step" (AS), which is shared by all cells and can be considered the "global clock" of the organism. Cells belong to two distinct categories: "normal" cells, which make up the bulk of the shape and "driver" cells, which are much fewer in number (typical value is one driver each 100 normal cells) and are evenly distributed in the shape volume. Driver cells have a Genome (an array of "instructions", composed of a left part and a right part) and a variable called cellular epigenetic type (CET, an array of integers). While the Genome is identical for all driver cells, the CET value is different in each driver cell; in this way, it can be used by different driver cells as a "key" to activate different instructions in the Genome. The CET value represents the source of differentiation during development, allowing driver cells to behave

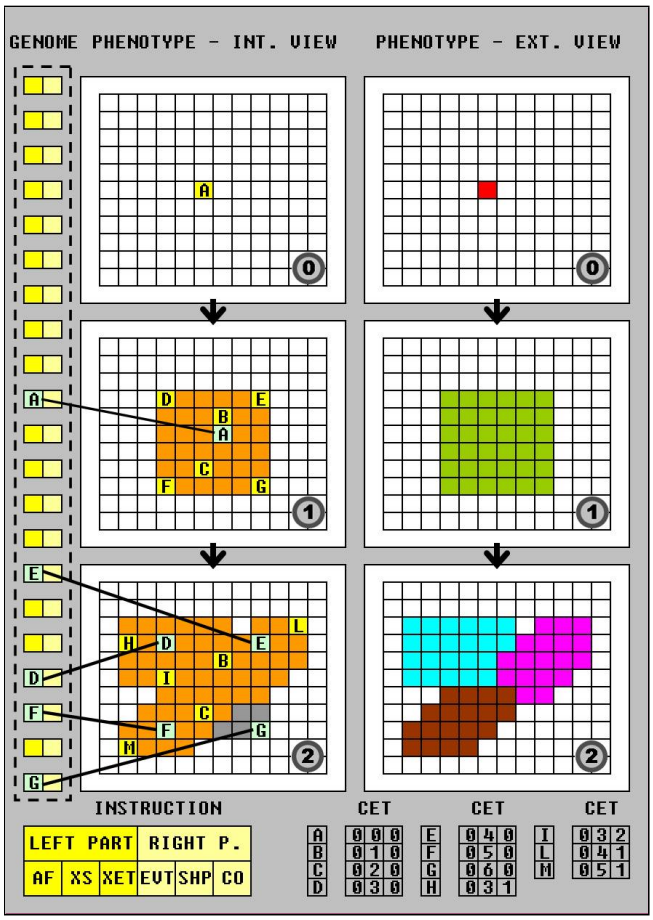


Figure 1: Example of development in three steps (AS=0,1,2) driven by five instructions: a proliferation triggered in step 1 on driver cell labelled with A, three proliferations triggered in step 2 on driver cells labelled with D, E and F and an apoptosis triggered in step 2 on driver cell labelled with G. Internal view on the left, external view on the right.

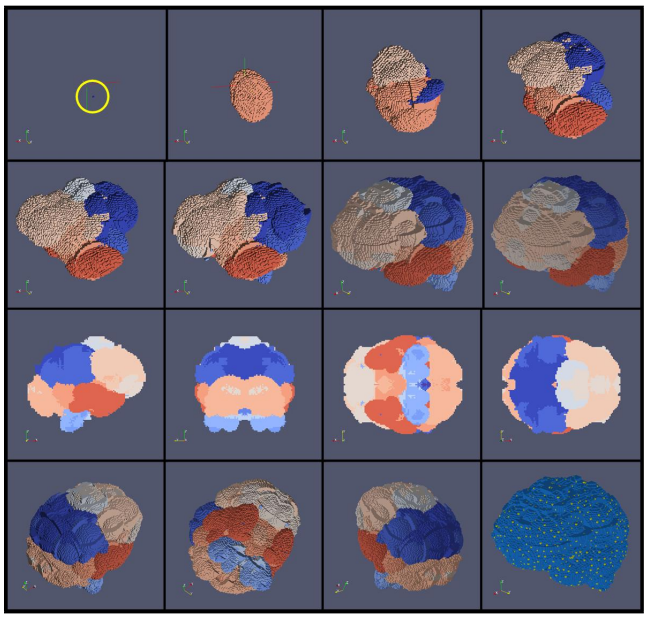


Figure 2: Example of development coded in a Genome composed of 360 instructions, evolved in 16000 generations; the shape represents an artificial brain, composed of 200.000 cells. In the upper part, the development sequence; in the lower part, some snapshots of the final phenotype taken from different angles.

differently despite sharing the same Genome. A shape can be "viewed" in two ways: in "external view" cells are shown with their colours; in "internal view" colours represent cell properties: blue is used for normal cells alive, orange for normal cells just (i.e. in the current age step) created, grey for cells that have just died, yellow for driver cells (regardless of when they have been created).

An instruction's left part is composed of the following elements: an activation flag (AF), indicating whether the instruction is active or not; a variable called XET, of the same type as CET; a variable called XS, of the same type as AS. At each step, for each instruction and for each driver cell, the algorithm tests if the instruction's XET matches the driver's CET and if the instruction's XS matches AS. In practise, XS behaves like a timer, which makes the instruction activation wait until the clock reaches a certain value. If a match occurs, it triggers the execution of the instruction's right part, which codes for three things: event type, shape and colour. Instructions give rise to two 'types' of events: "proliferation instructions" cause the matching driver cell (called "mother cell") to proliferate in the volume around it (called "change volume"), "apoptosis instructions" cause cells in the change volume to be deleted from the grid; the parameter 'shape' specifies the shape of the change volume, in which the proliferation/apoptosis events occur, choosing from a number of basic shapes called "shaping primitives"; in case of proliferation, the parameter 'colour' specifies the colour of the new cells.

Always in case of proliferation, both normal cells and driver cells are created: normal cells fill the change volume, driver cells are "sprinkled" uniformly in the change volume. To each new driver cell a new, previously unseen and unique CET value is assigned (consider for example proliferation triggered on A in figure 1), obtained by starting from the mother's CET value (the array [0,0,0] in the figure, labelled with A) and adding 1 to the value held in the ith array position at each new assignment (i is the current value of the AS counter); with reference to the figure, the new driver cells are assigned the values [0,1,0],[0,2,0],[0,3,0], ... , labelled with B,C,D, etc. (please note that labels are just used in the figures for visualisation purposes, but all operations are made on the underlying arrays). In practise a prolifer-

ation event does two things: first creates new normal cells and sends them down a differentiation path (represented by the colour); then creates other driver cells, one of which can become the centre of another event of proliferation or apoptosis, if in the Genome an instruction appears, whose XET matches such value. Figure 1 reports an example of development hand-coded.

The model of development described, coupled with a standard evolutionary technique, becomes an evo-devo method to generate arbitrarily shaped 2d or 3d cellular sets. The method evolves a population of Genomes that guide the development of the shape starting from a zygote initially present on the grid, for a number of generations; at each generation development is let unfold for each Genome and, at the end of it, adherence of the shape to the target shape is employed as fitness measure. *In silico* experiments (example in figure 2) have proved the effectiveness of the method in devo-evolving any kind of shape, of any complexity (in terms e.g. of number of cells, number of colours, etc.); being shape complexity a metaphor for organismal complexity, such simulations established the method's potential to generate the complexity typical of biological systems. The effectiveness of the method is, in our opinion, to be reconducted to the presence of a homogeneous distribution of driver cells, which keeps the shape "plastic" throughout development and allows artificial evolution to exploit physics to meet its ends.

Our model displays some similarities with L-systems; both models have productions that replace existing symbols with other symbols: the key difference lies in the mechanism for generating new symbols. In L-systems the new symbols have to be listed explicitly, in our model the number of new symbols is proportional to the size of the change volume, while the symbols themselves (the CET values) are created through an automatic procedure, which never changes and therefore is not encoded in the Genome: this feature allows a more compact representation of the productions in the Genome. Another important difference is that L-systems draw the symbols from a finite alphabet, while in the case of Epigenetic Tracking the alphabet is virtually unbounded and this "unboundedness" paves the way for open-ended evolution. CA-based models of development also have a cell state variable and again the key difference resides in the mechanism of assignment: while in CA-based models the value of the cell state is determined by the states of neighbouring cells, in our model it is assigned to cells as they are created (during a proliferation event); of course this is not the only difference: in CA models there is no distinction between normal and driver cells, etc.

In the current model version each cell can be considered as composed of two modules: 1) a "Morphogenetic Module", comprising all cellular variables related to morphology, such as spatial position and colour and 2) a "Change Module", consisting of the list of change instructions and

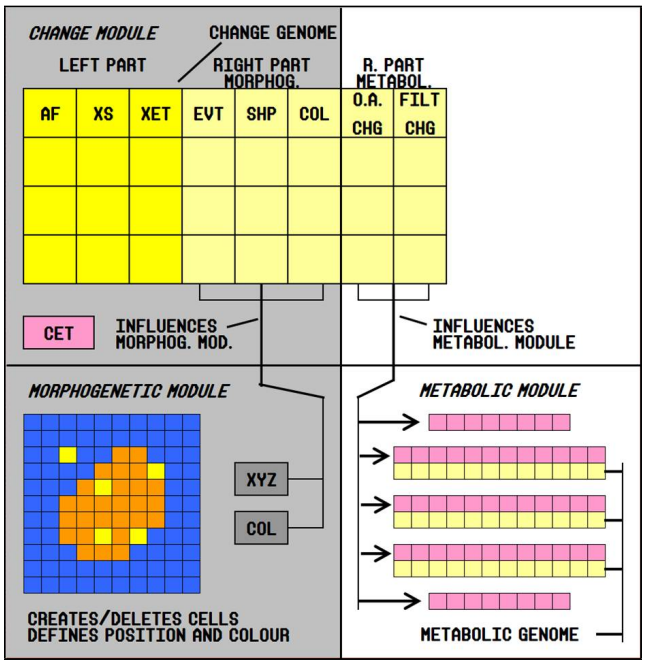


Figure 3: The old version of the model, dedicated to morphogenesis (on grey background); the new version of the model adds a part dealing with metabolic computation (on white background). Genetic elements are coloured in yellow; epigenetic elements are coloured in pink.

the CET (see left part of figure 3, on grey background); the Change Module's instructions code for changes affecting the Morphogenetic Module. Each module is in turn composed of "genetic" variables (unchanged during development and identical in all cells) and "epigenetic" variables (of genetic nature, but changed during development and potentially different in each cell). According to this definition, the Change Module is made up of a single block of genetic memory (the Genome, which will now be renamed "Change Genome") and an epigenetic variable (the CET). Besides possessing properties such as position and colour, cells do not perform any function; the present model has nonetheless served the purpose of modelling morphogenesis, a process by which an organism's external appearance -characterised by physical properties such as shape and colour- is created.

On the other hand, we know that real cells, besides having a position in space and a colour, are sophisticated micro-machines that carry out complicated chemical reactions, taking certain molecules as inputs and producing other molecules as outputs; the sum of these reactions, which represents the bulk of the cellular function, is referred to as the cell metabolism. Pancreatic cells, for instance, produce, among others, the hormones insulin, glucagon, and somatostatin; liver cells take in and degrade insulin, glycogen and hemoglobin and produce cholesterol and triglycerides, etc.

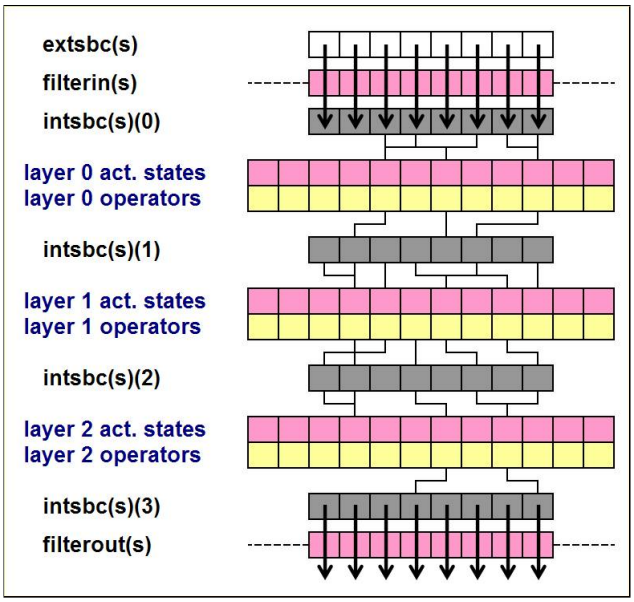


Figure 4: Metabolic computation in a cell. Operators are organised in layers: layer k converts the substance concentration array intsbc(s)(k) into intsbc(s)(k+1); each operator has an associated flag indicating the operator's activation state; the exchange of substances between the interior and the exterior of the cell is mediated by the arrays filterin and filterout.

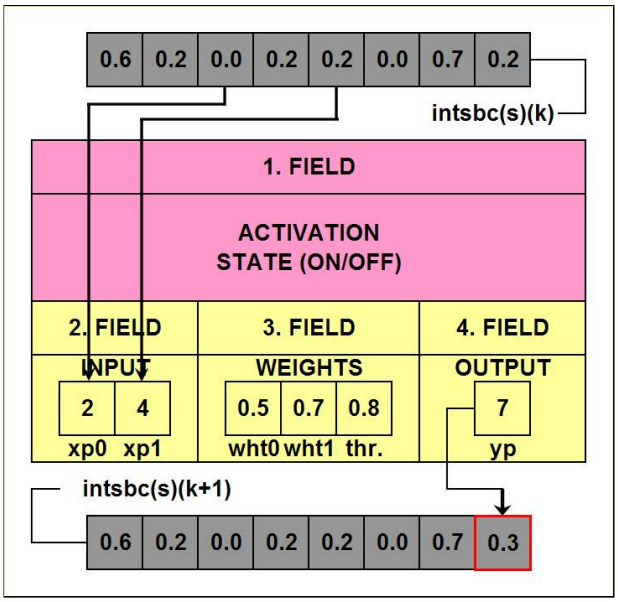


Figure 5: Details of a single layer k operator. The first field indicates the operator's activation state; the second field specifies which substances are to be loaded from intsbc(s)(k); the third field specifies the weights and the fourth field defines which susbtance is to be "influenced" in intsbc(s)(k+1).

The cellular metabolic machine is realised through the combined action of many simple "processors", each of which is dedicated to processing only few chemical subtances; such processors are implemented by genes that are turned on in the relevant cell. Different cell types have different patterns of gene activation, which allow cells to perform different specialised jobs; genes are by default active: the selective de-activation of specific genes is achieved primarily through a process called methylation, which prevents their transcription and their use in the gene network. The remainder of the paper will be dedicated to enriching the model with the ingredients necessary to realise the equivalent of a metabolic network.

Extended Model

The key innovation of the extended model (see figure 3) is the presence of a module, called "Metabolic Module", dedicated to carrying out the equivalent of metabolic operations. The elements responsible for such operations, called "operators", are arranged in layers and are grouped in a second Genome, called "Metabolic Genome"; to each operator a binary flag is associated, indicating the activation state; two other arrays, called filterin and filterout, are present, dedicated to managing the exchange of subtances of the cell with the external environment. The Change Genome present in the previous model version is still present in the new version in an extended form, in which the instructions' right parts, besides defining the events of proliferation and apoptosis and the shape and colour of the cells created, add some specifications relevant to changes affecting the cell metabolic dynamics.

Figure 4 gives a representation of the functioning of the Metabolic Module. As we said, the Metabolic Module is composed of a number of operators, each associated to a "layer number", so that the whole set of operators has the structure of a strictly-layered network. Each operator has a flag that indicates whether the operator is active or not: if not active, it is excluded from the computation. The operands are the equivalent of chemical substances and are grouped in two arrays called intsbc and extsbc ("internal" and "external" "substance concentrations"), whose values are real numbers comprised in the [0,1] interval representing substance concentrations; more precisely intsbc(s)(k) and extsbc(s)(k) are the concentrations relevant to substance s, to be processed by layer k operators. The arrays intsbc and extsbc represent the chemical mix present inside the cell and the chemical micro-environment present around the cell respectively.

The first processing step consists in copying the content of extsbc into intsbc; this copy operation is mediated by the array filterin, implementing a filter that allows only certain

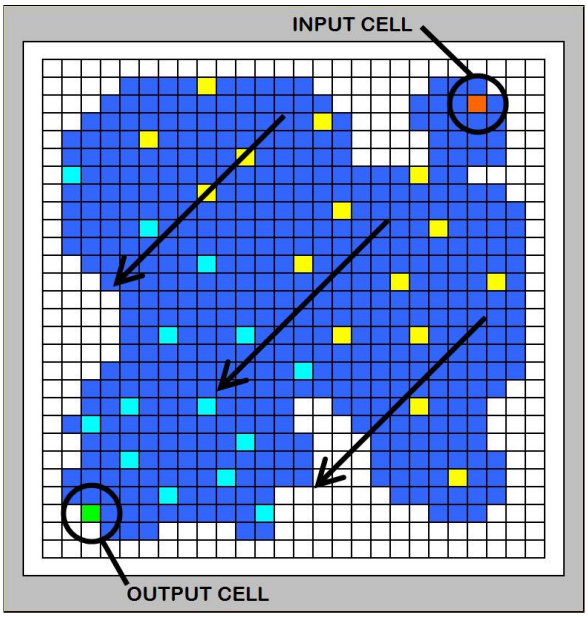


Figure 6: Each driver cell is assigned a number, depending on the distance from the input cell and the output cell (cells farther from the input and closer to the output have a higher number), so that the whole of driver cells make up a layered network; in the figure cells having different numbers are marked with different colours; arrows indicate the direction of the computation flow.

types of chemical substances to enter the cell: in practise intsbc(s)(0) is copied from extsbc(s) only if filterin(s)=1, otherwise (if filterin(s)=0) intsbc(s)(0) is initialised to zero. The computation is carried out one layer at a time; the initial state of intsbc (initialised from extsbc) is intsbc(s)(0); it is processed by layer 0 operators and the resulting array is intsbc(s)(1); subsequently intsbc(s)(1) is processed by layer 1 operators and the resulting array is intsbc(s)(2). This procedure is repeated K times (K=3 in our experiments), until the final state of the operand array intsbc(s)(3) is reached. At the end of the cycle, the content of intsbc "exits" the cell and is added to the extsbc of all other cells; the value intsbc(s)(3) to be added is multiplied by two factors: the first factor (filterout(s)) is a value that can be equal to -1 or +1; the second factor is a real number comprised in the [0,1] interval that depends on the distance between the cell and the other cell in whose extsbc the cell's intsbc is being copied. The function of filterout is analogous to that of filterin, only the set of possible values is different: (0,1) for filterin and (-1,1) for filterout.

The execution of an operation (performed by a single operator) is shown in figure 5. Each operator has four fields. The first field is a binary flag indicating whether the operator is active or not; the second field is an array of N integers (N=2 in our experiments), where the ith integer xp(i) represents the position of the ith input substance in the intsbc array; the third field is an array of N+1 float, being the ith float wht(i) the "weight" to be multiplied by the value contained in the ith position of the intsbc array; the products specified are summed together and then added to the (N+1)th weight (called "threshold"); the fourth field (yp) is an integer representing the position of the intsbc array to which the operator's output value (yv) is added. The operation implemented is described by the following equations (it is the classical nonlinear weighted sum neuron-like function; σ is the sigmoid function):

$$yv = \sigma(\sum(wht(i) * intsbc(xp(i)(k))) + threshold)$$
$$intsbc(yp)(k+1) = intsbc(yp)(k) + yv$$

For computational reasons the metabolic process has been so far implemented in driver cells only. In order to provide the shape with a direction for the computation flow, an input cell and an output cell are defined (actually, since the positions of driver cells are not known at the beginning of the experiment, two points in space are given and the two driver cells closest to such points are taken as input and output cell). Then, each driver cell is assigned a number which depends on its distance from the input cell and the output cell (cells farther from the input and closer to the output have a higher number -see figure 6). The initialisation of the input cell's extsbc with a set of input values triggers the start of the computation, which is executed for all number 1 cells, then for all number 2 cells etc., until the output cell is reached. The computation is repeated E times, where E is the number of examples (each example is made up by a set of input values and a set of target output values).

The Metabolic Module described provides cells with a computational tool able to carry out the equivalent of a metabolic process. So far, nevertheless, the set of operators (coded by the metabolic Genome) is identical for all cells; this leads to a biologically unrealistic behaviour, in which all cells carry out the very same computation and differences in the outputs are only determined by differences in the inputs. This is in contrast to what happens in biological organisms, where cells belonging to different organs have gene regulatory networks specialised to perform the metabolic reactions required by the organ's function in the body, despite the fact that all cells are endowed with the same set of genes. This specialisation is achieved through the selective inactivation of individual genes that, through multiple chemical mechanisms, are excluded from the network; the introduction in our model of the equivalent of such specialisation will require an extension to the right part of change instructions.

The extended right part is shown in the north-east quadrant of figure 3, on white background. The old right part (north-west quadrant, grey background) contains the code that specifies as usual the type of event (proliferation or apoptosis) and, in case of proliferation, the shape and colour

of the cells created. Besides position and colour, each cell has now also a set of operators each with an associated binary flag indicating its activation state; when a new cell is created during a proliferation event, the array of activation states is inherited from the mother cell. The first new right part field is a a P-dimensional binary array, called "operator activation changes" ("O.A. CHG" in the figure), specifying the P operator activation flags which have to be changed (0's are turned into 1's and 1's are turned into 0's); in this way the new cells end up having a set of active operators different from that of the mother, creating the potential for metabolic specialisation. Similarly, also the arrays filterin and filterout are inherited from the mother during proliferation and the second new field, called "filter changes" ("FILT CHG" in the figure) specifies the changes affecting such arrays. In other words, the new right part block contains the code that specifies the epigenetic part of the Metabolic Module, which can become different in every cell and which, together with the genetic part (equal in all cells), determines cell behaviour.

We end this section by showing how the Metabolic Module is integrated in the overall model of development. In the extended model age steps can be divided into a "change phase" and an "expression phase". In the change phase, the couple of variables (CET,AS) triggers the activation of proliferation and apoptosis instructions on a number of driver cells; as a consequence, some new cells are created and some existing cells are deleted from the grid. The newly created cells are given a position in space and a colour which are based on the position of the mother and the morphogenetic portion of the instructions' right parts; the daughter cells are also provided with a set of operators, a relevant set of activation states and filter arrays, all inherited from the mother. The code contained in the metabolic right part brings some changes to the activation pattern of the operators and to the filters, allowing specialisation to take place: this ends the change phase. In the expression phase the metabolic network carries out the cell's specialised metabolic function, processing input substances and producing output substances. These two phases can be thought of to correspond roughly to the 'mitosis' phase and the interphase of the cell cycle (the main difference being that in our model the cycle is syncronised for all cells, while in real cells it is not).

Simulation

The extended model of development has been tested with the same criterion used to test the previous version of the model, i.e. we have tested the model's susceptibility to produce a target result in combination with a standard evolutionary algorithm; in other words, we have tested the model's *evolvability*. In previous simulations concerned only with morphogenesis, we adopted a fitness function formula initially proposed by H. de Garis (De Garis, 1999):

$$\text{sfit} = (\text{ins} - \text{outs})/\text{des}$$

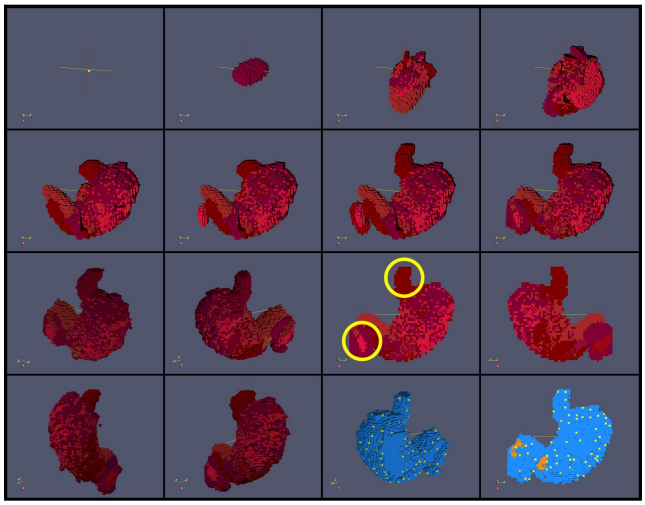


Figure 7: Morphogenesis of the artificial stomach. The upper part of the figure shows the development sequence, the lower part some snapshots of the final shape taken from different angles. Shape made up of 20.000 cells, genome composed of 300 instructions, evolved in 30000 generations.

where ins is the number of cells of the evolved shape falling inside the target shape, outs is the number of cells of the evolved shape falling outside the target shape, des is total number of cells of the target shape; for coloured target shapes, also the adherence to colours is taken into account (i.e. in order to add 1 to the ins count, a given cell must fall inside the target shape and its colour must be equal to that of the target cell in the same position).

To allow for the evolution of the metabolic part, a second fitness function has been introduced, defined through the following procedure. We define E examples, each composed of a set of input concentration values and a set of output target concentration values, indicated with tgtin(e)(s) and tgtout(e)(s). For each example, the extsbc of the input cell is initialised with the tgtin values; then the computation is carried out for all cells as described in the previous section, until the output cell is reached: let actout(e)(s) be the value of the output cell's extsbc relevant to the eth example and to the sth substance type. The computation is repeated for the total number of examples foreseen; the metabolic fitness function is defined as the sum of the differences between the target output and the actual output across all examples and substance types (normalised dividing by the number of terms):

$$mfit = \sum(abs(actout(e)(s) - tgtout(e)(s)))/(E \cdot S)$$

The overall fitness is then calculated as a weighted average of the shape fitness and the metabolic fitness (in the simulations performed coe1=coe2=0.5):

$$fit = coe1 \cdot sfit + coe2 \cdot mfit$$

	CELL #	OPERATORS 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ACTIV. STATE	0	1	0	1	1	1	1	1	1	1	1	0	1	1	0	1	1
ACTIV. STATE	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
ACTIV. STATE	2	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1
ACTIV. STATE	3	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	0
ACTIV. STATE	4	1	1	1	0	1	1	0	1	1	1	1	0	1	1	1	1
ACTIV. STATE	5	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
ACTIV. STATE	6	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1
ACTIV. STATE	7	1	0	1	1	1	1	1	1	0	1	1	0	1	1	1	1
ACTIV. STATE	8	1	1	1	1	1	1	0	1	1	1	1	1	0	0	1	1
ACTIV. STATE	9	1	1	1	1	1	1	0	1	1	1	1	1	0	0	1	1
ACTIV. STATE	10	1	1	1	0	1	1	0	1	1	1	1	0	1	1	1	1
ACTIV. STATE	11	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
ACTIV. STATE	12	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1
ACTIV. STATE	13	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
ACTIV. STATE	14	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1
ACTIV. STATE	15	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1
ACTIV. STATE	16	1	0	1	1	1	1	1	1	0	1	1	0	1	1	1	1
ACTIV. STATE	17	1	0	1	1	1	1	1	1	0	1	1	0	1	1	1	1
ACTIV. STATE	18	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
ACTIV. STATE	19	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1
ACTIV. STATE	20	1	1	1	1	1	1	0	1	1	1	1	1	0	0	1	1
ACTIV. STATE	21	1	1	1	1	1	1	0	1	1	1	1	1	0	0	1	1
ACTIV. STATE	22	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1
ACTIV. STATE	23	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
ACTIV. STATE	24	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1
ACTIV. STATE	25	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1
ACTIV. STATE	26	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	0
ACTIV. STATE	27	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
ACTIV. STATE	28	1	1	1	1	1	1	0	1	1	1	1	1	0	0	1	1
ACTIV. STATE	29	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1
ACTIV. STATE	30	1	1	1	0	1	1	0	1	1	1	1	0	1	1	1	1
ACTIV. STATE	31	1	1	1	0	1	1	0	1	1	1	1	0	1	1	1	1
ACTIV. STATE	32	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1
ACTIV. STATE	33	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1
ACTIV. STATE	34	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1
ACTIV. STATE	35	1	0	1	1	1	1	1	1	0	1	1	0	1	1	1	1
ACTIV. STATE	36	1	1	1	1	1	1	0	1	1	1	1	1	0	0	1	1
ACTIV. STATE	37	1	1	1	0	1	1	0	1	1	1	1	0	1	1	1	1
ACTIV. STATE	38	1	0	1	1	1	1	1	1	0	1	1	0	1	1	1	1
ACTIV. STATE	39	1	0	1	1	1	1	1	1	0	1	1	0	1	1	1	1

Figure 8: Operator activation states of the first 40 cells. As can be noted, there are sets of cells (those created in the same proliferation event) sharing the same pattern of operator activations (and the same filters).

	e	SUBSTANCES 0	1	2	3	4	5	6	7
INPUT	0	.40	.20	.00	.10	.10	.80	.60	.00
TARGET		.70	.20	.00	.30	.30	.50	.00	.70
OUTPUT		.50	.02	.12	.29	.00	.00	.00	.68
ABS(TGT-ACT)		.20	.18	.12	.01	.30	.50	.00	.02
INPUT	1	.90	.00	.50	.80	.00	.00	.00	.00
TARGET		.20	.50	.00	.10	.90	.00	.00	.00
OUTPUT		.29	.18	.18	.16	.00	.00	.00	.01
ABS(TGT-ACT)		.09	.32	.18	.06	.90	.00	.00	.01
INPUT	2	.20	.10	.00	.20	.40	.00	.00	.20
TARGET		.00	.00	.60	.00	.00	.60	.00	.20
OUTPUT		.34	.11	.19	.29	.00	.00	.00	.19
ABS(TGT-ACT)		.34	.11	.41	.29	.00	.60	.00	.01
INPUT	3	.50	.40	.30	.60	.80	.00	.00	.90
TARGET		.10	.40	.20	.00	.30	.10	.50	.40
OUTPUT		.15	.40	.19	.09	.00	.00	.00	.42
ABS(TGT-ACT)		.05	.00	.01	.09	.30	.10	.50	.02
INPUT	4	.80	.00	.70	.00	.00	.30	.00	.40
TARGET		.60	.10	.80	.50	.90	.70	.00	.00
OUTPUT		.62	.19	.14	.09	.00	.00	.00	.05
ABS(TGT-ACT)		.02	.09	.66	.41	.90	.70	.00	.05
INPUT	5	.30	.60	.00	.70	.50	.70	.10	.80
TARGET		.90	.00	.00	.80	.00	.00	.00	.90
OUTPUT		.12	.00	.04	.73	.00	.00	.00	.99
ABS(TGT-ACT)		.78	.00	.04	.07	.00	.00	.00	.09
INPUT	6	.00	.00	.60	.30	.00	.50	.00	.20
TARGET		.70	.20	.60	.30	.50	.00	.50	.20
OUTPUT		.46	.22	.18	.09	.00	.00	.00	.17
ABS(TGT-ACT)		.24	.02	.42	.21	.50	.00	.50	.03
INPUT	7	.10	.70	.90	.00	.10	.00	.90	.00
TARGET		.50	.30	.40	.90	.10	.00	.10	.00
OUTPUT		.49	.30	.17	.09	.00	.00	.00	.00
ABS(TGT-ACT)		.01	.00	.23	.81	.10	.00	.10	.00
INPUT	8	.90	.00	.70	.10	.40	.80	.60	.00
TARGET		.40	.80	.00	.00	.00	.80	.00	.30
OUTPUT		.34	.19	.19	.09	.00	.00	.00	.26
ABS(TGT-ACT)		.06	.61	.19	.09	.00	.80	.00	.04
INPUT	9	.40	.20	.00	.50	.00	.60	.00	.30
TARGET		.00	.00	.10	.70	.20	.00	.60	.70
OUTPUT		.34	.01	.10	.65	.00	.00	.00	.69
ABS(TGT-ACT)		.34	.01	.00	.05	.20	.00	.60	.01

Figure 9: Target-output comparison. The figure shows, for each example, input value, target value, actual output value and the absolute difference between target and output value.

In the following the values of some key parameters of the algorithm. The target shape is an artificial stomach composed of some 20.000 cells. The linear "driver to normal ratio" used in proliferation events is 4, meaning that one driver cell is created every 4 normal cells for each dimension (in three dimensions the ratio is thus $4^3 = 64$). As far as the metabolic part is concerned, the number of substances is 8, the number of operators is 16 and the number of examples is 10. In this experiment only one input cell and one output cell are foreseen: the driver cell closest to a predefined "input position" is defined as the input cell (analogously for the output cell). The genetic population is composed of 500 individuals (represented as strings of quaternary digits), undergoing elitism selection; GA parameters are 50% single point crossover, mutation rate of 0.1% per digit.

Simulation results are shown in figures 7-9. Figure 7 shows the development sequence of the artificial stomach from the single cell stage to its final shape and some snapshots taken from different angles; circles indicate the positions of the input and output cells. Figure 8 shows the operator activation state for the first 40 cells (for reasons of space); figure 9 shows the comparison of the target and actual values of the extsbc array for all examples. As can be seen, results are good both for the morphogenesis part and for the metabolic part; the final value of the shape fitness is 0.82, the final value of the metabolic fitness is 0.80; the total number of driver cells that make up the metabolic network is 848.

Biological correlates

In biology, the term epigenetics refers to changes in phenotype or gene expression caused by mechanisms other than changes in the DNA sequence. These changes may remain through cell divisions for the remainder of the cell's life and may also last for multiple generations. One way that epigenetic influences are implemented is through the remodelling of chromatin and one way chromatin remodelling is accomplished is through the addition of methyl groups to the DNA. DNA methylation in vertebrates typically occurs at CpG sites (cytosine-phosphate-guanine sites) and results in the conversion of the cytosine to 5-methylcytosine, catalysed by the enzyme DNA methyltransferase. The bulk of mammalian DNA has about 40% of CpG sites methylated

but there are certain areas, known as CpG islands -which are GC rich- where none are methylated: these are associated with the promoters of a high percentage of mammalian genes, including all ubiquitously expressed genes (in general there is an inverse relationship between CpG methylation and transcriptional activity).

If we stick to the definition of epigenetic cellular elements given in section 2 (variables of genetic nature changed during development and potentially not identical in different cells), the CET value (already present in the previous version of the model) qualifies as an epigenetic element. In the extended model two new epigenetic memories have been introduced: the operator activation states and the I/O substance filters. These two new memories have their biological counterparts in the DNA methylation marks and in the various "channels" present on the cell membrane (which mediate inside-outside cellular communication) respectively while, at the current level of knowledge, the CET has no biological equivalent. As far as the genetic part is concerned, in the new version of the model two Genomes are present: the Change Genome and the Metabolic Genome. This distinction appears to have no correspondence in nature, where a single Genome seems to be present, more similar to the Metabolic Genome in structure (genes are akin to metabolic operators). On the other hand, we can imagine to decompose the specifications contained in the change instructions into smaller units equivalent to operators, thus reconducting Change and Metabolic Genomes into a unitary representational framework: this will be a matter for future work.

The addition of computational capabilities to cells represents a significant step on the way to reducing the gap between Epigenetic Tracking and real biological systems. According to current knowledge, in multicellular organisms the behaviour of a single cell is determined by three factors: i) the genome; ii) the epigenome; iii) the influence of the chemical microenvironment surrounding the cell, created by all chemical signals generated by other cells. Cell behaviour can be further divided into a change (or "mitotic") part and an expression (or "interphasic") part; while the previous version of the model covered essentially only the change part, with genetic and epigenetic mechanisms, the extended version covers also the expression part, still with genetic and epigenetic mechanisms. The next logical step is represented by the addition of the cellular microenvironment as yet another determinant of cell behaviour.

Conclusions and future research

In the present paper the model of development called Epigenetic Tracking has been extended by adding to artificial cells computational capabilities (besides physical attributes -position and colour); cells endowed with such capabilities constitute the equivalent of a metabolic network. The extended model has been applied to the problem of devo coevolving both the shape and the metabolic network of an artificial organ (the stomach): the successful result of the simulation have been presented and discussed. Future research along this line is aimed at further reducing the gap between the model and real biological systems; in this respect, a key ingredient to be added to the model is represented by the influence of the surrounding chemical microenvironment, other than genetic and epigenetic factors, as another determinant of cell behaviour. I thank Perry for helping me reviewing the paper.

References

De Garis, H. (1999). *Artificial embryology and cellular differentiation*. Academic Press.

Doursat, R. (2008). Programmable architectures that are complex and self-organized: From morphogenesis to engineering. In *Proceedings of ALIFE XI*.

Fontana, A. (2008). Epigenetic tracking, a method to generate arbitrary shapes by using evo-devo techniques. In *Proceedings of EPIROB 2008*.

Fontana, A. (2009). Epigenetic tracking: Biological implications. In Kampis, G. and Szathmary, E., editors, *Proceedings of ECAL 2009*.

Gruau, F., Whitley, D., and Pyeatt, L. (1996). A comparison between cellular encoding and direct encoding for genetic neural networks. In *Genetic Programming 1996*.

Hornby, G. S. and Pollack, J. B. (2002). Creating high-level components with a generative representation for body-brain evolution. *Artificial Life*.

Joachimczak, M. and Wrobel, B. (2008). Evo-devo in silico: a model of a gene network regulating multicellular development in 3d space with artificial physics. In *Proceedings of ALIFE XI*.

Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*.

Lindenmayer, A. (1968). Mathematical models for cellular interaction in development. *Journal of Theoretical Biology*.

Miller, J. F. and Banzhaf, W. (2003). Evolving the program for a cell: from french flags to boolean circuits. In Press, A., editor, *On Growth, Form and Computers*.

Stanley, K. and Miikkulainen, R. (2003). A taxonomy for artificial embryogeny. *Artificial Life*.

Turing, A. (1952). The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society*.

Diversity From a Monoculture: Effects of Mutation-On-Copy in a String-Based Artificial Chemistry

Simon Hickinbotham[1], Edward Clark[1], Susan Stepney[1], Tim Clarke[2],
Adam Nellis[1], Mungo Pay[2], Peter Young[3]

[1]Department of Computer Science, [2]Department of Electronics, [3]Department of Biology
York Centre for Complex Systems Analysis, University of York, UK
sjh@cs.york.ac.uk www.yccsa.org

Abstract

We have developed an artificial chemistry that allows self-maintaining molecular systems to mutate and exhibit innovative behaviour. The molecular species in the chemistry are defined by strings of symbols that specify both the binding affinity and the reaction. We define a replicase molecule that can copy any other molecule that binds at a particular region on the replicase. Molecules are copied on a symbol-by-symbol basis. Occasional mis-copying of an individual symbol forms our mutation scheme. This paper describes the characteristics of the resulting evolutionary system. We ran 1,000 open-ended trials and observed an unexpectedly wide range of emergent phenomena, with many parallels to biological systems. We report these phenomena in qualitative terms, and give details of one of the most interesting among them: the emergence of co-dependent replicase hypercycles.

Introduction

Early-earth molecular systems are of interest due to their relatively simple replication mechanisms, gene multiplicity, and the blurring of the genotype-phenotype boundary. The simplicity of these systems make them a good target for models of chemical evolution. We have been working on an artificial chemistry called Stringmol [4, 3], which combines a stochastic chemistry, variable binding rates and a simple sequence-based programming language.

Stringmol is a rich intra-cellular RNA-world analogue in which there is no distinction between molecular template and molecular machine. We have recently been experimenting with a unimolecular system, where the molecule is capable of self-copying. We call this molecule a replicase. The sequence of symbols that specify a particular molecular species can be interpreted both as a template (a sequence of symbols) and as a program, which can be executed to carry out the reaction between molecules. If two molecules bind to each other by having a sufficiently "strong" match in their sequences, a handshaking process determines where the program that specifies the reaction starts. In our replicase example, this handshaking determines which molecule is copied and which molecule carries out the copying. In earlier work [5] we found that the function of simple molecular simulations is heavily influenced by bind affinity between molecules, so it is important that the representation of the molecules allows bind affinity to be specified on the genome.

String- or tape-based evolutionary simulations have been reported frequently in the literature, and there are many parallels between biology and computer science in the area. Turing machines make use of a tape and read-write heads [13]. They preceded von Neumann's self-reproducing automata [15]. Both of these architectures have interdependence of data and program, and use self-copying as key demonstrators of the function of the system. These are very simple state machines, with only a loose analogue to the concept of the organism. More recently, Ray's Tierra [11] and the AVIDA architecture [7] have expanded on the paradigm of organism-as-tape, with interesting emergent phenomena that mirror biology. A less well-known but related theme is that of expressing the organism as a container for a large *set* of strings, each of which contribute to the metabolism (and hence fitness) of the organism. Examples include Laing's kinematic machines from the 1970s [8], Hofstader's Typogenetics [6, 14], and Suzuki's string rewriting system [10]. The concept of mutation is realised only in Tierra and AVIDA. These two systems have a single tape per individual, mirroring the function of DNA in the organism. We believe that string systems have the potential to encode more than the genome of the system - the phenotypic machinery of gene expression can also be encoded on string-like agents and so lead to the evolution of effective machinery for genome organisation.

This paper concerns our early experiments with mutation in our replicase system. We believe that there should only be one form of "spontaneous" mutation in the system, and that this should occur when a symbol is copied from one sequence to another. We call this process "mutation-on-copy". In biology, mutation-on-copy certainly happens, especially when resources are running low; i.e. while the cell is under stress [16]. We believe that other forms of genome change should be effected by mechanisms intrinsic to the chemical model. For example it should be possible to construct

a transposon in the Stringmol language, which would allow macromutations whilst itself being a candidate for genomic control. Biological genomes are highly organised, and are responsible for their own expression. In other words, *the phenotype includes the genotype-reading structures, and is completely encoded in the genotype*. In yet other words, the genotype in its purest form is a sequence of symbols, and this encodes *everything else* that is manufactured in the cell, including the machinery for curating the genotype. We have preserved this property in our Stringmol model, and detail here a control experiment that attempts to determine the effects of single point mutations on such a system.

What might be expected of a single-container system that contains mutating molecular replicators? Our experiments confirm the prediction that a series of stable states would emerge, with eventual collapse of the system due to emergent selfish parasites. However, the observed range of reactive behaviour and the interesting dynamics were not expected to occur so rapidly in such a simple system. Analogues of parasitism, hypercycles, random drift, gene repression and co-evolution are reported. Unlike real biology, we are in a position to fully examine the system, and can detail the key events that led to the observed dynamics.

In an RNA-world analogue, such as the chemistry we present here, a molecule can act as both template and machine. Initially, two identical molecules come together, with one acting as the machine which makes a copy of the other. Mutants that are better templates subsequently sweep through the population, replacing the initial molecular species. More interestingly, we repeatedly observe the emergence of a molecular species that does not self-replicate but drives evolution to a state where the system is dominated for a long period by two co-dependent replicase species that are not self-maintaining. This is a catalytic hypercycle as defined by Eigen [2, fig.7].

It is interesting to consider the role of the container in these experiments. Many explanations for the origin of life include the use of membranes to keep the molecular template in close association with the machinery it specifies [9, 1], allowing selective advantage to operate on the machine-template complex as an entity. In early living systems, where mutation was rampant and much less tightly controlled, we observe that containers have a more time-critical role of preventing the rampant spread of emergent pathogens.

System overview

We give here a brief overview of our molecular system, which is described fully in [3] and [4]. A summary of the container metabolism is presented below, followed by a description and discussion of molecular structure. We pay particular attention to the role of sequence alignments and the mutation scheme in our chemistry.

Metabolism

A simulation can be considered as a set of reacting molecules whose movements inside a container are governed by a stochastic mixing function. All molecules are subject to *decay* (spontaneous destruction), which places a requirement upon the system to act in order to maintain itself in the face of entropy. Should molecules come sufficiently close to one another, then they can *bind* and *react*. The system has a clock. At each time step, all the molecules in the system are processed. Actions only occur if energy is available. Energy is consumed via binding and executing each instruction in a reaction. The likelihood of binding and the nature of the reaction is encoded in the string of each molecule in the encounter. Binding and reacting have an energy cost. At one particular time step, we specify that 25 energy units are available. Selection of which events consume the energy is stochastic. The balance between energy availability and the decay rate of the molecule maintains a population of around 350 molecules. We currently specify that only two molecules can ever participate in a single reaction, and that raw materials for the assembly of new molecules are available in saturation. These assumptions will be addressed in future work.

Molecular representation

Our molecular representation is a string of symbols. Each unique string is considered to be a unique molecular species. There are 33 symbols, most of which are non-functional. Maximum string length is 2000 symbols (to accommodate longer molecules with richer functionality), so there exists $n = \sum_{i=1}^{2000} 33^i \approx 10^{3037}$ potential molecular species. An important feature of the molecular representation is that it allows the possibility of several complementary subsequence alignments. Complementary alignments are necessary in order to prevent two identical molecules from binding to each other perfectly. Alignments have two key roles: firstly, they specify binding regions on molecules such that the more precise the alignment, the stronger the binding affinity; secondly they specify program flow in the functional region, commonly acting as placemarkers in "goto" statements. An important property of the representation is that the location of functional and binding regions is solely specified by the subsequences themselves, and different molecular species can bind at different sites on the sequence, so triggering different functions of the molecule. The sequence of the molecule is used to determine how likely a bind between molecules is via a process of Smith-Waterman alignment [12] of complementary symbols. Once a bind occurs, the sequence is treated like a program, commencing at the beginning of whichever aligned subsequence is furthest from the beginning of the string. There are 7 functional symbols, shown as non-alphabetical characters '`$`', '`>`', '`^`', '`?`', '`=`', '`%`', and '}'. Stringmol uses functional symbols to specify the manipulation of a set of point-

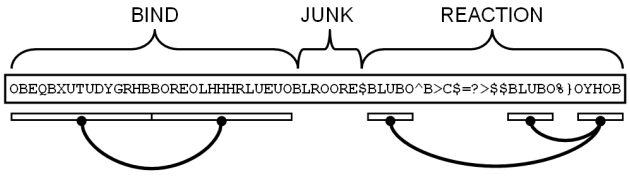


Figure 1: The seed replicase. The top line indicates the regions of the sequence. The sequence itself is shown in the centre box. Complementary alignments are indicated by black connecting lines at the bottom of the figure

ers which indicate positions on the molecular strings, and the symbols that the pointers index.

Mutation Scheme

One of the functional symbols is the copy operator '**=**'. This operator reads the symbol at the read pointer, and writes a copy of that symbol at the write pointer. To implement mutation-on-copy, we specify that a copy operation occasionally writes a different symbol to that being read with a probability $p_s = 0.00001$. More rarely still, insertion of an extra random symbol, or deletion of the symbol, take place with a much smaller probability $p_i = p_s/(10n)$, where n is the number of different symbol codes.

Experimental framework

We ran 1,000 simulations of a replicase environment under the mutation scheme described above. The goal was to evaluate whether the system would be robust to mutation, and if so, what effects it had on the molecular ecosystem. Each of the 1,000 trials had the potential to run indefinitely and only terminated when there were no molecules remaining in the system. This occurs when the replication mechanism deteriorates in some way so that the replicating molecules cannot copy themselves sufficiently quickly to counter the process of decay. In particular, we sought to identify emergent behaviours in the system that were not part of the original specification and arose by mutation.

The "seed replicase"

Here we describe the molecule used as the seed for the trial. It is one of many possible replicase molecules and is shown in figure 1. There are several features to note:

1. Two binding regions. Two are needed to allow a replicase to bind to a copy of itself because binding is *complementary*: a symbol is a perfect match to a different symbol in the set.

2. A junk region. Mutations here have no effect on the binding or reaction-program, allowing us to explore the effects of neutral mutation *drift*.

3. A functional region. This program specifies that the reaction involves creating a copy of the partner molecule in the reaction.

The seed replicase is 65 instructions long. The reactions takes 240 time steps to construct a new replicase molecule. All of the template codes in the seed replicase are more than one mutation away from a function code. Alignments in the functional region specify program flow. The two binding sites in our seed molecule do not align perfectly, which enables us to evaluate the evolutionary pressure on binding.

Analysis

As part of our evaluation, we developed several ways of representing the simulation data. Each molecule has a sequence of symbols. A particular sequence of symbols denotes a particular molecular species, which has an associated species number. The seed replicase is always species number 1. When a mutation occurs, a molecule with a novel sequence is generated, and this is assigned a new species number. In this way, we can record all new molecular species as they arise. We must also record the dynamics that ensue. Occasionally a new species increases in number and rises to dominance of the system, driving the previous dominant species to extinction. This is known in biology as a *sweep* event. We can capture these events by monitoring when the species number of the most abundant species changes (examples are shown in figure 4). We can record the reactions that exist between all species present in a system at any one time (see figure 6). Finally, we can record the *ancestry* of a molecular species: a new molecule is the product of a reaction between two other molecules, which belong to either one or two species types (see figure 7). These figures are described in more detail later.

With these tools to hand, we are able to demonstrate that our system is capable of producing innovative behaviour even from very simple starting conditions and with no external selection pressure. Essentially, the molecular community acts as a co-evolutionary system, in which the fitness of a particular molecular species is largely determined by the cohort of molecular species with which it shares the container. To demonstrate this, we present results on three levels. The first level gives summary observations and statistics from the 1,000 trials. Secondly, we offer a qualitative analysis of these trials, in which a range of emergent phenomena are qualitatively described. The third analysis gives details of a single trial with emergent phenomena and shows how a series of single-point mutations change the seed replicase system to a mutually-dependent "hypercycle" in which two molecular species cannot self-maintain, but maintain a population by copying each other.

General observations

The mutation rate delivers a mean time of 18,700 time steps for the creation of new molecular species. The majority of

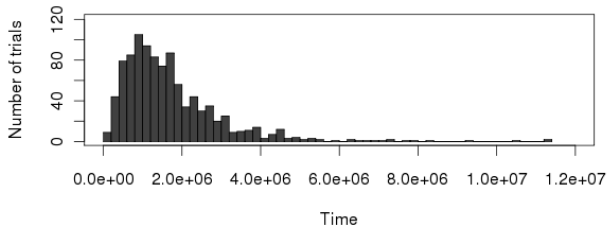


Figure 2: Distribution of extinction times for 1,000 trials

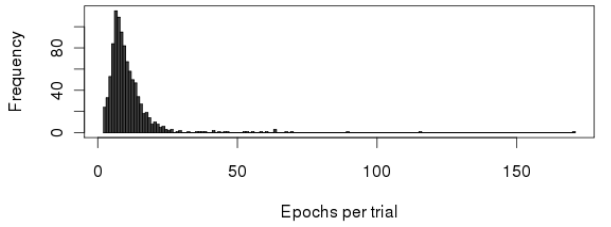


Figure 3: Histogram of number of epochs per trial

these new mutations are not "fixed" in the population and go extinct very quickly. Occasionally a new species arises that has some advantage over the current dominant species.

None of the 1,000 trials self-maintains indefinitely. The nature of extinction follows a uniform pattern as described below, but the timing of the extinction varies. Figure 2 shows the distribution of time to extinction for the molecular populations. The modal extinction time is 750000 time steps. In this time an average of 40 new species are produced.

Mutations occasionally produce molecules that rapidly multiply to become the dominant species in the system via the phenomenon of *invasion when rare*. We use the term *epoch* to describe the period over which a particular molecular species is dominant in the system; *sweep* describes a change in epoch. A histogram of the number of epochs per trial is shown in figure 3. The long tail on the histogram is a caused by runs where periods with co-dominant species that should be labelled as a single epoch are recorded by the analysis as a high number of very short epochs due to small fluctuations in abundance of the two species. This definition of the epoch is not particularly useful in situations where two species are co-dominant, but this behaviour was not predicted. Epochs for a single trial can be seen in figure 4.

A classification of emergent phenomena

In this section we give brief descriptions of the key phenomena we have observed in the 1,000 trials. These were identified by visual inspection of the plots of changes in the populations of molecular species, e.g. figures 4 and 5.

Extinction

All trials end when no molecules exist in the system. This occurs when there is a catastrophic decline in replicating molecules. The common cause of this is when a new 'parasitic' molecule arises that is 1) incapable of replicating itself, and 2) copied by the incumbent replicase at a higher rate than the replicase. Note that in order to be copied, a parasite must bind to the replicase sufficiently frequently. This tends to make the system more robust to molecular "junk" and explains why some of the trials continued for so long. A characteristic spike may be observed at the end of each run, which shows this new parasitic molecule as it rapidly increases and then declines when the last replicase molecules decay. Occasionally a parasite begins to overrun the replicase population, but it is unable to bind to a new replicase mutant that is created as the parasitic molecule is increasing. This is rare, occurring in only two of the trials.

Dynamics

Characteristic sweep. The majority of sweeps in our system take a constant form, as shown in figure 4. These are the the main cause of epoch change, and take less than 50,000 time steps for a new mutant to drive the previous dominant species to extinction.

Drift. Drift is observed when a neutral mutation of a dominant individual builds in numbers due to a random walk. Drift is common, occurring in 92 trials. It is plausible that sub-populations and slow sweeps (described below) are both commonly caused by drift. Species exhibiting drift tend to have mutations in the junk region, but can also show mutations in binding regions that do not change the bind affinity.

Sub-populations. These are species which persist in the community in fairly large numbers (more than 50 molecules of approximately 350 in the system). These are very common, occurring in nearly all runs. These sub-populations are nearly always wiped out when a new epoch begins, demonstrating the biological phenomenon of selective sweeps. *Enduring Sub-Populations*, that persist across more than one epoch, occur in 26 trials. This indicates that sub-populations tend to depend on some property of the dominant species in the system, essentially acting as non-lethal parasites. Co-dependence between dominant and sub-populations cannot be determined by examination of population numbers alone. In 2 trials we observed a sweep in a subpopulation whilst the dominant population remained stable.

Slow sweeps. A sweep can occasionally take much longer than the 50,000 time steps of a typical sweep. These are called "slow sweeps" and may be due to drift alone. An example can be seen in one of the hypercycle partners in figure 4 at around $t = 2,600,000$. Slow sweeps occurred in 52 trials.

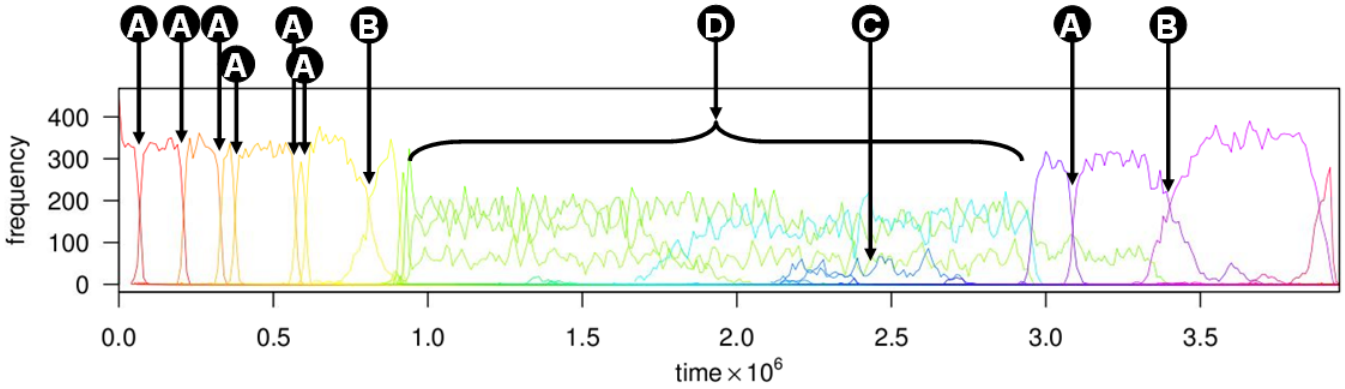


Figure 4: Dominant species in run 112. This trial exhibits (A) characteristic sweeps, (B) slow sweeps, (C) subpopulations, and (D) multispecies hypercycles.

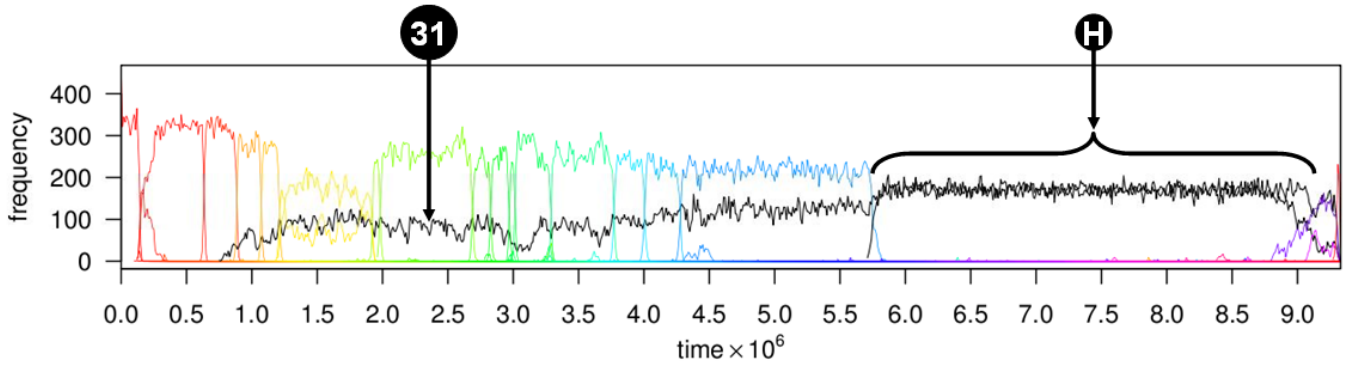


Figure 5: Dominant species in run 277. The short replicase (species 31) emerges at $t = 748,199$ and forms a hypercycle (H) at $t = 5,750,000$.

Rapid sweep sequences. Occasionally a mutant causes a "cascade" of new molecules by triggering a sequence of new unseen molecules that quickly dominate the population. The most common mechanism for this is a mutation that gives rise to a series of molecules that bind to a replicase such that less than their entire sequence is copied. This occurs in 31 trials.

Complex behaviour

Emergent hypercycles. A hypercycle occurs when an enduring sub-population increases in number until it becomes co-dominant with a dominant species. The species forming the enduring sub-population is not self-maintaining, but acts as a copier for the dominant species. The dominant species then repeatedly loses self-self affinity until it loses the ability to self-maintain altogether. The hypercycle occurs when the ability of the dominant population to self-maintain is lost, and the two species become co-dependent. This occurs in 8 trials. Hypercycles end with a sweep, but occasionally one of the partner molecules is still able to maintain a sub-population. A series of sweeps ensues, in which the sub-population declines slightly following each sweep. This occurs in 6 trials.

Spontaneous hypercycles. are the same as the emergent hypercycle, but forms from species that both arise in the immediately preceding epoch. The mechanism is under investigation. This occurs in 15 trials.

Multispecies hypercycles. occur in 14 trials, when there appears to be a mutual dependence among more than two chemical species, as shown in figure 4.

Detailed evaluation of a single trial

We present here details of one of the more interesting sequences of mutation that leads to a hypercycle of co-dependent molecular species. This was observed in trial 277 (figure 5), but hypercycles of one form or another occurred in 30 trials.

We classify this trial as an "emergent hypercycle". At $t = 748,199$ one of the eventual partners (species 31) is first produced via a mutation. This molecule exists as a sub-population for around $5,750,000$ time-steps before forming one partner in a co-dominant pair of molecular species. The

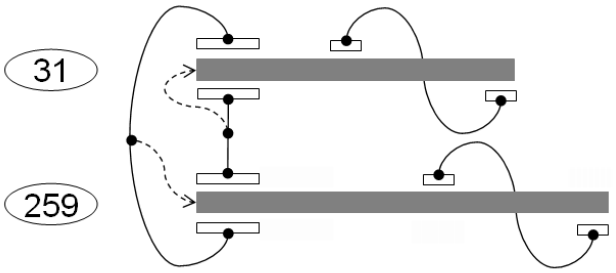


Figure 6: Reactions in the hypercycle. Molecules are represented by grey bars. Binding sites are shown as white boxes, with active binds shown above and passive binds shown below the molecule. Bind alignments are shown as black lines between molecules. Dashed lines show the product of the reaction (where one occurs).

partnership runs for approximately 3 million time steps before a parasitic molecule emerges to end the trial.

The molecular species in a hypercycle

The two molecular species (31 and 259) in the hypercycle are shown in figure 6. The bindings that occur between them are shown as black lines. The assignment of roles in the reaction (i.e. whether the molecule is passive (acts as the template) or active (acts as the program) occurs with equal probability for both molecules, meaning that for 50% of the time species 31 is produced and for the other 50% of the time species 259 is produced. Also note that species 31 is shorter than species 259 - it has lost one of the binding regions required for the reaction-program to initialise such that a copy of the replicase is created. This means it tends to be copied more quickly. Neither molecule is able to self-copy.

This phenomenon was neither foreseen in the original design nor expected to form without further design effort. It is particularly surprising that both partners in our hypercycle have no ability to self-copy. How could this have happened, and what is the evolutionary advantage of it?

Origin of the short partner

We need to explain how species 31, that is missing a key functional component, can rise to co-dominance in our system. We can trace the ancestry of the molecular species, and examine the reaction networks at key stages in any trial (figure 7). A white box indicates that a new species is synthesised *de novo* in the reaction, whereas a grey box indicates that the new species arises by modification of one of the reactants. Replicase molecules should act as catalysts, remaining unchanged when they emerge from a reaction. We can conclude that there is something in the reaction with molecules of species 29 that has produced species 30, which then reacts with species 9 to form species 31. The single

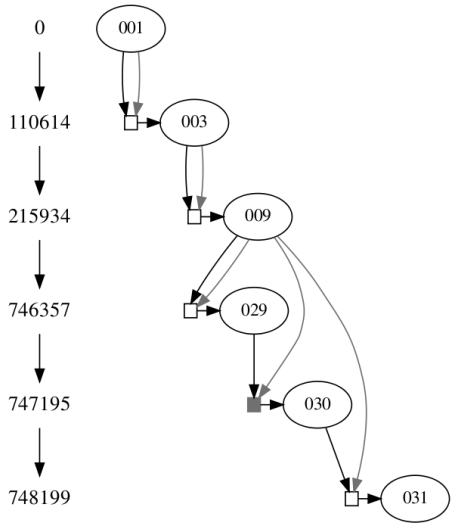


Figure 7: Ancestry of species 31. Numbers on the left indicate the time of reaction. Black arrows indicate the active partner. Grey arrows indicate the passive partner

point mutation of species 9 to create species 29 is shown below by a vertical line:

```
009 OBEQBX...LHHHRLUEUOBLROORE$BLUBOˆB>C$=?>$$BLUBO%}OYHOB
                                |
029 OBEQBX...LHHHRLUEUOBLROORE$BLUBPˆB>C$=?>$$BLUBO%}OYHOB
```

The subsequence `$BLUBO` has mutated to `$BLUBP`. The `$` symbol is a code for "seek", and (in this situation) positions the molecule's flow pointer at the end of the best complementary alignment for the sequence `BLUBO`, which is the sequence `OYHOB`. With the mutation in species 29, the alignment spans only the first four letters of `$BLUBO`, so the copy of the molecule is constructed one symbol in from the end of the molecule. When the construction is complete, the newly-created string must be cleaved from the active molecule's sequence. The pointers are arranged to achieve this via a second "seek" command with the same target (`OYHOB`). However, since the target has been overwritten in the original molecule, the seek command positions the pointer at the *end* of the newly copied molecule instead. The "cleave" command is applied to the far end of the string and is thus ineffective. The reaction-program terminates, and the new molecule (species 31) is created from most of a molecule of species 29 with a copy of species 9 pasted over the penultimate symbol.

In this manner, the reaction between species 29 and 9 creates species 30, which is nearly twice as long as the seed replicase, as shown in figure 8. Note there is only ever a single molecule of species 29, which is immediately transformed into species 30 when it reacts with a molecule from species 9. When species 9 binds to species 31, the bind site is shifted to a new position, as shown in figure 8. This changes the action of the replicase program such that the first 14 characters of the string are not copied. In this way,

```
030 OBEQBXUUUDYGRHBBOSEOLHHHRLUEUOBLROORE$BLUBPˆB>C$=?>$$BLUBO%}OYHOOBEQBXUUUDYGRHBBOSEOLHHHRLUEUOBLROORE$BLUBOˆB>C$=?>$$BLUBO%}OYHOB
Bind site:                                                  |--------------|
009                                                 OBEQBXUUUDYGRHBBOSEOLHHHRLUEUOBLROORE$BLUBOˆB>C$=?>$$BLUBO%}OYHOB
Product:                                                    |---------------------------------------------|
031                                                         BBOSEOLHHHRLUEUOBLROORE$BLUBOˆB>C$=?>$$BLUBO%}OYHOB
```

Figure 8: Origin of species 31

the single instance of species 30 can create many molecules of species 31 until it decays. Species 31 is then copied by dominant species in the system in 50% of reactions with it. Note that this cascade of reactions all occurs as a result of the single-point mutation on species 9.

Evolutionary pressure towards a hypercycle

Having established how a shorter molecule can arise via single-point mutations, we need to investigate how the molecule persists in the system, and what evolutionary pressure there is towards the formation of a hypercycle. It is important to note that in our replicase system a molecule that ensures it will always act as the template in a reaction is likely to sweep the population, as it will increase in numbers whenever it binds to another molecule. This is often achieved by *reducing* the bind probability for self-self reactions: as long as a bind is sufficiently likely, all the energy available in the system can be consumed. Binds stronger than this critical value have no advantage, whereas increasing any bias towards becoming the template in a reaction is clearly advantageous. For single-replicase systems, this is straightforward to understand, but with the introduction of species 31, the dynamics get more interesting.

Once present in the system, species 31 becomes a resource for other molecules. In all of the reactions with species 31, the chances of acting as a template are 50-50 (since the position of the alignment is the same on each string). This means that new species that bind to 31 can use it as a resource for increasing their number, even though half the time they will be exploited by species 31 to maintain its own population. Through a series of sweeps, each new dominant species binds increasingly strongly to species 31, thus flushing the previous incumbent from the system. Any new species that binds *less* strongly to species 31 than the previous dominant species is unsuccessful: it loses in the competition to exploit a valuable resource. Once bind affinity to species 31 is maximised, the old strategy of weakening self-self binds to guarantee template status in a reaction takes over again.

These processes are illustrated in figure 9, which plots binding rates for new dominant species in trial 277. The plots show the changes in bind probabilities with each successive sweep of the population as illustrated in figure 5. The line labelled "Bind to self" shows the probability of self-self binding for each new dominant species. The line labelled "Bind to 31" shows the bind probability between the new dominant species and species 31. There are three phases.

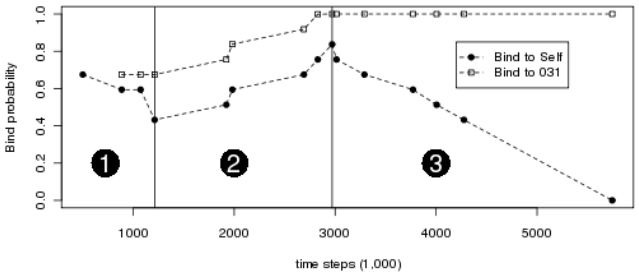


Figure 9: Change in binding rates as a precursor to hypercycle emergence

The first phase shows a decrease in self-binding probability between successive dominant species. We then see a second phase in which new species have an increasing affinity for binding to molecule 31. Once this is maximised, the third phase begins, in which successive dominant species sacrifice their self-bind probability to ensure they act as templates when reacting with the previous dominant species. In this way, dependence upon species 31 increases, until self-replication disappears altogether, and a hypercycle emerges.

The single-point mutations between dominant species are shown in figure 10. It shows that all mutations that confer an advantage occur in the binding regions of the molecule. Phases 1 and 3 of the run show changes in the second bind region, whereas phase 2 shows mutations in the first bind region. This corresponds with the change in phase noted for figure 9. The functional region of the molecule, which occupies the last half of the string, is preserved throughout. This is far from a random walk: the critical function of the replicase is preserved throughout, whilst a continual turnover of the binding site sequences illustrates the evolutionary pressure on the molecular species to act as a template for the molecule that the replicase builds.

Conclusions

We have presented an evaluation of the effect of mutation on an open-ended chemical system. The richness of behaviour we have shown is striking; indeed it was unexpectedly rich given that the only form of mutation is single-point. The need for such richness in complex systems was one of our main considerations during the design of this system. In addition, our chemistry reveals something of the dynamics of replicase systems that is very difficult to observe in biology. The decrease in binding affinity was not predicted, and the mechanism by which the hypercycle emerged was the result

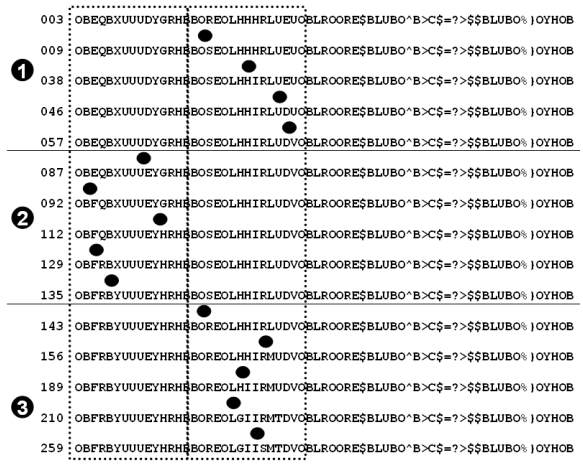


Figure 10: Mutations for the dominant species in run 277. Bind sites are indicated with dashed lines.

of a macromutation that was not "designed in" to the system.

Our replicase molecules are "imperfect replicators": they have a small chance of making an error when copying anything that binds to a certain region on the molecule. The imperfections in the copy process are not currently encoded on the genome; they are preset in the microcode of the copy instruction and thus unavailable for manipulation on the genome. In future work, we could represent the copy instruction at a finer level of granularity and use template codes to specify the accuracy of each sub operation, possibly including some cost for an increased accuracy of copy. We observed macro-mutations arising as a result of single-point changes that delivered emergent phenomena due to the wide heritable range of the system.

Finally, we must emphasise that these trials form a control experiment in which the effects of single-point mutation were evaluated. Future work will examine the effects of running a "population" of these trials, such that when a population of molecules collapses in an individual container, it can be replenished by a neighbour. This gives us a full model of early life, in which replicating templates and machinery self-maintain within membrane-bounded containers that can be replenished by neighbours.

Acknowledgements

This work was supported by EPSRC grant EP/F031033/1.

References

[1] Eigen, M. (1992). *Steps Towards Life: A Perspective on Evolution*. Oxford University Press.

[2] Eigen, M. and Schuster, P. (1977). The hypercycle, a principal of natural self-organization. part a: Emergence of the hypercycle. *Die Naturwissenschaften*, 64:541–565.

[3] Hickinbotham, S., Clark, E., Stepney, S., Clarke, T., Nellis, A., Pay, M., and Young, P. (2010). Specification of the stringmol chemical programming language version 0.2. Technical Report YCS-2010-458, Univ. of York.

[4] Hickinbotham, S., Clark, E., Stepney, S., Clarke, T., Nellis, A., Pay, M., and Young, P. (in press). Molecular microprograms. In *ECAL '09, LNCS 5777*.

[5] Hickinbotham, S., Clark, E., Stepney, S., Clarke, T., and Young, P. (2009). Gene regulation in a particle metabolome. In *CEC '09*, pages 3024–3031. IEEE Press.

[6] Hofstadter, D. R. (1979). *Godel, Escher, Bach: An Eternal Golden Braid*. Penguin.

[7] Johnson, T. J. and Wilke, C. O. (2004). Evolution of resource competition between mutually dependent digital organisms. *Artif. Life*, 10(2):145–156.

[8] Laing, R. (1977). Automaton models of reproduction by self-inspection. *J. of Theor. Biol.*, 66(3):437 – 456.

[9] Maynard-Smith, J. and Szathmary, E. (1997). *The Major Transitions in Evolution*. Oxford University Press.

[10] Ono, N. and Suzuki, H. (2002). Universal replication in a string-based artificial chemistry system. *Journal of Three Dimensional Images*, 16(4):154–159.

[11] Ray, T. (1991). An approach to the synthesis of life. In *Artificial Life II*, pages 371–408. Addison-Wesley.

[12] Smith, T. F. and Waterman, M. S. (1981). Identification of common molecular subsequences. *J Mol Biol*, 147(1):195–197.

[13] Turing, A. M. (1937). On computable numbers. *Proc. Lond. Math. Soc.*, 2:230–265.

[14] Varetto, L. (1998). Studying artificial life with a molecular automaton. *J Theor Biol*, 193(2):257–85.

[15] von Neumann, J. (1966). *Theory of self-reproducing automata*. University of Illinois Press.

[16] Wright, B. E. (2000). A biochemical mechanism for nonrandom mutations and evolution. *J Bacteriol*, 182(11):2993–3001.

Swarm Chemistry Evolving

Hiroki Sayama

Collective Dynamics of Complex Systems Research Group / Department of Bioengineering
Binghamton University, State University of New York
sayama@binghamton.edu

Extended Abstract

We report several recent extensions of Swarm Chemistry (Sayama 2008; Sayama 2009), an artificial chemistry model that uses kinetically interacting particle swarms as chemical reactants. Major modifications we newly implemented in the Swarm Chemistry model are as follows:

1. There are now two categories of particles, active (moving and interacting kinetically) and passive (remaining still and inactive). An active particle holds a recipe of the swarm (i.e., a list of kinetic parameter sets) in it (Fig. 1(a)).
2. A recipe is transmitted from an active particle to a passive particle when they collide, making the latter active (Fig. 1(b)).
3. The activated particle differentiates randomly into a type specified by one of the kinetic parameter sets in the recipe given to it (Fig. 1(c)).
4. Active particles randomly re-differentiate with small probability.

It has been demonstrated that these model extensions enable morphogenetic processes starting with a single particle containing a recipe (zygote) that grows into a fully developed self-organizing swarm pattern by "eating" other passive particles as raw materials through local recipe transmission (Sayama 2010). In addition, the stochastic re-differentiation introduced above (4) naturally achieves self-repair capability of swarms with simple open-loop linear control mechanisms (Sayama 2010).

Moreover, to demonstrate that macro-level ecological/evolutionary dynamics of self-organizing swarm patterns can arise out of micro-level processes embedded in particle interactions, we further introduced minimal mechanisms for variation and competition of recipes when they are transmitted between particles. Specifically, we implemented the following mechanisms to the model:

5. A recipe is transmitted between active particles of different types when they collide (*inheritance*). The direction of recipe transmission is determined by a competition function that picks one of the two colliding particles as a source (and the other as a target) of transmission based on their properties (*selection*) (Fig. 1(d)).
6. The recipe can mutate when transmitted (as well as spontaneously at other times) with small probability (*variation*) (Fig. 1(e)).

With these additional mechanisms, the Swarm Chemistry world has become capable of producing fully autonomous ecological and evolutionary behaviors of self-organized "super-organisms" made of a number of swarming particles. With a finite amount of resources (i.e., fixed number of particles) provided in a closed environment, we have observed behaviors of those macroscopic patterns that could be interpreted in ecological/evolutionary terms, such as reproduction, chasing, and predation, all emerging out of local interactions among individual particles (Fig. 1(f)).

We have tested a couple of different principles for the competition function, e.g.:

(i) The faster (or slower) particle wins (i.e., becomes the source).
(ii) The particle that hit the other one from behind wins.
(iii) The particle surrounded by more of the same type wins.

Each condition produced unique, distinct evolutionary dynamics. The most recent findings obtained from those different conditions are presented and discussed comparatively.

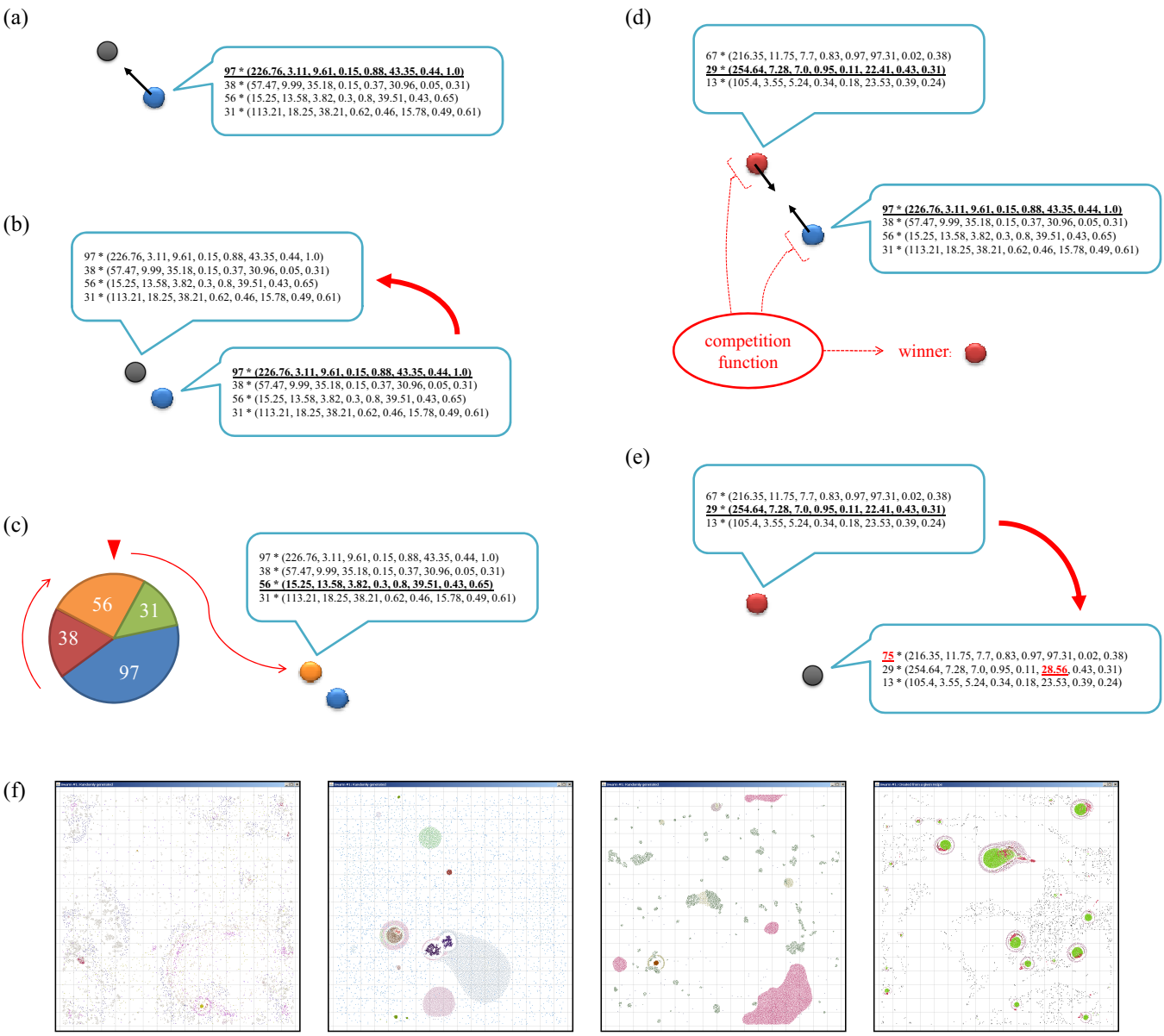


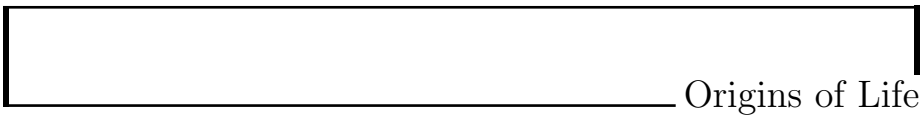
Figure 1: How particle interactions work in the revised Swarm Chemistry. (a) There are two categories of particles, active (blue) and passive (gray). An active particle holds a recipe of the swarm in it. (b) A recipe is transmitted from an active particle to a passive particle when they collide, making the latter active. (c) The activated particle differentiates randomly into a type specified by one of the kinetic parameter sets in the recipe given to it. (d) A recipe is transmitted between active particles of different types when they collide (*inheritance*). The direction of recipe transmission is determined by a competition function that picks one of the two colliding particles as a source (and the other as a target) of transmission based on their properties (*selection*). (e) The recipe can mutate when transmitted with small probability (*variation*). (f) Examples of ecologies of self-organizing patterns spontaneously formed in the Swarm Chemistry world (made of 10000 particles each).

References

Sayama, H. (2008). Self-organizing heterogeneous swarms designed through evolutionary methods. In Bullock, S., Noble, J., Watson, R. and Bedau, M. A., editors, *Artificial Life XI*, page 801. MIT Press, Cambridge, MA.

Sayama, H. (2009). Swarm chemistry. *Artificial Life*, 15:105–114.

Sayama, H. (2010). Robust morphogenesis of robotic swarms. *IEEE Computational Intelligence Magazine*, in press.


Origins of Life

Temperature Limit for the Emergence of Life-like System Deduced from the Prebiotic Chemical Kinetics under the Hydrothermal Conditions

Kunio Kawamura[1]

[1]Department of Applied Chemistry, Osaka Prefecture University
Gakuen-cho 1-1, Naka-ku, Sakai, Osaka, 599-8531, Japan.
kawamura@chem.osakafu-u.ac.jp

Abstract

The RNA world hypothesis and the hydrothermal origin of life hypothesis are contradictory to maintain life-like systems for these two hypotheses to be compatible by the following two main reasons. First RNA molecules are too labile and second the biologically important interactions would not be effective at high temperatures. The assumption can be applied to the protein-based life-like systems. We have continuously investigated the stability and the chemical evolution of RNA- and protein-based life-like systems by using our hydrothermal-monitoring techniques. According to these data, it has been found that two viewpoints are essential to discuss the temperature limit of RNA and/or protein-based life-like systems on the primitive earth. First, the accumulation of biomolecules should be determined by both the formation and degradation rates. Second, the reaction rates of the primitive life-like systems should be evaluated from the viewpoint of enzymatic reaction rates.

Introduction

The RNA world hypothesis has some drawbacks despite being supported by empirical data such as chemical evolution experiments using RNA and in vitro selection technique generating artificial ribozymes (Lohrmann and Orgel, 1980; Gilbert, 1986; Joyce et al., 1987; Sawai et al., 1989; Ellington & Szostak, 1990; Ferris and Ertem, 1992; Terfort and von Kiedrowski, 1992; Kawamura and Ferris, 1994). That is to say, the hypothesis that life originated near hydrothermal vent environments (the hydrothermal origin of life hypothesis) appears to be inconsistent with the RNA world hypothesis. The hydrothermal origin of life hypothesis was proposed based on the continuous investigations of thermophilic organisms (Corliss et al., 1981; Baross and Hoffman, 1985) and phylogenetic analysis of present organisms. The last common ancestor (LCA) is considered to have been a thermophilic organism (Pace, 1991; Forterre, 1994) although this is still disputed (Miller and Bada, 1988; Galtier et al., 1999).

It has been frequently concluded that RNA molecules are too labile under hydrothermal vent conditions for these two hypotheses to be compatible. Furthermore, biologically important weak interactions such as hydrophobic interactions and hydrogen bonding are weaker at higher temperatures. However, the most of simulation experiments have been carried out at low temperatures. In addition, there have been no practical techniques for the investigations of chemical evolution of RNA under hydrothermal conditions. These situations can be applied to the case that the protein-based life-like systems, such as GADV protein hypothesis (Ikehara, 2005), since the half-lives of proteins under hydrothermal environments are much shorter than the geological time scale; the formation of protein-like molecules has been examined under simulated hydrothermal vent conditions (Holm, 1992; Marshall, 1994; Imai et al., 1999; Kawamura et al., 2005).

Naturally, it is difficult to determine the temperature at which life originated while it is estimated that life on Earth originated 4600 to 3500 million years ago (Mojzsis et al., 1996). Frequent meteorite impacts could have raised the Earth's temperature significantly (Maher and Stevenson, 1988). Alternatively, some evidence suggests that the primitive ocean was frozen since the solar luminosity at that time was relatively less than at present (Sagan and Mullen, 1972). Thus, the temperature of the primitive ocean in which life originated remains speculative (Walker, 1985; Kasting and Ackerman, 1986).

Thus, investigations are required to evaluate the RNA- and/or protein-based life-like systems at different temperatures although the chemical evolution of RNA has been mainly studied at low temperatures. We have continuously studied the stability and prebiotic formation of RNA and protein-like molecules at high temperatures by using our monitoring methods of hydrothermal reactions. The systematic analyses of these data would provide insight into the possibility of a life-like system under hydrothermal conditions.

Conclusively, it has been found that the following viewpoints are essential to discuss the temperature limit of RNA and/or protein-based life-like systems on the primitive earth and to determine whether biomolecules are sufficiently stable or not under hydrothermal conditions.

View I: The accumulation of biomolecules should be evaluated under the thermodynamically open system, so that the accumulation of biomolecules should be determined by both the formation and degradation rates. Our experimental data suggested that the formation of RNA would be possible once the elongation of RNA starts from oligonucleotides longer than dimer at very high temperatures.

View II: The rate of the primitive reactions within the primitive life-like systems should be evaluated from the viewpoint of enzymatic reaction rates. Based on the

comparison between the reaction rates with and without enzyme, much higher temperature limit, such as 300 ºC, can be assumed for the emergence of a life-like system.

Chemical Evolution of Biopoymers under Hydrothermal Conditions

Monitoring methods of hydrothermal reactions

While hydrothermal reactions were normally investigated using batch reactors, it was difficult to monitor hydrothermal reactions within the millisecond to second time scale. To monitor such rapid reactions, the flow systems monitoring of hydrothermal reactions are becoming practical techniques (Kawamura, 2000; 2002). We have invented a real-time and *in situ* monitoring method of hydrothermal reactions using a micro-flow reactor system assembled with fused-silica capillary tubing, which enables the monitoring reactions at 0.002 - 200 s at 400 ºC at 50 MPa. For *in situ* monitoring of hydrothermal reactions, an optical window on the fused-silica capillary and a UV-visible detector are connected with high temperature-resistant optical fibers; it enables monitoring of 200 – 900 nm at 0.08 – 3.2 s at 400 ºC.

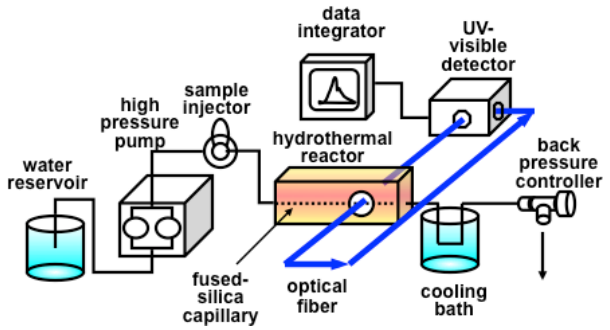


Figure 1. Hydrothermal flow reactor system using fused-silica capillary tubing. This system enables real-time monitoring and *in situ* UV-visible monitoring.

RNA- and protein-based life-like system

Discovery of ribozyme suggested that RNA-like molecules had a central role in the first life on earth. The plausible information flow in a life-like system consisting of RNA molecules is shown in Figure 1, where RNA molecules preserve both information and enzymatic activities. RNA world hypothesis is supported by chemical evolution experiments using RNA formation models (Lohrmann and Orgel, 1980; Joyce et al., 1987; Sawai et al., 1989; Ferris and Ertem, 1992; Terfort and von Kiedrowski, 1992). Activated nucleotide monomers (5'-phosphorimidazolide of nucleoside) were synthesized in the laboratory as model activated prebiotic nucleotide monomers that might also be formed under primitive Earth conditions and could produce RNA oligonucleotides (Lohrmann and Orgel, 1973). This technique has been successfully applied to the formation of RNA in the presence of polynucleotide template, metal catalyst, and clay mineral catalyst. On the other hand, the fact that *in vitro* selections can produce several ribozymes and aptamers support the speculation that different functional RNA could have spontaneously formed on primitive earth (Ellington and Szostak, 1990; Tuerk and Gold, 1990) although the same molecular machinery, which is used in the modern *in vitro* selection techniques, was not present on primitive Earth.

Naturally, proteins are important for the emergence of life-like systems while it is generally considered that proteins could not preserve biological information as RNA and DNA preserve information on the basis of Watson-Crick base-pair formation. In addition, simulation experiments on primitive Earth imply that the formation of protein-like molecules would be easier than that of RNA on the primitive Earth although it is indeed difficult to determine which formation of RNA or proteins is more difficult. The reason that the formation of proteins is frequently regarded to be easier than that of RNA may be due to the fact that the formation of RNA monomers consists of three steps of nucleotide bases, nucleoside, and nucleotides while amino acids are directly formed from primitive gas mixture using different energy sources.

Recently, GADV protein hypothesis has been proposed on the basis of analyses of the relationship between the structures of water-soluble granular proteins and nucleotide base compositions of genes regarding present organisms (Ikehara 2005). Conclusively, this hypothesis suggests that glycine (G), alanine (A), aspartic acids (D), and valine (V) could have been the most primitive protein, which could have formed a simpler transcription systems. The importance of the combination of G, A, D, V has been sometime pointed out from different viewpoints (Eigen et al., 1981). In addition, the difficulty that proteins would not readily preserve genetic information might be solved by assuming that the information

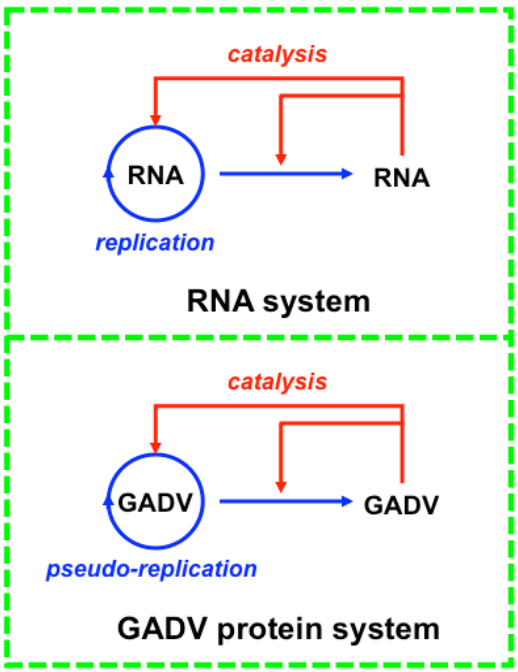


Figure 2. Prebiotic information flow for life-like systems consisting of RNA molecules and proteins.

could have been preserved on the basis of pseudo-replication mechanism. A possible information flow is illustrated in Figure 2, where GADV proteins could behave similarly to those assumed for an RNA based life-like system.

Naturally, this protein-based origin-of-life hypothesis should be evaluated from the viewpoint of hydrothermal origin of life hypothesis.

Prebiotic formations and stabilities of RNA and proteins under hydrothermal conditions

While simulation experiments for the formation of RNA oligomers on the primitive earth conditions have been extensively investigated using the phosphorimidazolides of nucleotide monomers (Lohrmann and Orgel, 1980; Joyce et al., 1987; Sawai et al., 1989; Ferris and Ertem, 1992), the most of studies were carried out at 25 ºC. Thus, we have investigated kinetic analyses of prebiotic formation models of RNA using the activated nucleotide monomers or water-soluble carbodiimide as a condensation reagent at temperatures up to 100 ºC. We successfully analyzed the following models (Figure 3), (1) the template-directed formation of oligoguanylate on a polycytidylic acid template (TD reaction) (Kawamura and Umehara, 2001), (2) the cyclization of oligonucleotides (CY reaction) (Kawamura et al., 2003), (3) the oligocytidylate formation in the presence of Pb^{2+} (ME reaction) (Kawamura and Maeda, 2007), and (4) the oligocytidylate formation in the presence of montmorillonite clay (CL reaction) (Kawamura and Maeda, 2008). The formations of oligonucleotides using these model reactions are basically difficult at high temperatures. Based on these empirical data, it was generalized that the reactions are expressed by the scheme shown in Figure 4.

The accumulation of oligonucleotides is determined by the relative magnitude of the processes. The kinetic analyses of the 4 types of RNA formation models suggested that the low efficiency of oligonucleotide formation at high temperatures is mainly due to the weak association between an activated nucleotide monomer and an elongating oligonucleotide since hydrogen bonding and hydrophobic interaction decrease with increasing temperature. This trend was observed for all the 4 types different prebiotic reactions. For the cases of TD, ME, and CL reactions, it is generally found that the association between an activated monomer and a monomer (or another activated monomer) for the formation of dimer becomes weak and the relative rate of the formation of dimer decreases notably as comparing to trimer and tetramer formations. On the contrary, for the cyclization of a linear oligonucleotide the association of 3'- and 5'-terminals is much easier since it is an innermolecular reaction. Thus, the rate constants of cyclization do not decrease notably as comparing to those of cleavage of phosphodiester bonding; naturally the cyclization of oligonucleotides would be disadvantageous for the formation of long oligonucleotides. According to these data, it was implied that the oligonucleotides could have formed at high temperatures if the association between the activated nucleotide monomer and the elongation oligonucleotide is facilitated by additives, such as, protein like-molecules, mineral surfaces, metal ions.

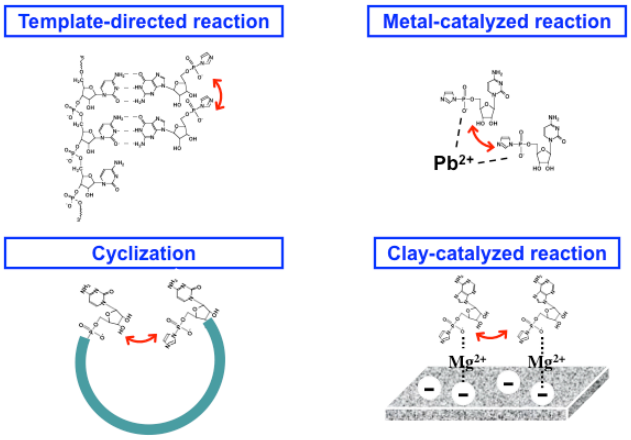


Figure 3. RNA formation models with and without activated nucleotide monomers.

On the other hand, it has been shown that the formation of protein-like molecules is possible under different conditions. Thermal condensations of amino acids mixtures including Asp and Glu have been frequently investigated as a formation model of protein-like molecules on the simulated dry surface model of primitive Earth (Fox and Harada, 1958). The formation of peptides was investigated in the presence and absence of condensation reagent while the investigations of peptide formation have been relatively weak as comparing to the formation of RNA (Ferris, et al., 1996).

The formation of protein-like molecules is also possible even under the hydrothermal conditions in the absence of condensation reagent while the efficiency is lower than that of the dry model (Imai et al, 1999). Actually, the formation of proteins from amino acids under hydrothermal conditions is not so easy, where the yield of oligopeptides formation is typically 0.1 – 1 %. One reason is that the dehydration of amino acids is principally difficult in aqueous solution. In addition, the cyclization of dipeptide to form diketopiperazine inhibits the further elongation of oligopeptides. Furthermore, the condensation reagent would facilitate the formation of oligopeptides while suitable prebiotic condensation reagents have not been yet discovered for the oligopeptide formation under hydrothermal conditions (Kawamura et al., 2009a). Based on our investigation of a condensation reagent for the formation of oligopeptides, it was found that the condensation

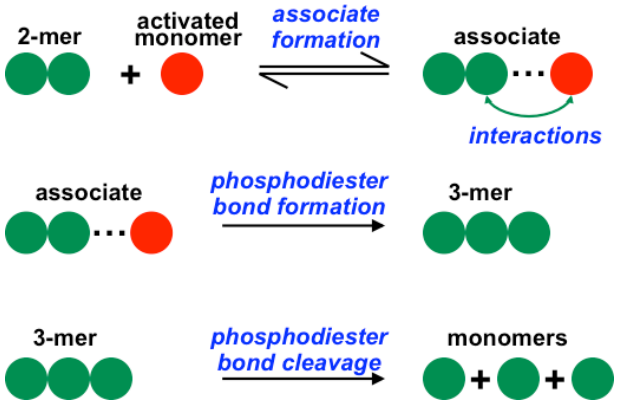


Figure 4. Generalized reaction model for the formation of prebiotic RNA molecules.

reagent is immediately destroyed under hydrothermal conditions.

By using the hydrothermal flow reactor, we have discovered two possible pathways, which enhance the elongation of oligopeptide. First, the elongation of oligoalanine readily proceeds within 10 – 30 sec at 250 – 330 ºC if its starts from 4-mer oligoalanine and longer (Kawamura et al., 2005). The efficiency of the formation of oligoalanine reaches to 10 %. Second, one-step formation of oligopeptides including 20 amino acids unit from Asp and Glu is possible within 3 min at 275 ºC (Kawamura and Shimahashi, 2008). However, it is generally true that the oligopeptides are not basically stable at high temperatures so the oligopeptides could not survive even if hydrothermal vent systems facilitate the formation of oligopeptides. Thus, it has been frequently proposed that oligopeptides could have accumulated in the surrounding cool ocean once the peptides are evacuated from the hydrothermal vent (Imai et al., 1999).

We have carried out kinetic investigations of the degradation of nucleotide bases, nucleosides, nucleotides, oligonucleotides, polynucleotides, amino acids, peptides, and proteins. The fastest process for the degradation of nucleotides as RNA monomers is the cleavage of triphosphate of nucleotides (Kawamura, 2000). Besides, the fastest process for the degradation of amino acids is racemization (Kawamura and Yukioka, 2001). The cleavage of phosphoester bonding is approximately 10000 times faster than that of racemization of amino acids. Moreover, the cleavage of phosphodiester bonding of RNA is approximately 100 times faster than that of peptide bonding (Kawamura, 2003a, 2003b; Kawamura et al., 2005). These facts indicate that the RNA and nucleoside monomers are less stable as comparing to proteins and amino acids. However, it should be noted that these reactions proceed within much shorter time scale than the geological time scale. For instance, the ribonuclease loses the catalytic activity within 30 s at 275 ºC (Kawamura et al., 2009b). This fact suggests the importance how to judge the stability of these biomolecules.

Table 1. Half-life calculated by the real-time monitoring of hydrothermal degradation for biomolecules.

half-life / s	Temperature / ºC		
	100	200	300
oligo17	4500	3.08	0.0268
$C^{3'}$pG	12900	28.8	0.542
$C^{2'}$pG	14100	37.4	0.789
dCdG	572000	45.7	0.0981
ATP	1290	0.37	0.00187
ADP	6830	1.61	0.0070
AMP	83500	8.65	0.022
adenosine	1610000	86.9	0.145
alanine	15900000	3380	13.7

Values of half-life were obtained from the previous investigations (Kawamura, 2000, 2003a, 2003b, Kawamura and Yukioka, 2001).

Interactions of biopolymers under hydrothermal conditions

Biologically important interactions, such as hydrogen bonding, hydrophobic interactions, π–π stacking, would decrease with increasing temperatures. However, it was normally difficult to analyze such interactions by using conventional techniques. Thus, we have attempted to measure such weak interactions of RNA and proteins using our *in situ* UV-visible monitoring system for hydrothermal reactions (Kawamura and Nagayoshi, 2007; Kawamura et al., 2010). Our accumulated data support quantitatively the assumption that the weak interactions, such as hydrogen bonding, hydrophobic interaction, π–π stacking, becomes weak.

It was confirmed that double-stranded DNA is readily denatured to single-stranded DNA at temperature lower than 100ºC by using our system (Kawamura and Nagayoshi, 2007). However, at higher temperatures it was found that single-stranded DNA form aggregate at higher temperatures up to around 200 ºC, where the solubility of DNA becomes low especially in the presence of Mg^{2+}. At higher temperatures, single-stranded DNA is cleaved so the solubility increases. This fact suggests that the solubility of DNA is an important factor to determine the limit temperature for life-like systems.

On the other hand, the interactions between proteins and chromogenic reagents were investigated using the *in situ* UV-visible monitoring system (Kawamura et al., 2010). Among a few kinds of proteins, the interaction of bovine serum albumin (BSA) with a water-soluble porphyrin (TPPS) was possible to investigate up to 150 ºC. The association constant between BSA and TPPS at 100 ºC was ca. 100 times smaller than that at 25 ºC. However, the interaction of TPPS with pyridine bases is not so reduced within this temperature range, where the association constants decrease only 2 – 6 times. Thus, we concluded that the decrease of the association constant of BSA with TPPS is due to the conformational change or denaturation of BSA at high temperatures. That is to say, BSA is a modern enzyme so that this is not suitable to interact with substrates at high temperatures, where denaturation occurs. Thus, it is important to investigate the interactions of prebiotic protein-like molecules with defferent substrates.

Temperature Limits of Primitive Life-life Systems

Viewpoints to determine whether prebiotic molecules are sufficiently stable

The viewpoints to determine whether biopolymers are stable or not have been briefly discussed regarding the RNA world in the previous publications (Kawamura, 2004). It is assumed that the conditions necessary for the emergence of a life-like system consisting of RNA and/or proteins are as follows: (1) a sufficient amounts of biomolecules are accumulated, (2) biological information is replicated, (3) a set of chemical assemblies controlling the rate of the reactions (primitive enzymes) exists within the system, and (4) the compartment of these chemicals would be necessary in a cell or a single unit. In modern organisms, for instance, RNA molecules are synthesized by RNA polymerases and degraded by ribonucleases. Similarly, in the RNA world, both the formation and degradation of RNA molecules had to be controlled by primitive enzyme-like molecules. Naturally, this principle should be applied to the case of life-like systems based on proteins or protein-like molecules. That is to say, the

accumulation of protein-like molecules would be controlled in the presence of primitive ribosome with a set of primitive enzymes, such as aminoacyl tRNA synthetase and primitive protease-like molecules. Thus, the following two views should be applied to examine the possibility of the accumulation of biopolymers. The term to express the restriction conditions was called as "Scale" in the previous paper for the purpose that we will find a way to determine quantitatively such scales on the basis of the reaction rate. The term "View" is used in the present paper.

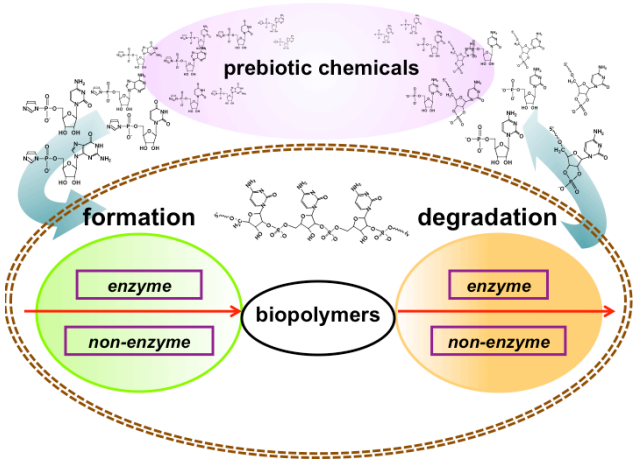


Figure 5. Accumulation of biopolymers kinetically controlled by the formation and degradation rates in the presence and absence of prebiotic enzymes.

View I: The accumulation of prebiotic biopolymers should be evaluated from the viewpoint of kinetics of the accumulation of prebiotic polymers. As mentioned above, the accumulation of biopolymers in a cell is determined by the formation + inflow and the degradation + outflow. This is illustrated as shown in Figure 5. Here, for simplification the sum of formation and inflow is called as formation and that of degradation and outflow is called as degradation. If the hydrothermal origin-of-life hypothesis is correct, the relative rates of biopolymer formation and degradation should determine the accumulation of the biopolymers under hydrothermal vent conditions as well as under mild conditions. If such primitive enzymes had existed on primitive Earth, the accumulation would had been possible without considering a pathway for surviving of biopolymers in the surrounding cool ocean for the biopolymers formed in the hydrothermal vent system.

View II: Since enzymes control reactions in modern organisms the rate of reactions in primitive life-like systems should be evaluated from the standpoint of possible primitive enzymatic reaction rates. The importance of the fact that enzymatic reaction rates are generally much greater than the uncatalyzed reaction rates has been addressed (Radzicka & Wolfenden, 1995). It is no doubt that enzymes are essential for controlling biological reactions in living systems. Based on this viewpoint, the comparison of the reaction rates with and without prebiotic enzymes should be essential for the evaluation of primitive life-like systems.

Possibility of RNA- and protein-based life-like systems at high temperatures

On View I, our data regarding the prebiotic formation of oligonucleotides show that the phosphodiester bond formation could be faster than that of the decomposition even at high temperature as mentioned above. Thus, these reaction models indicate that the oligonucleotides could have formed at high temperatures. As mentioned above, a strong association between the activated monomer and the elongating oligomer is required for the formation of the phosphodiester bond on the basis of the model shown in Figure 4. While we could have not detected prebiotic additives to facilitate the association, it is anticipated that the acceleration of phosphodiester bond formation with a strong association would be possible. Actually, chemical assemblies to enhance the association should exist at least up to 110-120 ºC in modern hyperthermophilic organisms (Stetter, 1982; Kashefi and Lovley, 2003). Presumably, potential prebiotic catalysts, such as protein-like molecules, clay minerals, and metal ions, could have facilitated the association of the monomer and the elongating oligomers for RNA based life-like systems. In addition, the supply of a sufficient concentration of the activated monomers that would be formed from bases, ribose, inorganic phosphate, and imidazole should be taken into account. View I is also applied for the accumulation of these resources for RNA molecules although the experimental evaluation would be difficult for simulating consecutive chemical evolution through these resources.

Table 2. Limit temperatures where the rate of oligonucleotide formation is faster than that of degradation.

Reactions	Limit temperature (ºC)
TD reaction	309
CY reaction	382
CL reaction	162

Calculations were performed on the basis of our previous investigations (Kawamura and Umehara, 2001; Kawamura et al., 2003; Kawamura and Maeda, 2008).

The temperatures where the formation of RNA becomes comparable to that of degradation of RNA were calculated on the basis of our previous data as shown in Table 2, where it is dependent on the type of prebiotic reaction models. The temperatures for CY reaction and TD reaction are somewhat higher than those of CL reaction. This value was not obtained for ME reaction. This is probably correlation with the yield of the phosphodiester bond formation, where the yields regarding TD and CY reactions are greater than those for CL and ME reactions. The association between two moieties to form phosphodiester bonding in CY is much easier than other reactions because it is innermolecular association. The association for TD reaction is efficient than that for CL and ME reactions. This finding suggests that the formation rate of RNA would be faster than the degradation rate of RNA even under hydrothermal conditions. The magnitude of the temperatures for these models is consistent with the efficiency of the phosphodiester bond formation of these reaction models.

For the case of protein-based life-like systems, several condensation reagents would facilitate the peptide bonding formation. However, there has been no data regarding

temperature dependence of the primitive formation rates of proteins from amino acids in the presence of condensation regents or by using activated amino acids.

On View II, enzymes control biological reactions in modern organisms. However, it is noted that the reactions can proceed even at very slow rates without enzymes as background reactions in organisms. The importance of this principle has been pointed out, where the ratio (k_{cat}/k_{non}) of the enzymatic reaction rate (k_{cat}) to the background reaction rate (k_{non}) represents the catalytic ability of the enzyme (Radzicka and Wolfenden, 1995; Kawamura, 2004). This fact indicates that the strong specificity of an enzyme to a substrate is due to the reduction of the activation energy for the enzymatic reaction. Thus, the specificity of enzyme is strongly dependent on the temperature since the background reaction rate increases with increasing temperature. Here, it is still difficult to compare the background rates of the reactions catalyzed by modern enzymes with that of primitive enzymes because the catalytic rate enhancement of primitive enzymes is unknown. The comparison of background reaction rates with modern enzymatic reaction rates was examined for thermophilic reactions in the previous study. In the present study, a continuous investigation on the basis of this concept has been carried out. The relationship between the enzymatic reactions and background reactions is illustrated in Figure 6.

In addition, there is a trend, which would be found even in biochemical text books, the magnitudes of the rate constants (k_{cat}) of reactions catalyzed by several enzymes are relatively narrow range of 10^{2} – 10^{6} s^{-1} while the uncatalyzed background rate constants (k_{non}) are in the range of 10^{-16} – 10^{0} s^{-1} (Radzicka and Wolfenden, 1995). We showed a similar relationship for the cases of ribonucleases and a RNA polymerase (Kawamura, 2004). Furthermore, we have examined the rate constants compiled from literature sources for several thermophilic enzymes (Kawamura, 2004), which was possible to incorporate the rate constants that were within the same range of other enzymatic rate constants. In addition, the rate constants (k_{cat}) of thermophilic enzymes do not largely differ from those of enzymes from mesophiles. According to this analysis, there is a general trend that the reaction rates with modern enzymes including thermophilic enzymes are in a relatively narrow range compared to the range of the background reaction rates. Conclusively, the enzymatic rate constants including mesophiles and thermophiles are shown in a trapezoid at the top-left corner and the uncatalyzed background rate constants are shown in a large trapezoid at the bottom (Figure 6).

The difference between the reaction rate with and without primitive enzymes should have been necessary for the accumulation of biopolymers. This principle would provide a

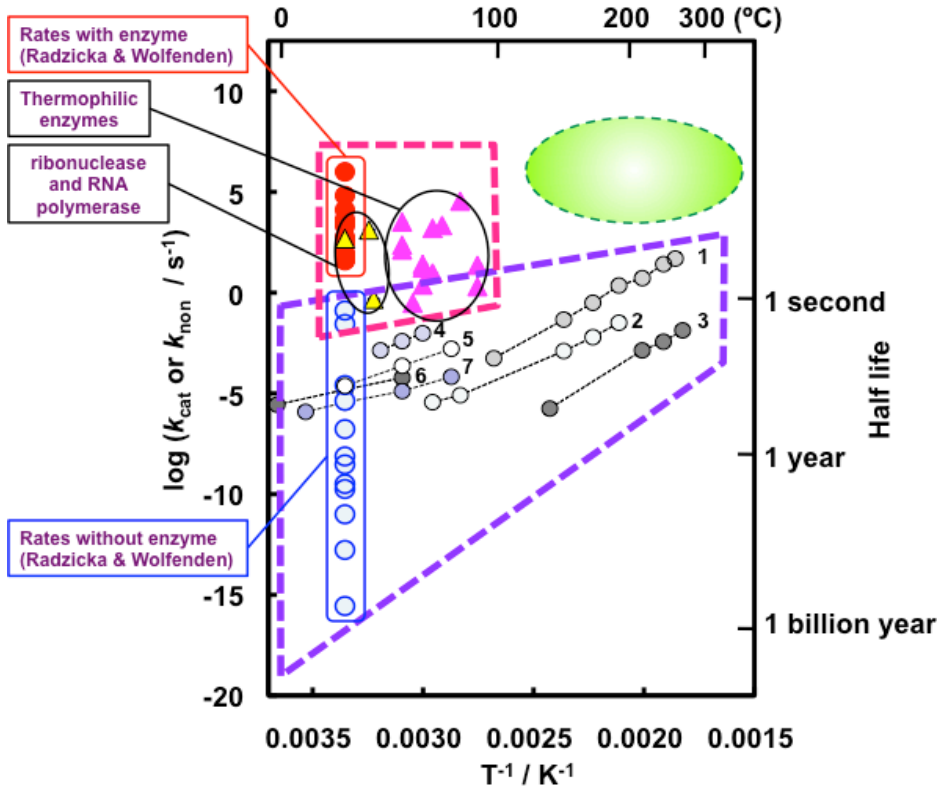


Figure 6. Comparison of the reaction rate with enzymes and without enzymes regarding prebiotic reactions. The horizontal axis indicates inverse values of temperature (T^{-1}) and the vertical axis indicates logarithmic values of reaction rates. The numbers show the reaction rates determined by our studies. 1: ATP hydrolysis, 2: $C^{3'}$pG cleavage, 3: racemization of alanine, 4: 4-mer formation by TD reaction, 5: cyclization of d(pGCGCG)rC, 6: CL 4-mer formation by reaction, 7: 3-mer formation by ME reaction. Top-right corner (green circle) would indicate the limit temperature and enzymatic reaction rate regarding the origin of life.

temperature limit for the primitive life-like system, where primitive enzymes could facilitate the target reactions with faster rates than the background reactions; this had to be chemically possible at the limit temperature. Besides, it is known that a fastest process in aqueous solution is proton transfer so that the enzymatic reaction could not be faster than the proton transfer process in aqueous medium. The proton transfer rates are plotted over the upper limit of enzymatic reaction rates. In addition, the interaction of a candidate biopolymer of primitive enzyme with a primitive substrate would decrease with increasing temperature.

This implies a weak specificity and an enhancement of the primitive enzymatic reaction. Naturally, there is no basis to determine how much difference between k_{cat} and k_{non} should have been essential for the primitive enzymes to construct a most primitive life-like system. Nevertheless, even a small difference between primitive enzymatic rates and background rates could be considered as candidates for a primitive enzyme activity. The large difference between the enzymatic rates and the background rates even at very high temperatures at the top-right corner (green circle) is impressive, where the background reaction rates merge to the extrapolation of modern enzymatic reaction rates. This might reflect that the evolution of enzymatic activities would have synchronized with the decrease of temperature.

By the way, the associate formation for elongating biopolymers would be facilitated by different additives while the assumption is now being evaluated. In addition, the compartment of chemicals for a life-like system would be also very important if we assume that the life-like system could have survived under hydrothermal conditions. In a compartment, that is, a cell, several advantageous are expected for the emergence of life-like systems (Figure 7). Chemical reactants could be concentrated so that the interactions among prebiotic chemicals would be enhanced. In addition, the stabilities of biomolecules would be facilitated by the associate formation with concentrated additives.

To evaluate the possibility of spontaneous formation of enzymatic activities in protein-like molecules, we have investigated kinetics of primitive enzymatic functions of protein-like molecules mainly focusing to the formation and degradation of RNA molecules; the protein-like molecules were prepared by the simulation reactions of amino acids condensation under dry conditions and hydrothermal conditions (Kawamura et al, 2004). However, no notable

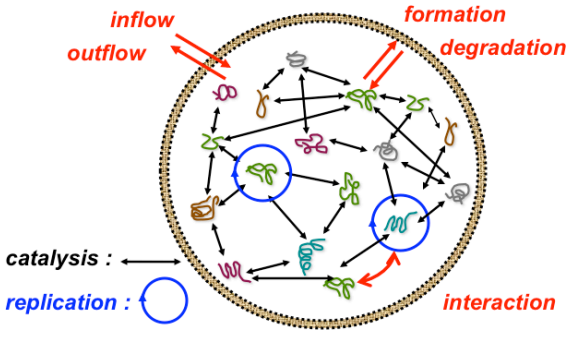


Figure 7. Compartment of biopolymers is important to facilitate the interactions between biomolecules, which would result efficient accumulation of biopolymers.

enzymatic activities have been detected so far within such randomly formed peptide-like molecules although a series of catalytic activities have been observed during the investigations of proteinoids (Fox, 1986). Less activity of protein-like molecules might suggest that enzymatic functions would have started from a very small catalytic effect and specificity at the initial stage.

Conclusions

This paper proposes the viewpoints to evaluate whether biopolymers, RNA and proteins, are compatible with primitive hydrothermal vent conditions. On View I, the relative magnitudes of the rates of degradation and formation of RNA were evaluated. The TD, CY, and CL reactions showed the fairly high temperatures where the rate of RNA formation could be greater than the rate of degradation. Naturally, chemical assemblies would have been required to facilitate the association to form biopolymers. On View II, the stabilities of biopolymes were evaluated based on the comparison between non-enzymatic and enzymatic reaction rates. The evaluation suggests that a life-like system consisting of RNA and/or proteins is possible at fairly high temperatures above 100 ºC.

In addition, the interactions and three-dimensional folding of biopolymers are important factor to determine the limit temperatures for life-like systems. From this viewpoint, the interactions between molecules would provide a limit temperature as well as View I and View II. Furthermore, the solubility of biopolymers is also important factor to determine the limit temperature for a life-like system. To evaluate the assumptions shown in the present paper, the experimental data on the kinetic accumulation of biopolymers, the primitive replication of RNA (possibly pseudo-replication by GADV proteins) and the primitive enzymatic functions under hydrothermal conditions should be explored in the future.

Acknowledgments

This study is supported by Grant-in-Aid for Scientific Research 21200004 (2009-2011) on Innovative Areas from the Ministry of Education, Culture, Sports, Science and Technology (MEXT) and Grant-in-Aid for Scientific Research 20540476 (2008-2010) from Japan Society for the Promotion of Science (JSPS).

References

Baross, J. A. and Hoffman, S. E. (1985). Submarine hydrothermal vents and associated gradient environments as sites for the origin and evolution of life. *Origins Life*, 15:327-345.

Corliss, J. B., Baross, J. A., and Hoffman, S. E. (1981). An hypothesis concerning the relationship between submarine hot springs and the origin of life on earth. *Ocean. Acta*, 4:59-69.

Eigen, M., Gardiner, W., Schuster, P., and Winkler-Oswatitsch, R. (1981). The origin of genetic information. Sci. Am., 244:88-118.

Ellington A. D. and Szostak, J. W. (1990). In vitro selection of RNA molecules that bind specific ligands. Nature, 346:818-822

Ferris, J. P. and Ertem, G. (1992). Oligomerization of ribonucleotides on montmorillonite reaction of the 5'-phosphorimidazolide of adenosine, *Science*, 257:1387-1389.

Ferris, J. P., Hill, A. R., Liu, J. R., and Orgel, L. E. (1996). Synthesis of long prebiotic oligomers on mineral surfaces. Nature, 381:59-61.

Forterre, P. (1995). Thermoreduction, Hypothesis for the origin of prokaryotes. *C. R. Sciences de la vie/Life sciences, Acad. Sci. Paris*, 318:415-422.

Fox, S. W. (1986). The evolutionary sequence: origin and emergences. *The American Biology Teacher*, 48:140-149,169.

Fox, S. W. and Harada, K. (1958). Thermal copolymerization of amino acids to a product resembling protein. *Science* ,128:1214–1215.

Gilbert, W. (1986). The RNA world. *Nature,* 319:618-618.

Galtier, N., Tourasse, N., and Gouy, M. A. (1999). Nonhyperthermophilic common ancestor to extant life forms. *Science*, 283:220-221.

Holm, N.G. editor (1992). Special issue-marine hydrothermal systems and the origin of life. *Origins Life Evol. Biosphere*, 22:5-242.

Ikehara, K. (2005). Possible steps to the emergence of life: The [GADV]-protein world hypothesis. *Chem. Rec.*, 5:107-118.

Imai, E., Honda, H., Hatori, K., Brack, A., and Matsuno, K. (1999). Elongation of oligopeptides in a simulated submarine hydrothermal system. *Science*, 283:831-833.

Joyce, G. F., Shwartz, A. W., Miller, S. L., and Orgel, L. E. (1987). The case for an ancestrial genetic system involving simple analogues of the nucleotides. *Proc. Natl. Acad. Sci. U.S.A.*, 84:4398-4402.

Kashefi, K. and Lovley, D. R. (2003). Extending the upper temperature limit for life. *Science*, 301:934-934.

Kasting, J. F. and Ackerman, T. P. (1986). Climatic consequences of very high-carbon dioxide levels in the earths early atmosphere. *Science*, 234:1383-1385.

Kawamura, K. (2000). Monitoring hydrothermal reactions on the millisecond time scale using a micro-tube flow reactor and kinetics of ATP hydrolysis for the RNA world hypothesis. *Bull. Chem. Soc. Jpn.*, 73:1805-1811.

Kawamura, K. (2002). *In situ* UV-VIS detection of hydrothermal reactions using fused-silica capillary tubing within 0.08 - 3.2 s at high temperatures. *Anal. Sci.*, 18:715-716.

Kawamura, K. (2003a). Kinetics and activation parameter analyses of hydrolysis and interconversion of 2',5'- and 3',5'-linked dinucleoside monophosphate at extremely high temperatures. *Biochim. Biophys. Acta*, 1620:199-210.

Kawamura, K. (2003b). Kinetic analysis of cleavage of ribose phosphodiester bond within guanine and cytosine rich oligonucleotides and dinucleotides at 65 - 200 °C and its implications on the chemical evolution of RNA. *Bull. Chem. Soc. Jpn.*, 76:153-162.

Kawamura, K. (2004). Behavior of RNA under hydrothermal conditions and the origins of life. *Inter. J. Astrobiol.*, 3:301-309.

Kawamura, K. and Ferris, J. P. (1994). Kinetics and mechanistic analysis of dinucleotide and oligonucleotide formation from the 5'-phosphorimidazolide of adenosine on Na^+-montmorillonite. *J. Am. Chem. Soc.*, 116:7564-7572.

Kawamura, K. and Maeda, J. (2007). Kinetic analysis of oligo(C) formation from the 5'-monophosphorimidazolide of cytidine with Pb(II) ion catalyst at 10 – 75 °C. *Origins Life Evol. Biospheres*, 37:153-165.

Kawamura, K. and Maeda, J. (2008). Kinetics and activation parameter analysis for the prebiotic oligocytidylate formation on Na^+-montmorillonite at 0 – 100 °C. *J. Phys. Chem. A*, 112:8015-8023.

Kawamura, K. and Nagayoshi, H. (2007). Behavior of DNA under hydrothermal conditions with $MgCl_2$ additive using an in situ UV-visible spectrophotometer. *Thermochimica Act*a, 466:63-68.

Kawamura, K. and Shimahashi, M. (2008). One-step formation of oligopeptide-like molecules from Glu and Asp in hydrothermal environments. *Naturwissenschaften*, 95:449-454.

Kawamura, K. and Umehara, M. (2001). Kinetic analysis of the temperature dependence of the template-directed formation of oligoguanylate from the 5'-phosphorimidazolide of guanosine on a poly(C) template with Zn^{2+}. *Bull. Chem. Soc. Jpn*., 74:927-935.

Kawamura, K. and Yukioka, M. (2001). Kinetics of the racemization of amino acids at 225 - 275 °C using a real-time monitoring method of hydrothermal reactions. *Thermochimica Acta*, 375:9-16.

Kawamura, K., Nakahara, N., Okamoto, F., and Okuda, N. (2003). Temperature dependence of the cyclization of guanine and cytosine mix hexanucleotides with water-soluble carbodiimide at 0 -75 °C. *Viva Origino*, 31:221-232.

Kawamura, K., Kuranoue, K., and Nagahama, M. (2004). Prebiotic inhibitory activity of protein-like molecules to the template-directed formation of oligoguanylate from guanosine 5'-monophosphate 2-methylimidazolide on a polycytidylic acid template. *Bull. Chem. Soc. Jpn.*, 77:1367-1375.

Kawamura, K. Nishi, T., and Sakiyama, T. (2005). Consecutive elongation of alanine oligopeptides at the second time range under hydrothermal condition using a micro flow reactor system. *J. Am. Chem. Soc.*, 127:522-523.

Kawamura, K., Takeya, H., and Kushibe, T. (2009a). Effect of condensation agents and minerals for oligopeptide formation under mild and hydrothermal conditions in related to chemical evolution of proteins. *Adv. Space Res.*, 44:267–275.

Kawamura, K., Nagayoshi, H., and Yao, T. (2009b). Stability of ribonuclease A under hydrothermal conditions in relation to the origin-of-life hypothesis: Verification with the hydrothermal micro-flow reactor system. *Res. Chem. Intermed*., 35:879–891.

Kawamura, K., Nagayoshi, H., and Yao, T. (2010). In situ analysis of proteins at high temperatures mediated by capillary-flow hydrothermal UV-Vis spectrophotometer with a water-soluble chromogenic reagent. *Anal. Chim. Acta*, 667:88–95.

Lohrmann, R. and Orgel, L. E. (1973). Prebiotic activation process. *Nature*, 244:418-420.

Lohrmann, R. and Orgel, L.E. (1980). Efficient catalysis of polycytidylic acid-directed oligoguanylate formation by Pb^{2+}. *J. Mol. Biol.*, 142:555-567.

Maher, K. A. and Stevenson, D. J. (1988). Impact frustration of the origin of life. *Nature*, 331:612-614.

Marshall, W. L. (1994). Hydrothermal synthesis of amino acids. *Geochim. Cosmochim. Acta,* 58:2099-2106.

Miller, S. L. and Bada, J. L. (1988). Submarine hot springs and the origin of life. *Nature*, 334:609-611.

Mojzsis, S. J., Arrhenius, G., McKeegan, K. D., Harrison, T. M., Nutman, A. P., and Friend, C. R. L. (1996). Evidence for life on Earth before 3,800 million years ago. *Nature*, 384:55-59.

Pace, N. R. (1991). Origin of life - Facing up to the physical setting. *Cell*, 65:531-533.

Radzicka, A. and Wolfenden, R. (1995). A proficient enzyme. *Science*, 267:90-93.

Sagan, C. and Mullen, G. (1972). Earth and Mars: evolution of atmospheres and surface temperatures. *Science*, 177:52-56.

Sawai, H., Kuroda, K. and Hojo, H. (1989). Uranyl ion as a highly effective catalyst for internucleotide bond formation. B*ull. Chem. Soc. Jpn.*, 62:2018-2023.

Stetter, K. O. (1982). Ultrathin mycelia-forming organisms from submarine volcanic areas having an optimum growth temperature of 105 °C. *Nature*, 300:258-260.

Terfort, A. and Von Kiedrowski, G. (1992). Self-replication by condensation of 3-aminobenzamidines and 2-formylphenoxyacetic acids. *Angew. Chem. Int. Ed. Eng.*, 31:654-656.

Tuerk, C. and Gold, K. (1990). Systematic evolution of ligands by exponential enrichment: RNA ligands to bacteriophage T4 DNA polymerase. *Science*, 249:505-510.

Walker, J. C. G. (1985). Carbon dioxide on the early earth. *Origins Life Evo. Biosphere*, 16:117-127.

The Origin of Informational Replicators by Serial Dilution of a Primordial Soup

Chrisantha Fernando[1,2,3]

[1]Center for Computational Neuroscience and Robotics, University of Sussex, Falmer, Brighton, BN1 9RH
[2]Collegium Budapest, 1014 Budapest, Szentháromság utca 2, Hungary
[3]The National Institute for Medical Research, The Ridgeway, Mill Hill, London, NW7 1AA, UK

Abstract

How can informational replicators (Zachar and Szathmáry 2010) such as template replicators, arise from non-informational autocatalysts (Szathmáry and Maynard Smith 1997; Szathmary 2000)? Variants of an informational replicator have a high probability of being autocatalytic, thus allowing potentially unlimited heritable variants to be replicated, for example, mutants of a DNA sequence have this property. Variants of non-informational replicators such as glycolaldehyde in the Formose cycle are *not* in general autocatalytic; therefore, there is little capacity for hereditary variation (Szathmáry 2006). This paper asks; what are the necessary and sufficient conditions for an increase in the probability that a variant of an autocatalyst will itself be capable of autocatalysis? Given some well-defined assumptions, serial dilution in a rich generative chemistry such as that found in the Miller experiment should result in the emergence of informational replicators, i.e. autocatalysts whose variants have a high probability of themselves being capable of autocatalysis.

Introduction

A reactor such as that of Millar's famous experiment (Miller 1953) contains reactions that are simple autocatalytic cycles (and probably more complex kinds of autocatalytic structure, e.g. reflexive autocatalytic sets (Farmer, Kauffman et al. 1986; Kauffman 1986)). An example of a simple autocatalytic cycle is the Formose reaction (Fernando, Santos et al. 2005), see Figure 1. It is known that this autocatalytic cycle is notoriously subject to side-reactions, the reaction of molecules external to the cycle with the intermediates of the cycle to produce new molecules. Some of these new molecules will themselves be autocatalytic with some probability p that we assume is a property of the parental autocatalytic cycle. The same fate of side-reactions befalls these newly produced autocatalysts.

Real chemistry is very complicated, but it is possible to get some idea of the dynamics of a growing chemical network of reactions by using simplified artificial chemistries. A typical abstraction is to use linear binary strings as molecules and allow ligation and cleavage reactions between these strings. This paper will use an even simpler artificial chemistry where a chemical is described by only two parameters. What is the motivation for this? In simulations carried out previously using a artificial chemistry (Fernando and Rowe 2007; Fernando and Rowe 2008) it was observed that the probability of an autocatalytic molecule producing another autocatalytic molecule in a side-reaction decreased with the size of the molecule. This is an inevitable consequence in a random chemistry of linear strings because longer strings are less likely to produce two copies of one reactant by chance, than are shorter strings, given random rearrangement of the monomers of in a bimolecular rearrangement reaction (the type used in the simulation). The reality for real organic molecules is of course much more complicated. Some classes of autocatalytic molecule will inevitably be more likely to produce autocatalysts than others (i.e. have different p values). The complexity of the chemical models that would be needed to determine these probabilities for various classes of molecule are bewildering and possibly beyond that which is currently feasible. Therefore, a model is presented that abstracts certain properties of this generative chemical process. The model assumes simply that an autocatalyst can be described by a small number of parameters. Firstly, a probability p that a side-reaction to the autocatalytic cycle produces an autocatalyst. Secondly, a structural parameter $\bar{p}$ that describes the mean of a lognormal distributed set of values from which is drawn the probability p' that an autocatalyst produced by a side-reaction will be capable of itself producing autocatalysts, see Figure 1. Thirdly, for some variants of the model it is assumed that each autocatalyst has some observable property f. f is drawn randomly for each autocatalyst from a normal distribution with mean 0 and s.d. = 1. In the models, there is no correlation between f of a parent and f of an offspring molecule. This f is intended to be some function that may contribute to fitness at a higher level.

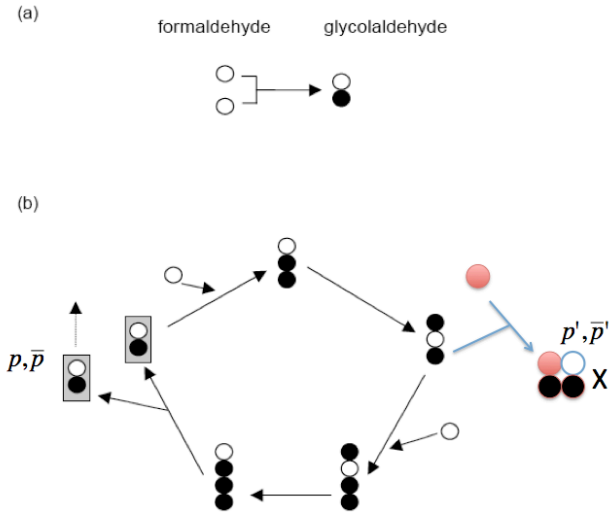


Figure 1 (Top) The molecule glycolaldehyde is autocatalytic using Formaldehyde as food, and making copies of itself, and growing in concentration exponentially. **(Bottom)** The intermediates of the glycolaldehyde autocatalytic cycle can undergo side-reactions with other species (red) to produce autocatalysts with a low probability *p*. Let these new autocatalysts have a probability *p'* of producing autocatalysts themselves by side-reactions. In the model, a structure parameter of the parental autocatalyst $\overline{p}$ determines stochastically the actual value of *p'* that an offspring autocatalyst will have.

Methods

A reactor is initialized with one core autocatalyst that has *p* drawn from a lognormal distribution with mean $\overline{p} = e^{-10}$. This is a small value, e.g. 0.001. The production of novel autocatalysts is simulated using a discrete time simulation. At each time-step, each existing autocatalyst has a probability *p* of producing another autocatalyst. If it does produce an autocatalyst, then this new autocatalyst has its *p'* value assigned by choosing a random number from the lognormal distribution defined by the $\overline{p}$ value of the "parent" autocatalyst. The new autocatalyst then has its $\overline{p}$' value defined based on the original $\overline{p}$ value of its parent. The crucial question in any realistic chemistry is whether there is a correlation between the $\overline{p}$ value of a parental autocatalyst and the $\overline{p}$' value of the autocatalyst produced from it. In other words, is the probability of producing an autocatalyst in a side-reaction a heritable parameter; is $\overline{p}$ heretable? It is clearly the case that there is no such simple correlation for all classes of molecule, although for some molecules there clearly is, for example, polymer template replicators. Such molecules have a very high probability that a variant will also be capable of replication. Several functions that relate the heretability $\overline{p}$ of parent and heritability $\overline{p}$' of offspring are examined in this paper. The simplest function assumes correlated $\overline{p}$ values where the $\overline{p}' = Norm(1,\sigma)\overline{p}_p$, where *Norm* is a Gaussian random number with mean 1 and standard deviation σ. An uncorrelated function is one in which $\overline{p}_c = e^{rand(-10,-9.5)}$, where rand(-10,-9.5) is a uniform random number between -10 and -9.5, the typical values evolved in the previous experiments when $\overline{p}$ was an evolvable parameter.

The reactor produces autocatalysts for a fixed time period T after which *M* random samples (containing autocatalysts) are taken from the reactor. Each autocatalyst has some probability *q* of being chosen for each sample, and let this value be fixed throughout a simulation. Let the chance of choosing an autocatalyst be low, e.g. 5%. In reality this probability *q* will depend on abundance, but here we have no model of chemical kinetics. Also, we do not allow the number of autocatalysts chosen to exceed some maximum *C* e.g. *50.* The sample will also inevitably contain many non-autocatalytic molecular species that are not modeled here.

One of the *M* samples are chosen based on maximizing the linear sum of *f* values of the autocatalyst species present in the reactor. Another valid option is just to choose a random sample. Both options are modeled here. The chosen sample then is used to reinitialize a new reactor. All autocatalysts not present in this sample are discarded. This is the serial dilution phase of the experiment.

Results

Correlated $\overline{p}$

Figure 2 shows the results obtained for a run in which selection is for highest summed *f*. The initial value of $\overline{p} = e^{-10}$, *q = 0.05, C = 50, M = 10.* In the function $\overline{p}' = Norm(1,\sigma)\overline{p}_p$, σ *= 0.001,* i.e. there are small correlated changes to the potential to produce autocatalysts (P1 in the diagram).

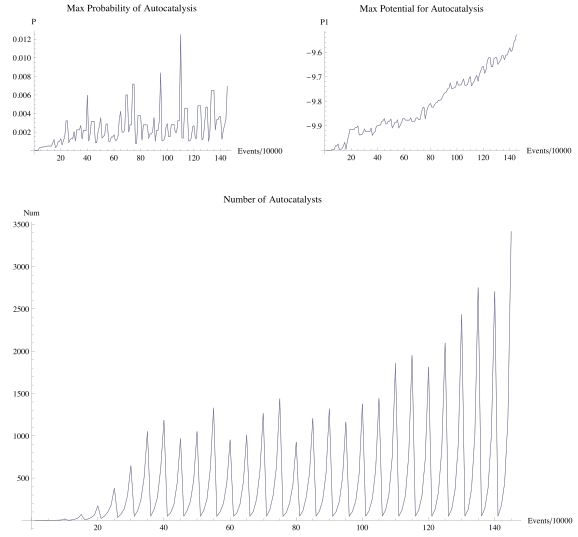


Figure 2. 28 serial dilutions, with selection for highest *f* sample. **(Top Left)** Maximum *p* value obtained. **(Top Right)** Maximum $\overline{p}$ value obtained. **(Bottom)** Total number of autocatalysts in the reactor.

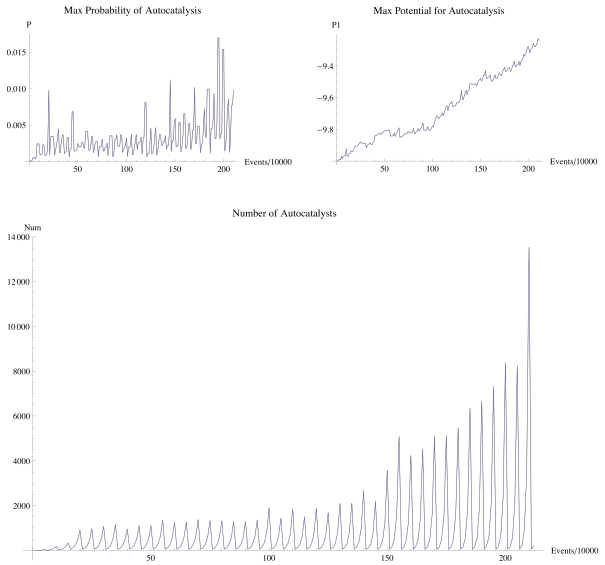


Figure 3. 34 serial dilutions, with random selection of a compartment. **(Top Left)** Maximum p value obtained. **(Top Right)** Maximum $\bar{p}$ value obtained. **(Bottom)** Number of autocatalysts.

After 28 serial dilutions of the system, the maximum value of $\bar{p}$ has increased significantly, and more autocatalysts are being produced in each round of network growth. Random compartment selection has a similar effect, see Figure 3. Selection for the compartment with the largest number of autocatalysts also has a similar effect (not shown). Next we consider the effect of making $\bar{p}$ a non-heritable structural parameter.

Uncorrelated $\bar{p}$

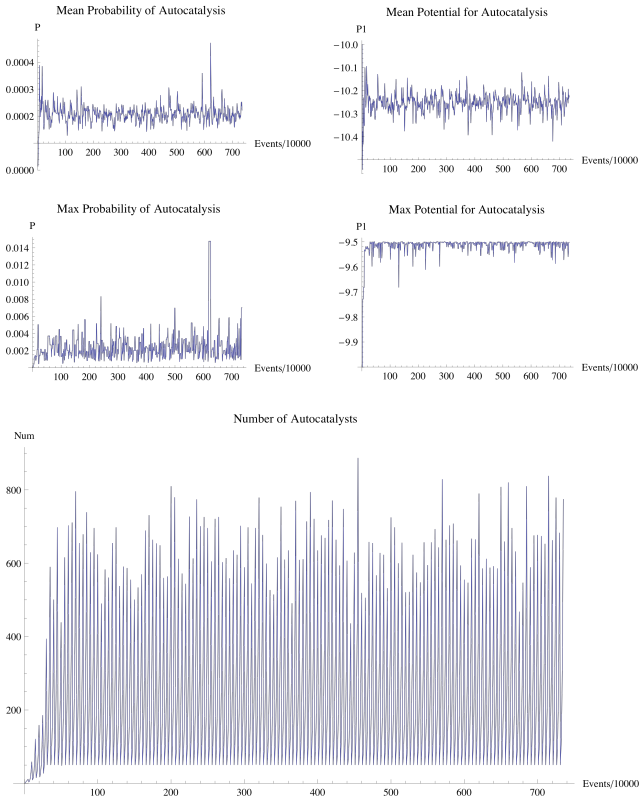


Figure 4. With uncorrelated $\bar{p}$**,** there is no improvement in autocatalysts over many serial dilutions. Selection is for the compartment with the largest number of autocatalysts.

Figure 4 shows the behaviour with an entirely uncorrelated potential for autocatalysis between successive autocatalysts $\bar{p}_c = e^{rand(-10,-9.5)}$. The probability p no longer tends to higher values because whilst a parental catalyst may occasionally produce an offspring with high p, this offspring has no tendency to itself produce offspring with high p. There exist autocatalysts in the population that do have high values of $\bar{p}$**,** but the mean value of $\bar{p}$ does not increase, as can be seen in the plot of mean $\bar{p}$ in Figure 4.
These results suggest that if the structural variability parameter $\bar{p}$ is not capable of being inherited, then there will be no tendency for the population of autocatalysts to tend towards becoming informational replicators.

Conclusions

The simple but fundamental principle demonstrated above is an example of the evolution of evolvability (Conrad 1990; Clune, Misevic et al. 2008; Parter, Kashtan et al. 2008), namely, that natural selection can act to select variants that are not of immediate benefit to the individual replicator, but confer improved variability properties, i.e. increase the chance that offspring will be fit. If there is variation (within generation differences) in variability (the capacity to produce variants during propogation) then there can be selection for variability properties that are beneficial to the lineage. This has been called lineage selection (Aboitiz 1991), and second order selection (Tenaillon, Taddei et al. 2001). Mark Toussaint has formalized the process of structuring phenotypic exploration distributions (Toussaint 2003) due to non-trivial neutrality, i.e. the capacity for the same phenotype p to be due to different genotypes $\bar{p}$**.** If some genotypes $\bar{p}$ tend to produce better variations in the phenotype p then those genotypes can be selected for. In this model it is shown that the capacity for non-trivial heritable neutral variation of $\bar{p}$ can allow increasing p.

The question remains, in chemistry, is there ever a circumstance in which $\bar{p}$ could be heritable within a lineage of autocatalysts? A conservative answer is sometimes yes, sometimes no. However, in this situation, the network dynamics would exhibit a tendency to select for that class of autocatalyst that did exhibit heredity of $\bar{p}$**.**

It is therefore proposed that experimentally it would be a matter of acute interest to take a rich generative chemistry such as that of Miller capable of producing a combinatorial explosion of polymers, and to take samples from the reactor once it had had a chance to generate this molecular diversity. These samples (selecting for the sample with the highest number of autocatalysts if possible) would be used to inoculate a new reactor. This cycle would be repeated for as many generations as possible. Each epoch should permit the generation of a new set of autocatalysts. This simple model predicts that such a protocol should be capable of generating informational replicators.

There are several simplifying assumptions of this model that must be examined. First we have ignored the fact that mass is finite. This means that exploration of the autocatalytic network may become limited if the mass of the reactor is used up producing non-autocatalytic molecules. Secondly we have completely ignored the existence of cross-catalytic interactions which may produce reflexive autocatalytic structures that can act as informational units. However, reflexive structures are only an intermediate step in what must eventually be selection for heretable $\bar{p}$ in the origin of

microevolution from macroevolution. An interesting addition to the model would be to allow species to be both autocatalytic and cross-catalytic with some probability. The interactions of the reactor would be described by a replication matrix. Adding a new species would involve producing a new row and column in this matrix. In addition to this matrix, each species would be described by structural parameters that determined the entries in the new row and column of the replication matrix for species that were produced in side reactions with it. Thirdly, the form of the structural parameter $\bar{p}$ (acting as a mean of a lognormal distribution to produce p') is somewhat arbitrary. A much more realistic method of describing the structural tendency for autocatalysis would be desirable.

Recent work by Ben Davis's group in Oxford has succeeded in enclosing a Formose cycle metabolism within lipid compartments. They are able to select for those compartments with certain chemical compositions (Gardner, Winzer et al. 2009). This paper is of some significance to them. If they were to simply choose small samples of each compartment and continue to test each sample for distinct autocatalytics, we predict that over many generations, one should find a greater diversity of independent autocatalysts.

Acknowledgements. Thanks to a Marie Curie Fellowship to work at Collegium Budapest, and the FP7 E-FLUX grant for evolutionary microfluidics.

Aboitiz, F. (1991). "Lineage selection and the capacity to evolve." Medical Hypotheses **36**: 155-156.

Clune, J., D. Misevic, et al. (2008). "Natural Selection Fails to Optimize Mutation Rates for Long-Term Adaptation on Rugged Fitness Landscapes." PLoS Computational Biology **4**(9): e1000187.

Conrad, M. (1990). "The geometry of evolution." BioSystems **24**(61-81).

Farmer, J. D., S. A. Kauffman, et al. (1986). "Autocatalytic Replication of Polymers." Physica D **22**: 50-67.

Fernando, C. and J. Rowe (2007). "Natural Selection in Chemical Evolution. ." Journal of Theoretical Biology **247**: 152-167

Fernando, C. and J. Rowe (2008). "The origin of autonomous agents by natural selection. ." Biosystems **91**: 355-373.

Fernando, C., M. Santos, et al. (2005). "Evolutionary potential and requirements for minimal protocells. ." Top. Curr. Chem. **259**: 167-211.

Gardner, P. M., K. Winzer, et al. (2009). "Sugar synthesis in a protocellular model leads to a cell signalling response in bacteria." Nature Chemistry **1**: 377-383.

Kauffman, S. A. (1986). "Autocatalytic Sets of Proteins." Journal of Theoretical Biology **119**: 1-24.

Miller, S. L. (1953). "Production of Amino Acids Under Possible Primitive Earth Conditions." Science **117**: 3046.

Parter, M., N. Kashtan, et al. (2008). "Facilitated Variation: How Evolution Learns from Past Environments to Generalize to New Environments." PLoS Computational Biology **4**(11): e1000206.

Szathmary, E. (2000). "The evolution of replicators. ." Phil. Trans. Roy. Soc. Lond. B **355**: 1669–1676.

Szathmáry, E. (2006). "The origin of replicators and reproducers." Philos. Trans. R. Soc. London. B. Biol. Sci. **361**(1474): 1761-1776.

Szathmáry, E. and J. Maynard Smith (1997). "From replicators to reproducers: the first major transitions leading to life." Journal of Theoretical Biology **187**(4): 555-571.

Tenaillon, O., F. Taddei, et al. (2001). "Second-order selection in bacterial evolution: selection acting on mutation and recombination rates in the course of adaptation." Res. Microbiol **152**: 11-16.

Toussaint, M. (2003). The evolution of genetic representations and modular adaptation. ND 04, 44780 Bochum---Germany.

Zachar, I. and E. Szathmáry (2010). "A New Replicator: A theoretical framework for analysing replication." BMC Biology **8**: 21.

Evolution, Selection and the Metabolic Foundations of the RNA World

Andrew J. Pratt[1]

[1]Department of Chemistry, University of Canterbury, Christchurch, PB4800, NZ
andy.pratt@canterbury.ac.nz

Abstract

The chemoton model of cells posits three sub-systems: metabolism, compartmentalization and information (Gánti, 2003). This paper describes a specific model for the evolution of a reproducing system with rudimentary versions of these three inter-dependent sub-systems. This is based on the initial emergence and reproduction of autocatalytic networks in hydrothermal micro-compartments containing iron sulfide. The driving force for life is catalysis of the dissipation of the intrinsic redox gradient of the planet (Russell and Kanik, 2010). The initial proto-metabolism was based on positive feedback loops associated with *in situ* carbon fixation in which the initial proto-metabolites modified the catalytic capacity and mobility of metal-based catalysts, especially iron-sulfur centres. A number of selection mechanisms, including catalytic efficiency and specificity, hydrolytic stability and selective solubilization, are proposed as key determinants for autocatalytic reproduction exploited in proto-metabolic evolution. This evolutionary process leads from autocatalytic networks within pre-existing compartments to discrete, reproducing, mobile vesicular protocells with the capacity to use soluble sugar phosphates and hence the opportunity to develop nucleic acids. Fidelity of information transfer in the reproduction of these increasingly complex autocatalytic networks is a key selection pressure in prebiological evolution that eventually leads to the selection of nucleic acids as a digital information sub-system and hence the emergence of fully functional chemotons capable of Darwinian evolution.

Introduction

Chemoton sub-systems and evolutionary pathways

Living cells are autocatalytic entities that harness redox energy via the selective catalysis of biochemical transformations. The complexity of cells requires that they emerged from evolutionary processes that predate life: a form of prebiological evolution (Szathmáry, 2007). The simplest model for cells is the chemoton model which regards them as fluid automata (Gánti, 2003). Chemoton theory proposes that living cells are comprised of three essential interconnected sub-systems associated with metabolism, compartmentalization and information. A metabolic sub-system is required to provide the building blocks and chemical energy for life. Compartmentalization is required for evolution to act on discrete competing entities. Finally, an information sub-system allows the evolution of levels of complexity that are a distinctive feature of life.

A theory of the origin of life based on the chemoton, or related, model must explain a clear pathway to the co-existence of these three interdependent sub-systems. (Szathmáry, 2007). Simultaneous creation of an entity with all three sub-systems in place is exceedingly improbable (Dyson, 1999); it is more likely that cells arose via a pathway involving accretion of one or two sub-system(s) by a simpler system. There are competing perspectives based on the assumed timing of events. What comes first: compartments, information and/or metabolism? The two main competing hypotheses both assume compartmentalization as an early feature, either via the self-assembly of lipids (Deamer, et al., 2006), or via surface adsorption (Wächtershäuser, 1988). They differ in the initially associated sub-system: information-first or metabolism first.

The closest synthetic models we have of partial chemotons are protocells based on lipid-encapsulated RNA molecules (Hanczyc, et al. 2003; Luisi, et al. 2006). These build on the demonstration of directed evolution in *in vitro* RNA systems (Kacian, et al. 1972) and the success of the RNA world hypothesis in exploring the dual ability of RNA molecules to act as both catalysts and stores of hereditary information (Gesteland, et al. 2006). However, an RNA world depends on the continued availability of complex raw materials, including sources of chemically activated nucleotides for polymerization, and of turnover of these materials in reproduction to allow selection of functional macromolecular structures. A significant challenge for this model is to understand the energy flux that created and sustained an RNA world; in particular the underpinning functional metabolism that harnessed redox energy for the evolution of the system and which provided the basis for contemporary biochemistry. In this model it is often assumed that metabolism emerges to replace spent pre-existing metabolites. This model for the engineering of metabolic pathways backwards to alternate starting materials is originally due to Horowitz (1945) but is out of step with recent insights into the evolution of biochemical metabolism (Zhang, et al., 2009) and unlikely to be the complete story.

The competing viewpoint is that the first steps to life were based on compartmentalized proto-metabolism that subsequently developed an information sub-system. Wächtershäuser, Russell, de Duve, Morowitz and others have developed models of this type in which proto-metabolic reactions are catalyzed and organized on iron sulfide surfaces (Wächtershäuser, 1988; Russell and Hall, 1997; de Duve, 1991; Trefil, et al. 2009).

A major challenge for models which base life on reproducing networks of catalysts, such as those envisaged in the GARD model (Shenhav, et al. 2007), is the limited evolvability of such systems (Vasas, et al. 2010). This paper presents a model that links metabolism-first and RNA models. It is proposed that self-organizing autocatalytic cycles did indeed provide the initial metabolic foundations that underpin a modified version of an RNA world, but that the latter emerged in response to the demands for fidelity of information reproduction. A prebiological (non-Darwinian) evolutionary account is presented that provides a series of specific chemical and physical selection mechanisms for the early stage development of a three sub-system RNA world chemoton.

Proto-metabolism in Pre-existing Compartments

Why and how did life emerge? Life depends on a continuous input of energy that can fuel redox chemistry. This theory for the origin of core metabolism, as a foundation for life, follows the hypothesis of Russell and Kanik (2010) in proposing that life emerged to exploit the intrinsic redox gradient of the earth that has existed since its origin. When the earth formed, an electron-rich core was physically segregated from a weakly oxidizing atmosphere containing carbon dioxide, nitrogen and other electron acceptors. By this model, life emerged in pores (Russell and Hall, 2006) within hydrothermal mineral deposits where there is a mixing of these otherwise segregated zones of the planet.

It is proposed that the critical features of this environment for the emergence of life are: (i) a continuous input of redox energy; (ii) a kinetic barrier to the dissipation of the intrinsic redox gradient; (iii) the availability of catalysts in a mixing zone that can speed dissipation of the gradient, but where initial catalysts are inefficient and capable of increased efficiency by diversification to networks of more specific catalysts; and (iv) protection against significant external shocks (*e.g.* protection against irradiation, variations in pH, ionic strength etc) to facilitate protocell evolution by allowing the reproduction of catalytic networks as discrete entities. This environment provides an evolutionary opportunity for the emergence of networks of catalysts of increasing complexity and is necessary, but not sufficient, for life. There is a limit to the complexity of simple catalytic cycles associated with limits to fidelity of reproduction (Vasas, et al. 2010). It is proposed that life, as we know it, emerges if and when a digital information sub-system evolves that transcends the information limits of simple chemical networks and allows open-ended Darwinian evolution with natural selection.

Iron sulfur species and the early evolution of catalytic centres. Following the patchwork model of evolution of biochemical catalysts (Jensen, 1976), the best starting point for evolution is the availability of generic, but inefficient catalysts that are capable of evolving increased specificity and efficiency (Szathmáry, 2007). One key issue for self-organising autocatalytic networks, highlighted by Orgel (2000), is the need for a series of catalysts that mediate all the processes of the network. Iron-sulfur based species (Beinert, et al., 1997) are well placed to fill this role since they are capable of catalyzing a diverse range of both redox and acid-base chemistry. Much of this chemistry is utilized in contemporary core metabolism via iron-sulfur clusters that resemble iron sulfide mineral structures (Figure 1) (Rickard and Luther III, 2007). Iron-sulfur clusters occur naturally in aqueous systems (Rozan, et al. 2000). Biochemical clusters of this kind mediate the following processes: (i) bioenergetic electron-transfer processes (*e.g.* Xia, et al. 1997; Cheng, et al. 2006) (ii) other metabolic redox chemistry, *e.g.* carbon fixation (Ragsdale, 1991), nitrogen fixation (Einsle, et al. 2002), reversible hydrogen formation (Nicolet, et al. 2000) and organic radical chemistry (Berkovich, et al., 2004; Nicolet and Drennan, 2004); and (iii) a diverse range of acid-base chemistry, including hydration-dehydration chemistry, *e.g.* aconitase, serine dehydratase and related enzymes (Flint and Allen, 1996).

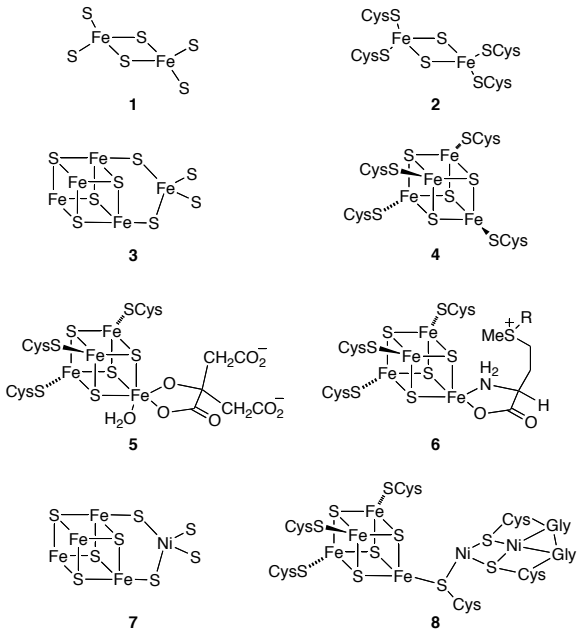


Figure 1: Iron sulfur minerals and catalytic biochemical clusters. **1** mackinawite sub-structure **2**: [2Fe,2S] electron-transfer cluster; **3**: greigite sub-structure; **4**: [4Fe,4S] electron-transfer cluster; **5**: acid-base catalyst (aconitase with citrate bound); **6**: radical generating cluster (with *S*-adenosylmethionine bound, R = adenosyl); **7**: model for Ni-substituted greigite; **8**: carbon fixing cluster of ACS.

The specific catalytic properties of iron-sulfur dependent enzymes is controlled by the composition of the metal-sulfur cluster and the details of the coordinating ligands (Figure 1). For example, iron-sulfur clusters completely coordinated by sulfur ligands (**2** and **4**) act as specific electron-transfer proteins in which the redox potential is moderated by cluster size and details (Rao and Holm, 2004). Clusters, such as the [4Fe,4S] cluster in aconitase (**5**), with one non-sulfur coordination site can undergo active metal and ligand exchange chemistry. Ligands, such as carboxylates, transiently bound to such clusters can undergo reactions involving acid-base catalysis (Flint and Allen, 1996). When

bound to an iron-sulfur cluster the amino acid derivative *S*-adenosylmethionine is a source of organic radicals (**6**). Iron sulfide minerals contain other metal ions (**7**) (Russell and Hall, 2006). The presence of adjacent metals ions, *e.g.* nickel, cobalt and molybdenum, provides new distinctive catalytic chemistry that can exploit the electron-transfer chemistry of iron sulfides. For example, nickel, iron sulfur clusters are utilized in a number of enzymes, including both key enzymes of the Wood-Ljungdahl carbon fixation pathway, CO dehydrogenase and acetyl-CoA synthase (**8**) (Volbeda, A. and Fontecilla-Camps, 2005); likewise, molybdenum, iron sulfur clusters are utilized in nitrogenase (Einsle, et al. 2002).

The ability to modify and control specific catalytic activities via coordination chemistry provides the potential for the evolution of catalysts of diversified specificity and activity in an emerging division of (proto-metabolic) labour.

Prebiotic Wood-Ljungdahl carbon fixation: the first step.

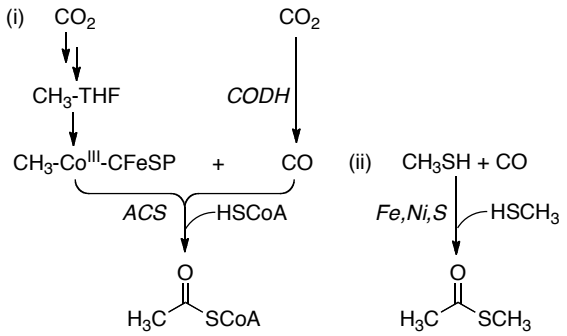


Figure 2: Overview of (i) Wood-Ljungdahl carbon fixation pathway and (ii) biomimetic geochemical analogue.

The shortest and simplest known route to biological carbon fixation is the Wood-Ljungdahl pathway (Figure 2) in which carbon dioxide is reduced to carbon monoxide at an iron, nickel sulfur centre of CO dehydrogenase (CODH). The carbon monoxide is then transferred directly to acetyl CoA synthase (ACS), another iron, nickel and sulfur-dependent enzyme, where it carbonylates a methyl-nickel species. The resulting acetyl nickel intermediate is intercepted by the thiol coenzyme A to produce acetyl CoA (Grahame, 2003; Hegg, 2004; Russell and Martin, 2004). The methyl group is delivered to this system by a cobalt corrinoid iron sulfur protein (CFeSP) (Svetlitchnaia, et al., 2006). In this carbon fixation pathway the key manipulations of carbon species are mediated by nickel and cobalt centres with adjacent iron-sulfur clusters supplying electrons. In geochemical systems the initially deposited iron monosulfide is nanoparticulate mackinawite, which adsorbs divalent metal ions (Wolthers, et al., 2003) such as nickel and cobalt. Huber and Wächtershäuser (1997) have shown that inorganic iron, nickel sulfide catalyses a simple analogue of acetyl CoA synthase chemistry in water, converting methanethiol to methyl thioacetate (Figure 2). The product thioester is hydrolysed under the reaction conditions to acetate which provides a strong overall thermodynamic driving force (Shock, 1992).

This simple geochemistry immediately provides a positive feedback mechanism that can underpin the generation of more complex catalytic networks. Carbon fixation involves the reductive formation of organic compounds and the concomitant oxidation of the iron sulfide. Mackinawite is a two dimensional semi-conductor with a layered structure (Rickard and Luther III, 2007). Surface oxidation processes, *e.g.* at a catalytically active nickel centre, will draw electrons from the iron sulfide. Oxidation of mackinawite produces greigite and other pyrrhotite iron sulfide minerals (Lennie, et al. 1997). Mackinawite oxidation is inefficient in the absence of suitable additives and it is known that redox-active organic compounds can facilitate such transformations (Rickard, et al., 2001).

$$\underset{\text{(mackinawite)}}{4FeS} \rightarrow \underset{\text{(greigite)}}{Fe_3S_4} + Fe(II) + 2e^- \qquad \text{(Equation 1)}$$

Mackinawite and greigite are both based on a close-packed sulfide lattice (Rickard and Luther III, 2007). In mackinawite the iron is in a tetragonal environment. In the transition to greigite some of the iron centres become octahedral. It is expected that this change will diversify the chemistry and catalytic properties of the iron sulfide local to the site of oxidation. In support of this view, Mike Russell has pointed out that mackinawite bears some resemblance to [2Fe,2S] clusters found in some simple electron transfer proteins, whereas greigite contains a sub-unit analogous to the [4Fe,4S] clusters found in many iron, sulfur dependent enzymes, including the key Wood Ljungdahl enzymes (Figure 1) (Russell and Hall, 2006).

Furthermore, interconversion of the two minerals involves a relocation of iron ions (Equation 1); these will presumably migrate to the surface. Organic compounds produced by the carbon fixation chemistry that are ligands will bind to the surface metal ions, including the newly exposed iron centres, modifying their chemistry. The generation of new catalytic centres which increase the overall activity with respect to carbon fixation will act as a positive feedback loop where the flux of oxidized carbon and reducing power, *e.g.* geochemically generated hydrogen, will be differentially turned over by catalytically active microporous domains within the hydrothermal rocks that contain both ligands and diverse catalytic metal centres.

Subsequent known iron-sulfur mediated transformations, can produce a suite of core proto-metabolites - ligands that can bind to and modify the catalytic chemistry of iron sulfur centres (Figure 3). Reductive carboxylation of thioesters from carbon fixation can produce α-keto acids, *e.g.* pyruvate (Cody, et al. 2000). These chelating ligands can undergo further chemistry once bound. Reductive amination of bound α-keto acids, using ammonia from the reductive fixation of nitrogen (Dörr, et al., 2003) and/or nitrate (Blochl, et al. 1992), can then give rise to α-amino acids via reductive amination (Huber and Wächtershäuser, 2003). Utilization of related substrates will produce a core of simple proto-metabolites which are selected on the basis of their being ligands for iron that modify the catalytic chemistry of exposed iron sites and hence the catalytic turnover of the emerging family of proto-metabolites. A family of diversified catalytic centres, with complementary activity, provides the basis for networks that are more productive than individual catalysts. In a porous hydrothermal mound a diverse variety of potential microenvironments will be evaluated as potential sources of autocatalytic networks. Individual pores with distinctive

mineral chemistry can develop distinctive chemical variants in an early form of compartmentalized proto-metabolism.

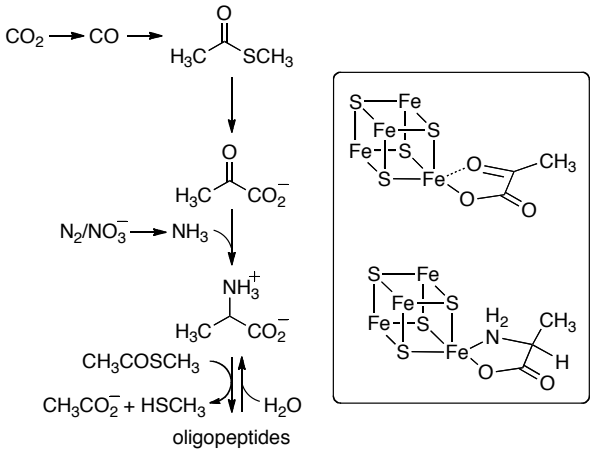


Figure 3: Generation of core proto-metabolites within an iron sulfide system. Binding of representative proto-metabolites to iron-sulfur centres is illustrated in the box.

The first oligomers and molecular evolution. Complex macromolecules are a key feature of biochemistry. All biological macromolecules are condensation polymers, created by dehydration of monomeric building blocks. In water, condensation polymers are unstable with respect to hydrolysis. These condensation polymers require biochemical energy, usually equated with ATP or related polyphosphates, for their synthesis. ATP is the archetypal water-compatible dehydrating agent (Westheimer, 1987).

A critical feature of the prebiotic Wood-Ljungdahl chemistry is that it generates thioesters as obligate intermediates. Thioesters are the other major class of water-compatible biochemical dehydrating agents and their intermediacy in carbon fixation chemistry provides dehydrating power that makes condensation polymers accessible. Since this chemistry was quickly associated with a growing pool of α-amino acids, oligopeptides were among the early oligomeric compounds (Figure 3). It has been shown that amides can be formed from amino acids in water using the intrinsic dehydrating power of prebiotic Wood-Ljungdahl catalysis (Huber and Wächtershäuser, 1998). Such oligopeptides are also ligands that are able to bind to iron-sulfur and other metal species and thereby modify the catalytic activity of the system by controlling coordination spheres. The production of condensation oligomers provides an explicit molecular selection mechanism. Since condensation oligomers are unstable with respect to hydrolysis in water, such condensation polymers only accumulate if they are generated faster than the rate at which they "die" via hydrolysis. Oligopeptides that facilitate the overall catalytic potential of the system will facilitate the production of further oligopeptides; condensation oligomers that participate in this feedback loop will be selected. Families of related oligopeptide-metal centres will emerge that can harness the chemistry of metal-sulfide clusters found in aqueous systems (Rozan, et al. 2000) and mediate distinct classes of chemical transformation with rudimentary specificity (*e.g.* acid-base chemistry vs redox chemistry). There will be some structural and metal-binding selectivity in these ligands, but they will lack the ordering and hence specificity available from contemporary enzymes.

Mobile Autocatalytic Networks

Solubility and prebiological evolution. The solubility of chemicals associated with catalytically active hydrothermal pores will play a critical role in the chemistry that evolves and in the reproduction of that chemistry. Solid minerals and bound ligands are retained within a finite location of a hydrothermal environment. Such a location has a finite lifetime for active chemistry until the supplies of raw materials are exhausted. A permanently localized autocatalytic network will eventually 'die' from starvation generating a selection pressure for mobility. Chemical products of autocatalytic networks will be leached from the system by solubilization. This is both a purifying mechanism and a seeding or reproduction mechanism. Chemicals, individually, or en masse, that are lost but not replaced are removed from the system as waste. However mobile components that seed neighbouring sites with autocatalytic chemistry are potentially a selectable means of reproduction.

As proposed by Mike Russell (2006), if the emerging autocatalytic networks develop in pores within hydrothermally deposited minerals, these discrete cavities provide an initial rudimentary compartmentalization mechanism. They prevent the free loss of soluble proto-metabolites allowing solution metabolism to emerge. Furthermore, proto-metabolites can accumulate in these pores by a hydrothermal concentration mechanism (Baaske et al., 2007; Budin, et al., 2009).

Iron encapsulation, phosphates and homeostasis. A significant challenge for the development of complex soluble chemistry within a specific pore of a hydrothermal deposit is the presence of high levels of free multivalent metal ions, including iron. Highly charged cations encourage precipitation of counter anions, notably phosphates. This facilitates localization of chemicals and surface catalysis but compromises the development of soluble metabolism, especially one that incorporates phosphate species (Pratt, 2006). It presents a fundamental challenge to the development of an RNA world within a hydrothermal environment.

Cells avoid this precipitation problem via a combination of encapsulation and exclusion of multi-valent metal ions. For example, essentially all iron within living cells is encapsulated within proteins. Calcium ions cannot be readily encapsulated because of their dynamic coordination chemistry and so they are actively pumped out of cells whereupon they form extracellular precipitates, *e.g.* calcium carbonate exoskeletons and bone. These extracellular deposits provide a homeostatic backdrop to the chemistry of cells (*e.g.* bone acts as a reservoir of calcium and phosphate) (Fraústo da Silva and Williams, 2001).

In biochemistry, iron is commonly encapsulated within oligopeptides either as iron mineral clusters or as porphyrin complexes. Both oligopeptides and porphyrins (Eschenmoser, 1988) are oligomers derivable from amino acid building blocks which are, in principle, accessible from plausible

prebiotic catalysis within the hydrothermal autocatalytic system. Templated synthesis (Costisor and Linert, 2004) of these oligomers on iron centres will provide selective routes to both classes of ligand which can sequester free iron ions within the system by competitive coordination chemistry. Oligomeric ligands will tailor the catalytic chemistry of iron sulfur catalytic centres by controlling the nature of the ligand coordination sphere. They will also control free metal ion levels and thereby allow partial solubilization of polyanionic species from pore surfaces.

In the presence of significant concentrations of free iron ions, inorganic phosphates precipitate, providing a concentration mechanism for this otherwise scarce resource. Surface-catalysed phosphoryl transfer from acetyl phosphate, available from acetyl thioesters (Weber, 1981), generates pyrophosphate that accumulates under these conditions (de Zwart, et al., 2004) and becomes a second source of dehydrating power in water (Baltscheffsky, 1997) once it can be solubilized. Iron(II) phosphates are sparingly soluble salts (Pratt, 2006); organic phosphates have significantly higher solubility than inorganic phosphates and, when the quantities of iron present are limiting, these are selectively desorbed into solution. For example, under conditions where there is competition for iron, phosphate and pyrophosphate are selectively precipitated in the presence of glycerol phosphates leaving the latter free in solution (Pratt, et al., 2009). Thus a selection mechanism for the utilization of soluble organo-phosphates, *e.g.* sugar phosphates, arises. As surface-bound inorganic phosphates react with organic species generated by proto-metabolism they selectively desorb into solution and become integrated with the thioester and amino acid based catalytic networks. Precipitated sparingly soluble iron phosphate, iron pyrophosphate and iron sulfide, provide a homeostatic backdrop to the emerging proto-metabolic networks, with concentrations adjusting as catalysis consumes proto-metabolites. This backdrop became an essential feature in the subsequent development of an RNA world.

Reproduction, mobility and selection. As individual pores evolve soluble proto-metabolic networks, some of the materials are washed to neighbouring pores where they can seed new autocatalytic networks: ligands can carry metal ions and influence the coordination chemistry, and hence catalytic activity, of metal sites; phosphates and other key proto-metabolites can be relocated. Productive autocatalytic networks will be more successful in seeding neighbouring pores. For simple catalytic networks this provides a selectable form of reproduction based on catalytic efficiency. However, the amount of proto-metabolic information that can be relocated in this piecemeal fashion is very limited in scope and so only simple autocatalytic networks can reproduce by this mechanism. Autocatalytic networks that develop the capability of relocating populations of catalytically active chemicals to neighbouring pores can reproduce more effectively and evolve to more complex systems.

There will be a range of solubilities amongst the components of the emerging autocatalytic networks: both the proto-metabolites and the oligopeptide-encapsulated metal catalysts. Amphipathic molecules that arise, such as some of the oligopeptide complexes and any fatty acids present, will aggregate to form higher order structures including micelles and vesicles (Deamer, et al. 2006). Hydrothermal concentration mechanisms will facilitate the generation of such structures (Budin, et al., 2009). The resulting micelles and vesicles will be heterogeneous aggregates of chemicals that will be relocated to neighbouring pores en masse. This will act as a selection mechanism for reproducing more complex networks. More sophisticated and productive networks will be relocated to new environments in which they will have access to renewed chemical feedstocks.

A stochastic corrector model of metabolic reproduction. Lipopeptide encapsulation allows relocation of multiple catalysts and proto-metabolites as envisaged by autocatalytic network theories, *e.g.* the GARD model (Shenhav et al., 2007). Individual components will be distributed between lipopeptide vesicles in a stochastic manner. As long as a representative sample of the constituents of the autocatalytic network are present then the catalytic cycles in the vesicle will be fully active. Such vesicles can relocate, grow and divide (Szostak, et al., 2001) in the buffered environment of the hydrothermal pores. Omission of any critical species will lead to compromised networks that will reproduce more slowly, if at all, and fail to compete with fully functional networks. This situation is analogous to the stochastic corrector model developed by Szathmáry to describe the group selection of populations of replicators in an RNA world scenario (Szathmáry and Demeter, 1987; Grey, et al. 1995). An analogous stochastic corrector model for catalysts (Figure 4) leads to the selection of functional reproducing networks of metabolic information (Shenhav, et al., 2007).

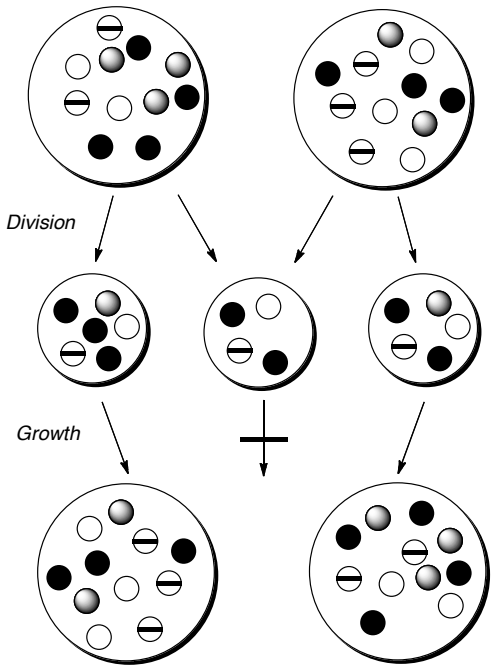


Figure 4: A stochastic corrector model of metabolic reproduction. Only vesicles containing representative populations of catalysts can grow and divide efficiently.

Early vesicular structures would be loose dynamic associations. These allow exchange of material with the

environment so new feedstocks can be taken up. Furthermore, discrete vesicles can fuse on contact allowing deficient vesicles to regenerate fully functioning autocatalytic networks and for growing vesicles to generate new combinations of metabolic processes via symbiotic events.

Two significant features limit the complexity of such systems: the statistical distribution of molecules provides a limit to the number of discrete components that can be reliably distributed during growth and division cycles; in addition the accuracy of metabolic turnover is limited by the lack of precision in the ordering of monomers in oligomer-based catalysts where specificity arises from simple chemical selectivity, rather than the degree of control that can be exerted by macromolecular catalysts (enzymes and ribozymes) of well-defined sequences. Such autocatalytic systems can develop general classes of proto-metabolic function involving the presence or absence of particular processes; however, as Szathmáry and colleagues have shown (Vasas, et al. 2010), these systems are not capable of open-ended Darwinian evolution where incremental variants can be selected and maintained in populations of competing entities. Nevertheless, the proto-metabolic history is likely to vary from one set of hydrothermal pores to another with the resulting autocatalytic networks being a function of the particular local geochemistry.

The Chemoton: Reproduction Fidelity and the Analogue-to-Digital Information Transition

Digital molecular information. Fidelity of the reproduction of biochemical information is a critical selection pressure for the development of complex organisms. Eigen's work has highlighted the critical role of error threshold limits in the reproduction of biochemical information in simple replicator systems (Eigen and Schuster, 1977). The fundamental discovery needed for the generation of digital information, in the form of well-defined macromolecular sequence information, was the generation of oligomers capable of carrying information but whose physical properties are approximately independent of composition. Benner (2004) has noted the importance of linear poly-ionic oligomers, built from monomeric units of similar size, structure and identical charges, in providing the requisite properties for genetic molecules. The ability of phosphate to link two units and retain a negative charge is critical to the structure and function of nucleic acids (Westheimer, 1987).

Some proto-metabolic networks provided a range of features that facilitated the development of RNA-based coding systems. They provided access to metallo-oligopeptide catalysts that generated both organic molecules and dehydrating power in water. They also manipulated phosphate precipitation equilibria, by encapsulating free divalent metal ions thereby allowing release of solubilized organo-phosphate species from precipitated stores. The ability of phosphate to channel sugar chemistry to useful metabolites (Muller, et al. 1990; Eschenmoser and Loewenthal, 1992) could then be exploited opening the way to nucleotide derivatives (Powner, et al. 2009). Once phosphate precipitation equilibria were made freely reversible by cation binding, pyrophosphate from autocatalytic iron-sulfur networks became a more general source of activated phosphate species (Baltscheffsky, 1997). It was also possible to exploit reversible surface binding of oligomeric sugar phosphate species, including oligonucleotides (Hatton and Rickard, 2008) to allow templated oligomer synthesis (Joshi, et al. 2007).

Once sugar phosphate derivatives, including rudimentary nucleotide analogues, became available to proto-metabolism their oligomerization was subject to the same molecular selection processes that refined the properties of simple oligopeptides. Oligomeric derivatives that provided useful catalytic activity enhanced the productivity of the protocells and were produced faster than they hydrolysed. They were initially selected on this basis. In this way mixed proto-metabolic networks arose in which catalysis was carried out by both oligopeptide complexes and oligonucleotide derivatives (White, 1976). The oligopeptide and oligonucleotide systems interfaced via simple amino-acylated nucleotide derivatives. Amino acids linked as esters to nucleotides could undergo a version of templated amide formation, facilitated by base-stacking of the nucleotide component. This provided a rudimentary precursor to translation.

Once catalytically useful oligomeric nucleotide derivatives emerged a second property was selected: namely the replication mechanisms associated with access to precise ordering of monomer units inherent in nucleic acid structures (Sievers and von Kiedrowski, 1994). This provided the basis for DNA replication. The co-evolution of translation occurred via increasingly precise versions of templated oligopeptide synthesis (Hsiao, et al. 2009). This was the final technology needed for the creation of replicators with a proto-metabolism built on an inter-dependent combination of iron sulfur catalysis, oligopeptides and oligonucleotides.

The continuing action of evolution, with replication fidelity as a key selection pressure (Eigen and Schuster 1977, 1978a and 1978b), set the stage for the emergence of a modified version of the RNA world (Gesteland, et al. 2006; Koonin and Martin, 2005) in which oligopeptide- and oligonucleotide-derived catalysts co-existed within reproducing vesicles. In these systems the oligonucleotides developed a unique function as a repository for precise replicable sequence information: open-ended Darwinian evolution had emerged. This was harnessed as the basis for coding oligopeptides of reproducible sequence via the refinement of translation. The resulting enhancement in the catalytic specificity of oligopeptides provided ever more efficient variants on metabolism. The same opportunities and evolutionary driving forces led to protocell membranes becoming more rigid barriers to the outside world once precise transport mechanisms became available via protein evolution. The resulting entities were the first true chemotons having the irreducible complexity associated with living cells.

Concluding remarks

The model presented here provides a plausible account of a combination of specific prebiological processes that explain the early steps by which a functional chemoton, with three interdependent sub-systems, can emerge. By this account life is not inevitable, but requires an ordered sequence of proto-metabolic innovations. Porous hydrothermal mineral mounds

provided an exceedingly large number of discrete geochemical environments that allowed parallel testing of vast numbers of chemical systems. Complex chemotons arose as a result of a series of molecular selection processes occurring within these environments. This model is potentially testable *e.g.* via combinatorial microfluidic technology (Kreutz, et al. 2010) with screening of diverse chemical systems for proposed proto-metabolic innovations.

It is proposed that the creation and selection of metabolic diversity occurred via simple chemical and physical steps. Initially selection was based on catalytic efficiencies of networks that emerged in specific pre-existing mineral micropore compartments. Encapsulation of metal species by organic ligands provided more active and specific catalysts and also allowed the development of a soluble proto-metabolism incorporating sugar phosphates. Systems that evolved the capacity to relocate en masse in lipopeptide vesicles, before their access to chemical feedstocks ends, selectively propagated. Protocells emerged with autocatalytic networks that included catalysts based on both oligopeptides and oligonucleotides which could then evolve complex oligonucleotide structures via molecular evolution. These first chemotons were the forerunners of an RNA world that evolved by open-ended Darwinian evolution.

Acknowledgments. I am deeply grateful to Mark Dörr, Vladimir Golovko, Janos Hajdu, Ant Poole and Mike Russell for insightful discussions.

References

Baaske, P., Weinert, F. M., Duhr, S., Lemke, K. H., Russell, M. J. and Braun, D. (2007) Extreme accumulation of nucleotides in simulated hydrothermal pore systems. *Proceedings of the National Academy of Sciences* USA 104:9346-9351.

Baltscheffsky, H. (1997) Major "anastrophes" in the origin and early evolution of biological energy conversion. *Journal of Theoretical Biology* 187:495-501.

Benner, S. A. (2004) Understanding nucleic acids using synthetic chemistry. *Accounts of Chemical Research* 37:784-797.

Beinert, H., Holm, R. H. and Münck, E. (1997) Iron-sulfur clusters: nature's modular, multipurpose structures. *Science* 277:653-659

Berkovitch, F., Nicolet, Y., Wan, J. T., Jarrett, J. T. and Drennan, C. L. (2004) Crystal structure of biotin synthase, an *S*-adenosylmethionine-dependent radical enzyme. *Science* 303:76-79.

Blochl, E., Keller, M., Wächtershäuser, G. and Stetter, K. O. (1992) Reactions depending on iron sulfide and linking geochemistry with biochemistry. *Proceedings of the National Academy of Sciences* USA 89:8117-8120.

Budin, I., Bruckner, R.J. and Szostak, J. W. (2009) Formation of protocell-like vesicles in a thermal gradient column. *Journal of the American Chemical Society* 131:9628-9629.

Cheng, V. W. T., Ma, E., Zhao, Z., Rothery, R. A. and Weiner, J. H. (2006) The iron-sulfur clusters in *Escherichia coli* succinate dehydrogenase direct electron flow. *Journal of Biological Chemistry* 281:27662-27668.

Cody, G. D., Boctor, N. Z., Filley, T. R., Hazen, R. M., Scott, J. H., Sharma, A. and Yoder Jr, H. S. (2000) Primordial carbonylated iron-sulfur compounds and the synthesis of pyruvate. *Science* 289:1337-1340.

Costisor, O. and Linert, W. (2004) *Metal mediated template synthesis of ligands* (World Scientific Publishing Co., Singapore).

de Duve, C. (1991) *Blueprint for a Cell: The Nature and Origin of Life* (Patterson, Burlington, NC).

de Zwart, I. I., Meade, S. J. and Pratt, A. J. (2004) Biomimetic phosphryl transfer catalysed by iron(II)-mineral precipitates. *Geochimica et Cosmochimica Acta* 68: 4093-4098.

Deamer, D., Singaram, S., Rajamani, S. and Kompanichenko, V. (2006). Self-assembly processes in the prebiotic environment. *Philosophical Transactions of the Royal Society* B361:1809-1818.

Dörr, M., Käßbohrer, J., Gunnert, R., Kreisel, G., Brandt, W. A., Werner, R.A., Geilmann, H., Apfel, C., Robl, C. and Weigand, W. (2003) A possible prebiotic formation of ammonia from dinitrogen on iron sulfide surfaces. *Angewandte. Chemie International Edition in English*, 42:1540-1543.

Dyson, F. (1999) *Origins of Life* (Cambridge University Press, Cambridge, UK) 2nd Ed.

Eigen, M. and Schuster, P. (1977) The hypercycle a principle of natural self-organisation. Part A: emergence of the hypercycle. *Naturwissenschaften* 64:541-565.

Eigen, M. and Schuster, P. (1978a) The hypercycle a principle of natural self-organisation. Part B: the abstract hypercycle. *Naturwissenschaften* 65:7-41.

Eigen, M. and Schuster, P. (1978b) The hypercycle a principle of natural self-organisation. Part C: the realistic hypercycle. *Naturwissenschaften* 65:341-369.

Einsle, O., Tezcan, F. A., Andrade, S. L. A., Schmid, B., Yoshida, M., Howard, J. B. and Rees, D. C. (2002**)** Nitrogenase MoFe-protein at 1.16 Å resolution: a central ligand in the FeMo-cofactor. *Science* 297**:**1696-1700.

Eschenmoser, A. (1988). Vitamin B_{12}: experiments concerning the origin of its molecular structure. *Angewandte Chemie International Edition in English*, 27:5-39.

Eschenmoser, A. and Loewenthal, E. (1992) Chemistry of potentiaIly prebiologicaI naturaI products. *Chemical Society Reviews* 21:1-16.

Flint, D. H. and Allen, R. M. (1996) Iron-sulfur proteins with nonredox functions. *Chemical Reviews* 96: 2315-2334.

Fraústo da Silva, J. J. R. and Williams, R. J. P. (2001) *The biological chemistry of the elements: the inorganic chemistry of life* (Oxford University Press, Oxford, UK) 2nd Ed.

Gánti, T (2003) *The Principles of Life*, (Oxford University Press, Oxford, UK).

Gesteland, R. F., Cech, T.R. and Atkins, J.F. eds. (2006) *The RNA World* (Cold Spring Harbor Lab. Press, Plainview, NY), 3rd Ed.

Grahame, D. A. (2003) Acetate C–C bond formation and decomposition in the anaerobic world: the structure of a central enzyme and its key active-site metal cluster. *Trends in Biochemical Sciences* 28:221-224.

Grey, D., Hutson, V. and Szathmáry, E. (1995) A re-examination of the stochastic corrector model. *Proceedings of the Royal Society of London B*262:29-35.

Hanczyc, M. M., Fujikawa, F. M. and Szostak, J. W. (2003) Experimental models of cellular compartments: encapsulation growth and division. *Science* 302:618-622.

Hatton, B. and Rickard, D. (2008) Nucleic acids bind to nanoparticulate iron (II) monosulphide in aqueous solutions. *Origins of Life and Evolution of the Biosphere* 38:257-270.

Hegg, E. L. (2004) Unraveling the structure and mechanism of acetyl-coenzyme A synthase. *Accounts of Chemical Research* 37:775-783.

Horowitz, N. H. (1945) On the evolution of biochemical syntheses. *Proceedings of the National Academy of Sciences* USA, 31:153-157.

Hsiao, C. L., Mohan, S., Kalahar, B. K. and Williams, L. D. (2009) Peeling the Onion: Ribosomes are ancient molecular fossils. *Molecular Biology and Evolution* 26:2415-2425.

Huber, C. and Wächtershäuser, G (1997) Activated acetic acid by carbon fixation on (Fe,Ni)S under primordial conditions. *Science* 276: 245-247.

Huber, C. and Wächtershäuser, G (1998) Peptides by activation of amino acids with CO on (Ni,Fe)S surfaces: Implications for the origin of life. *Science* 281:670-672.

Huber, C. and Wächtershäuser, G (2003) Primordial reductive amination revisited. *Tetrahedron Letters* 44: 1695-1697.

Jensen, R. A. (1976) Enzyme recruitment in evolution of new function. *Annual Review of Microbiology* 30:409-425.

Joshi, P. C., Pitsch, S. and Ferris, J. P. (2007) Selectivity of montmorillonite-catalysed prebiotic reactions of D,L-nucleotides. *Origins of Life and Evolution of the Biosphere* 37:3-26.

Kacian, D.L., Mills, D. R., Kramer F. R. and Spiegelman, S. (1972). A replicating RNA molecule suitable for a detailed analysis of extracellular evolution and replication. *Proceedings of the National Academy of Sciences* USA 69: 3038–3042

Koonin, E. V. and Martin, W. (2005) On the origin of genomes and cells within inorganic compartments. *Trends in Genetics* 21:647-654.

Kreutz, J. E., Shukhaev, A., Du, W., Druskin, S., Daugulis, O. and Ismagilov, R. F. (2010) Evolution of catalysts directed by genetic algorithms in a plug-based microfluidic device tested with oxidation of methane by oxygen. *Journal of the American Chemical Society*, 2010, 132:3128–3132.

Lennie, A.R. Redfern, S. A. T., Champness, P. E., Stoddart, C. P., Schofield, P. F. and Vaughan, D. J. (1997) Transformation of mackinawite to greigite: an *in situ* X-ray powder diffraction and transition electron microscope study. *American Minerologist*, 82:302-309.

Luisi, P. L., Ferri, F. and Stano, P. (2006) Approaches to semi-synthetic minimal cells: a review. *Naturwissenschaften*, 93:1-13.

Morowitz, H. J., Kostelnik, J. D., Yang, J. and Cody, G. D. (2000) The origin of intermediary metabolism. *Proceedings of the National Academy of Sciences* USA, 97:7704–7708.

Muller, D., Pitch, S., Kittaka, A., Wagner, E., Wintner, C. E. and Eschenmoser, A. (1990) Chemistry of α-aminonitriles – aldomerisation of glycolaldehyde phosphate to *rac*-hexose 2,4,6-triphosphates and (in presence of formaldehyde) *rac*-pentose 2,4-diphosphates – *rac*-allose 2,4,6-triphosphate and *rac*-ribose 2,4-diphosphate are the main reaction products. *Helvetica Chimica Acta* 73:1410-1468.

Nicolet, Y. and Drennan, C. L. (2004) AdoMet radical proteins - from structure to evolution - alignment of divergent protein sequences reveals strong secondary structure element conservation. *Nucleic Acids Research* 32:4015-4025.

Nicolet, Y., Lemon, B. J.,. Fontecilla-Camps, J. C. and Peters, J. W. (2000) A novel FeS cluster in Fe-only hydrogenases. *Trends in Biochemical Sciences* 25:138-144.

Orgel, L. E. (2000). Self-organizing biochemical cycles. *Proceedings of the National Academy of Sciences* USA 97:12503-12507.

Powner, M. W., Gerland, B. and Sutherland, J. D. (2009) Synthesis of activated pyrimidine ribonucleotides in prebiotically plausible conditions. *Nature* 459:239-242.

Pratt, A. J. (2006) The Curious Case of Phosphate Solubility, *Chemistry in New Zealand* October:78-80.

Pratt, A. J., Golovko, V. and Toombs-Ruane, H. (2009) FeS surface dynamics and molecular evolution. *Origins of Life and Evolution of Biospheres* 39:343-344.

Ragsdale, S. W. (1991). Enzymology of the acetyl-CoA pathway of CO_2 fixation. *Critical Reviews in Biochemistry and Molecular Biology* 26: 261–300.

Rao, P. V. and Holm, R. H. (2004) Synthetic analogues of the active sites of iron-sulfur proteins. *Chemical Reviews* 104:527-559.

Rickard, D., Butler, I. B. and Oldroyd, A. (2001) A novel iron sulphide mineral switch and its implications for earth and planetary science. *Earth and Planetary Science Letters* 189:85-91.

Rickard, D. and Luther III, G. W. (2007) Chemistry of iron sulfides. *Chemical Reviews* 107:514-562.

Rozan, T. F., Lassman, M. E., Ridge, D. P. and Luther III, G. W. (2000) Evidence for iron, copper and zinc complexation as multinuclear sulphide clusters in oxic rivers. *Nature* 406:879-882.

Russell, M. J. and Hall, A. J. (1997) The emergence of life from iron monosulphide bubbles at a submarine hydrothermal redox and pH front. *Journal of the Geological Society of London* 154:377–402.

Russell, M. J. and Hall, A. J. (2006) The onset and early evolution of life. *Geological Society of America Memoir* 198:1-32.

Russell, M. J. and Kanik, I. (2010). Why does life start, what does it do, where will it be, and how might we find it? *Journal of Cosmology*, 5:1008-1039.

Russell, M. J. and Martin, W. (2004) The rocky roots of the acetyl-CoA pathway. *Trends in Biochemical Sciences* 29:358-363.

Shenhav, B., Oz, A. and Lancet, D. (2007) Coevolution of compositional protocells and their environment. *Philosophical Transactions of the Royal Society* B362:1813-1819.

Shock, E.L. (1992). Chemical environments of submarine hydrothermal systems; marine hydrothermal systems and the origin of life. *Origins of Life and Evolution of the Biosphere* 22:67-107.

Sievers, D. and Kiedrowski, G. (1994) Self-replication of complementary nucleotide-based oligomers. *Nature* 369:221-224

Svetlitchnaia, T., Svetlitchnyi, V., Meyer, O. and Dobbekh, H. (2006) Structural insights into methyl transfer reactions of a corrinoid iron–sulfur protein involved in acetyl-CoA synthesis, *Proceedings of the National Academy of Sciences* USA 103:14331-14336.

Szathmáry, E. (2007) Coevolution of metabolic networks and membranes: the scenario of progressive sequestration. *Philosophical Transactions of the Royal Society* B362:1781-1787

Szathmáry, E. and Demeter, L. (1987) Group selection of early replicators and the origin of life. *Journal of Theoretical Biology* 128:463-486.

Szostak, J. W., Bartel, D. P. and Luisi, P. L., (2001) Synthesizing life. *Nature* 409:387-390.

Trefil, J., Morowitz, H. J. and Smith, E. (2009) The origin of life a case is made for the descent of electrons. *American Scientist* 97(3):206-208.

Vasas, V., Szathmáry, E. and Santos, M. (2010) Lack of evolvability in self-sustaining autocatalytic networks constraints metabolism-first scenarios for the origin of life. *Proceedings of the National Academy of Sciences* USA, 107:1470-1475.

Volbeda, A. and Fontecilla-Camps, J. C. (2005) Structure–function relationships of nickel–iron sites in hydrogenase and a comparison with the active sites of other nickel–iron enzymes. *Coordination Chemistry Reviews* 249:1609-1619.

Wächtershäuser, G. (1988) Before enzymes and templates – theory of surface metabolism. *Microbiological Reviews* 52:452–484.

Weber, A. L. (1981) Formation of pyrophosphate, tripolyphosphate, and phosphorylimidazole with the thioester *N*,*S*-diacetyl-cysteamine as the condensing agent. *Journal of Molecular Evolution* 18:24-29.

Westheimer, F.H. (1987) Why nature chose phosphates. *Science* 235:1173-1178.

White H. B. (1976). Coenzymes as fossils of an earlier metabolic state. *Journal of Molecular Evolution* 7:101–104.

Wolthers, M., Van der Gaast, S. J. and Rickard, D. (2003) The structure of disordered mackinawite. *American Mineralogist* 88:2007-2015.

Xia, D., Yu, C-A., Kim, H., Xia, J-Z., Kachurin, A. M., Zhang, L., Yu, L. and Deisenhofer, J. (1997) Crystal Structure of the Cytochrome bc_1 Complex from Bovine Heart Mitochondria *Science* 277:60-66.

Zhang, Y., Thiele, I., Weekes, D., Li, Z., Jaroszewski, L., Ginalski, K., Deacon, A. M., Wooley, J., Lesley, S. A.,Wilson, I. A. Palsson, B., Osterman, A. and Godzik, A. (2009) Three-dimensional structural view of the central metabolic network of *Thermotoga maritima*. *Science* 325:1544-1549.

In silico Evolution of Early Metabolism

Alexander Ullrich[1], Christoph Flamm[2], Markus Rohrschneider[3] and Peter F. Stadler[1,2,4,5,6]

[1]Bioinformatics Group, Department of Computer Science, University of Leipzig, D-04107 Leipzig
[2]Institute for Theoretical Chemistry, University of Vienna, 1090 Wien
[3]Image and Signal Processing Group, Department of Computer Science, University of Leipzig, D-04109 Leipzig
[4]Max Planck Institute for Mathematics in the Sciences, D-04103 Leipzig
[5]Fraunhofer Institute for Cell Therapy and Immunology, D-04103 Leipzig
[6]Santa Fe Institute, Santa Fe, NM 87501

Abstract

We developed a simulation tool for investigating the evolution of early metabolism, allowing us to speculate on the formation of metabolic pathways from catalyzed chemical reactions and development of characteristic properties. Our model consists of a protocellular entity with a simple RNA-based genetic system and an evolving metabolism of catalytically active ribozymes that manipulate a rich underlying chemistry. Ensuring an almost open-ended and fairly realistic simulation is crucial for understanding the first steps in metabolic evolution. We show here how our simulation tool can be helpful in arguing for or against hypotheses on the evolution of metabolic pathways. We demonstrate that seemingly mutually exclusive hypotheses may well be compatible when we take into account that different processes dominate different phases in the evolution of a metabolic system. Our results suggest that forward evolution shapes metabolic network in the very early steps of evolution. In later and more complex stages, enzyme recruitment supersedes forward evolution, keeping a core set of pathways from the early phase.

Introduction

Understanding the evolutionary mechanisms of complex biological systems is an intriguing and important task of current research in biology as well as artificial life. The formation of metabolic pathways from chemical reactions has been discussed for decades and several hypotheses have been proposed since the 1940s. Research on the TIM β/α-barrel fold architecture (Copley and Bork, 2000) shows that the evolution of modern metabolism is mainly driven by enzyme recruitment, as suggested by the patchwork model (Ycas, 1974; Jensen, 1976)). Nevertheless, many aspects of the evolutionary machinery are still not well understood. In particular, the first steps in early metabolism evade observation by conventional approaches. Studies on hypotheses of pathway evolution (Caetano-Anollés et al., 2009; Morowitz, 1999) suggest that metabolism has evolved in different phases and only traces or "shadows" are still observable from the events in the very distant past. Thus, there is a need for realistic models of early metabolism that consider all its components and scales. Simulation approaches have shown to be useful in finding and challenging explanations for the evolution of biological networks (Pfeiffer et al., 2005). We have recently proposed a computational framework for the evolution of metabolism (Flamm et al., 2010), modeling all its significant components in a realistic way. In this report we discuss first results from several simulation runs.

In the next section we recapitulate four scenarios of evolution that are of particular interest to understand the formation of metabolic pathways and assessing our own results. This will be followed by a brief introduction to our computational model that we use in this study. Then we will present some general results from a series of simulation runs and investigate some of the findings in more detail on two examples. We conclude with a short discussion on the comparison of our results with existing pathway evolution hypotheses.

Scenarios of Evolution

In this section, we elucidate four relevant hypotheses on the evolution of metabolism in general and formation of metabolic pathways in specific. For more a more detailed discussion of the theories of pathway evolution we refer to the reviews by Caetano-Anollés et al. (2009) and Schmidt et al. (2003) discussing further theories of pathway evolution.

Backward Evolution

Backward (or retrograde) evolution was one of the first theories for the evolution of metabolic pathways, proposed by Horowitz (1945). It assumes that an organism is able to make use of certain molecules from the environment. However, individuals that can produce these beneficial molecules by themselves gain an advantage in selection in the case of depletion of the "food source". Therefore, new chemical reactions are added that produce beneficial molecules from precursors that are abundant in the environment or that are produced in turn by the organism's metabolism. As a consequence, one should observe more ancient enzymes downstream in present-day metabolic pathways. Towards the entry point of the pathway, younger and younger going enzymes should be found (see Figure 1(a)).

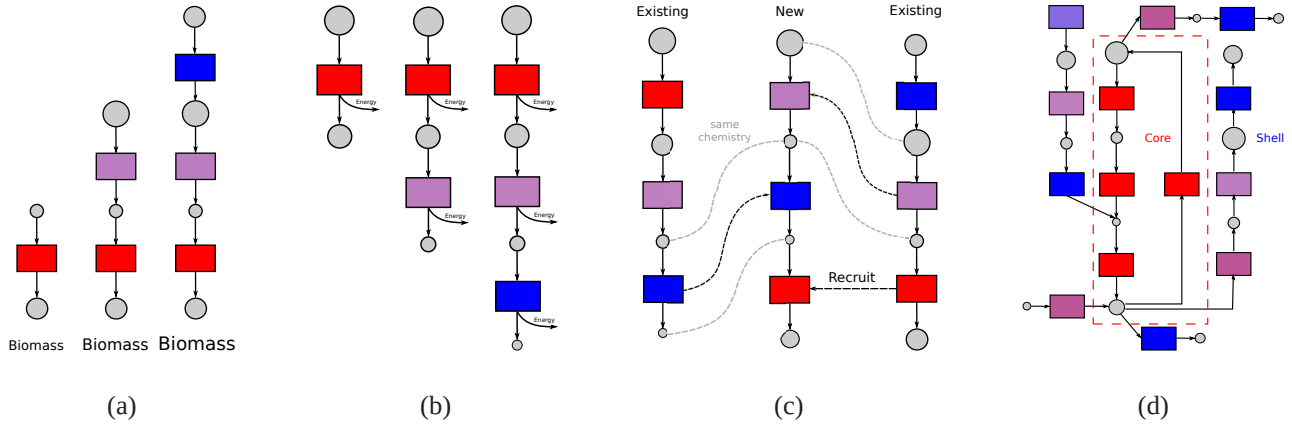


Figure 1: Hypotheses about the formation and evolution of metabolic pathways. (a) Backward evolution, (b) Forward evolution, (c) Patchwork model, (d) Shell hypothesis. Colored squares represent enzymes, gray circles are metabolites. Color encoding for enzymes stand for their age, red being older and blue being younger enzymes.

Forward Evolution

Forward evolution could be seen as an extension or counterpart of the backward evolution hypothesis, reversing the direction of pathway evolution. Granick (1957), and later Cordon (1990), argue for a pathway evolution in forward direction, requiring that the intermediates are already beneficial to the organism. This is in particular plausible for catabolic pathways, where the organism can extract more energy by breaking food molecules downs to simpler and simpler end products. Older enzymes are then expected to be upstream in the pathway, with younger enzymes appearing further downstream (see Figure 1(b)).

Patchwork Model

The patchwork model (Ycas, 1974; Jensen, 1976) explains the formation of pathways by recruiting enzymes from existing pathways. The recruited enzymes may change their reaction chemistry and metabolic function in the new pathways and specialize later trough evolution. This introduction of new catalytic activities lead to a selective advantage. Looking at the constitution of a pathway formed by enzyme recruitment, we should observe a mosaic-like picture of older and younger enzymes mixed throughout the pathway (see Figure 1(c)).

Shell Hypothesis

The shell hypothesis was proposed by Morowitz (1999). It argues for the case of the reductive citric acid cycle that in the beginning an auto-catalytic core is formed from which new catalytic activities and pathways could be recruited and fed. Thus a metabolic shell would form around this core. Enzymes in the core would likely be less prone to mutational changes because they are essential for the organism. Thus, one should still be able to observe a core of ancient enzymes (see Figure 1(d)).

Model

The computational model, summarized schematically in Figure 2, is composed of a genetic and a metabolic subsystem. The genetic subsystem is implemented as a cyclic RNA genome. A special sequence motif indicates the start of genes which are of constant length. The RNA sequence corresponding to the "coding sequence" of a gene is folded into the (secondary) structure using the `Vienna RNA Package` (Hofacker et al., 1994) (Step A in Figure 2).

During chemical reactions bond formation/breaking is confined to a small subset of atoms of the reacting molecules. A cyclic graph abstraction, called the imaginary transition state (ITS) (Fujita, 1986), can be used to capture the changes in the reactive center (Hendrickson, 1997). Furthermore, over 90% of all known organic reactions can be classified by their ITS (Hendrickson and Miller, 1990) and organized in a hierarchical structure (Herges, 1994). Sequence and structure features of the folded RNA gene products are mapped into the classification tree of organic reactions for functional assignment of the catalytic set (Step B in Figure 2). Thus we have implemented an evolvable sequence-to-function map (Ullrich and Flamm, 2009), allowing the metabolic organization to escape from the confines of the chemical space set by the initial conditions of the simulation.

The metabolic subsystem is built upon a graph-based artificial chemistry (Benkö et al., 2003) endowed with a built-in thermodynamics. To generate the metabolic reaction network, induced by the catalytic set (chemical reactions decoded from the genome) on the set of metabolites (chemical molecules of interest from user input), a rule-based stochastic simulation is performed, where the likelihood of a reaction being chosen depends on its reaction rate (Faulon and Sault, 2001). Reaction rates are calculated "on the fly" from the chemical graphs of the reactants.

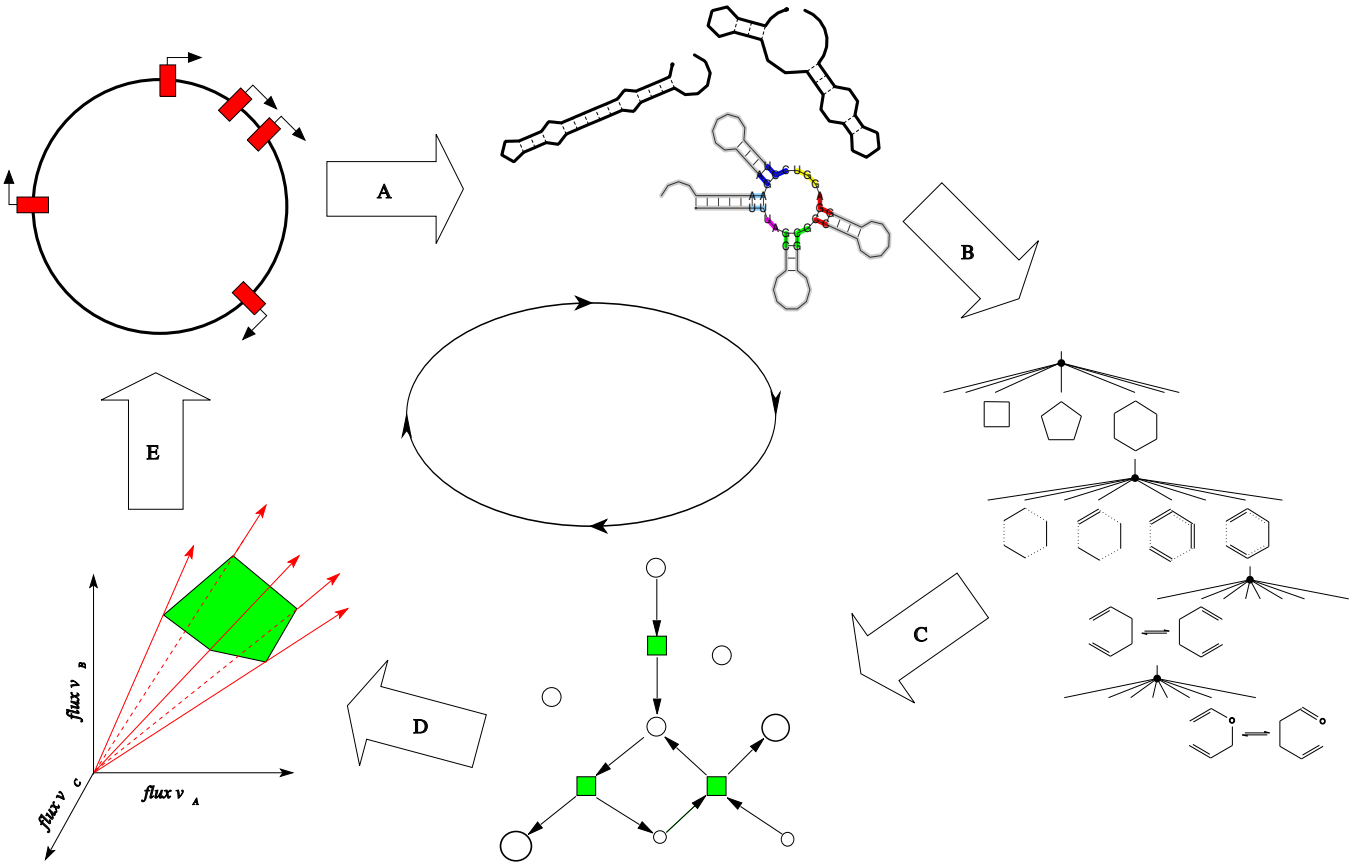


Figure 2: Scheme of the simulation system. (A) Decoding of (RNA-)genes to catalytic molecules; (B) Assignment of catalytic functions to "ribozymes", through mapping from structural and sequential information of the RNA molecule to a reaction logo in the hierarchy (Hendrickson, 1997); (C) Construction and stochastic simulation of the metabolic network; (D) Metabolic Flux analysis and fitness evaluation; (E) Application of genetic variation operators.

To identify the elementary flux modes, i.e., extreme pathways (Gagneur and Klamt, 2004), of the resulting reaction network, a metabolic flux analysis is performed. (Step D in Figure 2). The fitness of an organism is computed as the maximum of the (linear) yield function (e.g. biomass production) over all extreme pathways. Finally, genetic variation operators are applied to the genome (Step E in Figure 2). For a detailed discussion of the various steps of the computational model we refer the reader to Flamm et al. (2010).

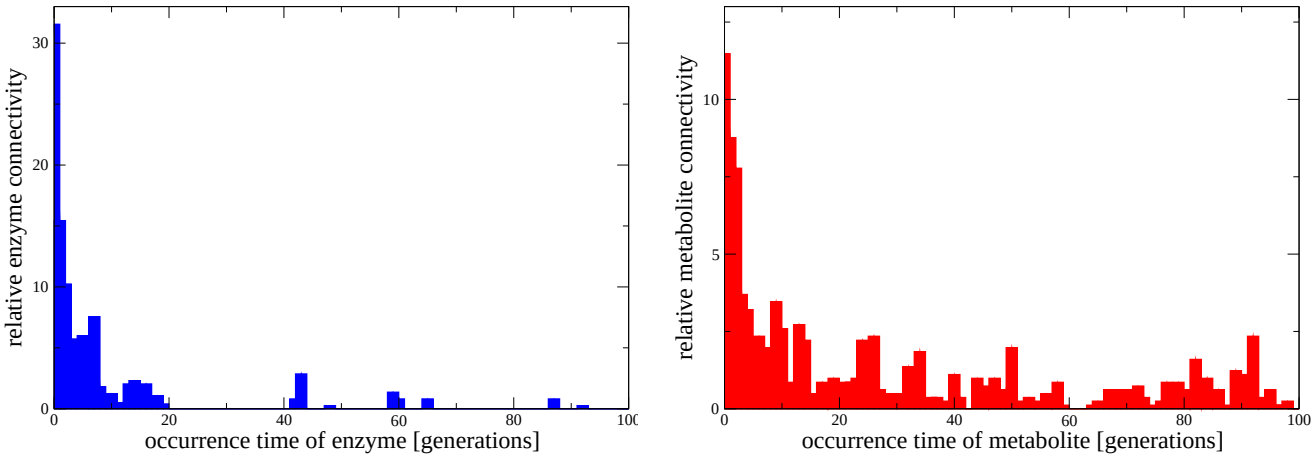
Simulations and Results

In this section we use the computational model described above to simulate the evolution of metabolic networks and analyze the change of its structure and components over several generations. All simulation runs performed for this paper were initialized with the full set of chemical reactions to chose from, the same configurations for genome length (5000 bases), and the same TATA-box constitution ("UAUA") and fixed gene length (100 bases). They differ in initial conditions, population size, environmental conditions, selection criteria, and simulation time (number of generations and stochastic simulation steps).

Quantitative Analysis

To gain some quantitative insights into the general principles of metabolic evolution we performed a series of simulation runs to investigate certain measures that give a picture of the evolutionary constitution of the metabolic networks throughout the evolution process.

In a previous study (Ullrich and Flamm, 2008), we already showed that our metabolic networks evolved certain properties such as a scale-free node degree distribution and the existence of hub-metabolites. An investigation of the enzyme connectivity suggested that enzymes from early stages show a higher connectivity than those from later stages. Here, we confirm these findings with a much larger sample of 100 simulation runs starting from the same set of initial metabolites (cyclobutadiene, ethenol, phthalic anhydride, methylbutadiene, and cyclohexa-1,3-diene). Figure 3(a) shows a clear trend for enzymes from the first gen-



(a) (b)

Figure 3: Average relative connectivity of (a) enzymes and (b) metabolites introduced in the same generation, for 100 generations. The height of the bars shows the fraction of the overall connections that are accounted by enzymes/metabolites from a particular generation. All values are averages over 100 simulation runs. Input molecules are not considered in the statistic, they account for nearly 50 percent of metabolite connectivity.

erations to be responsible for the major part of connections in the metabolic network. On the one hand, this can be explained simply due to the fact that enzymes that enter the system earlier have more time to form connections. On the other hand, this observation could also indicate that enzymes with higher and higher specificity evolve in the later stages. It could be anticipated, that enzymes with all specificities still appear in later generations but only specific enzymes catalyzing few reactions are taken to the next generation, while multi-functional enzymes are discarded because they would change the structure of the network too rigorously. Considering the connectivities of metabolites (see Figure 3(b)), we still find the highly connected nodes in the early steps, especially if we consider environment metabolites that are always abundant which account for about 50 percent of connectivity. However, there is constant production of metabolites potentially becoming highly connected.

In order to find arguments for some of the evolution hypotheses, we study the occurrence time (age) of reactions and metabolites along pathways. It is of particular interest to determine in which direction (downwards – with the flow of mass, or upwards – against mass-flow) pathways are form by addition of chemical reactions that recruit or produce new metabolites. We will use the term forward (backward) link if, in a pair of reactions in a pathway, the successor is evolutionary older (younger). In the same vein, a forward (backward) link between metabolites refers to a situation in which the products of a reaction are evolutionarily older (younger) than the educts. Accordingly, we define forward (backward) pathways as pathways in which there is at least one forward (backward) link and no backward (forward) link. Given these definitions, we compute the set of extreme pathways for every generation and all cells. For each pathway we then determine the percentage of forward and backward links and pathways, for both reactions and metabolites.

For this study, we performed 100 runs with a population size of 100 cells running for 100 generations and performing 100 network expansion (stochastic simulation) steps per generation, the input molecules were cyclobutadiene, ethenol, phthalic anhydride, methylbutadiene, and cyclohexa-1,3-diene. In Figure 4 we see the change from generation to generation in the constitution of the metabolic networks regarding our measures of forward/backward links and pathways. Considering the reactions of the networks, one can see that in the first generations, the networks consist mainly of links and pathways conforming to the forward evolution scenario. However, in later generations we observe a much more mixed mosaic like picture arguing in favor of the patchwork model. This trend becomes even more evident from the metabolite's point of view: almost all pathways consist of forward and backward links in equal numbers. Another observation from the reaction's point of view is that most forward pathways from the early stages remain even in the last stages, which could mean that they form a core of pathways that are not subject to evolutionary change. This supports the shell hypothesis. So far, our simulation results do not provide any support for the backward evolution scenario. However, so far we have not simulated an environment with temporary depletion of "food" metabolites, which is one of the major assumption of this theory. A future study considering this impact of variations in resource abundances might bring new insights on this matter.

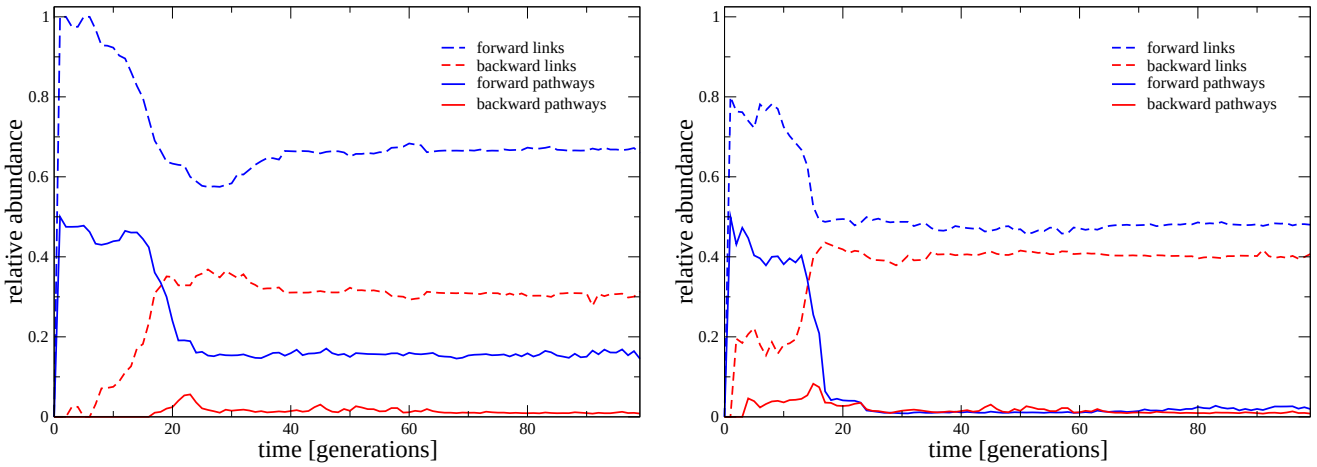


Figure 4: Evolutionary history of simulated metabolic networks. For the first 100 generations, we show the number of links and pathways that conform to the forward and backward evolution scenarios, respectively. Links are pairs of (a) consecutive reactions or (b) consecutive metabolites along a pathway. A pathway is identified as "forward-evolved" if at least one of its links is forward and none backward. In the first generations, the network consists predominantly of forward (reaction) links and pathways. After about 20 generations, the relative abundance of forward pathways decreases drastically but quickly reaches a persistent plateau value.

Example

In the following we illustrate some of our findings from the previous study in more detail for an example simulation. We use data from a simple simulation run, starting with only two input molecules and developing only few enzymes, for the visualization of an evolutionary time series (see Figure 5) an animation of the network evolution (see Additional Files) and the reaction- and metabolite-lifetime overviews (see Figure 6). The genome, and hence the set of enzymes, is chosen at random in the beginning. The two input molecules of this simulation are cyclic and sequential glucose. The simulation run is kept to 100 generations. We focus again on the evolutionary constitution of the metabolic network, i.e. investigating the relation between the occurrence time (age) of chemical reactions and their position in the network (downstream vs upstream) to draw conclusions about one of the evolution scenarios being at work. The four snapshots in Figure 5 showing the metabolic network in different stages are aligned to a union graph over all generations (Rohrschneider et al., tted). Thus, we can see that in the first steps the reactions upward in the network are added. The pathways are formed further in this forward direction. Looking at the last generation, basically all pathways from source to sink follow the forward evolution scenario. This observation is further supported by the interval graph for all chemical reactions in Figure 6. The reactions are here ordered according to their position in the graph. There is a clear trend of older reactions being on the top (upstream) and younger ones following more downstream. The colored bar next to the interval graph shows the pattern of the relation between age and position of reactions and metabolites for our example simulation run. The other three bars show the patterns for backward, forward evolution and the patchwork model, respectively. The forward evolution pattern comes closest to the simulated pattern.This illustrates again the speculation from the general analysis that in the early phase of metabolic evolution, forward evolution seems to be dominant. However, for metabolites we do not see a clear relation between the position along pathways or the network and their first appearance in the system. Similar to the general results, a much more mixed picture is observed for the metabolites. Therefore, no clear explanation can be made for the metabolite constitution.

Another, more complex, setting is used in a simulation run in which we investigate the evolutionary history of the involved genes/enzymes, depicted in the catalytic function genealogy for all generations (Figure 7). The simulation takes the same five input molecules from the above general study, but with a higher mutation and duplication rate and runs for a total of 2000 generations. Our simulation frameworks allows us to study the divergence and convergence of catalytic functions (Almonacid et al., 2010) since we can record the genealogy of each gene (reaction catalyst) throughout a simulation run, and we can utilize the ITS classification of the catalyzed reaction as a representation of the enzymatic function. Divergence of function is caused by

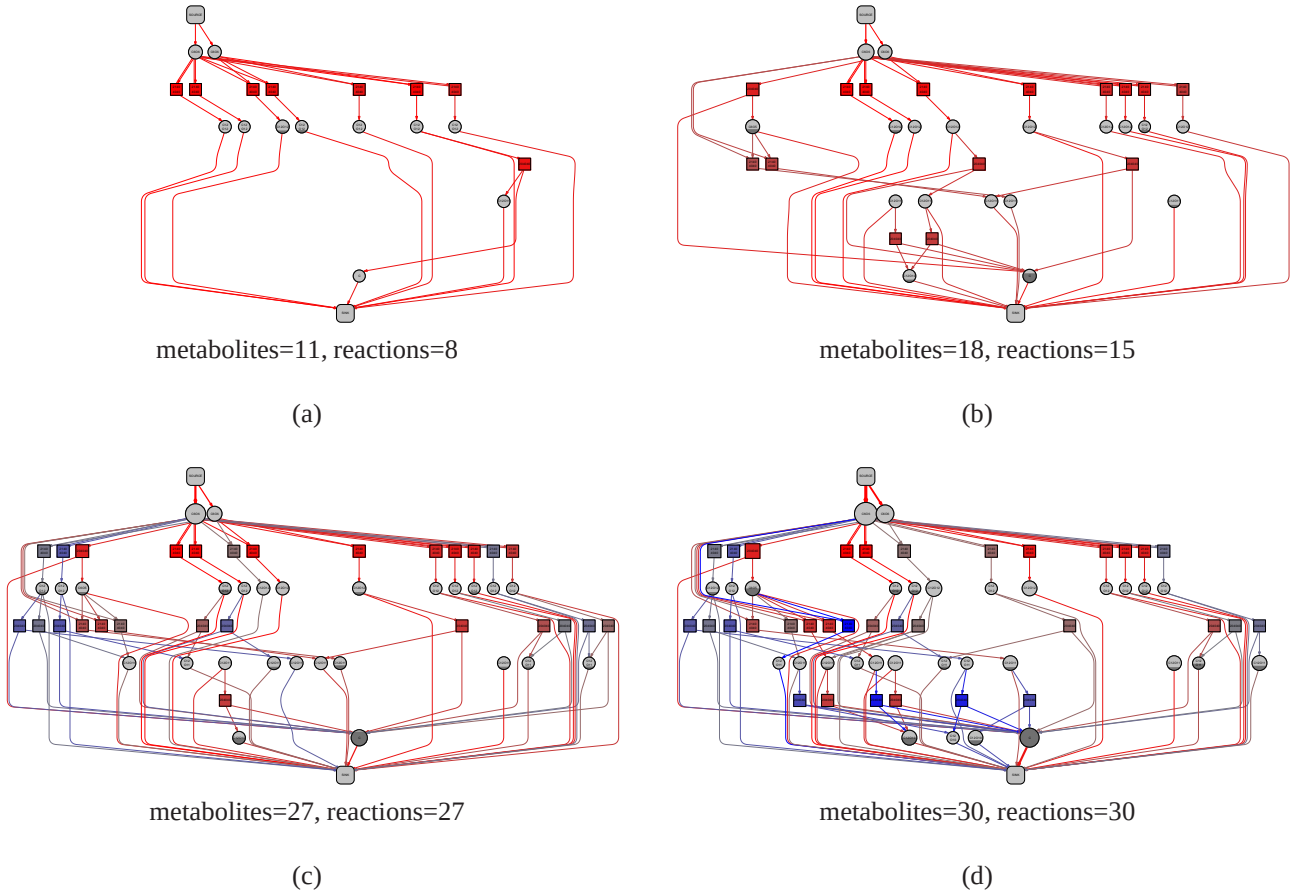


Figure 5: A series of simulated metabolic networks after (a) 10, (b) 30, (c) 66, and (d) 100 generations. Colored squares represent chemical reactions, gray circles represent metabolites. Metabolites involved in a reaction are connected to it in the network graph. The size of the nodes and the width of the edges encode for the number of extreme pathways in which the respective object is involved. The coloring for the reactions encode their age, where red stands for older (occurrence in early generation) and blue for newer (later generation) reactions.

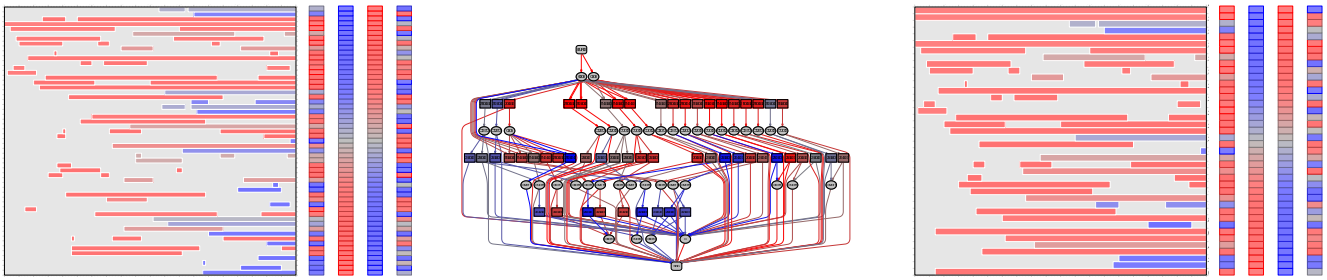


Figure 6: Life-time diagram for reactions and metabolites. (a) Life-time of reactions, (b) union network graph over all 100 generations, (c) life-time of metabolites. The reactions and metabolites (rows) in the life-time diagrams are positioned corresponding to their position in the union network graph, i.e. reactions/metabolites close to the source metabolites are in upper positions, reactions/metabolites close to the sink metabolites are placed at the bottom. The rows have colored entries if the corresponding reaction/metabolite was present at a certain generation (columns 1-100). We use the same coloring scheme as above, older reactions/metabolites are red, newer blue. The colored bars show the age distribution of reactions in the network in the same order as in the lifetime overview. The first bar represents our results, following the pattern for backward evolution, forward evolution and the patchwork model.

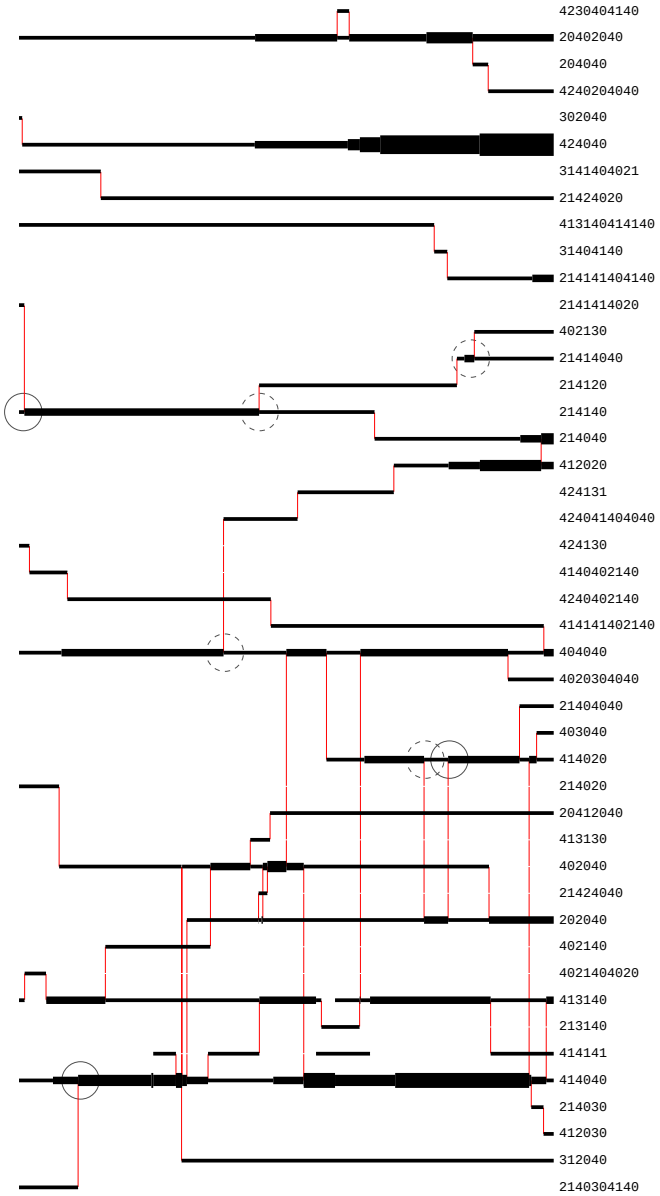


Figure 7: Genealogy of catalytic functions and gene dosage over 2000 generations. Each row represents an observed catalytic function. Black horizontal lines indicate time intervals in which genes coding for that catalytic function were present in the genome (0-200: from left to right). The thickness of the black lines indicates the number genes with a given function. Thin vertical red lines indicate points where the accumulation of mutations caused a transitions between catalytic functions. If the number of genes copies in a function class increases without a transition from another gene, then the increase is due to a gene duplication. A new gene can be created in the genome through the fortuitous formation of a TATA-box. Conversely, a gene can vanish if its TATA-box is destroyed by mutation. On the left of the chart a numerical encoding of the graph transformations performed by the "enzyme" is plotted.

gene duplication followed by sequence mutations, creating functionally different but structurally related catalysts. Convergence of function happens when catalysts from genealogically unrelated genes independently accumulate mutations resulting in the catalysis of the same reaction (or class of reactions). In Figure 7 convergence events are marked by circles. A small selection of divergence events, which are very frequent in our simulations, are marked by broken circles. Furthermore, the analysis of the functional transitions on the basis of the ITS graphs reveals that catalysts can alter their substrate specificity by small changes of the context of the graph rewrite rule, i.e. the necessary precondition for the applicability of the graph transformation rule.

Conclusions

We have introduced a simulation tool that models the early evolution of metabolism in a quite realistic setting and provides many tools for the detailed investigation of metabolic evolution. Using both simple example and a series of more complex simulation runs, the evolution of the components on the small scale (metabolites, enzymes) as well as on systems (pathways, networks) was investigated. The simulations allow to discriminate between different scenarios for the evolution of metabolic pathways. Based on the observations from this study, we argue that the different evolutionary hypotheses can be reconciled, in that they act in different phases of evolution, i.e. in different scenarios we might observe another strategy at work. Here, we suggest that forward evolution dominates in the earliest steps and is then superseded by a phase of enzyme recruitment, however, leaving behind a trace in form of a core set of forward evolved pathways.

To further test these hypotheses, we intend to simulate a number of different scenarios with changing parameters (mutation rate, duplication rate, "food" metabolite depletion), define other goals for the organisms (production of one specific metabolite, biomass or energy) and increase the complexity of the simulation runs (length and number of input molecules).

Albeit our simulation environment is still a drastic simplification of chemistry, it is realistic enough to investigate the evolution of early metabolism. Computer simulations like this one are likely to provide new insights about the general evolutionary mechanisms governing biological systems in particular in regimes that are not readily observable. Our approach of a realistic, yet computationally feasible, model appears to be a promising step in this direction.

Acknowledgments

This work has been funded by the Volkswagen Stiftung under grant I/82 719 and by the Vienna Science and Technology Fund (WWTF) MA07-30, and the COST-Action CM0703 "Systems Chemistry".

Additional Files

An animated movie of an example network evolution simulation, can be found here `http://www.bioinf.uni-leipzig.de/~alexander/animation.avi`.

References

Almonacid, D. E., Yera, E. R., Mitchell, J. B., and Babbitt, P. C. (2010). Quantitative comparison of catalytic mechanisms and overall reactions in convergently evolved enzymes: implications for classification of enzyme function. *PLoS Comput Biol*, 6(3):e1000700.

Benkö, G., Flamm, C., and Stadler, P. F. (2003). A graph-based toy model of chemistry. *J Chem Inf Comput Sci*, 43:1085–93.

Caetano-Anollés, G., Yafremava, L. S., Gee, H., Caetano-Anollés, D., Kim, H. S., and Mittenthal, J. E. (2009). The origin and evolution of modern metabolism. *Inter J Biochem & Cell Biol*, 41:285–297.

Copley, R. R. and Bork, P. (2000). Homology among $(\beta\alpha)_8$-barrels: implications for the evolution of metabolic pathways. *J Mol Biol*, 303:627–641.

Cordon, F. (1990). *Tratado evolucionista de biologa.* Aguilar Ediciones, Madrid, Spain.

Faulon, J.-L. and Sault, A. G. (2001). Stochastic generator of chemical structure. 3. Reaction network generation. *J Chem Inf Comput Sci*, 41:894–908.

Flamm, C., Ullrich, A., Ekker, H., Mann, M., Högerl, D., Rohrschneider, M., Sauer, S., Scheuermann, G., Klemm, K., Hofacker, I. L., and Stadler, P. F. (2010). Evolution of metabolic networks: A computational framework. *J. Syst. Chem.* in press.

Fujita, S. (1986). Description of organic reactions based on imaginary transition structures. 1. introduction of new concepts. *J. Chem. Inf. Comput. Sci.*, 26:205–212.

Gagneur, J. and Klamt, S. (2004). Computation of elementary modes: a unifying framework and the new binary approach. *BMC Bioinformatics*, 5.

Granick, S. (1957). Speculations on the origins and evolution of photosynthesis. *Ann NY Acad Sci*, 69:292–308.

Hendrickson, J. B. (1997). Comprehensive system for classification and nomenclature of organic reactions. *J Chem Inf Comput Sci*, 37:852–860.

Hendrickson, J. B. and Miller, T. M. (1990). Reaction indexing for reaction databases. *J. Chem. Inf. Comput. Sci.*, 30:403–408.

Herges, R. (1994). Coarctate transition states: The discovery of a reaction principle. *J Chem Inf Comput Sci*, 34:91–102.

Hofacker, I. L., Fontana, W., F, S. P., Bonhoeffer, S., Tacker, M., and Schuster, P. (1994). Fast folding and comparison of RNA secondary structures. *Mh. Chem.*, 125:167–188.

Horowitz, N. H. (1945). On the evolution of biochemical syntheses. *Proc Natl Acad Sci USA*, 31:153–157.

Jensen, R. A. (1976). Enzyme recruitment in evolution of new function. *Annu Rev Microbiol*, 30:409–425.

Morowitz, H. J. (1999). A theory of biochemical organization, metabolic pathways, and evolution. *Complexity*, 4:39–53.

Pfeiffer, T., Soyer, O. S., and Bonhoeffer, S. (2005). The evolution of connectivity in metabolic networks. *PLoS Biol*, 3:e228.

Rohrschneider, M., Ullrich, A., Kerren, A., Stadler, P. F., and Scheuermann, G. (2010 (submitted)). Visual network analysis of dynamic metabolic pathways. manuscript submitted for publication.

Schmidt, S., Sunyaev, S., Bork, P., and Dandekar, T. (2003). Metabolites: a helping hand for pathway evolution? *Trends Biochem. Sci.*, 28:336–341.

Ullrich, A. and Flamm, C. (2008). Functional evolution of ribozyme-catalyzed metabolisms in a graph-based toy-universe. In Istrail, S., editor, *Proceedings of the 6th International Conference on Computational Methodes in Systems Biology (CSMB)*, volume 5307 of *Lect. Notes Bioinf.*, pages 28–43.

Ullrich, A. and Flamm, C. (2009). A sequence-to-function map for ribozyme-catalyzed metabolisms. In *ECAL*, volume 5777/5778 of *Lect. Notes Comp. Sci.*

Ycas, M. (1974). On earlier states of the biochemical system. *J Theor Biol*, 44:145–160.

Dynamical Stability of Autocatalytic Sets

Rudolf M. Füchslin[1,3], Alessandro Filisetti[1], Roberto Serra[2,1],
Marco Villani[2,1], Davide DeLucrezia[1], and Irene Poli[4,1]

[1] European Centre for Living Technology
Calle del Clero 2940, 30124 Venice, Italy
[2] Dipartimento di Scienze Sociali, Cognitive e Quantitative Universita di Modena e Reggio Emilia,
via Allegri 9, 42100 Reggio Emilia, Italy
[3] Artificial Intelligence Lab Univ. Zürich
Andreasstr. 15, CH-8050 Zürich, Switzerland
[4] Dipartimento di Statistica, Universita Ca' Foscari,
San Giobbe - Cannaregio 873, 30121 Venezia, Italy

Abstract

Theoretical investigations of autocatalytic sets rendered the occurrence of self-sustaining sets of molecules to be a generic property of random reaction networks. This stands in some contrast to the experimental difficulty to actually find such systems. In this work, we argue that the usual approach, which is based on the study of static properties of reaction graphs has to be complemented with a dynamic perspective in order to avoid overestimation of the probability of getting autocatalytic sets. Especially under the, from the experimental point of view, important flow reactor conditions, it is not sufficient just to have a pathway generating a given type of molecules. The respective process has also to happen with a sufficient rate in order to compensate the outflow. Reaction rates are therefore of crucial importance. Furthermore, processes such as cleavage are on one hand advantageous for the system, because they enhance the molecular variability and therefore the potential for catalysis. On the other hand, cleavage may also act in an inhibiting manner by the destruction of vital components: therefore, an optimal balance between ligation and cleavage has to be found. If energy is included as a limiting resource, the concentration profiles of the components of autocatalytic sets are altered in a manner that renders a certain range for the energy supply rate as optimal for the realization of robust autocatalytic sets.

The results presented are based on a theoretical model and obtained by numerical integration of systems of ODE. This limits the number of involved molecular species which implies that the quantitative findings of this work may have no direct relevance for experimental situations, whereas the qualitative insights in the dynamics of the systems under consideration may generalize to systems of truly combinatorial size.

Keywords: Autocatalytic sets, autocatalytic metabolism, origin of life.

Introduction

In recent years, autocatalytic sets (ACS) Calvin (1956); Eigen (1971) have attracted interest from many different research directions. Probably most prominent are thereby investigations concerning the origin of life, but ACS proved to be a concept also of value e.g. for the study of transitions in general (non-chemical) systems of interacting production processes including the generation of knowledge, see Hanel et al. (2005).

Informally, the fundamental question with respect to chemical reaction networks is whether or not a given set of different, potentially catalytic molecules immersed into a suitable environment (most often some type of flow reactor) and provided with a sufficient supply of food or building blocks is able of maintaining the concentration of its members via mutual catalysis. The conditions under which such a self-maintaining or autocatalytic set can be expected to appear with sufficiently high probability are then those to be mimicked in an experiment e.g. concerned with the emergence of protolife.

Based on different models of catalytic networks, there is broad literature on the detection of ACS, see Letelier et al. (2006); Mossel and Steel (2005); Hordijk and Steel (2004). In Hordijk and Steel (2004) a polynomial-time algorithm for the detection of an important class of ACS has been presented. Hordijk and Steel applied this algorithm to a model by Kauffman (1986). By analyzing large numbers of randomly chosen networks, they corroborated a conclusion which Kauffman derived from combinatorial reasoning, namely that in sufficiently diverse populations of potentially catalytic chain molecules, an ACS will be present almost with certainty. Thereby, ACS will form independent of how sparse catalytic activity is distributed in the com-

binatorial variety of molecules, as long as this variety is big enough (usually limited by a maximal sequence length). Stated differently, given a certain variety of potentially catalytic molecules, there is always a threshold for the probability of catalytic activity such that above that threshold, ACS can be expected to emerge with high probability.

Despite some criticism (see Lifson (1997) and for a discussion of Lifson's arguments, see Steel (2000)) and the fact that more detailled models of catalysis may modify some results presented in Kauffman (1986), the main conclusions seem to generalize in one or the other form to a broad variety of models. The obvious question to ask then is, why ACS are not regularly discovered in the laboratory. In Filisetti et al. (2010), three possible answers were discussed. The first one (sometimes preferred by experimentalists) claims that the simplifications used in the formulation of the models on one hand make them tractable by analytical and/or computational means but on the other hand renders them unrealistic. The second answer (favored by some theorists) says that the basic statements derived from simplified models are also valid if the details of the physical and chemical world were considered, but that the threshold necessary for the emergence of ACS never has been reached. Finally, the third position (and also the one advocated in Filisetti et al. (2010) and in this work) highlights the fact that in investigations purely based on the properties of reaction graphs, dynamical and stochastic aspects are not considered. For some models, this is not necessary because their dynamics is basically (at least piecewise) determined by linear operators, e.g. Jain and Krishna (2001). But for most models (which are based on general reaction graphs), graph-theoretical methods may identify ACS which are only transient; this in the sense that the chemical dynamics eventually leads to a collapse of the ACS. This holds especially under flow reactor conditions, where e.g. a catalyst needs not only to be produced via some reaction path, but also at a sufficient rate in order to compensate for loss by outflow. Graph-theoretical means are able to identify whether or not a reaction path is present in a given network but not wether the dynamics establishes a non-trivial stationary ACS (In fact, one should speak of ACS exhibiting stationary or limit cycle behavior, but in practice one observes most models to yield almost exclusively stationary solutions. For a discussion, see e.g. Stadler et al. (1993)). In an experiment, however, it may be difficult to observe transient ACS, first because they may only be active during a very short period of time and second because their emergence may be highly susceptible to initial conditions. In contrast, stationary ACS which are able to produce a permanent deviation of some molecular concentrations from those one expects to result from the inflow and some non-catalytic background reactions offer a higher potential for being observable in a reproducible manner, as pointed out by Bagley and Farmer (1991). Whereas in Filisetti et al. (2010) the emphasis has been put on the investigation of the influence of stochastic fluctuations on the emergence and dynamics of ACS, this paper is concerned with the study of the influence of various parameters on the observability of stationary ACS.

The paper is organized as follows: In the second section, we discuss two different approaches for the definition of an ACS (or to be precise, the general and a more restrictive definition, the latter termed "autocatalytic metabolism") and motivate the choice being taken for the investigations in this work. In the third section, we briefly review the original model by Kauffman (1986) and present our implementation as a system of coupled ODEs. In the section reporting results, we show that the presence of a stationary ACS depends critically on the choice of parameters. We further study a derivative of the original model that takes energy considerations into account, means the different reactions compete for a, with a constant rate renewed, energy resource. We close with a discussion of the relevance of our results for experimental setups.

Autocatalytic Sets

We compare two different approaches for the analysis of autocatalytic sets. The first approach is especially appropriate for the study of reaction graphs and thoroughly discussed and formalized in Hordijk and Steel (2004). The second one, discussed in Bagley and Farmer (1991) takes into account the dynamics of the system but is less formal. Bagley and Farmer define an "autocatalytic metabolism" (ACM) as a coupled set of reactions which lead to permanent concentrations that are significantly departing from the values one would obtain without catalysis. As they point out, this definition is to some extent problematic, because what one regards as significant may depend on the experimental means. However, we will use a similar approach, because only those systems delivering a measurable deviation (both with respect to quantities as well as time) from some equilibrium distribution are of experimental interest. In order to highlight the difference between the two approaches, we briefly review the graph theoretical definition used by Hordijk and Steel and show that an ACS identified with their method needs not necessarily to be observable.

In Hordijk and Steel (2004) the main focus is laid on so called "reflexively autocatalytic and F-generated reaction systems (RAF)", whereby F denotes a set of "food"-molecules which are provided by the environment. For investigations concerned with the catalytic formation of chain molecules, F most often contains monomeric building blocks or a set of short oligomers. Informally, the concept of a RAF covers those sets of reaction systems R for which it holds that a) each reaction in R is catalyzed by a molecule being part of R and b) all reactants can be generated from a food set F by iterative applications of the reactions in R. In order to formalize the notion of a RAF in a rigorous manner, a number of definitions are required. We don't repeat

them here, but refer to the original work by Hordijk and Steel (2004)).

A RAF can be regarded as, once present, a potentially self-sustaining reaction system that in principle produces all the catalysts and intermediates it needs for its reactions. It is only potentially self-sustaining, because necessary molecules need not only to be produced but being produced with sufficient rates. Note further that the definition of a RAF does not require the system to emerge, given the molecules in F are supplied (In fact, the elements of F need not to be catalysts at all).

As shown in Hordijk and Steel (2004), there exists a polynomial-time algorithm for the detection of RAFs, given a system of catalytic reactions. That such a RAF is only potentially self-sustaining is demonstrated by a (completely artificial) reaction system given as follows (with respective catalyst and reaction rate above the arrows):

$$\begin{aligned} a + e &\xrightarrow{d,k} c \\ b + e &\xrightarrow{c,k} d \\ c &\xrightarrow{d,k} e + e \end{aligned} \tag{1}$$

With $F = \{a, b\}$, this system qualifies as a RAF (possibly being part of some bigger catalytic reaction system). It is possible (not shown here) to add further reactions representing the renewal of resources and outflow, the former taking place with unit rate, the latter with rate k_d. Setting $k = 1$ and $a(0) = b(0) = c(0) = d(0) = e(0) = 1$, the behavior of the system then depends critically on the size of k_d. As illustrated in Fig. 1, the system attains a stationary state for $k_d = 0.1$ and collapses for $k_d = 0.5$. This observation is of importance insofar that it shows that one tends to overestimate the probability for the observation of experimentally relevant ACM if one relies on static, graph theoretical methods yielding probabilities for the occurrence of ACS. Consequently, in what follows we employ dynamic reaction kinetics in order to decide whether a reaction system contains as a subsystem an ACM in the sense of Bagley and Farmer (1991).

The Model

A fundamental model for the study of the emergence of ACS has been proposed in Kauffman (1986); we will briefly review this approach and its main conclusions and present our own implementation which is used for the construction of a set of ODEs. These ODEs are solved numerically for various parameter settings in order to identify the relative importance of different reaction mechanisms. Thereby, we are interested in parameter combinations that exhibit non-trivial optima for the probability of the existence of an ACM, especially if these parameters offers the potential of being controllable in an experimental setting.

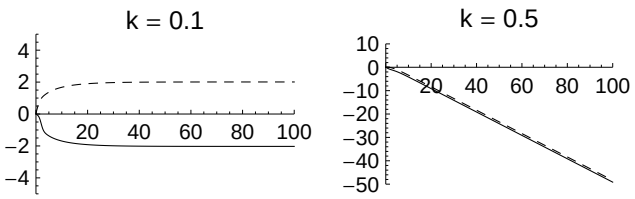


Figure 1: Time evolution of the system given by eqs. 1 for two different values of the outflow rate parameter k_d. Shown are the logarithms of the concentration of $c(t)$ (continuous line) and $d(t)$ (dashed line) as a function of time.

The Basic Model

In Kauffman (1986), the properties of sets of potentially catalytic di-block copolymers were investigated. Thereby, it was assumed

- Polymers consist of two different types of monomers A and B.
- There are two types of catalyzed reactions, namely ligation and cleavage.
- The probability for a polymer P_c to catalyze a ligation $P_1 + P_2 \xrightarrow{P_c} P_1P_2$ or a cleavage $P_1P_2 \xrightarrow{P_c} P_1 + P_2$ is given by a probability r.
- The number p_i represents the density of the polymer P_i.

This setting, basically a random reaction system, doesn't make any specific "helpful" assumptions supporting the emergence or existence of an ACM, and nevertheless, strong evidence was given that such a system should eventually contain an ACM, given only a sufficiently large variety of different polymers being included in the system (In case of block polymers, this can be achieved simply by allowing sequences of length up to a critical L_c).

Several implementations of random graph models using ODEs have been studied, see e.g. Farmer et al. (1986); Bagley and Farmer (1991). In this work, the dynamics of

the system is given by:

$$\begin{aligned}\frac{dp_i}{dt} &= k_{i,in} - k_{out}p_i \\ &+ \sum_{j,k,m} k_{j,k,L}L(j,k,i,m)p_jp_kp_m \\ &- \sum_{j,k,m} k_{i,j,L}L(i,j,k,m)p_ip_jp_m \\ &- \sum_{j,k,m} k_{j,i,L}L(j,i,k,m)p_jp_ip_m \\ &+ k_C \sum_{j,k,m} C(i,j,k,m)p_kp_m \\ &+ k_C \sum_{j,k,m} C(j,i,k,m)p_kp_m \\ &- k_C \sum_{j,k,m} C(j,k,i,m)p_ip_m.\end{aligned} \tag{2}$$

Thereby, p_i represents the density of a polymer with sequence P_i composed of two types of monomers A, B. The rate of influx $k_{i,in}$ is set to one for the monomers A, B and zero for all other sequences. Outflow is determined by the rate k_{out}, and the kinetic rates of ligation and cleavage are denoted by $k_{i,j,L}$ and k_C respectively. The arrays L and C represent the random graphs, chosen at the beginning of each run: This means that L, C are arrays representing fixed random reaction networks, which, once set, remain constant. Using the symbol $\oplus$ for sequence concatenation, it holds:

$$L(i,j,k,m) = \begin{cases} 0 & P_i \oplus P_j \neq P_k \\ 1 \text{with prob.} \quad r_L & P_i \oplus P_j = P_k \end{cases} \tag{3}$$

and

$$C(i,j,k,m) = \begin{cases} 0 & P_i \oplus P_j \neq P_k \\ 1 \text{with prob.} \quad r_C & P_i \oplus P_j = P_k \end{cases} \tag{4}$$

The index m represents the dependence on the catalyst P_m.

In all calculations subsequently shown, several additional assumptions have been made:

1. The monomers A, B must not act as catalysts; this in order to enhance chemical plausibility.

2. There is a maximal sequence length L. Ligations may well produce longer sequences, but those are assumed to fall out by precipitation. This is physically plausible and keeps the system tractable.

3. In order to capture steric effects, the ligation rate $k_{i,j,L}$ is length dependent. Shall $|P_i|$ denote the length of P_i, we set $k_{i,j,L} = k_L/(|P_i||P_j|)$ for some constant k_L. The idea behind this (crude) approximation is that in a well-stirred reactor, the collision frequency of two sequences is assumed to be independent of the length. The collision happens by the contact of two monomers, one out of each sequence. The chance that those are the ones that are able of mutual ligation because they mark the end and the start of the respective sequences is inversely proportional to the respective length of the sequences.

The system then contains $2^{L+1} - 2$ variables. This means, taking into account the non-catalycity of the monomers, that there are $(2^{L+1} - 2)^2(2^{L+1} - 4)$ potential ligation reactions and $(2^{L+1} - 4)\sum_{l=2}^{L} 2^l(l-1)$ possible cleavage processes. As it turned out, already values of $L = 6$ deliver systems of sufficient combinatorial variety in order to exhibit interesting dynamical effects. In all simulations, we set $\forall i : p_i(0) = 1$ as initial condition; this with the idea to give a potential ACM in a random graph sufficiently favorable starting conditions. Following Bagley and Farmer (1991), a random reaction graph qualifies as containing an ACM, if the concentration of at least one non-monomeric species is above a threshold T after a time interval longer than $10t_d$ with $t_d = -\log(T)/k_{out}$ denoting the typical decay time for T. As will be shown (and has already been discussed by Bagley and Farmer), the decision whether a reaction system contains an ACM is surprisingly insensitive to the choice of T. The numerical solutions were obtained by internal routines of the software package Mathematical^{TM} and a sample of solutions was verified with a standard adaptive fourth-order Runge-Kutta solver.

The Model with Explicit Consideration of Energy

Most of the investigations dealing with ACM don't take into account energy considerations, or more generally, the explicit competition for some limited resource other than the supplied monomers. As will be discussed in the result section, such an external limitation need not to be disadvantageous for the system, but may even help to stabilize it. We consider energy in a relatively simple manner. The ligation and cleavage terms in eqs. 2 are multiplied with the concentration $e(t)$ of some energy resource. Thereby, the energy resource is used up and permanently renewed by inflow with a rate k_E. The dynamics of the additional variable $e(t)$ is given by:

$$\begin{aligned}\frac{de}{dt} &= k_E - k_{out}e \\ &- \sum_{i,j,k,m} k_{i,j,L}L(p_i,p_j,p_k,p_m)p_ip_jp_me \\ &- k_C \sum_{i,j,k,m} C(p_i,p_j,p_k,p_m)p_kp_me.\end{aligned} \tag{5}$$

Results

In this section, we study the dependence of the dynamics of the models presented in the preceding section. Some of the parameters remain fixed for all simulations: $k_{out} = 0.02, k_L = k_C = 1$. Furthermore, each data point representing an average value has been computed using at least 20,

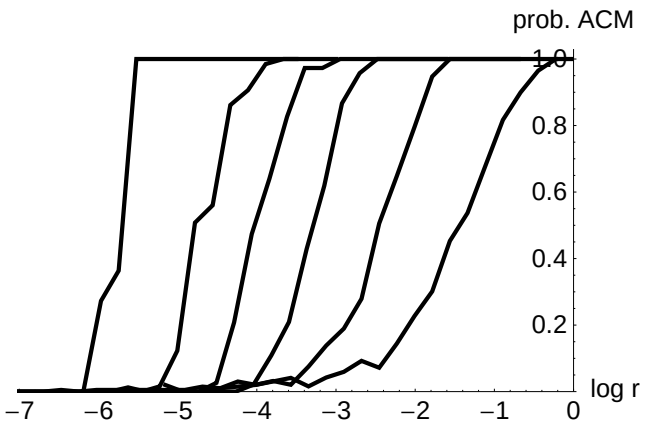


Figure 2: Probability for observing an ACM in a random reaction graph as a function of the catalytic reaction probability $r_L = r_C = r$ for different values of the maximal sequence length $L = 2, 3, 4, 5, 6, 8$. Starting from $L = 8$, graphs representing decreasing length exhibit increasing values for the transition value of r.

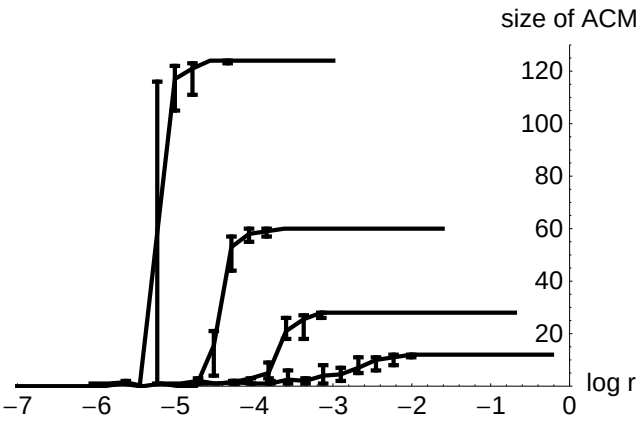


Figure 3: Average size of ACM (number of non-monomeric components bigger than $T = 10^{-6}$ after $t = 10^5$) as a function of r and for sequence length $L = 3, 4, 5, 6$ (bottom to top). Shown are the median values for the size of the ACM, the whiskers denoting the first and third quartile. Above the transition value of r, the system tends to be maximally diverse (A maximal sequence length L implies $2^(L + 1) - 4$ non-monomeric sequences).

but most often more than 50 samples. As a convention, logarithms are always taken to the base e. Whiskers, if shown, denote first and third quartiles.

The Fundamental Transition

As postulated in Kauffman (1986), for sufficiently large values of the probabilities for catalytic reactions r_L and r_C given in eqs. 3 and 4, the reaction graph should contain an ACM with high probability. In fig. 2, this transition is clearly observable and becomes sharper for longer sequences. Interestingly, the transition curves, giving the probability of observing at least one non-monomeric sequence with a concentration above the threshold value T look identically the same for T in the range from 10^{-12} to 10^{-2}, which means that if there is an ACM, at least one of its components will be present with a significant concentration. Fig. 3 shows the average size of the ACM, means the average number of components with concentration values above a threshold $T = 10^{-6}$ after an integration time $t = 10^5$ for sequences of maximal length $L = 3, 4, 5, 6$. We observe that above the transition value of r, the system becomes maximally diverse. This may be of relevance in an evolutionary context.

The Role of Cleavage

Given a certain fixed probability for ligation r_L, one may ask for the corresponding optimal value of r_C. It is clear that cleavage has some beneficial aspects for the appearance of an ACM, because cleavage tends to enlarge the variety of sequences. However, cleavage may as well destroy vital parts of an ACM. This is relevant especially under flow reactor conditions, where the generation of a specific sequence needs to be sufficiently powerful in order to compensate the outflow. And in fact, in fig. 4, a clear optimum for r_C can be observed, given a fixed $r_L = 0.01$ and $L = 6$. Notably, in our simulation, this optimum perfectly justifies the original choice of $r_L = r_C$ by Kauffman. The choice of r_L in the transition region is motivated by first taking into account that a system may be based only on ligation but not solely on cleavage (at least with monomeric input). A small value for r_L will most probably not yield an ACM. A large value is also not of big interest: A system with lots of ligation reactions already produces most sequences and does not profit from a further broadening of the sequence variety by cleavage. The transition region in fig. 2 is the domain in which an optimization of r_C will take the most effect.

Again, it is emphasized that the curve shown does not depend on the detection threshold T, though the average number of concentrations above the threshold does, see figs. 5 and 6. Note that whereas the curve in fig. 4 refers to the whole sample and shows the ratio of those reaction systems containing an ACM, the data in figs. 5 and6 give the average size of the ACM, provided there is one. Consequently, data points at the lower and higher end of the scale are of less statistical weight (and relevance) than those in the middle.

The Role of Energy

Controlling the influx of energy (or, to be chemically more accurate, the influx of molecular energy carriers) is a parameter easy to control in an experiment, therefore its influence is of interest. It is clear that below a certain threshold of the influx rate k_E the generation of non-monomeric

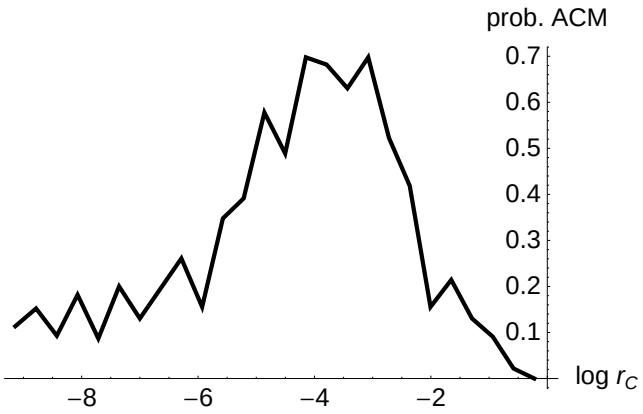


Figure 4: Probability for observing an ACM in a reaction graph with maximal sequence length $L = 6$ and $r_L = 0.01$ as a function of r_C. The detection threshold is set to $T = 10^{-6}$ (continuous line) and $T = 10^{-2}$ (dashed line).

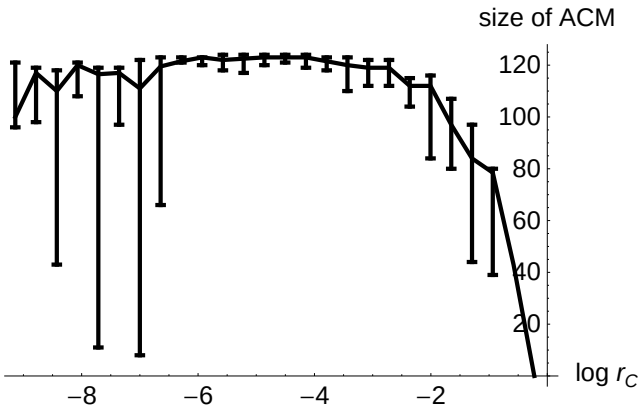


Figure 5: Average size of ACM for $L = 6$ and $r_L = 0.01$ as a function of r_C. The detection threshold is given by $T = 10^{-6}$. Shown are the median values for the size of the ACM and the whiskers denote the first and third quartile.

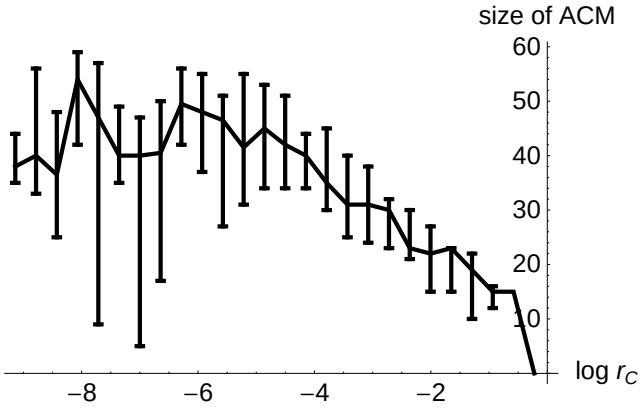


Figure 6: Same as fig. 5, but with $T = 10^{-2}$.

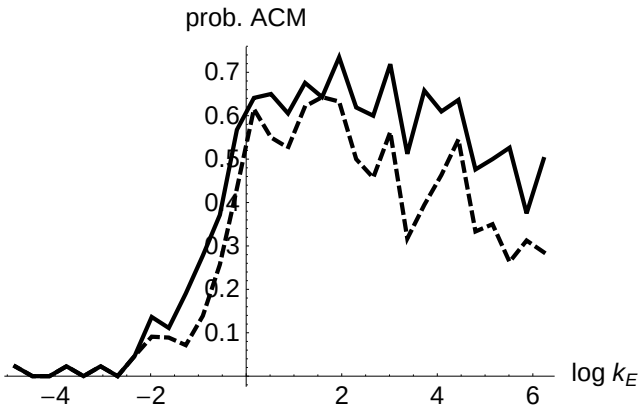


Figure 7: Probability for observing an ACM in a random reaction system with $L = 6, r_L = r_C = 0.01$ as a function of the rate of energy influx k_E.

sequences is not anymore powerful enough to compensate for the outflux. This can be seen in fig. 7. Given suitable system parameters, ACM are easy to observe at higher values of k_E. Interestingly, the average size of the ACM for a large threshold T shows a maximum for intermediate values of k_E, see fig. 8 (giving the average number of concentrations above $T = 10^{-6}$) and more prominently for $T = 10^{-2}$ in fig. 9. A possible explanation for this phenomenon is that the plenty abundance of energy alllows the generation of more or less all possible sequences, as suggested by the results shown in fig. 3. A more fierce competition for energy, however, may lead to the eventual extinction of some side branches of an ACM and consequently a boost of its "core" components. This externally controlled focussing is of relevance, because in more realistic scenarios with larger sequence lengths, the relative concentrations of core components may be much lower than in the (numerically tractable) model systems presented in this work. Consequently, stochastic fluctuations play a more important role and a mechanism strengthening the "backbone" of an ACM at the expense of some side reactions increases the robustness of the system which is of evolutionary and experimental importance (the consideration made here applies also to the scenario discussed in fig. 6). Studying stochastic effects in ACM with longer sequences requires, however, a particle based approach. For a detailed discussion, see Filisetti et al. (2010).

Summary and Discussion

We have shown the importance of the dynamics of a reaction system for answering the question whether it contains an autocatalytic metabolism. Many algorithms are based on the analysis of combinatorial properties of random graphs. Thereby, they are not considering that, especially in the situation of a flow reactor, there must not only be a pathway

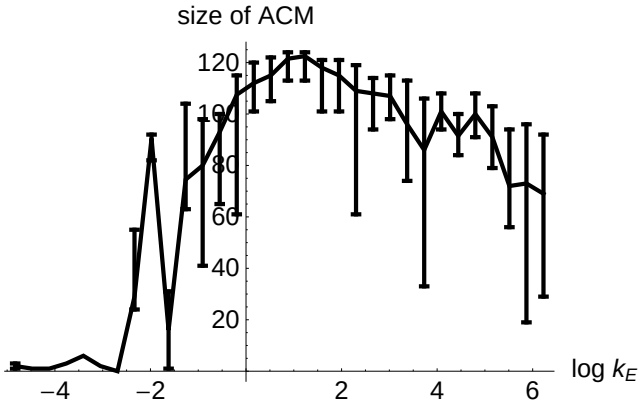


Figure 8: Average size of the observed ACM in a random reaction system with $L = 6, r_L = r_C = 0.01$ as a function of the rate of energy influx k_E and for a detection threshold $T = 10^{-6}$.

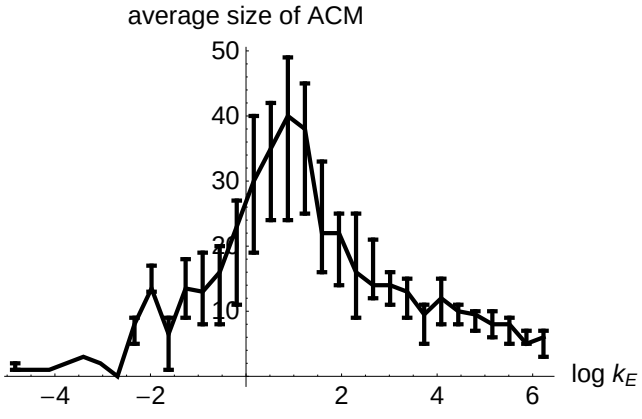


Figure 9: Average size of the observed ACM in a random reaction system with $L = 6, r_L = r_C = 0.01$ as a function of the rate of energy influx k_E and for a detection threshold $T = 10^{-2}$.

for the production of a given molecule but its production has in addition to happen at a rate that compensates for the loss by outflow. Studying the kinetic behavior of random reaction systems reveals the importance of a proper balancing of the probabilities for different types of reactions: We investigated cleavage and found that taking into account dynamics, cleavage does not only enlarge the variety of polymer species (which is desirable from the perspective of obtaining an ACM) but may also destroy components relevant for the system with a rate that cannot be compensated by their respective generation processes. We also investigated the role of energy consumption and found that the introduction of energy as a limiting factor strongly influences the concentration profile of the ACM. It turned out that whereas a large supply of energy leads to a broad variability of sequences, intermediate values seem to favor ACM with less, but, with respect to concentration also in absolute terms, more pronounced components. This means that such intermeidate values render ACM that are less susceptible to fluctuations, which is of relevance in the context of evlutionary processes.

We investigated systems with rather short sequences, mostly with a maximal sequence length of $L = 6$. The numerical values for the catalytic probabilities r_L and r_C need then to be of a size which is chemically not realistic. We claim that our results are of worth because whereas the quantitative features of the shown results heavily depend on L, the qualitative ones don't. Even more, data (partially not shown) suggests that the discussed effects become more pronounced with increasing L. According investigations need then to be performed in a particle based manner, see Filisetti et al. (2010). Another interesting perspective is presently investigated by DeLucrezia and coworkers. In their approach, the "monomers" are replaced by pre-prepared strands consisting of some ten amino acids. A sequence consisting of a combinatorial assembly of these strands may have a higher probability of exhibiting catalytic properties. However, the model presented in this paper is then only a "coarse-grained" approximation to the dynamics, because cleavage may well happen within one of the original monomeric strands.

Our choice of the initial conditions, namely to set the concentrations of all sequences to one at the start is certainly unrealistic and motivated by our focus on stability considerations. The discovery that the energy supply influences the concentration profile opens the perspective of "iterative" emergence. A very limited set of initially provided components may establish a first, still frail ACM which produces as side products some further, possibly catalytic components at low concentrations. A only temporal increase of the energy supply may enable the system to reach a new basin of attraction by a short-term increase of cleaving activity which in turn produce a passing wider variety of sequences at sufficient concentration in order to take effect, but without having to cope with the long-term presence of enhanced cleavage. We will address this scenario in a subsequent work fo-

cussed on issues of emergence, also considering aspects of stabilization against molecular parasites achieved by spatial organization with Filisetti et al. (2008) or without Füchslin and McCaskill (2001); Füchslin et al. (2004) explicit compartmentalization.

The problem of deciding whether or not a given reaction system contains an ACM may one remind to a similar problem in systems biology, namely the determination of possible fluxes in a only partially known metabolic networksVarma and Palsson (1994); Orth et al. (2010). In flux balance analysis, one basically determines the set of potential solutions for the fluxes, given that a) the stoichiometric matrix and a vector containing fluxes forms an underdetermined linear system and b) some (in practice usually linear) constraints have to be observed. Flux balance analysis provides a highly successful and efficient tool for e.g. the optimization of only partially known networks (By using linear programming). The problem we address in this work is, however, different. The networks are completely known and therefore, the flux balance equation are fully determined, which means that searching a stationary solution requires solving a non-linear system.

Taking into account dynamics shows that first, one of the reasons for the fact that spontaneously formed autocatalytic systems have not or only rarely been observed in the laboratory may not only be due to lack of catalytic activity. As a matter of fact, it could even be caused by too much catalysis, if cleavage is too frequent. Second, and probably more important, we need to shift our attention from focussing solely on catalysis (and respective probabilities) to a picture in which kinetics plays an important role too. Even if we had reaction system in which in principle an ACM could produce measurable signals, it only does if the kinetic parameters are suitably chosen. Some of these parameters, such as e.g. outflux rates, can easily be manipulated in an experiment and should be in the focus of future work.

Acknowledgments

This work has been supported by the Fondazione di Venezia, http://www.fondazionedivenezia.it, (DICE project).

References

Bagley, R. and Farmer, J. (1991). Spontaneous emergence of a metabolism. *Artifcial Life II*, pages 93–140. Santa Fe Institute Studies in the Sciences of Complexity X.

Calvin, M. (1956). Chemical evolution and the origin of life. *Amer. Sci.*, 44:248–263.

Eigen, M. (1971). Self-organization of matter and the evolution of biological macromolecules. *Naturwissenschaften*, 58:465–523.

Farmer, J. D., Kauffman, S. A., and Packard, N. H. (1986). Autocatalytic replication of polymers. *Physica D*, 22(2):50–67.

Filisetti, A., Serra, R., Carletti, T., Villani, M., and Poli, I. (2008). Synchronization phenomena in protocell models. *Biophysical Reviews and Letters*, 3(1/2):325–342.

Filisetti, A., Serra, R., Villani, M., Füchslin, R. M., Packard, N., Kauffman, S. A., and Poli, I. (2010). A stochastic model of autocatalytic reaction networks. submitted to ECCS10.

Füchslin, R. M., Altmeyer, S., and McCaskill, J. (2004). Evolutionary stabilization of generous replicases by complex formation. *Europ. Phys J. B*, 38(1):103–110.

Füchslin, R. M. and McCaskill, J. S. (2001). Evolutionary self-organization of cell-free genetic coding. *Proceedings of the National Academy of Science USA*, 98(16):9185–9190.

Hanel, R., Kauffman, S. A., and Thurner, S. (2005). Phase transition in random catalytic networks. *Physical Review E*, 72:036117.

Hordijk, W. and Steel, M. (2004). Detecting autocatalytic, self-sustaining sets in chemical reaction systems. *Journal of Theoretical Biology*, 227:451–461.

Jain, S. and Krishna, S. (2001). A model for the emergence of cooperation, interdependence, and structure in evolving networks. *Proceedings of the National Academy of Science USA*, 98(2):543–547.

Kauffman, S. (1986). Autocatalytic sets of proteins. *Journal of Theoretical Biology*, 119:1–24.

Letelier, J.-C., Soto-Andrade, J., Abarzua, F. G., Cornish-Bowden, A., and Cardenas, M. L. (2006). Organizational invariance and metabolic closure: Analysis in terms of (m;r) systems. *Journal of Theoretical Biology*, 238:949–961.

Lifson, S. (1997). On the crucial stages in the origin of animate matter. *J. Molecular Evolution*, 44:1–8.

Mossel, E. and Steel, M. (2005). Random biochemical networks: the probability of self-sustaining autocatalysis. *Journal of Theoretical Biology*, 233:327–336.

Orth, J. D., Thiele, I., and Palsson, B. O. (2010). What is flux balance analysis? *Nature Biotechnology*, 28:245–248.

Stadler, P. F., Fontana, W., and Miller, J. H. (1993). *Physica D*, 63:378.

Steel, M. (2000). The emergence of a self-catalysing structure in abstract origin-of-life models. *Appl. Math. Lett.*, 3:91–95.

Varma, A. and Palsson, B. O. (1994). Flux balancing: Basic concepts, scientific and practical use. *Biotechnology*, 12:994–998.

The EvoGrid: A Framework for Distributed Artificial Chemistry Cameo Simulations Supporting Computational Origins of Life Endeavors

Bruce Damer[1], Peter Newman[1], Richard Gordon[2], Tom Barbalet[3], David W. Deamer[4] and Ryan Norkus[1]

[1]DigitalSpace Corporation
[2]University of Manitoba
[3]Noble Ape and Biota.org, The Artificial Life Project
[4]University of California at Santa Cruz
bdamer@digitalspace.com

Abstract

The Evolution Grid, or EvoGrid is a computer simulation framework for distributed artificial chemistry (AC) supporting computational origins of life (COoL) research. The EvoGrid consists of a number of small experiments running on short time scales pruned by aggressive tree-branching searches supported by random parametric re-seeding and temporal back-tracking. The EvoGrid is designed to converge upon the observation of "cameo" simulations of key pre-biotic or simple biological structures or behaviors. These cameo simulations can then inform and feed larger AC simulations operating over biologically relevant time scales. In addition, the framework is designed to plug into a heterogeneous set of engines ranging from high fidelity molecular dynamics (MD) to more abstract AC techniques on the same set of data. The EvoGrid also provides shared web-based simulation management services and uniform, open standards for execution, storage and data analysis. We conclude by describing the first prototype implementation of the EvoGrid, early results, next steps and open questions in this and other COoL endeavors.

Introduction

In their seminal paper Open Problems in Artificial Life (Bedau et al., 2000) the authors set a challenge in the second open problem to "achieve the transition to life in an artificial chemistry *in silico*" (p. 364) while also identifying that "[b]etter algorithms and understanding may well accelerate progress… [and] combinations of… simulations… would be more powerful than any single simulation approach" (p. 367-68). The authors also point out that while the digital medium is very different from molecular biology, it "has considerable scope to vary the type of 'physics' underlying the evolutionary process" and that this would permit us to "unlock the full potential of evolution in digital media" (p. 369).

All of this potential awaits further progress in the computational challenges of high fidelity (i.e. accurate and predictive) artificial chemistries. Current state-of-the-art artificial chemistries (AC) (Dittrich, et al., 2001) including molecular dynamics (MD) projects utilize large centralized general-purpose computer clusters or, more recently, purpose built hardware, such as Anton, an MD supercomputer (Shaw, et al., 2009). Simulating tens of thousands of atoms for days to weeks on a commodity cluster will produce a number of nanoseconds of real-time equivalent chemistry. Optimized software running on Anton promises milliseconds of real-time equivalent ACs in weeks of computation (Shaw, et al., 2008).

To meet these challenges, proposals to unify efforts into larger computational origins of life (COoL) endeavors have been brought forth. Shenhav and Lancet (2004) propose utilizing the Graded Autocatalysis Replication Domain (GARD) statistical chemistry framework (Segre and Lancet, 1999, 2000). These authors have developed a hybrid scheme merging MD with stochastic chemistry. In GARD many short MD computations would be conducted to compute rate parameters or constraints for subsequent stochastic simulations. Thus, a federation of simulations and services was conceived which would also involve interplay with *in vitro* experiments. It is this vision for unifying efforts in COoL that has inspired our own work to build a framework for distributing and searching a large number of small chemistry simulation experiments.

As stated by Shenhav and Lancet, "the prebiotic milieu could best be characterized by a dense network of weak interactions among relatively small molecules" (p. 182). Simulating such a soup represents yet another scale of complexity beyond the targets set by even the builders of Anton. While the simulating of the full pathway to life *in silico* seems like a journey of a thousand miles, the first few steps can be taken and may become less daunting when helped along by some innovative algorithmic and architectural short cuts.

A fundamental property of large scale (in time duration and population of objects) simulations is that for the most part they use a homogeneous approach to optimize computation. On the opposite end of the spectrum we propose to run a large number of small simulations. Such an approach would in theory support a heterogeneous network of simulation techniques which vary physics, levels of abstraction and could even employ selection methods and replication of results inspired by the process of evolution. This is the approach

taken by the authors in developing the Evolution Grid (or EvoGrid), to be discussed next.

EvoGrid Search Function

The basic concept behind the EvoGrid is what we are terming *cameo simulations*. Cameo simulations are comprised of no more than a few hundred or thousand particles representing atoms and small molecules running over short time scales and in multiple instances. The existence of those instances is governed by a search tree function which permits variations of initial conditions and the branching of multiple, parallel simulations. Variation of parameters and branching are under control of an analysis step which looks for interesting structures or behaviors within each cameo simulation *frame*. Frames deemed less interesting may be terminated so as to permit other branches to be explored to a greater extent. This approach is inspired by the class of genetic algorithms (GA) combined with hill climbing algorithms widely used in Artificial Intelligence (Russell and Norvig, 2003). It is a form of importance sampling (Kalos and Whitlock, 2008), and its relationship to Maxwell's Demon requires careful scrutiny (Maruyama et al., 2009).

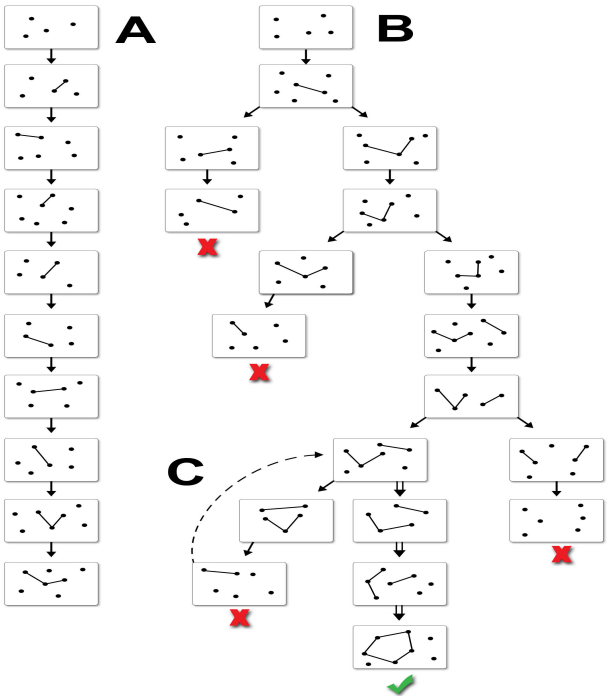


Figure 1: Illustration of the hill climbing search tree method employed by the EvoGrid

Figure 1 illustrates this method for a Control (A) which depicts a typical linear time sequence simulation and Test (B) which depicts the arising of simulation branches in this case due to selection for the phenomenon of more densely interconnected points. This illustration depicts another optimization called temporal back-tracking. If the simulation states of each frame can be stored through time, then a failed branch may be rolled back to the point at which "interesting" frames were still occurring. With a random seed applied, a new branch is started. This branch may yield a complex phenomenon forgone in the failed branch. In the example illustrated abstractly by C, that phenomenon might be a ring structure, as shown in the frame with the check mark. In this way, improbable occurrences may be guided across valleys of highly probable failure.

Genes of Emergence

Efforts to bridge nonliving and living matter and develop protocells from scratch (Rasmussen et al., 2003) will rely on bottom-up self assembly with commensurate self organization of classes of molecules. The development of repeatable self assembly experiments *in silico* (Rajagopalan, 2001) could serve as an important aid to *in vitro* protocell research. Self assembly in simulation may be purposefully designed into the experiment or may be an emergent phenomenon discovered by a directed search through multiple trial simulations. The initial conditions for a simulation could be equated to the coding sequences of a genetic algorithm (GA), and the simulation outputs seen as its expressed phenotype. The EvoGrid's search for self-assembly and other phenomena in cameo simulations is therefore a search for what we might term "genes of emergence" (GoE).

GoEs may be derived from within many different types of simulation, not just in the computationally intensive MD world. More abstract simulation modalities may yield shorter pathways to the production of important emergent phenomena than through computationally complex ACs (Barbalet et al., 2009). One could then see that the EvoGrid represents a "discovery system" operating on a continuum of techniques which might include: the execution of simulation modules that code for abstract universes yielding interesting results, to be then swapped out for a simple AC within which we would hope to reproduce the results, and finally, carrying the GoEs one step further into high fidelity MD, then which could inform validation through full scale *in vitro* experimentation.

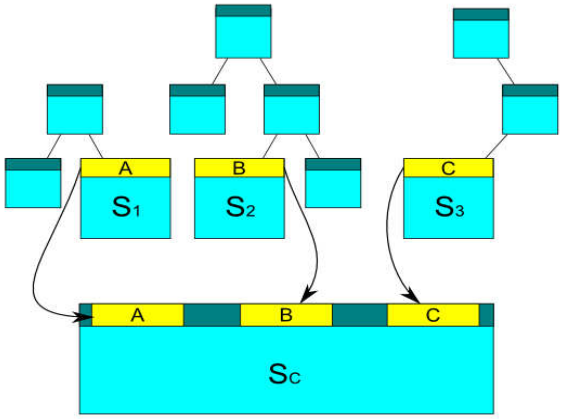


Figure 2: Illustration of the concept of cameo simulations feeding a larger composite simulation.

Figure 2 graphically illustrates the first two stages of this continuum. In the first stage, hill-climbing search functions (represented here as trees) process through a number of small cameo AC simulations. The end-point simulations, shown here as S1, S2 and S3, each meet some criteria for generating a structure or behavior of relevance to a larger composite simulation Sc. In the second stage, Sc is constructed from a mixture of content from each of the "feeder" cameo simulations and is driven by an amalgamation of the individual simulation experimental parameters A, B and C. The hope is that this amalgamation in simulation Sc, running with a much larger content store and over biologically significant time scales, would generate a rich mixture of phenomena, such as the formation of membranes, emergence of replicators, or the observation of autocatalytic reaction pathways. It is this enriched simulation environment which could be the basis for more ambitious computational origin of life endeavors. In another twist, an interesting phenomenon observed in Sc could be captured, its parameters and local contents extracted and cameo simulations run to characterize and fine tune the phenomenon more closely, enabling another ratchet in the emergent power of the larger simulation.

EvoGrid Design and Operation

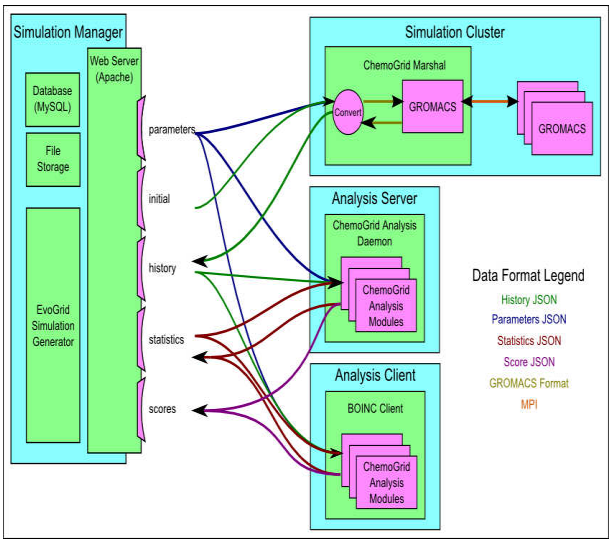


Figure 3: High level design and data flow of the EvoGrid

As depicted in Figure 3, the modular design of the EvoGrid encapsulates an MD simulation engine, in this case GROMACS (Van der Spoel, 2005), which we found to have good performance and was suitable to run as a plug-in component. GROMACS could be swapped out for other suitable simulation systems or the EvoGrid would support these systems running in parallel on the same data set. This architecture is designed to meet the challenge posed by Bedau et al. (2000) in which combinations of different simulation approaches might be a pathway to significant progress.

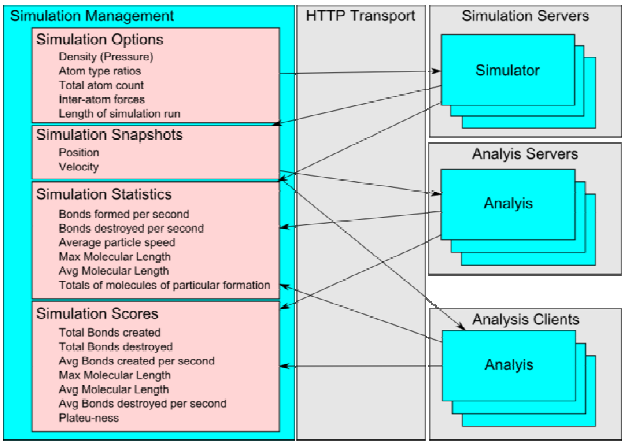


Figure 4: Lower level sequencing of data types through the EvoGrid

Other abstracted components depicted include an Analysis Server and an Analysis Client. Both of these components process inputs and outputs to the Simulation Cluster using the compact JSON format. The Simulation Manager running via HTTP/Web services sequences the simulation of and the analysis of individual frames (Figure 4). MD simulations typically have heavy compute loads in executing the time-steps for each force interaction of artificial atoms. In the EvoGrid, tens of thousands of frames are being executed and replicated through new branches. This generates terabytes of stored states for analysis. This could eventually call for a fully distributed simulation network, such as provided by the BOINC network (Anderson, 2004). BOINC supports many computationally intensive scientific applications, such as Folding@home (Pande et al., 2003). However, at this time we are relying on the centralized analysis server.

EvoGrid Prototype Runs and Results

A prototype of the EvoGrid architecture was built in 2009. Frames of 1,000 simulated atoms were run for 1,000 time steps within the GROMACS module with a uniform heat bath applied.

Initial conditions for GROMACS were:

- Density in particles per Angstrom: 0.01 - 0.1
- Temperature in Kelvin: 200 – 300, used for initial velocity and temperature bath
- Bond outer threshold in Angstrom: 0.1 - 1.0, distance, used for bond creation

The atoms ranged between three and ten randomly generated types. All their parameters (mass, charge, force interaction with other types, radius and volume) were selected from a uniformly distributed random range.

Forces between atom types included:
Pre-computed components of the Lennard-Jones force function:

- c6 0.0 - 0.1
- c12 0.0 - 0.00001

Covalently bonded (pre-computed components of the harmonic bond force function):

- rA 0.0 - 2.0
- krA 0.0 - 2.0
- rB 0.0 - 2.0
- krB 0.0 - 2.0

As an initial test case on a single instance of GROMACS when a bond was created, the Lennard-Jones forces would cease applying, and no new forces were applied. This was done to minimize real world constraints prior to having access to a computer cluster supporting covalent bond computations. The main focus of this prototype was to be able to test the architecture, not faithfully simulate the chemistry.

The position and velocity data was dumped every 1000 cycles and a naïve bonding applied to all atoms or atom-molecule or molecule-molecule objects. After a thousand of these dumps, this collected history was processed by the analysis server. Table 1 represents the scoring for frame number 144,204, the final frame in our trial run. The analysis was set up to look for the formation of "larger" virtual molecules, which in our simplistic interpretation meant a simple count of the greatest number of bonds between any two atoms. Employing Monte Carlo methodologies, the maximum search score reached in the trial was a simple sum of the entries in Table 1.

Measured values	**Final simulation scores**
Average molecular size	2.2303
Maximum average molecular size	4.47307
Average maximum molecular size	9.355
Maximum individual molecular size	17
Final maximum search score	33.0584

Table 1: Scoring produced by prototype analysis server for final simulation frame

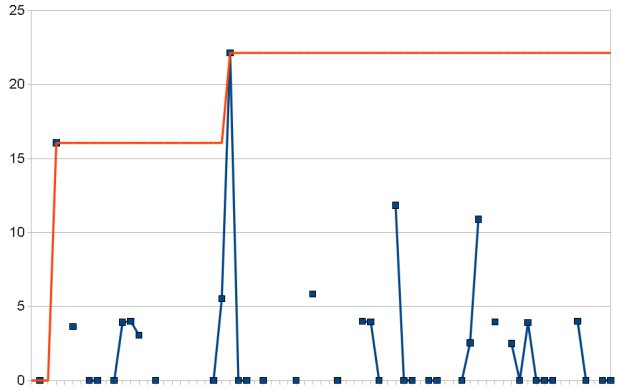


Figure 5: Scoring of experiments in "control" mode (random regeneration with no search tree function)

Figure 5 shows the "control" case (A) from figure 1 in which a random initial frame is simply run with a randomly seeded restarting of GROMACS for a duration of one thousand internal simulation steps (atom-atom interactions) with a thousand state dumps without the search function applied. As we can see, while there were some highly scored frames (red line), there is no maintained trend. Please note that the missing lines indicate cases where our software generated impossible simulation configurations and the execution was halted. This illustrated an area for improvement of how we were operating the GROMACS engine.

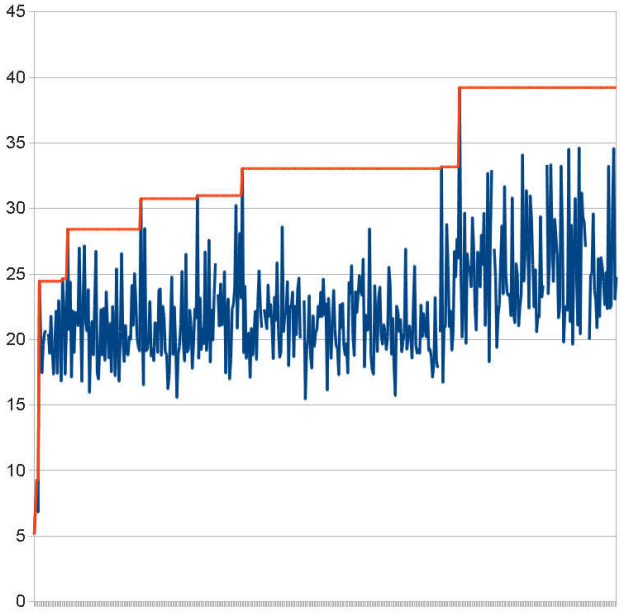


Figure 6: "Test" run showing trend toward higher "fitness" utilizing the search tree function

In Figure 6, the "test" case (B) from Figure 1 applies the search function, which clearly takes the initially high value produced by the same starting frame generated for the control case and improves on it over time. The strength of the search function is that subsequently generated frames eventually climb to a higher score-generating capacity ("fitness") over randomly generated control case frames. The search function will restart with lower performing simulations if all the potentially better options are exhausted. As seen in Figure 6, this causes a period where the evaluated simulation fitness (blue line) remains less than the best observed fitness (orange line). In this manner, the search function is operating as a Stochastic Hill Climbing algorithm in that the system has the ability to find its way out of traps set by local maxima.

EvoGrid Next Steps: Questions for the Computational Origins of Life

This very preliminary work poses far more questions than provides answers. However, as an early exemplar of computational origins of life (COoL) endeavors, the EvoGrid prototype and its proposed development path could serve as a roadmap to more fully functional platforms of the future. This roadmap also summons some broader issues, which might be considered a good start to a list of *open problems in computational origins of life.*

The greatest limitation in the EvoGrid prototype is our use of a naïve model of chemistry including the abstractness of our atom types, bond formation and the resulting "molecular

structures". Bonds are formed by simple proximity calculations using the positions, velocities and other data for objects exported from GROMACS. This situation may be improved by using the MOPAC7 library (Stewart, 2008) employed by GROMACS for covalent bond formation and the representation of other molecular affinities such as those produced by electrostatic and van der Waals forces:

1. Related to this first limitation is the need to go beyond the initial proof of concept prototype which is restricted to abstract atoms assembling into molecules. Our next steps must involve molecules assembling into larger structures that have the potential to exhibit properties of evolution. When this capability is prepared, a "real" set of experiments for testing the capabilities of the EvoGrid architecture should be attempted. Some proposed experiments include support for MD or coarse-grained simulation of lipid bilayer assembly reproducing the work of Fellerman (2009) using LAMMPS (Plimpton, 1995). Another good early test case would be to reproduce a simplified version of the groundbreaking experimental work by Bartel and Szostak (1993) in the isolation of new ribozymes from a large pool of random sequences.

2. The storage of frame states will be implemented in the near future. Temporal back-tracking is now being improved which will enhance the selective power of the search tree function. In addition, the computing resources of CALIT2 at the University of California at San Diego have been offered to the project, giving us critical storage and multiprocessor clusters for the next testing of the framework. A full work-up of computing and storage resources required by this architecture operating at different levels of simulation would be of value. Axes on a plot of EvoGrid computational complexity might include: number of particles and types of interactions handled for volume and time frame simulated, and desired level of fidelity to chemistry.

3. Another significant test of this concept would be the integration of simulation platforms other than GROMACS within the EvoGrid architecture to support heterogeneous simulations. For example, numerous engines, along the continuum of artificial chemistries from the highly abstract to the highly faithful to chemistry, are candidates to be integrated. In no particular order, candidate platforms are: The Organic Builder (Hutton, 2009), Avida (Adami and Brown, 1994), GARD (Segre and Lancet, 1999), NAMD (Philips et al., 2005), Desmond from Shaw et al (2008), and possible tie-ins to GPU-based hardware platforms (Anderson, 2008).

4. Bedau et al (2000) call for creating frameworks for synthesizing dynamical hierarchies at all scales. The heterogeneous nature of EvoGrid simulations would allow for coarse-graining procedures to focus simulation from lower levels to higher ones, saving computing resources by shutting off the less critical, more detailed simulations below. An example of this would be to switch to coarse grained simulation of an entire lipid vesicle, ceasing simulation of individual vesicle wall molecules. Conversely, fine grained simulations could be turned on for locally important details, such as diffusion of molecules through vesicle membranes. As exciting as this all sounds, a decade in the world of 3D simulation platforms has taught the authors of this paper that interfacing different software engines and representations of simulation space is extremely difficult. Running the same simulation space at multiple scales employing multiscale physics (e.g. from MD to dissipative particle dynamics, and beyond to smooth particle hydrodynamics) is also a very challenging problem that awaits future research.

5. A general theory of so-called cameo simulations needs to be developed to understand the minimum number of interacting objects and physical simulation properties required in these simulations for the emergence of "interesting" phenomena pertinent to life's building blocks. Our hypothesis that the GoEs in cameo simulations would apply to larger simulations also needs to be tested in the context of more ambitious COoL efforts capable of supporting artificial evolution thereby giving credence to the "Evo" in EvoGrid.

6. The EvoGrid cannot escape the meta-problem of all designed simulation environments: if we set up and simulate a system acting in the ways we accept as probable, then that system is much less likely to act in improbable and potentially informative ways, as results are always constrained by the abstractions and assumptions used. Another way of stating this very central conundrum is that as long as we do not know how chemical molecules might be able to exhibit emergence of important characteristics such as replication we will not be able to design the fitness functions to actually select for these molecules or their precursors. The fitness-function generation problem is as yet unsolved. However, the EvoGrid framework is being built to: 1) allow each potential experimenter to code in their own definition of fitness, accumulating knowledge applicable to the problem in an iterative fashion; and 2) support a more exotic solution in which the search functions themselves 'evolve' or 'emerge' alongside the simulation being searched. Actually building the second option would first require a much more extensive treatment from the field of information theory.

7. There are the deeper considerations that reach back to Langton who coined the term "artificial life" (Langton, 1986) and envisaged an investigation of *life as it could be*. COoL systems need not be constrained to models of the emergence of life on Earth. More abstract simulations may shine a light on *life as it might be* out in the universe (Gordon and Hoover, 2007), as a tool for use in the search for extraterrestrial intelligence (SETI) (Damer, 2010), or as a *technogenesis* within computing or robotic worlds.

8. A critic of theories of chemical evolution, cosmologist Sir Fred Hoyle used the statement about a ready-to-fly 747 aircraft being assembled by a tornado passing through a junk yard of parts (Hoyle 1984) to ridicule the idea of spontaneous generation of life at its origin. This idea today fuels creationist claims for irreducible complexity as one of their strongest arguments for the existence of a Creator. Like it or not, this flavor of debate will find its way to practitioners of COoL efforts. Gordon (2008), Damer (2008) and Barbalet and Daigle (2008) take this theme head on within a compendium of dialogues between creationists and scientists.

9. A corollary to Gordon's prediction (Gordon, 2008, p. 359) that Alife enthusiasts have an opportunity to solve the "Origin of Artificial Life" problem well before the chemists will solve the "Origin of Life" problem, is the very question of "what defines something as being life?". In the case of an *in silico genesis* we would ask "when will we know something is artificially alive?" Given latitude to speculate about these grand questions from such lofty heights of ignorance, it will be no surprise if emerging COoL endeavors attract a wide and vocal variety of converts and critics alike.

10. In the end the key question must be asked is: of what relevance is digital simulation to real chemistry or biology? Any given computational system might be able to show fascinating emergent phenomena but such discoveries might well stay trapped *in silico* and never transition over to inform experimentation *in vitro*. This would indeed be a shame and as such should motivate builders of systems like the EvoGrid to keep their eye on the ultimate prize: the transfer of concepts developed digitally into chemical experimentation. The inevitable marrying of these two media will produce one of the most powerful new tools for science and technology in the 21st Century.

Conclusion

A hybrid synthesis has been proposed between large scale high fidelity molecular dynamics simulations and distributed cameo simulations acting as an aggressive discovery system for the *genes of emergence* for some of life's building blocks. The EvoGrid is a framework under construction to support such distributed cameo simulations. Early results from a prototype implementation indicate that our search tree with temporal back-tracking optimization is performing as predicted as a stochastic hill climbing system. The EvoGrid software architecture has been shown to operate successfully with a large number of small, naïve chemical simulations run with the support of an industry standard MD engine. A listing of the current system's shortcomings and a roadmap for future development of the EvoGrid was presented. The authors concluded with a look at a few of the open questions applicable to the emerging field of computational origins of life (COoL) which is dedicated to "achieve the transition to life in an artificial chemistry *in silico*" (Bedau, et al. 2000).

Acknowledgements

We would like to thank Galen R. Brandt for a critical reading and Elixir Technologies Corporation for funding support of this work.

References

Adami, C., Brown, C.T. (1994). Evolutionary learning in the 2D artificial life system "Avida" In R. Brooks & P. Maes (Eds.), *Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, Cambridge, MA: MIT Press, 377–381.

Anderson, D.P. (2004), BOINC: A system for public-resource computing and storage, *Proceedings of the 5th IEEE/ACM international Workshop on Grid Computing* (November 08 - 08, 2004), International Conference on Grid Computing. IEEE Computer Society, Washington, DC, 4-10.

Anderson J.A., Lorenz C.D., Travesset A., (2008), General purpose molecular dynamics simulations fully implemented on graphics processing units, *Journal of Computational Physics*, 227 (10), 5342-5359.

Barbalet, T., Daigle J.P. (2008)
Biota Podcast 19, Retrieved April 8, 2010 from http://www.archive.org/download/biotapodcasts/biota_052308.mp3

Barbalet, T., Damer, B., Gordon, R. (2009)
Biota Podcast 44, Retrieved April 8, 2010 from http://www.archive.org/download/biotapodcasts/biota_032909.mp3

Bedau, M.A., McCaskill, J.S., Packard, N.H., Rasmussen, S., Adami, C., Green, D.G., Ikegami, T., Kaneko, K., & Ray, T.S. (2000). Open problems in artificial life. *Artificial Life*, 6, 363–376.

Bartel, D., Szostak, J., (1993), Isolation of new ribozymes from a large pool of random sequences, *Science*, 261, 1411-1418.

Damer, B., (2008), The God detector, In: *Divine Action and Natural Selection: Science, Faith and Evolution*, Eds.: R. Gordon & J. Seckbach. Singapore, World Scientific: 67-82.

Damer, B., (2010), SETI Institute Colloquium Lecture, retrieved June 10, 2010 from http://www.seti.org/csc/lecture/archive/2010

Dittrich, P., Ziegler, J., Banzhaf, W., (2001). Artificial chemistries - a review, *Artificial Life*, 7, 225–275.

Fellermann, H. (2009). Spatially resolved artificial chemistry, In: A. Adamatzky and M. Komosinski (eds.), *Artificial Life Models in Software 2nd Edition*, Springer, 2009: 343-370.

Gordon, R. (2008). Hoyle's tornado origin of artificial life, a computer programming challenge. In: *Divine Action and Natural Selection: Science, Faith and Evolution*. Eds.: R. Gordon & J. Seckbach. Singapore, World Scientific: 354-367.

Gordon, R., Hoover, R.B., (2007). Could there have been a single origin of life in a big bang universe? Proc. SPIE 6694, doi:10.1117/1112.737041.

Hoyle, F. (1984). *The Intelligent Universe*. New York, Holt Rinehart and Winston.

Hutton, T.J., (2009) The Organic Builder: A public experiment in artificial chemistries and self-replication, *Artificial Life*, 15(1), 21-28

Kalos, M.H. & P.A. Whitlock (2008). *Monte Carlo Methods*. Weinheim, Wiley-VCH Verlag GmbH & Co. KGaA, 2nd Edition.

Langton, C.G. (1986). Studying artificial life with cellular automata. *Physica D*, 22, 120–149.

Maruyama, K., Nori, F., Vedral, V. (2009). Colloquium: The physics of Maxwell's demon and information. *Reviews of Modern Physics* 81(1), 1-23.

Pande, V.S., Baker, I., Chapman, J., Elmer, S.P., Khaliq, S., Larson, S.M., Rhee, Y.M., Shirts, M.R., Snow, C.D., Sorin, E.J., and Zagrovic, B. (2003), Atomistic protein folding simulations on the submillisecond time scale using worldwide distributed computing. *Biopolymers*, 68(1), 91-109.

Phillips, J.C., Braun, R., Wang, W., et al., (2005), Scalable molecular dynamics with NAMD, *J. Comput. Chem*, 26(16): 1781-1802.

Plimpton, S.J., (1995), Fast parallel algorithms for short-range molecular dynamics, *J Comp Phys*, 117, 1-19.

Rajagopalan, R. (2001). Simulation of self-assembly systems, *Current Opinion in Colloid and Interface Science*, 6, 357-365.

Rasmussen S., Chen L., Nilsson M., Abe S., (2003), Bridging nonliving and living matter, *Artificial Life*, v.9 n.3, p.269-316, Summer 2003

Russell, S.J., Norvig, P. (2003), *Artificial Intelligence: A Modern Approach (2nd ed.)*, Upper Saddle River, New Jersey: Prentice Hall, 111-114.

Segre, D., Lancet, D. (1999), A statistical chemistry approach to the origin of life, *Chemtracts – Biochemistry and Molecular Biology* 12, 382–397.

Segre, D., Lancet, D. (2000), Composing life, *MBO Reports* 1, 217–222.

Shaw, D.E., Dror, R. et al. (2009), Millisecond-scale molecular dynamics simulations on Anton, *Proceedings of the ACM/IEEE Conference on Supercomputing (SC09)*, Portland, Oregon, November 14–20, 2009

Shaw, D.E., Martin M. et al. (2008), Anton, a special-purpose machine for molecular dynamics simulation, *Communications of the ACM*, 51 (7), 91–97.

Shenhav B., Lancet D. (2004), Prospects of a computational origin of life endeavor, *Origins of Life and Evolution of Biospheres*, 34(1-2), 181-94.

Stewart, J.P., MOPAC2009, Stewart Computational Chemistry, Colorado Springs, CO, USA, HTTP://OpenMOPAC.net (2008).

Van der Spoel, D., Lindahl, E., Hess, B., et al. (2005), GROMACS: Fast, flexible, and free, *Journal of Computational Chemistry*, 26(16), 1701-1718.

Adaptation Without Natural Selection

Richard A. Watson[1], Rob Mills[1], C.L. Buckley[2], Simon Powers[1], Alexandra Penn[1], Adam Davies[1], Jason Noble[1], Seth Bullock[1]

[1]Natural Systems group, ECS, University of Southampton, U.K.
[2]CCNR, Informatics, University of Sussex, U.K.
raw@ecs.soton.ac.uk

Extended Abstract

How can a system become better adapted over time without natural selection? Although some argue for 'organismic' properties such as robustness and self-sustaining regulation in non-evolved systems [1,5,11], others insist that natural selection is the only source of true adaptation [3]. We suggest that understanding how adaptation can occur without natural selection remains a fundamental open question for the Artificial Life community. For example, the origin of life, the origin of evolution, and the origin of new units of selection in the major evolutionary transitions/biological dynamical hierarchies, all seem to imply an adaptive process, or at least a non-arbitrary organisational process, that precedes the onset of natural selection proper (at each level of organisation).

In recent work we have been developing a number of inter-related concepts that approach this question from different angles [2,6,7,8,9,10,12,13,14,15]. In a general sense, it is known that a complex dynamical system can self-organise in a manner that reflects structure in external perturbations. But more specifically, we find that when variables in the system have a bi-modal distribution of decay constants (some fast and many slow), slow variables spontaneously act in a manner functionally equivalent to the weights of a neural network undergoing Hebbian learning, thereby modulating the behaviour of the fast variables such that the resultant internalised structure takes the form of an associative memory [4]. The proximal cause of these changes is merely that such a configuration is less resistant to, and hence less affected by, the perturbations to the system (c.f. homeostasis). But the system-scale consequences of this structuring is that such a system can 'recall', 'recognise' or 'classify' stimuli and, given appropriate structure in the perturbations, generalise to previously unseen stimuli, in just the same manner as a trained neural network [4].

This provides a framework to connect the concepts of a dynamical system merely 'doing what it does naturally' at one scale of explanation with interpretation as an *adaptive* system at another. In particular, in the joint phase space of both fast and slow variables the system merely decreases in energy, as one would expect from any purely mechanistic explanation. But induced structure in the slow variables improves the ability to dissipate energy from the fast state variables. Thus with respect to the fast system variables only, systems organised in this manner do not merely minimise system energy but *get better* at minimising energy over time. When the external environment of the system corresponds to an optimisation problem, the system thus improves its ability to solve that problem over time. It is in this sense that we can understand the system, not just as self-organised, but adapted. We present an abstract model and simulation of this process and discuss how it relates to a number of different domains: the evolution of evolvability in gene regulation networks [12], the evolution of new units of selection [10] via symbiosis [15] and 'social niche construction' [8,9], games on adaptive networks [2], distributed optimisation in multi-agent complex adaptive systems [13,14] and multi-scale optimisation algorithms [6,7].

References

1. Bateson, G. (1972). *Steps to an Ecology of Mind: Collected Essays in Anthropology, Psychiatry, Evolution, and Epistemology*. University Of Chicago Press.
2. Davies, A.P., Watson, R.A., Mills, R., Buckley, C. L., Noble, J. (2010) If you can't be with the one you love, love the one you're with: How individual habituation of agent interactions improves global utility. *ALife XII* (to appear).
3. Dawkins, R. (1982). *The Extended Phenotype*. Oxford, Freeman.
4. Hopfield, J.J. (1982) Neural networks and physical systems with emergent collective computational abilities, *PNAS USA*, 79 (8) 2554.
5. Lovelock, J.E. (1979). *Gaia: A New Look at Life on Earth.* Oxford, Oxford University Press.
6. Mills, R. (2010) *How Micro-Evolution Can Guide Macro-Evolution: Multi-Scale Search via Evolved Modular Variation.* PhD thesis, ECS, University of Southampton.

7. Mills, R., & Watson, R.A. (2009) Symbiosis Enables the Evolution of Rare Complexes in Structured Environments. To appear *Proceedings of European Conference on Artificial Life 2009*.
8. Powers, S. (2010) *Social Niche Construction: Evolutionary Explanations for Cooperative Group Formation*, PhD thesis, ECS Southampton (submitted).
9. Powers, S.T., Mills, R. Penn, A.S., Watson, R.A. (2009) Social niche construction provides an adaptive explanation for new levels of individuality (ABSTRACT), *Proceedings of Workshop on Levels of Selection and Individuality in Evolution, European Conference on Artificial Life.*(in press)
10. Powers, S.T., Penn, A.S. & Watson, R.A. (2007) Individual Selection for Cooperative Group Formation. *Proceedings of European Conference on Artificial Life 2007*. pp. 585-594.
11. Saunders, P.T. (1994) Evolution Without Natural Selection: Further Implications Of The Daisyworld Parable. *Journal of Theoretical Biology 166* (1994) 365-373.
12. Watson R.A., Buckley, C.L. & Mills, R. & Davies, A.P., (2010). Associative memory in gene regulation networks. *ALife XII* (in press).
13. Watson R.A., Buckley, C.L. & Mills, R. (2010). Optimisation in 'Self-modelling' Complex Adaptive Systems, *Complexity* (under revision)/ Tech. Report, ECS, University of Southampton.
14. Watson, R. A., Mills, R, and Buckley, C. L. (2009) *Global Adaptation in Networks of Selfish Components: Emergent Associative Memory at the System Scale*.. (submitted)/Tech. Report, ECS, University of Southampton.
15. Watson, R.A., Palmius, N., Mills, R., Powers, S.T., & Penn, A.S. (2009a) Can Selfish Symbioses Effect Higher-level Selection? *European Conference on Artificial Life 2009*. (in press).

The Blind Watchmaker's Workshop: three Artificial Chemistries in the context of Eigen's Paradox

Simon Hickinbotham[1], Adam Faulconbridge[2] and Adam Nellis[1]

[1]Department of Computer Science, [2]Department of Biology
York Centre for Complex Systems Analysis, University of York, Heslington, YO10 5DD, UK
sjh@cs.york.ac.uk www.yccsa.org

Abstract

We use Artificial Chemistries (ACs) as a way of addressing problems in Artificial Life (ALife) and evolution, by considering Eigen's paradox — small replicators with poor fidelity can not encode sufficient information to build a replicator with improved fidelity. We describe three AC case studies for different periods in the early evolution of the earth. From these, we discuss more general properties that are useful for ACs to possess for evolution, and compare our properties to those described by other authors.

We do not present a resolution of Eigen's paradox; rather we demonstrate a way of thinking about AC in the context of early evolution. Eigen's paradox is one key issue in this period. We use ACs as a model paradigm and from these we extract relevant properties that can be considered separately from the specific ACs that informed them; these properties can be used to inform design and analysis of future ACs.

Introduction

Artificial Chemistries (ACs) are a useful basis for experiments in Artificial life and evolution. Approaches to ACs in this area tend to emulate the 'central dogma' of biology, whereby information is encoded on macromolecules analogous to DNA, RNA, and proteins. This is a difficult modelling challenge due to the size of the molecules relative to their atomic constituents, and the complexity of the interactions between them. An alternative to this approach is to seek ACs that more closely resemble models of the early evolution of life on earth which do not have such a constrained linear flow of information. These stages may be easier to model due to their relative simplicity, and from these models, a set of properties can be derived that allow better models of the macromolecules of the central dogma of biology to be constructed. However, this pathway is not well understood in paleobiology and is therefore difficult to emulate. Recent work in paleobiology suggests that there were many different modes of evolution before the central dogma of biology became prevalent [25]. These modes exploit a more vague distinction between template (genotype-carrying) molecules and machine (phenotype) molecules. In this paper, we report work on ACs carried out separately by the three authors, that collectively emulate this period in the history of life.

One of the key problems an AC must handle is that any route from pre-biotic chemistry to the central dogma of biology must resolve *Eigen's paradox* [5]. This is Manfred Eigen's observation of the following cycle:

- Low-fidelity replicators are only able to preserve small genomes reliably.
- Small genomes limit the power of the phenotypes they express.
- So a small genome cannot encode a phenotype which contains a high-fidelity replicating mechanism

In essence, the poor copy fidelity of early genotypes could not encode the phenotype sufficiently accurately to preserve any improvements in copy fidelity.

We do not attempt to resolve Eigen's paradox here. Instead, we used the paradox as a challenge for AC design. This allows us to set ACs in a context and discuss their properties relative to this context. We argue for Goldberg's 'piecewise engineering' approach in the first instance [12] and take the view that a 'one size fits all' approach to AC design is not the most efficient way of approaching difficult problems. These problems are characterised by a system (such as chemistry, in the case of Eigen's paradox) that changes how it behaves as it develops through time. Before the resolution of Eigen's paradox, replicators were constrained in their size and therefore in their functionality; once the paradox has been resolved, this ceiling is lifted which allows for further evolution and adaptation, eventually leading to the central dogma of biology that we recognise today.

ACs can be used to produce Artificial Life (ALife) systems in which evolutionary features (such as reproduction or mutation) are not explicitly defined *a priori*. Instead, they are emergent properties of the system and as such are implicitly embedded:— they can be changed by the ALife system, rather than having to be pre-specified by a designer.

We investigate this by considering three different ACs which can represent the chemistry that existed before, after and during Eigen's paradox (figure 1). These chemistries come from recent work by the authors, developing ACs for three challenges: the origin of life [10]; the evolution of evolvability (*meta*-evolution) [21]; and as the basis for a self-maintaining genetic algorithm [16]. Note that the emphasis in these works is placed heavily on replication processes and do not consider the role of the container in the context of resolving Eigen's paradox. None of our chemistries currently model a cell membrane within the chemistry itself (but our chemistries do occupy a set volume and thus at least have the abstract concept of a container) although the emergence of membranes is linked to the emergence of replicators in models of the early earth. After describing these three chemistries, we discuss the properties they possess, how these relate to properties considered interesting by other authors [24] and how they relate to Eigen's paradox.

Finding a single chemistry to span these phases is much harder than finding different chemistries modelling each situation appropriately. The goal of our work in these three areas is to derive a new set of desired properties, to aid us in designing a series of ACs that together form an innovative artificial evolutionary platform. We are interested in finding which properties of ACs contribute to evolution and evolvability in general. Focusing on Eigen's paradox as an example of evolvability is a way in which we can tease out these properties.

The Context of Eigen's Paradox

A time-line of the beginnings of evolution on the early earth is shown in figure 1. This period is interesting to ALife researchers because it resolved Eigen's paradox [22], a key problem in evolution. The period begins with the 'late heavy bombardment' of the earth by debris from space as the solar system formed — only after this was the planet thought to be stable enough for life to prosper. Then come the well-known phases in the development of life on this planet, from the pre-biotic chemical 'soup' to the emergence of the central dogma of biology. The graphic in the middle of figure 1 illustrates the inheritance of genetic strategies over this period. Essentially, many different evolutionary strategies are prevalent, until the central dogma sweeps the planet as shown by the shaded region at the bottom of the graphic. Eigen's paradox is resolved before the emergence of replicator molecules that precede the central dogma of biology. The three chemistries forming the basis of the current contribution are shown to the right of the graphic in figure 1. These are described below.

From the perspective of the central dogma, Eigen's paradox is insoluble. It is not possible to construct a long genotype for an accurate copying phenotype from the basis of a short genotype that encodes an inaccurately-copying phenotype. And yet, the central dogma is common to all known life. Potential resolutions to Eigen's paradox are:

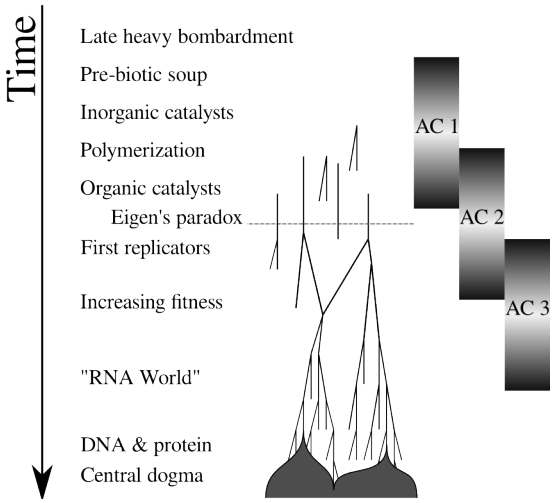


Figure 1: Timeline of the beginning of evolving systems. Events leading to the central dogma of biology are shown on the left. The resolution of Eigen's paradox is required for the emergence of competent replicators. The central graphic shows the myriad different evolutionary processes that are thought to have been prevalent before the central dogma. The three Artificial chemistries are shown on the right of the figure.

1. Stochastic processes throughout the planet over a billion years could ensure that, even though on average a short sequence does not copy well, given enough sequences, some might work well enough for long enough to encode a faithful genotype-copying arrangement.

2. Environment: there may have been local isolated environments where fidelity was higher and denaturation was reduced. If a long & accurate replicator could have arisen there, it could have spread to other locations; e.g. the presence of inorganic compounds such as clay crystals, could have aided replication [2].

3. The assumption that short sequences imply low fidelity is false. It may have been possible to construct some efficient copier from a short genome in some 'lost' chemistry. Alternatively, some collective property of the system does the job of forming an accurate template before the arrival of specialised template-carrying molecules.

Our chemistries explore the third possibility for resolution of the paradox. ACs for ALife could be used to find evolutionary mechanisms simpler than the central dogma of biology — this forms the central design objective of our ACs. It involves seeking simpler molecular machinery than DNA, RNA and protein, which will be easier to simulate computationally. However, by discarding the central dogma of bi-

ology, we have lost the ability to design replicators by looking at biology and attempting to copy what we see because these primitive replicators no longer exist on the Earth. We are faced with the task of designing from scratch an AC that can support recognisable evolution.

The paradox is related to ACs in two ways. Firstly, if we have an AC that cannot resolve this paradox, then the AC has a (small) maximum genome size that it can not overcome. If we want genomes larger than this size, then we must explicitly add in high-fidelity replicators. Secondly, the ACs may foster new theories about how Eigen's paradox can be resolved. We can design ACs to test these new theories.

Implementations

We now present a brief overview of the three chemistries referenced in figure 1. In its most basic form, an AC is defined as[4]:

- A set of molecules (both those present at a point in time and all possible molecules)
- Reactions that describe transformations between sets of molecules
- An algorithm which determines how the reactions are applied to the set of molecules present

A number of different ACs have been developed from this basis, without much consensus on which approach is 'best'. However, there have been a number of different properties and characteristics proposed as interesting features or requirements. ACs have also been applied in various other contexts [23, 20], but the power of ACs is limited if evolutionary processes are not implicit in the representation.

Our approach is to decompose the problem into three phases: emergence of self-replicators (AC1); evolution of evolvability (AC2); stable but primitive evolutionary system (AC3).

AC1: Emergence of Replicators

AC1 is an analogue of the pre-biotic soup in which early replicators emerged. It is designed as an source of open-ended chemical novelty and innovation, in which replicating molecular species may be initially formed. In this phase, replicators do not yet exist and therefore other processes and structures, such as autocatalytic sets [19] and hypercycles [6, 7, 8], are the focus of investigation.

One of the problems investigating the earliest phase of evolution is that there cannot be an assumption of a pre-existing replicating structure — it must be initially formed from other reactions. In order to achieve this, the chemistry must spontaneously generate sufficient novelty in order to describe templates and the molecular machinery to replicate them.

To implement an AC for this phase, we have developed a novel molecular representation classification, which we call

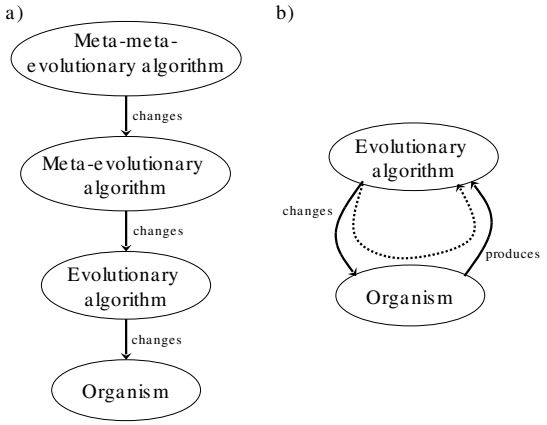


Figure 2: a) Naive meta-evolution suffers from the problem of how many meta-levels to use. b) Having the evolutionary algorithm as an emergent property of the organisms solves this problem. Evolution itself can choose how many levels of evolutionary algorithm to encode within the organism.

"sub-symbolic". Rather than reactants and products of reactions being defined in advance, they are determined by *bonding criteria* applied to *bonding properties* of the molecular species present; the bonding properties are themselves a emergent property of each atoms collection of sub-symbolic components. This means that for any molecule (either created within the system or provided by external input) all of its interactions can be generated dynamically.

Rather than try to specify a single AC that can achieve the emergence we seek, we have designed a framework within which many ACs can exist (RBN-World [10]). To find individual ACs that may achieve the goal of emergent replicators within this design space, we have developed a series of tests for desirable low-level properties. These form a set of 'stepping stones' that lead towards self-replicating systems. [9]

At the end of this phase, we anticipate a collection of molecules that form an autocatalytic set — production of every member of the set is catalysed by at least one member of the set. Taken as a cooperative collective, this forms a proto-organism capable of growth and replication.

AC2: Meta-Evolution

AC2 overlaps with AC1. AC2 is a meta-evolution phase in which speed and fidelity of replications increases as a loosely-replicating proto-entity becomes more capable of maintaining both its own fidelity and the fidelity of a larger reaction network [21]. The proto-entity will gradually evolve robust replication until it is widespread and prevalent.

AC2 implements an analogue of a traditional genetic algorithm (GA) in the same medium as the organisms themselves (figure 2). This requires the organisms and algorithm to be

implemented in a single representation, which a sufficiently rich AC can provide. We have identified the following requirements of an AC for meta-evolution:

- *template molecule(s)* that encode enzymes, including indirectly encoding the reactions that they can perform.
- *translation enzymes* that "read" the template molecule and construct the enzymes that are coded for.
- *replication enzymes* that can copy templates with some stochastic error so that mutations can occur.

We will encode initial examples of all of the above into template molecules within the system. This will allow meta-evolution to happen, because mutations occurring on the template molecule can cause the EA to change.

One part of evolving the EA is evolving the concept of mutation. We enable evolution of mutation because mutations can occur due to inexact copying of the template (mutation-on-copy). The replication enzymes are encoded on the template, and so the process of replication (and thus the process of mutation) can evolve under its own control.

The replication machines in this AC contain complex internal structure, and replication is a multi-step, character-by-character process. To replicate a template molecule, each character is replicated in turn by the following sequence of steps:

1. The next character from the template is read;
2. The replicator makes an internal representation of the next character;
3. Raw materials are picked up from the environment;
4. The raw materials are used to write the next character to the copy;
5. The replicator moves on to the next character on the template and the copy.

Because the copying process involves many steps, there are many ways in which is can go wrong. This means that many different types of mutation are possible, and also many different ways in which the replicator can evolve.

The replicators emerging from AC1 can be seen in AC2 as primitive and unstable with have low fidelity (high mutation). These will undergo metaevolution within AC2 to become the stable replicators of AC 3 exhibiting high fidelity (low mutation).

In relation to Eigen's paradox, this AC has a representation of replicating chemicals that can evolve their own copying fidelity. Therefore changes in the template and/or copy fidelity can be recorded over time and different conditions. This will enable examination of the conditions under which Eigen's paradox is resolvable and if it is inevitable.

AC3: "RNA world"

AC3 represents molecules that can copy with relatively high accuracy, even though there is not necessarily a distinction between template and machine.

AC 3 is called Stringmol . The Stringmol chemistry was developed to emulate molecular systems in such a manner that the binding and reactions between molecules could be varied using evolutionary approaches. In a nutshell, a molecule consists of a *sequence* along with a set of *flags* and *pointers* that allow the sequence to be executed as a program. Further details are available in [16] and [14]

There are two key features of the Stringmol system. The first is the *binding scheme*, which specifies the probability of two molecules joining together and creating a reaction. The second is the *mutation-reaction scheme*, which specifies how reactions occur under an environment of mutation, and determines what the products of the reaction are. Thus we have rules that handle the alignment of two strings of symbols (bound pair of molecules), and interprets the strings as a program and a data repository simultaneously.

Experiments with mutation in the Stringmol system have shown that a wide variety of phenomena can occur with no extenally-applied evolutionary pressure. In particular, we see the spontaneous emergence of autocatalytic sets from a basic replicase system [15].

Properties of Artificial Chemistries

It is useful to consider ACs in the light of the properties of ALife listed in [1]. ACs offer a route to generating "life" from the non-living by: A.2, exploring the transition to life *in silico*; A.3, discovering novel living organisations; A.4, determining how rules and symbols are generated from physical dynamics. Once a 'living' AC is constructed, then investigation can proceed, to: B.6, determine what is inevitable in open-ended evolution; B.7, explore evolutionary transitions (e.g. Eigen's paradox); B.8, provide the base layer of a hierarchical dynamical system; B.10, form the currency of an information processing theory for evolving systems. These ALife properties drive the properties of the underlying chemistry. One classification of desirable properties of an AC by Suzuki et al was published in [24] and is reproduced for convenience in table 1 alongside our summarised interpretations. We divide those ten properties into three groups: *molecule & reaction* properties, *membrane* properties and *mutation* properties.

New properties

Each of the three authors of this paper has independently developed ACs analogous to different stages in early evolution. We use these three 'case study' ACs to think about desirable properties of ACs in general.

In addition to the properties in table 1, there are some further properties we perceive to be desirable in an AC:

No.	Property	Interpretation	
1.	The symbols or symbol ingredients be conserved (or quasi-conserved) in each elementary reaction, at least with the aid of a higher-level manager.	Conservation of Mass	Molecules & reactions
2.	An unlimited amount of information be coded in a symbol or a sequence of symbols.	Molecules composed of atoms & bonds	
3.	Particular symbols that specify and activate reactions be present.	Catalysis	
4.	The translation relation from genotypes to phenotypes be specified as a phenotypic function.	Phenotypic gene expression	
5.	The information space be able to be partitioned by semi-permeable membranes, creating cellular compartments in the space.	Cells	Membrane
6.	The number of symbols in a cell can be freely changed by symbol transportation, or at least can be changed by a modification in the breeding operation.	Variable cell volume / concentration	
7.	Cellular compartments mingle with each other by some random process.	Cell movement	
8.	In-cell or between-cell signals be transmitted in the manner of symbol transportation.	Diffusion through membranes	
10.	Symbols be selectively transferred to specific target positions by particular activator symbols (strongly selective), or at least selectively transferred by symbol interaction rules (weakly selective).	Membrane pores & pumps	
9.	There be a possibility of symbols being changed or rearranged by some random process.	Spontaneous Mutation	Mutation

Table 1: The list of desirable AC properties from [24]. On the left is the original description, on the right is our summarised interpretation. NB: we classify property 10 as a membrane property along with 5-8 rather than a genome property with 9.

11. Novelty & innovation This is a property desired in evolutionary systems, and AC design should reflect this. If a new molecule is introduced to the chemistry, it should be able to interact with the other molecules present without requiring the AC to be changed. Furthermore, the AC should be able to generate novel molecules itself to allow innovative genetic architectures to emerge. This is related to Suzuki's properties #2: *Atoms and bonds* and #3: *Catalysis*, but rather than defining the function of molecules *a priori*, the possibility of novelty should be a general property of the molecular design. It is clear that ACs require this property in order to resolve Eigen's paradox, since without novelty there can be no transition between replicating systems. One can detect this property in absolute terms by asking whether it is possible to add a new molecular species to the system. If it is possible, one should then ask how *easy* it is to do so, and how easy it is for the system to *generate* new molecular species.

12. Range of Scales Although we do not think that all evolutionary phases should be supported by a single chemistry, we do think that chemistries should exhibit a wide range of scales — both spatially and temporally. Much of biology relies on reactions that proceed much slower than others, spanning several orders of magnitude in some cases. A large range of sizes of molecules are also present — from small metabolites consisting of a handful to atoms, to huge enzyme complexes with tens of thousands. Without such diversity, an AC would have limited scope for evolutionary exploration and therefore be restricted in terms of its potential behaviours and solutions to encountered problems.

A large range of spatio-temporal scales would also allow for smoother evolutionary slope climbing by gradual improvements once a solution has been found, for example with a faster rate or greater stability. Scale need not be measured in terms of size alone. Multi-scale representations are useful, because they offer a route to increase the efficiency of the system.

13. Dynamic environment History is littered with cases where an environmental change triggered an evolutionary breakthrough (punctuated equilibria [13]). There is also evidence that variation maintained by different environments can provide useful raw material for evolution, such as around deep-sea geothermal vents [11]. These dynamic environments can occur on many different scales; real-world biology varies from day/night cycles, to changing seasons and ice ages on a temporal scale and varies from micro-

environments between soil particles, through regional variations to continents (which themselves change over geological timescales). In order to utilise some of these dynamics, an AC should have parameters that can be varied (over time, space or both) to created different environments – analogous to temperature, pressure, pH, or other similar characteristics.

Dynamic environments allows a system to fully explore a chemistry, particularly if the rate of mutation varies. If the system can resolve Eigen's paradox locally within one environment, it can improve there and then spread to other environments — even if it could not evolve in those other environments directly.

14. Redundancy & degeneracy Successful evolutionary systems often contain *neutral mutation*. In an AC, this can be characterised by redundancy — multiple molecules that participate in equivalent reactions. However, neutral mutation is rarely completely neutral; it may have small side-effects. Degeneracy in an AC captures this by allowing two molecules to be equivalent for some reactions, but not for others.

In relation to Eigen's Paradox, redundancy and/or degeneracy can help by allowing multiple molecules to fulfil the same roles in the system. If one or more of these are lost through mutation, then the others may be able to partially or fully compensate. Techniques for measuring redundancy and degeneracy should be applicable to the AC, and give a feel for the expressive power of the system.

15. Emergent complex properties The reactions a molecular species participates in should be based on its structure, with similar molecules participating in similar reactions. However, there should be variation in this mapping such that while similar molecules in general have similar interactions, some similar molecules have very different interactions. This will allow an evolutionary landscape where gradual change generally occurs, yet there are some large changes in some regions. Combined with appropriate evolutionary pressures, this will lead to an efficient evolutionary engine.

16. Unified molecular representation There should be no 'special privileges' for template molecules — the property of holding genetic instructions should be an emergent property of the AC. This does not mean they have to be constructed from the same materials as other aspects of the chemistry, only that they should obey the same constraints and rules. In addition, if explicit membranes are used, they should also be represented without 'special privileges'.

The advantage of a unified molecular representation is that any part of the system can potentially interact with by any other part. This allows wider-ranging evolutionary changes and potentially highly innovative solutions to meta-evolutionary problems. It also means that the 'best' implementation of template molecules (or membranes) does not need to be hard-wired into the system beforehand — the system can be bootstrapped with an implementation that works and go on to optimise this itself.

17. Stochasticity Deterministic interactions between agents are a potential barrier to novel behaviour, and stochasticity can help smooth evolutionary changes by sampling the search space of possible alternatives. This leads to more efficient evolution when there are a large number of possible improvements.

18. Emergent mutation rates The replication mechanisms should enable the rate of error-on-copy to be modified. This allows the evolution of evolvability. A system that can reduce its own mutation rate in this manner can resolve Eigen's paradox by allowing larger templates to mutate less and so be more stable. But since the mechanism of genotype-encoding is changeable, the rate at which error accumulates cannot be set as an individual system-level parameter. Rather, the manifestation of error emerges from the reaction mechanism of the AC.

Mapping properties to three chemistries

Our three chemistries conform to the new properties listed in the previous section, thought no one chemistry contains all of them, but do not conform to some of the properties listed [24]. Below we show where our chemistries fit into Suzuki's and our own framework and the implications of those design decisions.

AC 1: Emergence of Replicators This AC analogue has a number of key properties within it. AC 1 implements #1: *conservation of mass* and #2: *atoms and bonds* of Suzuki's properties. Properties #3: *catalysis* and #4: *phenotypic gene expression* are deliberately not implemented in advance but are sought as emergent properties of the system. Our new property #11: *novelty & innovation* is the most important for this problem as we rely on novelty in order for replicators to emerge. Property #16: *unified molecular representation* is also key as we do not define what molecules fulfil which functions of the evolution of the system. #15: *Emergent complex properties* is another property that this systems is designed to exhibit, and is fundamental for the problem we are attempting to address.

Some properties we deliberately do not attempt to include in this AC. #18: *Emergent mutation* and Suzuki's #9: *spontaneous mutation* are not applicable to this phase, as there is not an explicit genome to be mutated; mutation-on-copy may appear as an emergent phenomenon however.

AC 2: Meta-Evolution The purpose of this AC is to investigate a rich mutation scheme, in particular #18: *emergent mutation* This is done by an enzyme-driven copying process with both #14: *redundancy and degeneracy* and #17: *stochastic* properties. This AC will display the emergent

complex property of meta-evolution when the copying machine is both encoded on the template being copied (which requires a #16: *unified molecular representation*) and situated in a #13: *dynamic environment* to provide a changing evolutionary pressure.

Relating this to Suzuki's properties, exploring #9: *spontaneous mutation* is also part of the purpose of this chemistry. In order for there to be a template to copy, this chemistry must satisfy #2: *atoms and bonds*. To be able to encode enzymes, we must satisfy #3: *catalysis*. The translation machines described above satisfy #4: *phenotypic expression* as they are both encoded and represented within the chemistry. To make evolution happen, this chemistry will enforce #1: *conservation of mass* through the atomic structures, which imposes additional restrictions upon the potential evolutionary solutions. As with AC 1 above, this chemistry is not especially concerned with membranes, and so properties #6, #7, #8 and #10 are not applicable to this chemistry. However, property #5: *containers* is satisfied in that membranes are implemented as simple containers, but their only function is to keep enzymes close to the templates they are acting on. There is no direct cell-cell interaction.

AC 3: "RNA world" Relating this AC to the the molecular and mutation-reaction properties described in table 1 [24] indicates that #1: *conservation of mass*, #2: *atoms and bonds*, #3: *catalysis*, and #4: *phenotypic gene expression* are all applicable to Stringmol . The mutation-reaction framework is more complicated however. In Stringmol mutation only occurs as new molecules are constructed, not spontaneously as specified by Suzuki et al. Mutations occur during the selective copy of symbols during a reaction of a particular type. This mimics biology more closely and can potentially be built into the AC to implement the meta-evolution described in AC 2.

Although this deviates from Suzuki *et al.*'s specification, mutation still occurs and it's rate can be controlled in a similar manner to the 'spontaneous' mutation in described (a 'cosmic ray rate'). Stringmol system allows reliable replication to be specified, but has a set mutation rate that allows adaptation to occur. These are the conditions in an 'RNA-world' which the Stringmol system was designed to emulate, and which has the capability to produce innovative responses.

Turning to the remainder of our new properties, #14: *Redundancy & degeneracy* are properties of this system, as well as #17: *stochasticity* due to the variable binding affinities. There is also the possibility for #11: *novelty & innovation* in terms of novel sequences with novel behaviours. Interestingly, the baseline mutation scheme allows a richer suite of macro-mutations to arise, with dramatic changes in the inter-molecular dynamics of the replication process. Stringmol therefore possesses our new property #18: *Emergent mutation rates*.

Conclusion

AC designs have to trade off between being rich enough to exhibit interesting behaviours and being simple enough to be computationally tractable. To address this, we develop abstractions with two goals: 1, to make the rich behaviour computationally tractable, and 2, to discover which properties underlie the richness. When using ACs to address evolutionary problems, the goals become further complicated. For example, in real chemistry the problems and solutions regarding survival of the organism have changed over time — the first forms of life were very different to modern populations of multi-cellular organisms. We use Eigen's paradox as an example of applying ACs to a evolutionary problem. We are not aiming to provide a resolution of Eigen's paradox: we provide a way of thinking about problems in which the properties and behaviours of the chemistry change over time (before, during and after the paradox).

In this work we have not looked at properties involving membranes and other spatial characteristics (#5: *cells with membranes*, #6: *variable cell volume / concentration*, #7: *cell movement*, #8: *diffusion through membranes*, and #10: *membrane pores & pumps* from Suzuki *et al.*). This is because these properties are predominantly under the control of the 'kinetics' used for any particular implementation of an AC. In our experiences, the kinetics component of the model can often be interchanged between different ACs depending on the features under investigation and available computational resources. For example, previous work on membranes in an AC [17, 18, 3], whilst clearly demonstrating interesting behaviours, poses computational challenges when used for investigations of evolution and novelty.

By considering specific ACs for three phases of evolution in the context of Eigen's paradox, we have concentrated on the properties needed for each phase. In all of these ACs, sub-symbolic atomic representations are useful because they preclude the need to create a set of reaction rules whenever a novel molecular species is produced, and so provide an appropriate platform for evolution to discover and preserve novel solutions which confer some benefit on the system. Effectively, using the sub-symbolic representation provides many properties for 'free'; #1: *conservation of mass*, #2: *atoms and bonds* and #3: *catalysis* from Suzuki's properties as well as #11: *novelty & innovation* and #16: *unified molecular representation* from our additional properties.

We have presented eight new properties in addition to the ten given in [24]. We have used Eigen's paradox as a context to map these properties onto our ACs to demonstrate how they can be used in the design and evaluation process. The resulting set of principles can be used for the design of a more generally applicable set of ACs.

Acknowledgements

Simon Hickinbotham and Adam Nellis are funded by the Plazzmid project, EPSRC grant EP/F031033/1. Adam

Faulconbridge is funded by BBSRC.

References

[1] Bedau, M. A., McCaskill, J. S., Packard, N. H., Rasmussen, S., Adami, C., Green, D. G., Ikegami, T., Kaneko, K., and Ray, T. S. (2000). Open problems in artificial life. *Artificial Life*, 6(4):363–376.

[2] Cairns-Smith, A. (1982). *Genetic takeover and the mineral origins of life*. Cambridge University Press.

[3] di Fenizio, P. S., Dittrich, P., and Banzhaf, W. (2001). Spontaneous formation of proto-cells in an universal artificial chemistry on a planar graph. *Lecture notes in computer science*, pages 206–215.

[4] Dittrich, P., Ziegler, J., and Banzhaf, W. (2001). Artificial chemistries-a review. *Artificial Life*, 7(3):225–275.

[5] Eigen, M. (1971). Self-organization of matter and the evolution of biological macromolecules. *Naturwissenschaften*, 58(10):465–523.

[6] Eigen, M. and Schuster, P. (1977). The Hypercycle: Part A. *Naturwissenschaften*, 64(11):541–565.

[7] Eigen, M. and Schuster, P. (1978a). The Hypercycle: Part B. *Naturwissenschaften*, 65(1):7–41.

[8] Eigen, M. and Schuster, P. (1978b). The Hypercycle: Part C. *Naturwissenschaften*, 65(7):341–369.

[9] Faulconbridge, A., Stepney, S., Miller, J. F., and Caves, L. (2010). RBN-world: The hunt for a rich achem. In *Proc. ALife XII*. MIT Press.

[10] Faulconbridge, A., Stepney, S., Miller, J. F., and Caves, L. (in press). RBN-world: A sub-symbolic artificial chemistry. In *ECAL 2009 (LNCS 5777)*. Springer.

[11] Gibson, R. N., Atkinson, R. J. A., and Gordon, J. D. M. (2001). *Oceanography and Marine Biology: An Annual Review*. CRC Press.

[12] Goldberg, D. E. (2002). *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA.

[13] Gould, S. J. and Eldredge, N. (1977). Punctuated equilibria: the tempo and mode of evolution reconsidered. *Paleobiology*, 3(2):115–151.

[14] Hickinbotham, S., Clark, E., Nellis, A., Pay, M., Stepney, S., Clarke, T., and Young, P. (2009). An abstract metabolism of molecular microprograms. Technical Report (in preparation), University of York.

[15] Hickinbotham, S., Clark, E., Stepney, S., Clarke, T., Nellis, A., Pay, M., and Young, P. (2010). Diversity from a monoculture: effects of mutation-on-copy in a string-based artificial chemistry. In *Proc. ALife XII*. MIT Press.

[16] Hickinbotham, S., Clark, E., Stepney, S., Clarke, T., Nellis, A., Pay, M., and Young, P. (in press). Molecular microprograms. In *ECAL 2009 (LNCS 5777)*. Springer.

[17] Hutton, T. (2004). Making membranes in artificial chemistries. In *Workshop and Tutorial Proceedings, Artificial Life IX*, pages 71–76. Citeseer.

[18] Ikegami, T. (2009). Shapes and Self-Movement in Protocell Systems. *Artificial Life*, 15(1):59–70.

[19] Kauffman, S. A. and Farmer, J. D. (1986). Autocatalytic sets of proteins. *Origins of Life and Evolution of the Biosphere*, 16(3-4):446–447.

[20] Meyer, T., Yamamoto, L., and Tschudin, C. (2008). An Artificial Chemistry for Networking. *Bio-Inspired Computing and Communication*, pages 45–57.

[21] Nellis, A. and Stepney, S. (2010). Meta Evolution (in preparation).

[22] Smith, J. M. and Szathmary, E. (1997). *The Major Transitions in Evolution*. New York: Oxford University Press.

[23] Straatman, B., White, R., and Banzhaf, W. (2008). An Artificial Chemistry-based Model of Economies. *Artificial Life*, 11:592.

[24] Suzuki, H., Ono, N., and Yuta, K. (2003). Several necessary conditions for the evolution of complex forms of life in an artificial environment. *Artificial Life*, 9(2):153–174.

[25] Vetsigian, K., Woese, C., and Goldenfeld, N. (2006). Collective evolution and the genetic code. *Proceedings of the National Academy Science*, 103:10696–10701.

Systems Biology

Systems Biology Approaches for Developing Artificial Pathogen Detection Networks

Anu Chaudhary[1], Geoffrey S. Waldo [1], William S. Hlavacek[2] and Chang-Shung Tung[2]

[1]Biosciences Division, Los Alamos National Laboratory, Los Alamos, NM 87544 USA
[2]Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87544 USA
anu@lanl.gov

Extended Abstract

Toll-like receptors (TLRs) offer the first line of host defense by recognizing the danger signals of pathogen and by inducing intracellular signaling that culminate in pathogen specific innate immune responses. We have been studying the early events that occur upon engagement of TLRs in cells. These events include protein phosphorylation and protein-protein interactions[1]. Intracellular protein interactions mediated by adapter proteins in the host are critical for generating an innate immune response. We have studied these interactions in both the cellular context, and by using isolated proteins. To minimize the complexity of working with cells, we are now developing a *bottom-up* approach to recreate the initial signaling that is triggered by TLRs, by generating protein assemblies *in vitro*. This will make it possible to directly and cleanly understand the prototypical signaling cascades involved in the ability of the host to detect pathogen components and mount an appropriate response. Although we are still very far from rationally assembling and understanding all of the design principles under which biological networks operate, tools of synthetic biology and computation developed by us and others offer the prospect of design and manufacture of networks with reportable and predictable properties.

To investigate the nature and specificity of interactions taking place in the host, we are using both cell-based and cell-free approaches. Cutting-edge reporter technologies help us design and analyze these systems. The split-luciferase protein technology can report various protein interactions in a high-throughput format [2]. The split-green fluorescence protein (GFP) technology, has allowed us to study protein folding and aggregations of protein domains [3,4], and is available in a multi-color format. Finally, the novel, triple-split GFP technology developed in the Waldo laboratory at LANL allows us to investigate specificities of protein-protein interactions by flow cytometry and imaging. Homology-based[5] and docking-typed[6] modeling approaches have allowed us to develop protein oligomer structures, and identify and validate critical interfaces that play a role these interactions. Finally, we are building predictive models of TLR signaling events and attempting to understand the design principles of cellular regulatory systems[7,8]. In summary, synergisms between experimental and theoretical approaches will allow us to develop artificial signal transduction systems that mimic the early steps of pathogen recognition by the host innate immune system. Such systems will allow us to understand, manipulate, and control early steps that play a role in pathogen detection.

References

1 Chaudhary, A., Fresquez, T. M. & Naranjo, M. J. Tyrosine kinase Syk associates with toll-like receptor 4 and regulates signaling in human monocytic cells. *Immunol Cell Biol* **85**, 249-256, (2007).
2 Remy, I. & Michnick, S. W. A highly sensitive protein-protein interaction assay based on Gaussia luciferase. *Nat Methods* **3**, 977-979, (2006).
3 Cabantous, S., Rogers, Y., Terwilliger, T. C. & Waldo, G. S. New molecular reporters for rapid protein folding assays. *PLoS One* **3**, e2387, (2008).
4 Cabantous, S. & Waldo, G. S. In vivo and in vitro protein solubility assays using split GFP. *Nat Methods* **3**, 845-854, (2006).
5 Tung, C. S., Joseph, S. & Sanbonmatsu, K. Y. All-atom homology model of the Escherichia coli 30S ribosomal subunit. *Nat Struct Biol* **9**, 750-755, (2002).
6 Tung, C. S., Walsh, D. A. & Trewhella, J. A structural model of the catalytic subunit-regulatory subunit dimeric complex of the cAMP-dependent protein kinase. *J Biol Chem* **277**, 12423-12431, (2002).
7 Hlavacek, W. S. *et al.* Rules for modeling signal-transduction systems. *Sci STKE* **2006**, re6, (2006).
8 Faeder, J. R., Blinov, M. L. & Hlavacek, W. S. Rule-based modeling of biochemical systems with BioNetGen. *Methods Mol Biol* **500**, 113-167, (2009).

(M,R) Systems and RAF Sets: Common Ideas, Tools and Projections

S. Jaramillo[1], R. Honorato-Zimmer[1], U. Pereira[1], D. Contreras[1], B. Reynaert[1], V. Hernández[1], J. Soto-Andrade[1], M.L. Cárdenas[2], A. Cornish-Bowden[2] and J.C. Letelier[1]

[1] Facultad de Ciencias, Universidad de Chile, Casilla 653, Santiago, Chile.
[2] Bioénergétique et Ingénierie des Protéines, CNRS, Marseille, France.
letelier@uchile.cl

Abstract

There are deep underlying similarities between Rosen's *(M,R)* systems as a definition of life and the *RAF sets* (Reflexive Autocatalytic systems generated by a Food source) introduced by Hordijk and Steel as a way of analyzing autocatalytic sets of reactions. Using RAF concepts we have systematically explored the set of possible small idealized metabolic networks, searching for instances of *(M,R)* systems. This exhaustive search has shown that the central requirement of Rosen's framework, unicity of Φ, becomes harder and harder to obtain as the network grows in size. In addition, we give an expression for operators f, Φ and β in terms of RAF sets.

Introduction

Metabolic closure is easy to introduce *informally* but rather difficult to define. Although it is crucial for understanding living organization it was neglected until late in the 20th century. The rebirth of the scientific study of biological organization can be traced back to the 30-year period from 1958 to 1987, which saw the publication of several distinct perspectives on closure, including *(M,R)* systems (Rosen, 1958), the chemoton (Gánti, 1975), hypercycles (Eigen and Schuster, 1977), autopoiesis (Maturana and Varela, 1980), autocatalytic sets (Kauffman, 1986), and the first Artificial Life conference in Los Alamos in 1987 (organized by Christopher Langton). There was, however, an almost complete lack of cross-fertilization between the different schools of thought, with each theory developed with almost no reference to any of the others (Letelier et al., 2006; Cornish-Bowden et al., 2007; Cárdenas et al., 2010). The most extreme case of isolation is represented by Robert Rosen (1934-1998), who introduced the concept of *(M,R)* systems early in his career to represent biological metabolic networks. His isolation was aggravated by the intricate nature of his writings, in which biological ideas were mixed with abstract mathematics. Furthermore, he expressed his mathematical ideas in non-standard notations and without any effort to help the reader by giving examples or offering many needed clarifications.

In recent years, we have undertaken a systematic attempt to understand and explain the core notions of Rosen's theory (Letelier et al., 2006). We have (a) clarified the relationship between *(M,R)* systems and autopoiesis (Letelier et al., 2003); (b) reframed Rosen's original formulation in terms of biochemical networks, with the introduction of the notion of "organizational invariance" for understanding Rosen's elusive mathematical operators (such as his β); (c) made a clear distinction between *(M,R)* systems in general and *(M,R)* systems with organizational invariance, a notion that is only implicit in Rosen's writing (he confusingly called these "replicative" *(M,R)* systems); (d) given mathematical and biological examples of simple idealized systems that can be understood within Rosen's intellectual framework; (e) clarified how these notions can be used to explore the origin of living systems and how they should be used in the context of what has come to be called "systems biology". Finally, we have also shown how our formulation of *(M,R)* systems can shed light on the problem of the computability of living systems (Cárdenas et al., 2010). This short summary is intended simply to underline how fruitful Rosen's view of metabolic closure has become, and to explain why we feel that the boundaries of our knowledge can be pushed to qualitatively new grounds by continuing the exploration of his ideas.

The systematic absence of examples (whether mathematical or biological) from Rosen's work has always been problematical, especially of simple examples that can serve as heuristic devices for enhancing theoretical research. In this paper we address the two points outlined above by pointing out the close relationship between *(M,R)* systems and a recent theory of living organization based on what have been called RAF sets. We show how many examples of simple *(M,R)* systems can be found by a computer algorithm constructed on the model of RAF sets. We discuss how the technical tools originating in RAF sets can be used to enhance the research of *(M,R)* systems, and specifically we address the problem of the nature and unicity of Rosen's Φ in the context of RAF sets.

(M,R) systems

Rosen's original formulation of *(M,R)* systems (Rosen, 1958), relied on a view of metabolism as a graph, and on a very abstract view of enzymes as functions (in the mathematical sense). The metaphor of metabolism as a graph, new in 1958, has subsequently been adopted by many people, without attribution to Rosen. The view of enzymes as functions has not attracted a wide following as Rosen's formulation seems unnecessarily abstract, without bringing practical or theoretical benefits. He used this approach in order to be able to use *category theory* for framing his important intuition about metabolic closure. Although this demanding mathematical approach has some advantages, as described in our previous work, we shall not use it here as the fundamental ideas exposed by Rosen can be explained using set theory, and thereby become accessible to mainstream biologists.

Our analysis of *(M,R)* systems, together with our examples, shows that the crucial aspect to understand organizational invariance is to understand the nature of the equation

$$\Phi(b) = f$$

Here Φ represents the aspect of biological organization that relates how catalysts are produced by the system. This equation seems to imply that a living system is organized in such a way that knowing b (right-hand side of biochemical equations) should be enough to unambiguously assign the catalysts (represented by f) to the reactions in the network.

Rosen, moreover, requires that there be only way to carry out this assignment, i.e., that there is only one mapping Φ such that $\Phi(b) = f$, a demanding assumption indeed. In other words, that we can reverse the procedure that gives f back from Φ. The reverse procedure is Rosen's β, so that

$$\beta(f) = \Phi$$

Mathematically, β is just the inverse of the "evaluation at b" operator that evaluates every function at b. Biologically, β represents the mechanisms that specify how the process of creating catalysts is maintained over time, i.e., *organizational invariance*.

To clarify these notions, we created a small metabolic network where they can be embodied in actual molecules that implement the functions Φ and β (Letelier et al., 2006).

RAF sets

We now give a brief introduction to the work of Hordijk and Steel (2004), who constructed a formal framework to study autocatalytic systems. Their main aim appears to have been to expand Kauffman's formalism about autocatalytic sets (Kauffman, 1993), to respond the criticisms that arose out of Kauffman's assumptions. At the same time, their analysis developed interesting algorithms that handle this expanded framework. As a result, they have produced a powerful approach that can be used to analyze a wide variety of systems, and here we shall describe how it applies to *(M,R)* systems. Their formalism depends on the following two sets: X, the set of molecules involved in metabolism as metabolites, catalysts or external input material (termed *food* in the formalism), and $\mathscr{R}$, the set of reactions that defines the metabolic network.

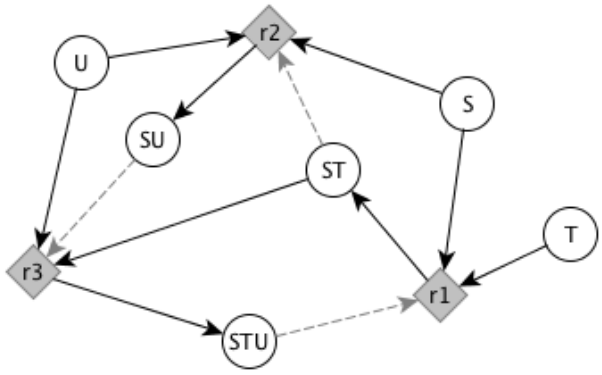


Figure 1: *(M,R)* system described by a *catalytic reaction graph*. Gray squares represent reactions and circles denote metabolites and enzymes. The black arrows represent chemical transformations while gray dashed arrows indicate catalyzations. This small network also contains a RAF set generated by the food set $(\mathrm{S}, \mathrm{T}, \mathrm{U})$.

Each reaction r is represented as a tuple (A, B), where $A, B \subset X$, $A \cap B = \varnothing$, A are the reactants and B the products of reaction r. This formalism is similar to Rosen's treatment of enzymes as transformations between two sets of molecules.

Further, to formalize the notion of catalysis, a specific set C (called the set of "catalyzations" by Hordijk and Steel), is introduced. Each catalyzation c is a tuple (x, r), where $x \in X$ is the catalyst and $r \in \mathscr{R}$ is the reaction catalyzed by x. The similarity with Rosen (1958) is evident, as any given catalyzation $c = (x, r)$ can be rewritten as $c = (x, r) = (x, (A, B)) = (A, x, B)$, making transparent the fact that molecule x catalyzes the reaction $A \to B$.

With the set of catalyzations defined, Mossel and Steel (2005) introduced a function γ that helps to simplify formulae in later sections:

$$\gamma_C(A, r) = \begin{cases} 1 \text{ if } \exists x \in A : (x, r) \in C, \\ 0 \text{ otherwise} \end{cases} \quad (1)$$

Additionally, a specific subset of X containing every molecule that is used but not produced by the metabolism is denoted F and it represents the food molecules.

Thus a *catalytic reaction system* over a food source F is composed by a triplet $\mathscr{L} = (X, \mathscr{R}, C)$ that defines the universe of molecules (X), the reactions occurring among these

molecules ($\mathscr{R}$) and the identity of the catalyst involved in each reaction (C) (see Figure 1). The following additional functions are defined: $\rho(r) = A$ and $\pi(r) = B$, which return the reactants and the products of any given reaction r, respectively. With the help of these elementary functions the same notion can be extended to a *set of reactions* $\mathscr{R}'$ as $\rho(\mathscr{R}') = \bigcup_{r\in\mathscr{R}'} \rho(r)$, where $\mathscr{R}' \subseteq \mathscr{R}$. This definition captures the conglomerate of molecules that participate as reactants for a set of reactions. A similar definition holds for $\pi(\mathscr{R}')$, the products of a subset of reactions. With these ideas, we can define the *closure* of a subset $X' \subseteq X$ relative to $\mathscr{R}' \subseteq \mathscr{R}$ ($cl_{\mathscr{R}'}(X')$) as the set of reachable molecules that can be synthesized by starting from X' and applying all the reactions in $\mathscr{R}'$ until no new molecule types appear. Then, a non-empty reaction subset $\mathscr{R}'$ of $\mathscr{R}$ is a *reflexively autocatalytic* network over F if $\rho(\mathscr{R}') \subseteq cl_{\mathscr{R}'}(F)$ and for each $r \in \mathscr{R}'$, $\gamma(\rho(\mathscr{R}') \cup \pi(\mathscr{R}'), r) = 1$. In other words every catalyst must be produced by a reaction in the same system or be part of the food set. This definition allows many reflexively autocatalytic networks in a *catalytic reaction system*. The network is F-generated if every reactant is either produced by the system or incorporated as a food item (i.e. formally $\rho(\mathscr{R}) \subseteq F \cup \pi(\mathscr{R})$). A network that is *reflexively autocatalytic* and F-generated is called a RAF set (see Figure 1).

RAF sets can be understood informally as an interdependent set of biochemical reactions where all of the metabolites are produced by the collection of reactions $\mathscr{R}'$. The advantage of this formalism is that it is precise enough to be coded in well defined algorithms that check whether a given reaction subset $\mathscr{R}' \subseteq \mathscr{R}$ is a RAF set over some food set F. We have implemented these algorithms, and we have created a simple framework in Lisp and Python, allowing us to carry out qualitative and quantitative analyses of *(M,R)* systems in terms of RAF formalism. Before discussing this, however, we need to show the extent to which RAF sets and *(M,R)* systems are equivalent.

RAF sets and *(M,R)* systems

Are *(M,R)* systems RAF sets? The original definition of an *(M,R)* system (Rosen, 1958) explicitly requires every catalyst (*M* in his original symbols) must be produced by the metabolism (*R* sub-systems are responsible for this task). This condition shows that *(M,R)* systems must be *reflexively autocatalytic* (RA) sets. Although, this does not necessarily imply that a RA set is an *(M,R)* system, because metabolic closure requires that no catalyst is given in the food set. In other words, a RA set is not in general an *(M,R)* system, but it may become one if all the catalysts in C are produced by the system and are not part of the food set F.

As *(M,R)* systems must be open to the flow of matter in order to satisfy thermodynamic requirements, their molecules derive ultimately from a food source, and they are, obviously, F-generated in the terminology of RAF sets. So *(M,R)* systems without organizational invariance are a subset of RAF sets, as are *(M,R)* systems with organizational invariance. The latter must, however, have additional features (in the context of RAF) to explain the unusual properties of operators Φ and β.

Algorithmic search for simple metabolic *(M,R)* systems

In this section we explore the probability of occurrence of an *(M,R)* system with a unique assignment of catalysts. For this purpose we characterized all the possible graphs describing a system consisting of a number $\#F$ of initial molecules and $\#\mathscr{R}$ synthesis reactions between any two molecules in the system. More specifically, we analyzed systems that conformed with the requirement of being *(M,R)* systems, that is, we did not allow any catalyst to be food, nor a reactant nor a product in the reaction it catalyzed.

Attention must be paid to avoid having two apparently distinct reaction networks exhibiting the same topological structure. The mathematical term for this is *graph isomorphism* (see Figure 3). Two graphs are said to be isomorphic when they can be transformed into each other by a simple relabeling of their vertices. Isomorphic metabolisms can be grouped under an equivalence class.

Thus, for a given pair $(\#F, \#\mathscr{R})$ we enumerate the number of all possible different equivalence classes of reaction networks. Next, for each one of these reaction networks, we generated the set of all possible assignments for the catalysts complying with the restrictions stated previously. But again, by the argument of relabeling, the set of assignments can be

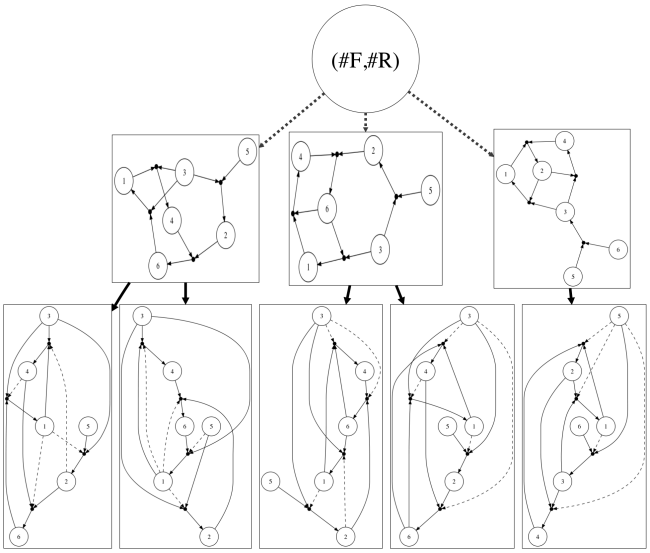
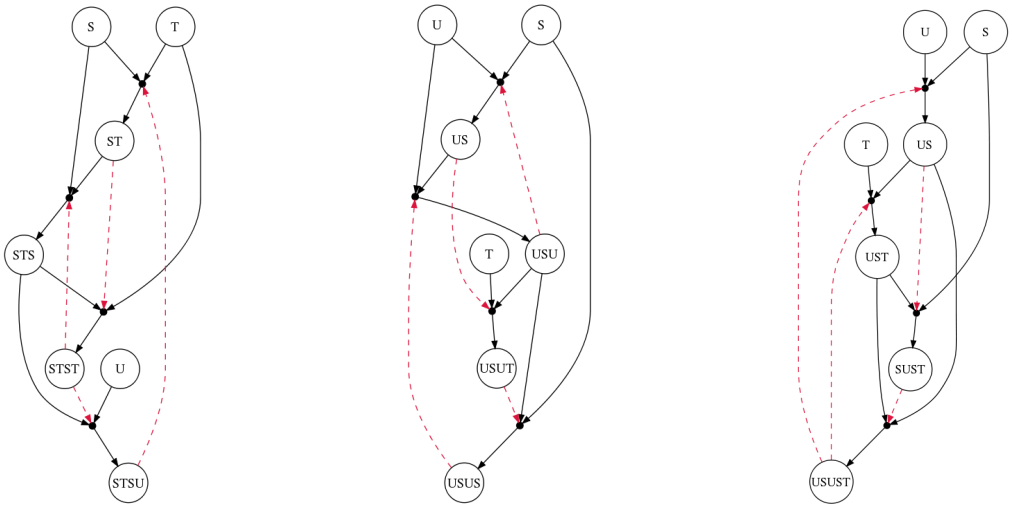


Figure 2: Diagram representing an example for the procedure to compute results from table 1. In the first step, the equivalence classes (3 in this example) are estimated for a given $(\#F, \#\mathscr{R})$; in the second step, all possible catalysts assignments for each equivalence class are calculated.



(a) (b) (c)

Figure 3: Three automatically generated RAF sets illustrating equivalence class and multiple catalyst assignments. Systems (a) and (b) have the same topological structure, i.e. there is an isomorphism from one to the other. Although this might not be obvious at first sight, a simple procedure of node relabeling transforms the reaction pathway in (a) to the one in (b). In spite of that, the systems differ in their catalyst assignments, i.e., even with the additional rules imposed by *(M,R)* systems, it is possible to make different choices when assigning the catalysts. System (c) has the same number of elements in the food set and the same number of reactions, but it belongs to another equivalence class.

also divided into equivalence classes (see Figure 2). Table 1 shows for $(\#F, \#\mathscr{R})$ the number of metabolic equivalence classes and the interquartile range[1] of the number of assignments. It can be seen that the number of possible assignments grows steeply with the number of reactions, so that it becomes more and more difficult to have a unique $\Phi(b) = f$ (Letelier et al., 2006).

There are some cases in which the range includes the critical value 1, which implies organizational invariance. Although, if we increase the number of food elements and leave the number of reactions unchanged, the generated reaction networks become shallower, and so we can consider the complexity of the network to be reduced and therefore the degrees of freedom of the assignation process are also reduced. In principle we could separate the trivial cases from those in which the unicity of the assignment reflects organizational invariance.

Rosen's triad in RAF formalism

The RAF formalism is not only useful for exploring the landscape of possible *(M,R)* systems, but it can also help to clarify some core concepts of *(M,R)* systems, namely Rosen's *triad*: f, Φ and β.

To explore the potential of the RAF formalism, we analyze the old problem in the theory of *(M,R)* systems of how to treat molecules as functions. Consider the following biochemical reaction:

$$\mathrm{x} + \mathrm{y} \xrightarrow{M} \mathrm{w} + \mathrm{z}$$

According to Rosen, this is the manifestation of the following function:

$$M \in Map(X \times Y, W \times Z)$$
$$M : X \times Y \to W \times Z$$
$$(x, y) \to (w, z)$$

The input elements are derived from the cartesian set $X \times Y$ that contains all the molecular types that, because of their structural similarities, can be used by the enzyme M as substrates. Our RAF-derived formalism extends the domain of function M to the whole set of molecules as follows: M is a function that, when given a set of molecules with the reactants, e.g. $(\ldots, x, \ldots, y, \ldots)$, returns a set containing elements w and z. But if the original input set lacks elements x or y, we have $M(\textit{input set}) = \varnothing$. Interestingly, with this formalism any molecule in the network ($x \in X$) can be treated as a function operating on any subset ($X' \subseteq X$) as follows:

$$x(X') = \pi(r_x) \text{ provided that } \rho(r_x) \subseteq X'$$

where r_x stands for the reaction that x catalyzes. If x catalyzes more than one reaction[2], then the above definition can

[1] This refers to the range in which data falls after removing lower and upper 25%, thus giving a notion of the amplitude of the mean values

[2] This multifunctionality seems to be necessary for *(M,R)* systems (Letelier et al., 2006).

Number of food molecules	Number of reactions					
	3		4		5	
2	4	2–2	19	12–24	136	144-216
3	10	1–4	72	12–31	685	216–324
4	8	1–6	75	1–36	933	204–432
5	2	1–1	37	1–34	577	1–432
6	1	1–1	11	1–1	212	1–1

Table 1: Number of metabolic equivalence classes and the interquartile range of the number of their possible assignments. The number of equivalence classes increases dramatically with the number of reactions.

be generalized to:

$$x(X') = \{x_i : x_i \in \pi(r) \mid (x,r) \in C \wedge \rho(r) \subseteq X'\} \quad (2)$$

Note that defining x only requires the set of reactions each molecule catalyzes, not the whole reaction network. This means that every molecule-as-a-function definition depends only on local information.

In our earlier work, the following small metabolism was used as a testbed for exploring concepts related to *(M,R)* systems.

$$\mathrm{S} + \mathrm{T} \xrightarrow{SU} \mathrm{ST} \quad (3)$$

$$\mathrm{S} + \mathrm{U} \xrightarrow{STU} \mathrm{SU} \quad (4)$$

$$\mathrm{ST} + \mathrm{U} \xrightarrow{SU} \mathrm{STU} \quad (5)$$

Then, treating every molecule as a function we have:

$$\mathrm{SU}(\mathrm{S},\mathrm{T}) = \{\mathrm{ST}\}$$
$$\mathrm{STU}(\mathrm{S},\mathrm{U},\mathrm{T}) = \{\mathrm{SU}\}$$
$$\mathrm{U}(\mathrm{S},\mathrm{T},\mathrm{U},\mathrm{ST},\mathrm{STU}) = \varnothing$$
$$\ldots$$

The last equation means that molecule U cannot transform the given mixture, because U is not a catalyst in the given metabolism. That said, we shall now analyze how concepts like f, Φ and β can be expressed with these ideas.

Metabolism: f

One of the basic equations in Rosen's model is $f(a) = b$, in which a represents the input materials (food set) needed by the organism to produce the complete set of metabolites and enzymes (b), i.e., every molecule reachable by the metabolism. Therefore, the function f is related to the notion of closure ($cl_{\mathscr{R}'}(X')$). To be able to define f in our terms, let us define function *expand*.

$$expand_X(X') = X' \cup \bigcup_{x_i \in X} x_i(X') \quad (6)$$

Moreover, let us define how a molecule set (X') can be applied to another molecule set (Y').

$$\overrightarrow{X'}(Y') = \begin{cases} Y' \text{ if } expand_{X'}(Y') = Y', \\ \overrightarrow{X'}(expand_{X'}(Y')) \text{ otherwise} \end{cases} \quad (7)$$

Thus, we use a molecular set as a function (distinguished from regular molecular set by a "semi-arrow") by repeatedly applying *expand* until no further additions occur. With these two last definitions, for any given catalytic reaction system $L = (X, \mathscr{R}, C)$, $f(a)$ can be defined as:

$$f(a) = \overrightarrow{catalysts(C)}(a) = b \quad (8)$$

where *catalysts* is a function that returns every catalyst in the given catalyzation set C ($catalysts(C) = \{x : (x,r) \in C\}$). The function *catalysts* is not required, as non-catalyst molecules do not modify the result. But it is used here as Rosen's formalism considers only catalysts as the core components of the metabolism.

Replacement: Φ

The formulation of Φ under RAF sets is more elaborate as we need to generate a function that using b as an input returns function f. The basic idea is to create mathematical objects that somehow keep track of which catalysts are produced and how these are created as a result of the metabolism. To begin we introduce operator Op. This operator returns the subset of molecules $X'' \subseteq X'$ that can act as catalysts upon the molecules in X' (the given molecule set).

$$Op(X') = \{x \in X' : x(X') \neq \varnothing\}$$

Then, for any given catalytic reaction system $L = (X, \mathscr{R}, C)$ over a food source F, $\Phi(b)$ will be defined as

$$\Phi(b) = \overrightarrow{Op(cl_{\mathscr{R}}(b) \cup F)} = f' \quad (9)$$

where $cl_{\mathscr{R}}(b)$ is the closure of b relative to the reaction set $\mathscr{R}$ as defined above. Therefore, Φ returns the catalyst set that are reachable from b as a function (f'), because the "semi-arrow" over the expression transforms the resulting set into a function. Thus, f' is operationally equivalent to function f.

Organizational invariance: β

Finally, it remains to define β, which should take the metabolism f as input and give us the replacement system Φ. The function β receives a hypothetical metabolism f' in the form of a function, thus our first step will be to find which catalysts can be related to that function f'. For that purpose, let us define the function ν that given a molecular set b and a function f', returns every reaction catalyzed by molecules in b, which produces part of the result of f' applied to F.

$$\nu(b, f', F) = \{r : \gamma(b \cup F, r) = 1\}$$

By using a new function μ, we filter out those reactions that cannot take place given the molecule set of interest ($b \cup F$).

$$\mu(b, f', F) = \{r \in \nu(b, f', F) : \rho(r) \subseteq b \cup F\} \quad (10)$$

This equation gives the reactions that are related to f', therefore β can be defined. For simplicity we shall define it as applied to a molecular set b.

$$\beta(f')(b) = \overrightarrow{Op(cl_{\mu(b,f',F)}(b) \cup F)} \quad (11)$$

This formula is similar to that of Φ, the main difference being that it uses function μ to obtain $\mathscr{R}$ instead of using $\mathscr{R}$ directly. In this way β returns a function that, used in an *(M,R)* system, would relate unequivocally to Φ.

Conclusion

A formidable challenge for using *(M,R)* systems as a framework for modeling biological systems has been the lack of operational definitions for the important functions f, Φ and β. Here we have presented various definitions for those functions that can be used for any catalytic reaction system.

An important unresolved matter is to make explicit how Rosen's equations can be fulfilled using concepts and definitions imported from RAF sets. Suppose that a given molecule set X and reaction set R compose an (M,R) system, how can that be proved using RAF-derived functions? First, let us distinguish a particular subset a of X, which contains every molecule that is not a product or a catalyst for any reaction. Then, we can write:

$$f(a) = b$$

This signifies "let the molecular system evolve until no further novelty can be produced". Now, we should expect that using the produced molecules as function will have the same effect as using f. In our terms, that means:

$$\Phi(b)(a) = b$$

This has the important consequence that f becomes equivalent (operationally) to $\Phi(b)$ in this molecular system.

β, as introduced here, does not explain Rosen's basic result ($\beta(f) = \Phi$, which means that Φ is uniquely determined by f). The definition of β and all associated formulae cannot explain Rosen's result, they merely serve as formal language that could help us to operate on modern metabolic data using Rosen's viewpoint.

Since the beginning of the 21st century there has been a resurgence of interest in the work of Robert Rosen, but it is not easy to understand and it is not apparent how to advance in a theory full of powerful but often obscure ideas (Letelier et al., 2006). Many attempts have been made to find the route to be followed in developing the theory (Wolkenhauer and Hofmeyr, 2007). Here we apply another formalism (RAF sets) that could be useful for clarifying the nature and properties of the operators f, Φ and β.

Finally, we have the caveat that living systems are not mere "soups of letters", and their complex properties are due to more than some combinatorics among molecules. It is apparent that to advance in our understanding of living organisms, it will be necessary to include further considerations into our current theory. These could be geometrical, thermodynamical, topological, or even merely historical, that is, relative to how life has come into existence, and later evolved here on Earth.

The RAF formalism may usher in an era in which the theory of *(M,R)* systems will demand reasoning tools that begin to resemble category theory more and more... Rosen would be amused!

References

Cárdenas, M. L., Letelier, J. C., Gutierrez, C., Cornish-Bowden, A., and Soto-Andrade, J. (2010). Closure to efficient causation, computability and artificial life. *J. Theor. Biol.*, 263:79–92.

Cornish-Bowden, A., Cárdenas, M. L., Letelier, J. C., and Soto-Andrade, J. (2007). Beyond reductionism: metabolic circularity as a guiding vision for a real biology of systems. *Proteomics*, 7:839–845.

Eigen, M. and Schuster, P. (1977). A principle of natural self-organization. *Naturwissenschaften*, 64(11):541–565.

Gánti, T. (1975). Organization of chemical reactions into dividing and metabolizing units: the chemotons. *Bio Systems*, 7(1):15–21.

Hordijk, W. and Steel, M. (2004). Detecting autocatalytic, self-sustaining sets in chemical reaction systems. *J. Theor. Biol.*, 227:451–461.

Kauffman, S. (1993). *The Origins of Order*. Oxford University Press, New York.

Kauffman, S. A. (1986). Autocatalytic sets of proteins. *J. Theor. Biol.*, 119(1):1–24.

Letelier, J. C., Marín, G., and Mpodozis, J. (2003). Autopoietic and (*M,R*) systems. *J. Theor. Biol.*, 222:261–272.

Letelier, J. C., Soto-Andrade, J., Guíñez Abarzúa, F., Cornish-Bowden, A., and Cárdenas, M. L. (2006). Organizational invariance and metabolic closure: analysis in terms of (*M,R*) systems. *J. Theor. Biol.*, 238:949–961.

Maturana, H. and Varela, F. (1980). *Autopoiesis and Cognition: the Realisation of the Living*. D. Reidel Publishing Company, Dordrecht, The Netherlands.

Mossel, E. and Steel, M. (2005). Random biochemical networks: the probability of self-sustaining autocatalysis. *J. Theor. Biol.*, 233(3):327–336.

Rosen, R. (1958). A relational theory of biological systems. *Bull. Math. Biophys.*, 20:245–260.

Wolkenhauer, O. and Hofmeyr, J.-H. (2007). An abstract cell model that describes the self-organization of cell function in living systems. *J. Theor. Biol.*, 246:461–476.

An Artificial Life Model for Carcinogenesis

Alessandro Fontana[1]

[1]IEEE
alessandro.fontana@ieee.org

Abstract

"Epigenetic Tracking" is the name of a model of cellular development that, coupled with an evolutionary technique, becomes an evo-devo method to generate arbitrary 2d or 3d shapes. The method evolves instructions contained in the genome inside cells, which guide the development of an artificial zygote into a mature phenotype: as such it belongs to the field of "artificial embryology", or "computational development". *In silico* experiments have proved its effectiveness in developing shapes of any kind and complexity, establishing its potential to generate the complexity typical of biological systems. Furthermore, it has also been shown how the underlying model of development is able to produce the artificial version of key biological phenomena such as embryogenesis, "junk DNA", and ageing. In this paper we show how malfunctions in the model lead to a phenomenon that can be considered the artificial equivalent of the process of carcinogenesis, which is explored through a simulation and analysed for two categories of tumours, teratomas and all other tumours, a distinction that emerges naturally from the framework.

Introduction

The previous work in the field of Artificial Embryology can be divided into two broad categories: the grammatical approach and the cell chemistry approach. In the grammatical approach development is guided by sets of grammatical rewrite rules; context-free or context-sensitive grammars, instruction trees or directed graphs (in place of actual grammars) can be used. L-systems were first introduced by Lindenmayer (Lindenmayer, 1968) to describe the complex fractal patterns observed in the structure of trees. The cell chemistry approach draws inspiration from the early work of Turing (Turing, 1952), who introduced reaction and diffusion equations to explain the striped patterns observed in nature (e.g. shells and animals' fur). This approach attempts to simulate cell biology at a deeper level, going inside cells and reconstructing the dynamics of chemical reactions and the networks of chemical signals exchanged between cells. Notable examples of grammatical embryogenies are (Lindenmayer, 1968) and (Gruau et al., 1996); among cell chemistry embryogenies, we recall (Kauffman, 1969) and (Bongard and Pfeifer, 2001).

"Epigenetic Tracking" (E.T.), first described in (Fontana, 2008), is the name of a model of cellular development that, coupled with an evolutionary technique, becomes an evo-devo method to generate arbitrary 2d or 3d shapes. The method evolves instructions contained in the genome inside cells, which guide the development of an artificial zygote into a mature phenotype; *in silico* experiments have proved its effectiveness in developing shapes of any kind and complexity (e.g. number of cells, number of colours, etc.), establishig its potential to generate the complexity typical of biological systems. Furthermore, it has also been shown how the underlying model of development is able to produce the artificial version of key biological phenomena such as embryogenesis, the presence of "junk DNA" and the phenomenon of ageing. The objective of this document is to use E.T. to explore another key topic in biology: the process of carcinogenesis. The rest of this document is organised as follows: section 2 provides a concise description of the model, section 3 gives a brief overview of the biological implications already analysed in previous work and outlines the main facts about carcinogenesis, sections 4 and 5 deal with artificial carcinogenesis, section 6 discusses the results and section 7 draws the conclusions.

The Model of Development

Shapes are composed of cells deployed on a grid; development starts with a cell (zygote) placed in the middle of the grid and unfolds in N age steps, counted by the variable "Age Step" (AS), which is shared by all cells and can be considered the "global clock" of the organism. Cells belong to two distinct categories: "normal" cells, which make up the bulk of the shape and "driver" cells, which are much fewer in number (typical value is one driver each 100 normal cells) and are evenly distributed in the shape volume. Driver cells have a Genome (an array of "instructions", composed of a left part and a right part) and a variable called cellular epigenetic type (CET, an array of integers). While the Genome is identical for all driver cells, the CET value is different in each driver cell; in this way, it can be used by different driver cells as a "key" to activate different instructions in the

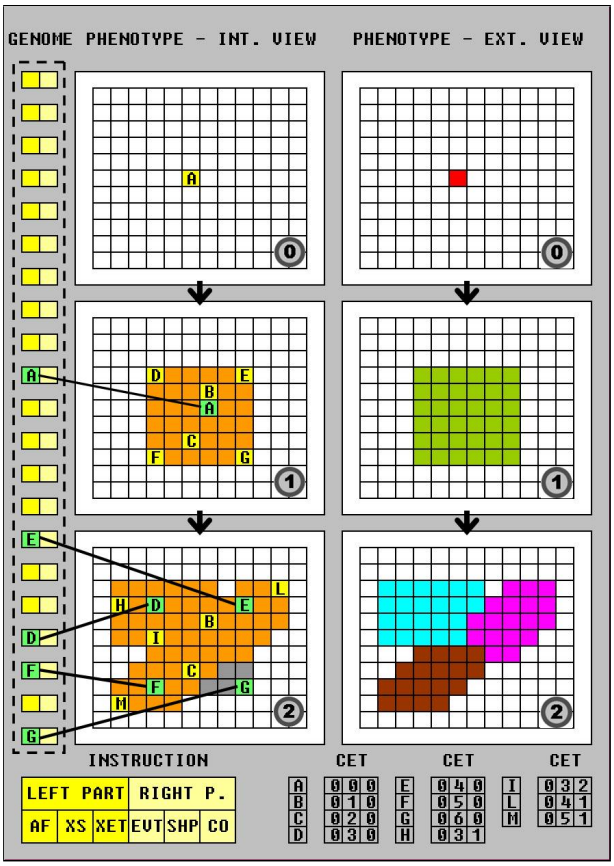


Figure 1: Example of development in three steps (AS=0,1,2) driven by five instructions: a proliferation triggered in step 1 on driver cell labelled with A, three proliferations triggered in step 2 on driver cells labelled with D, E and F and an apoptosis triggered in step 2 on driver cell labelled with G. Internal view on the left, external view on the right.

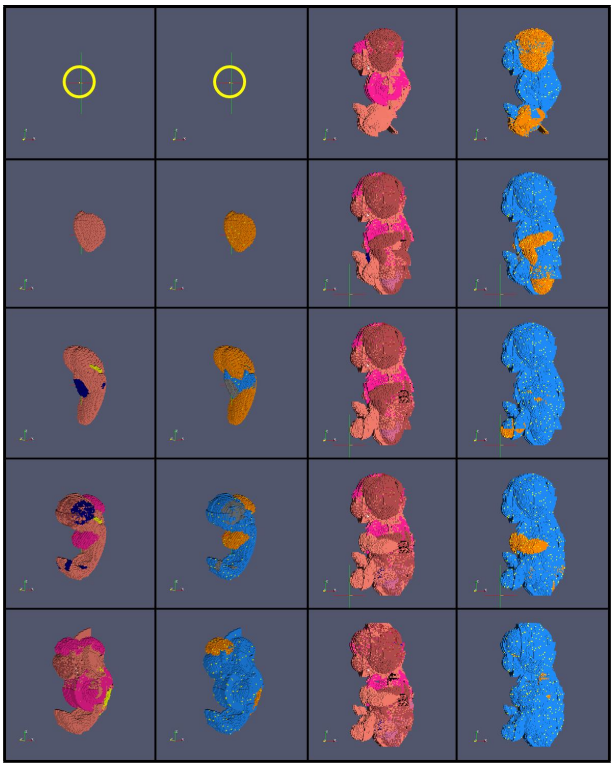


Figure 2: Development of an artificial human embryo of 200000 cells from a single cell (circled in yellow), generated with a Genome composed of 300 instructions, evolved in 40000 generations.

Genome. The CET value represents the source of differentiation during development, allowing driver cells to behave differently despite sharing the same Genome. A shape can be "viewed" in two ways: in "external view" cells are shown with their colours; in "internal view" colours represent cell properties: blue is used for normal cells alive, orange for normal cells just (i.e. in the current age step) created, grey for cells that have just died, yellow for driver cells (regardless of when they have been created).

An instruction's left part is composed of the following elements: an activation flag (AF), indicating whether the instruction is active or not; a variable called XET, of the same type as CET; a variable called XS, of the same type as AS. At each step, for each instruction and for each driver cell, the algorithm tests if the instruction's XET matches the driver's CET and if the instruction's XS matches AS. In practise, XS behaves like a timer, which makes the instruction activation wait until the clock reaches a certain value. If a match occurs, it triggers the execution of the instruction's right part, which codes for three things: event type, shape and colour. Instructions give rise to two 'types' of events: "proliferation instructions" cause the matching driver cell (called "mother cell") to proliferate in the volume around it (called "change volume"), "apoptosis instructions" cause cells in the change volume to be deleted from the grid; the parameter 'shape' specifies the shape of the change volume, in which the proliferation/apoptosis events occur, choosing from a number of basic shapes called "shaping primitives"; in case of proliferation, the parameter 'colour' specifies the colour of the new cells.

Always in case of proliferation, both normal cells and driver cells are created: normal cells fill the change volume, driver cells are "sprinkled" uniformly in the change volume. To each new driver cell a new, previously unseen and unique CET value is assigned, obtained by starting from the mother's CET value (the array [0,0,0] in the figure, labelled with A) and adding 1 to the value held in the ith array position at each new assignment (i is the current value of the AS counter); with reference to the figure, the new driver cells are assigned the values [0,1,0],[0,2,0],[0,3,0], ... , labelled with B,C,D, etc. (please note that labels are just used

in the figures for visualisation purposes, but all operations are made on the underlying arrays). In practise a proliferation event does two things: first creates new normal cells and sends them down a differentiation path (represented by the colour); then creates other driver cells, one of which can become the centre of another event of proliferation or apoptosis, if in the Genome an instruction appears, whose XET matches such value. This mechanism constitutes the "core" of the machine: a CET value produces a change event, which in turn produces other CET values, some of which produce other change events and so on, *in an indefinitely sustainable way*. Figure 1 reports a simple hand-coded example of development.

It may happen that the change volume is not empty; in this case the most realistic and physically plausible behaviour would be one in which the newly created cells push the existing cells outwards, which in turn would push other cells located in more external positions and so forth, until the moved cells find empty positions to settle without having to displace other cells. Since this approach has the drawback of involving the movement of most cells of the shape, being thus computationally demanding, a different solution has been undertaken. It consists of a procedure called "remove-redeploy" that, as the name implies, removes cells present in the volume before proliferatio, stores them in a temporary buffer and redeploys them back onto the grid after proliferation has occurred. The remred procedure plays the role of "physics", i.e. the set of rules by which cells are moved around and find their final position in the shape; based on our experience, the choice of the particular physics implemented has little impact on the effectiveness of the method, as long as physics behave predictably and consistently, as we all expect. This thanks to the distribution of driver cells throughout the shape, that enables the model of development to bend any kind of physics to its goals, keeping the shape plastic during development.

The model of development described, coupled with a standard evolutionary technique, becomes an evo-devo method to generate arbitrarily shaped 2d or 3d cellular sets. The method evolves a population of Genomes that guide the development of the shape starting from a small number of zygotes (usually one) initially present on the grid, for a number of generations; at each generation development is let unfold for each Genome and, at the end of it, adherence of the shape to the target shape is employed as fitness measure. *In silico* experiments have proved the effectiveness of the method in devo-evolving any kind of shape, of any complexity (in terms e.g. of number of cells, number of colours, etc.); figure 2 shows the development of an artificial human embryo, produced by a Genome composed of 300 instructions, evolved in 40.000 generations.

The effectiveness of the method is to be reconducted to four features of the model of development. The first key feature is the distinction between normal cells and driver cells; the latter represent the backbone of the developing shape and make it possible to steer development acting on a small subset of cells. The second feature is the implementation of the change events of proliferation and apoptosis in such a way that they create/delete many cells at once (instead of one). This increases the power of the single change event and allows a reduction of the number of change instructions needed to generate a given shape, speeding up the morphogenetic process. The third feature is the explicit presence of an epigenetic memory, i.e. a cell variable (the CET, only present in driver cells) that takes different values in different cells and represents the source of differentiation during development, leading different cells at different times to executing different portions of the Genome. The fourth feature is the mechanism of assignment of the CET values on the newly generated driver cells during a proliferation event, which ensures that each new driver cell is assigned a new, previously unseen CET value; the CET value represents the link by which these driver cells in subsequent steps can be picked up by the Genome and given other instructions to be executed.

Biological Implications

Embryogenesis. The interpretation of Epigenetic Tracking as a model of morphogenesis and cell differentiation is straightforward (the process of natural morphogenesis corresponds to the process of artificial morphogenesis, in which different cells types are represented by different colours); in this perspective, driver cells take the role of embryonic stem cells and have also much in common with the concept of Spemann's organiser. The Genome corresponds to the natural genome, while the cell epigenetic type (CET) corresponds to cellular epigenetic memory, representing in both the natural and the artificial world the portion of information which is different from cell to cell and, as such, constitutes the key ingredient necessary for cellular differentiation. A key difference is that, while embryonic stem cells are thought to be present only in the embryo, driver cells are present, evenly distributed throughout the body, for the entire duration of the organism's life.

Junk DNA. In molecular biology "junk DNA" is a collective label for the portions of the DNA sequence of a genome for which no function has been identified. In E.T., at any moment in the course of evolution, the set of driver cells/CET values generated during an individual's development can be divided into i) driver cells that activate an instruction during development and ii) driver cells that *do not* activate any instruction during development; in the same way the individual's Genome is composed by i) instructions that become active during development and by ii) instructions that *do not* become active during development. By analogy with real genomes, elements in the two categories labelled with ii) can be defined as "junk" driver cells and "junk" instructions respectively. The presence of junk information in both the set

of driver cells and the Genome was shown to be inescapably connected to the core of the Epigenetic Tracking machine, a requirement essential to its *evolvability*.

Ageing. As we said, at the end of an individual's development many junk driver cells are present, as well as many junk instructions; such stock of junk represents a reservoir of events that can potentially be triggered after the moment of fitness evaluation (in what can be called the period of "artificial ageing"). Since these events occur after fitness evaluation, they are by definition not affecting the fitness value; for this reason they will tend to have a random nature and their overall effect on the phenotype is more likely to be detrimental than beneficial: they can be thought of as a random noise superimposed on the phenotype created by the instructions subject to evolutionary pressure. In this perspective, the presence of a big stock of junk mediates both a species's evolvability and its susceptibility to ageing, which appear to be two sides (one good and one bad) of the same coin.

Carcinogenesis is the process by which normal cells are transformed into cancer cells. The standard theory of carcinogenesis states that carcinogenesis is a multi-step process that can take place in any cell, driven by damage (mutations) to genes (onco-genes and tumour-suppressor genes) that normally regulate cell proliferation, which in turn upsets the normal balance between cell proliferation and cell death and results in uncontrolled cell division and tumour formation. A few cancer-related genes, such as p53, do seem to be mutated in the majority of tumours, but many other cancer genes are changed in only a small fraction of cancer types, a minority of patients, or a subset of cells within a tumour; moreover, some of the most commonly altered cancer genes have inconsistent effects; for instance the oncogenes c-fosand c-erbb3 are strangely less active in tumours than they are in nearby normal tissues; the tumour suppressor gene rb was recently shown to be hyperactive -not disabled- in some colon cancers (Gibbs, 2003). In conclusion, the attempt to reconduct tumour formation to a subset of mutated genes, consistently found in all tumours, has so far been unsuccessful.

A more recent theory differentiates from the standard theory in tracing back the origin, the maintenance and the spread of a tumour to a relatively small subpopulation of cells called **cancer stem cells** (CSCs), whereas the bulk of the tumour would actually be composed of non-tumorigenic cells that, deprived of the cancer stem cells, would quickly shrink and disappear. CSCs possess characteristics associated with normal stem cells, specifically the ability to give rise to all cell types found in a particular cancer sample; CSCs may generate tumours through the stem cell processes of self-renewal and differentiation into multiple cell types. The implications of this hypothesis for therapy cannot be overstated: conventional chemotherapies kill differentiated or differentiating cells, which form the bulk of the tumor but are unable to generate new cells; a population of CSCs, which gave rise to it, could remain untouched and cause a relapse of the disease.

Mathematical models of cancer -see (Wodarz and Komarova, 2006) for a comprehensive review- have found application in three major areas: i) modelling in the context of epidemiology and other statistical data; ii) mechanistic modelling of avascular and vascular tumour growth (including physical properties of biological tissues); iii) modelling of cancer initiation and progression; basic mathematical tools used are ordinary differential equations, partial differential equations, stochatic processes, cellular automata and agent-based models. To our knowledge, most mathematical models stick to the standard theory, are based on differential equations and have the primary objective of explaining the dynamics of tumour growth, i.e. they try to answer to "how fast" tumours grow; our approach, instead, seeks to explain the mechanism of tumour formation from the very beginning.

Artificial Carcinogenesis I: Teratomas

In this section we will analyse a possibile malfunction of the model of cellular growth described in section 2 and we will show how such malfunction gives origin to a phenomenon that can be considered the artificial equivalent of carcinogenesis, with reference to a particular kind of tumour called teratoma. In the Epigenetic Tracking framework, a certain body part of an artificial organism is generated by a single driver cell that, once activated, proliferates, generating other driver cells, some of which in turn get activated, proliferating and generating other driver cells etc. (the same holds true for the entire organism). This process presupposes that each driver cell, at the moment of activation, find itself in the right position: only in this case is the cascade of events capable, along with physics, of generating the relevant body part. This delicate mechanism can be perturbed by both genetic mutations (affecting the Genome) and epigenetic alterations (affecting a driver cell's CET value). We will now focus our attention on a case characterised by an epigenetic mutation that, at step AS(J), turns the CET value (J) of a certain driver cell C(J), positioned at point P(J), into another CET value (K); if CET value K is not generated during normal development, or if it is generated but never activated, nothing happens.

If, on the contrary, CET value K does get activated during normal development to produce a certain body part -say at step AS(K), when cell C(K) finds itself at point P(K)- as a result of the mutation the cascade of events destined to give rise to such body part will start from both point P(K) at step AS(K) -right place and moment- and point P(J) at step AS(J) -ectopic place, wrong moment-. Being activated in the wrong place and moment, cell C(J) is not surrounded by the right micro-environment: as a result, the cascade of events originating from C(J) will only manage to mimic the development of the relevant body part in a grotesque fashion. Fig-

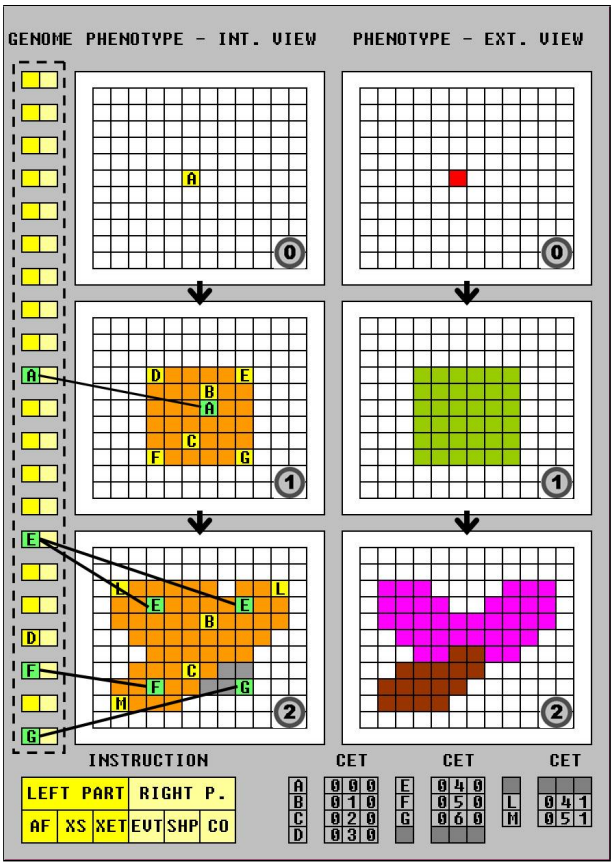


Figure 3: Example of artificial teratoma. In step 2 the driver cell bearing the CET value D is hit by an epigenetic mutation, that turns D into E. As a result, the cell starts behaving like the one bearing CET value E, triggering an arrow-shaped fuchsia proliferation, generating CET values, that can in turn trigger other proliferations, etc.

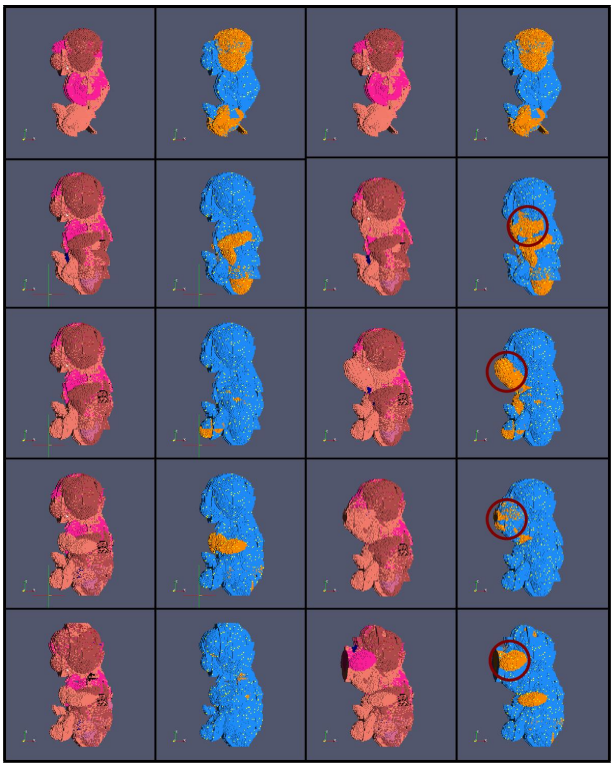


Figure 4: On the right: simulation of an artificial teratoma. In step 6 the CET value belonging to the driver cell circled in red is turned into the CET value of the zygote: as a consequence the development of the whole embryo starts over from the point indicated, producing a shapeless mass of cells in the neck region, composed of differentiated cells. On the left the normal development sequence for comparison.

ure 4 provides a hand-coded example of artificial teratoma, occurring to the shape whose development is shown in figure 1: in step 1 a mutation turns CET value D into CET value E: as a result, the same arrow-shaped fuchsia structure produced in the north-east part of the shape by CET value E is also produced in the north-west part, in place of the rectangle-shaped light blue structure produced by D (see figure 1); if some of the CET values produced by the proliferation originated from E trigger in turn other poliferation events, such events will occur both in the north-east and in the north-west part of the shape. The outcome of this scenario is an uncontrolled proliferation with a self-sustaining nature of limited duration (after a given number of steps, both sequences halt, as development does not go on forever).

A possible biological counterpart of this scenario is teratoma, a tumour with tissue or organ components resembling normal derivatives of all three germ layers. The tissues of a teratoma, although normal in themselves, may be quite different from surrounding tissues, and may be highly inappropriate, even grotesque: teratomas have been reported to contain hair, teeth, bone and very rarely more complex organs such as eyeball, torso, and hand; usually, however, a teratoma does not contain organs but rather tissues normally found in organs such as the brain, liver, and lung. Teratomas are thought to be present at birth, but small ones often are only discovered much later in life. Fetus in fetu is a rare form of teratoma that resembles a malformed fetus (it may appear to contain complete organ systems, even major body parts such as torso or limbs).

Figure 4 shows a simulation of an artificial teratoma, occurring to the artificial embryo shown in figure 2. In step 6, the CET value (J) of the driver cell marked with the circle (C(J)) is mutated into the CET value of the zygote (K) (hence AS(J)=6 and AS(K)=1); as a result, the development of the whole embryo starts over again from cell C(J): the cell proliferates, generating other CET values some of which, as occurred in normal development, trigger other proliferation events and so on. But, since in this case the zygote and all

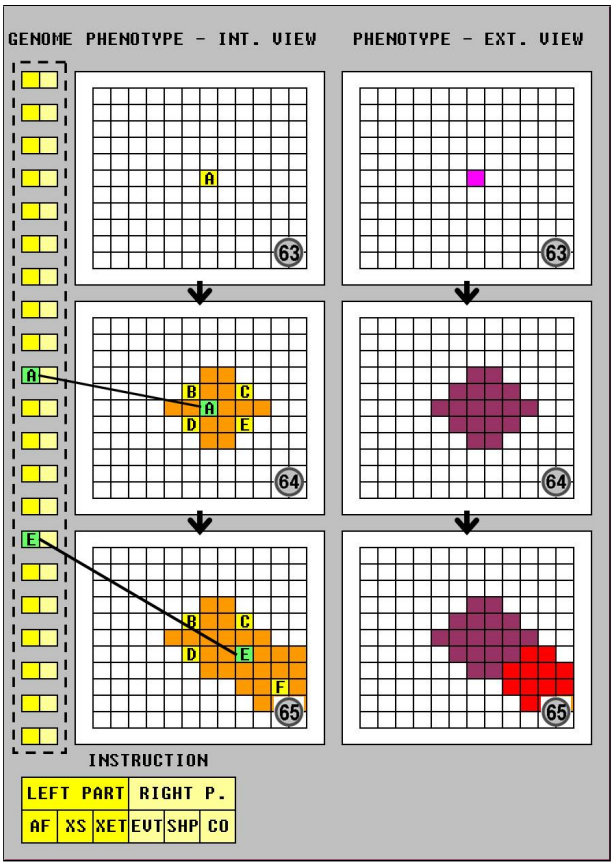


Figure 5: Ageing-related proliferation in a driver cell (the event is triggered during the ageing period). After step 2 the CET values generated do not trigger further events and the proliferation halts. The effects contribute to the ageing phenotype.

other CET values cascaded from it are in ectopic positions and are surrounded by wrong environments, while the different cell types (represented by different colours) continue to be created, the interactions with other cells -mediated by physics- prevent them from being arranged in the correct patterns; instead, an amorphous mass of differentiated cells is produced. The kind of epigenetic mutation reported in this simulation is only one among endless possibilities; another possible path leading to an (artificial) teratoma is the following: the CET value belonging to a driver cell of the developing (artificial) liver is turned into the CET value of a driver cell which in normal development is a precursor of the (artificial) hand; as a result, the mutated driver cell will try to generate the hand, etc. It is quite natural to hypothesise a direct link between the size of a teratoma and the depth of the tree of CET values at which the mutation occurs (the closer the latter is to the level of the zygote, the bigger the tumour).

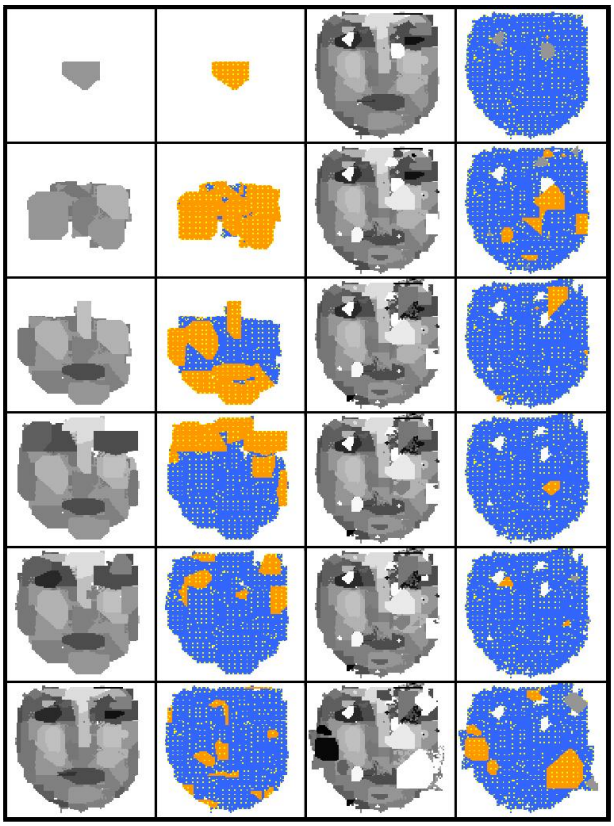


Figure 6: The "face". On the left the period of development (steps 0-5): the shape grows from a single cell to the mature phenotype in step 5, fitness is evaluated; on the right the period of ageing (steps 6-11): the picture quality deteriores steadily under the action of random instructions.

Artificial Carcinogenesis II: Other Tumours

As recalled in section 3, at the end of an individual's development many junk driver cells are present, as well as many junk instructions; such stock of junk represents a reservoir of events that can potentially be triggered after the moment of fitness evaluation, in the artificial ageing period. Since these events occur after fitness evaluation, they are by definition not affecting the fitness value; for this reason they will tend to have a random nature and their effects on the overall individual's fitness are more likely to be detrimental than beneficial: they can be thought of as a random noise superimposed on the phenotype created by the instructions subject to evolutionary pressure. An example is reported in figure 5: driver cell bearing CET value A triggers the activation of a proliferation instruction at step 64 (beyond fitness evaluation); at the subsequent step another proliferation is triggered on the driver cell bearing CET value E. Such random events represent indeed the essence of artificial ageing.

A simulation of artificial ageing is reported in figure 6 for a bi-dimensional "face" shape (picture of 100x100 size

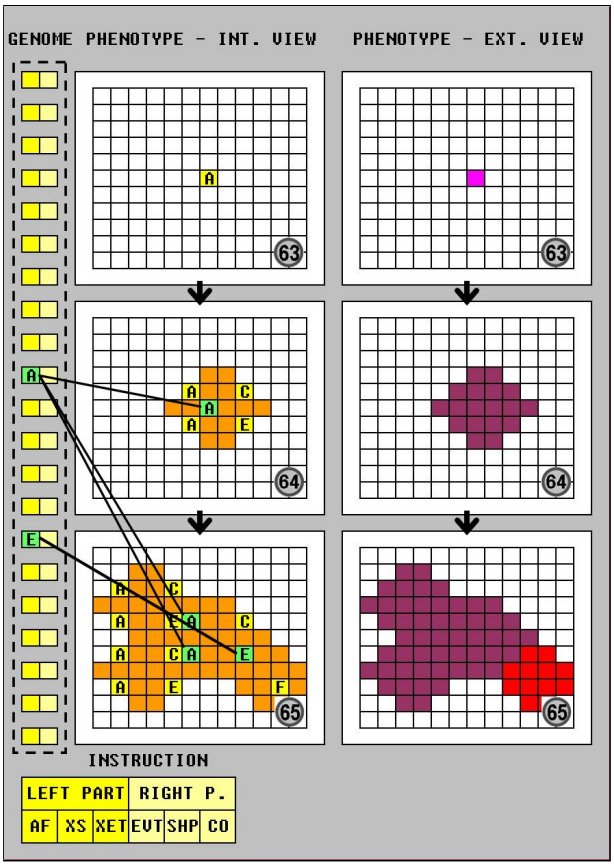


Figure 7: Tumorigenic proliferation in a driver cell. A damage is the CET generating mechanism has the effect of replacing CET values B and D with additional copies of A, which in turn trigger another proliferation in the subsequent steps. The amount of purple cells expands without limit.

with 16 grey shades); the left part shows steps 0-5, belonging to the period of development: the shape grows from the single cell stage to the mature phenotype in step 5, when fitness is evaluated; the right sequence refers to the period of ageing (steps 6-11), characterised by the accumulation of random events (of the type of that of figure 5), whose global effect causes a progressive deterioration of the quality of the image. In nature the moment of fitness evaluation can be thought to coincide with the moment of reproduction, even though, actually, an individual's fitness depends also on characteristics manifesting themselves after reproduction, as also those can affect the survival chances of its progeny; in other words the effect of changes on the fitness tends to decrease as the age of their appearance increases, rather than going abruptly to zero right after reproduction.

Now, the stage for a dangerous scenario is set if a fault arises in one of such "ageing" driver cells, affecting the mechanism used by the cell to generate new CET values during a proliferation event. Within this scenario many variants are conceivable (this mechanism can be damaged in many ways): in one possible variant the damage can be such that the CET value A (the mother's) appears among the CET values of the daughter cells, in one or more copies. Figure 7 shows the effect of such a damage on the same event of figure 5: CET values B and D have been replaced with CET value A: in this context the mother cell and its epigenetically identical progeny are stuck to execute the same proliferation instruction, leading to a situation in which the amount of purple cells tends to increase without limit. Along with the purple cells, also cells of a different type (in this case the red cells) may be present, leading to a heterogenous mix of cell types.

Discussion

The process of carcinogenesis is traditionally divided into three phases: initiation, promotion and progression. Initiation is linked to chemicals or physical stimuli that induce permanent alterations to DNA; a single exposure appears to be sufficient for the establishment of the initiated phenotype which, once in place, is irreversible. An initiated cell is susceptible to the effects of promoters; these compounds favour the proliferation of the cell, giving rise to a large number of daughter cells containing the mutation created by the initiator (if the cell has not been previously initiated promoters have no effect). The third stage, progression, refers to the stepwise transformation of a benign tumour into a malignant one (this framework is based on skin cancer studies, but it is thought to be valid for most tumour types).

As we said, the attempt to trace back carcinogenesis to a subset of mutated genes (oncogenes and tumour-suppressor -TS- genes) consistently found in all tumours, has so far been unsuccessful. Nevertheless, most tumours are undeniably correlated with specific patterns of mutations, affecting specific genes involved in cell-cycle regulation and cellular differentiation; individual genes are mutated in percentages that are tumour-specific, e.g. the rb gene is mutated in 50% of colorectal cancers, in 30% of adenocarcinomas, etc.: these correlations represent evidence a theory of carcinogenesis should seek to explain. According to current knowledge, TS genes are thought to act as checkpoints at some cell-cycle key moments, when they can stop the cycle upon detection of damages to DNA; oncogenes, on the other hand, are genes implicated in the cascade of chemical signals that drive the cell towards mitosis. While the supposed role of oncogenes appears to be realistic, the role of TS genes as "guardians of the genome" is, in our opinion, less firmly grounded; moreover, if they played this role, they should be mutated in 100% of cancers.

The hypothesis we wish to put forward here is that the cellular equipment dedicated to the generation of new CET values, which in our model is embedded in the cell structure, in real cells is implemented by means of TS genes; in other words, the CET values would be determined by the

interplay of the product of TS genes. In the light of this hypothesis, it is not surprising to find that the set of TS genes is tissue-specific, as it is the set of CET values dedicated to the differentiation of different tissues (the set of CET values needed to induce the differentiation of skin progenitor cells is different from the set of CET values needed to induce the differentiation of gut progenitor cells, for instance). This would explain why the set of mutated TS genes is different in different tumours, a fact that the "genome guardian" hypothesis is unable to account for. In the E.T. framework, the damage to the CET generation mechanism corresponds to initiation, a situation in which the number of CET values in the progeny which are equal to the CET value of the mother is altered.

The subsequent phase of promotion sets in once the conditions required for proliferation are met (if the cell does not proliferate, the effects of the damage to the CET generating mechanism do not become apparent, even if present). The progression phase corresponds to the drive towards the malignant phenotype, caused by mutations occurring to oncogenes (not included in the model's current version), which confer additional powers to the already transformed cells, e.g. the capacity to infiltrate tissues and to produce metastases. The presence in tumours of cells having different degrees of differentiation is a well documented phenomenon, coherent with the cancer stem cell theory and more difficult to explain with the standard theory (that postulates that tumour cells are clones of the cell originally affected by a number of mutations); this is a fact that, as we have seen, is easily accounted for by our model.

The proposed theory provides also a quite straightforward explanation for another well-known fact about cancer: the prevalence increasing with the age. The temporal patterns of ageing and cancer appear indeed to be perfectly superimposed: cancer is a rare occurrence in the young and becomes more and more common as the age progresses. This fact is easily accounted for by our theory, which hypothesises that the same events triggered in the artificial ageing period can contribute to the ageing phenomenon (if the CET generating machinery is intact) or give rise to a tumour (if the CET generating machinery is damaged). This can also explain the long latency observed between the exposure to mutagenic chemicals (e.g. tobacco smoke) and the manifestation of the tumour (e.g. lung cancer). As a matter of fact, even if the damage to the driver cell's CET generating mechanism occurs early in life, for its effects to become manifest we need to wait until a proliferation event is triggered on the relevant cell: if the instruction's timer is set to 60 years of age, the tumour will not appear until that moment.

According to the theory proposed, tumours originate from the artificial equivalent of embryonic stem cells, which in our model are present throughout the body for the entire life of the organism; a similar phenomenon could also originate from the artificial equivalent of adult stem cells, which at present are not included in the model. In such "adult driver cells" the CET value of the mother would normally be present in the progeny (to guarantee the renewal of the stem pool), in such an amount to keep the system in equilibrium (the renewal of progenitor driver cells -the equivalent of those having CET value A- would be counterbalanced by the disapperance of as many driver cells that differentiate to perform their specialised job in the body). In a pathological scenario, a damage to the CET generation mechanism would be such that the amount of new "A cells" outweighs the amount of differentiating cells, leading to a situation in which "A cells" become prevalent. In conclusion, we can say that our model of development is able to provide an explanation for some basic evidence relevant to tumours and fits well with the cancer stem cell theory.

Conclusions

In the present work the model of cellular development called Epigenetic Tracking has been employed to explore carcinogenesis; in this context, we have been able to show how malfunctions of model can produce the artificial counterpart of the process of carcinogenesis, broken down into two broad categories: one containing just a single tumour type called teratoma and one with all other tumours. In previous works it was shown how the model is able to produce the artificial version of key biological phenomena such as junk DNA and ageing; the addition of carcinogenesis to the repertoire of cellular behaviours strengthens the susceptibility of the model to be used as a universal model of cellular development, that can be succesfully employed as a tool to exploring a wide range of biological phenomena.

References

Bongard, G. and Pfeifer, R. (2001). Repeated structure and dissociation of genotypic and phenotypic. complexity in artificial ontogeny. In *Proceedings of GECCO-2001*.

Fontana, A. (2008). Epigenetic tracking, a method to generate arbitrary shapes by using evo-devo techniques. In *Proceedings of EPIROB 2008*.

Gibbs, W. W. (2003). Untangling the roots of cancer. *Scientific American*.

Gruau, F., Whitley, D., and Pyeatt, L. (1996). A comparison between cellular encoding and direct encoding for genetic neural networks. In *Genetic Programming 1996*.

Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*.

Lindenmayer, A. (1968). Mathematical models for cellular interaction in development. *Journal of Theoretical Biology*.

Turing, A. (1952). The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society*.

Wodarz, D. and Komarova, N. L. (2006). *Computational biology of cancer*. World Scientific.

Towards a simpler photoautotrophic cell: conserved and variable genes in *Synechococcus elongatus*

Luis Delaye[1], Carmen M. González-Domenech[2, 3] and Andrés Moya[3]

[1] CINVESTAV-Irapuato, México
[2] Facultad de Farmacia, Universidad de Granada
[3] Instituto Cavanilles de Biodiversidad y Biologia Evolutiva, Universitat de València
Andres.Moya@uv.es

Abstract

Simpler biological systems should be easier to understand and engineer. One way to achieve biological simplicity is through genome minimization. Here we have looked for genomic islands in the fresh water cyanobacterium *Synechococcus elongatus* PCC 7942 that could be used as targets for deletion for genome minimization. By using a combination of methods we have identified 184 genes that have been horizontally transferred into the genome of *S. elongatus* plus 127 ORFans (Figure 1). These genes have a combination of: a) unusual G+C content; b) unusual phylogenetic similarity; and/or c) a small number of a highly iterated palindrome 1 (HIP1) motif plus an unusual codon usage. We have also corroborated the existence of the largest genomic island by its lack of coverage among metagenomic sequences from a fresh water microbialite. Interestingly, most genes coding for proteins with a diguanylate cyclase domain are predicted to be xenologous, suggesting a role for horizontal gene transfer in the evolution of sensory systems in this cyanobacteria. In parallel we have identified 1401 highly conserved genes that might be essential for cell survival and should not be deleted. These two datasets (variable and conserved genes) comprises ~11.8% and 53.6% of annotated genes in *S. elongatus*. Our results set a guide to non-essential genes in *S. elongatus* PCC 7942 indicating a path towards the engineering of a simpler photoautotrophic cell.

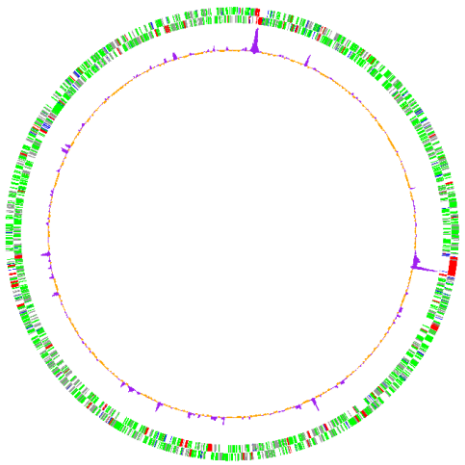


Figure 1. Conserved and variable regions in the genome of *S. elongatus* PCC 7942. Outer circle. Red: variable genes; green: conserved genes; gray: other. Inner circle. Regions of atypical tri-nucleotide composition are shown in purple.

An Agent-based Model for the Pattern Formation in Drosophila Melanogaster

Sara Montagna[1], Nicola Donati[1] and Andrea Omicini[1]
[1]ALMA MATER STUDIORUM–Università di Bologna
via Venezia 52, 47023 Cesena, Italy
sara.montagna@unibo.it, nicola.donati@studio.unibo.it, andrea.omicini@unibo.it

Abstract

The hierarchical organisation of biological systems plays a crucial role in the pattern formation of gene expression resulting from the morphogenetic processes. Being able to reproduce the systems dynamics at different levels of such a hierarchy might be very useful for studying such a complex phenomenon of self-organisation. In this paper we propose the adoption of the agent-based model as an approach capable of capture multi-level dynamics. We then realise an agent-based model of *Drosophila Melanogaster* morphogenesis demonstrating its capability of reproducing the expression pattern of the embryo.

Introduction

Developmental biology is an interesting branch of life science that studies the process by which organisms develop, focussing on the genetic control of cell growth, differentiation and movement. A main problem in developmental biology is understanding the mechanisms that make the process of vertebrates' embryo regionalisation so robust, making it possible that from one cell (the zygote) the organism evolves acquiring the same morphologies each time. This phenomenon involves at the same time the dynamics of – at least – two levels, including both cell-to-cell communication and intracellular phenomena: they work together, and influence each other in the formation of complex and elaborate patterns that are peculiar to the individual phenotype. This happens according to the principles of *downward* and *upward* causation, where the behaviour of the parts (down) is determined by the behaviour of the whole (up), and the emergent behaviour of the whole is determined by the behaviour of the parts (Uhrmacher et al., 2005).

Modelling embryo- and morphogenesis presents big challenges: *(i)* there is lack of biological understanding of how intracellular networks affect multicellular development and of rigourous methods for simplifying the correspondent biological complexity: this makes the definition of the model a very hard task; *(ii)* there is a significant lack of multi-level models of vertebrate development that capture spatial and temporal cell differentiation and the consequent heterogeneity in these four dimensions; *(iii)* on the computational framework side, there is the need of tools able to integrate and simulate dynamics at different hierarchical levels and spatial and temporal scales.

A central challenge in the field of developmental biology is to understand how mechanisms at intracellular and cellular level of the biological hierarchy interact to produce higher level phenomena, such as precise and robust patterns of gene expressions which clearly appear in the first stages of morphogenesis and develop later into different organs. How does local interaction among cells and inside cells give rise to the emergent self-organised patterns that are observable at the system level?

The above issues have already been addressed with different approaches, including mathematical and computational ones. Mathematical models, on the one side, are continuous, and use differential equations—in particular, partial differential equations describing how the concentration of molecules varies in time and space. A main example is the reaction-diffusion model developed by Turing, 1952 and applied to the *Drosophila Melanogaster* (*Drosophila* in short) development by Perkins et al., 2006. The main drawback of mathematical models is the inability of building multi-level models that could reproduce dynamics at different levels.

Computational models, on the other side, are discrete, and model individual entities of the system—cells, proteins, genes. The agent-based approach is an example of such a kind of models. Agent-based modelling (ABM) is a computational approach that can be used to explicitly model a set of entities with a complex internal behaviour and which interact with the others and with the environment generating an emergent behaviour representing the system dynamics. Some work has already been done which applies ABM in morphogenesis-like scenarios: a good review is proposed in Thorne et al., 2008. Most of these models generate artificial pattern – French and Japanese flags (Beurier et al., 2006) – realising bio-inspired models of multicellular development in order to obtain predefined spatial structures. At the best of our knowledge, however, few results have been obtained till now in the application of ABM for analysing real phenomena of morphogenesis.

In order to get the benefits of both approaches, hybrid frameworks has been developed. For instance, COMPUCELL 3D (Cickovski et al., 2005) combines discrete methods based on cellular-automata to model cell interactions and continuous model based on reaction-diffusion equation to model chemical diffusion. COMPUCELL 3D looks like a very promising framework whose main limitation is represented by the lack of a suitable model for cell internal behaviour—gene regulatory network in particular.

In this paper we present an agent-based model of the *Drosophila* embryo development, reproducing the gene regulatory network that causes the early (stripes-like) regionalisation of gene expression in the anteroposterior axis (Yamins and Nagpal, 2008; Perkins et al., 2006). The embryo is modelled as a set of agents, where each agent is a cell. Our approach allows the gene-regulatory network to be directly modelled as the internal behaviour of an agent, whose state reproduces the gene expression level and dynamically changes according to functions that implement the interactions among genes. It also allows the cell interacting capability mediated by morphogens to be modelled as the exchange of messages among agents that absorb and secrete – from and towards the environment – the molecules that are then able to diffuse over the environment.

The remainder of this paper is organised as follows: The role of hierarchy in the spatial self-organisation of gene expression during morphogenesis is first highlighted along with the main biochemical mechanisms taking place in this phenomenon. The agent-based approach is then presented with the modelling abstractions it provides. The third part describes the biological principles of *Drosophila* embryo development, while the fourth part reports the ABM we have developed and implemented. Simulation results are then discussed, followed by concluding remarks.

The Role of Hierarchy in Morphogenesis

Complex systems in general exhibit a hierarchical organisation that divide the system into levels composed by many interacting elements whose behaviour is not rigid, and is instead self-organised according to a continuous feedback between levels. Hierarchy has therefore a crucial role in the static and dynamic characteristics of the systems themselves. These properties are highly dependent by the principles of *downward* and *upward* causation, where the behaviour of the parts (down) is determined by the behaviour of the whole (up), and the emergent behaviour of the whole is determined by the behaviour of the part (Uhrmacher et al., 2005). An example is given by biological systems: an outstanding property of all life is the tendency to form multilevelled structures of systems within systems. Each of these forms a whole with respect to its parts, while at the same time being a part of a larger whole. Biological systems have different level of hierarchical organisation – *(1)* sequences; *(2)* molecules; *(3)* pathways (such as metabolic or signalling); *(4)* networks, collections of cross-interacting pathways; *(5)* cells; *(6)* tissues; *(7)* organs – and the constant interplay among these levels gives rise to their observed behaviour and structure. This interplay extends from the events that happen very slowly on a global scale right down to the most rapid events observed on a microscopic scale. A unique molecular event, like a mutation occurring in particularly fortuitous circumstances, can be amplified to the extent that it changes the course of evolution. In addition, all processes at the lower level of this hierarchy are restrained by and act in conformity to the laws of the higher level.

In this contest, an emblematic process is morphogenesis, which takes place at the beginning of the animal life and is responsible for the formation of the animal structure. Morphogenesis phenomena includes both cell-to-cell communication and intracellular dynamics: they work together, and influence each other in the formation of complex and elaborate patterns that are peculiar to the individual phenotype.

The biology of development

Animal life begins with the fertilisation of one egg. During the development, this cell undergoes mitotic division and cellular differentiation to produce many different cells. Each cell of an organism normally owns an identical genome; the differentiation among cells is then *not* due to different genetic information, but to a diverse gene expression in each cell. The set of genes expressed in a cell controls cell proliferation, specialisation, interactions and movement, and it hence corresponds to a specific cell behaviour and role in the entire embryo development.

One possible way for creating cells diversity during embryogenesis is to expose them to different environmental conditions, normally generated by signals from other cells, either by cell-to-cell contact, or mediated by cues that travel in the environment.

On the side of intracellular dynamics, signalling pathways and gene regulatory networks are the means to achieve cells diversity. Signalling pathways are the ways through which an external signal is converted into an information travelling inside the cell and, in most of the cases, affecting the expression of one or more target genes. The signalling pathways are activated as a consequence of the binding between *(i)* a cue in the environment and a receptor in the cell membrane, or *(ii)* two membrane proteins belonging to different cells. The binding causes the activation of the downstream proteins until a transcription factor that activates or inhibits the expression of target genes.

During embryo-morphogenesis few pathways are active. They work either as mutual inhibitors, or as mutual enhancers. The idea is that there are regions where the mutual enhancers are active and interact giving rise to positive feedbacks. Pathways active in different regions work probably as mutual inhibitors. There are then boundary regions where we can observe a gradient of activity of the different

sets of pathways, due to the inhibitory effect of the pathways belonging to neighbour regions.

The Agent-based Approach

In literature, agent-based systems – in particular Multi-Agent Systems (MAS) – are considered as an effective paradigm for modelling, understanding, and engineering *complex systems*, providing a basic set of high level abstractions that make be possible to directly capture and represent the main aspects of such complex systems, such as interaction, multiplicity and decentralisation of control, openness and dynamism (Michel et al., 2009; Merelli et al., 2007; Klügl et al., 2002). A MAS can be characterised by three key abstractions: *agents*, *societies* and *environment*. Agents are the basic *active* components of the systems, executing pro-actively and autonomously. Societies are formed by set of agents that interact and communicate with each other, exploiting and affecting the environment where they are situated. Such an environment plays a fundamental role, as a context enabling, mediating and constraining agent activities (Weyns et al., 2007).

By adopting an agent-based approach, biological systems can be modelled as a set of interacting autonomous components – i.e., as a set of agents –, whereas their chemical environment can be modelled by suitable agent environment abstractions, enabling and mediating agent interactions. In particular, MAS provide a direct way to model: *(i)* the individual structures and behaviours of different entities of the biological system as different agents (*heterogeneity*); *(ii)* the heterogeneous – in space and time – environment structure and its dynamics; *(ii)* the local interactions between biological entities/agents (*locality*) and their environment. An agent-based simulation means executing the MAS and studying its evolution through time, in particular: *(i)* observing individual and environment evolution; *(ii)* observing global system properties as emergent properties from agent-environment and inter-agent local interaction; *(iii)* performing in-silico experiments. The approach is ideal then for studying the systemic and emergent properties that characterise a biological system, which are meant to be reproduced *in virtuo*. In the context of biological system, agent-based models can therefore account for individual cell biochemical mechanisms – gene regulatory network, protein synthesis, secretion and absorption, mitosis and so on – as well as the extracellular matrix dynamic – diffusion of morphogens, degradation and so on – and their dynamic influences on cell behaviour.

The Drosophila Melanogaster Embryo Development

One of the best example of pattern formation during morphogenesis is given by the patterning along the anteroposterior axis of the fruit fly *Drosophila Melanogaster*. In this section we briefly propose a model for the pattern formation in the embryo. We reproduce the interaction among pathways inside the cell, that is responsible for its stabilisation into a specific genetic expression, and the cell-to-cell interactions mediated by cues, *i.e.*, transcription factors that enhance or inhibit the original cell activity and cause the formation of regions of cells with similar activity.

Biological background

The egg of *Drosophila* is about 0.5 mm long and 0.15 mm in diameter. It is already polarised by differently localised mRNA molecules which are called *maternal effects* The early nuclear divisions are synchronous and fast (about every 8 minutes): the first nine divisions generate a set of nuclei, most of which move from the middle of the egg towards the surface, where they form a monolayer called *syncytial blastoderm*. After other four nuclear divisions, plasma membranes grow to enclose each nucleus, converting the syncytial blastoderm into a *cellular blastoderm* consisting of about 6000 separate cells.

Up to the cellular blastoderm stage, development depends largely – although not exclusively – on maternal mRNAs and proteins that are deposited in the egg before fertilisation. After cellularisation, cell division continues asynchronously and at a slower rate, and the transcription increases dramatically. Once cellularisation is completed the gene expression regionalisation is already observable.

The building blocks of anterior-posterior axis patterning are laid out during egg formation thanks to the maternal effects. *Bicoid* and *caudal* are the maternal effect genes that are most important for patterning of anterior parts of the embryo in this early stage. They are transcription factors that drive the expression of *gap genes* such as *hunchback* (Hb), *Krüppel* (Kr), *knirps* (Kni) and *giant* (Gt), as shown in the diagram of Fig. 1; there, *tailess* (Tll) also appears as gap genes whose regulation we do not represent here. Gap genes together with maternal factors then regulate the expression of downstream targets, such as the *pair-rule* and *segment polarity genes*. The segmentation genes specify 14 parasegments that are closely related to the final anatomical segments (Alberts et al., 2002; Gilbert, 2006).

Methods

Our model consists of a set of agents that represent the cells, as well as of a grid-like environment representing the extracellular matrix. Agent internal behaviour reproduces the gene regulatory network of the cell, while agent interaction with the environment models the process of cell-to-cell communication mediated by the signalling molecules secreted in and absorbed by the extra-cellular matrix. Our model aims at reproducing the expression pattern of the gap genes, before the pair-rule genes are activated.

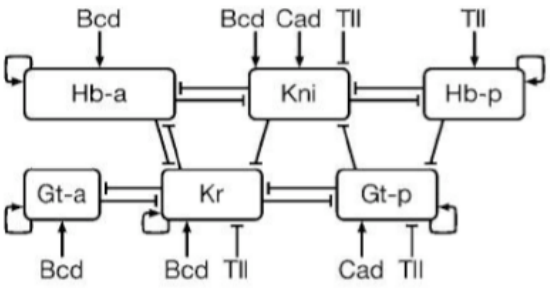


Figure 1: Gene regulatory network as in Perkins et al., 2006; Gursky et al., 2004

Model of the cell

We model different cell processes: secretion-absorption diffusion of chemicals from and towards the environment, cell growth and cell internal dynamics—gene regulatory network in particular.

Chemical diffusion Until cleavage cycle 13, there are no cell membranes surrounding cell cytoplasm and nucleus, and the transport of material mainly interests the nuclear membrane, and involves also cell membranes once they grow. We do not distinguish between the syncytial blastoderm and the cellular blastoderm stages, and model the process of molecule secretion and absorption as facilitated diffusion—the literature lacks of information about the transport mechanisms of such transcription factors and about the rate of diffusion.

Gene regulatory network Gene transcription begins with the binding at the gene promoter of one or more transcription factors. Gene transcription might also be repressed once transcription factors bind to other control regions called silencers. This activation/inhibition is stochastic (Kaern et al., 2005) and highly depends on the concentration of transcription factors. For those genes whose transcription is regulated by a set of other gene products we define a probability of transcription as a sum of positive and negative contributions from the concentration of enhancers and silencers, respectively. The probability of transcription of *hunckback*, according to the graph of Fig. 1, is then calculated as:

$$P_h = f([Bicoid]) + f([Hunchback]) + f([Tailess]) - f([Knirps]) - f([Kruppel])$$

where f is a linear function with the proportionality constant representing the strength of interaction. Then if $P_h > 0$ the protein is synthesised, otherwise the gene remains silent.

No distinction has been done in the model between anterior (a) and posterior (p) *hunckback* and *giant*, whose different expression only deals with the spatial distribution of maternal products.

Mitosis According to Fig. 2 where we show how the number of cells varies in the first four hours of embryo development – until the cleavage cycle 14, temporal class 8 – we computed the rate of division as a function of time: cell division is fast and synchronous until cleavage cycle 9, then slows down and becomes asynchronous. The rate of division is constant in the first hours of development (9.05 min^{-1}), then decreases until a low value (0.2 min^{-1}), as it appears in Figure 3.

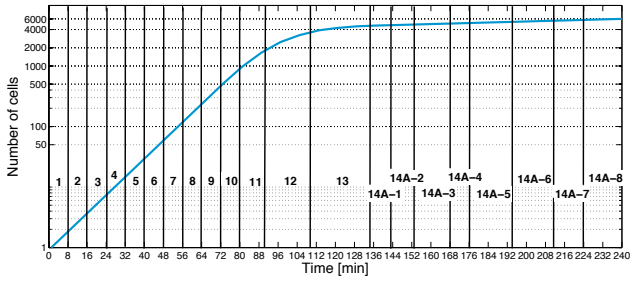


Figure 2: Number of cells varying from one to 6000 in the first 14 cleavage cycles

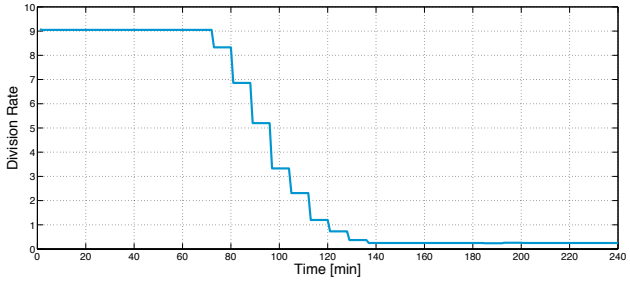


Figure 3: Rate of division in the first 14 cleavage cycles

Model of the environment

The 3D-tapered structure of the embryo, as in Figure 4, is modelled as a 2D-section of the embryo along the antero-posterior axis (c) under the assumption that the dynamics along the other two axis, a and b, does not influence what happens along the c axis. The space scale is 1:3.33 according to the real dimension of the embryo where the antero-posterior axis is almost three times the dorso-ventral one a. Space is not continuous but grid like, and each location might be occupied both by a set of morphogenes and by a cell.

The environment has its own dynamics, which mainly consists in the diffusion of morphogenes from region with bigger concentration to region with lower concentration, according to the *Fick's low* that the diffusive flux is pro-

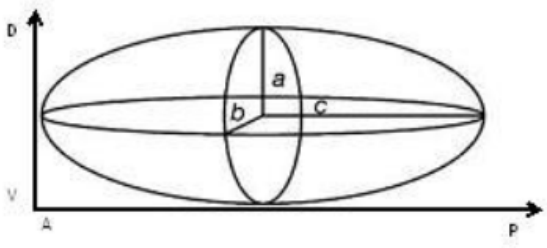


Figure 4: 3D-structure of real embryo

portional to the local concentration gradient (Smith and Hashemi, 2005). This law is used in its discretised form.

Model implementation and simulation procedure

The model is implemented on top of Repast Simphony[1], an open-source, agent-based modelling and simulation toolkit. It provides all the abstraction for directly modelling the agent behaviour and the environment. It implements a multi-threaded discrete event scheduler. In our simulations a time step corresponds to 4 seconds of the real system simulated. This is the smallest time-interval allowing for a good compromise between precision in the observation of the system dynamic and simulation execution time.

Simulations are executed from the cleavage cycle 11, when the zygotic expression begins. We used the experimental data available online in the FlyEx database[2]. The data contains quantitative wild-type concentration profiles for the protein products of the seven genes – *Bcd, Cad, Hb, Kr, Kni, Gt, Tll* – during cleavage cycles 11 up to 14A, which constitutes the blastoderm stage of *Drosophila* development. These data are used to validate the model dynamic. Expression data from cleavage cycle 11 are used as initial condition—see Fig. 6. The concentration of proteins are unitless, ranging from 0 to 255, at space point x, ranging from 0 to 100 % of embryo length.

Model parameters are: *(i)* diffusion constants of morphogenes motion; *(ii)* rates of gene interactions; *(iii)* rates of protein synthesis. Few data are available in literature for inferring the diffusion constants. We took inspiration from the work of Gregor et al., 2007 that calculates the diffusion rate for *Bicoid* and we imposed the value for all the morphogenes at 0.3 $\mu m^2/sec$. The rates of gene interactions and of protein synthesis are determined through a process of automatic parameter tuning. The task is defined as an optimisation problem over the parameter space. The optimisation makes use of *metaheuristics* – particle swarm optimisation – to find a parameter configuration such that the simulated system has a behaviour comparable with the real one (Montagna and Roli, 2009). We supported the automatic parameter tuning with a process of model refinement which slightly changed the topology of gene regulatory network, adding some edges that we found necessary for obtaining the real behaviour. An argumentation about the final model is provided in the Discussion.

[1] `http://repast.sourceforge.net/index.html`
[2] `http://flyex.ams.sunysb.edu/flyex/index.jsp`

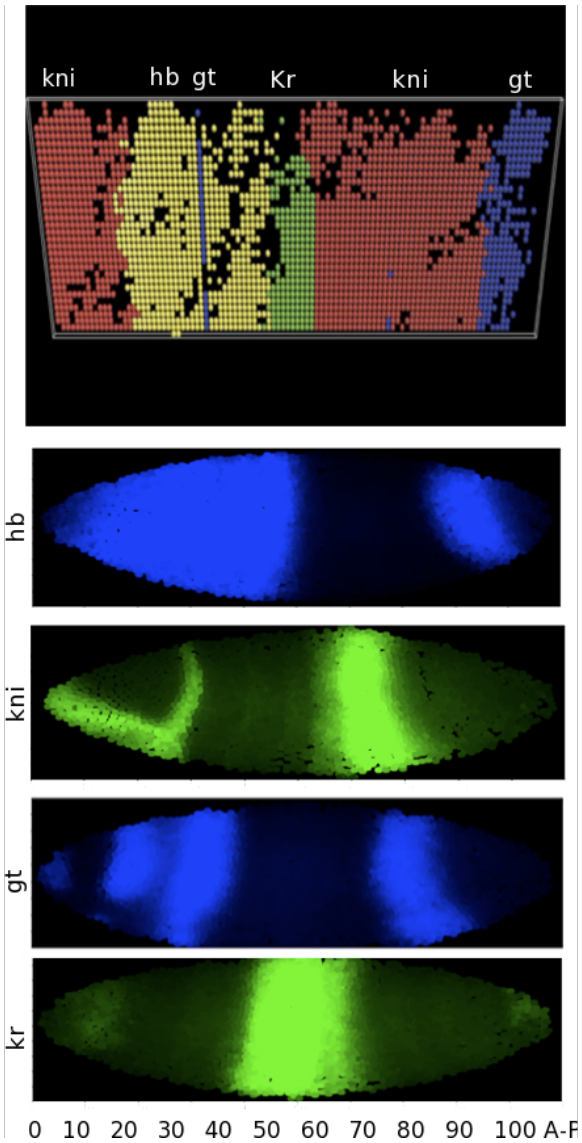


Figure 5: Qualitative results

Simulation results

Qualitative results charted in the 2D-grid are shown in Fig. 5 (top) for expression of *hb, kni, gt, Kr* at the eighth time step of cleavage cycle 14A. The image shows for each cell of the embryo the genes with higher expression. It clearly displays the formation of a precise spatial pattern along the A-P axis but it does not give any information about gene expression level. Experimental data are also provided in Fig. 5 (bot-

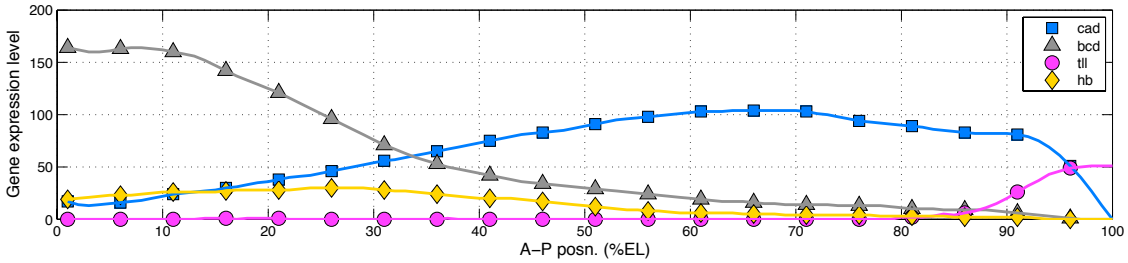


Figure 6: Experimental data at cleavage cycle 11 of genes with non-zero concentration: maternal genes *Bcd, Cad, Tll* and the gap gene *Hb*

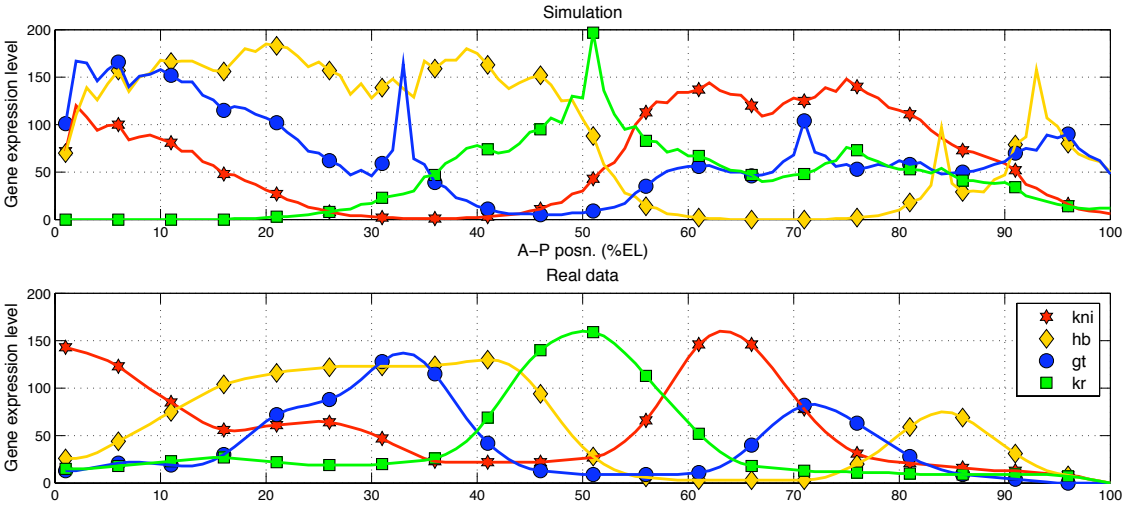


Figure 7: Quantitative simulation results for the four gap genes *hb, kni, gt, Kr* at a simulation time equivalent to the eighth time step of cleavage cycle 14A (top) and the corresponding experimental data (bottom)

tom) with 2D-Atlas reconstructing the expression level of the four genes in A-P sections of the embryo. More precise information about simulation behaviour are given with the quantitative results provided in Fig. 7. A comparison shows that the expression pattern of genes *Hb, Kni, Gt* and *Kr* nicely fit the spatial distribution shown in the experimental data: *Hb* is expressed in the left pole until about 45% of embryo length, while it does not appear on the right as it should between about 85% and 95%; *Kni* is correctly expressed on the extreme left and between 65% and 75% but it is slightly over-expressed on the right; *Gt* is reproduced in the correct regions but over-expressed in the extreme left and slightly under-expressed between 20% and 30%; finally, *Kr* properly appears between 40% and 60%.

Discussion

Through the model refinement we found the network showed in Fig. 8 where some more interactions are performed. The weight in sec^{-1} of each node is then reported in Fig. 9.

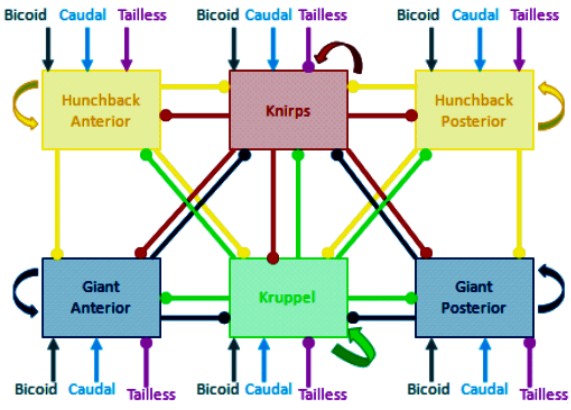


Figure 8: Gene regulatory network

	BICOID	CAUDAL	TAILLESS	HUNCHBACK	KNIRPS	KRUPPEL	GIANT
HUNCHBACK	0.0071	0.0018	0.0065	0.0400	-0.0080	-0.003	-
KNIRPS	0.077	0.0096	-0.0140	-0.0060	0.0700	-0.0055	-0.0037
KRUPPEL	0.0045	0.0123	-0.0240	-0.0002	-0.0073	0.0640	-0.0057
GIANT	0.0042	0.0124	-0040	-0.0032	-0.0030	-0.0096	0.0360

Figure 9: Rate of gene interactions

Bcd and *Cad* are activators of the gap genes. As maternal factor their central role is in fact to input the wave of zygotic expression. In particular, given the spatial distribution of their expression, *Bcd* is responsible for the activations on the left side of the embryo, while *Cad* in the opposite side. *Tll* enhances *Hb* expression while inhibits the expression of all the others as in the previous model. The interactions among gap genes are slightly different. As before *Hb* and *Kni* on one side and *Gt* and *Kr* on the other side inhibits one each other, and from the parameters found we infer that these are the strongest inhibitions among gap genes; *Hb* then weakly inhibits *Kr* and vice-versa, as well as *Gt* versus *Kni*. New weak edges have been found between *Kni* versus *Gt*, and *Kr* versus *Kni*.

As far as we know, there are no evidences in biological literature that already support the above results. It might be a starting point for new laboratory experiments.

Conclusion

The process of spatial organisation resulting from the morphogenesis process is demonstrated to be highly-dependent by the interplay between the dynamics at different levels of the biological systems hierarchical organisation. In modelling and simulating the phenomena of morphogenesis it might be appropriate to reproduce such a hierarchy. In this work we have described the application of ABM as an approach capable of supporting multi-level dynamics.

We studied the phenomenon of pattern formation during *Drosophila* embryo development, modelling the interactions between maternal factors and gap genes that originate the early regionalisation of the embryo. The possibility to model both the reactions taking place inside the cells that regulate the gene expressions, and the molecules diffusion that mediates the cell-to-cell communication, makes it possible the reproduction of the interplay between the two levels in order to verify its fundamental role in the spatial self-organisation characteristic of such a phenomenon.

The results presented show the formation of a precise spatial pattern which have been successfully compared with observations acquired from the real embryo gene expressions.

Future work will be firstly devoted to extending the model with the introduction of new phenomena on the side of both intracellular dynamics and cell-to-cell interaction. Gene regulatory network will be enlarged with other sets of genes which are downstream to gap genes such as the pair rule genes, *even-skipped* as first, whose expression gives rise at the characteristic segments of *Drosophila* embryo. Mechanisms regulating cell movements will then be added – cell adhesion and chemotaxis in particular – as soon as they are known to play a crucial role in cell sorting during morphogenesis.

Finally. we are planning to exploit the predictive power of the model analysing embryos that are not wild type, for instance performing in-silico Knock-Out experiments.

References

Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P. (2002). *Molecular Biology of the Cell*. Garland Science Textbooks. Garland Science, 4th edition.

Beurier, G., Michel, F., and Ferber, J. (2006). A morphogenesis model for multiagent embryogeny. In Rocha, L. M., Yaeger, L. S., Bedau, M. A., Floreano, D., Goldstone, R. L., and Vespignani, A., editors, *Artificial Life X: Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, pages 84–90. MIT Press, Cambridge, MA.

Cickovski, T. M., Huang, C., Chaturvedi, R., Glimm, T., Hentschel, H. G. E., Alber, M. S., Glazier, J. A., Newman, S. A., and Izaguirre, J. A. (2005). A framework for three-dimensional simulation of morphogenesis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2:273–288.

Gilbert, S. F. (2006). *Developmental Biology, Eighth Edition*. Sinauer Associates Inc., Eighth edition.

Gregor, T., Wieschaus, E., McGregor, A., Bialek, W., and Tank, D. (2007). Stability and nuclear dynamics of the bicoid morphogen gradient. *Cell*, 130(1):141–152.

Gursky, V. V., Jaeger, J., Kozlov, K. N., Reinitz, J., and Samsonov, A. M. (2004). Pattern formation and nuclear divisions are uncoupled in drosophila segmentation: comparison of spatially discrete and continuous models. *Physica D: Nonlinear Phenomena*, 197(3-4):286–302.

Kaern, M., Elston, T. C., Blake, W. J., and Collins, J. J. (2005). Stochasticity in gene expression: from theories to phenotypes. *Nature reviews. Genetics*, 6(6):451–464.

Klügl, F., Oechslein, C., Puppe, F., and Dornhaus, A. (2002). Multi-agent modelling in comparison to standard modelling. In Barros, F. J. and Giambiasi, N., editors, *Artificial Intelligence, Simulation and Planning in High Autonomy Systems*, pages 105–110. SCS Publishing House.

Merelli, E., Armano, G., Cannata, N., Corradini, F., d'Inverno, M., Doms, A., Lord, P. W., Martin, A., Milanesi, L., Möller, S., Schroeder, M., and Luck, M. (2007). Agents in bioinformatics, computational and systems biology. *Briefings in Bioinformatics*, 8(1):45–59.

Michel, F., Ferber, J., and Drogoul, A. (2009). Multi-Agent Systems and Simulation: a Survey From the Agents Community's Perspective. In Danny Weyns, A. U., editor, *Multi-Agent Systems: Simulation and Applications*, Computational Analysis, Synthesis, and Design of Dynamic Systems, pages 3–51. CRC Press - Taylor & Francis.

Montagna, S. and Roli, A. (2009). Parameter tuning of a stochastic biological simulator by metaheuristics. In *AI*IA*, volume 5883 of *Lecture Notes in Computer Science*, pages 466–475. Springer.

Perkins, T. J., Jaeger, J., Reinitz, J., and Glass, L. (2006). Reverse engineering the gap gene network of *Drosophila Melanogaster*. *PLoS Comput Biol*, 2(5):e51.

Smith, W. and Hashemi, J. (2005). *Foundations of Materials Science and Engineering*. McGraw-Hill, 4th edition.

Thorne, B., Bailey, A., Desimone, D., and Peirce, S. (2008). Agent-based modeling of multicell morphogenic processes during development. *Birth Defects Res C Embryo Today*, 81(4):344–353.

Turing, A. M. (1952). The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society (B)*, 237:37–72.

Uhrmacher, A. M., Degenring, D., and Zeigler, B. (2005). Discrete event multi-level models for systems biology. In Priami, C., editor, *Transactions on Computational Systems Biology I*, volume 3380 of *Lecture Notes in Computer Science*, pages 66–89. Springer.

Weyns, D., Omicini, A., and Odell, J. (2007). Environment as a first-class abstraction in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 14(1):5–30. Special Issue on Environments for Multi-agent Systems.

Yamins, D. and Nagpal, R. (2008). Automated global-to-local programming in 1-d spatial multi-agent systems. In *7th International Joint Conference on Agents and Multi-Agent Systems (AAMAS-08)*, pages 615–622, Estoril, Portugal. IFAAMAS.

Morphogen positioning by the means of a hydrodynamic engine

Sylvain Cussat-Blanc, Jonathan Pascalie, Hervé Luga and Yves Duthen

University of Toulouse - CNRS - UMR 5505
2 rue du Doyen-Gabriel-Marty
31042 Toulouse Cedex 9, France
{sylvain.cussat-blanc, jonathan.pascalie, herve.luga, yves.duthen}@irit.fr

Abstract

Artificial embryogeny aims at developing a complete organism starting from a unique cell. Nowadays many algorithms exist to synthesize artificial creature shapes or behaviours. With the purpose of shape and high-level behaviour joint evolution, one of the key aspects is the synthesis of positional information. Such pieces of information, called morphogens, are in many developmental models embedded in the environment and interactions are made through simple protein receptors. In this paper, we propose a new and original approach to solve the morphogen-positioning problem. We use a hydrodynamic model to replace the classical spreading algorithm. Mechanical constraints (the cell shape) and a dynamic activity are integrated. Thanks to this improvement, the cell behaviour can affect the spreading algorithm: cells can apply forces on the hydrodynamic environment to create substrate flows. Through experiments, this paper shows the way to develop complex shapes using this kind of simulator and proposes how to extend the simulation in a 3-D world in which physical laws are taken into account.

Introduction

Literature offers many developmental models able to develop several kinds of creatures starting from a single cell (Stanley and Miikkulainen, 2003). Many goals motivate that kind of research work: to develop a particular shape, to evolve a high-level behaviour, etc. or, at a higher level, to understand living systems by the use of such models to simulate their mechanisms. Nowadays, a complete research field axis is about shape development from a single cell. One of the major problems of this work is morphogen positioning. Morphogens are often used as positional information to lead cells in their development. In nature, positional information is a key aspect in morphogenesis, embryogenesis, organogenesis and in behaviour synthesis at last. Evolvable mechanisms should be used in developmental models to spread their positional information in the environment. This could allow the emergence of a complex structure and/or behaviour. Keeping this goal in mind, we choose to embed morphogen positioning in cellular activity thanks to a hydrodynamic simulator which cells are able to interact with.

Our previous work proposed a developmental model, named *Cell2Organ* (Cussat-Blanc et al., 2008), based on a strong simplification of mechanisms used by living systems. The developmental model is a chemical simulator where organisms have to develop a metabolism, may have self-repairing capacities and have to perform user-defined functions. In this paper, we show the plug of a hydrodynamic engine with the developmental model in order to solve one of its main limitations: manual morphogen positioning. In comparison to a classical spreading algorithm, widely used in developmental models in literature, the use of a hydrodynamic engine allows more possibilities. Organisms will have the ability to create fluid flows, to move substrates or structures to organize the environment at their convenience. Gastrulation stage of vertebrate embryos can be simulated with this kind of system. In this early development stage, morphogens are positioned thanks to a physical invagination that induces many flows in the environment, as explained by some physicists' theories such as (Fleury, 2009).

In our bio inspired approach, the use of a hydrodynamic engine has sense looking at the early development stage. Gastrulation stage is seen as the first step of the morphogenetic process. During this step, high dynamic is observed in the embryo. Undifferentiated cells migrate and the egg membrane invaginates itself. Hydrodynamic forces are generated with a combination of these mechanisms. These forces are constraints for the different actors of the system. The consequence is the positioning of a kind of "mechanical gradients", in other words growth lines take place thanks to the created mechanical constraints. These developmental axis could be seen as an embryogenic pre-pattern. This latter is, as the example of vertebrates, four members positioned in pairs on the anterior and posterior zones of the organism.

This paper is organised as follows. Section 2 gives the related works on artificial development and morphogen positioning. Section 3 summarizes the model *Cell2Organ*. Section 4 details the hydrodynamic layer we add to the model in order to set up morphogens in the environment. Section 5 presents some results we obtain thanks to this new layer. We first develop simple shapes like diamonds or rectangles and a mushroom-shaped creature. We then develop more complex shapes. We conclude these experimentations by hav-

ing a discussion on the practicality of such a morphogenesis process to generate bigger creatures that could populate a 3-D world based on newtonian dynamics. Finally, we expose several options to improve this work.

Related works

Over the past few years, more and more models concerning artificial development have been produced. A common method for developing digital organisms is to use Artificial Regulatory Networks (ARN). Banzhaf was one of the first to design such a model (Banzhaf, 2003). In his work, the beginning of each gene, before the coding itself, is marked by a starting pattern named "promoter". This promoter is composed of enhancer and inhibitor sites that allow the gene activations and inhibitions regulation. Another different approach is based on Random Boolean Networks (RBN) first presented by Kauffman (Kauffman, 1969) and re-used by Dellaert (Dellaert and Beer, 1994). An RBN is a network in which each node has a boolean state: activate or inactivate. The nodes are interconnected by boolean functions, represented by edges in the net. The cell function is determined during genome interpretation.

Several models dealing with shape generation have recently been designed (de Garis, 1999; Kumar and Bentley, 2003; Stewart et al., 2005; Chavoya and Duthen, 2008; Knabe et al., 2008; Joachimczak and Wróbel, 2009). Most of them use artificial regulatory network and morphogens to drive the development. With the latter approach, morphogens positioning in the environment is one of the main difficulties. In order to produce user-defined shapes as a French flag - that is one of the main benchmarks, a precise morphogen positioning is crucial. Two main methods exist to solve this problem: on the one hand, cells can produce morphogens by themselves that are spread in the environment with a simple spreading algorithm (Stewart et al., 2005; Knabe et al., 2008; Joachimczak and Wróbel, 2009) and, on the other hand, environment can contain built-in fixed morphogens (Chavoya and Duthen, 2008). Various shapes are produced, with or without cell differentiation. The well-known French flag problem was solved by Chavoya and Duthen, Knabe and recently in 3-D by Joachimczak. This problem shows the model differentiation capacity during multiple colour shifts.

Eggenberger was one of the first to propose a model that takes a leaf out of gastrulation (Hotz, 2003). In his work, both physics engine and artificial regulatory network (ARN) are used. The ARN controls cells behaviour whereas a physics engine allows to apply local constraints. Physical interactions could be observed between the cells and between the cells and the environment. Nevertheless, the substrate spread is made by cellular activity but is not influenced by the mechanical activity, that is to say movements made by cells do not spread any morphogen. Some biological theories about embryonic development bring out that hydrodynamic morphogen movements seem to be the basics of organogenesis (organ positioning the early embryo) and an explanation of most living being symmetric morphology (Cartwright et al., 2009; Fleury, 2009). To study the possible benefits of the morphogen flow creation in environments, we proposed to use a hydrodynamic layer whose activity is directly influenced by forces applied by cells.

This paper proposes a new morphogen positioning approach. More bio-inspired than biologically acceptable, we use a hydrodynamic engine to produce morphogen flows in the environment. Special cells have the ability to expulse morphogens with a given force whereas others will use the positional information to produce a defined shaped creature. Because our research axis is more focussed on creature development for virtual reality application than on cell mechanism realistic simulation, this bio-inspired approach is sufficient. Moreover, this kind of method could be used for future modular robots that could have the ability to expulse a substrate.

The next section presents our developmental model. It is based on action optimisation networks and on an action selection system inspired by classifier rule sets. It has been presented in details in (Cussat-Blanc et al., 2008).

Summary of *Cell2Organ*

We choose to implement the environment as a 2-D toric grid. This choice allows a significant decrease in the simulation complexity keeping a sufficient degree of freedom thus reducing the simulation computation time.

The environment contains several kinds of substrates. They spread within the grid, minimizing the variation of substrate quantities between two neighbouring points. These substrates can spread on the grid at several speeds and can interact with other substrates. Interactions between substrates can be viewed as a great simplification of a chemical reaction: using different substrates, the transformation will create new substrates, emitting or consuming energy. Formally, this chemical reaction can be written as follows:

$$a_1 s_1 + a_2 s_2 + ... + a_n s_n \rightarrow a'_1 s'_1 + a'_2 s'_2 + ... + a'_m s'_m \quad (\delta_{energy})$$

where s_i represents substrates, $a_i \in \mathbb{N}$ and $a'_j \in \mathbb{N}$ ($i \in 1..n$, $j \in 1..m$ are stoichiometric coefficients of the reaction and $\delta \in \mathbb{R}$ the quantity of energy produced (if positive) or consumed (if negative) during the reaction. For example, the reaction $2A + B \rightarrow C$ $(+50)$ produces one unit of C substrate from two units of A substrate and one of B's. The reaction also produces 50 units of energy.

To reduce the complexity, the environment contains a list of available substrate transformations. Only cells can trigger substrate transformations.

Cells

Cells act in the environment, more precisely on the environment's spreading grid. Each cell contains sensors and has

different abilities (or actions). An action selection system allows the cell to select the best action to perform at any moment of the simulation. Finally, a representation of an ARN is available inside the cell to allow specialization during division.

Each cell contains different density sensors positioned at each cell corner. Sensors allow the cell to measure the amounts of substrates available its Von Neumann neighbourhood. The list of available sensors and their position in the cell are described by the genetic code.

To interact with the environment, cells can perform different actions: perform a substrate transformation, absorb or reject substrates in the environment, divide (see later), wait, die, etc. This list is not exhaustive. The addition of an action is simplified by model implementation. As with sensors, not all actions are available for the cell: the genetic code will give the available action list.

Cells contain an action selection system. A system based on a set of rules is inspired by classifier systems. It uses data given by sensors to select the best action to perform. Each rule is composed of three parts: (1) The *precondition* describes when the action can be triggered. A list of substrate density intervals describes the neighbourhood in which action must be triggered. (2) The *action* gives the action that must be performed if the corresponding precondition is respected. (3) The *priority* allows the selection of only one action if more than one can be performed. The higher the coefficient, the more probable the rule selection.

Division is a particular action performable if the next three conditions are respected. First, the cell must have at least one free neighbour to create the new cell. Secondly, the cell must have enough vital energy to perform the division. The vital energy level needed is defined during the environment specification. Finally, during the environment modelling, a condition list can be added.

Action optimisation

A new cell created after division is totally independent and interacts with the environment. During a division, the cell can optimize a group of actions. In nature, this specialisation seems to be mainly carried out by a gene regulatory network (GRN). In our model, we imagine a mechanism that plays the role of an artificial GRN. Each action has an efficiency coefficient that is linked to the action optimisation level: the higher the coefficient, the lower the vital energy cost. Moreover, if the coefficient is null, the action is not yet available for the cell. Finally, the sum of efficiency coefficients remains constant during the simulation. In other words, if an action is optimised by increasing its efficiency coefficient during a division, another (or a group of) efficiency coefficient has to be decreased. A network represents the transfer rule during a division stage. In this network, weighed nodes represent cell actions with their efficiency coefficients and weighed edges representing efficiency coefficient quantities that will be transferred during the division. Efficiency coefficient variations during division stage allow cell specialisation over divisions.

Creature's genome

To find the best-adapted creature to a specific problem, we use a genetic algorithm. Each creature is tested in its environment. This latter returns the fitness at the end of the simulation. Each creature is coded with a genome composed of three different chromosomes: the list of *available actions*, an encoding of the *action selection* system and an encoding of the *optimisation network*.

Because of the complexity of developed creatures, the genetic algorithm had to be improved. First, we have decided to parallelise it on a computation grid. We used a middleware, named ProActive, that allows a total abstraction of grid infrastructure (Caromel et al., 2006). We applied a Master/Worker algorithm to parallelise our genetic algorithm. This algorithm is well suited to artificial evolution because the creature genome is small and the fitness computing cost is very important. Because of the small size of the genome, the network bottleneck induced by a Master/Worker architecture deployed on a computational grid will not heavily increase the computation time. Moreover, because the Master/Worker algorithm preserves the properties of a classical genetic algorithm, the number of generations needed by the algorithm to converge and the final solution quality are exactly the same with or without parallelisation.

A second optimisation of our genetic algorithm consists in leading the algorithm in its search. In our experimentation, the fitness function can be broken up with sub-objectives to describe the different evolution stages of the creature. This approach, commonly named *incremental evolution*, has been used in different domains such as behaviour simulation (Kodjabachian and Meyer, 1998; Mouret and Doncieux, 2008) or genetic programming (Walker, 2004). Authors generally conclude that global computation time is the same in comparison to a classical fitness but this algorithm gives more adapted solutions. In our problem, we generally break the fitness up in the three following stages: *metabolism* that is the lowest level function needed by the creature, *cell birth quantity* during the simulation shows the capacity of the organism to develop itself in the environment and *global fitness* that gives the efficiency of the organism to solve the problem (can also be broken up into sub-objectives).

Example of generated creatures

Different creatures have been generated using this model. For example, we develop a *harvester*, a creature able to collect a maximum of substrate scattered all over the environment and to transform it into division material and waste. The creature has to reject the waste because of each cell limited substrate capacity. Another creature is the *transfer system*. Presented in (Cussat-Blanc et al., 2008), this crea-

ture is able to move substrate from one point to another. This creature is interesting because it has to alternate its behaviour between performing its function and developing its metabolism to survive. Finally, different *morphologies*, such as a starfish, a jellyfish or any user-designed shape, have been obtained (Cussat-Blanc et al., 2008). Once again, the organism must develop its metabolism to be able to sustain its activity.

All generated creatures have a common property: they are able to repair themselves in case of injury (Cussat-Blanc et al., 2009). This feature is an inherent property of the model. It shows the phenotype plasticity of produced creatures.

The last model's interesting feature is organ cooperation capacity to produce bigger structures. We have developed organs separately and built an organism composed of these organs that has a higher-level purpose. We create for example a self-feeding structure composed of four organs: two transfer systems and two producer-consumers.

Concerning the morphology development, one limitation of the model is the necessity to position morphogens by hand in the environment. In order to solve this problem, we propose a hydrodynamic layer that allows morphogen flow creation by cells. The organism has to make a morphogenetic blueprint of the shape in the environment before it develops itself by following division information. The next section details the hydrodynamic model we use and its set up options. The integration to the developmental model is also detailed.

Hydrodynamic layer

This simulator manages hydrodynamic substrate interactions of our model. Its main aim is to propose a method inspired by the gastrulation of some living beings to position morphogens. This early stage of the organism development allows the morphogen positioning of the embryo in its immediat environment. It then allows the development of its organs. By the use of a hydrodynamic simulator in our model, we can produce the apparition of flows in the environment that correspond to flows created by the organism when it performs its actions (division, substrate absorption or rejection in particular). Thus, cells can for example expulse a substrate to be positioned in the environment in a specific direction and with a specific strength.

Hydrodynamic model

Because of the computation cost induced by the hydrodynamic simulator complexity, we use a method that reduces the resource usage of the hydrodynamic layer on our simulation but keeps enough realism and degree of freedom. We base our work on Jos Stam's solver (Stam, 2003). This model is mainly used for image processing. This quite simple approach is interesting because its ability to solve Navier and Strokes' equations has been proved.

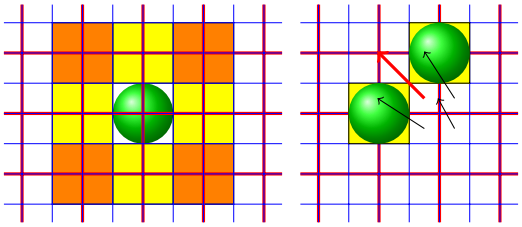


Figure 1: (a) Relative positioning of the chemical (red bold lines) and hydrodynamic (blue thin lines) environment. (b) Velocity vectors (red bold arrow) allow the spreading of few substrates on the other side of the cellular membrane.

In this model, the environment is a grid on which fluids particles are moving following speed vectors. Particles here represent our substrates. Our simulated cells are impassable obstacles. When a particle hits a cell membrane, the speed vector that corresponds to the collision point is modified in order to redirect the particle along the cell edge. In a first step, to simplify the simulation, all substrates will be spread separately, that is to say independently of one another. In other words, substrate flow interactions are not simulated with model. In our experimentation of morphogen positioning, this limitation has been overtaken bringing together all morphogens in a unique substrate and then breaking it up in the developmental model into several morphogens.

To ripen border conditions, the hydrodynamic simulator grid size has been doubled in comparison with the chemical simulator grid. Indeed, the smaller the grid subdivision, the more precise the border condition computation. In other words, fluid flows will be more precisely described. Because the grid subdivision strongly increases the computation cost, the hydrodynamic grid has only been subdivided by two in comparison to the chemical grid. The algorithm has also been adapted to take into consideration the inter-cell spreading allowed by our previous spreading algorithm. Because obstacles represented by cells are stuck together, no fluid flow is possible between cells. In our model, the organism's external speed vectors are able to modify the organism's internal speed vector in order to create internal flows. Figure 1 is a scheme of the subdivision grid and force applications in the environment.

The non-conservation material quantity is one of the main limitations of this model. Indeed, during the simulation, the hydrodynamic engine can generate a small loss of material. Such a loss could be unacceptable for the developmental model on little quantities or on application linked with real data such as real cell simulation. The main aim of the hydrodynamic engine is to spread morphogens in the environment in order to develop a shaped creature. Such a loss of material could generate a non-desired growth of the organism. However, several methods exist to fix the problem. The first

one consists in the implementation of an energy conservation law, which equilibrates the substrate leaks due to equation reductions. A proportional distribution of lost material on the entire grid has been preferred because the energy conservation method is expensive in computation resources and will be difficult to apply to our simulator.

The number of adjustable parameters is another strength of this model. Many properties are implied in fluid movements. The first parameter is the viscosity coefficient. This coefficient is used to describe the fluid movement. the higher the coefficient, the easier the outflow on its support. The second parameter of the model is the substrate density. This latter represents the capacity of the substrate to be spread during its spread. The higher the coefficient, the higher the links between substrates particles. Finally, the last parameter on which the user should act is the intensity of the force applied on the environment. The higher the force intensity, the bigger the induced activity.

The integration in our cellular simulation is simple: the hydrodynamic engine totally replaces the traditional spreading algorithm previously used to spread substrates. Cells interact with the environment, in particular by absorbing or rejecting substrates. Without a hydrodynamic layer, their actions could not create the fluid flows due to molecular movement. Now, the hydrodynamic engine can simulate this kind of phenomenon. Expulsion strength with a particular direction can be given to the cell. According to hydrodynamic forces, cells can position now a substrate everywhere in its environment. Cells can also create flows to produce global movement in the environment. Substrate absorption can create suctions in the same way. Lastly, as defined in the developmental model *Cell2Organ*, during a division stage, future cell position must be empty before the daughter cell creation. In other words, substrates in the mother cell neighbourhood must be spread in the close environment in order to clean up the space to the daughter cell. The addition of a hydrodynamic engine instead of a classical spreading algorithm induces the creation of multiple complex flows (vortex in particular) near the division that can modify the behaviour of close cells.

Preliminary results of such an engine use with our developmental model has been presented in (Cussat-Blanc et al., 2010). Through several experimentations, we showed the capacity of this kind of model to create hydrodynamic flows by using a cell that rejects substrates in a chosen direction. We also showed the possibility to lead the flow with the use of other cells, these latter acting as obstacles in the environment. Finally, we showed a possible extension of the model *Cell2Organ* in a physical world through the experimentation of a muscular joint.

In this paper, the previously presented hydrodynamic engine is used to position morphogens in the environment. A cell able to reject morphogens in the environment by giving them a defined force is used to create a pattern that an organism endowed with a shape generation genome will follow. Thanks to this method, we develop several shapes presented in the next section.

Experiments

Experimental conditions

To provide comparable results, the environment composition is the same in all next experiments. In order to develop several shaped creatures, several hydrodynamic engine parameters (viscosity, expulsion force and density) and initial cell possibilities are tested. We first present the used environment and cell capacities, which are always the same in next experimentations. The results of these experimentations are then presented.

The environment is composed of 5 substrates: energetic substrate W that provides energy to cell by chemical reaction $W \rightarrow Energy$ (30), morphogen substrates NE, NW, SE, SW that provide division information to cells. Whereas W can spread and is massively present in the environment to develop an easy and efficient metabolism (the latter is not the main goal of the experiments), few morphogens are positioned in the environment to be only expulsed by cells.

Two kinds of cells are available in the environment.

Pusher cells have two actions: reject morphogen in the environment and wait for a signal. Because the cells' genome is very simple, it is hand-coded: cells can reject morphogens while they have units into their membranes; when they have no more substrate, they wait indefinitely.

Development cells can follow morphogens to develop a shaped-creature. The used genome has been evolved by a genetic algorithm and is detailed in (Cussat-Blanc et al., 2008). To summarize its functioning, cells have to manage their metabolisms provided by the energetic substrate W and their development functions (follow morphogens to produce a shape). A good genome has been found by a genetic algorithm and can produce any desired shape if morphogens are correctly positioned in the environment.

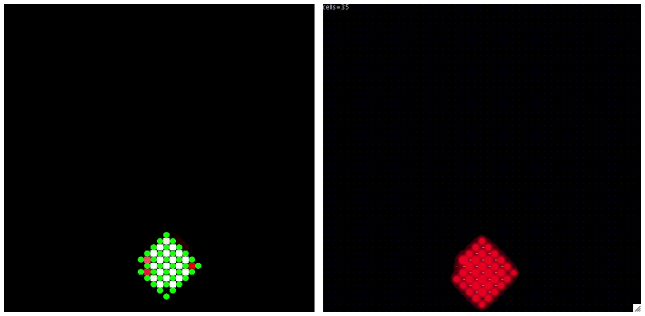
The rest of this section presents three experiments: simple shapes development, the development of a mushroom-like shaped creature and a four-armed creature. The aim is to study the impact of the hydrodynamic engine parameter modifications on the developed shapes. Videos of all these experiments are available on the website http://www.irit.fr/∼Sylvain.Cussat-Blanc.

Simple shapes

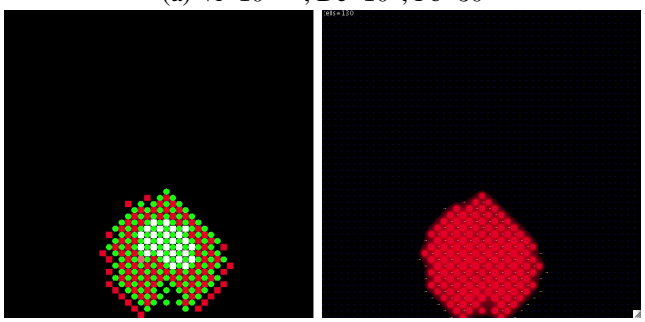
The aim of this first experiment is to give a range of possible shapes that can be produced by the model and to evaluate the

Viscosity	Density	Force
$10^{-6} < Vi < 10^{-28}$	$1 < De < 10^5$	$30 < Fo < 50$

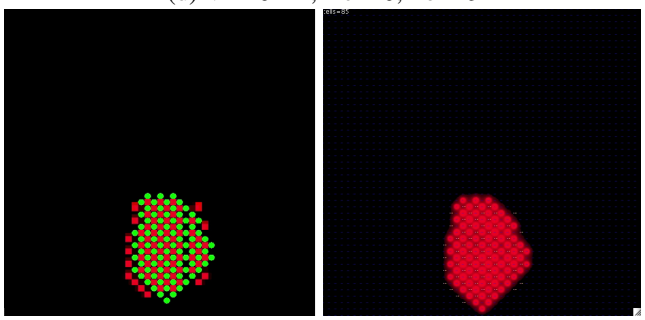
Table 1: Parameter acceptable value ranges



(a) Vi=10^{-26}, De=10^{5}, Fo=30



(d) Vi=10^{-18}, De=10, Fo=45



(c) Vi=10^{-6}, De=10, Fo=50

Figure 2: Influence of viscosity (Vi), density (De) and expulsion force (Fo) on developed shapes. On the left, hydrodynamic world where cells (in green) are obtacles and morphogen densities are represented with a gradient from white to red. On the right, the chemical world where cells (in red) are developping by following morphogens.

acceptable range of each parameter. In a first step, we empirically modify the parameters to develop as many shapes as possible. The parameter ranges are presented in table 1.

Figure 2 shows examples of produced shapes. As expected, parameter variations allow the development of different shape sizes (width) and statures (height). It is interesting to notice that figure 2(a) shows the capacity of the model to develop a square, a common problem of the literature (first step of the French flag problem). A high-density value ($De = 100000$) has been used here to keep morphogens grouped and make the production of such a shape possible.

With a low-density value, we develop the mushroom-shaped creature presented in figure 3. As previously introduced, the density parameter configures the stickiness force between substrates. The result is the development of a mushroom "cap" on the top of the shape, due to the vortex formation along the "stalk" that creates depressions. This accumulation produces two big vortexes of substrates on the top that produce the "cap".

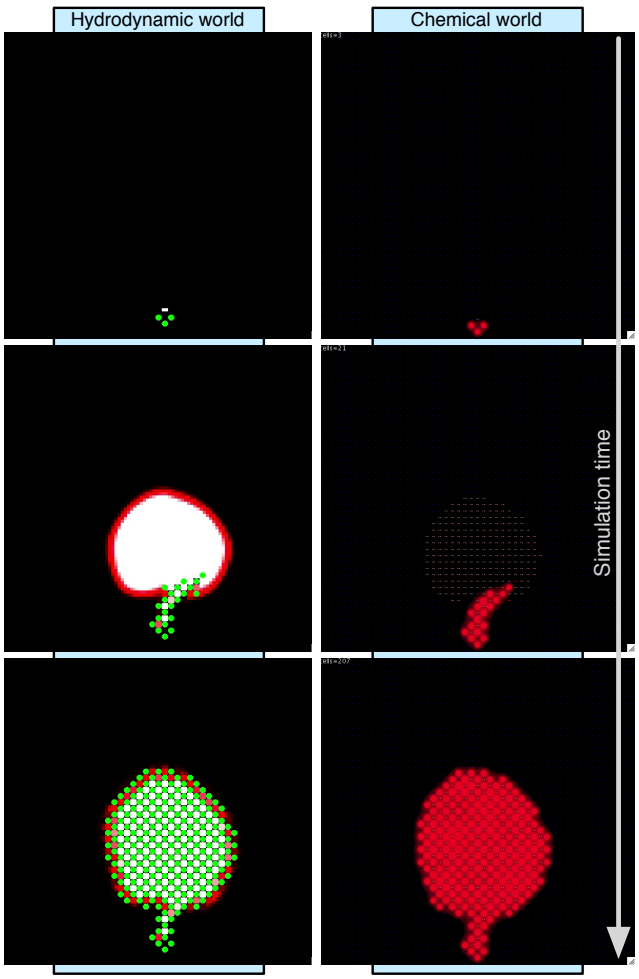
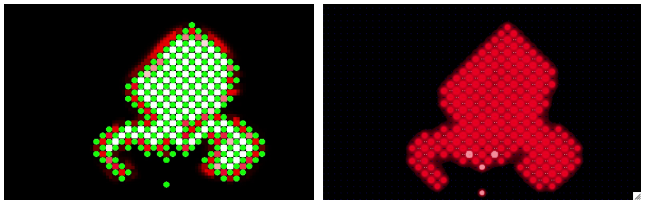


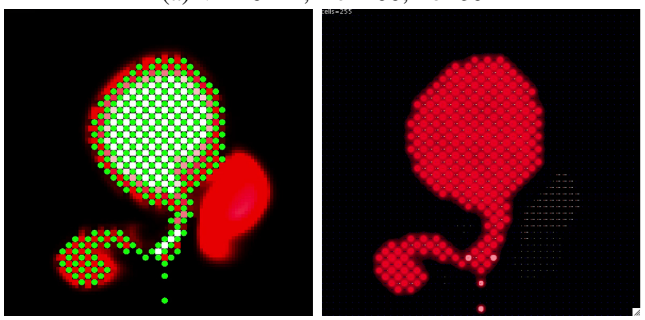
Figure 3: Development of a mushroom with morphogens positioning: a high fluid viscosity allows the cap formation.

Cell configuration influence on morphogen flows

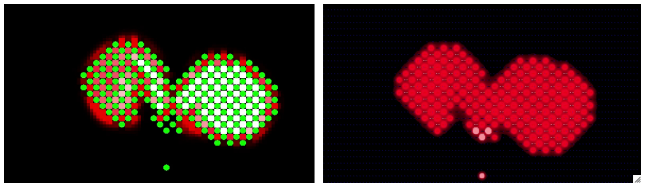
Modifying the initial cell configuration in the environment strongly influences the produced shape. Because cells are considered as obstacles in the hydrodynamic world, when a morphogen flow hits one of them, it is automatically divided in two flows that interfere. In these experiments, medium values of viscosity, density and expulsion forces are used. Depending on the cell position and the hydrodynamic engine parameters, many shapes can be obtained. Figure 4 presents some examples of initial configurations influences. Some interesting shapes appear in this figure: a kind of body endowed of tow tentacles in figure 4(a), an stomach-like shape



(a) Vi=10^{-28}, De=100, Fo=50



(b) Vi=10^{-10}, De=100, Fo=50



(c) Vi=10^{-22}, De=10, Fo=50

Figure 4: Influence of viscosity (Vi), density (De), expulsion force (Fo) and initial configuration on developed shapes (initial cells are highlighted in the chemical world).

on figure 4(b) and two wings on figure 4(c). This kind of shapes can be mixed to produce a complex creature and allow to jiggle in a simulated physical world. We will present an idea of such an improvement later in this paper.

The four-armed creature

In order to produce a bigger creature that could move and act in a physical world, we develop a creature endowed with four arms. Based on the same environment as before, we modify the pusher cell to give it the possibility to expulse substrates in the four cardinal directions (up, down, left and right) in order to produce four morphogen flows in the environment. According to previous results, we choose the hydrodynamic parameters to produce rectangular sets of cells that will represent the arms. The initial configuration is also based on a simple shape development: a 4-direction pusher cell is set in the centre of the environment and four development cells are positioned on its diagonals, all around the pusher cell. Figure 5 shows the development of this four-armed creature.

Artificial creatures, with a morphology such as the four-armed creature previously presented, could be endowed with locomotive abilities in a simulated physical world. We already develop a physics engine that we plug in our model. This simulator, presented in details in (Cussat-Blanc et al., 2010), is linked to the chemical environment (*Cell2Organ*) and allows the simulation in a 3-D physical world of these developed organisms. We already showed the movement of a "muscular joint" where two "bones" rotate around a "kneecap" thanks to a "muscular fibre". All these components are produced by the developmental model and then linked in the physical world. Muscular fibre cells are able to change their shapes in order to produce a global movement. This kind of mechanism could be applied to the four-armed creature: each cell could be able to rotate around each other in order to produce a global movement of such a structure. With the intention of realising this behaviour, a high-level controller (neural network, classifier system, etc.) must be added to the cell to manage the rotation.

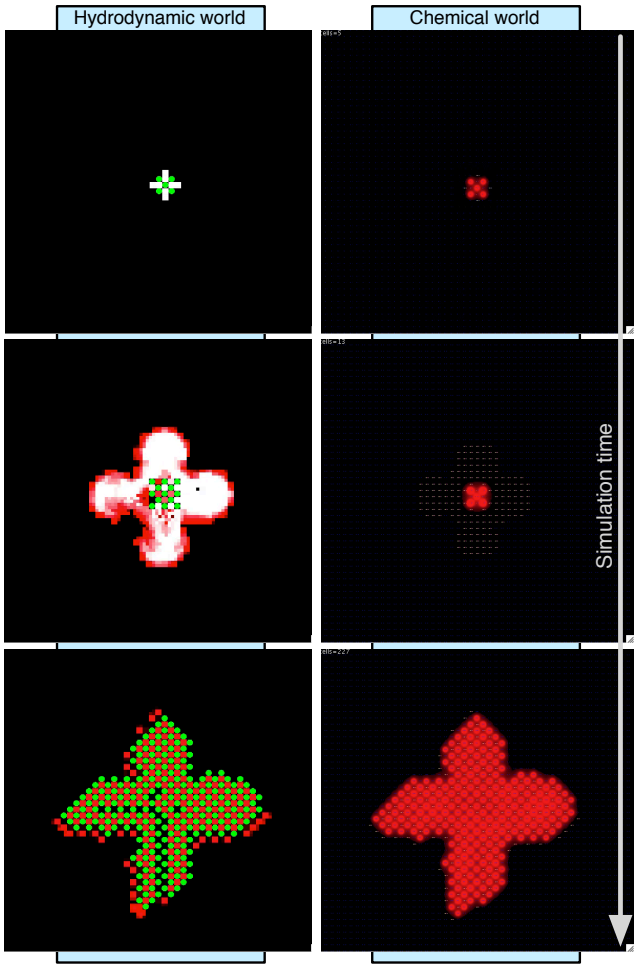


Figure 5: Development of a four-armed creature

Conclusion

In this paper, we have presented the last features added to our developmental model. We have plugged a hydrodynamic en-

gine to automatically position morphogens in the environment. This first stage prepares the environment by positioning morphogens in the environment. A creature can then develop its morphology by following division information. Thanks to this add-on, we develop various shapes, simple or more complex. The hydrodynamic model we choose for a simulation allows us an interesting parameterisation of fluid properties: whereas most models are hard to tune, Stam's model allows a simple modification of viscosity, density and forces applied to substrates. We show that several morphologies can be obtained.

This work can be improved in many ways. First, it could be interesting to evolve the presented parameter set with an evolutionary algorithm. The use of such a research algorithm could help us to produce user-defined morphologies just by giving a fitness function that describes the shape of the creature (that is a common problem in literature).

To produce more complex creatures, we imagine a cell differentiation inspired from nature: in real living systems, after a given number of divisions, embryonic stem cells can produce differentiated cells (neurons, epithelial cells, etc.). The mechanism could be used in our model to produce rotations or morphology modifications in creatures: a pusher cell produces an initial morphogenetic pattern. Developing cells have a given division credit to produce a shape. When this credit is depleted, the developing cell turns into a pusher cell that produces a new morphogenetic pattern. Surrounding developing cells continue the shape development following the previously produced pattern and so on. A grammar based on L-Systems could give the division credit and pusher parameters (expulsion force and direction) and could be evolved by an evolutionary algorithm in order to produce complex creature morphologies.

Lastly, as presented at the end of the previous section, creatures must also be simulated in a 3-D physical world to produce high-level moves. This feature will bring us closer to our goal: producing a creature from a single cell able to move in a 3-D environment.

References

Banzhaf, W. (2003). Artificial regulatory networks and genetic programming. *Genetic Programming Theory and Practice*, pages 43–62.

Caromel, D., Delbe, C., di Costanzo, A., and Leyton, M. (2006). Proactive: an integrated platform for programming and running applications on grids and p2p systems. *Computational Methods in Science and Technology*, 12.

Cartwright, J., Piro, O., and Tuval, I. (2009). Fluid dynamics in developmental biology: moving fluids that shape ontogeny. *Frontiers of Interdisciplinary Research in the Life Sciences*.

Chavoya, A. and Duthen, Y. (2008). A cell pattern generation model based on an extended artificial regulatory network. *Biosystems*, 94(1-2):95–101.

Cussat-Blanc, S., Luga, H., and Duthen, Y. (2008). From single cell to simple creature morphology and metabolism. In *Artificial Life XI*, pages 134–141. MIT Press, Cambridge, MA.

Cussat-Blanc, S., Luga, H., and Duthen, Y. (2009). Cell2organ: Self-repairing artificial creatures thanks to a healthy metabolism. In *Proceedings of the IEEE Congress on Evolutionary Computation (IEEE CEC 2009)*.

Cussat-Blanc, S., Pascalie, J., Luga, H., and Duthen, Y. (2010). Three simulators for growing artificial creatures. In *Proceedings of the IEEE Congress on Evolutionary Computation (IEEE CEC 2010), to be published*.

de Garis, H. (1999). Artificial embryology and cellular differentiation. In Peter J. Bentley, e., editor, *Evolutionary Design by Computers*, pages 281–295.

Dellaert, F. and Beer, R. (1994). Toward an evolvable model of development for autonomous agent synthesis. In *Artificial Life IV*, Cambridge, MA. MIT press.

Fleury, V. (2009). Clarifying tetrapod embryogenesis, a physicist's point of view. *The European Physical Journal Applied Physics*, 45(3):30101–30101.

Hotz, P. (2003). Genome-physics interaction as a new concept to reduce the number of genetic parameters in artificial evolution. In *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, volume 1.

Joachimczak, M. and Wróbel, B. (2009). Evolution of the morphology and patterning of artificial embryos: scaling the tricolour problem to the third dimension. In *10th European Conference on Artificial Life (ECAL09)*. Springer Verlag.

Kauffman, S. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theorical Biology*, 22:437–467.

Knabe, J., Schilstra, M., and Nehaniv, C. (2008). Evolution and morphogenesis of differentiated multicellular organisms: autonomously generated diffusion gradients for positional information. *Artificial Life XI*, 11:321.

Kodjabachian, J. and Meyer, J. (1998). Evolution and development of neural controllers for locomotion, gradient-following, and obstacle-avoidance in artificial insects. *IEEE Transactions on Neural Networks*, 9(5):796–812.

Kumar, S. and Bentley, P. (2003). Biologically inspired evolutionary development. *Lecture notes in computer science*.

Mouret, J. and Doncieux, S. (2008). Incremental Evolution of Animats' Behaviors as a Multi-objective Optimization. *Lecture Notes in Computer Science*, 5040:210–219.

Stam, J. (2003). Real-time fluid dynamics for games. In *Proceedings of the Game Developer Conference*, volume 18.

Stanley, K. and Miikkulainen, R. (2003). A taxonomy for artificial embryogeny. *Artificial Life*, 9(2):93–130.

Stewart, F., Taylor, T., and Konidaris, G. (2005). Metamorph: Experimenting with genetic regulatory networks for artificial development. In *ECAL'05*, pages 108–117.

Walker, M. (2004). Comparing the performance of incremental evolution to direct evolution. In *Second International Conference on Autonomous Robots and Agents*, pages 119–124.

Critical Properties of Complex Fitness Landscapes

Bjørn Østman[1], Arend Hintze[1], and Christoph Adami[1]

[1]Keck Graduate Institute, Claremont, CA 91711
bostman@kgi.edu

Abstract

Evolutionary adaptation is the process that increases the fit of a population to the fitness landscape it inhabits. As a consequence, evolutionary dynamics is shaped, constrained, and channeled, by that fitness landscape. Much work has been expended to understand the evolutionary dynamics of adapting populations, but much less is known about the structure of the landscapes. Here, we study the global and local structure of complex fitness landscapes of interacting loci that describe protein folds or sets of interacting genes forming pathways or modules. We find that in these landscapes, high peaks are more likely to be found near other high peaks, corroborating Kauffman's "Massif Central" hypothesis. We study the clusters of peaks as a function of the ruggedness of the landscape and find that this clustering allows peaks to form interconnected networks. These networks undergo a percolation phase transition as a function of minimum peak height, which indicates that evolutionary trajectories that take no more than two mutations to shift from peak to peak can span the entire genetic space. These networks have implications for evolution in rugged landscapes, allowing adaptation to proceed after a local fitness peak has been ascended.

Introduction

The structure of the fitness landscapes that populations find themselves in determines to a large extent how those populations will evolve. In introducing the concept of an adaptive fitness landscape, Sewall Wright (1932) sought to illustrate the idea that some combinations of characters will give rise to very high fitness (peaks) while some others do not (valleys), and to study the processes that allow a population to shift from peak to peak. Evolution in simple smooth landscapes (where each site or locus contributes independently to fitness) is trivial, because the ascent of a single fitness peak is largely deterministic (Tsimring et al., 1996; Kessler et al., 1997). At the other extreme lie "random" landscapes (Derrida and Peliti, 1991; Flyvbjerg and Lautrup, 1992), which are characterized by an absence of any fitness correlations between genotypes, and whose dynamics can likewise be solved using statistical approaches. In between these two extremes lie fitness landscapes that are neither smooth nor random, where mutations at different loci interact in complex patterns, giving rise to variedly rugged and highly *epistatic* landscapes (Whitlock et al., 1995; Burch and Chao, 1999; Phillips et al., 2000; Beerenwinkel et al., 2007; Phillips, 2008). Experiments with bacteria and viruses (Elena and Lenski, 2003) have revealed that real fitness landscapes are of this nature: they are neither smooth nor random, and consist of a large number of fitness peaks.

Unfortunately, while experiments with bacteria and viruses have taught us a lot about evolutionary dynamics, they can only probe very limited regions of the fitness landscape, confined to the genotype space surrounding those of living organisms. In artificial landscapes we are not constrained by generation time or the specific genotypic space that organisms happen to occupy, but can place organisms anywhere in the fitness landscape, thus enabling us to examine the statistical properties of fitness landscapes.

If realistic fitness landscapes are neither smooth (a single peak) nor random (very many randomly placed peaks in the landscape), what is the structure of complex landscapes in "peak space"? Are most peaks confined to one region of genotype space, leaving other areas empty? Are peaks clustered or are they evenly distributed? One hypothesis about the structure of fitness landscapes was proposed by Kauffman (1993), who posited that peaks are not evenly distributed, but that high peaks are correlated in space, forming a *Massif Central*, and presented numerical evidence supporting this view. According to this observation, the best place to look for a high fitness peak is near another high fitness peak. A corollary to this hypothesis is that large basins with no peaks surrounds the central massif. If fitness peaks are indeed distributed in this manner, it would have profound implications for the *traversability* of the landscape, and for evolvability in general (Altenberg and Wagner, 1996).

Here we strive to study this question in much more detail, by analyzing all the peaks in a landscape in which the ruggedness can be tuned from smooth to random. In particular, we would like to know whether the highest peaks form clusters of connected walks that can *percolate*, i.e., form connected clusters that span the entire fitness landscape. Such clusters are very different from the neutral net-

works studied elsewhere (van Nimwegen et al., 1999; Wilke, 2001), and we briefly argue that peak networks may be more important for evolvability.

NK Landscape

Kauffman's NK model (Kauffman and Levin, 1987, see also Altenberg, 1997) has been used extensively to study evolution because it is a computationally tractable model of N binary interacting loci where the ruggedness of the landscape can be tuned by varying K, the number of loci that each locus interacts with. Typically N is of the order of 10-30, but larger sets can be studied if a complete enumeration of genotypes is not necessary. If $K = 0$, the smooth landscape limit is reached, because if loci do not interact, then there is a single peak in the landscape that can be reached by optimizing each locus independently. If $K = N - 1$, on the other hand, the model reproduces the random energy model of Derrida (Derrida and Peliti, 1991). The N loci are usually thought of as occupying sites on a circular genome, while the interactions occur between *adjacent* sites (see Fig. 1), but the identity of the interactors are immaterial and the results do not depend on their physical location on the genome. The example genome in Fig. 1 shows the interactions between loci in an $N = 20$ and $K = 2$ model, where the width and darkness of the lines reflects the strength of the epistatic interactions between sites for the global peak of that landscape.

While clearly the NK model should not be thought of as describing the genome of whole organisms, the model has been used extensively to study the evolution of a smaller set of sites, such as the residues in a protein (Macken and Perelson, 1989; Perelson and Macken, 1995; Hayashi et al., 2006; Carneiro and Hartl, 2010) or the set of interacting genes coding for a pathway or a module (Kauffman and Weinberger, 1989; Sole et al., 2003; Yukilevich et al., 2008; Østman et al., 2010).

In the original NK model, the fitness contribution of each locus is calculated as the arithmetic mean of the fitness contributions of each locus $w(x_i)$, which itself is a function of the value of the bit at that locus ('1' if the gene is expressed, '0' if it is silent) and the allele of the K genes it interacts with. This fitness landscape is constructed by obtaining uniformly distributed independent random numbers for all the possible combinations of the $K+1$ sites (2^{K+1} numbers for each locus), so that the fitness contribution for any combinations of alleles can simply be found by looking up that value in the table. Here, we modify this model slightly, by replacing the customary arithmetic mean by the geometric one, so that the fitness of genotype $\vec{x} = (x_1, ..., x_N)$ is given by

$$W(\vec{x}) = \left(\prod_{i=1}^{N} w(x_i) \right)^{1/N} . \quad (1)$$

This modification better captures the nature of real genetic interactions (see, e.g., St Onge et al., 2007), and it makes it possible to introduce lethal mutations by setting one or more numbers in the fitness lookup-table to zero. Taking the geometric mean skews the distribution of genotype fitness to the left, resulting in a mean of about 0.4, rather than the value of 0.5 when using the arithmetic mean (see Fig. 2). Of course the logarithm of $W(\vec{x})$ reduces to the usual arithmetic mean of the log-transformed fitnesses.

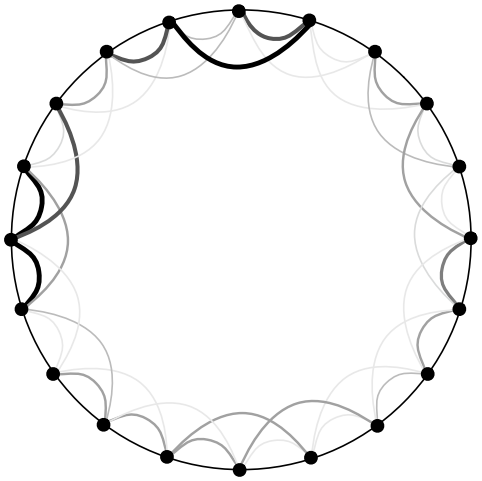


Figure 1: Genome and epistatic interactions between sites for the peak genotype of an $N = 20$ and $K = 2$ model. While all sites within a "radius" of two interact (light grey), the strength of interaction can be very different depending on the actual landscape that was formed. Here, the strength of epistatic interactions was calculated by performing all single-site and pairwise knockouts on the global peak genotype, and calculating the deviation of independence using a standard method (Bonhoeffer et al., 2004; Elena and Lenski, 1997; Østman et al., 2010).

In the NK model we can easily compute the fitness of all genotypes as long as N and K are not too large, and we can also identify *fitness peaks* as those genotypes whose N one-mutation neighbors all have lower fitness. Increasing K creates landscapes that are increasingly rugged, containing more and higher peaks with deeper valleys in between. The waiting time to new mutations becomes a determining factor in how much the population can evolve before it risks becoming stuck on a peak of suboptimal fitness. Visualizing natural fitness landscapes is difficult since it requires probing genotype-space by measuring the fitness of organisms whose genomes are fully sequenced. Even worse, natural fitness landscapes are rarely static, making such an endeavor even more futile. In computational models all genotypes can sometimes be enumerated, and we can thus learn about the global properties of the fitness landscape. This exciting possibility is muted by the fact that we cannot easily visualize high-dimensional spaces, and we are forced to resorting to

statistical methods to probe the landscape.

How Peaks Cluster

In Fig. 2 we show the fitness distribution of all genotypes of an $N = 20, K = 4$ landscape (this distribution is virtually identical for different realizations of landscapes with the same N and K). Of those 2^{20} genotypes, less than 0.07% are peaks (this fraction depends on the particular realization of the landscape), and are also roughly normally distributed in fitness. Note that while the highest-fitness genotypes are very likely peaks, there are peaks whose fitness is significantly smaller, down to the mean fitness of genotypes in the landscape. The number of peaks scales approximately exponentially with N (when K is fixed), but only about linearly with K for K sufficiently large, and at fixed N (data not shown).

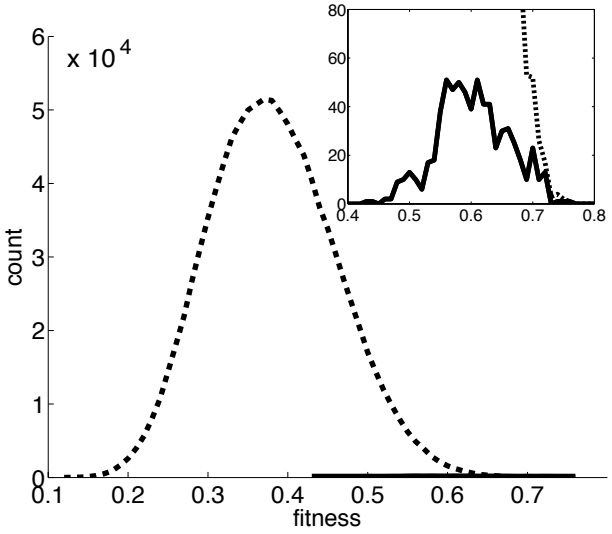


Figure 2: Fitness distribution of all $1,048,576$ genotypes (dashed line) in a typical landscape of $N = 20$ and $K = 4$. This landscape contains 679 peaks whose fitness distribution is shown as a solid black line. In the inset we have zoomed in on the peaks.

Pairwise distances

Because the "Massif Central" hypothesis says that the neighborhoods of high peaks are the best places to look for other high peaks, it is natural to also look at the pairwise distance of all peaks in a landscape. As we now know the genotypes of all the peaks in the landscape, we can ask whether peaks have a tendency to be located close to each other by studying the distribution of *Hamming distances* between peaks, which counts the number of differences in the binary representation of the sequences. In fact, this is how Kauffman validated his hypothesis: by plotting the fitness of peaks as a function of the Hamming distance of all peaks to the highest peak he found (Kauffman (1993), page 61), for a landscape with $N = 96$ and $K = 2$, 4, and 8. As it is not possible to enumerate $2^{96} \approx 8 \cdot 10^{28}$ genotypes, Kauffman found high peaks using random uphill walks. Here, we instead use $N = 20$, for which we can compute the fitness of all genotypes and thus locate all peaks. After computing the Hamming distance between all pairs of peaks, we can compare the distribution of these distances to a control distribution constructed with the same number of *random* genotypes, which are not expected to show any bias in the distribution of distances. (It is easy to see that the distribution of pairwise distances of random binary sequences of length $N = 20$ peaks at $d = 10$.)

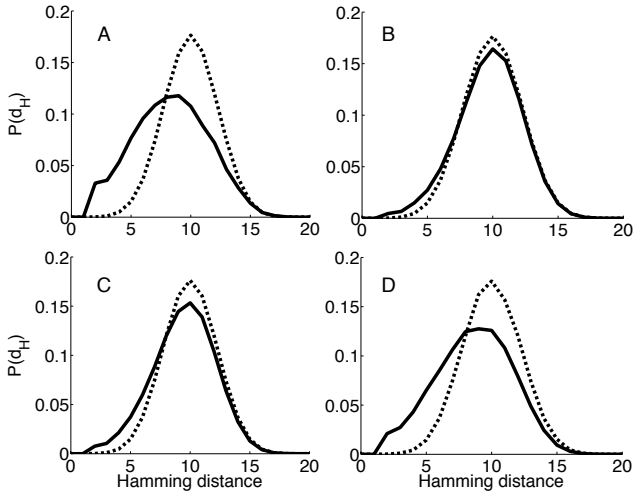


Figure 3: Distributions of pairwise Hamming distances between all peaks (solid) and between random "control" genotypes (dashed). The distributions shown are the averages of 50 different landscapes with genomes of length $N = 20$. (A) $K = 2$ landscapes containing an average of 98 peaks. (B) $K = 4$ landscapes containing an average of 720 peaks. (C) $K = 4$ landscapes including only an average of 363 peaks with a fitness above a threshold: $W \geq \Theta = 0.60$. (D) $K = 4$ landscapes including only an average of 95 peaks with a fitness above a threshold of $\Theta = 0.66$. As the samples include fewer and higher peaks, the pairwise distributions of $K = 4$ landscapes begin to resemble that of the $K = 2$ landscapes, suggesting that the highest peaks do cluster in genotype space, whereas the distribution of lower peaks is less biased.

We find that for $K = 2$, peaks are generally closer to each other than expected, indicating that peaks cluster in genotype space (see Fig. 3A). This alone does not tell us whether high peaks are more frequently associated with other high peaks (as opposed to peaks of lower fitness). Moreover, when examining $K = 4$ landscapes (that contain over seven

times as many peaks on average as for $K = 2$) we notice that the tendency for peaks to cluster close to each other is nearly gone, that is, the distribution closely resembles the random control (Fig. 3B). However, the bias reappears when we filter the peaks so that we only include those of high fitness (Figs. 3C and D), reaffirming the hypothesis that in complex epistatic landscapes, there is something special about being a high peak, genotypically speaking.

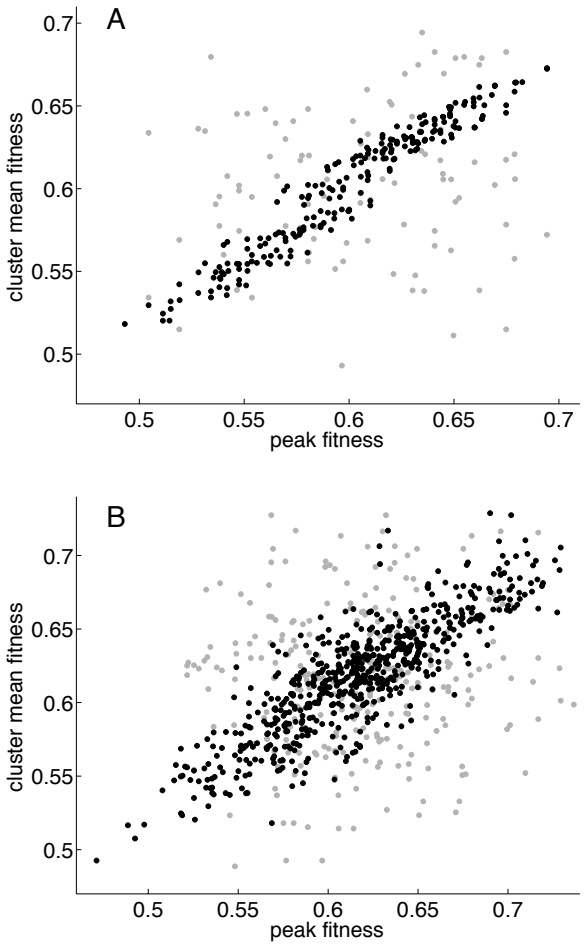


Figure 4: Mean fitness of peaks in circular clusters of radius $d = 2$ as a function of the fitness of the peak in the center of the cluster. (A) One landscape of $K = 2$ with 166 peaks (black dots). All landscapes show a strong correlation between cluster mean fitness and peak fitness, while the same analysis of assigning random genotypes to the peaks (but keeping the fitness) shows no such correlation (gray dots). The random data are from ten samplings. (B) One landscape of $K = 4$ with 679 peaks (black dots), and random genotypes (gray dots) obtained by sampling four times.

Peak neighborhood

If we want to know whether peaks with high fitness are likely to be found near other such peaks, we should study the mean fitness of peaks within a specified radius of that peak. These "circular" clusters contain all peaks within a Hamming distance d of a chosen peak (not counting the peak at the center). For the smallest possible distance between peaks $d = 2$, the size of a cluster is limited to 210 genotypes, but since peaks must be at least two mutations away from each other, there can be at most 190 peaks within a Hamming distance of two.

Fig. 4A depicts the mean fitness of adjacent peaks in circular clusters of radius $d = 2$ (black dots, for $K = 2$), showing a tight correlation between peak fitness and average adjacent peak fitness that indicates that the immediate neighborhood of high peaks is populated by other peaks of high fitness. On the contrary, when we randomize the location of the 166 peaks in genotype space without changing their height, this relationship vanishes (light gray dots in Fig. 4A). For $K = 2$ random peaks are far apart, resulting in only very few peaks within a distance $d = 2$ of each other. The $K = 4$ landscape has four times as many peaks as the $K = 2$ landscape, and the effect persists (Fig. 4B). The observed relation between mean fitness of these circular clusters and peak fitness persists even when the radius in increased to $d = 6$ (data not shown). We observe a similar correlation between mean cluster fitness and maximum peak height in network clusters (data not shown).

Adjacency matrices

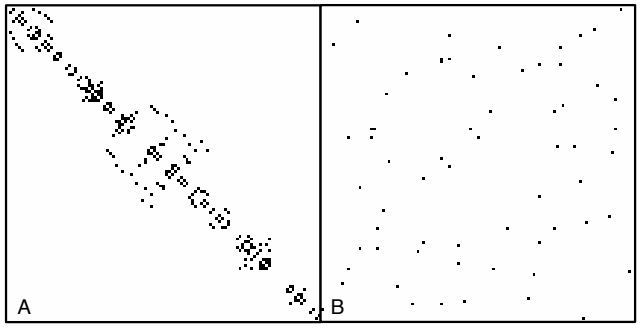


Figure 5: Adjacency matrices showing clusters of peaks. (A) Single $K = 4$ landscape with peaks of Hamming distance $d = 2$ connected. The peaks are ordered according to which network cluster they belong to. This landscape consists of 109 peaks with fitness above $\Theta = 0.66$ that are grouped into nine clusters (not counting singletons). (B) Random $K = 4$ landscape with $d = 4$ and $\Theta = 0$, showing only the first 109 genotypes.

While circular clusters can tell us whether high peaks are surrounded by peaks that are higher than expected, they do

not allow us to examine certain critical properties of the landscape. To do this, we should think of peaks in the genetic landscape as nodes in a random graph, and study the size of clusters of peaks that are formed by connecting all those peaks that are within a distance d of each other. Connecting such *networks clusters* of peaks creates a *percolation problem* (see, e.g., Bollobas and Riordan (2006)). In statistical physics, systems where nodes are connected by edges that are placed with a fixed probability undergo a geometric phase transition as a function of the edge placement probability. One of the quantities studied in percolation theory is the size of the largest cluster, because this variable rises dramatically at the critical point so that it takes up most of the system once past the critical point. If the largest cluster takes up most of the nodes, the system is said to "percolate", which implies that the cluster spans the entire system (allowing you to walk across connected nodes from any part to any other in the system). We will study the percolation properties of the fitness landscape by using the peak height as the critical parameter. Clearly, if only the highest few peaks are considered the system is far from percolation, as these peaks are unlikely to be connected. But if the highest peaks are closer to each other than expected in a random control, then the peaks could percolate far earlier.

Let us begin by computing the Hamming distance between all pairs of peaks with fitness greater than Θ, and connect those peaks that are a distance of no more than d away from each other. In Fig. 5A, we show the *adjacency matrix* of clusters, which we obtained by placing a dot for every two peaks that are with a distance d (that is, immediately adjacent). Peaks are ordered in such a way that peaks that fall into the same cluster are placed next to each other. This procedure allows us to the visualize the structure of clustered peaks in the landscape. In contrast, if the same peaks are assigned random locations in the landscape, there is no apparent structure, and clusters of peaks are on average very small (Fig. 5B). For $K = 4$ and $d = 2$ very few peaks are connected in a random landscape, and because of this the adjacency matrix shown in Fig. 5B is for $d = 4$, and includes peaks of any height. Only the first 109 peaks are shown.

Percolation phase-transition

In Fig. 6 we show the average relative size of the largest network cluster as a function of the peak threshold Θ, defined as the ratio of the largest number of connected peaks with fitness above Θ to the total number of peaks in the landscape. The relative size of the largest connected component (also called the "giant cluster" in percolation theory) increases dramatically as the critical threshold is reached, much like the size of the giant component increases when the critical probability of edges is reached in percolation theory. But what is remarkable about this transition is that it only occurs because the high peaks in the landscape occur near other high peaks: if the peaks were not clustered, the largest network cluster size would not increase when we lower Θ, as is the case when we reassign peaks to random genotypes (gray lines in Fig. 6).

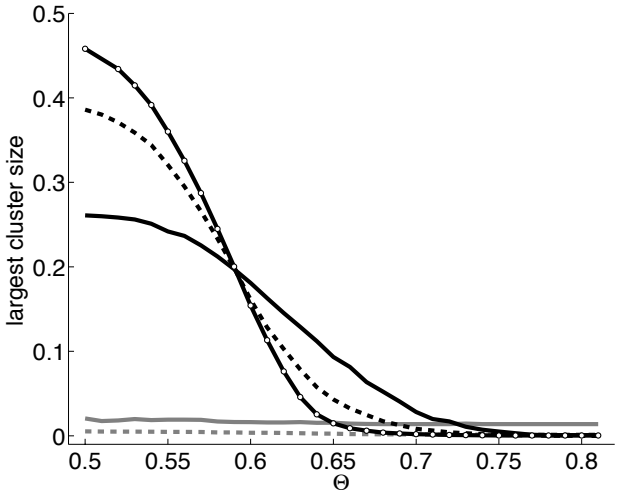


Figure 6: Size of the largest network cluster in the landscape averaged over 50 landscapes for each K as a function of fitness threshold, Θ. $K = 2$ (solid black line), $K = 4$ (dashed black line), and $K = 6$ (solid black line with white circles). The more rugged the landscapes are, the more abrupt the transition is from small network clusters to one cluster dominating the landscape. Random genotypes for $K = 2$ (solid gray line) and $K = 4$ (dashed gray line) show no increase in cluster size.

When we include enough peaks, either by setting Θ low for $K = 4$ (or else for $K = 6$ or higher) we find that for $d = 2$ there are always *two* largest network clusters, while the third largest cluster contains significantly fewer peaks. Both large clusters percolate genotype space and the diameter of both graphs is 18, not 20 (in general, $N-2$), while the shortest distance between the two clusters is always 3. This is peculiar to the way clusters are formed in this particular percolation problem. It is a rewarding exercise to determine the root cause of this peculiarity, which we leave to the interested reader. The transition seen in Fig. 6 suggests that in more rugged landscapes there are several clusters containing high peaks (high Θ), and that these high-peak clusters are connected by the peaks of lower fitness (lower Θ).

The percolation of genetic space by peaks with a sufficiently low height is reminiscent of the percolation of genetic space by arbitrary shapes in the RNA folding problem (Grüner et al., 1996), except that in that case structures with different genotypes form a neutral network that can be traversed by single point mutations. The giant cluster of peaks in the NK landscapes cannot be traversed like that: rather, it requires a minimum of two mutations to jump

from peak to peak, and because some of the peaks have inferior fitness, such mutations can only be tolerated for a finite amount of time–long enough to jump to the next highest peak. Thus, deleterious mutations are likely to be important to reach distant areas in genotype space, and the importance of these is slowly being realized (Lenski et al., 2003, 2006; Cowperthwaite et al., 2006; Østman et al., 2010).

Discussion

Using several methods we have shown that the rugged fitness landscapes that epistatic interactions create in the NK model consist of fitness peaks that are distributed in a manner that strongly affects evolution. High peaks are more likely to be found near other high peaks, rather than near lower peaks or far from peaks altogether. Similarly, lower peaks are predominantly located near each other in genotype space. Cluster analysis reveals that peaks tend to cluster (as compared to the same peaks placed randomly in genetic space) giving rise to large basins of attraction that are effectively devoid of peaks. This feature is especially prominent for moderately rugged landscapes ($K = 2$), while the addition of many more smaller peaks in more rugged landscapes ($K = 4$ or higher) makes this trend less significant. To the extent that we think that the NK landscape is an accurate model for real fitness landscapes of proteins and genetic pathways or modules, the discovery that these landscapes possess a remarkable structure that appears to be conducive to adaptation is highly informative about the process of evolution. Clustering of peaks makes a difference when the environment changes in a way that is unfavorable to the population, and forcing the population to adapt anew. If the landscape consists of evenly distributed peaks, then the risk of becoming stuck on a low fitness peak is high, and the population risks extinction. On the other hand, if peaks are unevenly distributed, then the ascent of one peak may not be where adaptation ends, making it possible to locate the global peak or another high fitness peak.

The more rugged a landscape is, the more peaks it contains, and the larger the space of genotypes that the largest network cluster spans. In smooth landscapes with only one or a few peaks, populations can evolve from genotypes of low fitness and move across genotype space toward high fitness. In rugged landscapes, the population always risks becoming stuck on a suboptimal peak. However, networks of closely connected peaks that percolate genotype space may still make it possible to traverse the fitness landscape jumping from peak to peak (given a sufficiently high mutation rate). If peaks are evenly distributed in genotype space, the chance to jump from peak to peak and thereby eventually locate the global peak is virtually nil. It is important, however, to remember that there are limits to the realism of the NK landscape as a model of realistic genetic or protein landscapes. For example, it is known that a significant percentage of substitutions in proteins or mutations in genetic pathways are neutral, while the NK landscape has virtually no neutrality (even though most mutations do not change the fitness significantly). Neutrality plays an important role to enhance traversability, and will facilitate the transition between peaks so that deleterious mutations are not essential for the shift from peak to peak. However, one could maintain that deleterious mutations are more promising for adaptation than neutral mutations are, because they may be what separate important phenotypes (Lenski et al., 2006).

The observation that peaks form clustered networks, and that these networks percolate, implies that the risk of becoming stuck on a suboptimal peak is significantly mitigated, because all it takes is the two right mutations to locate a new peak. Thus, it appears that evolvability comes for free in complex rugged landscapes of interacting loci. We should note, however, that the reason *why* peaks cluster in landscapes with epistatic interactions is not immediately apparent, and is a subject of ongoing investigations.

Acknowledgements

The authors thank Nicolas Chaumont for contributing code. This work was supported by the National Science Foundation's Frontiers in Integrative Biological Research grant FIBR-0527023.

References

Altenberg, L. (1997). Nk fitness landscapes. In Back, T., Fogel, D., and Michalewitcz, Z., editors, *The Handbook of Evolutionary Computation*, New York. Oxford University Press.

Altenberg, L. and Wagner, G. P. (1996). Complex adaptations and the evolution of evolvability. *Evolution*, 50:967–976.

Beerenwinkel, N., Pachter, L., and Sturmfels, B. (2007). Epistasis and shapes of fitness landscapes. *Statistica Sinica*, 17:1317–1342.

Bollobas, B. and Riordan, O. (2006). *Percolation*. Cambridge University Press, Cambridge, UK.

Bonhoeffer, S., Chappey, C., Parkin, N. T., Whitcomb, J. M., and Petropoulos, C. J. (2004). Evidence for positive epistasis in HIV-1. *Science*, 306:1547–1550.

Burch, C. L. and Chao, L. (1999). Evolution by small steps and rugged landscapes in the RNA virus ϕ6. *Genetics*, 151(3):921–927.

Carneiro, M. and Hartl, D. L. (2010). Adaptive landscapes and protein evolution. *Proc Natl Acad Sci U S A*, 107 Suppl 1:1747–1751.

Cowperthwaite, M. C., Bull, J. J., and Meyers, L. A. (2006). From bad to good: Fitness reversals and the ascent of deleterious mutations. *PLoS Computational Biology*, 2:1292–1300.

Derrida, B. and Peliti, L. (1991). Evolution in a flat landscape. *Bulletin of Mathematical Biology*, 53:355–382.

Elena, S. F. and Lenski, R. (1997). Test of synergistic interactions among deleterious mutations in bacteria. *Nature*, 390:395–397.

Elena, S. F. and Lenski, R. E. (2003). Evolution experiments with microorganisms: the dynamics and genetic bases of adaptation. *Nat Rev Genet*, 4:457–469.

Flyvbjerg, H. and Lautrup, B. (1992). Evolution in a rugged fitness landscape. *Physical Review*, A 46:6714–6723.

Grüner, W., Giegerich, R., Strothmann, D., Reidys, C., Weber, J., Stadler, I. L. H. P. F., and Schuster, P. (1996). Analysis of RNA sequence structure maps by exhaustive enumeration II. structures of neutral networks and shape space covering. *Monatshefte für Chemie*, 127:375–389.

Hayashi, Y., Aita, T., Toyota, H., Husimi, Y., Urabe, I., and Yomo, T. (2006). Experimental rugged fitness landscape in protein sequence space. *PLoS One*, 1:e96.

Kauffman, S. and Levin, S. (1987). Towards a general theory of adaptive walks on rugged landscapes. *J Theor Biol*, 128(1):11–45.

Kauffman, S. A. (1993). *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press US.

Kauffman, S. A. and Weinberger, E. D. (1989). The NK model of rugged fitness landscapes and its application to maturation of the immune response. *J Theor Biol*, 141(2):211–245.

Kessler, D., Levine, H., Ridgway, D., and Tsimring, L. (1997). Evolution on a smooth landscape. *Journal of Statistical Physics*, 87:519–544.

Lenski, R. E., Barrick, J. E., and Ofria, C. (2006). Balancing robustness and evolvability. *PLoS Biology*, 4:e428.

Lenski, R. E., Ofria, C., Pennock, R. T., and Adami, C. (2003). The evolutionary origin of complex features. *Nature*, 423(6936):139–144.

Macken, C. A. and Perelson, A. S. (1989). Protein evolution on rugged landscapes. *Proc Natl Acad Sci U S A*, 86(16):6191–6195.

Østman, B., Hintze, A., and Adami, C. (2010). Impact of epistasis and pleiotropy on evolutionary adaptation. arxiv.org: arXiv:0909.3506v2.

Perelson, A. S. and Macken, C. A. (1995). Protein evolution on partially correlated landscapes. *Proc Natl Acad Sci U S A*, 92(21):9657–9661.

Phillips, P., Otto, S., and Whitlock, M. (2000). Beyond the average, the evolutionary importance of gene interactions and variability of epistatic effects. In Wolf, J., Brodie III, E., and Wade, M., editors, *Epistasis and the Evolutionary Process*, pages 20–38. Oxford University Press.

Phillips, P. C. (2008). Epistasis - the essential role of gene interactions in the structure and evolution of genetic systems. *Nature Reviews Genetics*, 9:855–867.

Sole, R. V., Fernandez, P., and Kauffman, S. A. (2003). Adaptive walks in a gene network model of morphogenesis: insights into the Cambrian explosion. *Int J Dev Biol*, 47(7-8):685–693.

St Onge, R. P., Mani, R., Oh, J., Proctor, M., Fung, E., Davis, R. W., Nislow, C., Roth, F. P., and Giaever, G. (2007). Systematic pathway analysis using high-resolution fitness profiling of combinatorial gene deletions. *Nat Genet*, 39(2):199–206.

Tsimring, L., Levine, H., and Kessler, D. (1996). RNA virus evolution via a fitness-space model. *Phys Rev Lett*, 76(23):4440–4443.

van Nimwegen, E., Crutchfield, J. P., and Huynen, M. (1999). Neutral evolution of mutational robustness. *Proc. Natl. Acad. Sci. USA*, 96:9716–9720.

Whitlock, M. C., Phillips, P. C., Moore, F. B.-G., and Tonsor, S. J. (1995). Multiple fitness peaks and epistasis. *Annu. Rev. Ecol. Syst.*, 26:601–29.

Wilke, C. O. (2001). Adaptive evolution on neutral networks. *Bulletin of Mathematical Biology*, 63:715–730.

Wright, S. (1932). The roles of mutation, inbreeding, crossbreeding, and selection in evolution. In *Proceedings of the Sixth International Congress on Genetics*, volume 1, pages 355–366.

Yukilevich, R., Lachance, J., Aoki, F., and True, J. R. (2008). Long-term adaptation of epistatic genetic networks. *Evolution*, 62(9):2215–2235.

Analysis of Gene Regulatory Network Motifs in Evolutionary Development of Multicellular Organisms

Lisa Schramm[1], Vander Valente Martins[2], Yaochu Jin[3,*] and Bernhard Sendhoff[4]

[1]Technische Universität Darmstadt, Karolinenplatz 5, 64289 Darmstadt, Germany
[2]Escola Politécnica da Universidade de São Paulo, Av. Prof. L. Gualberto, trav. 3, 380, 05508-970 São Paulo, Brazil
[3]University of Surrey, Guildford, Surrey, GU2 7XH, UK
[4]Honda Research Institute Europe, Carl-Legien-Str. 30, 63073 Offenbach, Germany
lschramm@rtr.tu-darmstadt.de

Abstract

Biological development is governed by gene regulatory networks (GRNs), although detailed genetic and cellular mechanisms remain unclear. By means of analyzing biological data, it is believed that some GRN motifs have played an important role in the evolution of biological development. In this work, we investigate in a computational model for development to verify if these motifs can also be evolved as in biology, which can not only help understand biological development and improve simulated evolution as well. The goal of the evolution is to evolve an elongated body plan using a cellular developmental model controlled by a GRN. We count the number of network motifs during the evolution and try to relate the changes of these network motifs to the fitness profile of the evolution. We find for the number of most motifs an increase in the beginning of the evolution and a decrease as the evolution proceeds. We hypothesize that at the beginning a high number different motifs is helpful for the evolution, however, motifs that are not used for the targeted development, i.e., an elongated body morphology in this work, will get lost later on. Finally, we examined two individuals before and after a fitness jump to analyze which genetic changes have contributed to the large fitness improvement.

Introduction

Recent advances in computational systems biology suggest that computational models for development may help us to gain more insights into the genetic and cellular mechanisms underlying biological development. Among other research efforts, analysis of small, frequently occurred network structures, often known as network motifs, have attracted much interest as described by Alon (2007, 2006). Analysis of biological data revealed that such motifs can widely be identified in bacteria and yeast, see e.g., Babu et al. (2004). Most recently, it has been found that some motifs may have played an essential role in evolution. For instance, Kwon and Cho (2008) analyzed the role of feedback loops and found that more positive feedback loops and less negative feedback loops contribute to the robustness of the regulatory system. However, the analysis of motifs on an evolutionary scale requires the data of many individuals from different evolutionary stages. These data are (currently) not available in biology. Therefore, it seems advisable to support the biological analysis with the results from computational models. Even though these models are usually abstract and the analysis is computationally expensive, it is the target to identify patterns that relate the emergence of motifs to the evolutionary progress in computational models.

Some computational models for artificial development have been proposed (see Harding and Banzhaf (2008)) based on various computational models of GRNs (de Jong, 2002; Geard and Willadsen, 2009). In models of artificial development, one or a few single cells divide and proliferate in a 2D or 3D environment. These cells interact with each other, developing into a pattern, a structure or a shape.

One major concern in cell-based developmental models under the control of GRNs is a self-stabilizing cell growth and the ability to self-heal after a damage. The French flag problem is a popular benchmark used in artificial development, see e.g., Joachimczak and Wròbel (2009); Wolpert (2004). Andersen et al. (2009) managed to evolve a stable development and demonstrate the capacity of self-repair using a GRN based on cellular automata. In their model, cells are fixed on a grid and contact inhibition is adopted, i.e., if a cell is surrounded by other cells, it will not divide any more.

In this work, we have used a cellular growth model described by Steiner et al. (2007), which was inspired by an artificial development model suggested by Eggenberger Hotz et al. (2003). We use a GRN network model that defines the actions of the cells. The cells interact with each other through diffusion of external transcription factors. In contrast to other work, our cells are not fixed on a grid and can move via cell-cell physical interactions. In addition, cells can divide as long as the gene for cell division is active. Therefore, the model has fewer assumptions and the developmental process is less constrained. This model has been employed for simulating neural development in a hydra-like animat (Jin et al., 2008). Stable cell growth has also evolved in a co-evolution of morphology and control of swimming

*The work was conducted while Yaochu Jin was with the Honda Research Institute Europe.

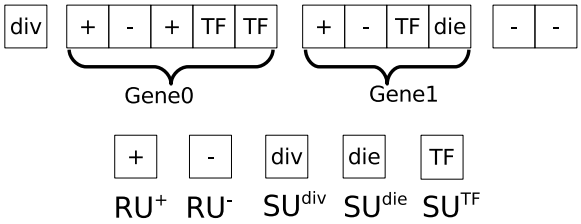


Figure 1: An example chromosome for the development.

animats (Schramm et al., 2009). Additionally, stable and lightweight structures have evolved in (Steiner et al., 2009) using this cellular model.

In (Steiner et al., 2007), the authors showed that the emergence of a negative feedback motif helps to enhance the mutational robustness. In this paper, we analyze the motifs of the GRNs in the best individuals of the whole evolutionary run to see how various network motifs have contributed to the evolution of cellular development. We examine the change in the number of motifs during evolution. Additionally, we analyze the difference in the structure of the GRNs of two related individuals before and after a fitness jump.

We describe our model in the next section followed by an introduction to the widely studied network motifs. Then we present the experimental results of the evolutionary runs together with the number of motifs during the evolution. We conclude the paper with an analysis of two individuals, a summary and an outlook.

The Computational Model for Morphological Development

The morphological development starts with a single cell that can perform a few cellular actions, e.g. cell division or cell death. The cell is placed in the center of a two-dimensional computation area of size 100×80, the cells are not fixed on a grid and can be at all positions inside the computation area. The cells interact physically with each other and can produce transcription factors (TFs) that are used for cell-cell communication. A gene regulatory network (GRN) defines the behavior of the cells.

The genes of the virtual DNA in each cell consist of regulatory units (RUs) and structural units (SUs), see Schramm et al. (2009) for details, as illustrated in Figure 1. The SUs of a gene define the cellular behaviors, in this paper cell division, cell death or the production of TFs. The RUs define whether a gene is activated (expressed). All RUs have an activation level depending on the TF concentrations inside and outside a cell. The activation of a gene is defined by a sum of the activation levels of its RUs, which can be activating (RU^+) or inhibiting (RU^-). If the difference between the affinity values of a TF and a RU is smaller than a predefined threshold ϵ (in this work ϵ is set to 0.2), the TF can bind to the RU to regulate the gene activation. The affinity similarity ($\gamma_{i,j}$) between the i-th TF and j-th RU is defined by:

$$\gamma_{i,j} = \max\left(\epsilon - \left|\text{aff}_i^{TF} - \text{aff}_j^{RU}\right|, 0\right). \quad (1)$$

If $\gamma_{i,j}$ is greater than zero, then the concentration c_i of the i-th TF is checked whether it is above a threshold ϑ_j defined in the j-th RU:

$$b_{i,j} = \begin{cases} \max(c_i - \vartheta_j, 0) & \text{if } \gamma_{i,j} > 0 \\ 0 & \text{else} \end{cases}. \quad (2)$$

Thus, the activation level contributed by the j-th RU (denoted by $a_j, j = 1, ..., N$) can be calculated as follows:

$$a_j = \sum_{i=1}^{M} b_{i,j}, \quad (3)$$

where M is the number of TFs that bind to the j-th RU. Assume the k-th gene is regulated by N RUs, the expression level of the gene can be defined by a summation of the activations of all RUs

$$\alpha_k = 100 \sum_{j=1}^{N} h_j a_j (2s_j - 1),\ s_j \in (0, 1). \quad (4)$$

$2s_j - 1$ denotes the sign (positive for activating and negative for repressive) of the j-th RU and h_j is a parameter representing the strength of the j-th RU.The k-th gene is activated if $\alpha_k > 0$ and its corresponding behaviors coded in the SUs are performed.

The SU for cell division (SU^{div}) encodes where the new cell is placed in comparison to the mother cell. A cell with an activated SU for cell death dies at the developmental timestep which it is activated. When SUs for both cell death and cell division are simultaneously active, the cell dies without division. Two additional SUs are reserved for other possible behaviors, which are not used in this work. As a result, it can happen that some genes perform no action.

An SU that produces a TF (SU^{TF}) also encodes all parameters related to the TF, such as the affinity value, the decay rate D_i^c, the diffusion rate D_i^f, as well as the amount A_i of the TF_i to be produced:

$$A_i(\alpha_k) = \begin{cases} \beta\left(\frac{2}{1+e^{-20 \cdot f \cdot \alpha_k}} - 1\right) & \text{if } \alpha_k > 0 \\ 0 & \text{otherwise} \end{cases}, \quad (5)$$

where f and β are both encoded in the SU^{TF}. Which TF_i is produced is defined in terms of the affinity value.

A TF produced by an SU can be partly internal and partly external. To determine how much of a produced TF is external, a percentage ($p^{\text{ext}} \in (0, 1)$) is also encoded in the corresponding gene. Thus, $\Delta c_i^{\text{ext}} = p^{\text{ext}} \cdot A_i$ is the amount of external TF to be produced and $\Delta c_i^{\text{int}} = (1 - p^{\text{ext}}) \cdot A_i$ is that of the internal TF.

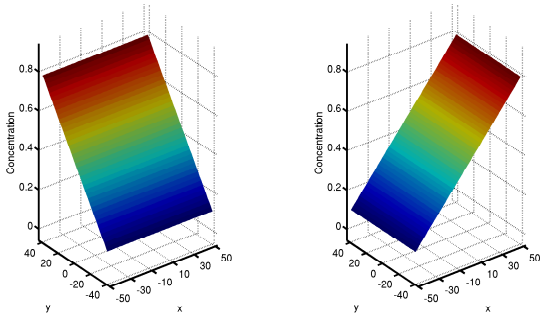


Figure 2: Concentrations of the prediffused TFs.

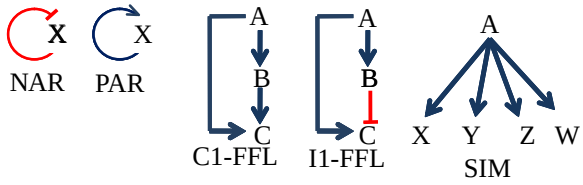


Figure 3: Network motifs (adapted from Alon (2006)).

External TFs are put on four grid points around the center of the cell. They undergo first a diffusion and then a decay process:

$$\text{Diffusion:}\quad \mathbf{u}_i^*(t) = \mathbf{u}_i(t-1) + 0.1 D_i^f \cdot (\mathbf{G} \cdot \mathbf{u}_i(t-1)), \quad (6)$$

$$\text{Decay:}\quad \mathbf{u}_i(t) = ((1 - 0.1 D_i^c)\, \mathbf{u}_i^*(t)), \quad (7)$$

where $\mathbf{u}_i$ is a vector of the concentrations of the i-th TF at all grid points and the matrix $\mathbf{G}$ defines which grid points are adjoining. The internal TFs underlie only a decay process:

$$c_i^{\text{int}}(t) = (1 - 0.1 \cdot D_i^c)\, c_i^{\text{int}}(t-1). \quad (8)$$

All internal and external concentrations of TFs are limited to an interval of $[0, 1]$.

In our experiments, we put two prediffused, external TFs without decay and diffusion in the computation area. The first TF (preTF00) has a constant gradient in the y-direction and the second (preTF01) in x-direction (see Figure 2 and Figure 13).

Static and Dynamic Network Motifs

Network motifs are sub-networks that occur more often in biological gene regulatory networks than expected at random. In this work, we analyze the occurrence of different types of regulatory motifs, such as autoregulation, feed-forward-loops and single input modules, see Figure 3. In the following, we describe the function of a few network motifs, as described in Alon (2006, 2007):

- **Negative autoregulation (NAR)** defines a gene whose product directly inhibits its own expression. Such motifs can speed up the response time compared to a gene without NAR with the same steady state. It leads to steady states with a rapid rise and a sudden saturation. NAR also promotes robustness.
- The **positive autoregulation (PAR)** slows down the response time and can lead to bi-stability.
- The **coherent feed-forward loop 1 (C1-FFL)** results in a fast convergence to a steady state but a slow decrease of the concentration.
- The **incoherent feed forward loop 1 (I1-FFL)** can act as a pulse generator. It can turn a concentration very fast on with an overshoot, and then it converges to its steady state.
- The **Single input module (SIM)** consists of one gene regulating many other genes. Temporally sequential cellular events can be controlled with a SIM.

There are a lot of different FFLs, among which C1-FFL and I1-FFL are the most frequent ones in E. coli and yeast. The functional analysis described above is performed on isolated motifs, and therefore their behavior in a whole network can be very different.

All possible connections of a GRN define the *static network*. Therefore, the *static network motifs* are all possible network motifs regardless of whether they are actually used during cell operations. In this paper, we want to analyze only the network connections that are really used during development, which constitute the *dynamic network*. The related motifs are then termed the *dynamic network motifs*. In order for a static motif to be counted as a dynamic motif, all motif connections have to have been activated (above the threshold) in at least one cell at anytime during development. Thus the dynamic motif must play an active role during cell operations and not just a potential role as the static motif. Of course dynamic motifs are a subset of static motifs.

Experimental Settings

We use an extended evolution strategy, (μ, λ)-ES with elitism for evolving the developmental model, where μ and λ are parent and offspring population size, respectively (Beyer and Schwefel, 2002). In this work, $\mu = 30$, $\lambda = 200$, and 3 elitists are adopted. The strategy parameter σ is fixed to $\sigma = 10^{-4}$ in our work.

in addition to mutation, we use gene duplication, gene transposition and gene deletion as genetic variations. Gene duplication randomly copies a sequence of RUs and SUs in the chromosome and then inserts it, again randomly, into the chromosome. In the case of gene transposition or deletion, this randomly picked out sequence of RUs and SUs is moved to another randomly chosen site on the chromosome, or simply removed.

Mutation is always performed, while gene duplication, transposition and deletion are exclusive, i.e., only one of

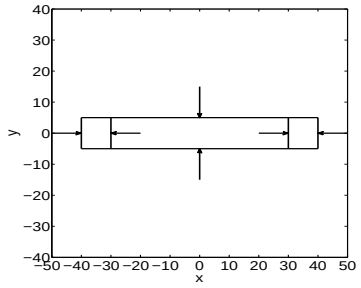


Figure 4: The target shape for the cellular growth model.

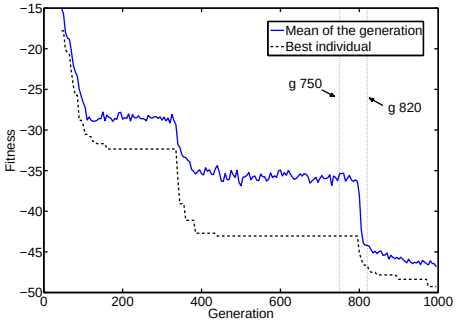


Figure 5: Fitness curves of the analyzed evolutionary run. Solid line: mean of the generation. Dotted line: best individual.

them can be performed to the same chromosome in one generation. The probabilities for gene duplication, gene transposition, and gene deletion are $p_{dup} = 0.05$, $p_{trans} = 0.02$, and $p_{del} = 0.03$. These values are not particularly motivated, however, the algorithm is not sensitive to the choice of probabilities.

The goal of the evolution is to obtain an elongated shape resulting from the cell growth process controlled by the GRN. To this end, we define a target shape, as described in Figure 4. The target shape has an approximated width-to-height ratio of $a : b$, which in the experiment, we set $a_{max} = 10$, $b_{min} = 60$ and $b_{max} = 80$. Thus, the fitness function can be defined as follows:

$$f = p_1 - p_2 - \min\left\{\min_i\left\{\boldsymbol{x}^i(1)\right\}, -\frac{a_{max}}{2}\right\} + \max\left\{\max_i\left\{\boldsymbol{x}^i(1)\right\}, \frac{a_{max}}{2}\right\}, \quad (9)$$

where $\boldsymbol{x}^i$ represents the position of the i-th cell and

$$p_1 = \begin{cases} 70 + \min_i\left\{\boldsymbol{x}^i(0)\right\} & \text{if } \min_i\left\{\boldsymbol{x}^i(0)\right\} < -\frac{b_{max}}{2} \\ -30 & \text{if} -\frac{b_{max}}{2} < \min_i\left\{\boldsymbol{x}^i(0)\right\} < -\frac{b_{min}}{2} \\ \min_i\left\{\boldsymbol{x}^i(0)\right\} & \text{otherwise} \end{cases} \quad (10)$$

and

$$p_2 = \begin{cases} 70 + \max_i\left\{\boldsymbol{x}^i(0)\right\} & \text{if } \max_i\left\{\boldsymbol{x}^i(0)\right\} > \frac{b_{max}}{2} \\ 30 & \text{if } \frac{b_{max}}{2} > \max_i\left\{\boldsymbol{x}^i(0)\right\} > \frac{b_{min}}{2} \\ \max_i\left\{\boldsymbol{x}^i(0)\right\} & \text{otherwise} \end{cases}. \quad (11)$$

To achieve a computationally tractable size of the body morphology, the number of cells (n_c) is constrained between 10 and 500. A penalty of $600 - n_c$ is applied if $n_c < 10$ and a penalty of n_c if $n_c > 500$. If the cells in the developed morphology are not fully connected, this means there exists one or several cells with a high distance to all other cells, a fitness of 50 is assigned.

Experimental Results

The best and mean fitness curves of an evolutionary run are presented in Figure 5. We can observe two fitness jumps

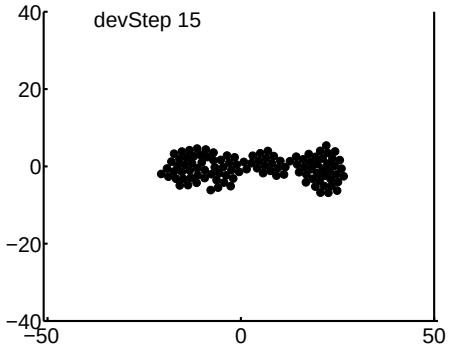


Figure 6: Resulting shape of the best individual.

around generations 350 and 800 during the whole evolution. The resulting shape of the best individual in the last generation is shown in Figure 6. The morphologies of the individuals of the first generations all result in either no cell or too many cells (we aborted the runs with more than 700 cells). In Figure 7 the total number of genes is shown. The number of genes is nearly constant, there is only one huge jump at the end of the evolution.

Dynamic Network Motifs

We count the different network motifs for all selected individuals every 5th generation. The motifs of the best individual and the mean of the parent generation are presented in Figures 8 - 11. Our algorithm counts all occurrences of one gene activating two others as one SIM (which is then a three node motif). When there is one gene activating more than two other genes, the algorithm counts more SIMs, according to the combinatorial possibilities $\binom{N}{2}$. E.g. for 4 genes

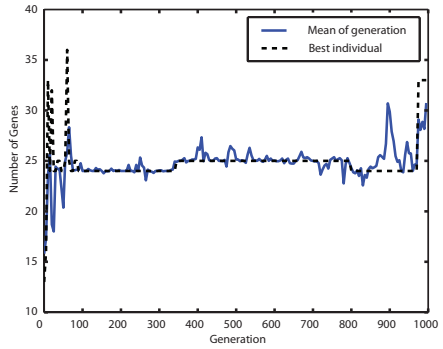
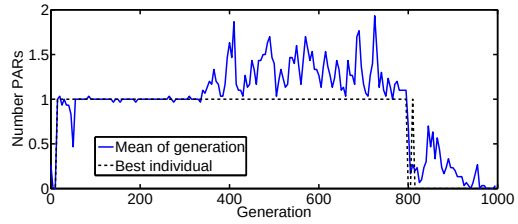
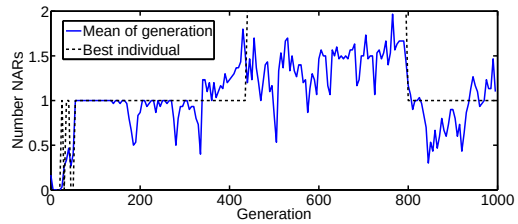


Figure 7: Number of genes of the best individual and the mean of the generation.



(a) Positive autoregulation (PAR)

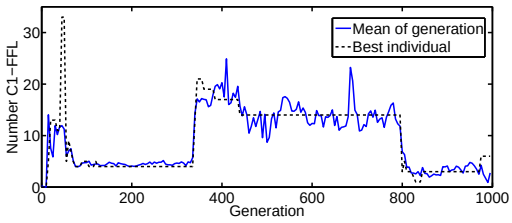


(b) Negative autoregulation (NAR)

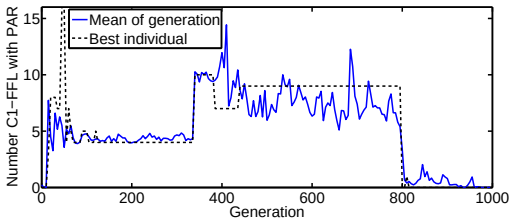
Figure 8: Number of autoregulations (AR).

our algorithm counts $\frac{4!}{2!(4-2)!} = 6$ SIMs. This masks on the one hand the number of SIMs, but on the other hand the size of the SIM is taken into account.

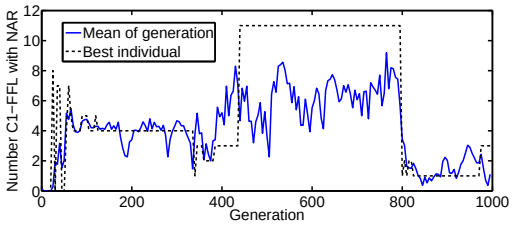
Regarding the number of most motifs, we find an increase in the beginning of the evolution and a decrease in later generations. An increase in the number of motifs is observed often between generation 300 and 500, while a considerable decrease of most motifs is observed around generation 800. The number of some motifs, e.g., I1-FFL, I1-FFL with NAR and SIM with NAR, increases again in the last generations, which can be explained with the increase in the number of genes (see Figure 7). The two large changes in the number of motifs correlate with two large fitness jumps. A change in the number of genes is not the reason, though the number of genes is nearly constant (see Figure 7). We hypothesize that on the one hand, evolution attempts to increase the number of motifs to perform better, whereas on the other hand, motifs that are not helpful are lost in later generations.



(a) C1-FFL



(b) C1-FFL with PAR

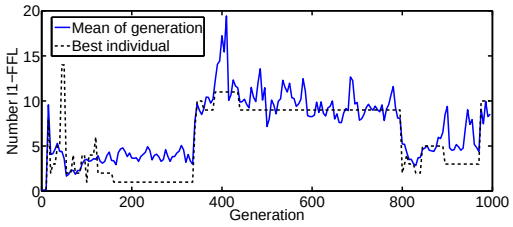


(c) C1-FFL with NAR

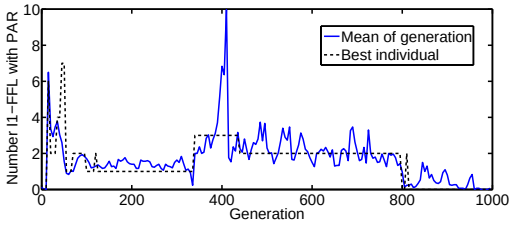
Figure 9: Number of coherent feed-forward loops (C1-FFL) with only activating connections.

In the following, we discuss in greater detail the change of the number of the motifs:

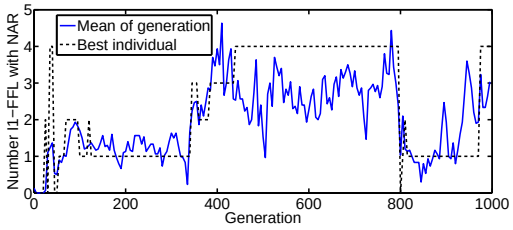
- **PAR**: One PAR exists in the best individual until generation 800, then the PAR is lost. On average over the generations, the number of PAR increases between generation 300 and 400 from about one to between one and two and becomes zero around generation 800. PAR seems to be important during evolution but is lost in later generations.
- **NAR:** The number of NARs is very low throughout the evolution. It starts from one, goes up to two at about generation 450 and falls back to one again at generation 800.
- The number of **C1-FFL**is high during the evolution compared to that of the PARs and NARs. There is a considerable increase of this motif between generation 300 and 400 and a decrease around generation 800. The numbers of **C1-FFL with PAR** and **C1-FFL with NAR** are smaller but have a similar trend as **C1-FFL**.
- The number of **I1-FFL** is very low at the beginning and also increases between generation 300 and 400 to about 10 and decreases again around generation 800. At the end of the evolution, there is again an increase in the number



(a) I1-FFL



(b) I1-FFL with PAR



(c) I1-FFL with NAR

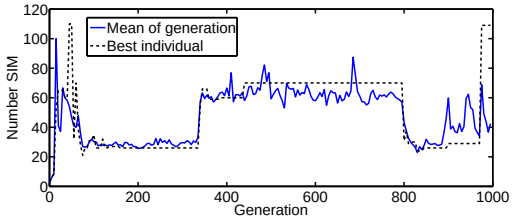
Figure 10: Number of incoherent feed-forward loops (I1-FFL) with one negative connection from B to C.

of this motif. The number of **I1-FFL with PAR** and **I1-FFL with NAR** is much lower than that of the **I1-FFL**.

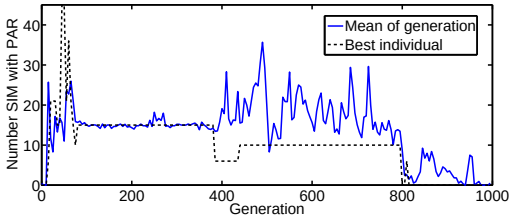
- The number of **SIMs** and **SIMs with NAR** is much higher than that of the other motifs. Note that we count all three-node SIMs, and consequently the larger the SIM, the more three node SIMs are counted. The change of SIMs during the evolution is comparable to that of the I1-FFL. The **SIM with PAR** is the only motif that decreases between generation 300 and 400, and reaches zero at generation 800 (because the PARs decrease to zero).

To relate the changes in the number of motifs to the occurrences of the genetic operators during evolution, including duplication, deletion or transposition, we traced back the ancestors of the best individual in the final generation and analyzed which genetic operators are selected over the generations. The results are given in Figure 12.

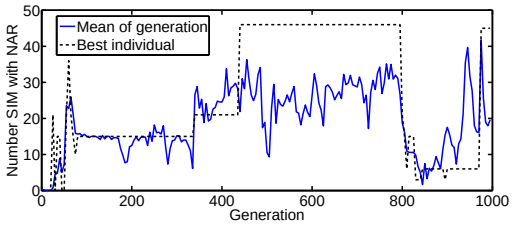
The gene deletion selected in generation 800 correlates with a strong fitness increase and a decrease of a lot of motifs. To better understand what happened during these generations, we analyze the best individual in generation 750 at the fitness plateau before the deletion and the best individual in generation 820 after the deletion in the next section.



(a) SIM



(b) SIM with PAR



(c) SIM with NAR

Figure 11: Number of single input modules (SIM) during the evolution. We count three nodes SIMs, so that larger SIMs result in a higher number of SIMs.

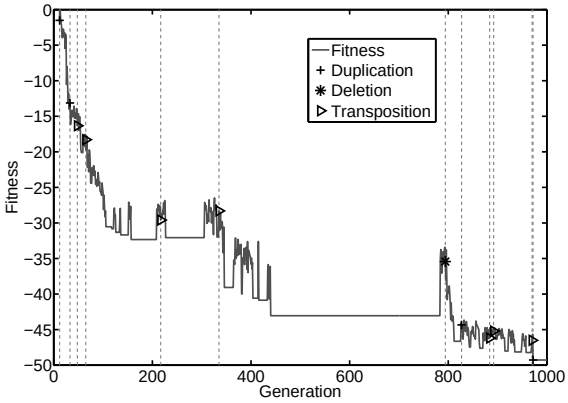


Figure 12: The fitness of the ancestors of the best individual in the last generation. Symbol '+' denotes a gene duplication, '*' a deletion and a triangle a gene transposition.

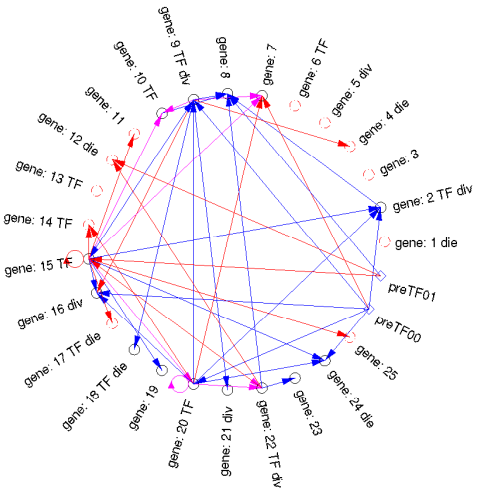


Figure 13: The genes and their used connections of the best individual in generation 750. The circles represent the different genes. Genes that are active during development are denoted with black (solid) circles. Red (dashed) circles indicate genes that are never active. The arrows represent the interactions between the genes, where blue represents an activating, red an inhibiting and magenta both an activating and inhibiting connections. The two diamonds represent the predefined TFs.

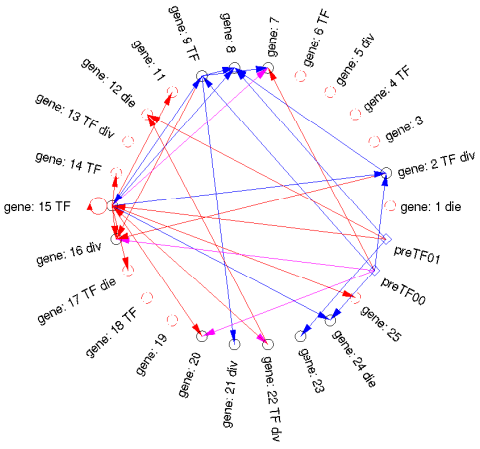


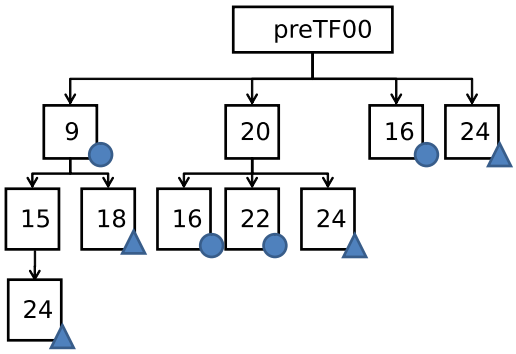
Figure 14: The genes and their used connections of the best individual 820. Notation as in Figure 13. The genes are numbered, and number 10 is skipped for an easier comparison betwen the two individuals of generation 750 and 820, because gene 10 was deleted in between.

Detailed analysis of two individuals

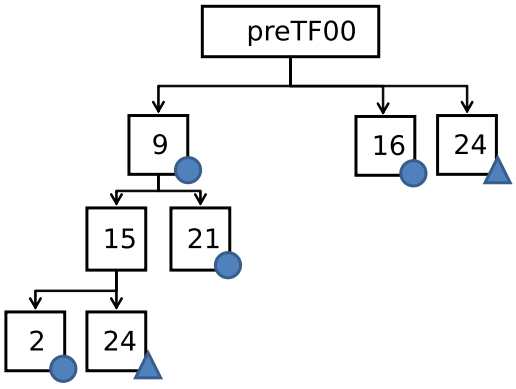
The genes and their activations of the best individual in generation 750 and 820 are presented in Figure 13 and 14.

Note that only the dynamic activations are shown, and there are much more static activations.

The deleted regulatory and structural units belong to genes 9 and 10 of the best individual of generation 750. The SU for cell division of gene 9 and the complete gene 10 are deleted. We skipped gene number 10 in the second individual to ease the comparison of the two individuals. Another difference is that the SU of gene 20 of the best individual in generation 750 changes from TF production to an unused SU through mutation. Though gene 10 of the best individual in 750 has no further influence on the development (no arrows starting from this gene in Figure 13), the more important change seems to be the mutation of gene 20. Figure 15 shows the activations of the different genes in temporal hierarchies. The inhibitions are not shown and the inactivated genes are omitted. There are only temporal hierarchies and one feedback loop. The mutation to gene 20 resulted in a deletion of the whole sub-tree. The deletion of gene 9 has no further effect on the development. Gene 20 in the best individual of generation 750 has a lot of connections to other genes and is a member of a lot of motifs. Interestingly, the loss of gene 20 resulted in an increase in fitness from generation 750 to generation 820.



(a) Individual 750_0



(b) Individual 820_0

Figure 15: The activating relations of the different genes. Genes for cell division are marked with a circle, genes for cell death with a triangle. Only some important activating effects are shown, inactivated genes and inhibiting connections are omitted.

Summary and Conclusion

In this work, we have analyzed the change in the number of network motifs in the gene regulatory network during evolution of a cell growth model for an elongated body morphology. A general trend is that the overall number of motifs increases significantly at the beginning of evolution. During the evolutionary process the numbers of all motifs have increased with the exception of PAR.

Since the genome length does not change significantly during evolution, it seems that it is not just the increase of genetic material but of structured genetic material, i.e., dynamic network motifs, that is important during the evolutionary process. At the same time, motifs that do not influence development are lost again during evolution. Therefore, it seems that the frequency of motifs is under selectional control and that the increase of dynamic network motifs is related to the evolvability of the process.

We analyzed the genetic changes that contributed to the fitness jump around generation 800 and compared the genes of two individuals before and after the genetic change. We found that the fitness increase and the decrease of the number of dynamic motifs were due to one mutation that changed a gene from producing an important TF to a gene without function. Contrary to intuition, the correlated gene deletion neither influenced the fitness nor the number of motifs.

A more detailed interpretation of our results is restricted by the fact that only observations from one experiment are available. Needless to say that a more statistically sound analysis would be desirable, however, the considerable computational expense of the described process makes it difficult to run a larger number of experiments.

For the analysis of static motifs, other authors have normalized their results to the motifs one can find in random networks (Kashtan and Alon, 2005). For dynamic motifs this is difficult, because most static motifs in random networks will not be dynamical, simply because the developmental process terminates very early. Frequently, this is due to the early activation of cell death by a prediffused TF in random networks. More precise, most random networks result in an activation of cell death in the first developmental step and the development stops. This results in nearly no network motifs, because no TFs are produced. In order to make sure that during the evolutionary process not just the raw genetic material is increased we compared the number of dynamic motifs to the genome length during evolution.

Acknowledgements

We thank Nazli Bozoglu for her work on static analysis of network motifs and Till Steiner for inspiring discussions. This work was supported by the Honda Research Institute Europe.

References

Alon, U. (2006). *An Introduction to Systems Biology: Design Principles of Biological Circuits*. Chapman & Hall.

Alon, U. (2007). Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, 8:450–461.

Andersen, T., Newman, R., and Otter, T. (2009). Shape homeostasis in virtual embryos. *Artificial Life*, 15(2):161–183.

Babu, M. M., Luscombe, N. M., Aravind, L., Gerstein, M., and Teichmann, S. A. (2004). Structure and evolution of transcriptional regulatory networks. *Current Opinion in Structural Biology*, 14(3):283–291.

Beyer, H.-G. and Schwefel, H.-P. (2002). Evolution strategies –a comprehensive introduction. *Natural Computing: an international journal*, 1(1):3–52.

de Jong, H. (2002). Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9(1):67–103.

Eggenberger Hotz, P., Gómez, G., and Pfeifer, R. (2003). Evolving the morphology of a neural network for controlling a foveating retina - and its test on a real robot. In *The Eighth International Conference on Artificial Life*, pages 243–251. MIT Press.

Geard, N. and Willadsen, K. (2009). Dynamical approaches to modeling developmental gene regulatory networks. *Birth Defects Research Part C*, 87(2):131–142.

Harding, S. and Banzhaf, W. (2008). Artificial development. In Würz, R. W., editor, *Organic Computing*, chapter 9, pages 201–219. Springer.

Jin, Y., Schramm, L., and Sendhoff, B. (2008). A gene regulatory model for the development of primitive nervous systems. In *Proc. of ICONIP 2008*, pages 48–55. Springer.

Joachimczak, M. and Wròbel, B. (2009). Evolution of the morphology and patterning of artificial embryos: scaling the tricolour problem to the third dimension. In *Proc. of ECAL 09*.

Kashtan, N. and Alon, U. (2005). Spontaneous evolution of modularity and network motifs. *Proceedings of the National Academy of Sciences*, 102(39):13773–13778.

Kwon, Y. K. and Cho, K. H. (2008). Quantitative analysis of robustness and fragility in biological networks based on feedback dynamics. *Bioinformatics*, 7(24):987–994.

Schramm, L., Jin, Y., and Sendhoff, B. (2009). Emerged coupling of motor control and morphological development in evolution of multi-cellular animats. In *Proc. of ECAL 09*.

Steiner, T., Schramm, L., Jin, Y., and Sendhoff, B. (2007). Emergence of feedback in artificial gene regulatory networks. In *Proc. of CEC*, pages 867–874.

Steiner, T., Trommler, J., Brenn, M., Jin, Y., and Sendhoff, B. (2009). Global shape with morphogen gradients and motile polarized cells. In *Proceedings of the 2009 Congress on Evolutionary Computation*, pages 2225–2232.

Wolpert, L. (2004). *Principles of Development*. Oxford University Press.

Minimal Synthetic Cells

Designing a Protocell: Attempt at a Systemic Design Linking Information, Metabolism and Container

S.E.Maurer[1], A. Albertsen[1], J. Cape[2], J.M. Boncella[2], L. Spencer[2], H.-J. Ziock[2], S. Rasmussen[1], and P.-A. Monnard[1]

[1] FLinT center, University of Southern Denmark, 5230 Odense M, Denmark
[2] Los Alamos National Laboratory, Los Alamos, NM 87545, USA
monnard@ifk.sdu.dk

Extended Abstract

Living cells are in many respects the ultimate nanoscale chemical system. Within a very small volume they can produce highly specific and useful products by extracting resources and free energy from the environment. They are self-assembled and self-organized, as well as capable of self-repair and self-replication.

Designing artificial chemical systems bottom up (artificial cells [1] or protocells [2-4]) endowed with these powerful capabilities are being intensively investigated. Usually such chemical systems are designed around the encapsulation of a set of genes along with a gene translation and protein generation unit, all confined within the boundaries of liposomes/vesicles [3, 4]. The generated artificial systems have many of the basic characteristics of a living system, but usually completely lack the gene mediated regulation functions that natural cells possess [5-7].

To address this issue, we are attempting to implement a simple, chemical system in which the regulation of the metabolism is truly mediated by information molecules [8, 9]. Our proposed system is composed of a chemical mixture of fatty acids that form bilayers (compartment), amphiphilic information molecules (polymerized nucleic acids -NAs), and metabolic complexes (photosensitizers). Due to the intrinsic properties of all its components, a chemical system will self-assemble into aqueous, colloid mixtures conducive to the necessary metabolic steps, as well as the non-enzymatic polymerization of the building blocks of the information unit. The metabolic reaction products (e.g., the container molecules) will in turn promote system growth and information replication.

In this scheme, the polymerized NAs acts as an information molecule mediating the metabolic catalysis (electron donor/relay system) with a ruthenium metal complex as a cofactor and sensitizer. The metabolic catalyst converts the hydrophobic precursor container molecules into amphiphiles, thus directly linking protocell metabolism with information. In a first experimental design, the NA chain has been replaced by a single nucleobase, 8-oxoguanine, which is tethered to one of the bipyridine ligands of the metal center [10].

We report the following major steps towards this chemical protocell: (1) the spontaneous formation (self-assembly) of chemical structures consisting of decanoic acid, its precursor, and the simplified NA-ruthenium complexes; (2) metabolism mediation by a nucleobase to effectively promote the photochemical assisted amphiphile synthesis, which continuously drive the system away from equilibrium; (3) the demonstration of reaction selectivity dependent on the nature of the information molecule since only one specific nucleobase has the required redox potential to allow the metabolism to function; (4) photochemical formation of amphiphiles that functions efficiently within the membrane, i.e., the protocell compartment; and (5) a demonstration of continued metabolic functionality after extrusion mediated container division.

The next steps are the integration of short nucleic acid oligomers as opposed to a single nucleobase as the information material to study their photocatalytic activity and attempts to adopt the underlying metabolic reaction to drive the polymerization of the oligomers, thereby yielding replication of the information molecules.

References

[1] Pohorille A. and Deamer D. W. (2002). Artificial cells: prospects for biotechnology. *Trends in Biotechnology*, 20, 123-128.
[2] Szostak, J. W., Bartel, D.P. and Luisi, P.L. (2001) Synthesizing life. *Nature*, 409, 387-390.
[3] Luisi P. L., Ferri F. and Stano P. (2006) Approaches to semi-synthetic minimal cells: a review. *Naturwissenschaften*, 93, 1-13.
[4] Monnard, P.-A., Ziock, H. and Declue, M.S. (2008) Organic nano-compartments as biomimetic reactors, and protocells. *Curr. Nanoscience*, 4, 71-87.
[5] Noireaux V. and Libchaber A. (2004) A vesicle bioreactor as a step toward an artificial cell assembly. *Proc Natl Acad Sci U S A*, 101, 17669-17674.
[6] Monnard P.-A., Luptak A., and Deamer D.W. (2007). Models of primitive cellular life: Polymerases and templates in liposomes. *Phil. Trans. R. Soc. B*, 362, 1741-1750.
[7] Yutetsu, K., Pasquale, S., Ueda, T., and Luisi, P. L., (2009) A synthetic biology approach to the construction of membrane proteins in semi-synthetic minimal cells. *Biochim. Biophys. Acta,* 1788, 567- 574.
[8] Rasmussen S., Chen L., Deamer D. Krakauer, D. C., Packard, N. H., Stadler, P. F. and Bedau, M. A. (2004) Evolution. Transitions from nonliving to

living matter. *Science*, 303, 963-965.
[9] Rasmussen S., Chen L., Nilsson M. and Abe S. (2003) Bridging nonliving and living matter. *Artif Life*, 9, 269-316.
[10] Declue, M.S., Monnard, P.-A., Bailey, J. A., Maurer, S.E., Collis, G.E., Ziock, H.-J., Rasmussen, S., Boncella, J. M. (2009) Nucleobase Mediated, Photocatalytic Vesicle Formation from an Ester Precursor. *J. Am. Chem. Soc.,* 131, 931-933.

Physically Grounded Simulations of a Self-Replicating Chemical Aggregate

Harold Fellermann[1,2] and Steen Rasmussen[1,3]

[1]Dep. of Physics and Chemistry, University of Southern Denmark, Campusvej 55, 5230 Odense, Denmark
[2]ICREA-Complex Systems Lab, Universitat Pompeu Fabra, Dr. Aiguader 80, 08003 Barcelona, Spain
[3]Santa Fe Institute, 1399 Hyde Park Road, Santa Fe NM 87506 USA
harold@ifk.sdu.dk

Extended Abstract

Self-replicating structures have been studied as models of living organisms since the very onset of Artificial Life research, particularly in the abstract mathematical framework of cellular automata (von Neumann (1966); Langton (1984)). Here, we study self-replicating structures in the 3D space-time continuous and physically grounded framework of dissipative particle dynamics (DPD). DPD is essentially a numerical solver of the Navier-Stokes equations with incorporated thermal fluctuations. The framework is particularly suited for coarse grained simulations of complex liquids and soft condensed matter systems on microscopic length scales. (Groot (1997))

Such a DPD based physical embedding allows us to study self-replicating structures not only as abstract mathematical entities, but to regard them as models of real-world physical objects. In particular, we model super-molecular lipid aggregates (surfactant-coated oil droplets) equipped with an internal metabolism that drives their replication due to a natural aggregate instability. In addition, the aggregate is equipped with inheritable carriers of regulatory chemical information that enables the container-metabolism-information system (commonly referred to as protocell) to undergo Darwinian evolution (Fellermann (2007,b)). Our model is directly related to the minimal protocell design of Rasmussen and coworkers that is currently being pursued both experimentally and through theory (Rasmussen (2008)).

The simulation generates spontaneous self-assembly and self-replication of the entire container-metabolism-information aggregates as well as a fitness function for the inheritable information carriers. These findings are emergent, generic, and robust properties of the systems dynamics.

We analyze the performance of the system for all steps of the replication cycle consisting of (i) nutrient feeding, (ii) information-regulated metabolic turnover, (iii) template-directed replication of the information component, and (iv) aggregate replication by growth and division (see Figure). Interestingly, the model predicts that the most difficult obstacle to be overcome in the life-cycle of this protocell model is product inhibition of the replicating information molecules - a well-known issue from experimental studies (Sievers (1994)).

In conclusion, we argue that physical embedding allows for self-replicating structures of seemingly unanticipated simplicity. Furthermore, the physical foundations of the model opens up for applications of established knowledge and methods, e.g. from statistical physics and, therefore, allows to relate model findings to laboratory results in a qualitative manner. As such, the model provides a systemic consistency check for laboratory implementation issues (which enabled us to discover an earlier ”design bug” with consequences for the experimental implementation).

References

Fellermann, H. and Solé, R. (2007). Minimal model of self-replicating nanocells: a physically embodied, information-free scenario. *Phil. Trans. R. Soc. Ser. B* 362(**1486**):1803-1811.

Fellermann, H., Rasmussen, S., Ziock, H.-J., and Solé, R. (2007). Life-cycle of a minimal protocell: a dissipative particle dynamics (DPD) study. *Artif. Life* 13(**4**):319-345.

Groot, R. D. and Warren, P. B. (1997). Dissipative particle dynamics: Bridging the gap between atomistic and mesoscale simulation. *J. Chem. Phys.* 107(**11**):4423-4435.

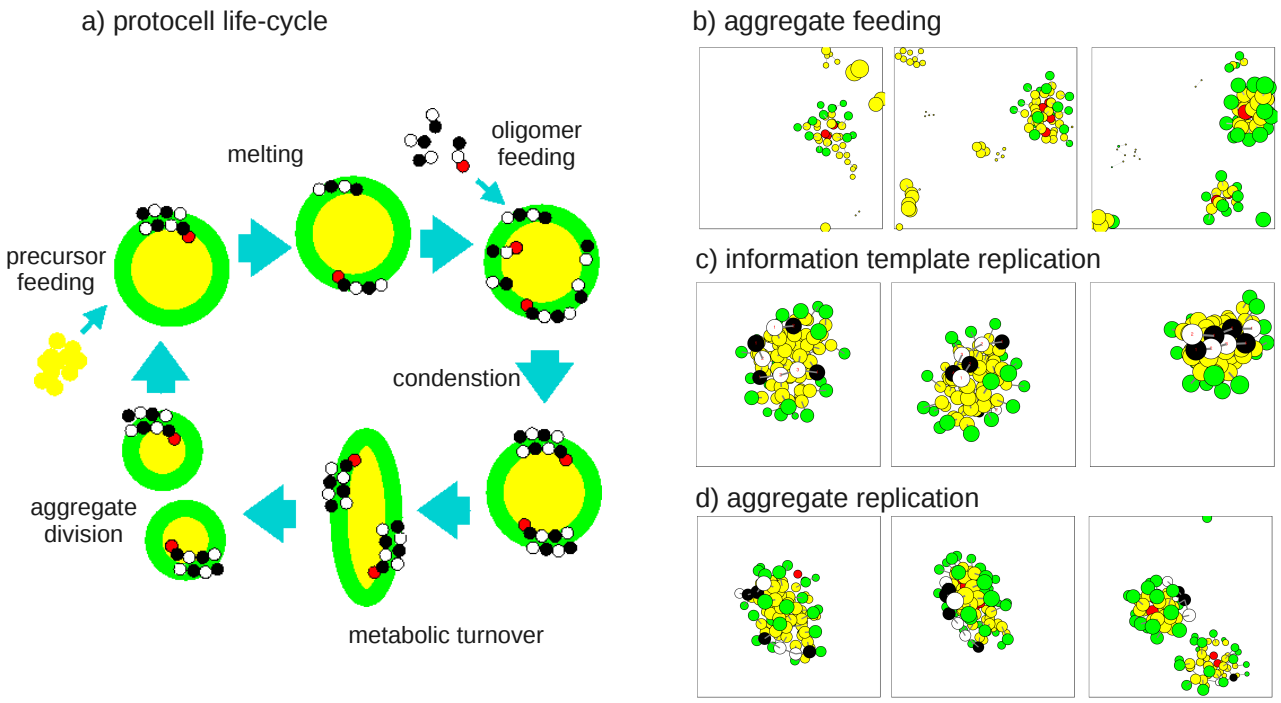


Figure 1: *(a)* The life-cycle of the protocell: Precursos molecules (yellow), surfactants (green), information polymers (black and white), and a photo-sensitizer (red) spontaneously self-assemble in water to form protocells (lower left). Feeding additional precursors increases their volume and stabilizes them when melting the information double strands. Feeding complementary oligomers allows for template-directed replication through condensation. Metabolic turnover of precursors into surfactants induces an aggregate instability that leads to division. Panels *(b)* through *(d)* show simulation snapshots of these processes.

Langton, C. G. (1984) Self-replication in cellular automata. *Physica D.* 10:135-144

von Neumann, J. (1966). Theory of Self-Reproducing Automata. University of Illinois Press, Champaign, IL.

Rasmussen et al. (2008). Assembly of a minimal protocell. In Rasmussen et al., editors, *Protocells: bridging nonliving and living matter* pages 125-156, MIT Press, Cambridge, MA.

Sievers, D. and von Kiedrowski, G. (1994). Self-replciation of complementary nucleotide-based oligomers. *Nature* 369:221-224.

Chemical Approaches to Synthetic Biology: From Vesicles Self-Reproduction to Semi-Synthetic Minimal Cells

Pasquale Stano[1], and Pier Luigi Luisi[1]

[1]Biology Department, University of RomaTre, Viale G. Marconi 446, 00146 Rome, Italy
stano@uniroma3.it; luisi@mat.ethz.ch

Abstract

The recent advent, success and diffusion of synthetic biology (SB) are mainly related to its application as markedly bioengineering-oriented discipline. In addition to this classical view, SB also means "constructive" biology, and it is aimed to the construction of synthetic (artificial, man-made) biological-like systems, at the aim of understanding basic concepts of living systems and of their parts. In the last years, we have investigated lipid vesicles (liposomes) as cell models, by studying different aspects of their general reactivity, from their self-reproduction to the hosting of simple and complex biochemical reactions. In the attempt of modeling simple autopoietic systems by vesicle populations, it was firstly shown that simple vesicles may grow and divide according to physical laws, also revealing an unexpected pattern recognized as a "matrix effect", consisting in the conservation of the average size in a population of self-reproducing vesicles. Semi-synthetic minimal cells, on the other hand, are defined as liposome-based synthetic cells that contain the minimal and sufficient number of macromolecular components in order to be defined as "alive". Clearly, the design and the construction of minimal living cells require the establishment of the minimal number of life criteria. These have been generally described as self-maintenance, self-reproduction and evolution capability. The current experimental approach to semi-synthetic minimal living cells exploits the combination between cell-free protein expression and liposome technology, and it is conceptually based on autopoietic theory. In the FP6 SYNTHCELL project, we have investigated the expression of functional proteins inside lipid vesicles by using a minimal set of enzymes, t-RNAs and ribosomes (PURESYSTEM) at the aim of constructing functional cell models. In this contribution, we will discuss recent experimental advancements in the field of synthetic cell constructions, giving emphasis to their relevance in synthetic biology, self-organization and biocomplexity, and in origins of life studies.

1. Chemical Approaches to Synthetic Biology

In the last fifty years of biological research we have been "much better at taking cells apart than putting them together'' (Liu and Fletcher, 2009). Recently, however, also thanks to great amount of detailed information gained by the analytic approach, we have the unprecedented opportunity to develop a new kind of biological understandings, namely by the synthetic (constructive) approach. Synthetic biology (SB) aims at ''designing and constructing biological parts, devices, and systems that do not exist in the natural world and also at the redesign of existing biological systems to perform specific tasks'' (http://syntheticbiology.org). SB is generally seen as a bioengineering discipline, based on design, simulation and construction of novel biological systems, but it also embodies the novel concept, perhaps not fully recognized, of gaining knowledge by constructing biological systems. This attitude is particularly relevant in those cases where the analytical (dissecting) approach cannot be undertaken, as in the case of primitive and minimal living systems.

Classic SB studies deal with the generation of new devices, systems, organisms which are supposed to perform novel "useful" tasks, like the production of fuels, of hydrogen, of a chemical species, for bioremediation, and so on. Notice that in such studies a determined goal is set at the very beginning, and all routes and tools are bent and focused for the purpose of obtaining that goal. Methodologically, SB operations *on* biological systems can be tentatively classified as additions, eliminations, substitutions, combinations, modifications (change, inversion, minimization, adaptation, etc.). They reflect the above-mentioned engineering approach, but are indeed synthetic operations, that define a constructive act and bring about novel systems.

Seen with the eyes of a chemist, SB means the construction of biological systems as in the case of molecules and molecular systems. Molecules react together according to their intrinsic chemical reactivity and environmental conditions, giving rise to complex molecules starting from simpler ones. Supramolecular chemistry describes the self-assembly and self-organization of molecules into structures, kept together by non-covalent interactions. Autocatalytic systems, oscillating reactions, reaction networks, and reactions in micro-compartments are other chemical examples of increasing complexity. The main aim of chemical SB is therefore not the achievement of a specific goal or function, but the study of the properties of a certain construct, which has been built to be tested. Clearly, as in the bioengineering approach to SB, here also the concepts and the methodologies of assembling are central, as well as the functional and structural integration among the parts.

There are several examples of possible applications of chemical synthetic biology, as recently reviewed (Luisi, 2007; Chiarabelli et al., 2009), but in this contribution we would like to focus on the attempts to make minimal living systems, in particular primitive cell models and semi-synthetic cells. Much of the discussion presented here has been published recently in a more extensive form (Luisi et al., 2006; Stano and Luisi, 2010; Stano 2010). We will first introduce the

concept of *autopoiesis*, the theoretical framework that guides the construction of minimal living cells, then we will shortly comment recent results on the self-reproduction of lipid vesicles. Then we shift the focus on more complex constructs, i.e., semi-synthetic minimal cells. Finally, we discuss our latest finding on the assembly of cells from lipids and solutes.

2. Vesicle Self-Reproduction

Studies on vesicles self-reproduction started about 20 years ago in the Luisi's group at the ETH (Zurich), together with other investigations on micelle and reverse micelle self-reproduction. These studies are linked to (and actually inspired by) the theory of autopoiesis, which accounts for the dynamical process at the basis of living entities. The self-reproduction of synthetic compartments, like those listed above, is a pre-requisite for projects aimed to construct synthetic/artificial cells in the laboratory. In fact, since synthetic compartments can grow and divide only due to physical forces, it becomes plausible to design and try to build a minimal living system that self-reproduce thanks to the interplay between chemical transformation and supramolecular reactivity, as shown in the case of micelles and vesicles. Ultimately, projects as the Minimal Cell, Synthcells, Los Alamos Bug, and similar ones are related to such reactive pattern.

2.1 Autopoiesis

The term *autopoiesis* (self-production) refers to the description of the behavior of all biological systems, and especially cells, the simplest organisms. This theory was introduced in the Seventies by the two Chilean biologists Humberto R. Maturana and Francisco J. Varela (Maturana and Varela, 1980). Within the context of SB and the construction of synthetic cells, autopoiesis is a powerful conceptual tool for defining in general terms what are the structural and functional requirements of a molecular biosystems in order to mimic the basic living features of natural ones. The simplest autopoietic dynamics is shown schematically in Figure 1 (Luisi, 2003). The autopoietic unit is a self-bounded material structure, where boundary components (**L**) are formed by internal chemical transformations mediated by the network **E**. In such way, the precursor(s) **P** enter the autopoietic unit and are then transformed into **L**. Eventually **L** decays to a waste product **W**. At the same time, the chemical network **E**, which can be composed by few or several components (not shown) is not static, but also continuously destroyed and reconstructed at the expenses of building blocks **Q** (giving the by-products **Z**). Overall, the autopoietic unit stays out of equilibrium but maintains its identity despite the continuous transformation of its components. Its existence relies on environmental conditions, due to the need of assimilation of components from outside. For this reasons the autopoietic cells establish a sort of minimal cognitive relationships with its environment.

Notice that the "shell" (the boundary formed by **L** molecules) as well as the "core" components (the **E** sub-system) are simultaneously produced by the internal autopoietic dynamic, i.e. the autopoietic system actually produces its own compounds and its own processes.

Living cells are autopoietic units, but the contrary is not necessarily true (for a discussion, see Bitbol and Luisi, 2004).

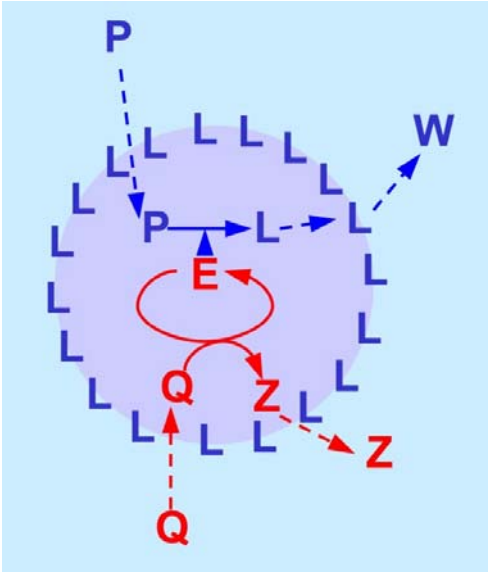


Figure 1. Schematic drawing of an autopoietic cell.

Clearly, in living cells **L** molecules are the lipids and the proteins of cell membranes, whereas **E** is the genetic/metabolic network. **P** and **Q** are the basic nutrients for cell growth, and **W**, **Z** the waste materials. Is it possible to build a (minimal) autopoietic cell in the laboratory? To answer this question, we firstly have to conceptually simplify the structure shown in Figure 1 by reducing the complexity of the elements involved in the autopoietic dynamics (reducing their number, and simplifying their structure/function).

One first answer to this question has been provided in terms of vesicle self-reproduction, which consists in an autopoietic growth (and division) based on the scheme indicated in Figure 1. In particular, it has been demonstrated that a supramolecular assembly of **L** molecules (a vesicle, but also a micelle or a reverse micelle) can grow at the expenses of a precursor **P**, without any internal metabolism (without the red sub-system shown in Figure 1).

We will see later how synthetic cells are now designed in order to display a similar autopoietic mechanism, based on a minimal DNA/RNA/enzyme genetic/metabolic network (**E** in Figure 1).

2.2 Recent advancements in vesicles self-reproduction

We have recently reviewed the whole field of vesicles self-reproduction, from the historical and scientific viewpoints (Stano and Luisi, 2010). The mechanism underlying vesicle self-reproduction is based on the following points: (1) existence of a proper precursor **P**, that can be chemically converted into the membrane-forming compound (**L**) by hydrolysis, oxydation, deprotonation, and other simple transformations; (2) uptake of **P** by existing vesicles, and transformed into **L** therein; (3) the vesicle growth must

proceed in a way that an unstable physical state is soon reached, which precedes the division into two or more daughter vesicles. It has been shown long ago that fatty acid vesicles can grow and self-reproduced at the expenses of fatty acid anhydride (Walde et al., 1994), and fatty acid micelles (Bloechliger et al., 1998). Oleic acid systems are typically used in this context. In these systems, the above-mentioned conditions (1-3) are satisfied. In particular, condition 3 is thought to derive from unbalanced surface-to-volume growth, which brings about to vesicle instability (Fiordemondo e Stano, 2007; Luisi et al., 2008). One of the most intriguing results from such studies is known as the "matrix effect" (Bloechliger et al., 1998; Lonchin et al. 1999; Berclaz et al., 2001; Rasi et al., 2003). During the investigation of vesicles self-reproduction it was discovered that the size of pre-existing vesicles was somehow conserved in the next vesicle generation. In particular, it was shown that the size distribution of vesicles (formed after addition of **P** to a pre-existing vesicles population) was very similar to the size distribution of pre-existing vesicles, as if the vesicle size acts as a "template". The mechanism of matrix effect is not yet understood, but a recent investigation brings about evidences on possible intermediates. Freeze-fracture electron-micrographs suggest the transitory existence of elongated "twin" vesicles (Stano et al., 2006) resembling bacteria during binary division. Previous results obtained with ferritin-containing vesicles (Berclaz et al., 2001) indicate that in some conditions the solute molecules are redistributed among daughter vesicles. An interesting report on self-reproduction of *giant* fatty acid vesicles has been recently provided by Szostak and coworkers (Zhu and Szostak, 2009), who demonstrated that elongated tubular vesicles, derived from micelle uptake, can divide into into several smaller vesicles. Interestingly, experiments done with a permeable buffer indicate that vesicle pure-growth or vesicle growth/division is indeed governed by the surface-to-volume growth ratio. Experiments from Sugawara's group (Kurihara et al., 2010) with synthetic surfactants show that self-reproduction can also occurs by a translocation mechanism, i.e., a new vesicle, born inside the mother one, comes out *via* a not well understood physical translocation through the parent membrane.

3. Minimal Cells

As noticed before, although the details of vesicle self-reproduction are yet unknown, such studies prompted the development of more complex models of minimal self-reproducing systems, namely the construction of vesicle–based cell–like systems, with the final aim of creating living cells in the laboratory. These constructs, which are called protocells, artificial cells, minimal cells, synthetic cells or semi-synthetic cells, are the subject of flourishing research into the origins of life and synthetic biology communities. Among the most active groups in the field, we must recall David Deamer at the University of California, Jack Szostak at Harvard, Tetsuya Yomo at the Osaka University, Steen Rasmussen at the FLinT (Southern Denmark University).

We limit ourselves to the discussion of our current approach, known as the semi-synthetic one (Luisi et al., 2006). Such approach (Figure 2) consists in using lipid vesicle as cellular model, and implement a sort of minimal metabolism based on DNA/RNA/enzyme components. The philosophy behind minimal cells lies again in the autopoietic theory. In particular, emphasis is placed on the need for a cellular system of minimal complexity.

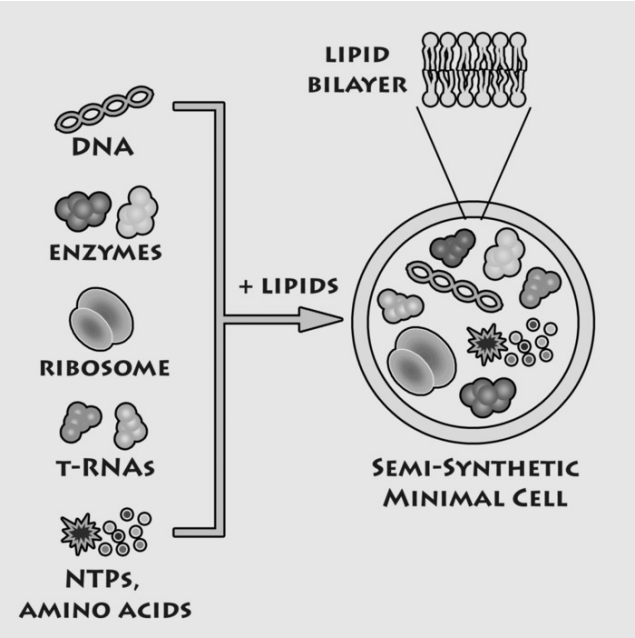


Figure 2. Semi-synthetic approach. Reproduced with permission from Elsevier from Chiarabelli et al. (2009).

Minimal cells are thus composed of the minimal number of genes, enzymes, ribosomes, tRNAs and low molecular weight compounds that are encapsulated within a synthetic compartment as in the case of lipid vesicles. The resulting construct, which is similar to a living cells and displays minimal living properties (self-maintenance, self-reproduction and possibility to evolve) is generally designed on the basis of the minimal number of functions required and on the minimal complexity of the biochemical elements needed for its construction.

Conceptually, therefore, semi-synthetic minimal cells come from one of the operations mentioned as typical of SB approaches (elimination of unnecessary elements in a system). The result of such simplification resembles very much the biological notion of *minimal genome*, i.e., the minimal number of genes requested to make a living organism. Classical studies based on comparative genomics (reviewed by Luisi et al., 2002, 2006) suggest that such number lies between 200-300 genes, and the figure of 204 genes has been proposed by Moya and coworkers on the basis of a recent study (Gil et al., 2004). A similar result (151 genes) has been obtained by Forster and Church (2006) by reasoning on the minimal biochemical requirements of a minimal cell.

In principle, therefore, it would be possible to build a synthetic cell by inserting a minimal genome inside liposomes, as well as all the macromolecules and low molecular-weight compound required for decoding the genome. This has not been done yet, and although several

advancements have been recorded in the recent years, this goal appears to be not easily reachable. We describe below some key milestones along the road-map to minimal cells, according to the semi-synthetic approach. We then conclude this contribution by giving a summary of most recent results from our group, and a survey of some general aspects and modern trends of minimal cell studies.

3.1 Pioneering studies

The first report dates back to 1999, and describes the first proved ribosomal polypeptide synthesis (poly(Phe) from poly(U)) inside liposomes (Oberholzer et al., 1999). The demonstration that ribosomal protein synthesis can occurs inside vesicles actually allows the design of more complex systems, based on DNA transcription into messenger RNA and translation of the latter into protein (therefore developing a *function*). Semi-synthetic minimal cells approaches are based on this idea. From the experimental viewpoint they consist into a convergence of *in vitro* biochemical systems and liposome technology. By using cell extracts or – more recently – reconstituted transcription/translation kits, as the PURE System introduced by Ueda and coworkers (Shimizu et al., 2001), functional proteins can be expressed inside vesicles. The basic idea is the following. Firstly, the protein expression cover about 50% of the minimal genome; second, it has a sufficient complexity to be used as a (partial) model of a whole cell metabolism; third synthesizing *functional* proteins inside liposomes, e.g. enzymes, structural proteins and so on, paves the way to implementing minimal cellular functions, like genomic replication, lipid synthesis, environment sensing, membrane functionalization, active transportation of nutrients inside, motion, etc.

Since the report from Yomo's group in 2001 (Yu et al., 2001) there have been several reports on the synthesis of a functional soluble protein (GFP, green fluorescent protein) inside lipid vesicles (reviewed in Luisi et al., 2006, Chiarabelli et al. 2009; Stano 2010). This can be considered a standard achievement. Recent investigations are instead devoted to more quantitative studies (Hasoda et al. 2008; Saito et al. 2009; Amidi et al. 2010; Sunami et al., 2010).

3.2 Recent advancements

It is useful to mention here two of the most recent results, that differ technically and conceptually from the standard achievement described in the previous paragraph. The first is our report on the synthesis of transmembrane protein inside lipid vesicles, without the help of specialized proteins, but simply exploiting the self-assembly properties of the protein and lipid membrane (Kuruma et al., 2009). The work aimed to construct a minimal cell capable of synthesizing lipid molecules from inside, as shown in Figure 1. The underlying biochemistry is the two-steps transformation of glycerol-3-phosphate into phosphatidic acid, a membrane-forming compound. In order to carry out these transformations, two active enzymes need to be synthesized inside a lipid vesicle, namely the glycerol-3-phosphate acyltransferase (G3PAT, a transmembrane enzyme) and the lysophosphatidic acid acyltransferase (LPAAT, a membrane-associated enzyme) (Figure 3).

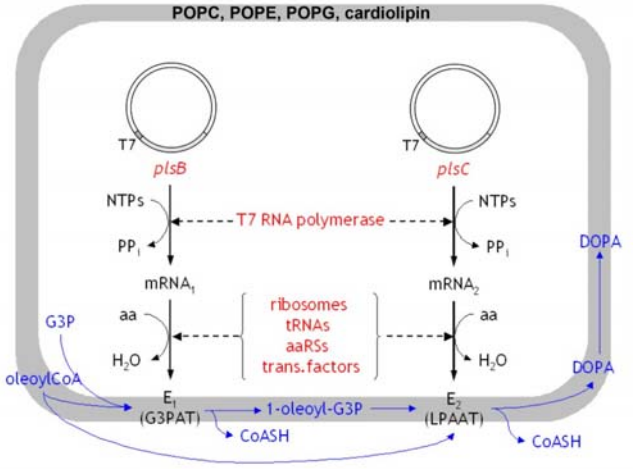


Figure 3. Lipid-synthesizing minimal cell. All translational factors are encapsulated inside liposome, which is composed by four kinds of phospholipids. The composition of lipid membrane is a key factor for obtaining simultaneously a good entrapment of molecules inside liposomes, high yield of protein synthesis, and functional forms (correct folding, insertion) of the target enzymes (G3PAT and LPAAT).

The desired two-steps reaction could be achieved only by changing the redox conditions, and unfortunately the amount of produced phosphatidic acid was too low to observe a macroscopic change on vesicles. This study represents, however, an important advancement along the roadmap to minimal self-reproducing cells.

The second most recent result deals instead with the attempt of synthesizing a functional protein (GFP) inside small vesicles (diameter 200 nm) (Souza et al., 2009). This study was intended as an experimental investigation on the *minimal size* of cells, an old debated question in biology. By using the protein synthesis as a paradigm of the whole cellular metabolism, we have indeed successfully demonstrated that 200 nm vesicles (plausible models for small ancient cells) actually support a complex metabolism as the transcription/translation one. Interestingly, a careful analysis of the statistics of co-entrapment of all macromolecular components (ca. 80) involved in the protein synthesis revealed a surprising conclusion. In fact, according to the classical description of solute entrapment, the Poisson probability of co-encapsulating the ca. 80 different molecules (0.1-1 μM each) inside 200 nm (diam.) vesicles is practically zero (10^{-26}). Nevertheless, the protein was synthesized in some compartments, and therefore the apparent contrast between observed and predicted behavior represents a conundrum. In order to explain the observations, we made the hypothesis that local (internal) solute concentration was ca. 20 times higher than the nominal (bulk) one. We have recently investigated

this phenomenon by entrapping ferritin inside liposomes, and analyzing the occupancy frequency in each liposome by means of cryo-TEM visualization (Luisi et al. submitted), see below for a short comment on such study.

3.3 On the entrapment of solutes

Projects on the construction of minimal cells foresee, as basic assembly step, the formation of solute-containing lipid vesicles. It is interesting to notice that such important process has not been studied in great detail. It is clearly recognized that the entrapment process depends on the mechanism of vesicle formation, on the nature of lipids and solutes, and by the concentrations used in the experiment. The general hypothesis is that the average number of entrapped molecules (N_0) depends on the concentration of solutes (C_0) used and on the vesicle volume (V), i.e. $N_0 = C_0V$. Deviations from the expected average number are typically modeled by a Poisson distribution. In our recent investigation on the encapsulation of ferritin inside lipid vesicles – a study that was triggered by the conundrum of simultaneous multiple entrapment of several components inside liposomes, see above – we discovered that the description of entrapment phenomena is not well described by the standard model (Luisi et al., submitted). When vesicles are allowed to form spontaneously in the presence of solutes, the surprising result is that the classical description fails (at least for submicrometric vesicles) with respect to: (*i*) the average number of solute per vesicle, (*ii*) the expected occupancy distribution.

In particular, we have observed that a small fraction of vesicles are filled by several solute molecules, confirming our working hypothesis of high internal solute concentration, and that the occupancy profile does not follow the Poisson distribution, being aligned instead as in a long-tail distribution. Experiments are currently in progress to fully characterize the vesicle system.

This result indicates that SB studies on the construction of synthetic or semi-synthetic cells actually drives also advancements in basic science. In fact, thanks to such approach it becomes evident that our simple model of vesicle formation needs a revision, since there are suggestions that membrane closure into a vesicle is not a passive event, but might bring about solute recruitment with the consequent formation of high internal solute concentration, which is a pre-requisite for the spontaneous formation of functional cells.

3.4 Next developments and conclusions

In conclusion, there has been a big progress in the ability of constructing minimal cells by the semi-synthetic approach. The state of the art is represented by the synthesis of water-soluble as well as membrane proteins. This will allow the realization of more complex systems that are capable of implementing additional function, especially in the direction of constructing a minimal autonomous cell, and a self-reproducing cell. As evident in Figure 1, the final goal will be the simultaneous and possibly functionally coupled core-and-shell reproduction.

In order to discuss next development, we have to distinguish among conceptual advancements and technical ones. Moreover, it is also useful to discuss the general aspects of semi-synthetic approach, within SB and with respect to other research lines.

New directions in minimal cell research, as anticipated, should focus on the self-reproduction of the genetic/metabolic molecules as well as a more efficient lipid synthesis, the so-called *core-and-shell reproduction*. Such goal can be reached by duplicating DNA and by implementing the *in situ* ribosome synthesis. The other two set of key macromolecules, tRNAs and aa-tRNA synthase need also to be synthesized inside vesicles. Lipid synthesis is particularly relevant, and together with phospholipid synthesis, fatty acid synthesis should be considered (for a preliminary report, see Murtas 2009). The study on the cell-free synthesis of transcription factors (Asahara and Chong, 2010), and on a short biosynthetic pathway (UDP-*N*-acetylglucosamine pathway, by Zhou et al., 2010), point toward the realization of more complex systems by the *in vitro* gene expression approach. Another interesting direction has been pioneered by Davis and coworkers, who let synthetic cells send a chemical message (ribose-borate complex, synthesized inside the synthetic cell via the formose reaction) to a bacteria population, stimulating a quorum sensing response (Gardner et al., 2009). It is expected that further development may concern a two-way communication between synthetic and natural cells (for a discussion, see also Cronin et al. 2006, for a potential application as drug delivery systems, see Zhang et al. 2008). Further studies might be devoted to the explicit investigation of stochastic effects within synthetic cells (such concept has been only marginally discussed in Tsuji and Yoshikawa, 2010; Saito et al., 2009; Yamaji et al., 2009; Carrara et al. 2009, Sun and Chiu, 2005; Dominak and Keating, 2007; Lohse et al., 2008), as well as an explicit approach that take into account the whole vesicle population instead of focusing on single vesicles (competition and selection, see Stano, 2007; Chen and Szostak, 2004; Cheng and Luisi, 2003; and cooperation). From the technical viewpoint, it is remarkable the use and the possible future developments of microfluidic devices for producing and filling giant vesicles (Ota et al. 2009).

A more general discussion, on the other hand, must focus on the relevance of semi-synthetic cells as primitive cell models. Clearly, the compounds used to build a semi-synthetic cell are not primitive, and the resulting semi-synthetic cell is "minimal" in the sense of minimal number of functions. In other words, simplicity of minimal cell does not necessarily translate into primitiveness. In other words, one has to also point to simpler cellular models, highlighting chemical and physical aspects of minimal cells, which are still not completely clear. Some efforts have been done in this direction by the group of Szostak, who recently reviewed the main results of his research and the issue of constructive approach (Mansy and Szostak, 2009; Schrum et al., 2010). In order to build more primitive cell models it is necessary to complement the notion of minimal cells with more basic models, and several strategies can be tested. For instance, one

could focus on the synthesis of very simple polypeptides, or by implementing some small metabolic network, or exploiting the catalytic properties of small peptides (such as Ser-His, see Li et al., 2000; Gorlero et al., 2008), peptide-membrane interaction, and the reduction of ribosome complexity. For example, Chris Thomas, a former PhD student of Luisi's group, and Erica D'Aguanno (graduate student), studied the interaction of rRNA with poly-L-arginine, showing that stable complexes, in definite molar ratio, form rapidly and spontaneously by simple mixing the two components. The resulting complexes show a compact structure as evident by cryo-TEM imaging and dynamic light scattering, and have similar dimension and gross form of ribosomes. This may suggest a simple origin for ribosome particles as ribonucleic acid/basic peptide complexes.

In summary, research on synthetic cells is now flourishing after a long "incubation" stage. Although limited, the number of groups interested in such research is increasing, and the issue of creating compartment-based cell model is approached from the experimental as well as modeling (Solé et al., 2007; Rasmussen et al., 2009) viewpoints. We are confident that synthetic cell studies will impact on basic biological knowledge, especially in revealing physico-chemical and dynamic aspects of cell-like functions, as well as by becoming important tools in biotechnology and drug delivery.

Acknowledgments. This work has been funded by the SYNTHCELLS project (Approaches to the Bioengineering of Synthetic Minimal Cells, EU FP6 Grant #FP6043359); by the Human Frontiers Science Program (RGP0033/2007-C) and by the Italian Space Agency (Grant Nr. I/015/07/0). It is also developed within the COST Systems Chemistry CM0703 Action.

References

Amidi, M., de Raad, M., de Graauw, H., van Ditmarsch, D., Hennink, W. E., Crommelin, D. J. A., and Mastrobattista, E. (2010). Optimization and quantification of protein synthesis inside liposomes. *Journal of Liposome Research*, 20:73-83.

Asahara, H., and Chong, S. (2010). In vitro genetic reconstruction of bacterial transcription initiation by coupled synthesis and detection of RNA polymerase holoenzyme, *Nucleic Acid Research*, doi:10.1093/nar/gkq377.

Berclaz, N., Bloechliger, E., Mueller, M., and Luisi, P. L. (2001). Matrix effect of vesicle formation as investigated by cryo-transmission electron microscopy. *Journal of Physical Chemistry B*, 105:1065-1071.

Bitbol, M., and Luisi, P. L. (2004). Autopoiesis with or without cognition: defining life at its edge. *Journal of the Royal Society Interface*, 1:99-107.

Blochliger, E., Blocher, M., Walde, P., and Luisi, P. L. (1998). Matrix effect in the size distribution of fatty acid vesicles. *Journal of Physical Chemistry*, 102:10383-10390.

Carrara, P., Stano, P., and Luisi, P. L. (2009). Giant vesicles and w/o emulsions as biochemical reactors. *Origins of Life and Evolution of Biospheres*, 39:308-308.

Chen, I. A., Roberts, R. W., and Szostak, J. W. (2004). The emergence of competition between model protocells. *Science*, 305:1474-1476.

Cheng, Z., and Luisi, P. L. (2003). Coexistence and mutual competition of vesicles with different size distributions. *Journal of Physical Chemistry B*, 107:10940-10945.

Chiarabelli, C., Stano, P., and Luisi, P. L. (2009). Chemical approaches to synthetic biology. *Current Opinion in Biotechnology*, 20:492-497.

Cronin, L., Krasnogor, N., Davis, B. G., Alexander, C., Robertson, N., Steinke, J. H., Schroeder, S. L., Khlobystov, A. N., Cooper, G., Gardner, P. M., Siepmann, P., Whitaker, B. J., and Marsh, D. (2006). The imitation game – A computational chemical approach to recognizing life. *Nature Biotechnology*, 24:1203-1206.

Dominak, L. M., and Keating, C. D. (2007). Polymer encapsulation within giant lipid vesicles. *Langmuir*, 23:7148-7154.

Fiordemondo, D., and Stano, P. (2007). Lecithin-based water-in-oil compartments as dividing bioreactors. *ChemBioChem*, 8:1965-1973.

Forster, A. C., and Church, G. M. (2006). Towards synthesis of a minimal cell. *Molecular Systems Biology*, 2:45.

Gardner, P. M., Winzer, K., and Davis, B. G. (2009). Sugar synthesis in a protocellular model leads to a cell signalling response in bacteria, *Nature Chemistry*, 1:377-383.

Gil, R., Silva, F. J., Peretó, J., and Moya, A. (2004). Determination of the core of a minimal bacteria gene set. *Microbiology Molecular Biology Reviews*, 68:518-537.

Gorlero, M., Wieczorek, R., Adamala, K., Giorgi, A., Schininà, M. E., Stano, P., and Luisi, P. L. (2008). Ser-His catalyses the formation of peptides and PNAs. *FEBS Letters*, 583:153-156.

Hasoda, K., Sunami, T., Kazuta, Y., Matsuura, T., Suzuki, H., and Yomo, T. (2008). Quantitative study of the structure of multilamellar giant liposomes as a container of protein synthesis reaction. *Langmuir*, 24:13540–13548.

Kurihara, K., Takakura, K., Suzuki, K., Toyota, T., and Sugawara, T. (2010). Cell-sorting of robust self-reproducing giant vesicles tolerant to a highly ionic medium. *Soft Matter*, 6:1888-1891.

Kuruma, Y., Stano, P., Ueda, T., and Luisi, P. L. (2009). A synthetic biology approach to the construction of membrane proteins in semi-synthetic minimal cells. *Biochimica et Biophysica Acta*, 1788:567–574.

Li, Y., Zhao, Y., Hatfield, S., Wan, R., Zhu, Q., Li, X., McMills, M., Ma, Y., Li, J., Brown, K.L., He, C., Liu, F. and Chen, X. (2000). Dipeptide Ser-His and related oligopeptides cleave DNA, proteins and a carboxyl ester. *Bioorganic Medical Chemistry*, 8:2675–2680.

Liu, A. P., and Fletcher, D. A. (2009). Biology under construction: in vitro reconstruction of cellular function. *Nature Reviews*, 10:644–650.

Lohse, B., Bolinger, P.-Y., and Stamou, D. (2008). Encapsulation efficiency measured on single small unilamellar vesicles. *Journal of the American Chemical Society*, 130:14372–14373.

Lonchin, S., Luisi, P. L., Walde, P., and Robinson, B. H. (1999). A matrix effect in mixed phospholipid/fatty acid vesicle formation. *Journal of Physical Chemistry B*, 103:10910-10916.

Luisi, P. L. (2003). Autopoiesis: a review and a reappraisal. *Naturwissenschaften*, 90:49-59.

Luisi, P. L. (2007). Chemical aspects of synthetic biology. *Chemistry and Biodiversity*, 4:603-621.

Luisi, P. L., Allegreti, M., Souza, T., Steiniger, F., Fahr, A., and Stano, P. (submitted) Spontaneous protein crowding in liposomes: A new vista for the origin of cellular metabolism.

Luisi, P. L., Ferri, F., and Stano, P. (2006). Approaches to semi-synthetic minimal cells: a review. *Naturwissenschaften*, 93:1-13.

Luisi, P. L., Oberholzer, T., and Lazcano A. (2002). The notion of a DNA minimal cell: A general discourse and some guidelines for an experimental approach. *Helvetica Chimica Acta*, 85:1759-1777.

Luisi, P. L., Souza, T., and Stano, P. (2008). Vesicle behavior: In search of explanations. *Journal of Physical Chemistry B*, 112:14655-14664.

Mansy, S. S., and Szostak J.W. (2009). Reconstructing the emergence of cellular life through the synthesis of model protocells. *Cold Spring Harbor Symposia on Quantitative Biology*, doi: 10.1101/sqb.2009.74.014

Maturana, H. R, and Varela, F. J. (1980). *Autopoiesis and cognition: the realization of the living*. Reidel, Dordrecht

Murtas, G. (2009). Internal lipid synthesis and vesicle growth as a step toward self-reproduction of the minimal cell. *Systems Synthetic Biology*, doi: 10.1007/s11693-009-9048-1.

Oberholzer, T., Nierhaus. K. H., and Luisi, P. L. (1999). Protein expression in liposomes. *Biochemical Biophysical Research Communications*, 261:238-241.

Ota, S., Yoshizawa, S., and Takeuchi, S. (2009). Microfluidic formation of monodisperse, cell-sized, and unilamellar vesicles. *Angewantde Chemie International Edition English*, 48:6533-6537.

Rasi, S., Mavelli, F., and Luisi, P. L. (2003). Cooperative micelle binding and matrix effect in oleate vesicle formation. *Journal of Physical Chemistry B*, 107:14068-14076.

Rasmussen, S., Bedau, M. A., Chen, L., Deamer, D., Krakauer, D. C., Packard, N. H., Stadler, P. F. editors (2009). *Protocells: Bridging Nonliving and Living Matter*, MIT Press, Cambridge, Massachusetts.

Saito, H., Kato. Y., Le Berre, M., Yamada, A., Inoue, T., Yoshikawa, K., and Baigl, D. (2009). Time-resolved tracking of a minimum gene expression system reconstituted in giant liposomes. *ChemBioChem*, 10:1640–1643.

Schrum, J. P., Zhu, T. F., and Szostak, J. W. (2010). The origins of cellular life. *Cold Spring Harbor Perspectives Biology*, doi: 10.1101/cshperspect.a002212.

Shimizu, Y., Inoue, A., Tomari, Y., Suzuki, T., Yokogawa, T., Nishikawa, K., and Ueda, T. (2001). Cell-free translation reconstituted with purified components. *Nature Biotechnology*, 19:751-755.

Solé, R. V., Rasmussen, S., and Bedau, M. editors (2007). *Towards the artificial cell. (Vol 362) Philosophical Transaction of the Royal Society B.*

Souza, T., Stano, P., and Luisi, P. L. (2009). The minimal size of liposome based model cells brings about a remarkably enhanced entrapment and protein synthesis. *ChemBioChem*, 10:1056–1063.

Stano, P. (2007). Question 7: New aspects of interactions among vesicles. *Origins of Life and Evolution of Biospheres*, 37:439-444.

Stano, P. (2010). Synthetic biology of minimal living cells: primitive cell models and semi-synthetic cells. *Systems and Synthetic Biology*, doi: 10.1007/s11693-010-9054-3.

Stano, P., and Luisi, P. L. (2010). Achievements and open questions in the self-reproduction of vesicles and synthetic minimal cells. *ChemComm*, 46:3639-3653.

Stano, P., Wehrli, E., and Luisi, P. L. (2006). Insights on the oleate vesicles self-reproduction. *Journal of Physics: Condensed Matter*, 18:S2231-S2238.

Sun, B., and Chiu, D. (2005). Determination of the encapsulation efficiency of individual vesicles using single-vesicle photolysis and confocal single-molecule detection. *Analytical Chemistry*, 77:2770-2776.

Tsuji, A., and Yoshikawa, K. (2010). Real-time monitoring of RNA synthesis in a phospholipid-coated microdroplet as a live-cell model. *Chembiochem*, 11:351-357.

Walde, P., Wick, R., Fresta, A., Mangone, A, and Luisi, P. L. (1994) Autopoietic self-reproduction of fatty acid vesicles. *Journal of the American Chemical Society* 116:11649-11654.

Yamaji, K., Kanai, T., Nomura, S. M., Akiyoshi, K., Negishi, M., Chen, Y., Atomi, H., Yoshikawa, K., and Imanaka, T. (2009). Protein synthesis in giant liposomes using the in vitro translation system of Thermococcus kodakaraensis. *IEEE Transactions on Nanobioscience*, 8:325-331.

Yu, W., Sato, K., Wakabayashi, M., Nakatshi, T., Ko-Mitamura, E. P., Shima, Y., Urabe, I., and Yomo, T. (2001). Synthesis of functional protein in liposome. *Journal of Bioscience and Bioengineering*, 92:590–593.

Zhang, Y., Ruder, W. C., and LeDuc, P. R. (2008). Artificial cells: building bioinspired systems using small-scale biology. *TRENDS in Biotechnology*, 26:14–20.

Zhou, J., Huang, L., Lian, J., Sheng, J., Cai, J., and Xu, Z. (2010). Reconstruction of the UDP-*N*-acetylglucosamine biosynthetic pathway in cell-free system. *Biotechnology Letters*, doi: 10.1007/s10529-010-0315-8.

Zhu, T. F., and Szostak, J. W. (2009). Coupled growth and division of model protocell membranes. *Journal of the American Chemical Society*, 131:5705–5713.

Ribocell Modeling

Fabio Mavelli

Chemistry Department University of Bari – Via Orabona 4 – 70125 Bari Italy
mavelli@chimica.uniba.it

Extended Abstract

A minimal living cell, or protocell, is a minimal supra molecular self-bounded structure that can exhibit self-maintenance, self-reproduction and evolvability (Luisi 2003). Some years ago, Szostak and colleagues proposed a minimal cell prototype called Ribocell: ribozymes based cell (Szostak et al. 2001) that, in principle, can exhibit all these three properties. This model cell consists in a self-replicating minimum genome coupled with a self-reproducing lipid vesicular container. The genome is composed by two hypothetical ribozymes: R_{Lip} able to catalyze the conversion of molecular precursors into membrane lipids and R_{Pol} able to duplicate RNA strands. Therefore, in an environment rich of both lipid precursors and activated nucleotides the Ribocell can self-reproduce if both processes: the genome self-replication and the membrane reproduction (growth and division), are somehow synchronized. In a recent work (Mavelli et al in press) we have presented and discussed a detailed and as realistic as possible kinetic mechanism for the Ribocell based on a previously published in silico model of self-replicating vesicles (Mavelli and Ruiz-Mirazo 2007):

$$\begin{aligned}
&R_{Pol} + {}_cR_{Pol} \underset{k_S}{\overset{k_{SS}}{\rightleftharpoons}} R_cR_{Pol} \\
&R_{Lip} + {}_cR_{Lip} \underset{k_S}{\overset{k_{SS}}{\rightleftharpoons}} R_cR_{Lip}
\end{aligned} \qquad (1)$$

$$\left.\begin{aligned}
R_{Pol} + S &\xrightarrow{k_{R@S}} R@S \\
R@S_cS_n + NTP_{n+1} &\xrightarrow{k_{NPT}} R@S_cS_{n+1} + W \\
R@S_cS &\xrightarrow{k_{R@SS}} R_{Pol} + S_cS
\end{aligned}\right\} \qquad (2)$$

$$P + R_{Lip} \xrightarrow{k_L} L + R_{Lip} + W \qquad (3)$$

$$\left.\begin{aligned}
NTP_{Ex} &\xrightarrow{P_{NTP}} NTP \\
P_{Ex} &\xrightarrow{P_P} P
\end{aligned}\right\} \qquad (4)$$

Scheme 1: The Ribocell metabolism: (1) reversible association of RNA polymerase (R_{Pol}) and RNA-synthase (R_{Lip}) strands with the respective complement cRPol and cRLip; (2) catalytic cycle of the RNA replication (S= R_{Pol}, ${}_cR_{Pol}$, R_{Lip} and ${}_cR_{Lip}$); (3) conversion of the precursor P into the membrane lipid L catalyzed by the ribozyme RLip; (4) transport processes across the lipid membranes.

Using a deterministic approach, we showed that synchronization between genoma duplication and membrane reproduction can spontaneously emerge within the used approximations and the adopted kinetic parameters, all derived from the literature (see Table 1), only if the k_L constant is increased of five orders of magnitude (Mavelli *et al* in press).

Kinetic Patameters	Values	Process Description	References
$k_{SS}[s^{-1}M^{-1}]$	$8.8\cdot10^{6}$	Formation of dimers R_cR_{Pol} and R_cR_{Lip}	Christensen 2007
$k_S[s^{-1}]$	$2.2\cdot10^{-6}$	Dissociation of dimers R_cR_{Pol} and R_cR_{Lip}	Christensen 2007
$k_{R@S}[s^{-1}M^{-1}]$	$5.32\cdot10^{5}$	Formation of R@S	Tsoi and Yang 2002
$k_{R@SS}[s^{-1}]$	$9.9\cdot10^{-3}$	Dissociation of Complexes $R@S_cS$	Tsoi and Yang 2002
$k_{NTP}[s^{-1}M^{-1}]$	0.113	Nucleotide Polymerization in Oleic Vesicle	De Frenza 2009
$k_L[s^{-1}M^{-1}]$	0.017	Catalyzed Lipid Precursor Conversion	Stage-Zimmermann and Uhlenbeck 1998
$k_{in}[dm^2s^{-1}]$	$7.6\cdot10^{19}$	Oleic acid association to the membrane	Mavelli et al.2008
$k_{out}[dm^2s^{-1}]$	$7.6\cdot10^{-2}$	Oleic acid release from the membrane	Mavelli et al.2008

P_P [$cm \cdot s^{-1}$]	$4.2 \cdot 10^{-9}$	Membrane Permeability to Lipid Precursor	Sacerdote and Szostak 2005
P_{NTP}[$cm \cdot s^{-1}$]	$1.9 \cdot 10^{-11}$	Membrane Permeability to Nucleotides	De Frenza 2009
$P_W=P_S$	0.0	Membrane Permeability to W and genetic staff	
P_{aq}[$cm \cdot s^{-1}$]	$1.0 \cdot 10^{-3}$	Oleic Acid Membrane Permeability to Water	Sacerdote and Szostak 2005

Table 1: Kinetic Constants and Permeability of the Ribocell *in silico* model at room temperature (S= R_{pol}, $_cR_{pol}$, R_{lip} and $_cRLip$).

In this contribution we will focus the attention on the role of random fluctuations on the Ribocell time behaviour by using a Monte Carlo program developed in recent years for simulating chemically reacting compartmentalized systems (Mavelli *et al* 2008). The random nature of reacting events (*intrinsic stochasticity*) can highly differentiated the time course of each single protocell in the population, since the effect of fluctuations is enlarged by the autocatalytic character of genome replication. Moreover, another source of time course dispersion is the random distribution of the cell internal content after each division (*extrinsic stochasticity*). Also in this case, displacement from the deterministic equality of the genetic staff amount in both the daughter cells is amplified by the nature of the internal metabolism. However, while intrinsic stochasticity can determine equivalent behaviours with different time scales (Fig.1A), the extrinsic randomness can produce completely different outcomes bringing to the death for dilution of the Ribocell if a complete segregation of ribozymes in diverse protocells takes place (Fig 1B,C).

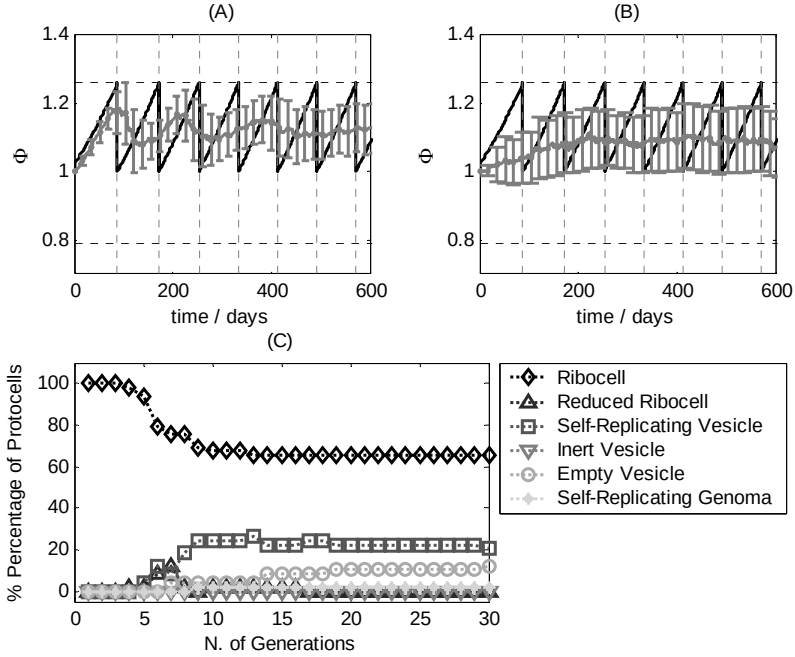


Figure 1: Comparison between deterministic curves (black lines) and stochastic simulation data (gray lines with error bars) of the Ribocell reduced surface Φ obtained setting (A) $k_L=1.7x10^4$ $s^{-1}M^{-1}$ and (B) $k_L=1.7x10^5$ $s^{-1}M^{-1}$ (Vertical dashed lines are the deterministic division times). (C) Composition of the Ribocells population against the generation number ($k_L=1.7x10^5$ $s^{-1}M^{-1}$).

References

Christensen U. (2007) Thermodynamic and Kinetic Characterization of Duplex Formation between 2'-O, 4'-C-Methylene-modified Oligoribonucleotides. DNA and RNA. *Biosci Rep* 27:327–333

De Frenza A. (2009) private communication.

Luisi P.L. (2003) Autopoiesis: A review and a reappraisal. *Naturwissensch* 90:49-59

Mavelli F., Della Gatta P., Cassidei L. and Luisi P.L (in press) Could the Ribocell be a feasible proto-cell model? To appear in *Orig. Life Evol. Biosph.*

Mavelli F., Lerario M., Ruiz-Mirazo K. (2008) 'ENVIRONMENT': a stochastic simulation platform to study protocell dynamics. In: Arabnia HR et al. (ed) *BIO-COMP'08 Proceedings, Vol II*. CSREA Press, New York

Mavelli F., Ruiz-Mirazo K. (2007) Stochastic simulations of minimal self-reproducing cellular systems. *Phil. Trans. Royal Soc. B* 362:1789-802

Sacerdote M.G., Szostak J.W. (2005) Semipermeable lipid bilayers exhibit diastereo-selectivity favoring ribose. *PNAS* 102:6004-6008.

Stage-Zimmermann T.K., Uhlenbeck O.C. (1998) Hammerhead ribozyme kinetics. *RNA* 4:875-889.

Szostak J.W., Bartel D.P., Luisi P.L., (2001) Synthesizing Life, *Nature* 409, 387-390.

Tsoi P.Y., Yang M., Surface plasmon resonance study of human polymerase β binding to DNA. *Biochem. J.* 361:317-325

To Burst or Not To Burst: Osmotic Regulation in a Protocell Model Through Precursor Mechano-Sensitive Channels

Ben Shirt-Ediss[1,2], Fabio Mavelli[3], and Kepa Ruiz-Mirazo[1,4]

[1]Department for Logic and Philosophy of Science, University of The Basque Country, Spain
[2]ICREA-Complex Systems Lab, Universitat Pompeu Fabra, Spain
[3]Chemistry Department, University of Bari, Italy
[4]Biophysics Research Unit (CSIC - UPV/EHU), Spain
kepa.ruiz-mirazo@ehu.es

Extended Abstract

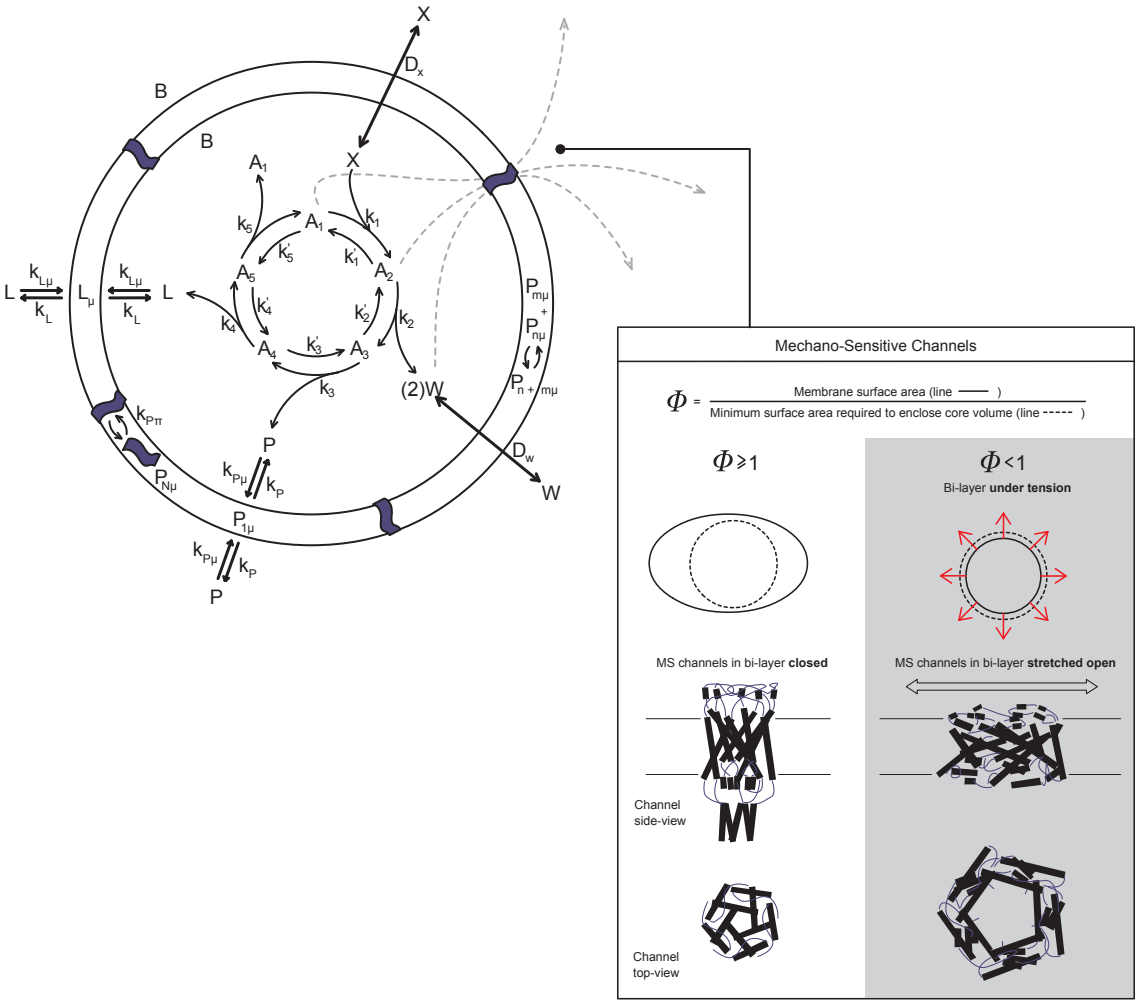


Figure 1: Protocell model with rudimentary Mechano-Sensitive (MS) membrane channels. In osmotic crisis, internal turgor causes tension in the membrane, opening the MS channels and allowing internal solutes to disperse, re-stabilising the system.

We are interested in exploring plausible mechanisms which could enable a simple lipid bi-layer protocell system for more robust and possibly richer self-maintenance dynamics in variable environmental conditions.

One fundamental problem faced by all compartments with a selectively semi-permeable membrane is the ever present threat of *osmotic burst*. For various and sometimes unexpected reasons, internal or external conditions for a cellular system can suddenly change (e.g. an *E. coli* bacterium caught in a rain shower), resulting in the appearance of a large osmotic potential across the membrane. This potential drives a 'shock' flow of water into the cellular compartment, quickly expanding the internal volume and possibly rupturing the membrane. Mechano-Sensitive (MS) channels are one prudent mechanism of increasing interest (Kung (2005)) by which a cell can detect and respond to forces in it's lipid bi-layer. These intricate structures (composed of folded protein helixes) span the membrane, and open a water-filled pore like an iris (see box on Fig. 1) in response to increasing local membrane tension. In the case of the unlucky *E. coli* bacterium caught in the rain shower, the MS channels act as 'emergency valves', releasing internal solutes until osmotic equilibrium is restored again. More generally, MS channels can be thought of as a tranducer mechanism, converting mechanical fluctuations in the membrane (local tensions) into a chemical signal (by way of modulating compartment solute permeability).

This work aims to explore more fully some ideas seeded at ECAL 2007 (Ruiz-Mirazo and Mavelli (2007)) as to how a protein channel feedback system could be useful for cellular stability at a very early stage in the origin of life i.e. in a protocell scenario. In the previous work, one case considered was protein channels becoming aligned and active in the protocell membrane only when the system was in osmotic crisis conditions ($\Phi < 1$, Fig. 1). When open, these channels accelerated the diffusion of an internal waste product out of the protocell compartment, at a rate dependent on a diffusion constant, the number of proteins channels in the membrane and the concentration gradient of the waste.

This study seeks to model the protein channels above as slightly more realistic MS channels. Instead of channels opening indiscriminately whenever there is *some* membrane tension (as in the previous case), now channels open in proportion to the *relative* membrane tension ($1 - \Phi$, when $\Phi < 1$), and each channel has a more realistic binary switching behaviour, remaining effectively closed until a tension transition barrier is crossed, after which it snaps to a fully open conformation. A second objective of this work is to investigate the dynamic implications of the MS channels facilitating not only the diffusion of waste out of the compartment, but also the diffusion of the molecules involved in the internal Ganti (Ganti (2002)) reaction cycle. This direct negative feedback on the growth of the internal cycle presents an interesting dynamical scenario not tested before with the protocell model. Simulations are again being carried out with the ENVIRONMENT (Mavelli et al. (2008)) platform. Results are to be presented at the conference.

References

Kung, C. (2005). A possible unifying principle for mechanosensation. *Nature*, 436, 647-654.

Ruiz-Mirazo, K. and Mavelli, F. (2007). Simulation Model for Functionalized Vesicles: Lipid-Peptide Integration in Minimal Protocells. In: *Advances in Artificial Life (Proceedings of the 9th European Conference on Artificial Life, 2007)*, Almeida e Costa et al. (eds.), pp. 32-41. Springer, Berlin.

Mavelli, F., Lerario, M. and Ruiz-Mirazo, K. (2008). 'ENVIRONMENT': A Stochastic Simulation Platform to Study Protocell Dynamics. In: *Proceedings of the 2008 International Conference on Bioinformatics and Computational Biology*, H. R. Arabnia et al. (eds.), pp. 934-941. CSREA Press.

Ganti, T. (2002). On the early evolutionary origin of biological periodicity. *Cell Biol. Int.*, 26, 729-735.

Constructing an Artificial Self-replication System of Genetic Information from RNA and Proteins

Norikazu Ichihashi[1,2], Yohsuke Bansho[3], Tomoaki Matsuura[1,2], Tetsuya Yomo[1,2,3]
[1]Graduate School of Information Science and Technology, Osaka University
[2]ERATO JST
[3]Graduate School of Frontier Bioscience, Osaka University
yomo@ist.osaka-u.ac.jp

Abstract

Self-replication of genetic information is one of the central functions of living systems. This function enables the living system to reproduce itself, introduce mutations, and evolve. How could a self-replication system be constructed from non-living materials on the earth? What conditions are required? The answers to these questions are largely unknown. Here, we attempted to construct an artificial self-replication system of genetic information from biological materials, such as RNA and proteins, to identify the conditions necessary to establish self-replication and enable the system to evolve. Based on previous reports, we constructed a self-replication system of genetic information from RNA (genetic information) encoding RNA replicase (Qβ replicase) and a cell-free translation system (PURE system). During the reaction, RNA replicase was translated from the RNA, and then bound to the original RNA and catalyzed its replication. These successive reactions are referred to here as self-replication of genetic information. This system consisted of more than 100 components, all of which were identified. Therefore, we can control all the components independently and quantitative analysis is possible. The reaction efficiency was markedly lower than expected from the activity of the replicase and the translation system. This poor efficiency suggests that there are as yet unknown conditions required for efficient self-replication. To clarify the problems, we analyzed the self-replication system by mathematical modeling, which indicated three limiting factors: 1) competition between translation and replication for RNA; 2) parasitic RNA amplification; and 3) inactive double-stranded RNA formation. Overcoming these problems will be necessary for realization of an *in vitro* self-replication system. To resolve the first problems, we measured the affinity of RNA with replicase and ribosome, and adjusted the ribosomal concentration to the optimum level. To resolve the second problem, we compartmentalized the reaction into a micrometer-sized water-in-oil emulsion. This was considered to confine the parasitic RNA to minor compartments, so that the other major compartments were free from parasite where self-replication continued. Although the third problem is now under investigation, the self-replication efficiency has improved significantly. These result demonstrated that establishment of an efficient self-replication system requires coordination of internal reactions and a mechanism for repression of parasitic replicator.

Simulation of Self-reproduction Phenomenon of Cells in Two-dimensional Cellular Automata

Takeshi Ishida[1]

[1] Nippon Institute of Technology
ishida06@ecoinfo.jp

Extended Abstract

Understanding the generalized mechanism of self-reproduction is considered to be fundamental for application in various fields such as mass-production of molecular machines of nanotechnology and artificial synthetic of biology (synthetic biology). Furthermore, it is considered that large, complex machine systems of over a certain size are difficult to construct by the top-down approach. Therefore, these complex systems are required to be constructed by the bottom-up approach, by applying the phenomenon of biological self-organization. Thus we have to elucidate not only the details of the cellular reaction network but also the condition for simulating self-organized, self-replicating cells.

Fifty years ago, von Neumann initiated the study of the phenomenon of self-reproduction from a mathematical point of view. This study theoretically proved the possibility of constructing a self-reproducing machine by cell state and transition rules of two-dimensional square cells. On the other hand, Neumann' self-reproducing machine was large in size; therefore, it is difficult to implement this machine perfectly in a computer system (Mange et. al. (2004)). Thereafter, Langton (1989) developed a simple machine capable of self-reproduction abandoning the completeness of Neumann's self-reproducing machine. Although the shape was very simple, the rules of transition are complicated and it could reproduce specific shapes.

In our study, we developed a model for simulating cellular self-reproduction in a two-dimensional Neumann-type cellular automaton. We demonstrated that the following 3 functions can be realized by the transition of 2 adjacent cells in a cellular automaton.

(1) Formation of a border similar to a cell membrane.
(2) Self-replication is achieved while maintaining a carrier containing information (information carrier).
(3) The division of the cell membrane is achieved while maintaining the total structure of the cell.

This study demonstrated the self-reproducing ability of a shape that was similar to that of real cell. This is not a study to clarify all the necessary and sufficient conditions of self-reproduction. It is considered that it is possible to simulate self-replication in a real dynamic chemical reaction environment by applying the transition rules determined in this study.

A two-dimensional triangular grid model was used in this study. The cell automaton was constructed by transition rules such that the state of the next step was decided by the state of the cell and that of 6 neighboring cells. Each cell has a state (0–19) and direction (6 directions) as an attribute. In the triangular grid, calculation starts from a certain initial condition. The transition rules were divided into the following 4 phases: state transition concerning cell membrane formation, division of the information carriers, movement of the information carriers, and formation of the nuclear membrane surrounding the information carriers. In other words, first we applied transition rules of cell membrane formation and settled the total states in all cells. Then, we applied the transition rules for the division of information carriers, following which we applied the transition rule of movement of the information carriers and formation of the nuclear membrane.

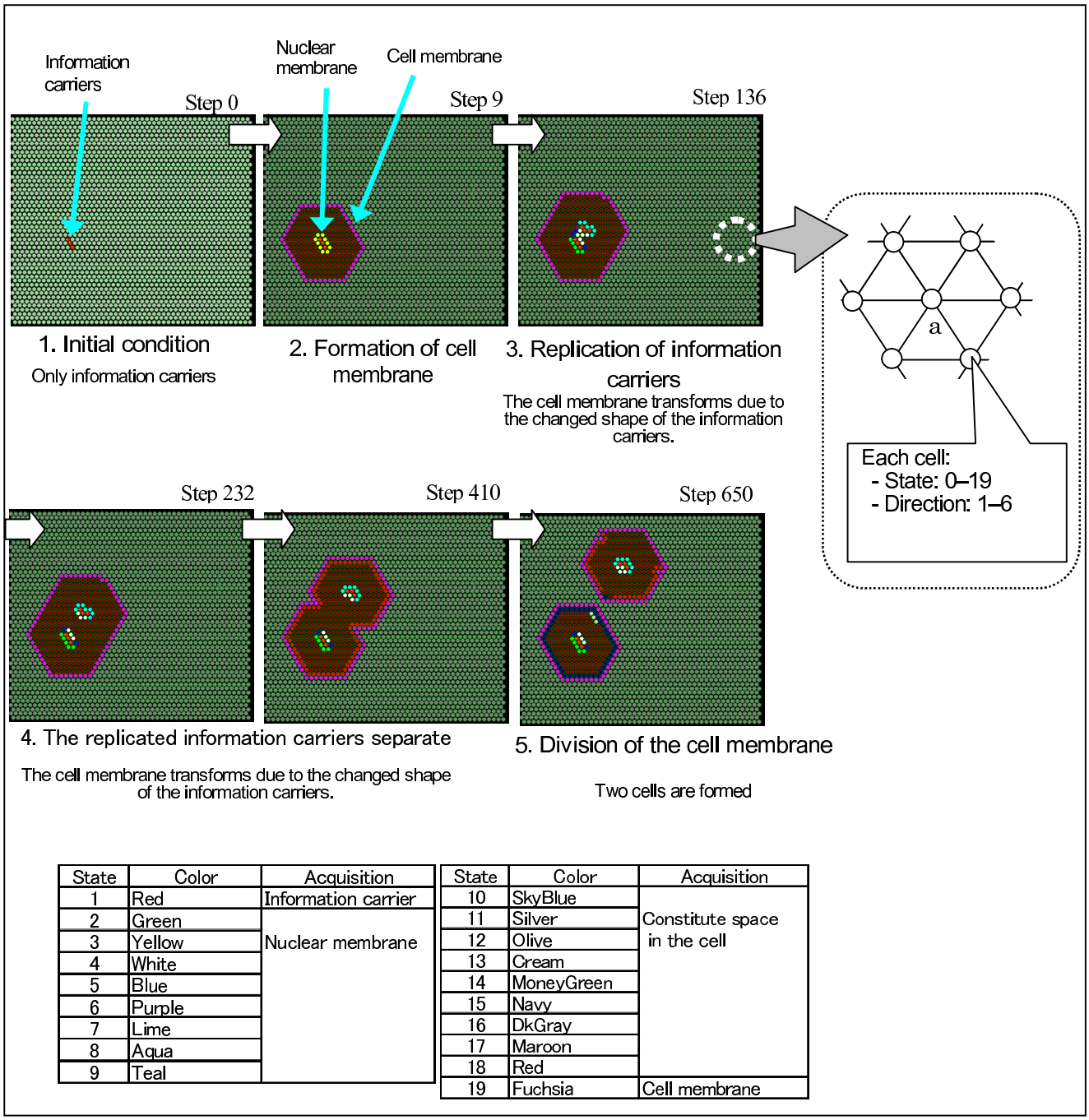
Using the model mentioned above, we demonstrate a calculation result with transition rules and the initial condition. Our model was capable of producing a self-reproducing phenomenon in a cell-like shape with few state transition rules (Figure 1).

References

Mange, D., Stauffer, A., Peparolo, L. and Tempesti; G. (2004) A macroscopic view of self-replication, Proceeding of the IEEE 92, 1929–1945, 2004

Langton., C. ed. (1989) Artificial Life, 1–48, Addison-Wesley, 1989

Takeshi Ishida (in press). Simulating self-reproduction of cells in a two-dimensional cellular automaton. To appear in the *Journal of Robotics and Mechatronics*. Vol.22, No.5



State	Color	Acquisition
1	Red	Information carrier
2	Green	Nuclear membrane
3	Yellow	
4	White	
5	Blue	
6	Purple	
7	Lime	
8	Aqua	
9	Teal	

State	Color	Acquisition
10	SkyBlue	Constitute space in the cell
11	Silver	
12	Olive	
13	Cream	
14	MoneyGreen	
15	Navy	
16	DkGray	
17	Maroon	
18	Red	
19	Fuchsia	Cell membrane

Figure 1 Results of a cell-type self-reproducing two-dimensional cellular automaton. Pink grids are cell membranes, and central red grids are information carriers. This figure shows the process of formation of cell membrane, and the process of division of the information carriers with the cell membrane.

Approach to Synthetic Cell with a Cell-free Toolbox

Jonghyeon Shin[1] and Vincent Noireaux[1]

[1]University of Minnesota, 116 Church Street SE, Minneapolis, MN 55455, USA
noireaux@umn.edu

Abstract

Cell-free protein synthesis is increasingly used to produce large amounts of proteins *in vitro*. Cell-free systems combine a strong bacteriophage transcription, in most cases the T7 RNA polymerase, to a cytoplasmic extract from an organism, such as *E. coli*, that provides the translation machinery. These systems have been prepared for many types of applications, mostly in biotechnology, such as proteomics and directed evolution. Recently, cell-free protein synthesis was used to reconstitute informational processes outside living organisms (Noireaux, et al 2003, Noireaux and Libchaber, 2004, Isalan, et al 2005). These studies were limited, however, by the current properties of cell-free systems, which have not been optimized for synthetic biology purposes. In particular, transcription is restricted to bacteriophage RNA polymerases and no procedures to accelerate messenger RNA and protein degradations have been described.

Our laboratory has developed a new cell-free expression system to specifically reconstitute biological information processes *in vitro*. This efficient transcription/translation *E. coli* cell-free system works with nine different transcription mechanisms: seven *E. coli* sigma factors and two bacteriophage RNA polymerases with their respective promoters. This set of cell-free transcriptions offers a unique modularity to engineer synthetic gene circuits. Although high protein production is required to reconstitute interesting gene networks, degradation is also an essential characteristic of gene expression. Our system includes a control of the mRNA lifetime and of the protein degradation rates. The dynamics of synthetic circuits is tuned by adjusting gene concentrations, promoter strengths, synthesized messengers and proteins lifetime.

This cell-free toolbox is used for two purposes: (i) the construction and the study of elementary gene circuits and (ii) the synthesis of an artificial cell. Multiple stage transcription cascades, AND gates and negative feedback loops have been engineered. The output signals of these circuits can be tuned in a wide dynamics range depending on the mRNA and protein degradation rates. We are currently investigating how this cell-free expression system can be used to approach biopolymer physics problems such as the DNA binding protein search problem. The cell-free extract can be encapsulated into synthetic phospholipids vesicles, which form a sort of artificial cell system. One of the main questions addressed by this research is: how can we develop the properties of these synthetic vesicles from the internal gene expression? The perspectives and the limitations of this approach will be discussed.

References

Noireaux, V., Bar-Ziv, R. and Libchaber, A. (2003). Principles of cell-free genetic circuit assembly. *Proc. Natl. Acad. Sci. U S A,* 100:12672-12677.

Noireaux, V. and Libchaber, A. (2004). A vesicle bioreactor as a step toward an artificial cell assembly. *Proc Natl Acad Sci U S A,* 101:17669-17674.

Isalan, M., Lemerle, C., Serrano, L. (2005). Engineering gene networks to emulate Drosophila embryonic pattern formation. *PLoS Biol,* 3:e64.

‘In Vitro’ and ‘In Silico’ Complementary Studies on the Physical Properties of Prebiotic Compartments

Gabriel Piedrafita[1], Ben Shirt-Ediss[2], Fabio Mavelli[3], Pierre-Alain Monnard[4] and Kepa Ruiz-Mirazo[2,5]

[1] Dept. Biochemistry and Molecular Biology, University Complutense of Madrid, Spain.
[2] Logic & Phil. Science Department, University of the Basque Country, Spain.
[3] Chemistry Department, University of Bari, Italy.
[4] FLinT Center, Institute of Physics and Chemistry, University of Southern Denmark, Denmark.
[5] Biophysics Research Unit (CSIC – UPV/EHU), Spain.
kepa.ruiz-mirazo@ehu.es

Extended Abstract

Fatty-acid vesicles are being extensively studied as experimental models of prebiotic compartments. These supramolecular structures have shown a variety of interesting dynamic properties (spontaneous self-assembly, autocatalytic growth, potential reproductive and/or competitive regimes – for a review see [1]). Nevertheless, their high dynamism presents at the same time some drawbacks: compared to compartments made of standard phospholipids (or, so-called, liposomes), fatty-acid vesicles are more permeable and less stable; they require higher monomer concentration thresholds (cvc values) and are rather sensitive to external factors, such as pH, temperature, or ionic strength [2, 3].

However, several recent experiments (e.g., [4, 5, 6]) carried out with mixtures of simple amphiphiles (i.e., both mixtures of fatty-acids and mixtures of fatty-acids with other simple surfactants or lipid derivatives), have demonstrated that certain combinations provide higher stability to this type of compartments and indicate the relevance of diverse factors, such as the packing density or irregularities between polar heads on the membrane surface, in their physical properties (e.g., in their permeability). This research is opening a whole new panorama, in which different mixtures of plausible prebiotic amphiphiles need to be explored.

In this context, we have been studying various theoretical models of plausible prebiotic compartments with ENVIRONMENT, a computational platform that was developed some years ago to simulate protocell dynamics [7]. In particular, we have started to analyze the hypothetical transition from ‘self-assembling’ fatty acid vesicles to ‘self-producing’ lipid protocells [8], focusing on the corresponding changes in the cvc and the permeability of the compartment, as well as its implications for the general stability of the protocell. In the preceding simulations, as a first approximation, membrane permeability was assumed to change linearly with its mixed composition. But, although the values of the permeability coefficients for the pure cases were derived from real data, we are aware that such an assumption for intermediate cases (i.e., for different ratios of the binary mixture) may not truly hold.

Therefore, we are currently exploring a more realistic scenario in which changes in the cvc and permeability of the compartment are a non-linear function of the membrane composition. Our approach involves the combination of ‘in vitro’ methods (wet experiments) and ‘in silico’ techniques (stochastic simulations), since we are convinced that any theoretical protocell model should be empirically grounded and, in turn, the interpretation of experimental data can be greatly clarified by means of theoretical modelling and simulation tools. Our aim is to present the results of this combined effort in the conference.

References

1. Morigaki, K., and Walde, P. (2007). Fatty acid vesicles, *Cur Opin Coll & Interface Science*, 12:75–80.
2. Monnard, P. A., Apel, C. L., Kanavarioti, A., et al. (2002). Influence of ionic inorganic solutes on self-assembly and polymerization processes related to early forms of life: implications for a prebiotic aqueous medium, *Astrobiology*, 2:139-152.
3. Thomas, J. A., and Rana, F. R. (2007). The influence of environmental conditions, lipid composition, and phase behavior on the origin of cell membranes, *Orig Life Evol Biosph*, 37(3):267-85.
4. Namani, T., and Deamer, D. W. (2008). Stability of model membranes in extreme environments, *Orig Life Evol Biosph*, 38:329-341.
5. Mansy, S., Schrum, J. P., Krishnamurthy, M., et al. (2008). Template-directed synthesis of a genetic polymer in a model protocell, *Nature*, 454:122-126.
6. Maurer, S. E., Deamer, D. W., Boncella, J. M., et al. (2009). Chemical evolution of amphiphiles: glycerol monoacyl derivatives stabilize plausible prebiotic membranes, *Astrobiology*, 9:979-987.
7. Mavelli, F., and Ruiz-Mirazo, K. (2007). Stochastic simulations of minimal self-reproducing cellular systems. *Phil Trans R Soc B*, 362(1486):1789-802.
8. Ruiz-Mirazo, K., Piedrafita, G., Ciriaco, F., et al. (in press). Stochastic simulations of mixed-lipid compartments: from self-assembling vesicles to self-producing protocells. In Arabnia, H. R., editor (2010), *Software tools and algorithms for biological systems*. Book series: Advances in experimental medicine & biology. Springer.

Biological & Chemical Information Processing and Production

'Doctor in a Cell': Vision and Accomplishments

Binyamin Gil [1], Maya Kahan-Hanum[1], Natalia Skirtenko[1], Rivka Adar[1] and Ehud Shapiro[1,2]

[1]Department of Biological Chemistry, Weizmann Institute of Science, Rehovot 76100 Israel
[2]Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot 76100 Israel
binyamin.gil@weizmann.ac.il, ehud.shapiro@weizmann.ac.il

Abstract

The "holy grail" of medical treatment is early detection and in situ cure, or destruction of malfunctioning cells. Such task could be achieved by intelligent nanometer devices capable of operating *in vivo*, sensing disease markers, correctly identifying the abnormal cells, and curing them or causing their destruction.

Our laboratory's long-term objective is to develop a 'Doctor in a cell': molecular-sized device that can roam the body, equipped with medical knowledge and treatment potential. It would diagnose a disease by analyzing the data available in its biochemical environment, and treat it by synthesizing, or activating, the appropriate drug molecules *in situ*. This kind of device might, in the future, be delivered to all cells in a specific tissue, organ or the whole organism, and cure or kill only those cells diagnosed with a disease.

As an important milestone towards realizing this desirable long-term goal, we have developed a molecular system shown to perform the abovementioned tasks *in vitro* (Benenson *et al*.). Although this system was initially limited to mRNA based disease indicators as input, we are now developing new input mechanisms that expand the spectrum of possible inputs. One input mechanism enables the detection of microRNA and almost any protein or small molecule. Another input mechanism enables the sensing of active DNA binding proteins, such as transcription factors. These new abilities may facilitate the detection of important intracellular and intercellular disease markers.

While operating this system inside living cells remains a major challenge, expanding the capabilities of molecular computers and investigating their theoretical and practical attributes might be rewarding in the long term.

Reference

Yaakov Benenson, Binyamin Gil, Uri Ben-Dor, Rivka Adar and Ehud Shapiro (2004). An autonomous molecular computer for logical control of gene expression. *Nature,* 429:423-429.

En Route to Signal Inversion in Chemical Computing

Michael Heymann[1], Kyle I. Harrington[2] , Jordan Pollack[2] and Seth Fraden[1]

[1]Physics Department, Brandeis University, Waltham, MA, USA
[2]DEMO Lab, Volen Center for Complex Systems, Brandeis University, Waltham, MA, USA
heymann@brandeis.edu

Extended Abstract

We investigate the Belousov-Zhabotinzky (BZ) reaction as a substrate for computation. Expanding on previous research we present a new technique that utilizes two modes of the BZ reaction, excitation and oscillation, and selective diffusive coupling. We show in simulation that this technique can be used to invert input signals, providing the logical operator, NOT. Our system can readily compute NOR, which when connected in multiples is sufficient for simulating any other logical operator. Furthermore, progress to experimentally implement these operators and to wire them into circuits using soft lithography and replica molding is presented.

To synthesize living systems the field of artificial life has explored numerous substrates, physical and virtual. Chemical substrates have been gaining in popularity with recent advances in chemical computation (Adamatzky, 2009; Gorecki, 2009) and cognition (Dale and Husbands, 2010). In Braitenberg's series of vehicles of increasing cognitive complexity a key turning point is the introduction of inhibitory threshold devices, allowing for the use of *numbers*, *logic,* and *basic memory* (Braitenberg, 1986). Though to an extent the latter two properties have been introduced in our choice substrate, the Belousov-Zhabotinzky (BZ) reaction, true inhibition in the BZ has not been achieved. Here we applied the novel concept of inhibitory coupling (Toiya et al. 2008) to design signal inverting logic gates.

Using BZ substrate, various logic gates have been implemented experimentally or by computer simulation. Gorecki has simulated the gates AND and OR, as well as the MAJORITY function. Adamatzky showed XOR and AND in a related experimental substrate. Collision dynamics of BZ waves have also been exploited to annihilate signals (de Lacy Costello, 2009). To our knowledge, binary negation-based gates such as the computationally universal gates NAND and NOR (Sheffer, 1913) have not been implemented. We simulated the computation of NOT and NOR in a heterogeneous BZ substrate and synthesized a NOT gate prototype.

We designed negation-based gates using a light-sensitive implementation of the BZ reaction (Vanag and Epstein, 2009). Our system is composed of two elements: excitatory and oscillatory domains connected through a filter. Both domains are chemically identical, but differ in the amount of projected light. The illumination was tuned such that induction of a small perturbation (input) into the excitatory domain can ignite a full excitation. The oscillatory domain follows an unsuppressed periodic trajectory.

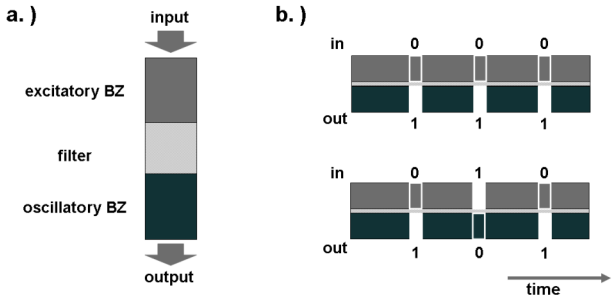


Figure 1: Inverter circuit and idealized space-time plots for signal inversion. The excitatory domain is conducting input waves into the oscillatory patch (a). Without input, the oscillatory domain transitions between oxidized (white, logic state true) and reduced (dark, logic state false) state (b, top). Due to the inhibitory coupling incoming waves will suppress and delay oscillations in the oscillatory domain into a later reading frame (b, bottom).

Using oil as a chemical filter allows for signal inversion. The filter is selective and only non-polar species such as bromine (Br_2) can permeate across (Toiya et al. 2008). Thus, a wave traveling from the excitable towards the oscillatory domain will temporarily increase the Br_2 in the oscillatory domain. Br_2 is then readily converted back to the inhibitor Br-, which will delay the oscillation in the oscillatory domain (Figure 1).

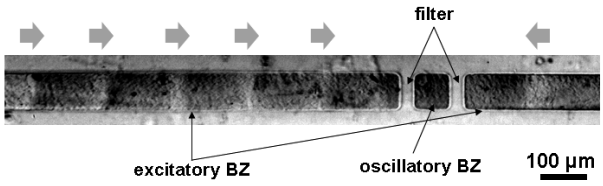


Figure 2: NOR gate prototype. Catalyst immobilized on silica gel was cast into patterned PDMS slabs. Hydrophobic PDMS walls separate BZ domains and act as chemical filters. Action potential like input waves (indicated by grey arrows) propagate towards and couple into the central oscillatory domain.

We verify our concept by simulating a simplified reaction-diffusion system of the light-sensitive BZ reaction (Vanag and Epstein, 2009). We integrate chemical turnover numerically in each BZ domain and compute the flux between compartments. Assuming fast diffusion within compartments, we reduce their size to a single point. Though a single inverter is sufficient for an inhibitory connection, we extend upon simple signal inversion to realize a NOR gate by combining two inverters. Prototypes were constructed by casting BZ catalyst immobilized on silica gel into patterned PDMS slabs (Figure 2). Hydrophobic PDMS walls were designed to separate BZ domains and act as selective chemical filters. Preliminary experimentation suggests our substrate can couple BZ domains within circuits.

The BZ reaction offers a wide range of interesting dynamics. We have described a technique capable of inverting input signals, and presented supporting simulations along with preliminary experimental results. This work suggests that the BZ reaction may be a useful substrate for the synthesis of minimally cognitive agents. Future work will utilize finite element analysis to quantitatively identify parameters for optimal input timing and delay strength. Experimental efforts will focus on increasing the robustness of single logic operators as well as connecting them into functional circuits to achieve universal computation at the microscopic scale in a chemical substrate.

Acknowledgements

We thank Andrew Adamatzky, Milos Dolic, Irving Epstein, Marcin Leda, Ning Li, and Hector Ochoa Gonzales for discussions.

References

Braitenberg, V. (1986). *Vehicles Experiments in Synthetic Psychology* MIT press, MA.

Dale, K., and Husbands, P. (2010). The evolution of reaction-diffusion controllers for minimally cognitive agents. *Artificial Life*, 16(1):1-19

de Lacy Costello, B., Toth, R., Stone, C., Adamatzky, A., and Bull, L. (2009). Implementation of glider guns in the light-sensitive Belousov-Zhabotinsky medium. *Physical Review E* 79(2):26114

Gorecki, J., Gorecka, J.N., Igarashi, Y. (2009). Information processing with structured excitable medium. *Natural Computing* 8(3):473-492

Sheffer, H.M. (1913). A set of five independent postulates for Boolean algebras, with application to logical constants. *Transactions of the American Mathematical Society* 14(4):481-488

Toiya, M., Vanag, V.K., and Epstein, I. (2008). Diffusively Coupled Chemical Oscillators in a Micro-fluidic Assembly. Angew. *Chem*. 120:7867-7869

Vanag, V. K., and Epstein, I. R. (2009). A model for jumping and bubble waves in the Belousov-Zhabotinsky-aerosol OT system. *The Journal of Chemical Physics* 131:104512

The Emergence of Replication in a Digital Evolution System using a Secondary Structure Approach

Uwe Tangen

BioMIP, Org. Chem. I, Ruhr-University-Bochum, 44780 Bochum, Germany
uwe.tangen@ruhr-uni-bochum.de

Abstract

Spontaneous emergence of non self-replication in a micro-controller based artificial chemistry model, with replication being a concerted action of several sequential micro-processes or instructions, is a difficult problem. The choice of programming language that is used to realize replication as a sequence of instructions is to a certain extent arbitrarily. The question is, how many bits have to be found by a dynamical system in the right space- and time-context to instantiate this replication. A secondary structure is introduced to allow complex instruction sets to be used. The secondary-structure folding mechanism, a directed graph or Moore automaton, allows replication to emerge with an arbitrary instruction-width.

The question of whether there is anything before emergence of replication has a tentative answer: early precursors of replication probably do not exist. Replication only starts when at least two replicating programs are in the same neighborhood replicating each other. A "cloud" of potential precursors of replication is not visible.

Introduction

The desire to create hitherto unknown information from scratch is at least as old as information processing machines, cf. e.g. Menabrea (1842). The proof that a machine can hold its own description and be able to replicate itself, together with its own description, has been provided by von Neumann (1966). The spontaneous emergence of higher-order structures was already studied with first-generation computers by Barricelli (1962). The α-universe designed by Holland (1976) was the first attempt to show spontaneous emergence of self-replicating structures using a formal language concept. But the first to convincingly show the evolution of higher order structures and processes was Ray (1991). The demonstration of spontaneous emergence of self-replication was made by Pargellis (1996). He streamlined the Tierra instruction set Ray (1991) in such a way that there were about one in 100 000 random sequences of five instructions which resulted in a self-replicator. Artificial chemistry as a field of research emerged when desktop computers had become ubiquitous McCaskill (1988); Fontana (1991) (see Dittrich et al. (2001) for a review). These works attempted to connect chemical systems with information processing at the molecular level. A promising idea was to use graph rewriting as a chemical representation and processing, McCaskill and Niemann (2001); Benkö et al. (2005). Unfortunately no evolutionary studies could be realized because of the excessive computational processing required. Also, the inherent brittleness of digital evolution made evolutionary studies with Turing machines infeasible Yoshii et al. (1998). It is nearly impossible for self-replicating programs in Turing- or register- machines to degrade smoothly.

In biochemistry, on the other hand, when an amino-acid sequence of a natural enzyme is altered, the functionality of the enzyme is extremely robust, with mostly just the catalytic rate decreasing. However, sometimes mutations in the active center of an enzyme knock-out its catalytic activity altogether. Despite this remarkable robustness, in non-linear complex networks of enzymes, drastic reactions can occur when these are altered, or when environmental conditions change.

How is it possible in principle to evolve such robust behavior? A minimal requirement for the evolution of robustness seems to be a powerful instruction set (or equivalently: a multitude of different, even redundant, operators). Then evolution can take several different pathways to solve a problem and react flexibly to changing conditions. It seems obvious that with only 16 discrete operations available Tangen (2010), such a smooth "action-landscape" cannot be achieved. A possible way out of this dilemma will be presented in the sequel.

The evolutionary model

The evolutionary task to be solved in this model is much harder than in previous models of self-replication, Ray (1991); Pargellis (1996); Adami and Brown (1994). In self-replication, the question of self and non-self is not relevant. This is the reason why in a mixed system, self-replicators will always prevail. Non self-replication requires at least two cooperating entities before a replication cycle can happen. They have to solve the problem of kinship, otherwise

they will go extinct due to parasitism. Furthermore, these two entities – in our case micro-controllers – must be sufficiently shielded from the disruptive activity of other micro-controllers in the vicinity. Therefore, asking for the emergence of non self-replication enlarges the effective evolutionary search-space greatly.[1] Everything must fit into the right spatio-temporal environment for all the programs involved in the replication cycle.

On the other hand, merging all the functionality of a program into one replication operator, as done in Tangen (1994) which we here call *atomic replication*, is a simplistic answer to the question of how new information is created from scratch. This means that a gap exists between the number of bits required in a program to encode a successful replication cycle and the size of the search space allowed for finding the correct bits, Tangen (2006, 2010); e.g., for two different sets of instructions, see Table 1. This gap can be closed with the secondary structure approach taken here.

Ribozymes or DNAzymes are biochemical equivalents to the micro-controllers used here as active components, Levy and Ellington (2003). They combine both properties: the ability to store and to process information, that is, to catalyze certain reactions. The goal of understanding the properties of ribozyme replication is also the reason why this model neglects the much easier approach of self-replication. It is the hope that non self-replication does not show the early convergence of self-replicating entities, Tangen (2002).

The model in a nutshell

Micro-controllers are situated in a spatially environment and can interact with each other. Interaction occurs through a recognition procedure. Each micro-controller can recognize a pattern, which is defined as a concatenated sequence of *Site* instructions, Table 2, in a neighboring micro-controller's program, which after recognition is then attached to the active micro-controller. The attaching micro-controller puts the address of the recognized micro-controller into its own read- or write-slot, Figure 1. The second recognition-based interaction is realized when program control is transferred from the active micro-controller to another micro-controller: this is equivalent to a subroutine call, see instruction *Call* in Table 2. The third recognition event is a register access event where the accumulator of the foreign micro-controller acts as a local register.

[1]Three different terms dealing with replication are used throughout this work: (a) self-replication means an active entity is reading its own description, allocating, or creating a new empty container and after putting a copy of the description into this new container, so releasing it into the environment, (b) non self-replication is essentially the same except that the active entity is not able to read its own description but instead the description of a neighboring entity, which it makes a copy of, and (c) atomic replication means that a single instruction in the program can perform the whole replication.

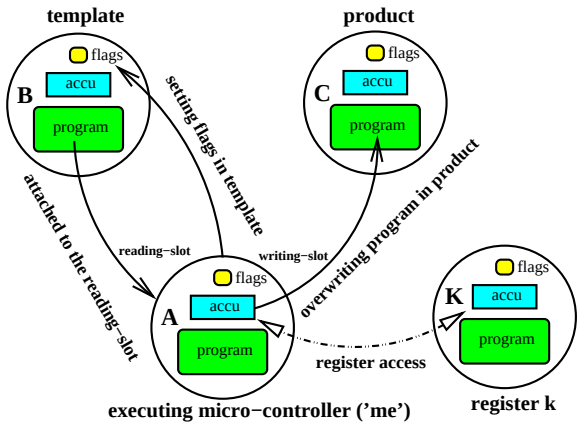


Figure 1: *How micro-controllers interact with each other.* Each interaction is realized via a recognition procedure with s concatenated bases (see *Site* instruction in Table 2). Two attachment slots are available per micro-controller. Micro-controllers attached to the reading slot (see *Load* instruction in Table 2) serve as templates, and micro-controllers attached to the writing slot (see *Store* instruction in Table 2) serve as products. Flags in other micro-controllers can be set if they are attached to the reading-slot. The standard registers are accumulators from other micro-controllers. The address of the register is the recognition site which a neighboring micro-controller exhibits.

The micro-controller[2] has input-ports (registers or a read-attached program) and output-ports (registers or the write-attached program of another micro-controller). Each instruction is divided into three parts, the cargo, conditional, and special parts. The cargo part is the parameter for the instruction in the special part, which is executed if allowed by the conditional part, see Figure 2.

A further bit is needed to allow conditional execution. The instructions in row J1 of Table 1 are also executed if J2 is specified and the ZF-flag (accumulator value 0) is active or if row J3 is specified and the PF1-flag is active. Only a few instructions have side-effects during execution, namely *Search*, *SetFA*, *SetFB,* and *Site*, see Table 2.

To summarize, each instruction is at least six bits wide, see Table 1 (left part). The data and program width are of size two bits. These two-bit words will be called nucleotides. Each replicated instruction thus requires three nucleotides and three copying operations.

The environment and physics of the simulations

The minimum case of sustained replication in this work occurs when two machines replicate each other – in that

[2]A Harvard architecture (`http://en.wikipedia.org/wiki/Harvard_architecture`) has been chosen because it naturally allows to use different data widths without affecting the instruction sequence.

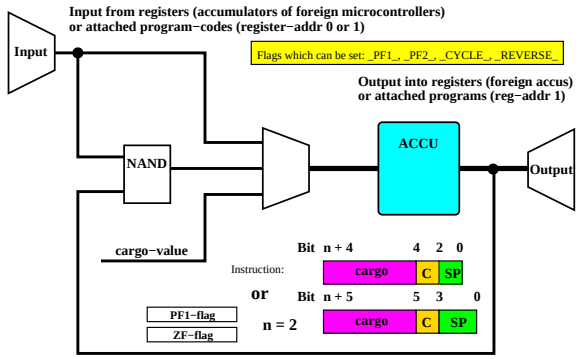


Figure 2: *Structure of the micro-controller.* The micro-controller uses two bits of the special instruction section (SP). The condition part (C) is two-bits wide. The width of the cargo depends on the experiments, usually $n = 2$ using quaternary encoding. This leads to a six bit micro-controller in the simple case. The zero-flag (ZF) and PF1-flag (PF1) are used for conditional execution (see Table 4). Input and output either comes from or is sent to other micro-controllers.

	S0	S1	S2	S3
J0	*Set*	*Site*	*Load*	*Store*
J1	*End*	*Call*	*SetFA*	*SetFB*
ZF	“	“	“	“
PF1	“	“	“	“

	S0	S1
J0	*Set*	*Site*
J1	*Load*	*Store*

Table 1: *Instruction sets which exhibit emergence of replication.* The left set is the most powerful case which is still able to develop emergence of replication. The right set shows the simplest non-trivial case. Though emergence of replication is possible, the diversity of the emerging population is limited. The NAND-instruction, shown in Figure 2, was omitted in this particular instruction set. Many different instruction sets can be chosen as long as they represent a superset of the minimal instruction set given in the right table.

Instr.	Description
Load	Load a value from a register into the accumulator. The cargo specifies the address of the register. Register 0 points to the micro-controller attached to the reading slot. Register 1 points to the micro-controller attached to the writing slot. With no micro-controller attached, a search is initiated. Prepended *Site* instructions increase the specificity of register addressing. When there are no previous *Site* instructions or accesses to registers 0 or 1, a random search is done. If no suitable micro-controller is found, this instruction has no effect.
Store	Store the accumulator in a register. The cargo specifies the address of the register. Register 1 points to the micro-controller attached in the writing slot. Register 0 points to the micro-controller attached to the reading slot. With no micro-controller attached a search is ignited. Prepended *Site*-instructions increase the specificity of register-addressing. When there are no previous *Site* instructions or accesses to registers 0 or 1, a random search is done. If no suitable micro-controller is found and address 1 is accessed, the program is stopped to reduce processing costs.
Call	Transfer execution to the micro-controller specified in the cargo part of this instruction. Accumulator and attachment slots are transferred to the new micro-controller. The current program is stopped after this call. Prepending *Site* instructions increase the specificity of the micro-controller addressing, where these *Site* instructions are combined with the cargo part of the *Call* instruction to one big virtual recognition-site. If no appropriate micro-controller is found, the instruction has no effect.
Set	Preset the accumulator with the value provided by the cargo part.
Site	Define a recognition site, either to be recognized by others or to actively serve as an address. Used with instructions *Call*, *Load* and *Store*.
SetFA	If a machine is attached to the reading-slot (e.g. after accessing register 0) then certain flags can be set in the machine Tangen (2010).
SetFB	Set flags in the executing machine Tangen (2010).
End	This instruction is required with fixed-length programs. Variable length programs can omit the declaration of the end of the program because it is already physically given.

Table 2: *A few basic instructions understood by the micro-controllers.* Currently 64 different instructions are implemented and used by the secondary structure approach. The *End*-instruction is a special case, only needed with fixed-length programs.

Bits	cargo	C	SP
Site	xx	00	01
Load	00	00	10
Store	01	00	11
End	–	01	00

Bits	cargo	C	SP
Site	xx	-	01
Load	00	-	10
Store	01	-	11

Table 3: *Two examples of minimum replicator programs.* The relevant instruction sets are given in Table 1. The bits (xx) in the cargo part of the *Site*-instructions are arbitrary but needed to be stabilized throughout evolution. The minimal program on the left requires the system dynamics to find 22 correct bits. The right program needs 12 bits and a program length of three instructions.

Bit 1	Bit 0	Meaning
0	0	Instructions are always executed.
0	1	Instructions are always executed.These instruction have conditional counterparts, see Table 1.
1	0	Only executed if ZF-flag (ACCU == 0) is set.
1	1	Only executed if the PF1-flag is set.

Table 4: *Conditional part of an instruction.* Instructions can be executed if certain conditions are fulfilled, such as the accumulator (ZF-flag) being zero or the flag PF1 being set in the status-register of a micro-controller.

sense a machine can be thought of as a ribozyme, although in reality only the programs and a few state-variables are copied! Many former studies McCaskill (1988); Tangen (1994) found that if self-replicators compete with non self-replicators, self-replicators prevail, simply because perturbations caused by missed templates are not possible in the self-replicating case. The minimal programs are shown in Table 3.

So far the best way to minimize the number of bits needed is to mimic physical behaviors and make use of this assumption by introducing side-effects for some appropriate instructions Tangen (2010). A universal, programmable instruction set was devised in that work, which needed only 22 significant bits in the minimum-replicator case. With these 22 bits, a non-trivial evolution from the starting point was shown. An even simpler program, Table 3 (right table), needs only to have 12 bits specified by the system. Even though the system has programs such as these, with their simple, non-trivial instruction sets that have the potential to exhibit replication, it is unlikely that in any particular collective execution in this system that replication will occur. This difficulty is due to the large class of perturbations that can be exerted by uncoordinated *Store*-instructions.

Convolution of programs (secondary and tertiary structure)

Mapping the primary structure of a program onto a secondary or tertiary structure promises better evolvability of the resulting replication system Kimura (1990); Wagner (1985). A simple approach is to use a graph whose nodes represent instructions and whose edges represent traversals according to the nucleotides given. Consider a graph for which n is the data-/cargo-width and m is the instruction-width in bits, and whose outbound degree is k. This graph describes a kind of machine known as a Moore automata[3], and in the case where $n < m$, it provides the simplest method which allows redundancy in the secondary landscape. Figure 3 describes a simple, non-trivial version of this automaton with $k = 2$. It is a matter of choice whether the accumulator values of the micro-controller are considered part of the Moore automaton, as shown in part (a), or are defined by the program, as in part (b). The latter case is most natural for the Harvard architecture, with its strict separation of data- and command- path. The first variant is more akin to the natural biochemical situation where accumulator values are only indirectly present in the form of co-factors. From an evolutionary point of view, the search space decreases considerably in the first case as does the number of degrees of freedom.

On the other hand, the extreme case of a fully connected graph is equivalent to the situation where $k = n \geqq m$, which is nothing but a system without any secondary structure.

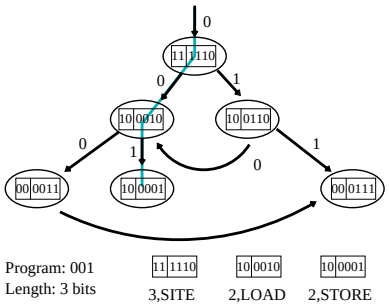
A quaternary system is chosen here[4]: $n = 4$. The number of instructions is $m > 16$. Furthermore, the values of the accumulator are still set by the program directly, as in case (b) in Figure 3.

Nucleotides in a program no longer represent instructions. They represent commands to move along the instructions in the graph and thus change the current state of the Moore automaton. Increasing the power of the instruction set means inserting further nodes with the corresponding edges into the graph. Of course, the graph must not contain nodes which cannot be reached.

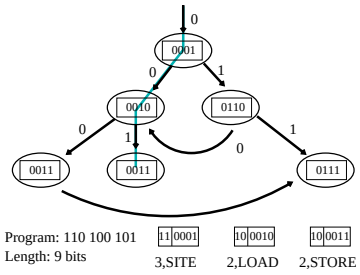
Trivial replicators If for example the size of a random graph is sufficiently large, replicators will trivially emerge, even without using replication. Two cases can happen: Firstly, it is conceivable that a program emerges which contains only a sequence of zeros, where these instructions are interpreted as write-zero operators. Such a program represents a simple auto-catalytic process without any special notion of evolution or information processing. Preliminary experiments have shown that most random graphs with a minimum size do exhibit such a 'chemical'-nature. Secondly,

[3] http://en.wikipedia.org/wiki/Moore_machine

[4] Schuster and Stadler (1994) argue that quaternary RNA encodings have best evolutionary properties. Their assumptions on RNAs certainly do not hold in the current model, but without further investigation, taking a quaternary system is an initial choice.



a) Values and commands per state



b) Only commands per state

Figure 3: *Two variants of a simple non-trivial directed graph.* These graphs can be interpreted as Moore automata: (a) with a program length of only three bits the commands *R*elease, *Load* and *Site* can be issued with their respective accumulator values (3, 2, 2) and (b) accumulator values are not part of the Moore automaton and have to be provided by the program. This increases the program length to 9 bits, giving the same functionality as in the upper part.
Of course, the number of degrees of freedom in case (a) is much less than in case (b). On the other hand, the search space in case b) is much larger than in case a) and evolution needs to search longer to find the specific functional sequences. With only one bit available the graphs must have an outgoing connectivity degree $k = 2$. Each node can have arbitrarily many inbound connections.

a sequence of zeros only can be equivalent to a replicator program. To create programs with identical nucleotides is much easier than to sustain a complicated sequence of zeros and ones [5].

To avoid these trivial solutions, the Python script creating these graphs looks for short cycles. They are eliminated via a randomization procedure. A detected cycle is broken up by the overwriting of one node on the cyclic path with a random node. After several passes through the whole graph, almost no short cycles remain. A successful example of emergence of replication of a non-trivial replication can be seen in Figure 6.

[5]Problem of frame shifts `http://en.wikipedia.org/wiki/Frameshift_mutation`

Site	*Load*	*Store*
	Loadwa	*Storewa*
	Loadwb	*Storewb*
	Loadf	*Storef*
	Cload	*Cstore*
	Zf_cload	*Zf_cstore*
	Pf1_cload	*Pf1_cstore*

p	n
0.0	309
0.01	468
0.03	1302
0.05	1932
0.1	4057

Table 5: *Searching potentially replicating programs.* To increase the probability of creating replicative programs additional instructions (*Site* and variants of *Load* and *Store* instructions, left table) have been added with certain probabilities given in the table on the right. A recursive search algorithm finds all occurrences of potential replicator programs and marks these as possible starting nodes in the secondary structure. The second column in the right table shows the frequency of possible replicator programs in a graph of 8192 nodes and the probabilities given in the first column. Only the operators as such are considered and not the instruction parameters in the cargo values, see Figure 2. In this case a program with three instructions and a cargo-width of 2 bits has six unspecified bits to be found by the dynamics of the system.

Means to increase the probability of emergence Furthermore this Python script looks for possibly viable minimal replicator-programs in the graph. It searches recursively for instruction sequences [*Site*, *Load*, *Store*] and variants, see Table 5, to extract suitable entry points for newly created micro-controllers. Suitable entry points into the Moore automaton (i.e., starting nodes) increase the probability of emergence of replication. From an evolutionary point of view, these entry points are neutral: they do not change the physics in the system but rather provide hints for the dynamics to find replicative sequences.

To further increase the probability of starting replication, additional {*Site*, *Load*, *Store*} instructions can be inserted at random into the graph. Table 5 (right) shows how many suitable entry points are found by the Python script depending on the probability of adding one of these three instructions or their relatives. If all three instructions are inserted equiprobably with, e.g., probability of 10%, then one of the three instructions will be chosen with a probability of 30%. As expected, the higher the probability, the more replication programs there are in reach of an arbitrarily chosen entry point (state) for the automaton. This can also be seen in Figure 4, where the distances of viable minimal programs from each node in the graph are plotted. These distances measure the effort of the recursive search algorithm to find such viable programs. Only the special part of the instructions, as such, are taken into account and not the cargo values, see Figure 2. This means that the programs found are probably not replicating at all but have a high propensity if the cargo values can be altered by the dynamics. If no viable program

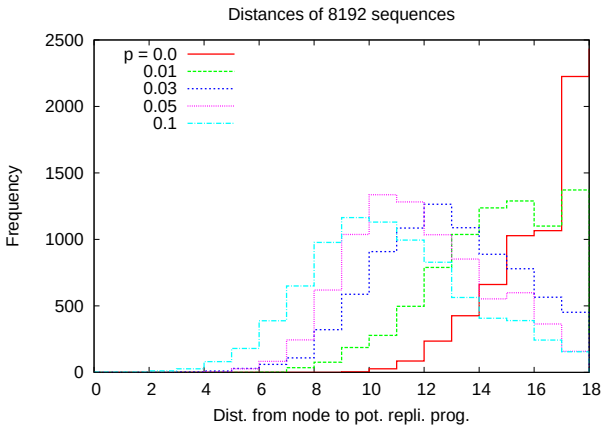


Figure 4: *Costs between each node of the graph to the next potential replicating program.* The more additional *Site*, *Load* and *Store* instructions are added, the more probable it is to find a replicative program by accident. The numbers on the x-axis are arbitrary and essentially only reflect the inner properties of the recursive-search algorithm.

is found by a node, the maximum cost is assumed, see the right box in Figure 4. The distribution of viable programs from a given node in the graph is not a sharp one. There is reason to hope that the wide distribution helps to find a new niche for replication, but this has still to be demonstrated.

Computational results

The software used (EvoCpu_i686) is custom-developed[6]. The space is divided into containers which are randomly selected and processed. Each micro-controller in a processed container is allowed to execute a certain number of instructions. Each executed instruction needs a certain amount of energy. Several instructions and their "physico-chemical" effects can be fine-tuned by such energy coefficients.

To illustrate how replication emerges, an extract of four containers from a successful experiment with approximately four million micro-controllers (i.e., 18 non-zero micro-controllers, with two of them having a minimal replicator program) were put into an empty, smaller system, and evolution was started again. Eight consecutive generations are shown in Figure 5. Common features of these replicating systems are: (a) they do not use all the available space and (b) irregular spatial structures emerge right from the beginning. If these experiments are done on a single CPU, then the evolutionary outcome is deterministic. No mutations or other typical genetic algorithm operators are involved here.

[6]The software is available for download at `http://www.biomip.de/Uwe/projects/EvoCpu`. It is suitable for SMP (symmetric multiprocessor)-machines. Further details on the model are also provided.

An old question asks whether the emergence of replication has any precursors and whether supporting these precursors can increase the probability of the emergence of replication. The first occurrence of replication in the above experiment has been traced down to two micro-controllers, see Figure 6, one of them a ligating program (center picture) and the second a minimal replicator (lower picture). With a high probability ($p = (70/100) = 0.7$ in this example) these two programs are sufficient to develop two minimal programs which will then be able to replicate each other, commencing the evolutionary process. As one can see from the colors in Figure 5, the diversity in the system is high right from the beginning, and remains so with many interesting structures developing (data not shown).

Discussion and conclusions

The work presented shows that with the help of customized micro-controllers, non self-replicating programs can and eventually will emerge. This is a much harder task for evolving systems than in the former models of self-replication.

Replication can only be realized if two replicating programs (or in biological terms, two ribozymes) cooperate in such a way that both of them replicate each other simultaneously and that no other entities interfere. This scenario of non self-replication seems to be more suitable when studying the transition from non-living to living matter. Self-replication requires a protecting hull, and this hull or membrane has to be encoded also by the self-replicating system, otherwise an exponential proliferation would not be possible. In addition, the problem of nutrients or waste passing the hull or membrane needs to be solved right at the start in the self-replicating system.

Convolution of a program into a secondary structure solves the notorious problem of missing bits to code for the many operators required and to circumvent the brittleness problem. Furthermore, and even more important, physical and chemical properties of the system can be naturally encoded (mapped) into the secondary structure without having to change the micro-controller machinery. Having said that, this particular solution of a secondary structure can hardly be found in nature. Understanding the emergence of replication would make it possible to incorporate further biochemical details. The secondary structure also provides a way to abstract the details of physics and chemistry. This can facilitate higher forms of organization because they are no longer perturbed by detailed settings.

When looking at the transition from non-living matter to living matter, the question arises of whether there are any precursors to replication. However, this appears to be unlikely. In the example shown, a non-replicating program works in conjunction with a replicator to create the second required replicator, see Figure 6 (center). But restarting the extracted system only one generation earlier fails to show any emergence, even if very large parts of the original sys-

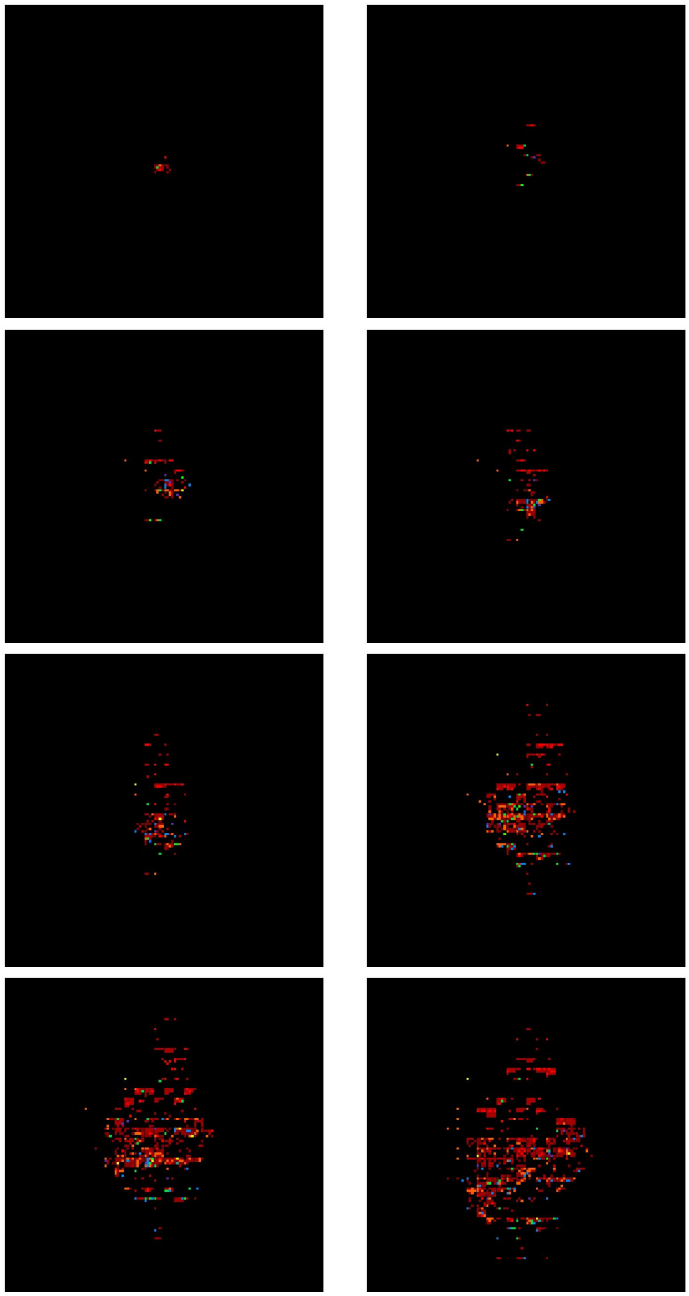
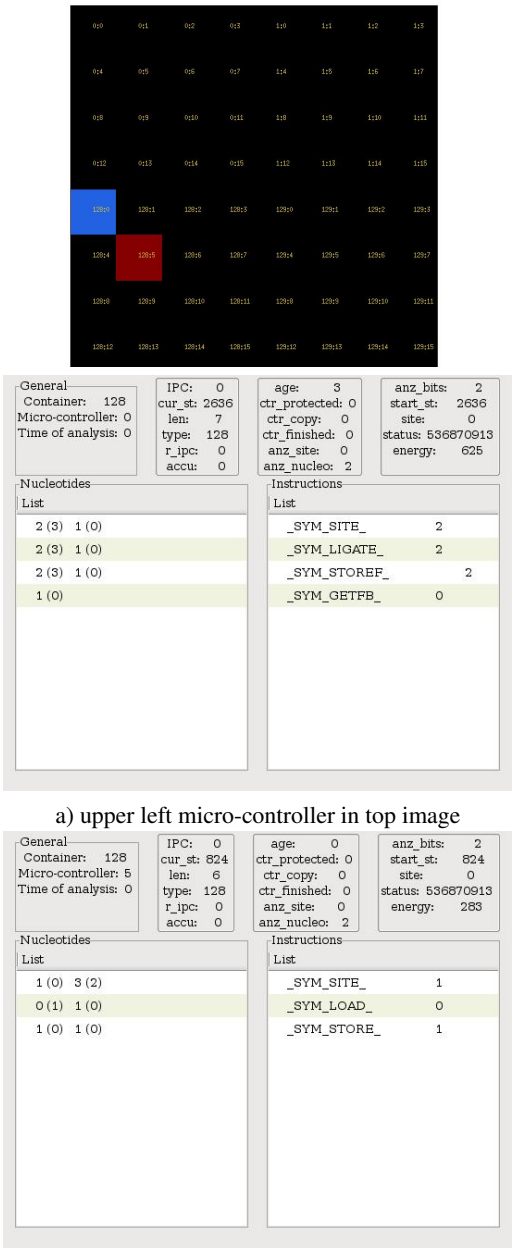


Figure 5: *Sequence showing the spatial fingerprints of the replicating programs at the onset of replication.* An area of 2x2 containers was extracted from a data-log shortly after replication emerged. This area was transplanted into a new, smaller, empty system and each image shows the fingerprints of replication in consecutive generations. The asymmetric growth of the cluster is a consequence of activity of perturbing parts in programs.



a) upper left micro-controller in top image

b) lower-right micro-controller in top-image

Figure 6: *Seed extracted from the very first generation ($t = 347$) of the emergence of replication (system size 2048⋆2048 micro-controllers).* In the top image, the first two micro-controllers (red and blue) are shown acting as seeds for replication. From these two programs, two copies of the red program shown (machine-id 128:5) are likely develop with high probability. The parameter of the instructions is shown to the right of the mnemonics (cargo value, see Figure 2). The left part gives the same information, but now from a nucleotide point of view, without printed mnemonics. The numbers in parentheses are the complements of the nucleotides (in this case Watson-Crick complement). See also the irregular bits set in the instructions, which prove that this onset of replication is not due to a trivial unchanging sequence of only zero- or one-bits.

tem are extracted and replayed. If there are precursors before replication, they cannot be numerous and only occur right before the onset of non self-replication. Or they are so special and specific that they do not exhibit a broad basin of attraction. The major bottleneck does not seem to be the occurrence of replicator-programs as such (in the example shown above, only twelve specific bits have to be available twice in a neighborhood) but the disturbance of unrelated micro-controllers interfering with the replicating process – in the above case there are five containers with $4 \cdot 16 + 14 = 78$ machines.

In previous work, non self-replication emerged only if there were at most the 22 unknown bits (without secondary structure) required for the shortest replicator program Tangen (2010). The number 22 is not important, but it gives a hint as to the difficulty of the search problem. Using the secondary structure allows us to adjust the physics of the system from a few bits per minimal replicator to a potentially arbitrarily large number of bits. However, the most important advantage of the secondary structure is the ability to use many more instructions than there are bits available for encoding, and to fine-tune the physical environment as needed. Furthermore, different areas in the directed graph represent different physics, thus allowing multiphysical experiments to be conducted. Species with their center-points moving along the directed graph represent a case of hardware evolution.

Acknowledgments This work was partly financed by the European Center of Living Technology in Venice, Italy and the Programmable Artificial Cell Evolution project PACE EU-IST-FP6-FET-002035 of the European Union. In addition, the research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under Grant Agreement No. 222422, ECCell - Electronic Chemical Cell (a project in the EU FP7-IST-FET Open Initiative).

References

Adami, C. and Brown, C. T. (1994). Evolutionary learning in the 2d artificial life system "avida". In Brooks, R. and Maes, P., editors, *Artificial Life IV*, pages 377–381. MIT Press, Cambridge, MA.

Barricelli, N. A. (1962). Numerical testing of evolution theories Part I: Theoretical introduction and basic tests. *JournalActa Biotheoretica*, 16:69–98.

Benkö, G., Flamm, C., and Stadler, P. F. (2005). Explicit collision simulation of chemical reactions in a graph based artificial chemistry. *Lec. Note Comp. Sci*, 3630:725–733.

Dittrich, P., Ziegler, J., and Banzhaf, W. (2001). Artificial chemistries - a review. *Artif. Life*, 7:225–275.

Fontana, W. (1991). Algorithmic chemistry: A model for functional self-organization. In Langton, C. G., editor, *Artificial Life II*, pages 159–202. Addison-Wesley, Reading, Massachusetts.

Holland, J. H. (1976). Studies of the spontaneous emergence of self-replicating systems using cellular automata and formal grammars. In Lindenmayer, A. and Rozenberg, G., editors, *Automata, Languages, Development*, pages 385–404. North Holland Publishing Company, Amsterdam.

Kimura, M. (1990). Some models of neutral evolution, compensatory evolution, and the shifting balance process. *Theo. Pop. Biol.*, 37:150–158.

Levy, M. and Ellington, A. D. (2003). Exponential growth by cross-catalytic cleavage of deoxyribozymogens. *Proc. Natl. Acad. Sci. USA*, 100:6416–6421.

McCaskill, J. S. (1988). *Polymer Chemistry on Tape: A Computational Model for Emergent Genetics*. Max–Planck–Society, Göttingen, Germany. Report.

McCaskill, J. S. and Niemann, U. (2001). Graph replacement chemistry for DNA processing. *Lec. Note Comp. Sci*, 2054:263–270.

Menabrea, L. F. (1842). The analytical engine: Invented by Charles Babbage. *Bibliothèque Universelle de Genève*, No. 82.

Pargellis, A. N. (1996). The spontaneous generation of digital "life". *Physica D*, 91:86–96.

Ray, T. S. (1991). An approach to the synthesis of life. In Langton, C. G., Taylor, C., Farmer, J. D., and Rasmussen, S., editors, *Artificial Life II*, pages 371–408. Addison-Wesley, New York.

Schuster, P. and Stadler, P. F. (1994). Landscapes - complex optimization problems and biopolymer structures. *Computers Chem*, 18:295–324.

Tangen, U. (1994). The extension of the quasi-species to functional evolution. In *PhD Thesis*. Uni Jena, Jena.

Tangen, U. (2002). An evolvable micro-controller or what's new about mutations? In Langdon, W. B. e. a., editor, *Genetic and Evolutionary Computation Conference GECCO 2002*, pages 178–187. Morgan Kauffmann.

Tangen, U. (2006). From evolving software towards models of dynamically self-assembling processing systems. In Jost, J., Reed-Tsochas, F., and Schuster, P., editors, *Proceedings of ECCS 2006*, pages 50, p85.pdf. ECSS, Paris.

Tangen, U. (2010). Enzyme-like replication de novo in a micro-controller environment. *Artificial Life Journal*. To appear.

von Neumann, J. (1966). *Theory of Self-Reproducing Automata*. Burks, A. W. University of Illinois Press, Urbana.

Wagner, A. (1985). Robustness and evolvability: a paradox resolved. *Proc. R. Soc. B*, 275:91–100.

Yoshii, S., Ohashi, S., and Kakazu, Y. (1998). Self-organized complexity in computer program ecosystem. In *Artificial Life 6*, pages 483–488. Springer, Berlin, Heidelberg.

What do You Want to do Today? Relevant-Information Bookkeeping in Goal-Oriented Behaviour

Sander G. van Dijk, Daniel Polani and Chrystopher L. Nehaniv

Department of Computer Science, University of Hertfordshire, Hatfield AL10 9AB, United Kingdom
s.vandijk@herts.ac.uk

Abstract

We extend existing models and methods for the informational treatment of the perception-action loop to the case of goal-oriented behaviour and introduce the notion of *relevant goal information* as the amount of information an agent necessarily has to maintain about its goal. Starting from the hypothesis that organisms use information economically, we study the structure of this information and how goal-information parsimony can guide behaviour. It is shown how these methods lead to a general definition and quantification of sub-goals and how the biologically motivated hypothesis of information parsimony gives rise to the emergence of behavioural properties such as least-commitment and goal-concealing.

Introduction

The world is a complex place. Millions of years of evolution have created an environment with intricate relationships, structure and many things that an organism living in it has to look out for. It is no surprise then that organisms invest a lot of energy in the processing of all the information available to them. For instance, the retina of a resting blowfly accounts for 10% of its energy consumption and for the human brain this amount is estimated to be 20% (Laughlin et al., 1998).

It is unlikely that an organism would spend all this energy if it is not crucial; individuals that limit their information intake and processing to the necessary minimum and allocate the rest of their energy to behaviour that is more relevant to survival or reproduction will outperform ones that waste energy on useless information processing. Also, even though this means an organism uses information economically, it is plausible that an organism still often operates at the limit of its information processing bandwidth and that there is an evolutionary drive to do away with unused capacity, similar to the degeneration of useless eyes in cave-dwelling fish (Jeffery, 2001). We will refer to these assumptions as the *information parsimony* hypothesis.

We are interested in the necessary principles of life and lifelike behaviour. The hypothesis of information parsimony hints that information acquisition and processing capabilities are part of these fundamental requirements. In the vein of the Alife motto "life as it could be", we use minimal models of agents and their informational properties to study these basic requirements of life. The substantial history of this approach shows that clear statements can be made about information processing bounds and how these influence the structure of sensory and behavioural systems and embodiment (Barlow, 1961; Brenner et al., 2000; Nehaniv et al., 2007; Pfeifer et al., 2007; Polani, 2009).

The information parsimony hypothesis has given rise to a body of research on the informational treatment of the perception-action loop of agents and the interactions with their environment. It has been shown that this can lead to global, fundamental insights in necessary bounds on behaviour (Polani et al., 2006), evolution of coordination (Sporns and Lungarella, 2006), intrinsic drives (Klyubin et al., 2008), successful search strategies for tasks with sparse information (Vergassola et al., 2007), and behaviour structuring (van Dijk et al., 2009). These results are general in the sense that they do not require a specific model of brain mechanics. In this paper we will extend this previous work to the more specialised, though sufficiently general case of *goal-oriented* behaviour.

Goals

There are many cases, both in biological and in artificial settings, where the environment can be seen as offering rewards for certain types of behaviour. These rewards can range from as clear-cut as a treat given by a dog trainer to as diffuse as persistence. When such a reward measure is available to an agent, it can often be regarded as performing a certain task with an accompanying end-goal (Montague et al., 2004).

Although successful behaviour that appears goal-oriented is achieved, note that we do not want to imply that the organism or agent necessarily maintains an explicit representation of this goal. However, there is evidence for the case that human adults encode actions in terms of their outcomes (Hommel et al., 2001). Furthermore, brain structures have been located where activity is highly correlated to the goal of observed behaviour (Hamilton and Grafton, 2006), indicating an evolutionary drive towards goal-centred thought.

Moreover, recent research is beginning to show evidence for neural correlates of an individual's own goals, not limited to human brains, e.g. Saito et al. (2005); Spiers and Maguire (2006). Therefore we will adopt the viewpoint that certain behaviour, or in any case episodes of behaviour, can be seen as being driven by a concrete, identifiable goal.

Goal Information

We extend methods for informational treatment of the perception-action loop to explicitly include goal-directed behaviour. Here an agent needs to actively maintain information about its current goal. In the case of human beings it has been consistently argued that this is performed by the pre-frontal cortex (Montague et al., 2004). As any information processing this takes effort and consumes energy, thus, following the information parsimony hypothesis, it is expected that organisms attempt to optimise this process. Here therefore we study the necessary bounds of goal-information that has to be maintained at a given time. We show how these bounds can guide behaviour and that they can give rise to the emergence of certain behaviour properties, such as least-commitment planning, which traditionally is explicitly designed into computational approaches (Weld, 1994), and goal-concealing.

In the following two sections we will give a short introduction to concepts and notation used in this paper and an overview of the informational methods used to study the perception-action loop. Next, we introduce the main concept of the research presented here: *relevant goal information*. The effects of this quantity on behaviour and interpretations of these effects are then presented using a navigation-task example. Subsequently, we show how relevant goal information gives rise to a natural notion of *transition points*. Finally, we will relate our results to previous work and give a general discussion in the last section.

Concepts and Notation

When we talk about information, we refer to the information-theoretical formalism introduced by Shannon (1948). Here, the main elements are random variables, which we denote with capital letters, e.g. X. Such a variable can assume a specific value (small letter, x) from a given alphabet (curved capital, $\mathcal{X}$), subject to a probability distribution over the possible values: $\sum_{x\in\mathcal{X}} Pr(X = x) = 1$. To improve legibility we will, by abuse of notation, write $p(x)$ for both the entire distribution and for the probability that variable X assumes the value x, determined by the context. We use $p(x, y)$ and $p(y|x)$ for joint and conditional probabilities, respectively.

A probability distribution implies an 'uncertainty' about the value of a random variable. This uncertainty is quantified as the *entropy* $H(X) = -\sum_x p(x)\log p(x)$. We take 2 as the base of the logarithm, so that the unit of entropy is bits. Alternatively, the entropy can be seen as how much information on average is gained when learning the value of a random variable. The conditional entropy $H(Y|X) = -\sum_{x,y} p(x, y)\log p(y|x)$ determines the amount of uncertainty left about Y when the value of X is know.

The amount of information that on average is available both in X and Y can be calculated with the *mutual information* $I(X;Y)$. The mutual information can be defined as $I(X;Y) = H(Y) - H(Y|X) = H(X) - H(X|Y)$, which leads to the interpretation that it is the decrease in uncertainty about one variable when the value of the other one is known.

Finally. the expected value of a random variable is written as $\mathbb{E}[X]$, or $\mathbb{E}[X|\theta]$ when the value is conditioned on some parameters θ. The expected value is equal to the sum of the possible values, weighed by their probability: $\mathbb{E}[X] = \sum_x p(x)x$. Similarly, we can for instance write the conditional expected value of a function as $\mathbb{E}[f^\theta(X, y)|\theta] = \sum_x p(x|y, \theta)f^\theta(x, y)$.

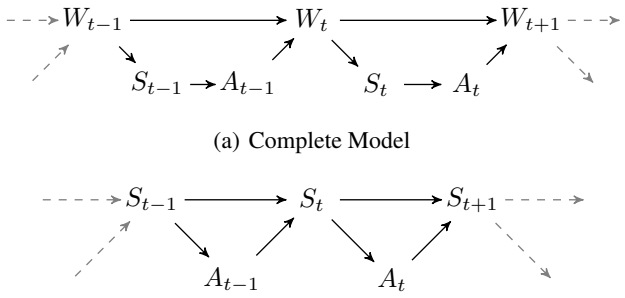
For a more elaborate background on the information-theoretical concepts and notation used in the current paper see Cover and Thomas (1991).

The Perception-Action Loop

An agent is embodied and situated in an environment; it has direct contact to the environment through its sensors and actuators. Information about the world is obtained through the sensors and influence the agent's actions, which in turn can affect the environment. This results in a *Perception-Action loop (PA-loop)* and, following Klyubin et al. (2004), we model this loop as a *causal Bayesian network* (CBN), as shown in Fig. 1(a). Such a network represents the relationship between the agent and the environment. At each time step t the agent perceives part of the state of the world w_t, resulting in a sensor state $s_t \in \mathcal{S}$. A fully reactive agent chooses its action $a_t \in \mathcal{A}$ based solely on this state. Its *policy* π defines the probability of performing these actions: $\pi(a_t|s_t) = p(a_t|s_t)$. When the agent performs an action, the world state is changed according to the *state transition probability distribution* $\mathcal{P}^{a_t}_{w_t,w_{t+1}} = p(w_{t+1}|w_t, a_t)$.

Without loss of generality, in the rest of this paper a simplified version of this model is used. It is assumed that the world is fully accessible to the agent, i.e. the sensor state reflects the full state of the world. For the CBN, this means that the world and sensor nodes can be collapsed, resulting in the network shown in Fig. 1(b). Consequently, we will use the term 'state' interchangeably for both world and sensor state.

As outlined in the introduction, we consider agents that operate in an environment that rewards certain behaviour. We are interested in how in this case the combined structure of the world and rewards can influence the structuring of behaviour. We assume that the reward that the agent receives is quantifiable. For instance, in a food-searching task the



(a) Complete Model

(b) Fully Accessible World

Figure 1: Causal Bayesian network of the perception-action loop, unrolled in time, showing (a) the complete model and (b) the case when the world is fully accessible.

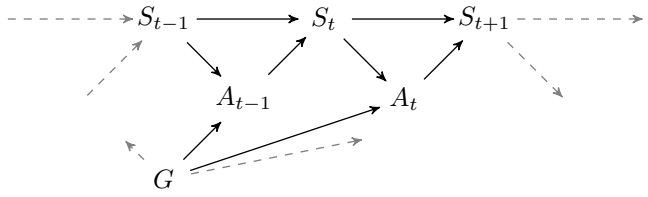


Figure 2: Causal Bayesian network of the perception-action loop, extended with the goal node.

agent can be presented a reward related to the nutritional value of the food when it is found. Another commonly used scheme is to represent the energy spent to perform a task as a penalty or negative reward for each time step that the goal is not reached. We will use the first model, as detailed further on.

These rewards are modelled by an *immediate-reward function* (Sutton et al., 1999) which gives the immediate reward that an agent will receive for performing action a_t when in state s_t and consequently finding itself in state s_{t+1}: $\mathcal{R}^{a_t}_{s_t,s_{t+1}} \in \mathbb{R}$. Given this function we can define the *state-action value function* (or *utility function*) $U^\pi(s_t, a_t)$ which gives the expected future reward of taking action a_t when in state s_t and subsequently following policy π (Sutton et al., 1999):

$$U^\pi(s_t, a_t) = \sum_{s_{t+1}} \mathcal{P}^{a_t}_{s_t,s_{t+1}} \left[\mathcal{R}^{a_t}_{s_t,s_{t+1}} + \gamma \mathbb{E}[U^\pi(s_{t+1}, A_{t+1})|\pi]\right], \quad (1)$$

where $\gamma \in [0, 1]$ is a discount factor to model preference for short term (low γ) or long term reward (high γ).

In this setting, a rational agent that performs goal-directed behaviour will try to gather as much reward as it can as fast as possible, effectively attempting to find an optimal policy π^* maximising the expected value of (1):

$$\pi^* = \arg\max_\pi \mathbb{E}\left[U^\pi(S_t, A_t)|\,\pi\right] \quad (2)$$

$$= \arg\max_\pi \sum_{s_t,a_t} p(s_t, a_t) U^\pi(s_t, a_t) \quad (3)$$

$$= \arg\max_\pi \sum_{s_t,a_t} \pi(a_t|s_t) p(s_t) U^\pi(s_t, a_t). \quad (4)$$

Information in the PA-Loop

With the formalisms outlined in the previous sections in place, we can look at the informational properties of the PA-loop. The arrows in the CBNs of Fig. 1 can be regarded as channels; the world 'transmits' information which the agent receives through its sensors and in turn the agent 'injects' information into the world through its actuators. The well established field of information theory then provides us with the tools to answer questions about the PA-loop in a concrete way in the terms of Shannon information (Shannon, 1948).

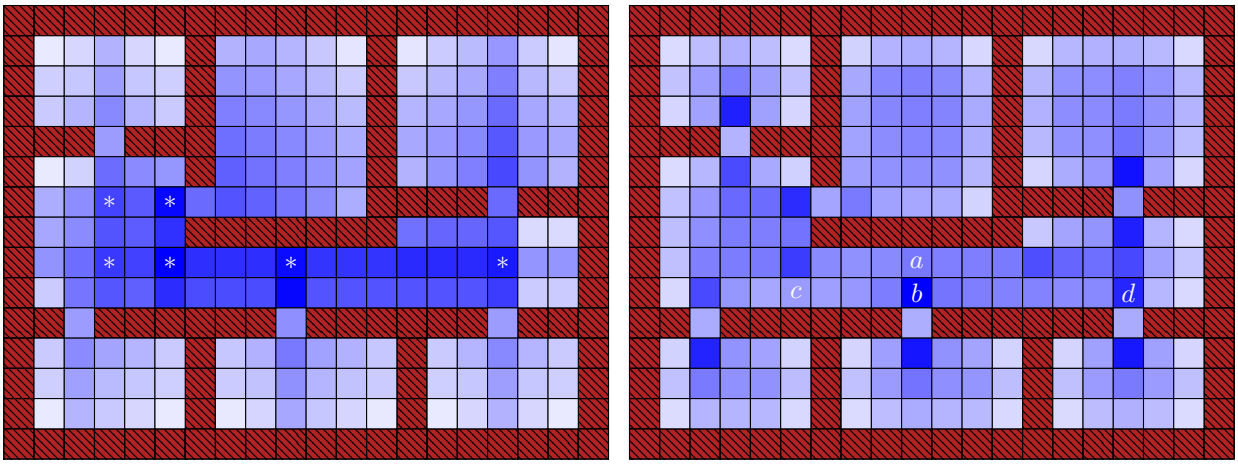
For instance, we can determine the amount of information that an agent on average takes in through its sensors to determine its actions using the mutual information between sensor states and actions $I(S_t; A_t)$. Not all information that is available in S_t is relevant to its current task and, following the hypothesis of information parsimony as discussed in the introduction, we assume that the agent will aim to minimise this quantity. The lower bound of the necessary amount of information intake to be able to achieve a certain level of utility can be quantified using the paradigm of *relevant information* (Polani et al., 2006), and is done by solving the following problem:

$$\min_{\pi(a_t|s_t)} \Big[I(S_t; A_t) - \beta \mathbb{E}\left[U^\pi(S_t, A_t)|\pi\right]\Big]. \quad (5)$$

The solution is a policy which minimises the state-information used to select actions while maximising the expected utility achieved by this policy. The parameter β can be varied to trade-off utility and information requirement; low β promotes information parsimony, high β puts more weight on utility. When β goes to infinity, the policy found will become optimal and the minimum amount of state information needed to act optimally is given by $I(S_t; A_t)$. As shown by Polani et al. (2006), the problem of (5) can be solved with an iterative algorithm that interleaves traditional algorithms of information theory (rate-distortion (Blahut, 1972)) and reinforcement learning (value iteration (Sutton and Barto, 1998)). This algorithm has the important property that the solution of (5) simultaneously fulfils (1).

Relevant Goal Information

The methods for relevant information are generally applicable to any case where a reward function can be defined. However, it is restricted to the analysis of a single task. Here



(a) Relevant Goal Information (b) Goal Information Transitions

Figure 3: Grid world example for relevant goal information. Walls are denoted with a brown, hashed background. The remaining free cells comprise the set of states $\mathcal{S}$. The goal G is uniformly distributed and its alphabet $\mathcal{G}$ consists of the empty cells within the six rooms. The agent can perform four actions: move north, east, south or west. When such an action would move the agent to an occupied cell the action has no effect. The shading of the background of the free cells indicates (a) the total amount of relevant goal information for each cell and (b) the amount of new relevant goal information when arriving in a cell. Dark blue shading for high amount, light blue or white for low amounts The meaning of the asterisk and letter marks is explained in the text.

we will extend the model of the PA-loop to enable us to handle an agent that could perform different tasks. To do so, we focus on the common case where this task can be determined by reaching a distinct goal. Here we do not discern how the current goal of an agent is selected; it can be imposed externally, such as a command given to a dog by its master, or it may be an intrinsically determined goal, as in the case of a hungry predator that decides to catch a certain prey. Instead, we only are concerned about the decision making process once a goal is given.

We introduce the new random variable G. The value of this variable, g, represents the current goal of an agent. Figure 2 shows how the CBN of the PA-loop is extended with this new variable. Note that we do not aim to study the case of an agent having several simultaneous goals. Rather, we concentrate on agents that select a specific goal from a discrete set of possible goals $\mathcal{G}$. After this selection the goal is fixed, until the goal is achieved or abandoned.

The new CBN shows that the policy now also depends on the current goal: $\pi(a_t|s_t, g) = p(a_t|s_t, g)$. Also, each separate goal gives rise to a distinct immediate reward function and thus to a separate goal-dependent utility function $U^\pi(s, g, a)$.

This extension of the model introduces an additional information source; apart from *sensory* information the agent now also needs to maintain and process *goal* information to guide its actions. Per the information parsimony hypothesis this is assumed to be costly and therefore we are interested in determining lower bounds on this amount of information needed to achieve a given performance. Analogous to the sensory case we term this the relevant *goal* information. In contrast, we will denote the traditional relevant information with relevant *sensory* information.

Whereas the relevant sensory information determines the minimum amount of sensory information necessary for a certain goal, we can also determine the minimum goal information necessary on average to achieve a certain utility, given the current state. By analogy to (5), this is done by solving the following minimisation problem:

$$\min_{\pi(a_t|s_t,g)} \Big[I(G; A_t|S_t) - \beta \mathbb{E}[U(S_t, G, A_t)|\pi] \Big] \qquad (6)$$

The solution to this problem, which is a policy trading off goal information parsimony with utility, controlled by the trade-off parameter β, can be found using the same iterative procedure used for relevant sensory information as described in (Polani et al., 2006).

As an example we use a navigation task in the grid world shown in Fig. 3(a). The set of states $\mathcal{S}$ and the set of goals $\mathcal{G}$ both consist of all unoccupied cells, and the goal variable G is assumed to be uniformly distributed; any of the goals is as likely as another. The agent is rewarded when it achieves the current goal ($\mathcal{R}^{a_t}_{s_t,s_{t+1}} = 1$ if $s_{t+1} = g$, 0 otherwise) and a discount factor of $\gamma = 0.9$ is used.

As with relevant sensory information, we can study the trade-off between utility and relevant goal information by varying the value of β in (6). Figure 4 shows that the results

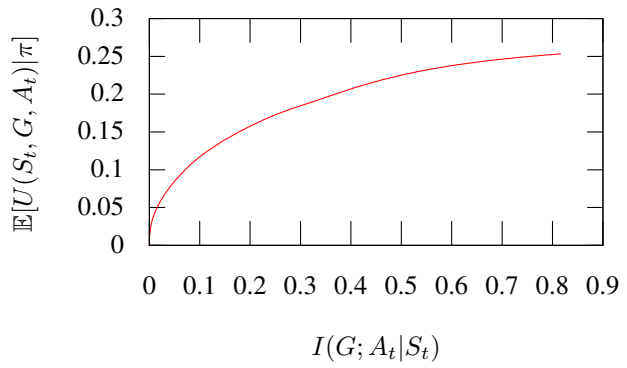


Figure 4: Trade-off between goal information (horizontal axis, bits) and expected utility (vertical axis).

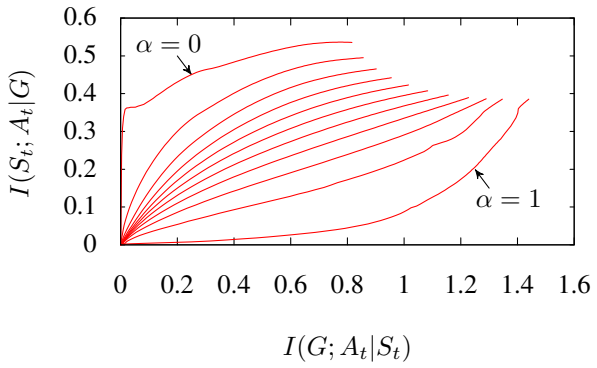


Figure 5: Trade-off between goal information (horizontal axis, bits) and sensory information (vertical axis, bits) for different values of $\alpha \in [0, 1]$, which controls preference for goal (low α) or sensory (high α) information parsimony.

of this trade-off are similar to that found for relevant sensory information; expected utility rises monotonically with higher goal information bandwidth, but the agent can still achieve a performance close to 90% of the maximum with as little as half of the optimal amount of information.

Besides utility, goal information may also have to be traded off against sensory information; a policy that minimises relevant goal information could require a higher average bandwidth for the sensors. We can combine equations (5) and (6) to take into account both costs:

$$\min_{\pi(a_t|s_t,g)} \Big[(1-\alpha)I(G; A_t|S_t) + \alpha I(S_t; A_t|G) - \beta\mathbb{E}[U(S_t, G, A_t)|\pi]\Big], \quad (7)$$

where α can be varied from 0 to 1 to reflect the relative cost of each process; low α promotes goal information parsimony, high α indicates sensor information is deemed to be more costly. Figure 5 shows that generally more relevant goal information is linked to an increase in sensory information, but that different weights result in different trade-offs.

We can extract the relevant goal information for each state separately, $I(G; A_t|s_t)$, as is shown in Fig. 3(a) for the policy achieving maximum expected utility. This example shows some interesting properties of relevant goal information. Firstly, in central states the agent tends to require more goal information than in more remote states or states close to walls. This is easily explained by the fact that in the central states the a priory probability of the direction the goal is in is roughly uniformly distributed; the goal can be on any side. When in the more distant states, however, the goal tends to be in a single direction. Only in exceptional cases does the agent need to deviate from going in this default direction and thus use extra goal information. Directly next to the walls the agent even only has to choose from the limited set of actions that do not make it run into a wall. Here the relevant goal information is bounded from above by the cardinality of this limited set. This also explains why the amount of relevant information in doorways is found to be often lower than in neighbouring states; here only two actions are useful.

Another observation is that local peaks in relevant goal information, marked with an asterisk in Fig. 3(a), can be found in front of doorways, even several cells away, most notably at 'crossing points' between different doorways. Trajectories of the agent tend to go from one of these peak cells to another. We will give an interpretation and explanation for this effect in the global discussion at the end of this paper.

Goal Information Transitions

In the example of the previous section we have only looked at single step scenarios. It shows that in different states the amount of goal information needed can vary. An interesting question is whether there is also a qualitative difference between the relevant goal information in different states. For instance, a bee flying out to search for food at first only has to consider which patch in its habitat is its target. Only when arrived at this patch it has to take into account the several individual resources (Bell, 1990). As another example, in our grid world, when the agent is in front of a doorway, it has to take into account whether the goal is in the neighbouring room or not. However, when it has just entered the room, this information is no longer relevant and it now has to focus on where exactly in the room the goal is. The model of relevant goal information given here can be used to analyse this development of goal information through time.

Given the single-step goal-information parsimonious policy as found in the previous section, we can determine how much of the relevant goal information in a certain state was not needed during the sequence leading to that state:

$$I(G; A_t|\mathbf{A}_0^{t-1}, s_t) = H(G|\mathbf{A}_0^{t-1}, s_t) - H(G|\mathbf{A}_0^{t}, s_t), \quad (8)$$

where $\mathbf{A}_0^t = (A_0, \ldots, A_t)$ denotes the sequence of actions from the start of the task to time step t. This amount of new relevant goal information is shown for our grid world case in Fig. 3(b), averaged over sequences of up to 5 time steps.

As one would expect, some of the cells where the total amount of relevant goal information is high (those marked in Fig. 3(a)) also stand out here; if in a cell more goal information is required than in the neighbouring cells, naturally a relatively high amount of this information is new. However, there are some notable differences: although the states where much new goal information is needed also require much total goal information, the opposite argument does not hold.

For instance, the cells marked a and b in Fig. 3(b) are shaded darkest in Fig. 3(a) and so require the most amount of information, with only a small difference between them. But there is a clear difference in how much of this information is new and different from the goal information that on average is required in the past before arriving in these cells. At cell b, in front of the doorway, the qualitative transition in goal information is much more pronounced. This same difference can be seen in the cells marked c and d; again, the total amount of relevant information for these cells is approximately the same, but for cell c more of this information is the same as already maintained by the agent in previous steps, showing a much less defined transition. All in all, we can note that the largest transitions are at doorways and at corners.

Discussion

Two Viewpoints

The result of minimisation of goal information is a policy where the agent often takes the same action, regardless of the goal; e.g. if going north works for all goals and going east only for a part of them the agent can always select going north and it can disregard all goal information. This leads to two complementary viewpoints for relevant goal information.

One is what we call the *least-commitment* (in the sense of least-commitment planning (Weld, 1994)) viewpoint. Because the actions taken by the agent are optimal for as many goals as possible, the amount of goals excluded by the actions are minimal. Although, in the methods described here, the goal does not change during a single run, because of the least-commitment property of the agent's policy, the agent will have a higher probability of still having behaved optimal if such a change does happen. The policy of the agent can be seen as keeping as many options open as possible. Thus, minimisation of relevant goal information causes the emergence of a least-commitment strategy.

This shows the relatedness of relevant goal information to *empowerment* (Klyubin et al., 2008). This quantity defines the maximum amount of possible observable control an agent has on its environment and is based on the same kind of informational treatment of the PA-loop as put forward in this paper. In a task-less setting empowerment leads to an intrinsic drive to least-commitment behaviour, whereas relevant goal information gives rise to such a drive in a goal-oriented agent.

The least-commitment viewpoint leads to the interpretation of states where relevant goal information is high as necessary decision points. If the goal can be in either of two rooms, the agent will not move towards one or the other until it has no other option. This occurs at the crossing points between doorways, where the agent has to make a decision and commit to one of the rooms.

Such an approach to delay decision making may not always be optimal, such as a driver who risks an accident by steering for a corner at the last moment at high speed. However, here these risks are assumed to be contained in the reward function, rendering such policies suboptimal and thus no longer considered by the agent.

Another interpretation arises from the *goal-concealing* viewpoint. This viewpoint is obtained by noting that the mutual information between goal and action can not only be seen as how much goal information is needed to decide on an action, or how much information the goal gives about the action, but also how much information the actions give about the goal (a similar viewpoint for sensory relevant information is taken by Salge and Polani (2010)). This means that by minimising relevant goal information the agent gives away as little information as possible about its goal to an external observer. This observer could see this as the emergence of a goal-hiding strategy.

From this viewpoint the peaks in relevant goal information at crossing points can be explained by noting that the actions taken here give away a lot of information about the goal of the agent. When the agent is at a crossing point between two rooms, the observer does not know in which room the goal is, but after seeing the action he can exclude all the cells in the room the agent moved away from.

Sub-Goals

In the field of Reinforcement Learning (RL) there has been a lot of recent activity on the subject of higher level behaviour structuring, task decomposition and automatic sub-goal discovery (Barto and Mahadevan, 2003). A large amount of algorithms for automatic behaviour structuring have resulted from this. For instance, the intuition that so called 'bottleneck' or 'funnel' states in an environment, such as doorways, are salient sub-goals has led to methods being developed based on visitation count (McGovern and Barto, 2001; Kretchmar et al., 2003; Asadi and Huber, 2005) and graph-theoretical techniques (Şimşek et al., 2005; Kazemitabar and Beigy, 2009; Şimşek and Barto, 2009). Other approaches that are also based on assumptions about the structure of the world, but using less strict definitions of what may constitute a 'good' sub-goal, include state space segmen-

tation/clustering (Bakker and Schmidhuber, 2004; Mannor et al., 2004), relative novelty (Şimşek and Barto, 2004), sensation/action co-occurrence (Digney, 1996) or transitions (Hengst, 2002; Kozlova et al., 2009), causal-graph decomposition (Jonsson and Barto, 2006) and the use of data-mining techniques (Kheradmandian and Rahmati, 2009). Finally, a separate class of algorithms does not focus on structure of goals, but on segmentation, clustering and abstracting common state-action sequences (Sun and Sessions, 2000; Pickett and Barto, 2002; Girgin et al., 2006).

All these methods indicate their usefulness by showing increased learning performance in certain RL tasks. Also, they show that skill transfer, made possible by task segmentation, can be highly beneficial (Perkins and Precup, 1999; Konidaris and Barto, 2007). However, hardly any comparison of the performance of different approaches has yet been done. This is not surprising, since the methods can differ greatly and, more importantly, they are based on different, designer imposed, assumptions about what is a good way to structure a task. In these papers the structural properties of a sub-goal or sub-task are defined for a particular domain of interest, after which a solution is engineered for these specific properties.

The results of the current paper, however, suggest a more fundamental, biologically/Alife motivated definition of sub-goals: a sub-goal is achieved when a significant qualitative change of the task at hand occurs, which is when the actions of an agent are guided by a new component of, or new information about, the goal not taken into account earlier. As shown earlier, the notion of relevant goal information can be used to identify such transitions. Note that the informational treatment of the PA-loop is independent of domain, architecture and particular implementations and therefore we do not need any of the assumptions made in the engineering solutions. The biologically plausible hypothesis of information parsimony is sufficient for the treatment of emergence of sub-goals.

References

Asadi, M. and Huber, M. (2005). Accelerating Action Dependent Hierarchical Reinforcement Learning Through Autonomous Subgoal Discovery. In *Proceedings of the ICML 2005 Workshop on Rich Representations for Reinforcement Learning*.

Bakker, B. and Schmidhuber, J. (2004). Hierarchical Reinforcement Learning Based on Subgoal Discovery and Subpolicy Specialization. In *Proceedings of the 8-th Conference on Intelligent Autonomous Systems, IAS-8*, pages 438–445.

Barlow, H. B. (1961). Possible Principles Underlying the Transformations of Sensory Messages. In Rosenblith, W., editor, *Sensory Communication*, chapter 13, pages 217–234. MIT Press, Cambridge, MA.

Barto, A. G. and Mahadevan, S. (2003). Recent Advances in Hierarchical Reinforcement Learning. *Discrete Event Dynamic Systems*, 13(1-2):41–77.

Bell, W. J. (1990). Searching Behavior Patterns in Insects. *Annual Review of Entomology*, 35:447–467.

Blahut, R. E. (1972). Computation of Channel Capacity and Rate-Distortion Functions. *IEEE Transactions on Information Theory*, 18:460–473.

Brenner, N., Bialek, W., and de Ruyter van Steveninck, R. (2000). Adaptive Rescaling Maximizes Information Transmission. *Neuron*, 26(3):695–702.

Cover, T. M. and Thomas, J. A. (1991). *Elements of information theory*. Wiley-Interscience, New York, NY, USA.

Digney, B. (1996). Emergent Hierarchical Control Structures: Learning Reactive / Hierarchical Relationships in Reinforcement Environments. In *Proceedings of the Fourth Conference on the Simulation of Adaptive Behavior: SAB 98*, pages 363–372. MIT Press.

Girgin, S., Polat, F., and Alhajj, R. (2006). Learning by Automatic Option Discovery from Conditionally Terminating Sequences. In *ECAI*, pages 494–498.

Hamilton, A. F. d. C. and Grafton, S. T. (2006). Goal Representation in Human Anterior Intraparietal Sulcus. *Journal of Neuroscience*, 26(4):1133–1137.

Hengst, B. (2002). Discovering Hierarchy in Reinforcement Learning with HEXQ. In *ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning*, pages 243–250, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Hommel, B., Müsseler, J., Aschersleben, G., and Prinz, W. (2001). The Theory of Event Coding (TEC): A framework for perception and action planning. *Behavioral and Brain Sciences*, 24(05):849–878.

Jeffery, W. (2001). Cavefish as a Model System in Evolutionary Developmental Biology. *Developmental Biology*, 231(1):1–12.

Jonsson, A. and Barto, A. (2006). Causal Graph Based Decomposition of Factored MDPs. *Journal Of Machine Learning Research*, 7:2259–2301.

Kazemitabar, S. J. and Beigy, H. (2009). Automatic Discovery of Subgoals in Reinforcement Learning Using Strongly Connected Components. *Advances in Neuro-Information Processing*, 5506:829–834.

Kheradmandian, G. and Rahmati, M. (2009). Automatic Abstraction in Reinforcement Learning Using Data Mining Techniques. *Robotics and Autonomous Systems*, 57(11):1119–1128.

Klyubin, A. S., Polani, D., and Nehaniv, C. L. (2004). Tracking Information Flow through the Environment: Simple Cases of Stigmergy. In Pollack, J., Bedau, M., Husbands, P., Ikegami, T., and Watson, R. A., editors, *Artificial Life IX: Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems*, pages 563–568. The MIT Press.

Klyubin, A. S., Polani, D., and Nehaniv, C. L. (2008). Keep Your Options Open: An Information-Based Driving Principle for Sensorimotor Systems. *PLoS ONE*, 3(12):e4018.

Konidaris, G. and Barto, A. (2007). Building Portable Options: Skill Transfer in Reinforcement Learning. In *IJCAI'07: Proceedings of the 20th International Joint Conference on Artifical intelligence*, pages 895–900, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Kozlova, O., Sigaud, O., and Meyer, C. (2009). Automated Discovery of Options in Factored Reinforcement Learning. In *Proceedings of the ICML/UAI/COLT Workshop on Abstraction in Reinforcement Learning*, pages 24–29, Montreal, Canada.

Kretchmar, R., Feil, T., and Bansal, R. (2003). Improved Automatic Discovery of Subgoals for Options in Hierarchical Reinforcement Learning. *Journal of Computer Science & Technology*, 3:9–14.

Laughlin, S. B., de Ruyter van Steveninck, R. R., and Anderson, J. C. (1998). The Metabolic Cost of Neural Information. *Nature Neuroscience*, 1:36–41.

Mannor, S., Menache, I., Hoze, A., and Klein, U. (2004). Dynamic Abstraction in Reinforcement Learning via Clustering. In *ICML '04: Proceedings of the twenty-first International Conference on Machine Learning*, page 71, New York, NY, USA. ACM.

McGovern, A. and Barto, A. G. (2001). Automatic Discovery of Subgoals in Reinforcement Learning using Diverse Density. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 361–368, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Montague, P. R., Hyman, S. E., and Cohen, J. D. (2004). Computational Roles for Dopamine in Behavioural Control. *Nature*, 431(7010):760–7.

Nehaniv, C. L., Polani, D., Olsson, L., and Klyubin, A. S. (2007). Information-Theoretic Modeling of Sensory Ecology: Channels of Organism-Specific Meaningful Information. In Laubichler, M. D. and Müller, G. B., editors, *Modeling Biology: Structures, Behaviors, Evolution (The Vienna Series in Theoretical Biology)*, pages 241–282. MIT Press.

Perkins, T. and Precup, D. (1999). Using Options for Knowledge Transfer in Reinforcement Learning. Technical report, University of Massachusetts, Amherst, MA, USA. UM-CS-1999-034.

Pfeifer, R., Lungarella, M., Sporns, O., and Kuniyoshi, Y. (2007). On the Information-Theoretic Implications of Embodiment – Principles and Methods. In *Proc. of the 50th Anniversary Summit of Artificial Intelligence*, volume 4850, pages 76–86. Springer-Verlag.

Pickett, M. and Barto, A. G. (2002). PolicyBlocks: An Algorithm for Creating Useful Macro-Actions in Reinforcement Learning. In *ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning*, pages 506–513, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Polani, D. (2009). Information: Currency of Life? *HFSP Journal*, 3:307–316.

Polani, D., Nehaniv, C. L., Martinetz, T., and Kim, J. T. (2006). Relevant Information in Optimized Persistence vs. Progeny Strategies. In *Artificial Life X: Proceedings of The 10th International Conference on the Simulation and Synthesis of Living Systems, Bloomington IN*.

Saito, N., Mushiake, H., Sakamoto, K., Itoyama, Y., and Tanji, J. (2005). Representation of Immediate and Final Behavioral Goals in the Monkey Prefrontal Cortex During an Instructed Delay Period. *Cerebral Cortex*, 15(10):1535–46.

Salge, C. and Polani, D. (2010). Digested Information as an Information Theoretic Motivation for Social Interaction. *Journal of Artificial Societies and Social Simulation (JASSS)*. accepted.

Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, 27:379–423 and 623–656.

Şimşek, Ö. and Barto, A. G. (2004). Using Relative Novelty to Identify Useful Temporal Abstractions in Reinforcement Learning. In *ICML '04: Proceedings of the Twenty-First International Conference on Machine Learning*, page 95, New York, NY, USA. ACM.

Şimşek, Ö. and Barto, A. G. (2009). Skill Characterization Based on Betweenness. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 21*, pages 1497–1504.

Şimşek, Ö., Wolfe, A. P., and Barto, A. G. (2005). Identifying Useful Subgoals in Reinforcement Learning by Local Graph Partitioning. In *Proceedings of the Twenty-Second International Conference on Machine Learning*, pages 816–823.

Spiers, H. J. and Maguire, E. A. (2006). Thoughts, Behaviour, and Brain Dynamics During Navigation in the Real World. *Neuroimage*, 31(4):1826–40.

Sporns, O. and Lungarella, M. (2006). Evolving Coordinated Behavior by Maximizing Information Structure. In *Artificial Life X*, pages 323–329. MIT Press.

Sun, R. and Sessions, C. (2000). Self-Segmentation of Sequences: Automatic Formation of Hierarchies of Sequential Behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 30(3):403–418.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press.

Sutton, R. S., Precup, D., and Singh, S. (1999). Between MDPs and Semi-MDPs: a Framework for Temporal Abstraction in Reinforcement Learning. *Artificial Intelligence*, 112(1-2):181–211.

van Dijk, S. G., Polani, D., and Nehaniv, C. L. (2009). Hierarchical Behaviours: Getting the Most Bang for your Bit. In *Proceedings of the 10th European Conference on Artificial Life*. Springer (In Press).

Vergassola, M., Villermaux, E., and Shraiman, B. I. (2007). 'Infotaxis' as a Strategy for Searching Without Gradients. *Nature*, 445(7126):406–9.

Weld, D. S. (1994). An Introduction to Least Commitment Planning. *AI Magazine*, 15(4):27–61.

Engineered Microbial Communication for Population-Level Behaviour

Angel Goñi Moreno[1] and Martyn Amos[2]

[1]Natural Computing Group, Faculty of Computer Engineering, Universidad Politécnica de Madrid, Spain.
[2]Department of Computing and Mathematics, Manchester Metropolitan University, United Kingdom.
agmoreno@gcn.upm.es

Extended Abstract

Quorum-sensing (QS) has been extensively studied in the context of synthetic biology (Basu *et al*., 2005; Danino *et al.*, 2010; Garcia-Ojalvo *et al*., 2004). It enables a community-level response to emerge once a certain signal *concentration threshold* has been reached. We use QS to design a *multi-strain*, engineered bacterial community with autonomous behaviour. We model our system on the familiar "client-server" architecture, with a single central server and two clients (one "red" and the other "green"). The task we define is that of *oscillation* (Tigges *et al.,* 2009); by engineering feedback between three different strains, we obtain indefinite switching between "red" and "green" outputs. The system is not restricted to simple oscillation, as server cells may be introduced with much more complex behaviours.

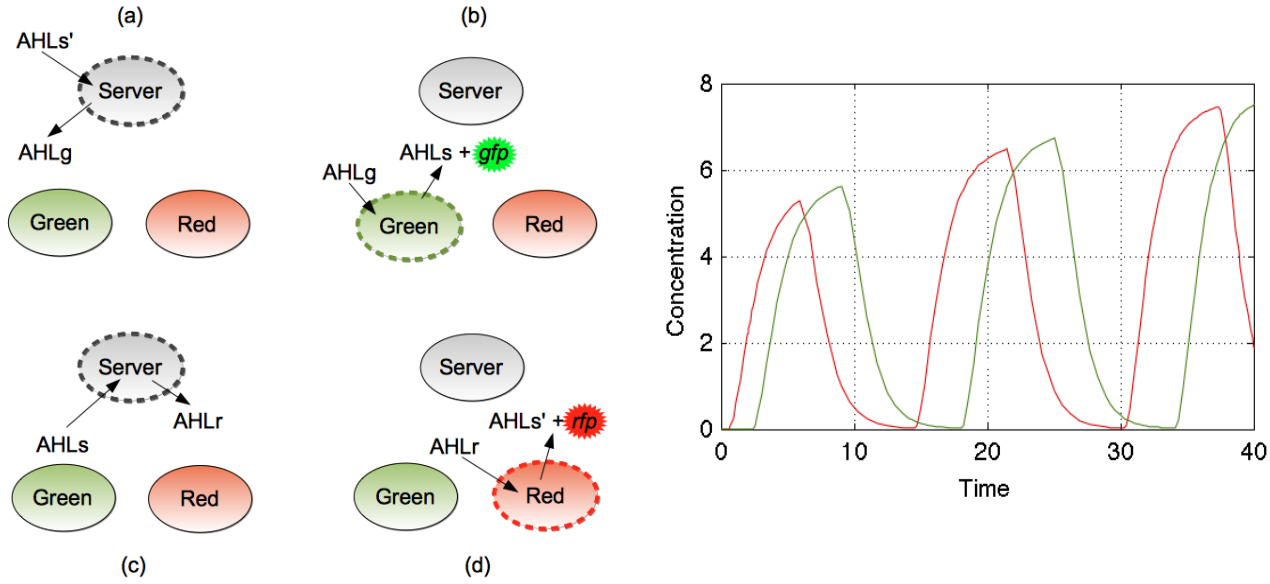


Figure 1: System architecture (left), simulation results (right).

In Figure 1, we show the server and two clients; the *server* is activated by selected signalling molecules, labelled AHLs and AHLs', (producing either AHLr or AHLg respectively); the *green* client is *activated* by AHLg, producing AHLs and green fluorescent protein, and the *red* client is activated by AHLr, producing AHLs' and red fluorescent protein. We can see how this machine lies dormant until either AHLg or AHLr is added to the nutrient, after which one of the clients is activated and the system enters a period of oscillation. This is achieved by the server cells switching “turns” between red and green client cells. We also see the results of system simulations, with plots of AHLs' and AHLs over time.

Our key contribution is the design of the server, which is extremely noise-resistant, and robust in the face of differential client behaviour (e.g., if one client's “off” signal degrades much more slowly than another's). Future work will focus on experimental testing of the system, and investigation of its real-world applicability.

References

Basu, S., Gerchman, Y., Collins, C. H., Arnold, F. H., and Weiss, R. (2005). A synthetic multicellular system for programmed pattern formation. *Nature*, 434(7037), 1130-4.

Danino, T., Mondragón-Palomino, O., Tsimring, L., and Hasty, J. (2010). A synchronized quorum of genetic clocks. *Nature*, 463(7279), 326-30.

Garcia-Ojalvo, J., Elowitz, M. B., and Strogatz, S. H. (2004). Modeling a synthetic multicellular clock: Repressilators coupled by quorum sensing. *Proceedings of the National Academy of Sciences of the United States of America*, 101(30), 10955-60.

Tigges, M., Marquez-Lago, T. T., Stelling, J., and Fussenegger, M. (2009). A tunable synthetic mammalian oscillator. *Nature*, 457(7227), 309-12.

Design Principles of Transcriptional Logic Circuits

Nicolae Radu Zabet[1], Andrew N. W. Hone[2] and Dominique F. Chu[1]

[1]School of Computing, University of Kent, CT2 7NF, Canterbury, UK
[2]School of Mathematics, Statistics and Actuarial Science, University of Kent, CT2 7NF, Canterbury, UK
N.R.Zabet@kent.ac.uk

Abstract

Using a set of genetic logic gates (AND, OR and XOR), we constructed a binary full-adder. The optimality analysis of the full-adder showed that, based on the position of the regulation threshold, the system displays different optimal configurations for speed and accuracy under fixed metabolic cost. In addition, the analysis identified an optimal trade-off curve bounded by these two optimal configurations. Any configuration outside this optimal trade-off curve is sub-optimal in both speed and accuracy. This type of analysis represents a useful tool for synthetic biologists to engineer faster, more accurate and cheaper genes.

Introduction

The desire to control is a recurring theme of human nature and the control of biological systems represents the ultimate goal for synthetic biologists. Towards achieving this goal, researchers have modelled and engineered genes in bacterial cells that perform basic computational tasks. These tasks mainly mimic the behaviour of simple electronic components, such as logic gates, oscillators, toggle switches and counters (Gardner et al., 2000; Elowitz and Leibler, 2000; Guet et al., 2002). However, when attempting to increase the complexity of these engineered genetic systems, certain limitations of the components are likely to hamper their construction. Thus, there is an urgent need for an extensive analysis of the biophysical limits of the elementary components.

Synthetic biologists showed that binary logic gates can be engineered in living cells using transcriptional logic (Guet et al., 2002; Kramer et al., 2004; Yokobayashi et al., 2002; Cox III et al., 2007; Anderson et al., 2007; Sayut et al., 2009). Transcriptional logic gates are genes which can integrate multiple signals at the level of cis-regulatory transcription control using various binary logic functions (AND, OR, NAND, NOR, XOR, etc.). To implement binary logic, both the input and the output of these genes needs to have two abundance levels corresponding to the two logical levels, a high and a low abundance level. Biological modellers successfully identified and described various designs of these logic gates (Weiss et al., 2003; Buchler et al., 2003; Hermsen et al., 2006; Schilstra and Nehaniv, 2008; Silva-Rocha and deLorenzo, 2008). However, what is still missing is a complete analysis of how these logic gates can be used as building blocks for more complex logical systems and what are the parameters which ensure optimal design in terms of speed and accuracy under limited (constant) energetic resources.

There are three properties of a genetic system that we use in our analysis: speed, accuracy and cost. We define the *propagation time* as the time required by the output species in a logical system to reach the new steady state after an instantaneous change of the inputs. This is directly connected with *speed* in the sense that fast system are described by short propagation times and conversely. Due to low copy number and slow chemical reactions, genetic systems are stochastic and, thus, they are affected by *noise* (Kaern et al., 2005). The noise reduces the ability to distinguish between different logical outputs of a gate and, because of that, it reduces *accuracy*. Finally, the *metabolic cost* is usually measured as the required number of ATP molecules. We are interested in the scaling properties of this measure, rather than in the exact value. Hence, we measure cost as the maximum synthesis rate of a gene.

Recently we investigated speed and accuracy in the case of single binary genes (genes with two expression levels, high and low) (Zabet and Chu, 2010). The analysis revealed that these genes display a trade-off curve between switching time and noise under fix metabolic cost, i.e., lower noise is achieved at lower speeds and conversely. This trade-off is controlled by the decay rate, in the sense that higher decay rate means higher speed but also lower accuracy.

In this contribution, we extend this analysis to gene networks by considering a specific binary logic system, the full-adder. The full-adder is a system able to perform binary addition (to produce both the sum and the carry) for three binary inputs, two of which are the two operands and the third allows plugging in the carry from a previous full-adder module. We constructed the required logic gates by considering genes that can be regulated by two proteins in an independent fashion, i.e., binding of any of the inputs does not alter

the binding of the other input. Moreover, these logic gates need to ensure *interconnectivity*. Assuming that the two inputs that regulate a gene can have two possible abundance levels, high (H_{in}) and low (L_{in}), then, in order to connect an *arbitrary* number of logic gates, the output has to have two possible abundance levels (H_{out} and L_{out}) with at least the same signal strength, $(H_{in} - L_{in}) \leq (H_{out} - L_{out})$ (Magnasco, 1997). Usually the output levels are identical with the input one or very close to them, $H_{out} \geq H_{in}$ and $L_{out} \leq L_{in}$. Based on these requirements, we found the set of parameters which ensures interconnectivity of the required logic gates and then we constructed the full-adder showing the correct functioning of the system.

Gene regulation is usually modelled by a Hill function (Ackers et al., 1982; Bintu et al., 2005; Chu et al., 2009). The Hill function is a sigmoid function described by two parameters: the threshold K (which represents the input abundance required for half activation of the gene) and the Hill coefficient l (which determines the steepness of the function). The results show that, for step-like regulation functions ($l \to \infty$), the system displays an optimal position of the threshold in terms of speed and accuracy, while, for finite Hill coefficients, there is a trade-off between these two properties and the trade-off is controlled by the position of the threshold.

Model

We selected a design for the full-adder with five logic gates: two XOR gates, two AND gates, and one OR gate (see Fig. 1).

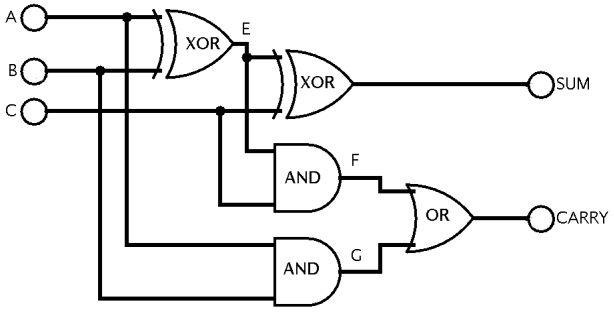


Figure 1: *Full-adder*. The logic gate diagram of the full adder.

To construct this full-adder from genes, we need first to construct transcriptional logic gates. We model a transcriptional logic gate as a gene G_z, which synthesises protein z, the output of the gate. This gene is regulated by two proteins x and y, which are considered as the inputs of gate. Species z is described by the following deterministic differential equation

$$\frac{dz}{dt} = \alpha + \beta f(x, y) - \mu z \qquad (1)$$

where α is the basal synthesis rate, $\alpha + \beta$ the maximum synthesis rate, $f(x, y)$ is the regulation function of gene G_z, and μ is the decay rate.

Although there are many scenarios for promoter regulation that mimic the behaviour of different logic gates, we selected independent binding (binding of one TF does not influence in any way the binding of the other TF). In this scenario there are two operator sites O_x and O_y, each of them having l binding sites. On each operator site only molecules of a specific transcription factor can bind, and they do this in a homo-cooperative maner. The probabilities that an operator site is full is described by a Hill function (Ackers et al., 1982; Bintu et al., 2005; Chu et al., 2009)

$$p_x(x) = \frac{x^l}{x^l + K^l}, \quad p_y(y) = \frac{y^l}{y^l + K^l} \qquad (2)$$

where K is the regulation threshold (the required input value for half activation of the gene) and l is the Hill coefficient (indicates steepness of the function). We assumed that the two operator sites (O_x and O_y) have identical parameters (K and l).

Assuming that the gene is turned on when any of the two TF are present, then the regulation function will mimic the behaviour of an OR gate. Analogously, assuming that a gene can be turned on only when both of the transcription factors are present, then the regulation function will mimic the behaviour of an AND gate. Finally, if the gene is turned on when any of the TF is present, but when both of them are present their effects cancels out and the gene is turned off, then the gene will behave as an XOR gate. The corresponding forms of the regulation functions are

$$\begin{aligned} f_{AND} &= \frac{(xy)^l}{(xy)^l + (Kx)^l + (Ky)^l + K^{2l}}, \\ f_{OR} &= \frac{(xy)^l + (xK)^l + (yK)^l}{(xy)^l + (Kx)^l + (Ky)^l + K^{2l}}, \\ f_{XOR} &= \frac{(Kx)^l + (Ky)^l}{(xy)^l + (Kx)^l + (Ky)^l + K^{2l}}. \end{aligned} \qquad (3)$$

Fig. 2 confirms that these regulation functions display the desired behaviour.

Using these three logic gates, the full-adder, can be constructed as a set of chemical reactions. Since the full-adder contains five logic gates, then we need five species to implement this system (e, f, g, sum and carry). The chemical reactions which describe all these species are given by

$$\emptyset \underset{\mu_e}{\overset{\alpha_e + \beta_e f_{XOR}(a,b)}{\rightleftharpoons}} e, \quad \emptyset \underset{\mu_f}{\overset{\alpha_f + \beta_f f_{AND}(c,e)}{\rightleftharpoons}} f,$$

$$\emptyset \underset{\mu_g}{\overset{\alpha_g + \beta_g f_{AND}(a,b)}{\rightleftharpoons}} g,$$

$$\emptyset \underset{\mu_s}{\overset{\alpha_s + \beta_s f_{XOR}(e,c)}{\rightleftharpoons}} \text{sum}, \quad \emptyset \underset{\mu_{co}}{\overset{\alpha_{co} + \beta_{co} f_{OR}(f,g)}{\rightleftharpoons}} \text{carry}$$

where a, b and c are three input species.

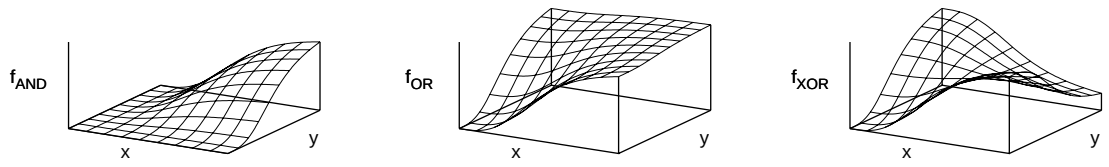


Figure 2: *Regulation functions that mimic logic gate behaviour*. The threshold was set to $K = 0.5\ [\mu M]$ and we considered a Hill coefficient of $h = 3$.

Results

First we need to identify the sets of parameters which allow interconnection of gates and then we need to identify the sub-set of parameters which allows optimal functioning of the full-adder in terms of speed and accuracy under fixed metabolic cost. We will apply these two analyses for two cases: (i) step-like regulation functions ($l \to \infty$) and (ii) finite Hill coefficients.

To keep the mathematics tractable, and without losing too much generality, we consider identical gates, i.e., all genes are affected by the same decay rate (μ), have the same synthesis rates (α and β) and the same Hill parameters (l and K). The only thing that differentiates the gates is the regulation function, which, in the case case of the full-adder, can be f_{AND}, f_{OR} or f_{XOR}.

Step Regulation Functions

We start our analysis by considering the ideal case, the system where the regulation functions have infinite Hill coefficient.

The interconnectivity property can be met by considering the output signal strength to be kept constant, $H_{out} = H_{in} = H$ and $L_{out} = L_{in} = L$. In the case of the OR gate, the system has the following steady state behaviour

$$\begin{aligned} L &= \frac{1}{\mu}\left[\alpha + \beta f_{OR}(L, L)\right], \\ H &= \frac{1}{\mu}\left[\alpha + \beta f_{OR}(L, H)\right], \\ H &= \frac{1}{\mu}\left[\alpha + \beta f_{OR}(H, H)\right]. \end{aligned} \tag{4}$$

For infinite Hill coefficient the solution is given by $\alpha = L$ and $\beta = (H - L)$. Analogously, it can be shown that the solution is the same for all gates. This synthesis rates ensure a correct steady-state behaviour of the full-adder (see Fig. 3(a)).

System Performance We investigate two properties of a logic system, namely speed and accuracy, under the constraint of fix metabolic cost. The metabolic cost of a gene Z can be defined as the maximum synthesis rate of that gene, $\zeta_z = \alpha + \beta f_z^H$, where f_z^H is the highest value which $f(x, y)$ takes. Thus, by keeping the synthesis rate fixed the metabolic cost is kept constant. Note that this is just an approximation to the actual metabolic cost, and that the metabolic cost of the maintenance of the entire machinery was not included in it. However, this measure indicates how the metabolic costs scales with different parameters.

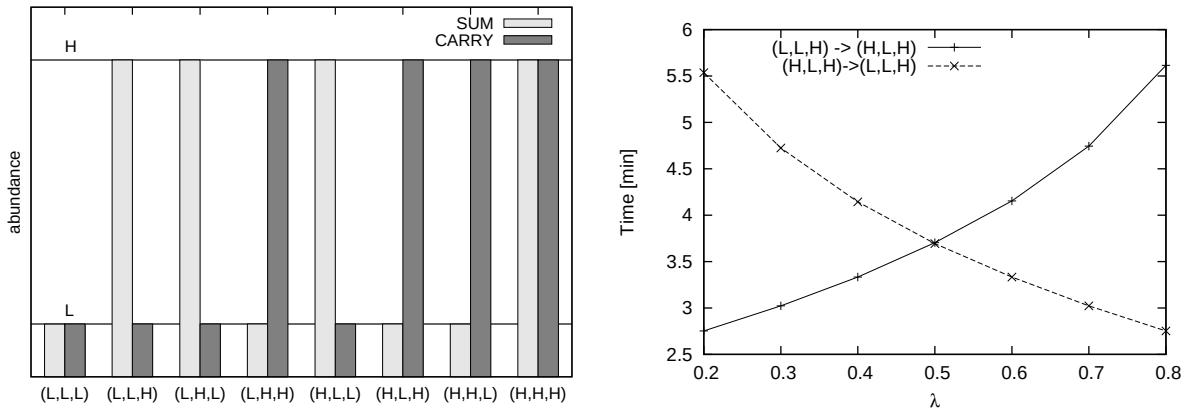
The *propagation time*, T_{gene}, of a gene is the time required to reach the steady state to within a fraction θ of $H - L$. Assuming instant change of the input, Eq. (1) can be solved analytically and the time to reach $L + (H - L)\theta$ or $H - (L - H)\theta$ can be computed as

$$T_i = \tau \cdot ln\left(\frac{1}{1-\theta}\right) \tag{5}$$

where $\tau = 1/\mu$ represents the average life time of the species.

The propagation time through a single gate can only be reduced by reducing the average life time of the protein (τ). In the case when the two logical steady states are kept constant (so the signal strength is not reduced) and the synthesis rate is kept constant (so we do not increase the metabolic cost) then also the decay rate is kept constant. Thus, there is no optimization that one could attempt to perform on individual gates under fix metabolic cost without reducing signal strength. However in the case of logic gates systems, like the case of the full-adder, the input is not changed instantaneously in all gates and the position of the threshold influences the propagation time.

The threshold is located between the low and the high state, $K = L + (H - L)\lambda$, ($\lambda \in [0, 1]$). λ indicates the position of the threshold; for $\lambda < 0.5$, K is closer to L and for $\lambda > 0.5$, K is closer to H. Note that by considering K to be outside the interval $[L, H]$ the regulation is removed, i.e., the gene is always in the same state no matter whether the input is L or H. In order for a gene to change state, one of the inputs, has to cross over or under K. Using Eq. (5) one can compute the time it takes one species to move from low state to the threshold ($L \to K$) and from the high state



(a) steady state behaviour (b) optimum K for speed

Figure 3: *Full-adder with step-like regulation function.* (a) The output abundance based on the input abundance for step-like regulation functions. (b) We plotted the propagation time when switching between (L, L, H) to (H, L, H). The following set of parameters have been used: $\mu = 1\ min^{-1}$, $l = 50$, $L = 0.2\ \mu M$, $H = 1.2\ \mu M$, $K = 0.7\ \mu M$, $\alpha = 0.2\ \mu M \cdot min^{-1}$, $\beta = 1.0\ \mu M \cdot min^{-1}$ and $\theta = 0.9$.

to the threshold ($H \rightarrow K$) as

$$t_{LK} = \tau \cdot ln\left(\frac{1}{1-\lambda}\right), \quad t_{HK} = \tau \cdot ln\left(\frac{1}{\lambda}\right). \quad (6)$$

Assuming that the longest cascade in the system has n gates, then a general formula for the propagation time is given by

$$T = \sum_{i=1}^{n-1} t_{iK} + T_n \quad (7)$$

where t_{iK} is equal to t_{LK} if species ith was in low state before changing the input in the system, and t_{iK} is equal to t_{HK} if species ith was in high state before changing the input in the system. Hence, the propagation time in a cascade equals a sum of t_{LK} and t_{HK} terms and a fix time representing the last gene in the cascade T_n.

Fig. 4 confirms that based on the threshold position, the system can be faster when switching in one direction and slower in the opposite direction. When the switching direction is not important, the problem of optimizing propagation time becomes a minimax problem, i.e., minimize the maximum time to switch. In the context of step-like regulation functions, the optimum threshold, according to Eq. (6), resides at the midpoint between high and low states, $\lambda_T = 0.5$ (see Fig. 4).

Analysing the circuit diagram of the full-adder 1 one can notice that the longest path through the circuit consists of three gates, and this is used when computing the carry . This path is followed, for example, when switching between (L, L, H) and (H, L, H). Fig. 3(b) confirms that the optimum threshold, in the case of step-like regulation function, resides at the midpoint between high and low state

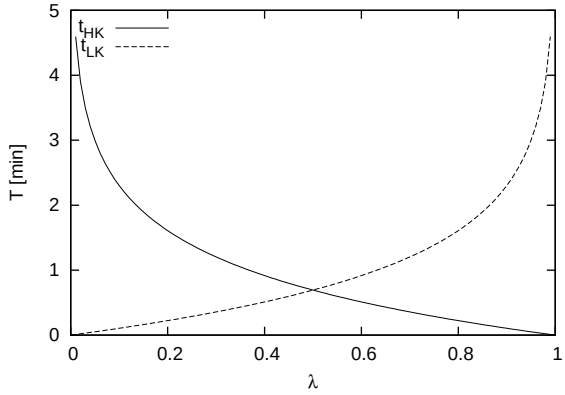


Figure 4: *The time to reach the threshold.* The protein average life time to $\tau = 1\ [min]$. The two steady states are $L = 0.2\ [\mu M]$ and $H = 0.8\ [\mu M]$, and the corresponding synthesis rates were considered. Both switching directions were consider.

($\lambda = 0.5$). Also note, that Eq. (7) and Eq. (6) correctly predict the propagation time in the full-adder in the case of high Hill coefficients.

Next, we need to investigate the accuracy of the system. At steady state the *variance* of the output z of a logic gate, which has two inputs x and y, can be written as (van Kampen, 2007; Elf and Ehrenberg, 2003; Paulsson, 2004)

$$\sigma_z^2 = \underbrace{z}_{\text{intrinsic}} + \underbrace{\overbrace{\left[\beta_z \frac{\partial f(x,y)}{\partial x}\tau_z\right]^2}^{\Gamma_{zx}} \overbrace{\frac{\tau_x}{\tau_x+\tau_z}}^{T_{zx}} \sigma_x^2}_{\text{upstream from } x} + \underbrace{\overbrace{\left[\beta_z \frac{\partial f(x,y)}{\partial y}\tau_z\right]^2}^{\Gamma_{zy}} \overbrace{\frac{\tau_y}{\tau_y+\tau_z}}^{T_{zy}} \sigma_y^2}_{\text{upstream from } y} \quad (8)$$

The intrinsic component is generated by the randomness in the birth-death processes and it can be approximated by a Poisson process (Bar-Even et al., 2006; Newman et al., 2006). The upstream component is the noise transmited from the upstream species (the species that regulate the gene) (Pedraza and van Oudenaarden, 2005). The upstream noise is composed of three terms: the regulation factor (Γ_{zx} and Γ_{zy}), the time average factor (T_{zx} and T_{zy}), and the variance of the upstream species (σ_x^2 and σ_y^2).

In this contribution, we are interested in how noise affects our ability to distinguish between the two known output states, H and L. To get a meaningful measure of this, we will normalise the variance by the square of the signal strength, $\eta_z \doteq \sigma_z^2/(H-L)^2$, rather than by the square of the mean (which is often used as a definition of noise).

$$\eta_z = \frac{z}{(H-L)^2} + \left[\beta_z\tau_z\frac{\partial f(x,y)/\partial x}{(H-L)}\right]^2 T_{zx}\sigma_x^2 + \left[\beta_z\tau_z\frac{\partial f(x,y)/\partial y}{(H-L)}\right]^2 T_{zy}\sigma_y^2 \quad (9)$$

For step-like regulation function the derivatives in (9) will be zero, and the only contribution to the noise is the intrinsic component. Thus, the noise of the output depends only on the steady state abundance (high and low), but is independent of the number of gates in the system or of the threshold position. However, if the threshold is close enough to one of the steady states (H or L), then small fluctuations in the input generates high fluctuations in the output and the analytical method is not accurate any-more. Assuming that the threshold is positioned at the midpoint (optimum position for speed) and the two steady states are far enough from each other, then the noise will be determined only by the intrinsic component. Hence, in the case of step-like regulation functions, the system displays an optimum threshold position ($\lambda = 0.5$) which ensures optimality both for speed and accuracy.

Finite Hill Coefficients

Due to the fact that Hill coefficients are bounded above by the number of regulatory binding sites (Chu et al., 2009), and genes have a small number of binding sites (Hermsen et al., 2006), biologically realistic Hill coefficients are finite and have low values.

For low Hill coefficients, Eq. (4) has only one solution, $H = L$. This is not a useful solution because it removes the binary logic. Therefore, we search for parameters which ensure that the signal strength is not reduced, $(H_{out} - L_{out}) \geq (H_{in} - L_{in})$, and this can be achieved by solving only the first two equations in Eq. (4):

$$\begin{aligned} \frac{\alpha_{OR}}{\mu} &= \frac{Lf_{OR}(L,H) - Hf_{OR}(L,L)}{[f_{OR}(L,H) - f_{OR}(L,L)]}, \\ \frac{\beta_{OR}}{\mu} &= \frac{H-L}{[f_{OR}(L,H) - f_{OR}(L,L)]}. \end{aligned} \quad (10)$$

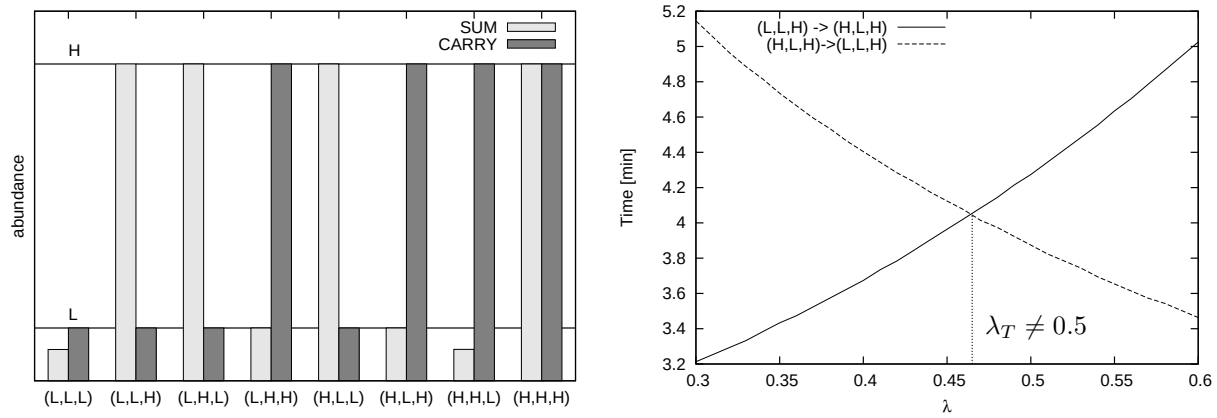
Note that not for all sets of parameters (l, K, μ, H, L) the synthesis rates will have positive values. Interestingly, increasing the Hill coefficient increases the space of allowed parameters, and in the limit case of a step function ($l \to \infty$) any values of the other parameters will generate positive synthesis rates. For Hill coefficient less than or equal to 1 there is no solution for this system. Analogously one could use the same mechanism to determine the synthesis rates for all the other gates. For AND and XOR gates the solution is given by

$$\begin{aligned} \frac{\alpha_{AND}}{\mu} &= \frac{Lf_{AND}(H,H) - Hf_{AND}(L,H)}{[f_{AND}(H,H) - f_{AND}(L,H)]} \\ \frac{\beta_{AND}}{\mu} &= \frac{H-L}{[f_{AND}(H,H) - f_{AND}(L,H)]} \end{aligned} \quad (11)$$

$$\begin{aligned} \frac{\alpha_{XOR}}{\mu} &= \frac{Lf_{XOR}(L,H) - Hf_{XOR}(H,H)}{[f_{XOR}(L,H) - f_{XOR}(H,H)]} \\ \frac{\beta_{XOR}}{\mu} &= \frac{H-L}{[f_{XOR}(L,H) - f_{XOR}(H,H)]} \end{aligned} \quad (12)$$

Fig. 5(a) confirms that the signal is not decreased and shows that in two cases the actual output low state (L_{out}) is lower than the desired one (L).

System Performance For low Hill coefficients the optimum threshold in terms of speed in not positioned any more at the midpoint between high state and low state (see Fig. 5(b)). This is a consequence of the fact that for low Hill coefficient the Hill function loses the symmetry around the threshold. Hence, when designing a specific system, one could use numerical solutions to determine the



(a) steady state

(b) optimum K for speed

Figure 5: *Full adder with low Hill coefficients*. (a) The output abundance based on the input abundance for low Hill coefficients. (b) We plotted the propagation time when switching between (L, L, H) to (H, L, H) for low Hill coefficient. The following set of parameters have been used: $\mu = 1\ min^{-1}$, $l = 6$, $L = 0.2\ \mu M$, $H = 1.2\ \mu M$, $K = 0.7\ \mu M$ and $\theta = 0.5$.

optimal threshold position for any specific set of parameters. Also, one can notice that decreasing the Hill coefficient increases the propagation time due to the fact that a gene is not instantly turned on/off when an input species crosses over/under the threshold (compare Fig. 3(b) and Fig. 5(b)). Increasing the Hill coefficient asymptotically reduces the propagation time to the one of the step-like regulation function and, thus, the optimal threshold asymptotically approaches the midpoint, $\lambda_T = 0.5$ (data not shown).

Next, we investigated the accuracy of the full-adder. The output sum for the input (H, L, L) produces the highest noise levels independent of the threshold position. Considering this case we determined the dependence of noise on the threshold position. The mathematical formula of the noise is too complicated to give any information about the system, but we can use it to generate numerical solutions. Fig. 6(a) shows that there is an optimal position of the threshold in terms of noise which differs from the optimal position in terms of speed, $\lambda_\eta \neq \lambda_T$. However, around the optimal threshold position in terms of noise (λ_η) the noise does not vary significantly (see Fig. 6(a)).

The system displays two optimal threshold positions, one for speed (λ_T) and one for noise (λ_η). If these two positions coincide ($\lambda_T = \lambda_\eta$) then the system has on optimal set of parameters and the engineer needs to set up the threshold to this position.

However, it is most likely, that these two threshold positions will differ, as it is the case with our full-adder. In this case, there is an optimal trade-off curve when the threshold resides between these two optimal positions (λ_T and λ_η). In addition any other trade-off curve is suboptimal comparing to this one.

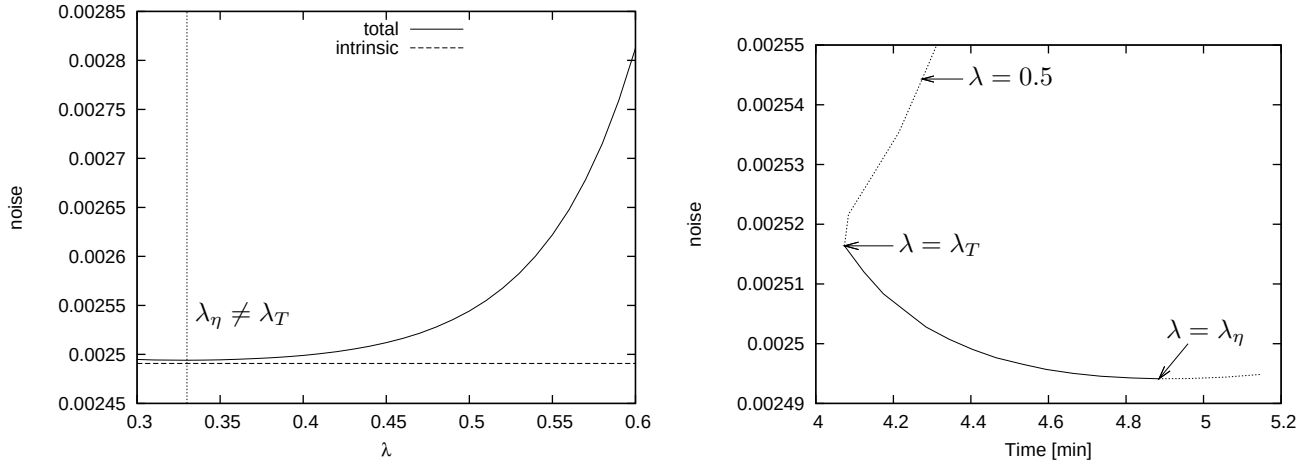
In our example of the full adder $0.5 \leq \lambda_\eta \leq \lambda_T$. Fig. 6(b) graphically represents the trade-off between noise and time based on the threshold position. We identified the optimal trade-off curve determined by $\lambda_\eta \leq \lambda \leq \lambda_T$. Any threshold in this interval can optimize the system either in speed or in accuracy, but never in both. However, for threshold positions outside this interval the system display sub-optimal trade-off curves; for $\lambda < \lambda_\eta$ or $\lambda > \lambda_T$ both the propagation time and the noise are worst compared to the ones in the optimal trade-off curve.

Discussion

In this contribution, we presented a general method for constructing arbitrarily large logical systems based on binary genes. For exemplification purpose, we designed a full-adder system formed of five genes. The approach modelled logic gates constructed using two cis-regulatory transcription control regions. This type of logic gates has been already synthetically engineered by synthetic biologists (Guet et al., 2002; Kramer et al., 2004; Yokobayashi et al., 2002; Cox III et al., 2007; Anderson et al., 2007; Sayut et al., 2009). We propose the tuning of the synthesis/decay rates in such a way that will permit interconnectivity of different gates/genes. This tuning represents basic requirement for a correct functioning of the logic system.

Recently we showed that leak free systems are optimal in terms of speed and noise (Zabet and Chu, 2010). However, Eq. (10) and Eq. (11) indicate that basal vanishing leak rates are very difficult to obtain. This suggests that leak free systems, although optimal in speed and noise are not always desirable, because they are likely to reduce the signal strength when thinking about interconnecting genes.

We also presented here an approach for selecting the set of parameters which optimizes the system in terms of speed



(a) optimum K for noise

(b) trade-off curves

Figure 6: *Optimum K for noise*. (a) The noise dependence on the threshold. The following set of parameters have been used: $V = 8 \times 10^{-16}\ l$, $\mu = 1\ min^{-1}$, $l = 6$, $L = 0.2\ \mu M$, $H = 1.2\ \mu M$, $K = 0.7\ \mu M$ and $\lambda = 0.5$. We assumed a Poisson noise of the three input species.

and accuracy under constant metabolic cost. Increasing the Hill coefficient will optimize both the speed and the accuracy, but this is not usually at the direct reach of synthetic biologists. However, the threshold can be altered by mutations of the regulatory binding sites (Buchler et al., 2005). We show that the threshold position, for a fixed Hill coefficient, influences both the speed (see Fig. 5(b)) and the noise (see Fig. 6(a)).

In an ideal system, a system with gates that display step-like regulation functions (infinite Hill coefficients), we found that the system has an optimal set of parameters (threshold positioned at the midpoint between the two steady states). This set of parameters maximizes both speed and accuracy for a fix cost. Moreover, the speed and the accuracy achieved in this type of system is the asymptotic limit that any biological real system can aim towards.

Real genes have finite low Hill coefficients and, in this case, a logic system will display two optimal sets of parameters: one in speed λ_T and another one in noise λ_η. We found that there is a trade-off curve between speed and accuracy which is bounded by these optimal sets of parameters (λ_T and λ_η) and any point between these two can optimize the system in either speed or accuracy. Nevertheless, any other set of parameters (the threshold outside this interval) is sub-optimal with respect to accuracy or speed.

This analysis showed that for finite low Hill coefficients there are two sets of parameters, one optimizing in terms of speed and the other on in terms of noise, when the metabolic cost is not increased. However, this analysis addressed only logic gates formed of individual genes. It was widely recognized, that network motifs can play a significant role in both speed and noise (Alon, 2007). Thus, further optimization can be achieved by considering logic gates built from more than one genes that form a network motif. Nevertheless, the details of this analysis need to be left for further research.

References

Ackers, G. K., Johnson, A. D., and Shea, M. A. (1982). Quantitative model for gene regulation by lambda phage repressor. *PNAS*, 79:1129–1133.

Alon, U. (2007). *An Introduction To System Biology. Design Principles of Biological Circuits*. Chapman & Hall/CRC Mathematical and Computational Biology Series.

Anderson, J. C., Voigt, C. A., and Arkin, A. P. (2007). Environmental signal integration by a modular and gate. *Molecular Systems Biology*, 3.

Bar-Even, A., Paulsson, J., Maheshri, N., Carmi, M., O'Shea, E., Pilpel, Y., and Barkai, N. (2006). Noise in protein expression scales with natural protein abundance. *Nature Genetics*, 38(6):636–643.

Bintu, L., Buchler, N. E., Garcia, H. G., Gerland, U., Hwa, T., Kondev, J., and Phillips, R. (2005). Transcriptional regulation by the numbers: models. *Current Opinion in Genetics and Development*, 15:116–124.

Buchler, N. E., Gerland, U., and Hwa, T. (2003). On schemes of combinatorial transcription logic. *PNAS*, 100(9):5136–5141.

Buchler, N. E., Gerland, U., and Hwa, T. (2005). Nonlinear protein degradation and the function of genetic circuits. *PNAS*, 102(27):9559–9564.

Chu, D., Zabet, N. R., and Mitavskiy, B. (2009). Models of transcription factor binding: Sensitivity of activation functions to model assumptions. *Journal of Theoretical Biology*, 257(3):419–429.

Cox III, R. S., Surette, M. G., and Elowitz, M. B. (2007). Programming gene expression with combinatorial promoters. *Molecular Systems Biology*, 3.

Elf, J. and Ehrenberg, M. (2003). Fast evaluation of fluctuations in biochemical networks with the linear noise approximation. *Genome Research*, 13:2475–2484.

Elowitz, M. B. and Leibler, S. (2000). A synthetic oscillatory network of transcriptional regulators. *Nature*, 403:335–338.

Gardner, T. S., Cantor, C. R., and Collins, J. J. (2000). Construction of a genetic toggle switch in escherichia coli. *Nature*, 403:339–342.

Guet, C. C., Elowitz, M. B., Hsing, W., and Leibler, S. (2002). Combinatorial synthesis of genetic networks. *Science*, 296:1466–1470.

Hermsen, R., Tans, S., and ten Wolde, P. R. (2006). Transcriptional regulation by competing transcription factor modules. *PLoS Computational Biology*, 2:1552–1560.

Kaern, M., Elston, T. C., Blake, W. J., and Collins, J. J. (2005). Stochasticity in gene expression: from theories to phenotypes. *Nature Reviews Genetics*, 6:451–464.

Kramer, B. P., Fischer, C., and Fussenegger, M. (2004). Biologic gates enable logical transcription control in mammalian cells. *Biotechnology and Bioengineering*, 87(4):478–484.

Magnasco, M. O. (1997). Chemical kinetics is turing universal. *Physical Review Letters*, 78(6):1190–1193.

Newman, J. R. S., Ghaemmaghami, S., Ihmels, J., Breslow, D. K., Noble, M., DeRisi, J. L., and Weissman, J. S. (2006). Single-cell proteomic analysis of s. cerevisiae reveals the architecture of biological noise. *Nature*, 441:840–846.

Paulsson, J. (2004). Summing up the noise in gene networks. *Nature*, 427:415–418.

Pedraza, J. M. and van Oudenaarden, A. (2005). Noise propagation in gene networks. *Science*, 307:1965–1969.

Sayut, D. J., Niu, Y., and Sun, L. (2009). Construction and enhancement of a minimal genetic and logic gate. *Applied and Environmental Microbiology*, 75(3):637–642.

Schilstra, M. J. and Nehaniv, C. L. (2008). Bio-logic: Gene expression and the laws of combinatorial logic. *Artificial Life*, 14:121–133.

Silva-Rocha, R. and deLorenzo, V. (2008). Mining logic gates in prokaryotic transcriptional regulation networks. *FEBS Letters*, 582:1237–1244.

van Kampen, N. (2007). *Stochastic processes in physics and chemistry*. North Holland, 3rd edition.

Weiss, R., Basu, S., Hooshangi, S., Kalmbach, A., Karig, D., Mehreja, R., and Netravali, I. (2003). Genetic circuit building blocks for cellular computation, communications, and signal processing. *Natural Computing*, 2:47–84.

Yokobayashi, Y., Weiss, R., and Arnold, F. H. (2002). Directed evolution of a genetic circuit. *PNAS*, 99(26):16587–16591.

Zabet, N. R. and Chu, D. F. (2010). Computational limits to binary genes. *Journal of the Royal Society Interface*, 7:945–954.

Associative memory in gene regulation networks

Richard A. Watson[1], C.L. Buckley[2], Rob Mills[1], Adam Davies[1]

[1]Natural Systems group, ECS, University of Southampton, U.K. [2]CCNR, Informatics, University of Sussex, U.K..
raw@ecs.soton.ac.uk

Abstract

The pattern of gene expression in the phenotype of an organism is determined in part by the dynamical attractors of the organism's gene regulation network. Changes to the connections in this network over evolutionary time alter the adult gene expression pattern and hence the fitness of the organism. However, the evolution of structure in gene expression networks (potentially reflecting past selective environments) and its affordances and limitations with respect to enhancing evolvability is poorly understood in general. In this paper we model the evolution of a gene regulation network in a controlled scenario. We show that selected changes to connections in the regulation network make the currently selected gene expression pattern more robust to environmental variation. Moreover, such changes to connections are necessarily 'Hebbian' – 'genes that fire together wire together' – i.e. genes whose expression is selected for in the same selective environments become co-regulated. Accordingly, in a manner formally equivalent to well-understood learning behaviour in artificial neural networks, a gene expression network will therefore develop a generalised associative memory of past selected phenotypes. This theoretical framework helps us to better understand the relationship between homeostasis and evolvability (i.e. selection to reduce variability facilitates structured variability), and shows that, in principle, a gene regulation network has the potential to develop 'recall' capabilities normally reserved for cognitive systems.

Evolvability

How natural selection results in the evolution of complexity, if it is natural selection that is responsible, is not yet understood [1,2]. It is easy to see how natural selection increases the frequency of fit phenotypes from a given distribution of phenotypic variants. But this is only part of the explanation. Although continued adaptation does not require that the available distribution of phenotypes is fitter than the parent on average (that would imply directed variation), continued increases in fitness and functionality require that this distribution includes at least some phenotypes that are fitter than the parent. This is often taken for granted, but experience in evolutionary algorithms and artificial life experiments suggests that such variants are quickly exhausted by selection, precluding further adaptation [2]. Thus the evolution of significant biological complexity requires that we explain how the distribution of phenotypes, resulting as they do from random variation in genotypes, includes phenotypes that are, not merely different from, but fitter than the parental type. The explanation might be, at least in part, that in natural organisms the distribution of phenotypic variants itself becomes better adapted over time [3] – hence enhancing *evolvability*, the ability of a population to evolve [4,5,6,7]. Since the processes of development, mapping genotype to phenotype, is itself genetically specified and subject to natural selection, this seems like a possibility, at least in principle.

However, although it is easy to say that natural selection should favour more evolvable genotypes, without a proximal account for the selective gradients that would produce such an outcome this is just wishful thinking. It is not so easy to pin down the source of a selection pressure that increases evolvability. For example, enhanced evolvability ought to mean that a genotype evolves *better*, not just that it evolves, and given that adaptive variants from a given phenotypic distribution are quickly exhausted it is hard to see how a variant genotype in a population that is stuck at a local optimum can be said to have better evolvability than another. This implies that the evolution of evolvability might require a constantly varying selective environment and multiple opportunities to generate and exploit variant phenotypic distributions. Moreover, if the environment changes in an entirely arbitrary fashion, a genotype to phenotype mapping cannot evolve to exploit it, so we are lead to the conclusion that such a mapping could only be adaptive if it exploits some kind of structure or regularity observed in the distribution of selective environment [8].

A simple way in which this might work is as follows. Different genotypes with the same phenotype might (nonetheless) have a different distribution of *phenotypic neighbours* - phenotypes produced through small mutations to the genotype. In a selective environment that varies from one selective regime to another (Fig.1), natural selection might favour genotypes that have phenotypes that are fit in one regime *and* have phenotypic neighbours that are fit in the other (over genotypes that have phenotypes that are equally fit in the first regime but do not have phenotypic neighbours that are fit in the other) [8]. In a sense, we can understand the propensity to produce phenotypes that are not currently selected for but have been selected for in the past as a kind of 'memory' of past selective environments [8], and under certain conditions evolved genotypes may even "generalise to future environments, exhibiting high adaptability to novel

goals". But exactly how this might happen, what the selective pressures are that might produce this outcome, and the limitations and affordances of such a process are poorly understood in general.

Part of the process might involve the evolution of modularity, for example [9,10]. That is, certain phenotypic features might become tightly integrated units (clusters of phenotypic features that co-vary), whilst others remain, or become, separated and vary independently. Such modularity might then provide, in effect, higher-level variation – i.e. variation at a higher-level of organisation [11]. Such high-level variability might in principle provide new combinations of modules with high probability (compared to the original distribution of 'atomic' character combinations) even though some particular combination of modules that is fit may not previously have been selected for.

Wagner *et al* [10] explain part of the proximal mechanism that might be involved in this process. Referring to genetic loci that affect the correlation of phenotypic traits [12], they state that "natural selection can act on [such loci] to either increase the correlation among traits or decrease it depending on whether the traits are simultaneously under directional selection or not. …[Resulting in] a reinforcement of pleiotropic effects among co-selected traits and suppression of pleiotropic effects that are not selected together" [10].

Wagner *et al* do not seem to notice, however, that this suggests intriguing parallels with Hebbian learning familiar in computational neuroscience [13,14]. Hebb's rule, in the context of neural network learning, is often represented by the slogan *neurons that fire together wire together*, meaning that synaptic connections are strengthened between neurons that have correlated activation in response to a stimulus. Formally, a common simplified form of Hebb's rule states that the change in a synaptic connection strength ω_{ij} is $\Delta\omega_{ij} = \delta s_i s_j$ where $\delta>0$ is a fixed parameter controlling the learning rate and s_n is the current activation of the n^{th} neuron. This learning rule has the effect of transforming correlated neural activations (created by an external stimulus) into causally linked neural activations. From a dynamical systems perspective, this has the effect of enlarging the basin of attraction for the current activation pattern/system configuration created by the stimulus. This type of learning can be used to train a recurrent neural network to store a given set of training patterns [15] thus forming what is known as an 'associative memory' of these patterns. A network trained with an associative memory then has the ability to 'recall' the previously seen training pattern that is most similar to a new partially specified or corrupted test pattern.

In this paper we investigate the possibility that a gene regulation network, capable in principle of exhibiting the same kind of dynamics as a recurrent neural network, is subject, over evolutionary timescales (not lifetimes [16]), to modifications in connections that are in principle the same as those produced by Hebbian learning familiar in neural network models. Thus *genes that fire together wire together* - i.e. genes whose expression is selected for in the same selective environments become co-regulated. Accordingly, the previously external cause of correlations in phenotypic characters (i.e. direct selection on expression patterns) becomes internalised (i.e. the result of a regulatory connection). A developmental trajectory determined by such an evolved network will then be able to reproduce a previously selected phenotype ballistically from an arbitrary initial condition using purely internalised dynamics, i.e. using a memory of what phenotypic characters work well together.

This analogy helps us to understand how a gene regulation network can modify the distribution of phenotypes in a manner that reflects structure in the selective environment. Specifically, we argue that evolved changes in regulatory connections will tend to cause the regulatory network as a whole to form an associative memory [15] of locally optimal phenotypes that have been visited in the past [17,18]. The evolved network has a dynamical behaviour which models the historical selective pressures on phenotypes (in the sense of having the same attractors) and can thereby create phenotypic distributions that are especially fit. In particular, an evolved network can produce a distribution of phenotypes that enables a population to escape locally optimal phenotypes (i.e. phenotypes that were locally optimal prior to the development of this regulation) in favour of superior optima. We also show that the proximal cause of these changes is not the teleological anticipation of future reward but something much more mundane – merely selection for robustness or canalisation of the current phenotype [5]. By analogy with the Baldwin effect [19], the internalised memory of previously found solutions enables previously evolved phenotypes to be produced innately by the developmental process. We therefore argue that selection for homeostasis on an immediate timescale (i.e. the ability to regulate a constant condition [20]), is the proximal cause of increased evolvability on larger timescales (i.e. increased ability for adaptation), as we will discuss.

Self-modelling dynamical systems

In related work [17,18] we have been developing the concept of a 'self-modelling' dynamical system – a complex adaptive system that creates a memory of its past dynamical behaviour. We have shown that if changes to connections are Hebbian and slow compared to the system's state dynamics, a complex adaptive system will form an associative memory of its own dynamical attractors that enables it to lower its energy more efficiently and completely when subjected to repeated perturbation [17]. The 'training patterns' in such a scenario are the configuration patterns that are commonly experienced under the network's intrinsic dynamics, hence '*self*-modelling' [18] – and if the system spends most of its time at locally optimal configurations, it is these configurations that the associative memory stores. From a neural network learning point of view, a network that forms a memory of its own attractors is a peculiar idea. Forming an associative memory means that a system forms attractors that represent particular patterns or state configurations. For a network to form an associative memory of its own attractors therefore seems redundant; it will be forming attractors that represent attractors that it already has. However, in forming an associative memory of its own attractors the system will nonetheless alter its attractors; it does not alter their positions

in state configuration space, but it does alter the size of their basins of attraction (i.e. the set of initial conditions that lead to a given attractor state via local energy minimisation). Specifically, the more often a particular state configuration is visited the more its basin of attraction will be enlarged and the more it will be visited in future, and so on. Because every initial condition is in exactly one basin of attraction it must be the case that some attractor basins are enlarged at the expense of others. Accordingly, attractors that have initially large basins of attraction will, with continued positive feedback, eventually out-compete all others until there is only one attractor remaining in the system.

Variation in the selective targets/initial conditions

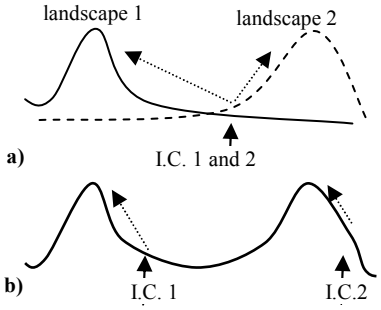


Fig.1. a) Adaptation to two different targets from the same initial condition (I.C.), b) Adaptation to one multi-modal target from two different initial conditions.

Before introducing our model, we briefly discuss an equivalence between multiple evolutionary episodes in different selective environments (Fig.1.a) and multiple evolutionary episodes from different initial conditions in a static (but multi-modal) selective environment (Fig.1.b). Parter *et al*, for example, conduct experiments using the former – and construct by hand different selective targets that are drawn from the same 'language' of tasks [8] (varying in a modular manner). We prefer the latter; using a single multi-modal landscape (created by modular epistasis) with repeated radical 'perturbations' of the evolved solution causing it to visit different local optima. What matters for our purposes is only the similarity or differences of the multiple 'targets'/ 'local optima', and the latter method has the advantage that, when the landscape is produced from the superposition of many low-order epistatic interactions (see methods), it does not require such explicit hand-crafting in this respect since structural similarity in the local optima results naturally.

A model for the concurrent evolution of gene expression patterns and regulation networks

Overview. Our model is intended to be as simple as possible. Presumably, the evolution of a gene expression network that is capable of creating correlated gene expression patterns and potentially sophisticated dynamical attractors was preceded by the evolution of static (unregulated) gene expression patterns. Likewise, the evolution of robust cell types in single-celled organisms, and gene expression networks that (partially) determine those cell types, presumably preceded the evolution of multi-cellular development and programmed cell differentiation. Accordingly, our model addresses the evolution of a gene expression pattern, and subsequently a regulation network, in a single-celled organism. By 'phenotype' we therefore simply mean a particular pattern gene expression, and by 'development' we simply mean the dynamical gene regulation process that creates the 'adult' gene expression pattern.

The model is not intended to be a literal model of biological processes. The critical features include a continuous-valued state vector representing a pattern of gene expression and a matrix of positive and negative connections representing up- and down-regulating connections between genes. These are subject to random variation and a selective environment that favours particular gene expression correlations. These components are linked together in a manner representing the concurrent evolution of a gene expression pattern and a gene regulation network but we aim to keep this protocol as simple as possible (see Fig. 2).

We assume that a pattern of gene expression is (epigenetically) inherited from one cell to the descendant cell and that a selection pressure on this phenotype causes it to evolve over many reproductions. A regulation network is also (genetically) inherited and subject to evolution via selection on the gene expression pattern that it modifies. We assume that every gene has the potential to regulate any other gene but that there is no significant regulation in the ancestral cell type (i.e. initially zero connections). Random variation in the connections of the network can introduce positive or negative correlations in the expression of genes which may or may not be beneficial given the current selective environment. So, in the lifetime of the cell, its initial gene expression pattern is inherited from the parent cell with random variation, this pattern of expression then forms the initial condition of the gene regulation network, which is then run for a number of time-steps (usually one) creating a slightly altered pattern of gene expression, and it is this pattern of expression which is interpreted as the phenotype of the organism and evaluated by the fitness function.

Evolutionary adaptation. The idea of evolved *correlations* between the expression of one gene and that of another invokes the notion of a *distribution* of phenotypes. When there are many copies of each genotype in a population, each one producing a phenotype from this distribution, selection on these individual phenotypes implicitly selects for genotypes that produce high fitness phenotype distributions [10]. However, we find that an explicit population with multiple copies of a genotype is more complicated than necessary. It is sufficient to merely compare the phenotype of a mutant to the phenotype of the original type and retain whichever is fitter. Hence we model the evolutionary process with a simple random mutation hill-climber (or '(1+1)ES'[21]) rather than a population-based evolutionary algorithm [3]. The latter merely adds additional stochastic fluctuations and unnecessary conceptual complications.

The overall architecture of the evolutionary model is depicted in Fig. 2. and detailed in Fig.3. Note that the gene

expression network does not so much represent a mapping from genotype to phenotype, as it is popularly conceived, so much as a mapping from an initial gene expression pattern to an 'adult' gene expression pattern. This adult gene expression pattern and the gene expression network is passed on the next generation (with random variation).

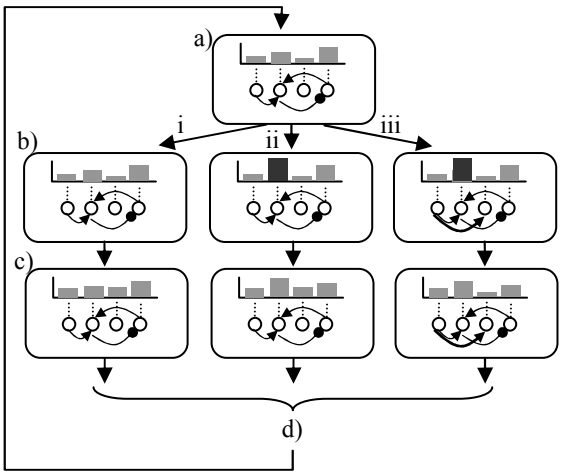


Fig.2: Schematic overview of the inheritance, regulation and selection processes (i.e. an iteration of the evolutionary hill-climber). a) A cell contains both an expression pattern and a genetically specified gene regulation network. b) Its descendents include individuals that are i) identical to the parent, ii) have a perturbed expression pattern (black), iii) have both a perturbed expression pattern and a genetically mutated regulation network (here depicted by an additional connection). c) The pattern of gene expression in each of these descendent cells is 'developed' or 'run' through their regulation networks creating three slightly different 'adult' gene expression patterns. d) The cell with the most fit gene expression pattern replaces the ancestral cell type.

The gene regulation network, *R*, (Fig. 3) is a matrix of connection strengths initialised to 0. The expression pattern, *E*, is set to a random configuration each t^*=5000 iterations (each gene expression level is set to a value drawn uniformly and independently in the range (-1,1)). This represents a radical environmental perturbation of the expression pattern and allows the expression pattern to visit the slopes of different local optima in the fitness landscape (Fig. 1) hence commencing a new evolutionary 'episode'. *E1*, *E2* and *E3* are the three modified expression patterns that result from the three descendents of the ancestral type (having no mutations, mutation to the expression pattern only, and mutation to both the expression pattern and the regulation network, respectively. We assume that mutation to the regulation network without mutation to the regulation pattern is unlikely). *mut* is a mutation function that introduces a small perturbation to the expression pattern or a small mutation to the regulation network. Specifically one of the existing expression levels or connection strengths (selected at random) is modified by adding a value drawn uniformly in the range (-1,1). (In test cases where the regulation network is not evolved, lines 2.c and 2.g are omitted.) *run*(*E,R*) is a function that 'develops' the initial expression pattern *E* by running the regulation network *R* for *p* time steps (*p*=1 by default) and returns a new expression pattern. For each time step the new activation level, s_i(t+1), of gene, *i*, is calculated using the old value with a decay term and a sum of weighted (positive or negative) inputs from the other genes in the network, as follows [22]:

$$s_i(t+1) = s_i(t) + T\left(\sum_j^N w_{ij}\sigma(s_j(t)) - s_i(t)\right) \quad \textbf{(1)}$$

where T=0.001 is a time constant, w_{ij} is the connection from gene *j* to gene *i*, $\sigma(x)=\tanh(x/10)$ is a sigmoidal output function determining the expression level of a gene with activation level *x* (representing the tendency of expression levels to saturate).

1. initialise regulation network, *R*.
2. *t*=0, repeat
   a. if (*t*=0) expression pattern, *E*=random, *t*=*t**;
   b. E'=*mut*(*E*);
   c. R'=*mut*(*R*);
   d. E1=*run*(*E*, *R*); E2=*run*(*E*', *R*); E3= *run*(*E*',*R*')
   e. *m*= max(*f*(*E1*),*f*(*E2*),*f*(*E3*))
   f. if (*f*(*E2*)=*m*) *E*=*E*';
   g. if (*f*(*E3*)=*m*) *E*=*E*', *R*= *R*';
   h. *t*=*t*-1

Fig. 3. Pseudocode of the inheritance, regulation and selection processes depicted in Fig. 2.

The selective environment. The fitness landscape is (initially) carefully controlled so that we can assess easily whether an evolved regulation network is creating appropriate correlations in the gene expression pattern. The minimal conceivable scenario is one where there are only two genes with selection for correlated expression in these two genes [10]. If we do not have any intrinsic preference for absolute gene expression levels, only for correlations, this means that there will be two locally optimal gene expression patterns of equal fitness – 'HH' and 'LL' (representing 'High' or 'Low' expression levels for the first and second genes). Alternatively, if we select for anti-correlation then these will be 'HL' and 'LH'. However, although we might be able to evolve a gene regulation network that supports correlation or anti-correlation in such a scenario, the evolutionary outcome will be somewhat degenerate in the sense that each of the two locally optimal gene expression patterns will have equal fitness and be equally likely to arise (from a random initial condition) without a regulation network.

Accordingly, we will examine the next simplest case; a system of four genes in two pairs. Here we can define a fitness function where 'HHHH' and 'LLLL' are maximally fit, but where 'HHLL' and 'LLHH' are local optima of lower fitness. Favouring pairs of co-expressed genes in this manner thus enables us to define a system with different-fitness optima without introducing a preference for absolute expression levels, or any asymmetries that would make one gene more important than any other. It also represents a minimally 'modular' fitness function. Naturally, we do not imagine that such a fitness landscape represents any realistic biological scenario – its structure is chosen merely to avoid obfuscating the significance of an evolved regulation network

with a complex adaptive landscape, and to test whether a network can create correlations that support co-regulation and create high-fitness phenotypes (we later investigate evolution on a 30-variable randomised landscape).

We construct a fitness function of this type using a sum of low-order (pair-wise) epistatic interactions [23] creating a locally smooth (but multi-modal) fitness landscape. Specifically, the fitness of an expression pattern, $S=<s_1,s_2\ldots s_N>$, is given by:

$$f(S) = \sum_{i}^{N}\sum_{j}^{N} e_{ij}\sigma(s_i)\sigma(s_j) \qquad \textbf{(2)}$$

where N is the number of genes in the system, s_i is the activation of the i^{th} gene, e_{ij} is the epistatic interaction between genes i and j, defined below and $\sigma(s)=\tanh(s/10)$ is the expression level of the gene, as before. The epistatic matrix is as follows: $e_{12}=e_{34}=1$, $e_{13}=e_{14}=e_{23}=e_{24}=0.1$, else $e_{ij}=0$ – thus defining the two pairs of strongly interacting genes (s_1/s_2 and s_3/s_4), with only weak interactions between these pairs as discussed above.

Results

Evolution of expression patterns without evolved regulation. Fig. 4 (right) illustrates the evolution of an expression pattern (without evolved regulation) over 10^5 evolutionary time steps (therefore showing 20 evolutionary episodes between radical perturbations of the expression pattern). This clearly shows the four locally optimal expression patterns (HHHH, HHLL, LLHH, and LLLL) and that patterns where the four genes are all high or all low have the highest fitness. The fitness values at each of the evolutionary local maxima attained (i.e. at each t=1 time step) may be either in the lower class or the higher class (see Fig. 4). The proportion of high and low fitness optima found indicates the size of the evolutionary basin of attraction for each class of optima. For these parameters under these conditions (without a regulation network) we find that the evolutionary basin of attraction for the fitter local optima accounts for about 73% of the initial configuration space (averaged over 300 evolutionary episodes).

Evolved regulation. Under natural selection, evolved changes to the connections in the regulation network must be those that change the expression pattern in the direction that increases fitness; and that direction may be different depending on the currently selected expression pattern. Since the evolved expression pattern very quickly settles into one attractor or the other, most evolution of the regulation network will occur when the expression pattern is at or near a locally optimal configuration. So, as a first step to investigating the evolution of a regulation network we evolve the regulation network when the expression pattern is 'clamped' at a single locally optimal configuration. Specifically, in line 2.a of Fig.3, E is set to $<s,s,s,s>$ (s=5) instead of a random configuration. We find that after 100,000 more evolutionary steps the evolved connections in the regulation network are all positive (Table 1). In contrast, when the clamped expression pattern is HHLL ($E=<s,s,-s,-s>$), the evolved connections are positive on the block diagonal (shaded) and negative elsewhere (Table 2).

It is crucial to note that the signs of these connections do not directly reflect the epistatic interactions in the fitness landscape – the intrinsic epistasis in the landscape does not change between the HHHH and HHLL test cases. Rather the evolved connections reflect the expression states experienced when the regulatory connection is altered (i.e. s_i=H/s_j=H and s_i=L/s_j=L expression levels create selection for positive connections, whereas s_i=H/s_j=L and s_i=L/s_j=H expression levels evolve negative connections). This clearly follows Hebbian principles – when equal gene expression levels are selected together they wire together positively, when one is selected to be high and the other low, they wire together negatively.

However, the sign of the connection is really just a labelling convention – what really matters with respect to demonstrating Hebbian learning is that these evolved connections increase the basin of attraction for the current expression pattern. Fig. 5 shows, for example, the effect of the connections evolved at the HHLL expression pattern (i.e. Table 2). We see that the evolved connections change the size of the HHLL attractor basin to fill 100% of the configuration space (conversely, when regulation is evolved at the HHHH expression pattern, Table 1, this pattern comes to occupy 100% of the configuration space).

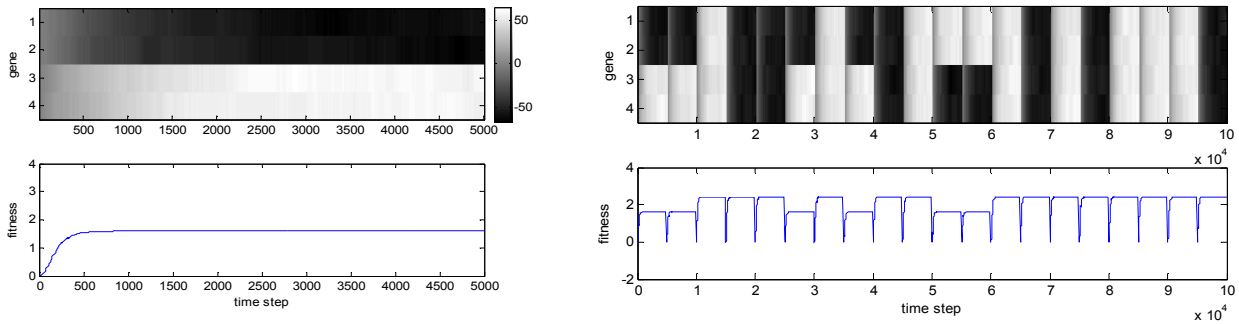


Fig.4. left) Evolution of a gene expression pattern without regulation for one evolutionary episode (5000 time steps). This happens to arrive at the locally optimal expression pattern where genes 1 & 2 are low, and 3 & 4 are high. Right) A longer run (100,000 time steps) including 20 evolutionary episodes, again without evolved regulation. Note that with these parameters, each evolutionary episode very quickly reaches a locally optimal expression pattern (i.e. transients are short). Note that fitnesses at evolutionary attractors fall into two classes (roughly those below a fitness of 2 and those above).

i/j	1=H	2=H	3=H	4=H
1=H	89.13	160.18	126.02	104.35
2=H	120.42	58.95	87.40	152.94
3=H	163.49	76.60	152.08	79.10
4=H	197.69	56.58	158.36	159.87

Table 1: evolved connections when the expression pattern is HHHH.

i/j	1=H	2=H	3=L	4=L
1=H	80.93	105.81	-60.99	-146.92
2=H	153.02	120.27	-94.84	-108.03
3=L	-157.65	-125.27	69.33	163.97
4=L	-156.00	-140.19	84.13	69.17

Table 2: evolved connections when the expression pattern is HHLL.

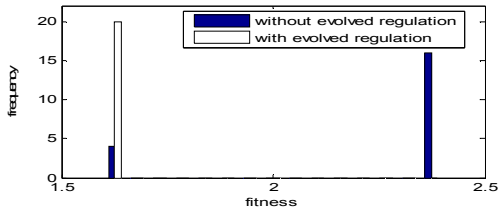


Fig. 5. Number of evolutionary episodes (from 20) finding each locally optimal phenotype before and after evolution of the regulation network. When the gene expression pattern is held at a low fitness attractor, the evolved regulation network canalises this pattern.

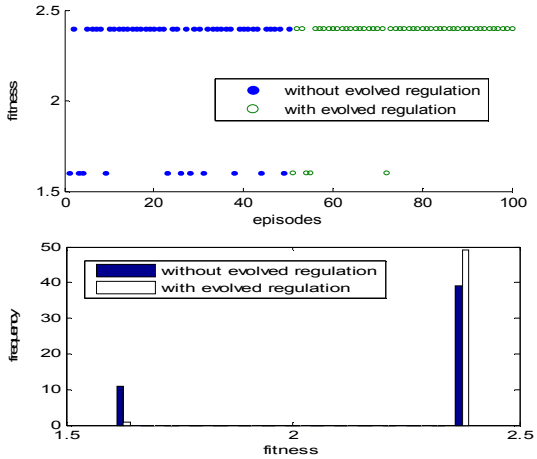


Fig. 6. When the gene expression pattern is evolved freely, evolved regulation canalises the fitter pattern (since it is visited more often). Upper) The evolution of a gene expression pattern without evolvable regulation (episodes 1-50) and with evolvable regulation (episodes 51-100). Each point represents a locally optimal expression pattern found via a single evolutionary episode from a random initial condition. Lower) see Fig.5.

i/j	1	2	3	4
1	437.37	566.40	60.50	72.32
2	269.72	389.88	253.21	212.56
3	184.52	98.54	270.58	351.04
4	448.46	-25.23	373.18	246.46

Table 3: Evolved regulatory connections when the expression pattern is not clamped. Although there is a lot of variation, the average value in the block diagonal (shaded) is 363 and elsewhere 163. The generally positive values mean that both the superior HHHH/LLLL attractor (Table 1) and the inferior HHLL/LLHH attractor (Table 2) have been reinforced, but the lower values off the diagonal retain a reflection of the underlying modularity.

Note that the evolved regulation network does not necessarily increase the basin of attraction for the fitter phenotypes, but rather for the phenotype present at the time that changes to the regulation network were evolved. Next, we evolve the regulation network without clamping the expression pattern. Without regulation the fitter phenotype is already found 73% of the time, so if the evolved regulation network reinforces the fitter attractor 73% of the time and the less fit attractor only 27% of the time then on average the fitter attractor should be enlarged more often than the less fit attractor in a positive feedback manner and it will eventually outcompete it (Fig. 6, Table 3).

Collectively, these results demonstrate that selection favours changes to regulation connections that reflect co-expression in the current phenotype, and that these connections increase the basin of attraction for that expression pattern, as expected for Hebbian changes to connections. They also show that in a fitness landscape where fitter patterns have larger basins (as is necessarily the case when the fitness landscape is created from the superposition of many low order interactions [18,24,25]) enlargement of these fitter basins will outcompete lower fitness basins and create a regulation network that produces fit phenotypes more reliably. Although this result is somewhat underwhelming in this almost trivial (two attractor) system, in addition to the basic Hebbian principles, it also illustrates a further vital point. Specifically, the fact that the basin of attraction for the superior phenotypes is now almost 100% means that there are some initial conditions that used to lead natural selection of expression patterns to find the inferior phenotype but now evolution of expression patterns from these same initial conditions leads to the superior phenotype. That is, random variation in the expression pattern that would increase fitness by moving toward the inferior phenotype is being suppressed by the regulation network, and variation that moves the expression pattern toward the superior phenotype is being supported. This means that given the evolved regulation network, the evolutionary trajectory of the expression pattern is able to 'climb out' of the basin of attraction for the inferior phenotype and secure adaptation in the direction of the superior phenotype. Evolution of regulation that avoids sub-optimal phenotypes in a larger system is shown in Fig.7[1].

Ballistic development. Thus far the developmental network is only run for one time step (p=1) per application of natural selection. This is sufficient to induce significant correlations and redirect the evolutionary trajectory of expression patterns, as we have shown. But in general one might expect a regulation network to 'develop' an initial expression pattern into a fit adult expression pattern for many time steps without the need for selection to act on the result of every intermediate step. We therefore examine a 'ballistic' developmental trajectory (i.e. *run*(E,R) with p=5000, rather

[1] Here fitnesses are measured on thresholded expression values (>0→1, <0→-1) to ensure that an increase in fitness is the result of increasing the basin of attraction for a fit configuration pattern and not merely the result of increasing the magnitude of the expression levels (see measuring energy with the original weights rather than the learned weights [18]).

than 5000 iterations of the evolutionary cycle with p=1) using the regulation network evolved in Fig.7, applied to an initially random expression pattern. We find that even though selection is not being applied the fitness of the phenotype increases monotonically at each developmental step, and in fact the phenotypic attractor that is reached by this ballistic developmental process is the same attractor that is reached when selection was applied (Fig. 8). Thus selection on intermediate phenotypes (and epigenetic inheritance) has become redundant because development can now 'recall' the result of, or recapitulate, what was previously an entire evolutionary episode from any initial condition. Analogy with the Baldwin effect, where phenotypes that were previously acquired by lifetime learning are latterly exhibited innately [19], is provocative.

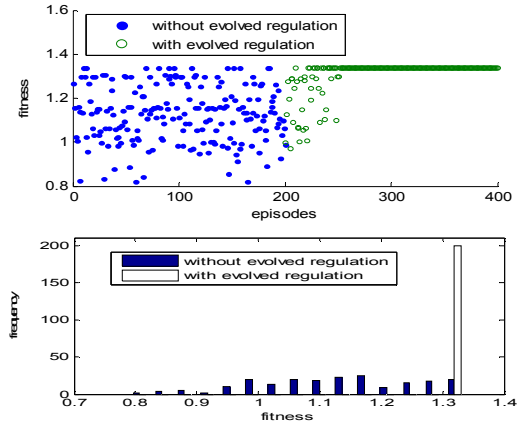


Fig. 7. As per Fig. 6 for a system of 30 genes with random epistasis in the fitness function (Eq.2 with each e_{ij} drawn randomly (-1,1)). The basin of attraction for the highest fitness optima is initially only 9.5%, meaning that 90.5% of episodes get stuck at some other sub-optimal phenotype. After the regulation network is evolved all of these inferior phenotypes are reliably evaded regardless of the initial gene expression pattern.

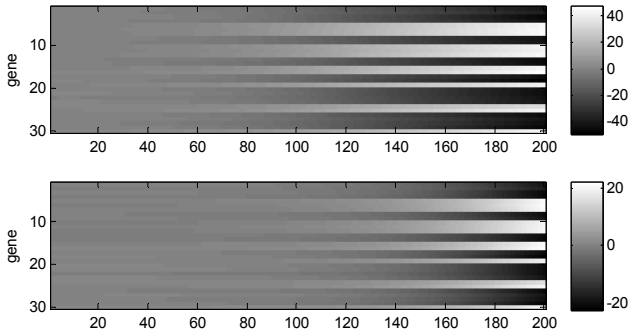


Fig. 8. 200 steps of an evolutionary episode with the evolved regulation network (upper) are accurately mimicked by ballistic (unselected) multi-step development using the same network (lower).

Discussion

Distal 'explanation'? On the one hand, the result of Fig. 7 is just what one might expect – selection favours fit phenotypes and if there are regulation networks that produce fit phenotypes reliably then they will be selected for. But this distal reasoning is misleading and obscures the proximal mechanism by which this result is produced. Note that a regulation network can preclude fit phenotypes just as easily, if not more so, than it might support them – it has 'masking' as well as 'guiding' possibilities [26] – and the evolution of a useful regulation network must not be taken for granted.

The point we illustrate in the initial results (Tables 1 & 2, Fig. 5) is that the evolved regulation network is not favouring fit phenotypes in a direct sense, it is merely canalising the *current* phenotype. This is not an obvious route to finding fit regulation networks and one might expect that, at best, it will ultimately result in canalising an average-fitness phenotype, *not* the fittest phenotype. But when the distribution of phenotypes visited over many evolutionary episodes has some correlations (or anti-correlations) that occur more frequently than others, it is these correlations that are ultimately reinforced by the regulation network (Fig. 6). If these correlations appropriately reflect the epistatic structure in the fitness landscape then they can enhance evolvability. In this manner the regulation network comes to represent the structure of the epistasis (or more exactly, the structure of the correlations between phenotypic characters produced by the epistasis) in the selective history over which the regulation network was evolved. But by the same reasoning, when the correlations in characters in the phenotypes visited do not reflect the epistatic structure of the fitness landscape in general, and instead reflect arbitrary phenotypic correlations, the regulation network will evolve to represent correlations that are not of especially high fitness. We demonstrate this by increasing the mutation rate on the regulation network, and/or increasing the duration of each evolutionary episode, such that the evolutionary history does not visit a representative sample of phenotypic attractors before the regulation network fixes on a particular attractor. On average this causes the regulation network to fix a phenotype with an average fitness rather than the highest fitness. Accordingly, it is not to be taken for granted that a gene regulation network will evolve to enhance high-fitness phenotypes just because such a network exists in the space of possible networks.

Proximal explanation. We should therefore investigate the proximal selection pressures involved in the initial result of Tables 2 & 3 (i.e. these data show that the selected changes to regulation connections are Hebbian but they do not explain why). Why is it that connections that reinforce the *current* phenotype are evolved instead of, say, connections that enlarge the basin of attraction for the fittest possible phenotype? (And how does this ultimately result in fit phenotypes?) To probe this issue we must consider the immediate selective gradients in the vicinity of the current phenotype. Specifically, for a change to a regulation connection to confer a selective advantage it must change the configuration of expression levels in a manner that increases fitness. However, most of the time, the current phenotype is a locally optimal configuration of gene expression levels. Thus, it might seem that the only way for a change to a connection to confer a fitness advantage would be when such a change moves the current phenotype out of the current local optimum and into a better one in a single mutation. But such a possibility is highly unlikely when the nearest phenotype of higher fitness is not an immediate neighbour.

In fact, something much more subtle is at work. Although most of the time the phenotype is *almost* locally optimal it is in fact constantly perturbed by the small environmental perturbations (line 2.b in Fig. 3). Changes to the regulation network can therefore be favoured by selection if they have the effect of returning the phenotype to the local optimum more quickly or more completely after this minor perturbation. In other words, we argue that changes to the regulation network are selected for merely because they make the current (almost locally optimal) phenotype more robust or more homeostatic. We test this hypothesis by removing line 2.a., the small environmental perturbations, and repeating the experiment shown in Table 2. In this case we find that there are no changes to the regulation network that are selected, in fact all changes are either neutral or deleterious. Thus the small environmental perturbations serve a dual role – they first provide (unregulated) phenotypic variation that selection can act on to find locally optimal phenotypes, but they also create instability in these phenotypes creating a selective gradient that favours a regulation network that canalises these phenotypes. We argue that this dual role of variation is not special to this particular model but will necessarily occur whenever random variation, necessary for evolution to act at all, is present.

From proximal causes to distal consequences. This proximal mechanism is also not very surprising given what one might expect from natural selection – if natural selection can act on the distribution of phenotypes in such a way as to narrow that distribution onto the fitter phenotypes, then a regulation network, for example, that provides such an outcome will be selected for. But canalisation – a *reduction* in the distribution of phenotypic characters – seems opposed to concepts of evolvability and *increases* in adaptability. However, a selection pressure for robustness can result in increased adaptability – in essence evolvability is the complement of canalisation [5]. The basic conceptual link is that restricting variation in phenotypic characters that are detrimental, whilst permitting continued variation in characters that have the potential to be beneficial, enhances adaptation rather than restricts it. But it is crucial to realise that in the current model the canalisation provided by the regulation network does not merely restrict variation in some characters but rather it reduces the degrees of freedom in the *correlation* of phenotypic characters [4].

In contrast, note that in Hinton and Nowlan's model [19] for example, canalisation acts to reduce the variation in each phene independently. This therefore cannot act like an associative memory – it is not a memory of what things have *co-occurred* (i.e. have been selected together in the same environments) only of what things have *occurred* (been selected). The fact that the memory in our evolved regulation networks is associative is evidenced by the fact that variation in all phenes is still possible (when the network canalises the fitter attractor it actually canalises both HHHH and LLLL). This is crucial because if no further variation in phenotypic characters was possible we would conclude that canalisation had precluded further adaptation, but when canalisation creates correlations in phenotypic variation it is plausible to interpret this as smarter adaptation, i.e. a more evolvable genotype, rather than an unevolvable genotype. This is really a matter of perspective however, since both types of canalisation (associative and non-associative) necessarily reduce the space of phenotypic possibilities.

Limitations and further work

Our gene expression network uses signed expression levels to facilitate straightforward comparison with Hebb's rule, but negative expression levels are biologically unnatural. We have also hinted at the sensitivity of the results to the timescales of evolutionary changes to expression patterns and to the regulation network, and to the period of the perturbations/ evolutionary episodes, but we have not yet examined this sensitivity carefully.

In related work we are interested in the question of whether individual agents in a complex adaptive system that can alter the strength of connections with one another will tend to do so in a Hebbian manner [17,27,28]. In this paper we have shown that selection on a network as a whole produces Hebbian changes to connections, but we suspect that the same effect occurs if each gene in the network is evolved independently. This hints at an explanation for how a network of 'selfish' genes can coordinate with one another in a manner that creates fit phenotypes despite being selected as individuals in sexual organisms. This then parallels work we are developing in the context of co-evolving species in an ecosystem where species may evolve the coefficients of a Lotka-Volterra system [27] or evolve symbiotic relationships [29], and connects with 'social niche construction' concepts [30].

The fact that natural selection is involved in this model should not to be mistaken for evidence of how 'clever' natural selection is. On the contrary, we have shown that given an appropriate (i.e. association-based) representation, a hill-climber can produce these results. Moreover, the proximal cause of these results is that selection is decreasing variability which is something that hardly warrants natural selection at all [17,18,31]. We think it more fruitful to ascribe the 'cleverness' of the result to the ability of an appropriate substrate to 'yield' or 'relax' to structured perturbation in a manner that reduces or dampens the effects of such perturbations [31]. This is supported by the observation that Hebbian changes to connections are equivalent to changes in connections that reduce the energy of a system [17].

Conclusions

Wagner *et al* [10] suggest that phenotypic correlations will evolve in a manner we recognise as Hebbian. Our conclusions, originating from separate motivations [11,17], agree but differ in emphasis – whereas Wagner *et al* address the rate of adaptation created by a correlated phenotypic distribution we emphasise the robustness or stability of a phenotype under environmental perturbation. But the mechanisms are deeply related because resilience is just another way to say that a phenotype 're-adapts' quickly. All of the other results we have shown – the enlargement of the basin of attraction for the current phenotype, the ability to 'recall' fit phenotypes that have been selected for in the past,

and the ability for a developmental trajectory to recapitulate what was previously an evolutionary trajectory – follow from this basic observation and dynamics that are already well-understood in neural networks. This theoretical framework helps us to better understand the relationship between homeostasis and evolvability (i.e. selection to differentially reduce variability facilitates structured variability), and shows that, in principle, a gene regulation network has the potential to exhibit 'recall' capabilities normally considered to be the exclusive purview of cognitive systems.

Acknowledgments. Chrisantha Fernando, Jason Noble, Alexandra Penn, Simon Powers, Marie Redron.

References

1. Bedau, M.A., McCaskill, J.S. *et al*., (2000) Open problems in artificial life, *Artificial Life*, 6(4):363-76)
2. Bedau, M.A. (2006) The Evolution of Complexity. Symposium on "The making up of organisms" Ecole Normale Supérieure, Paris.
3. Toussaint, M., & von Seelen, W. (2007) Complex adaptation and system structure, *BioSystems* 90: 769–782
4. Wagner G.P. & L. Altenberg, L. (1996). Complex adaptations and the evolution of evolvability. *Evolution* 50:967–976.
5. Kirchner, M. and Gerhart, J. (1998). Evolvability. Proc. Natl. Acad. Sci. USA, 95:8420–8427.
6. Izquierdo, E.J. & Fernando, C. (2008) The Evolution of Evolvability in Gene Transcription Networks. Proceedings of Alife XI.
7. Crombach A, Hogeweg P (2008) Evolution of Evolvability in Gene Regulatory Networks. *PLoS Comput Biol* 4(7).
8. Parter, M., Kashtan, N., Alon, U. (2008) Facilitated Variation: How Evolution Learns from Past Environments To Generalize to New Environments. *PLoS Comput Biol* 4(11): e1000206.
9. Lipson, H., J.B. Pollack, & N.P. Suh. (2002).. On the origin of modular variation. *Evolution* 56, 1549-1556
10. Wagner, G.P., Pavlicev, M., & Cheverud, J.M. (2007) The road to modularity. *Nature Reviews Genetics* 8: 921-931.
11. Mills, R. (2010) *How Micro-Evolution Can Guide Macro-Evolution: Multi-Scale Search via Evolved Modular Variation.* PhD thesis, ECS, University of Southampton.
12. Pavlicev, M, Kenney-Hunt, J.P., Norgard, E.A., Roseman, C.C., Wolf, J.B. & Cheverud, J.M. (2008) Genetic variation in pleiotropy: differential epistasis as a source of variation in the allometric relationship between long bone lengths and body weight. *Evolution.* 2008 Jan;62(1):199-213.
13. Hebb, D.O. (1949) *The organization of behaviour*. New York: Wiley.
14. Ackley, D.H., Hinton, G.E., Sejnowski, T.J. (1985) A Learning Algorithm for Boltzmann Machines, *Cognitive Science*, 9: 147-169.
15. Hopfield, J.J. (1982) Neural networks and physical systems with emergent collective computational abilities, *PNAS USA*, 79 (8) 2554.
16. Fernando, C. Liekens, A.M.L., Bingle, L.E.H., Beck, C., Lenser, T., Stekel, D.J. ,Rowe, J.E. (2008) Molecular circuits for associative learning in single-celled organisms. *Journal of the Royal Society Interface.* 6:463-469.
17. Watson, R. A., Mills, R, and Buckley, C. L. (2009) *Global Adaptation in Networks of Selfish Components: Emergent Associative Memory at the System Scale.* Submitted/Tech. Report, ECS, University of Southampton.
18. Watson R.A., Buckley, C.L. & Mills, R. (2010). Optimisation in 'Self-modelling' Complex Adaptive Systems, *Complexity* (under revision)/ Tech. Report, ECS, University of Southampton.
19. Hinton, G.E. & Nowlan, S.J. (1987) How Learning Can Guide Evolution. *Complex Systems*, 1, 495-502
20. Ashby, W.R. (1952) *Design for a Brain*, Chapman & Hall
21. Rechenberg, I., *Evolutionsstrategie: Optimierung technischer Systeme und Prinzipien der biologischen Evolution* , Frommann-Holzboog, Stuttgart, 1973.
22. Beer, R.D. (1995) On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior*. 3(4).
23. Kauffman, S.A. (1989) Adaptation on rugged fitness landscapes. *Lectures in the Sciences of Complexity*, 527-618.
24. Fontanari, J.F. (1990) Generalization in a Hopfield network. *Journal de Physique*, 51, 2421-2430.
25. Kryzhanovsky, B. & Kryzhanovsky, V. (2008) Binary Optimization: On the Probability of a Local Minimum Detection in Random Search. *ICAISC 2008*: 89-100.
26. Mayley, G. (1997). Guiding or hiding: Explorations into the effects of learning on the rate of evolution. *Procs. ECAL 4*, 135-144.
27. Lewis, M. (2009) *An Investigation Into The Evolution Of Relationships Between Species In An Ecosystem*. MSc Dissertation, ECS, University of Southampton.
28. Davies, A.P., Watson, R.A., Mills, R., Buckley, C. L., Noble, J. (2010) If you can't be with the one you love, love the one you're with: How individual habituation of agent interactions improves global utility. *ALife XII* (to appear).
29. Watson, R.A., Palmius, N., Mills, R., Powers, S.T., & Penn, A.S. (2009a) Can Selfish Symbioses Effect Higher-level Selection? *European Conference on Artificial Life 2009*. (in press).
30. Powers, S.T., Mills, R. Penn, A.S., Watson, R.A. (2009) Social niche construction provides an adaptive explanation for new levels of individuality (ABSTRACT), *Procs of Workshop on Levels of Selection and Individuality in Evolution*, *ECAL 2009*.
31. Watson, R.A., Mills, R. Buckley, C.L., Powers, S.T., Penn, A.S., Davies, A.P., Noble, J. & Bullock, S.G. (2010) Adaptation without natural selection (ABSTRACT). *ALife XII* (to appear).

Processing Signals with Evolving Artificial Gene Regulatory Networks

Michał Joachimczak[1] and Borys Wróbel[1,2]
[1]Computational Biology Group, Institute of Oceanology, Polish Academy of Sciences
[2]Laboratory of Bioinformatics, Adam Mickiewicz University in Poznań, Poland
{mjoach,bwrobel}@iopan.gda.pl

Abstract

Computational properties of gene regulatory networks (GRNs) are of great interest in the field of systems biology and, increasingly, in the field of artificial life. Understanding how GRNs work and evolve may help in elucidating the properties of real biological networks and in designing new biological networks for practical applications. Here we investigate the possibility to evolve artificial GRNs that can generate or process continuous signals represented by concentrations of artificial substances. We use a biologically-inspired model of regulatory networks. The way the nodes in the GRN (regulatory units) are connected and the weights of connections are encoded in a linear genome. A genetic algorithm is used to obtain GRNs that can solve problems with increasing difficulty. Some of these problems require performing simple mathematical operations and sustaining memory. We analyse if the solutions are general by presenting the GRNs with input patterns that were not used for fitness evaluation during evolution. We also briefly discuss the advantages of using biologically-inspired GRN-like systems for control problems and compare them with systems inspired by neural networks.

Introduction

The genes in the genomes (DNA) of all organisms encode indirectly 3-dimensional structures of complex chemical polymers (RNA, proteins). When the genes are expressed, these polymers are produced in the cell. Cells consist of a genome, gene products, and the chemical substances these products help to construct (by chemical reactions) and/or transport into the cell from the outside environment. Chemical substances in the cell are a part of an intricate control mechanism. The presence of particular gene products and chemical substances in the cell at a particular moment determines what genes will be expressed at the next moment, and thus what will be produced. The regulation of gene expression occurs first of all at the level of transcription: formation of RNA molecules with the sequence corresponding to the DNA sequence in the genome. Some of these RNA molecules later determine the sequence of proteins. Some proteins (called transcription factors, TFs) have chemical affinity to particular regions in the DNA. Binding of such proteins to DNA may lower or increase the expression of the genes nearby. This is just one example of chemical interactions that regulate gene expression, but others follow similar rules.

A network of such regulatory processes is known as a gene regulatory network (GRN). GRNs can be thought of as life's primary computers, organizing all cellular processes. The regulatory properties of such networks and their use for control of artificial and biological systems are of great interest for the Artificial Life and the Systems/Synthetic Biology research community. Biological GRNs are robust to external interferences and to damages caused by mutations. They are able to control the development of an organism consisting of billions of cells. In a developing or adult multicellular organism, each cell is controlled by a GRN with essentially the same structure. It is the state of the network (concentration of substances) that makes the cells behave differently, depending on their local environment.

Artificial models of GRNs were previously used to investigate statistical properties of GRNs, such as the small world property or the dominant motifs (Kuo et al., 2006; Nicolau and Schoenauer, 2009). Network dynamics and evolution of networks with certain patterns of gene expression has also been explored to some extent (Banzhaf, 2003; Knabe et al., 2006; Kuo et al., 2004; Reil, 1999). So was the application of artificial GRNs for control problems, such as animat control (Bentley, 2004; Taylor, 2004; Quick et al., 2003) and artificial multicellular development. Indeed, we have originally formulated the GRN model used in this work to control multicellular patterning of 3-dimensional artificial embryos (Joachimczak and Wróbel, 2009), inspired by the model presented by Eggenberger (1997). Similar models have been proposed (e.g. Schramm et al., 2009; Andersen et al., 2009), so it is interesting to explore the computational properties of such networks.

GRN topology in our model is encoded in a linear genome which consists of genetic elements forming regulatory units (nodes in the network). Connections between nodes are defined by interactions between artificial TFs and regulatory regions ("promoters"). The concentrations of TFs increase and decrease in a continuous manner. There is no limit on

the number of nodes, number of connections per node or total number of connections. Defining such limits would be beneficial from the engineering point of view (it would decrease the vast search space of possible solutions). However, we are not interested here in solving a particular engineering problem, but rather in investigating the computational properties and evolvability of artificial but biologically realistic regulatory networks.

In this paper we will aim to evolve systems in which the expression of genes marked as the GRN output follows a predefined target pattern. In most of the experiments the target will depend on the input to the network. From the biological point of view the input can be understood as a concentration of a chemical substance in the environment. From the engineering point of view, the input is a continuous signal. In other words, we will describe networks evolved to generate or process signals, in particular, signals in which information is encoded in chemical pulses: coupled increases/decreases of substance concentration.

Artificially designed regulatory networks that can perform desired tasks and react to external input are of recent interest of the field of Synthetic Biology. Biological GRNs in which gene expression oscillates and GRNs created to count subsequent external signals (Elowitz and Leibler, 2000; Friedland et al., 2009) are a step towards engineering networks to produce proteins or RNAs in an intelligent and designed manner, for therapeutic or industrial purposes.

In the following section, our model is briefly described. The evolvability in various signal processing tasks and the generality of the solutions is then discussed for each task separately. General conclusions and the perspectives for future work follow.

The model

Genome and genetic elements

Genomes are composed of a list of genetic elements. Several genetic elements form a regulatory unit, which corresponds to a node in a regulatory network. Genetic elements fall into three classes. "Genes" are elements that code products (transcription factors, TFs). Products can bind to "promoters" (a generic term for regulatory regions). "Special elements" code for either external inputs or outputs of the regulatory network.

The genome is parsed sequentially and divided into regulatory units whenever a series of promoters followed by a series of genes is found (Fig. 1). In other words, each regulatory unit can be composed of one or several regulatory elements and one of several genes encoding TFs. In the next step, special elements are assigned to inputs or outputs, according to their type. The first special element of type one is assigned to the first input, and so on. The same goes for special elements of type two and the outputs. The number of inputs/outputs depends on the particular experiment. If there are more special elements of a particular type than inputs/outputs, they are ignored.

By computing affinities between all products and all promoters, connections between regulatory units are formed. This is how a gene regulatory network (GRN) emerges, with each regulatory unit becoming a single node.

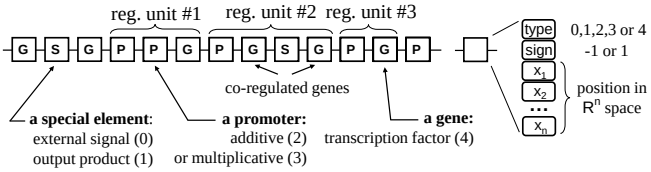


Figure 1: The genome and the structure of a single genetic element. Each element consists of a type field, a sign field, and a sequence of N real values used to determine affinity to other elements ($N = 2$ was used in this paper).

Each genetic element in our system encodes a point in N-dimensional space (Fig. 1). This allows to calculate product-promoter affinity, based on the Euclidean distance between these points (the affinity is high when the distance is small). If the distance is larger than a cut-off value, there is no affinity. This prevents full connectivity in the network. The product of sign fields of the two elements determines the sign of the connection (which can be activatory or inhibitory). The coordinates coded in genetic elements can mutate, so as the genomes evolve, the points in N-dimensional space that correspond to the elements approach one another or move away. Neutral mutations result in a random walk in this space, so only selection limits spreading of the points over time.

The activation of a promoter is a sum of the concentration of all products that bind to it, weighted by their affinities. Promoters in our systems can be either additive or multiplicative. The presence of a multiplicative promoter in a regulatory unit results in a strict requirement for the presence of a binding product, otherwise the unit is not expressed. To compute expression of a given regulatory unit, the sum of activations of its additive promoters is multiplied by the activation of its every multiplicative promoter. The result (A) allows to calculate the synthesis/degradation rate of all products in a given regulatory unit: $\frac{dL}{dt} = f_A(A) - L$, where L is the current concentration, and $f_A(A) = \frac{2}{1+e^{-(A-1)}}$. This sigmoid function can give positive or negative values. The concentration will increase if synthesis rate is higher than that of spontaneous degradation. Otherwise, the degradation will be slowed down or indeed increased (when the $f_A(A)$ is negative). Fig. 2 provides an overview of the time scale of spontaneous product degradation in our system.

Special elements in our system, as any other genetic elements, are associated with points in N-dimensional abstract space. If a particular special element corresponds to an input, it means that the concentration of this artificial chemical substance is driven externally. Apart from that, the sub-

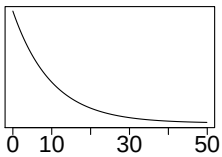


Figure 2: Time scale of product degradation. The product concentrations are in the range $< 0, 1 >$. The intrinsic degradation can increase if a gene is negatively regulated.

stance behaves as any other TF in the system and regulates other genes, with one exception: it cannot directly control the output node of the network. Although this could be beneficial for some problems, we decided to prevent trivial solutions by requiring all signals to be processed by at least one internal node. For all the experiments presented here, at least one external special substance was provided in this manner, having a fixed concentration of "1". This is because it is necessary to have a substance with a non zero concentration to start the GRN activity. For networks evolved to react to changing concentrations of external substances, additional input elements were provided.

If an input element can be seen as a regulatory unit with one gene and zero promoters (its concentration is driven externally), an output element is treated as a regulatory unit with only one promoter and a gene that does not code for a TF. The concentration of the output gene product is thus a clearly defined exit point for all information processing in the system, even though the fact that connections between the output node and the internal nodes are not permitted is expected to have a minor detrimental effect on evolvability. Only one output was allowed.

Genetic algorithm

Genetic operators can act on the level of single elements or multiple elements. On the level of single elements, particular fields can be mutated, changing element type, sign bit, or disturbing the coordinates of an associated point in space. Single or multiple elements can be deleted or duplicated. A series of duplications and deletions can lead to changes in the order of the elements. Changes in the order of promoters within a regulatory unit are neutral, the same goes for the changes in the order of genes. Changing the order of regulatory units does not lead to changes in the topology of the network so it is also neutral. Any type change is permitted. In particular, new input and output elements can be created from other elements (genes, promoters) when the type field of an element is changed by mutation. Type mutations can in principle lead to the loss of inputs or outputs. Obviously, in the experiments described here, such loss would be highly deleterious.

The results shown in this work were obtained using a fairly standard genetic algorithm with a population size of 300, elitism, tournament selection, and multipoint crossover for sexual reproduction (for 30% of the individuals in each generation). Evolutionary runs were initiated with individuals consisting of 5 randomly created regulatory units. The runs were terminated after no improvement over the last 500 generations was detected (typically, after $2500 - 10000$ generations). Shorter runs would often indicate lower evolvability (genetic algorithm stuck in a local optimum rather than continuously improving the network).

Fitness function

The target for evolution was to obtain desired expression patterns as a response to particular input signals. A straightforward approach would be to aim to minimize the difference between the desired (d_t) and obtained (o_t) expression levels over time: $\sum_t |o_t - d_t|$. However, this often lead us to unsatisfying, suboptimal solutions. This is because many of the target patterns require keeping output product expression at 0 for some time, so lack of expression during the whole time results in higher fitness than, for example, a pattern that is shifted but otherwise correct. Once such trivial solution is reached, little can be improved by evolution: there is no regulation that can be fine tuned. We alleviated this problem by including the terms that give higher weight for correctly expressing output product when its concentration is expected to be higher and for the correct number of oscillations in periodic expression patterns:

$$\sum_{t=p}^{L} |o_t - d_t|(1 + kd_t)\frac{1}{1+S} \quad (1)$$

where L is the number of GRN simulation steps (between 600 and 1000 clock ticks, depending on the experiment), and k increases the weight of properly expressed high concentrations ($k = 2$ was used). Parameter p ("propagation time") allows to set the number of simulation steps after which the activity of the output is evaluated. Because some time is needed to build up TF concentrations, it is not reasonable to penalize the network whatever its activity during this time. Propagation time was set to 50 clock ticks: this is a rough estimate of the time needed to form a response. The last term promotes evolution of oscillatory patterns. S was set to 1 when the desired number of oscillations was obtained or to 0 when there was no oscillations or too many (more than twice the desired number). Imperfect matches resulted in intermediate values. To keep the matters simple, the number of events when the expression crosses the level of 0.5 was counted (the events when $d_{t-10} < 0.5$ and $d_t \geq 0.5$ or $d_{t-10} \geq 0.5$ and $d_t < 0.5$). The minimum distance between countable events was set to 10 clock ticks to prevent trivial fluctuations around 0.5. Inclusion of this term in the error function promotes the correct number of oscillations from the very beginning, even if not timed correctly.

Calculated error was further normalized, so that a perfect match in expression pattern would result in individual scoring 0 and the worst possible would score 1. For experiments where multiple training pairs were used, the final fitness would be an average of every test case.

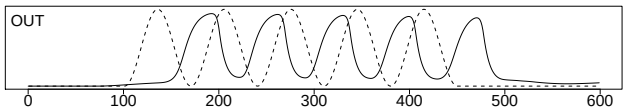


Figure 3: Behaviour of an evolved network that gives a sine wave expression pattern lasting for five periods (the best network in 10 runs); dashed line: the desired response.

Results

Internally induced oscillations

We have first analysed if our system allows for evolution of networks in which an output product level oscillates. Oscillating gene expression has been previously investigated in somewhat similar artificial GRN models (Kuo et al., 2004; Knabe et al., 2006). This task can be made easier by providing the network with a periodically changing input of the same frequency as the target. However, no such input was made available in our experiments: the only external signal was a special product with a constant maximum concentration, so the obtained dynamics was internally induced.

It proved very easy to evolve oscillating expression with almost perfect match to the target pattern (sine waveforms) in a large range of frequencies and amplitudes. The oscillations were stable: they persisted also when the number of simulation steps was increased beyond the network lifespan used at the evaluation stage during evolution.

In a more challenging task, the target was a sine wave starting at a certain time point and ending after 5 periods. The oscillations in the best networks found in 9 independent runs out of 10 had proper frequency but did not terminate. Only in one run a good solution was obtained (Fig. 3), even though the phase of the output signal does not match the target phase. This is penalized by the error function, but the solution is rewarded because the number of pulses is correct (Eq. 1). Perhaps the difference in fitness between a solution in which oscillations terminate and a solution in which they do not is too small and this is why most runs got stuck in a local minimum. If so, simple extension of the lifespan beyond 600 clock ticks would improve evolvability.

Doubling the input frequency

Apart from the task described above, all the others involved processing continuously changing input signals. In the first such task, the networks were expected to double the frequency of the input oscillations (sine wave). Three training inputs were provided at the evaluation stage in the GA: two sinusoidal curves with different frequencies and an input in which the signal was kept at 0 (requiring an empty response). The "no signal" input was included to facilitate emergence of solutions that are active only when external signal is present.

In 10 out of 10 runs the evolved networks displayed the correct behaviour for the training set. Fig. 4ab shows the behaviour of the best network obtained. The solutions were general: intermediate frequencies were also doubled. Even very low frequencies posed no problem (Fig. 4c, note that the time scale is different in different panels). Indeed, for the best individuals we were not able to find a frequency that would be too low to elicit the proper response. Generalizing to frequencies above the range in the training set proved more challenging. The networks did not behave as desired when the frequency was increased more than about 40% (Fig. 4d); interestingly, the best GRN in an experiment in which the frequencies in the training examples were two times lower had about the same relative upper limit.

The behaviour of the best GRN was tested using an input pattern in which frequency changed multiple times (in the training patterns, frequency was constant). The network showed correct behaviour: matching the output frequency to the input frequency (not shown). However, less general solutions were obtained in some runs: these GRNs would lock their outputs to the frequency present at the beginning of a complex input pattern.

It is difficult to analyse how exactly the output of the best GRN is calculated because of the high density of the networks, about 0.5-0.6 (30-50 regulatory units linked with about 1000 edges, encoded with roughly 250 genetic elements). However, a hint on inner mechanics can be obtained by replacing the sinusoidal input with a trapezoid waveform and changing its duty cycle. It can be seen (Fig. 4e) that a spike of the output expression is generated for each raising and each falling edge in the input. This suggests that the poor generalization for higher frequencies may result from the fact that the rate of output product accumulation and degradation is adjusted to the rates used in the training set. If so, concentrations will increase and decrease too fast when the frequency is low; indeed, this can be observed in Fig. 4c).

Low pass frequency filter

Filtering input frequency is a problem well suited for regulatory networks: limited speed of accumulation and degradation of TFs will work as an RC circuit. In this task the networks were expected to regenerate in the output the frequency of the input sinusoid, but only if this frequency was below a certain threshold. Five inputs were provided in the training set: two with frequencies below the threshold, two with frequencies above it, plus the "no signal" input which was again expected to give no output signal. It was easy to obtain networks with correct behaviour that generalized for frequencies higher and lower than those in the training set. However, providing these network with a sum of two sinusoids with only one frequency below the threshold (an example of such input is provided in Fig. 5cd) would result in no output signal. This suggests that these networks simply detected the high rising slope in the input and blocked the output if it was too high.

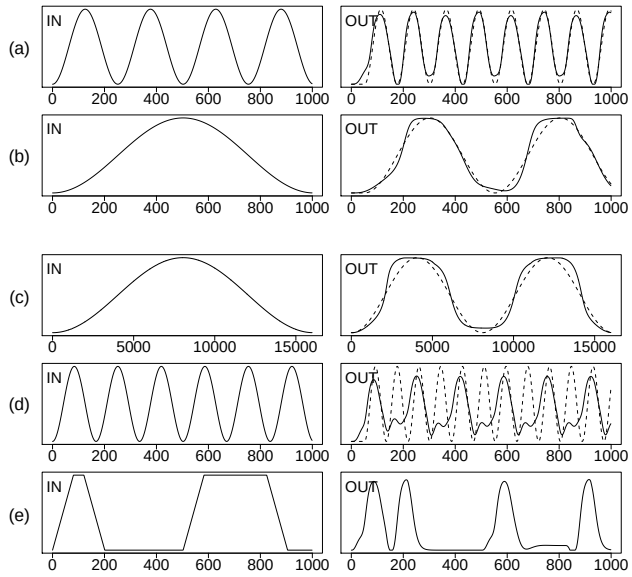


Figure 4: Behaviour of the network evolved to double the frequency of the input signals (the best solution in 10 evolutionary runs, obtained after 6191 generations): (ab) the response for the inputs in the training set (the correct response for the "no signal" input is not shown), (c) this network behaves correctly for an input with much lower frequency than in the training set (note that the time scale was changed), but fails to generalize for inputs with slightly higher frequency (d), the response for the signal in panel (e) hints on the way in which the output is calculated. Dashed lines in (a-d): the desired ideal response.

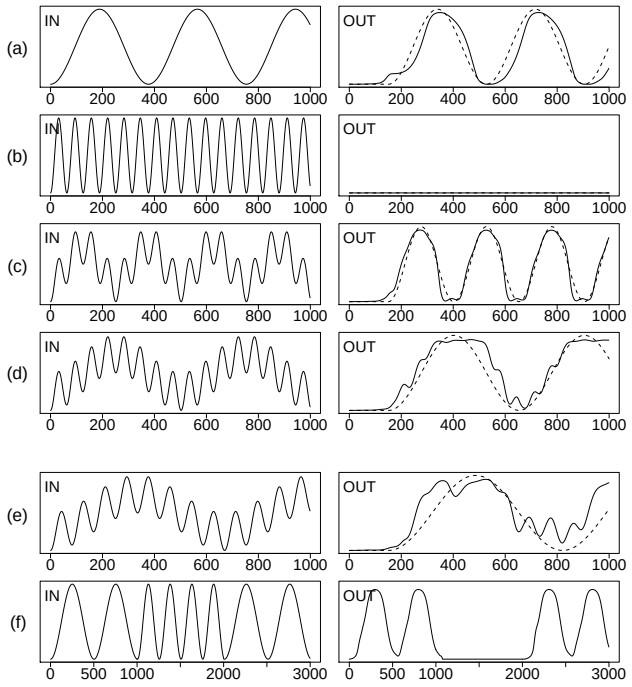


Figure 5: Behaviour of a GRN (the best individual in 10 evolutionary runs, obtained in generation 8839) acting as a low pass filter for the inputs in the training set (a-d; only half of the training examples is shown) and the inputs for which the network was not evaluated during the genetic algorithm (ef). The dashed lines correspond to the desired response.

.

To improve generality of the solutions, we have added such inputs to the training set, requiring the network to filter out just the higher frequency component. Fig. 5e shows the behaviour of a network that correctly if imperfectly filters the high frequency component even for an input not in the training set. This network shows correct behaviour also when another input not in the training set was used (Fig. 5f), adjusting "on the fly" the output signal to the changing frequency in the input. However, such behaviour was observed for the best GRNs only in some of the runs. The best networks in other runs failed to generalize and locked to the frequency present at the beginning of a complex input pattern. This is similar to what was observed in the previous task.

Doubling the pulse length

In the tasks described above, obtaining the solution did not require the explicit memory of the input signal. This is not the case for the task in which the networks were expected to respond with a square pulse twice the length of the square pulse in the input after 50 simulation steps. Three input patterns plus the "no signal" input were used in the training set (Fig. 6a-c). Good solutions were obtained in all 10 evolutionary runs. The best network (Fig. 6a-c) behaved correctly also when the square pulses in the inputs occurred at different times than in the inputs used in the training set. It also behaved as expected when the input pattern consisted of subsequent square pulses.

Good generalization was observed for pulses with other (intermediate) lengths than the pulses in the training set. Pulses up to 50% shorter (Fig. 6d-f) than the shortest training pulse gave the desired response, but pulses longer than the longest training pulse gave responses shorter than desired (Fig. 6e), exposing leaky nature of the GRN-based memory. When the pulses in the input had half the height of those in the training set (Fig. 6f), the length of the output pulse would be close to that of the input pulse. This suggests that the network acts as a simple integrator (e.g. by slowly building up some concentrations) instead of reacting to raising and falling edge of the input signal like frequency doubling networks.

When the networks were required to output a square pulse with doubled length after 300 time steps instead of 50, the behaviours were less accurate, though proper generalization was still observed. The average value of error function (considering only the best individuals in each independent run

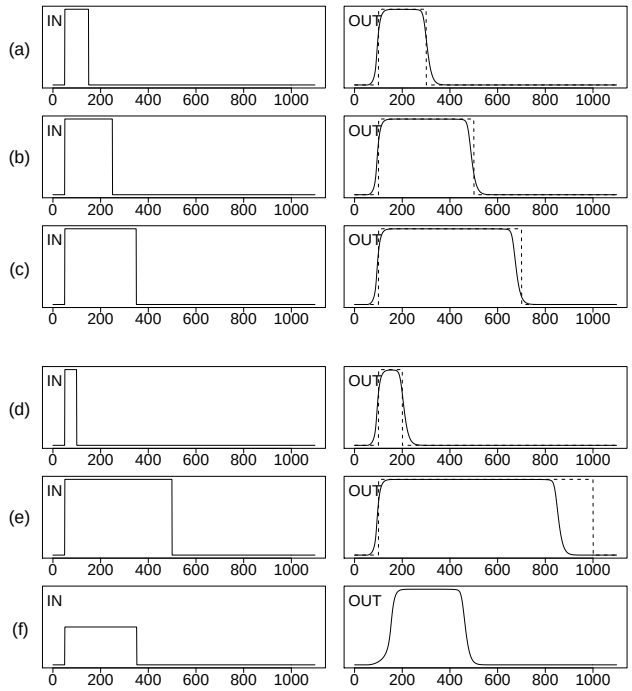


Figure 6: Behaviour of the network evolved to double the input pulse length (the best individual in 10 evolutionary runs, obtained in generation 7295): (a-c) the responses for the inputs in the training set (the response to the "no signal" input was not shown) and (d-f) for the inputs used when testing for generality. Dashed lines correspond to the desired ideal response.

out of 10) was worse: 0.054 for 300 steps vs. 0.017 for 50. The values were also more variable (standard deviation was 0.027 and 0.002, respectively). This further demonstrates the leaky nature of evolved GRN-based memories: the longer the networks have to store the information, the more degraded it becomes.

Doubling the number of input pulses

From the biological point of view, the GRNs discussed thus far could be seen as responding to continuously raising and falling concentration of chemical substance (pulses in the input). What was relevant was the frequency or the length of the pulses. In the next two problems, the number of pulses will be important. The first task, doubling the number of pulses, can be seen as more difficult than the previous problem. The response still requires performing multiplication, but the number of subsequent pulses needs to be counted, not the pulse length.

Fig. 7a-c shows that the best network obtained in 10 runs correctly doubles the number of pulses in the training set inputs when this number is one or two. The solution when the expected number of subsequent oscillations is six is almost correct. However, the generalization is imperfect: seven in-

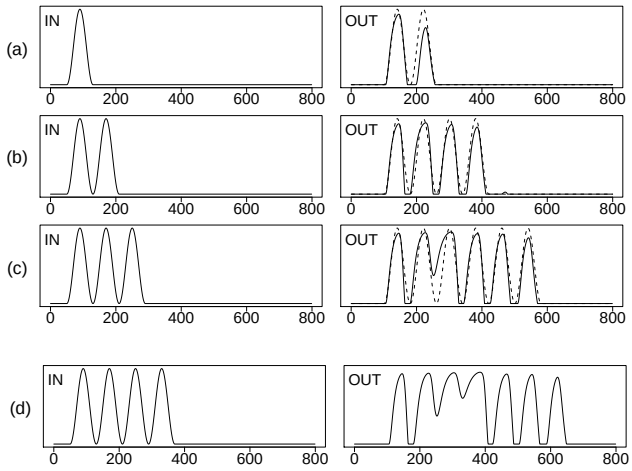


Figure 7: Behaviour of a GRN that doubles the number of spikes (the best individual in 10 evolutionary runs, obtained in generation 2794): (a-c) the network behaves correctly or almost correctly for the training set input, but (d) responds with less spikes than expected when the generality of the solution is tested with a higher number of spikes in the input.

stead of eight pulses for four pulses in the input (Fig. 7d), a response shorter than expected. This reminds the behaviour of GRNs evolved to double pulse lengths when presented with input pulses longer than the longest in the training set.

Integrating information from two separate signals: counting pulses

The experiment described above indicates that a task that involves processing concentration pulses allows to approach the limits of our system in terms of searching for networks with desired signal processing properties. To make the task even more difficult, the networks were required to process signals from two inputs instead of one. The task was to respond with the number of output pulses equal to the number of pulses on both inputs within a certain time window (see Fig. 8a-e for the training set). No response was expected when no input pulses were present in the pattern. Fig. 8 shows the behaviour of the best GRN in 10 runs. This network is able not only to count correctly the pulses in the training set but is also general enough to work in a continuous manner (Fig. 8f).

Modifying the system time step

Product accumulation and degradation in our system is simulated in discrete steps. Changes in concentration are computed with every iteration with a time step $dt = 0.1$. The step size is a compromise between accuracy and computation cost. In principle, it would be possible for some of the evolved networks to exploit inaccuracies that would occur if some concentrations were to change rapidly due to over-regulation and wrongly chosen dt. To test if this is an issue

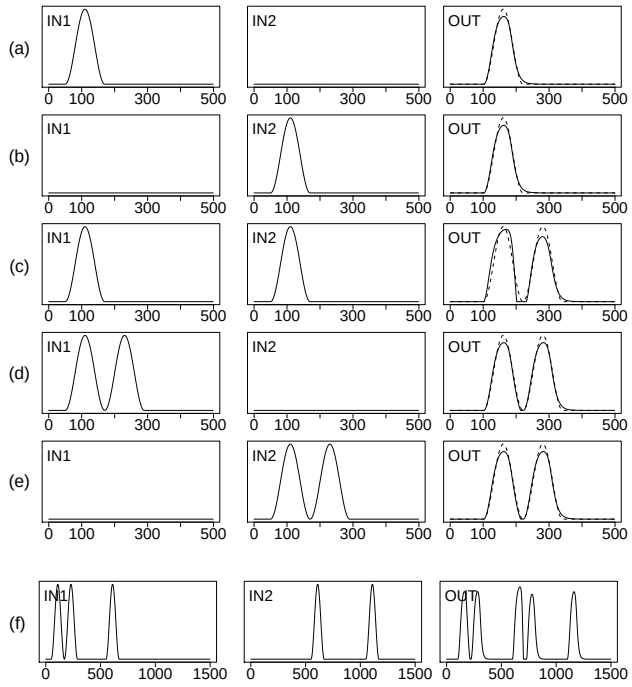


Figure 8: Behaviour of the GRN evolved to count the pulses in two inputs (the best individual in 10 evolutionary runs, obtained in generation 2168): (a-e) the network gives an expected output for the the inputs in the training set and the (f) inputs used to test for generality.

we decreased dt by an order of magnitude and increased 10-fold the number of simulation steps. This increased simulation accuracy but did not affect the behaviour of any of the networks discussed above.

The importance of continuous TF accumulation/degradation

In the GRN model used here the TF concentration at a particular time point is determined by its synthesis and degradation rates and its concentration at the previous time step. In order to test if this GRN property is important for signal processing tasks, we have modified the model so that the gene expression was determined only by the activation of associated promoters in the previous time step. More precisely, the function $f_A(A)$, instead of being treated as current product synthesis level (with the range $< -1, 1 >$), would be shifted right and scaled to $< 0, 1 >$ so that it could be treated as a new expression level for the given time step. This allows genes to change its activity instantly. In this model GRNs behave similarly to recurrent networks of perceptron-like neurons (similar regulatory networks were used by us Joachimczak and Wróbel (2008) and other researchers, e.g. Eggenberger (1997). To see if this change affects evolvability, we compared the average fitness for the best individuals in 10 runs using the problem of doubling

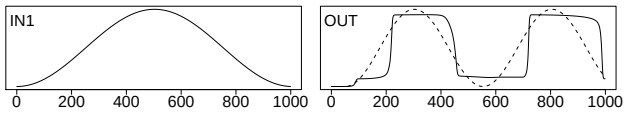


Figure 9: The best individual obtained in 10 evolutionary runs using a modified model in which product built-up and degradation is not simulated (response to one of the training signals is shown).

the input frequency. The behaviour of the best individual for a non-continuous model Fig. 9 can be compared with that observed in Fig. 4. Even though a good solution was found, the evolvability itself was clearly worse. Average error for 10 runs with a modified model was 0.075 (sd: 0.025). For the model with continuous TF synthesis/degradation the error was 0.026 (sd: 0.005).

Discussion

The goal of this work was to investigate in a qualitative and exploratory manner the possibility to evolve artificial GRN that can generate or process continuous signals provided as externally driven concentrations of chemical substances. We have tested if the way we have formulated the encoding of the structure of the networks in a linear genome and the genetic algorithm allows for evolvability in several problems of various difficulty. Several attempts have been made previously by us and other researchers to employ artificial GRNs for various tasks (such as development). It is thus interesting to investigate what kind of information processing can be performed by single cells equipped with such networks.

In general, given enough simulation steps, artificial GRNs can be expected to be similar to perceptron-like artificial neural networks (ANNs) with recurrent connections in terms of computational properties, even though the biological inspiration is different. Perhaps the most important difference between the GRN model used here and commonly used ANN models is that here the state of a regulatory node, represented by the concentration of associated products (transcriptional factors) is influenced by the rate of product synthesis and degradation. This limits the response time of the network. On the other hand, smoothness of gene expression provides an advantage for generating gradually changing outputs, such as sine waves (compare Fig. 4b and Fig. 9). One could also expect that such inherent dynamics of each node could be exploited by biological GRNs when dealing with noisy external signals and with the inherent noisiness of gene expression itself. Obviously, "no free lunch" theorem applies: GRNs may provide an advantage in a certain class of problems, but one should not expect them to universally outperform other approaches.

In particular, computations that required counting pulses of input substance concentration proved more difficult than other tasks (which also involved simple mathematical cal-

culations and memory). Processing information encoded in pulses is superficially similar to information processing in spiking neural networks. However, in GRN-based systems the pulses result from simulated product accumulation followed by degradation not by simulation of ion transport through the membrane, often extremely simplified (so that a spike results when a threshold potential is reached). It is reasonable to assume that this kind of information encoding is far from optimal for processing signals with regulatory networks. In other words, problems that require pulse counting can help to find the limit of what can be evolved using GRN-based systems such as ours.

Introducing more realistic molecular dynamics could make evolving artificial GRN models a useful tool for obtaining synthetic regulatory networks (see e.g. Friedland et al., 2009; Elowitz and Leibler, 2000). Such networks might find applications for example in intelligent delivery of therapeutic chemical substances (small molecules, proteins, regulatory RNAs), regulated by external signals. Artificial evolution would allow to design such networks and optimize them by various criteria, such as the number of regulatory elements and genes or robustness to noise.

The evolvability in signal processing tasks could be also improved by changes in the error function or reformulation of the tasks themselves. For example, it would probably help to look for the best match of the output expression pattern within a certain range of allowable response times instead of requiring the pattern to appear after a predefined response delay.

Although it would be very interesting to further explore the areas hinted above, the next step in our work will be to investigate the statistical properties of evolving artificial GRNs and to employ the model described here in other control problems, for example, animat navigation.

Acknowledgements

This work was supported by the Polish Ministry of Science and Education (project N519 384236). The computational resources used in this work were obtained thanks also to the support of the project N303 291234, the Tri-city Academic Computer Centre (TASK) and the Interdisciplinary Centre for Molecular and Mathematical Modelling (ICM, University of Warsaw; project G33-8).

References

Andersen, T., Newman, R., and Otter, T. (2009). Shape homeostasis in virtual embryos. *Artificial Life*, 15(2):161–183.

Banzhaf, W. (2003). On the dynamics of an artificial regulatory network. In *Proceedings of the 7th European Conference (ECAL 2003) Dortmund, Germany, September 14-17, 2003*, Lecture Notes in Artificial Intelligence, pages 217–227. Springer Berlin / Heidelberg.

Bentley, P. J. (2004). Adaptive fractal gene regulatory networks for robot control. In *Workshop on Regeneration and Learning in Developmental Systems in the Genetic and Evolutionary Computation Conference (GECCO 2004)*.

Eggenberger, P. (1997). Evolving morphologies of simulated 3D organisms based on differential gene expression. In *Proceedings of the Fourth European Conference on Artificial Life*, pages 205–213, Cambridge, MA. MIT Press.

Elowitz, M. B. and Leibler, S. (2000). A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(6767):335–338.

Friedland, A. E., Lu, T. K., Wang, X., Shi, D., Church, G., and Collins, J. J. (2009). Synthetic gene networks that count. *Science*, 324(5931):1199–1202.

Joachimczak, M. and Wróbel, B. (2008). Evo-devo *in silico*: a model of a gene network regulating multicellular development in 3D space with artificial physics. In *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, pages 297–304. MIT Press, Cambridge, MA.

Joachimczak, M. and Wróbel, B. (2009). Evolution of the morphology and patterning of artificial embryos: scaling the tricolour problem to the third dimension. In *Proceedings of 10th European Conference on Artificial Life (ECAL 2009)*. Springer.

Knabe, J. F., Nehaniv, C. L., Schilstra, M. J., and Quick, T. (2006). Evolving biological clocks using genetic regulatory networks. In *Artificial Life X: Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, pages 15–21. MIT Press/Bradford Books.

Kuo, D., P., Banzhaf, W., and Leier, A. (2006). Network topology and the evolution of dynamics in an artificial genetic regulatory network model created by whole genome duplication and divergence. *BioSystems*, 85(3):177–200.

Kuo, D. P., Leier, A., and Banzhaf, W. (2004). Evolving dynamics in an artificial regulatory network model. In *Parallel Problem Solving from Nature - PPSN VIII*, volume 3242 of *Lecture Notes in Computer Science*, pages 571–580. Springer Berlin / Heidelberg.

Nicolau, M. and Schoenauer, M. (2009). On the evolution of scale-free topologies with a gene regulatory network model. *BioSystems*, 98(3):137–148.

Quick, T., Nehaniv, C. L., Dautenhahn, K., and Roberts, G. (2003). Evolving embodied genetic regulatory network-driven control systems. In *Advances in Artificial Life*, pages 266–277.

Reil, T. (1999). Dynamics of gene expression in an artificial genome - implications for biological and artificial ontogeny. In *Proceedings of the 5th European Conference on Artificial Life*, volume 1674 of *Lecture Notes In Computer Science*, pages 457–466, London, UK. Springer-Verlag.

Schramm, L., Jin, Y., and Sendhoff, B. (2009). Emerged coupling of motor control and morphological development in evolution of multi-cellular animats. In Kampis, G. and Szathmáry, E., editors, *Proceedings of 10th European Conference on Artificial Life (ECAL 2009)*. Springer.

Taylor, T. (2004). A genetic regulatory network-inspired real-time controller for a group of underwater robots. In *Proceedings of the Eighth Conference on Intelligent Autonomous Systems (IAS-8)*, pages 403–412.

Minimal Cognition and Physical Intelligence

Behavioral Metabolution: Metabolism Based Behavior Enables New Forms of Adaptation and Evolution

Matthew D. Egbert[1], Xabier E. Barandiaran[1] and Ezequiel A. Di Paolo[1,2]

[1]Center for Computational Neuroscience and Robotics, University of Sussex, Brighton, U.K.
[2]Ikerbasque: Basque Foundation for Science
Department of Logic and Philosophy of Science, University of the Basque Country, Spain.
mde@matthewegbert.com, xabier.academic@barandiaran.net, ezequiel@sussex.ac.uk

Abstract

Both metabolism and behavior play a key role in biological theory and artificial life modelling. Yet, despite their centrality there has been very little exploration of the relationship between these concepts and almost no exploration of how the interaction between the two could impact on evolution or instantiate alternative mechanisms for evolutionary processes. We present a simulation model of bacteria capable of metabolism-based chemotaxis: a minimal metabolic system capable of modulating behavior by influencing the probability of flagellar rotation (like in *E. coli* chemotaxis). We perform two illustrative experiments. In the first, the incorporation of a chemical compound into metabolism qualitatively improves the chemotactic strategy. In the second, an encounter with a specific chemical compound leads to a reaction that opens up a new metabolic pathway while automatically regulating chemotaxis towards that same compound. Both experiments illustrate the adaptive potential of metabolism-based behavior and can be used to explore the idea of "Behavioral Metabolution," a co-evolutionary synergy between behavior and metabolism. We abstract some principles of behavioral metabolution and discuss its application to early prebiotic evolution.

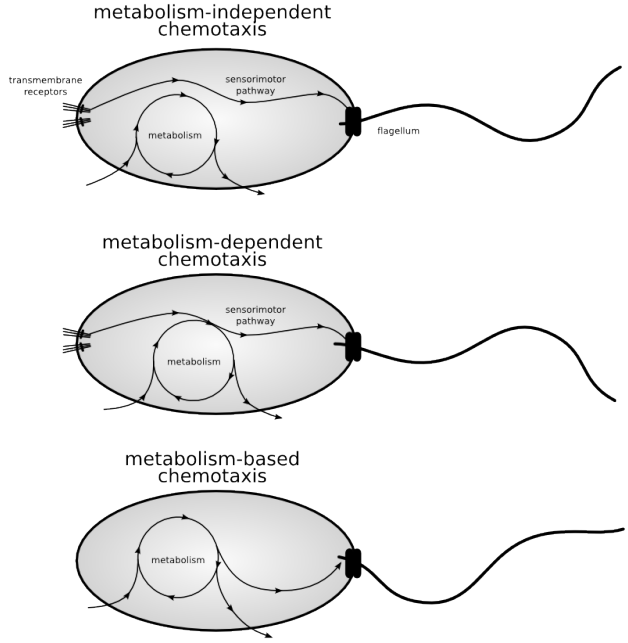


Figure 1: Three different relationships between metabolism and chemotaxis. Arrows indicate only short-term dynamical influence between processes. See text for details.

Introduction: metabolism and behavior

There is a long tradition in artificial life of investigating the origins and essence of life through the study of metabolism. Metabolism is understood as the far from thermodynamic equilibrium organization of chemical networks that produce and sustain their components from available energetic and material resources (Ganti, 1975; Kauffman and Farmer, 1986; Morowitz, 1999). Recent work on protocellular systems (Rasmussen et al., 2008) has re-framed research on metabolism within the framework of minimal forms of (proto)cellular compartments capable of self-maintenance.

Rarely is the environment of such early-life scenarios considered to be controlled or selected by a behaving or moving proto-life-form. However, recent artificial models of self-moving protocellular (autopoietic) systems (Suzuki and Ikegami, 2009; Egbert and Di Paolo, 2009) and real, self-propelled chemical systems (Toyota et al., 2009) suggest that even extremely simple forms of proto-life may have been capable of selectively modulating their environment through behavior.

In parallel to the omission of behavior in the study of the origin of life, studies of minimal adaptive behavior have almost completely ignored the role of metabolism as sustaining or modulating behavioral patterns. In particular, research on bacterial chemotaxis (the paradigmatic case of "minimal adaptive behavior") has long proceeded under the assumption that behavior generating mechanisms operate in an metabolism independent manner (i.e., while behavior subserves metabolic survival, sensorimotor pathways are not influenced by short-term metabolic dynamics). This assump-

tion can be traced back to the pioneering work of Julius Adler (1969) and has since remained almost unquestioned even in the most detailed and systematic simulation models of bacterial chemotaxis (Bray et al., 2007). It is, of course, not the only possible relationship between metabolism and chemotaxis. Figure 1 indicates three different possibilities for this relationship, *independent*, *dependent* (mechanisms in a sensorimotor loop are created by the metabolism) and *based* (metabolism itself modulates behavior). Recently, the growing evidence for metabolism-dependent chemotaxis in many bacteria (Alexandre and Zhulin, 2001), including *E. coli*, has attracted renewed attention to the interplay between metabolism and behavior.

In short, the interaction between behavior and metabolism remains currently under-explored even though empirical and modelling work has begun to address its possible implications. In particular, an aspect that deserves further examination is the effect of this interaction on early (and not so early) evolutionary dynamics. The goal of this paper is to present a model that investigates some potential implications of the interaction between metabolism and behavior in both directions (behavior $\rightarrow$ metabolism and metabolism $\rightarrow$ behavior) as well as the potential impact of these interactions upon evolutionary processes.

We shall first present a model of *metabolism-based chemotaxis* consisting of a minimal metabolism coupled to a simplified motor system inspired by *E. coli*. We use this model to demonstrate, through two experiments, that: 1) metabolism can modulate behavior in an adaptive manner, 2) behavior can change the metabolism by changing the environment in which it exists and, 3) changes in metabolism can produce new types of behavioral patterns. Next, we abstract away some general principles and implications of metabolism-based chemotaxis. Finally, we conclude with some discussion regarding the evolutionary dimension of metabolism-based chemotaxis, what we term "behavioral metabolution", and its potential application to the question of early evolution of life.

Metabolism-based chemotaxis, the model

We consider metabolism as the self-production of a chemical network through the transformation (by the network) of available energetic and material resources into constituents of the network. This process is most simply realized through an auto-catalytic reaction whereby energetic and material resources (E and M respectively) are transformed by network constituent C into more C and a low energy waste V thus: $M + E \xrightarrow{C} C + 2V$. This single reaction may be understood as a higher level abstract representation of a whole network of processes, considering that the essence of metabolism is that of an auto-catalytic network. To capture the requirement of far-from-thermodynamic equilibrium, C and V are considered thermodynamically unstable and degrade rapidly. Their continued presence is therefore only possible through a *dynamic* equilibrium of degradation countered by production. We label this reaction the "core metabolism" and expose it to various other reactants in different experiments. Table 1 and Figure 2 show all of the chemical reactions that can be active in the bacteria simulated in our model. The upper-left square indicates the core metabolism described in this section. The other pathways are described in the experiments and results section.

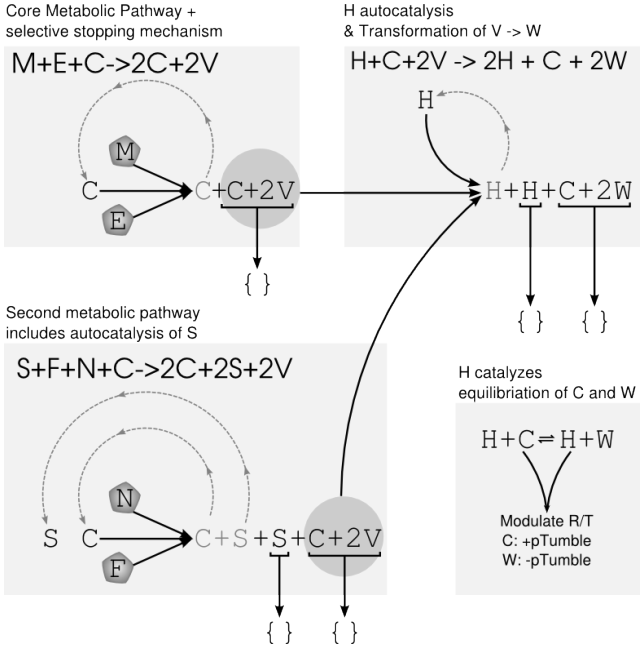


Figure 2: Reactions grouped conceptually by their 'role' in the model. Resources are surrounded by pentagons. Autocatalytic reactions are indicated by circular paths. Degradation of reactants is indicated by an arrow to the empty set.

The metabolic dynamics are described by the differential equations in Table 2. These equations include some reactants that are only used in some of our experimental scenarios and are explained later in the text. The rate con-

#	reactants		products	k_f	k_b
0:	M + E + C	$\rightleftharpoons$	2C + 2V	0.61	≈ 0
1:	H + C	$\rightleftharpoons$	H + W	0.006	0.006
2:	H + C + 2V	$\rightleftharpoons$	2H + C + 2W	0.37	≈ 0
3:	C + 2V	$\rightarrow$	{}	0.006	n/a
4:	C + 2W	$\rightarrow$	{}	0.006	n/a
5:	H	$\rightarrow$	{}	0.02	n/a
6:	S	$\rightarrow$	{}	0.0001	n/a
7:	S + F + N + C	$\rightleftharpoons$	2C + 2S + 2V	0.99	≈ 0

Table 1: A list of the chemical reactions in each simulated metabolism. Also indicated are the reaction rates (forward and backward). These rates are referred to in Table 2.

$$
\begin{aligned}
dE/dt &= -k_{f0}EMC + k_{b0}C^2V^2/4 + k_d[E](\mathbf{x}) \\
dM/dt &= -k_{f0}EMC + k_{b0}C^2V^2/4 + k_d[M](\mathbf{x}) \\
dC/dt &= -k_{f0}EMC + k_{b0}C^2V^2/4 \\
&\quad -2k_{b0}C^2V^2/4 + 2k_{f0}EMC \\
&\quad -k_{f1}CH + k_{b1}HW \\
&\quad -k_{f3}CV^2/2 - k_{f4}CW^2/2 \\
&\quad -k_{f7}CFNS + k_{b7}C^2V^2S^2/6 \\
&\quad -2k_{b7}C^2V^2S^2/6 + 2k_{f7}CFNS \\
dV/dt &= -2k_{b0}C^2V^2/4 + 2k_{f0}EMC \\
&\quad -2k_{f2}CHV^2/2 + 2k_{b2}CH^2W^2/4 \\
&\quad -2k_{f3}CV^2/2 \\
&\quad -2k_{b7}C^2V^2S^2/6 + 2k_{f7}CFNS \\
dW/dt &= -k_{b1}HW + k_{f1}CH \\
&\quad -2k_{b2}CH^2W^2/4 + 2k_{f2}CHV^2/2 \\
&\quad -2k_{f4}CW^2/2 \\
dH/dt &= -k_{f2}CHV^2/2 + k_{b2}CH^2W^2/4 \\
&\quad -2k_{b2}CH^2W^2/4 + 2k_{f2}CHV^2/2 - k_{f5}H \\
dF/dt &= -k_{f7}CFNS + k_{b7}C^2V^2S^2/6 + k_d[F](\mathbf{x}) \\
dN/dt &= -k_{f7}CFNS + k_{b7}C^2V^2S^2/6 + k_d[N](\mathbf{x}) \\
dS/dt &= -k_{f6}S - k_{f7}CFNS + k_{b7}C^2V^2S^2/6 \\
&\quad -2k_{b7}C^2V^2S^2/6 + 2k_{f7}CFNS + k_d[S](\mathbf{x})
\end{aligned}
$$

Table 2: Differential equations specifying how chemical concentrations change within each simulated bacterium (excluding influence of the environment). k_{fn} and k_{bn} represent the reaction rate constants for the nth reaction in the forward or backward direction. $[\rho](\mathbf{x})$ represents the local environmental concentration of the resource ρ.

stants (k_{fn} and k_{bn}) in the differential equations have been determined by assigning *free-energies* to each reactant and *activation-energies* for each reaction such that the system adhered to the constraints defined in our definition of a minimal metabolism. Given chemical free-energies and reaction activation-energies, reaction rates can be calculated according to $k_f = \exp(A)$ and $k_b = \exp(A + R - P)$ which indicate the reaction rate for a forward (exergonic) reactions and backward (endergonic) reactions respectively. A represents the activation energy of the reaction and R and P represent the combined energy levels of the reactants and the products respectively of the reaction. Figure 3 indicates why the forward and backward equations are different. This method of determining reaction rates allows the exploration of abstract chemistries while remaining congruent with the 2^{nd} law of thermodynamics.

Resources encountered in the environment diffuse into bacteria at a rate proportional to the local concentration of the environmental resource. The rate constant for this diffusion, $k_d = 0.04$, is the same for all resources.

The chemical reactions are simulated as occurring in a compartment surrounded by a membrane that includes a set of flagella. The clockwise and counter-clockwise flagellar rotation is determined by the relation between the concentrations of C and W compounds. In analogy to the working of flagellar rotation in *E. coli* chemotaxis, when the overall movement of flagellar rotation is counter-clockwise the bacterium is propelled in straight direction (what is generally called the "running mode"), whereas when flagella rotate clockwise, the bacterium rotates on its axis changing direction randomly ("tumbling mode"). Bacteria are simulated in a 2D square'petri-dish' of 200 units. By default, bacteria are always running, i.e., moving in a straight line in the direction of their orientation, α, thus: $\frac{dx}{dt} = 0.05 \cdot \cos(\alpha)$, $\frac{dy}{dt} = 0.05 \cdot \sin(\alpha)$. A baseline probability of tumbling allows for the random direction to be changed occasionally. Tumbling bacteria remain at the same location, with α changed to a random value selected from a flat distribution between 0 and 2π. The effect of the influence of C and W concentrations on flagellar rotation is abstracted and summarized in the following equation governing the probability of tumbling of the bacteria (i.e. the probability of the bacteria changing direction randomly): $P_{\text{tumble}} = 0.001 * \max(-0.1 + [C]^2 - 0.9[W]^2, 0.01)$.

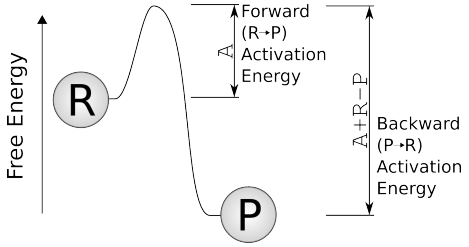


Figure 3: Energy required for a reaction to take place. The line traces the free energy of the reactants as the reaction takes place.

Experiments and results

The goal of these two experiments we now present is to provide a proof of concept of how, in metabolism-based chemotaxis, small changes in metabolism can lead to qualitative changes in behavior (experiment 1) and how behavior can lead to fixation of new metabolic pathways (experiment 2).

E1. Influence of metabolic change in behavior

In this experiment, we demonstrate how a small change in metabolism can lead to a substantial, qualitative difference in behavior. Specifically we demonstrate a scenario whereby one form of chemotaxis (selective-stopping) is transformed into a more sophisticated form (gradient-climbing) through exposure to a new reactant. To do this, we compare two different types of bacteria, placing 100 of each type evenly distributed on a petri dish containing at its center a resource of $M + E$; the concentration of which decays with distance following a Gaussian distribution (indicated in the histograms). The control group starts with only reactant $[C] = 0.5$ which provides a functioning core metabolic pathway. The experimental group is the same as the control except that it starts

with an additional reactant, $[H] = 1.0$. The presence of this chemical produces a self-maintaining gradient-climbing mechanism by enabling reactions 1 and 2 (see Table 1 and Figure 2 top-right and lower-right). These two conditions allow us to examine the differences between bacteria that have not encountered H (control group) and those that have (experimental group).

Figure 4 indicates the behavior of the control group which demonstrates the selective-stopping mechanism accomplishing a simple form of chemotaxis. The histogram at the top indicates the number of bacteria at different distances from the peak resource at the end of the trial, (data taken from 10 trials, each of 100 bacteria). The three plots at the bottom of the figure indicate the spatial distribution of the bacteria in the petri dish at the start, halfway through, and end of a typical trial. The behavior of these bacteria is a simple result of the metabolism and its influence on motion. In the absence of W, the concentration of C will drive the behavior of the bacterium: if the metabolic activity (i.e., the production of C) is high the probability of tumbling will increase and the bacterium will remain in the local area. If C is low the probability of tumbling will decrease and the bacteria will move, still in a random walk, but with increasingly long durations of directional movement until C is produced again (e.g., when the bacterium finds a place where M and E are abundant). The mechanisms resembles the Ashbian principles for adaptation (Ashby, 1952) except that the system is simply altering its relation to the environment, instead of reconfiguring itself internally. In this way, behavior is directly modulated by the rate of metabolic production in a "selective stopping" manner that is beneficial for metabolism: "stay where you are if the metabolism is running sufficiently well, otherwise run". This is the simplest example of what we call *metabolism-based chemotaxis* where the "sensorimotor" pathway is the metabolism itself.

Bacteria with $[H] > 0$ are capable of the the more sophisticated "gradient climbing" strategy (widely found in bacterial chemotaxis) whereby the bacteria are capable of comparing, as they move, the current concentration of a chemical compound with its concentration earlier. To explain how this is accomplished, we must describe the dynamics of the new reactant, H. H is auto-catalytic in the presence of C and V, so once a functioning metabolism encounters H, its concentration will be maintained above 0. In this simulation, H performs two roles. It catalyzes an equilibration between C and W, $(H + C \rightleftharpoons H + W)$ and additionally, in its auto-catalysis, transforms V into W which inhibits tumbling. These equations produce a system that is described conceptually in Figure 6 whereby 1) stoichiometry and reaction rates cause W to change more rapidly than C, 2) W and C tend to equilibriate to equal concentrations, and 3) W inhibits the probability of tumbling and C enhances it. These properties produce an adaptive gradient climbing mechanism (adaptive in the sense used by bacteriologists to describe the ability to adapt to a wide range of base levels of stimulus). It can be seen how in both conditions bacteria approach the resource center but H produces a more efficient result due to its adaptation; as is evident when comparing Figures 4 and 5 where the gradient-climbing bacteria move to the highest concentration of resource, unlike the selective-stoppers that stop when the resources are above a threshold. (In both cases, a secondary peak around a distance of 190 can be observed due to the effect of the petri dish wall).

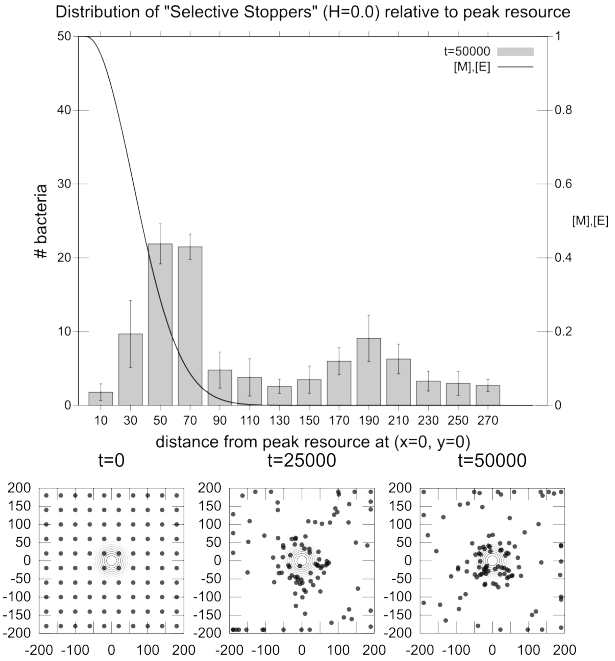


Figure 4: Selective-stopping bacteria distance from peak resource (top) and spatial distribution (bottom).

The experiment shows how changes in the metabolic network of a metabolism-based chemotactic agent can lead to qualitative adaptive changes and improvement on its behavior, through relatively simple means. While moving through its environment, a bacterium can potentially encounter a new component H that is incorporated into the metabolism through its self-catalytic activity and through its capacity to improve the adaptive behavior of the bacterium. The chances of this event happening are enhanced by the self-movement of the bacterium. Note that the specific changes that have occurred here have been designed to make the system as simple to understand as possible, not to suggest that the transformations described have occurred in this way in biology.

E2. Influence of behavioral change on metabolism

In this new experiment we include a second metabolic pathway. In this pathway, energetic and material resources (F

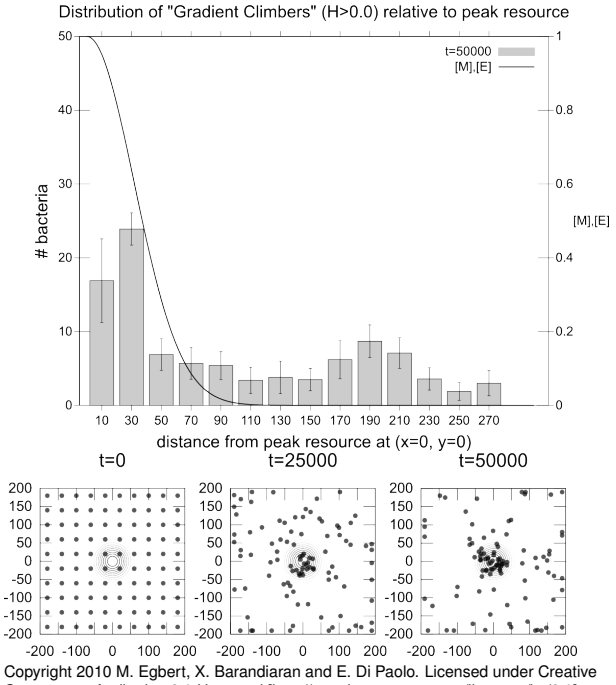


Figure 5: Gradient-climbing bacteria distance from peak resource (top) and spatial distribution (bottom).

and N respectively) are converted into C and V. Like the core metabolic pathway, this is an auto-catalytic production requiring C to be present to occur. However, unlike the core metabolic pathway this reaction is also auto-catalytic with respect to S. This means that S is both produced by the reaction and required for the reaction to occur (see Figure 2 bottom-left).

Bacteria, (initialized with $C = 0.5$, $H = 1.0$ and $S = 0.0$) are placed evenly distributed around a petri dish containing two sources of E and M, located at $(x = -75, y = 0)$ and $(x = 75, y = 0)$. One source of F and N is located at $(x = 0, y = 0)$. There is no S in the environment except within a circle of radius 0.5 around the left peak of resource E and M $(x = -75, y = 0)$, where $[S] = 1.0$.

Figure 7 indicates the distribution of the bacteria over the course of the simulation. The bottom figures are as in Figures 4 and 5, but the histogram now indicates the distribution of bacteria along the x-axis, comparing the distributions of bacteria that have zero and non-zero concentrations of S. Data have been collected at the end of 10 different trials, each of 100 simulated bacteria. As before, at the start of the simulation, the bacteria are evenly distributed around the arena. The gradient climbing mechanism attracts the bacteria to one of the sources of E/M. At this stage, none of the bacteria have any S, so F/N is not metabolizable and has no effect on the behavior of the bacteria as the metabolism based mechanism automatically ignores resources that are

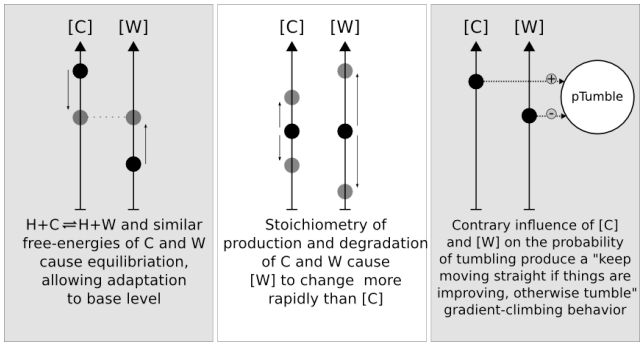


Figure 6: Implementation of gradient climbing mechanism.

irrelevant to the metabolism. As time progresses, bacteria tend to gravitate towards the highest concentrations of E/M, and those that are at the left source have an increasingly high chance of encountering the pocket of S. Those bacteria that come into contact with S become capable of auto-catalyzing S. Their metabolism has been changed and the odds of this change occurring have been significantly influenced by their behavior. Those bacteria with $[S] > 0$ have gained a new metabolic pathway. They are now capable of metabolizing F/N and as time progresses, those bacteria that through their random walk are brought close enough to "taste" F/N, now also climb that gradient. Bacteria that were initially attracted to the right-most source of E/M never encounter S and accordingly never are drawn away from their initial F/N resource source and at the end of the simulation there are in a certain respect two 'species' of bacteria – one that consumes and is attracted to both pairs of resources and one that is only attracted to, and only consumes the original pair.

Discussion: Behavior, metabolism, evolution

The adaptive power of metabolism-based chemotaxis

Adaptive behavior is generally understood and modelled as optimizing some value function or as maintaining essential variables under viability constraints. However, there is generally no reference to the dynamics of the biological organization (e.g., metabolism) that serves as the basis of these viability constraints —see Egbert et al. (2009) for a discussion. When metabolic dynamics are directly coupled to behavior a number of adaptive phenomena come to the surface that generally pass unnoticed due to the typical abstractions made in adaptive-behavior models.

From the previous experiments we can generalize that, despite its simplicity (or perhaps thanks to it), metabolism-based behavior can enable a number of powerful adaptive capacities:

1. The metabolic **consequences of behavior can be eval-**

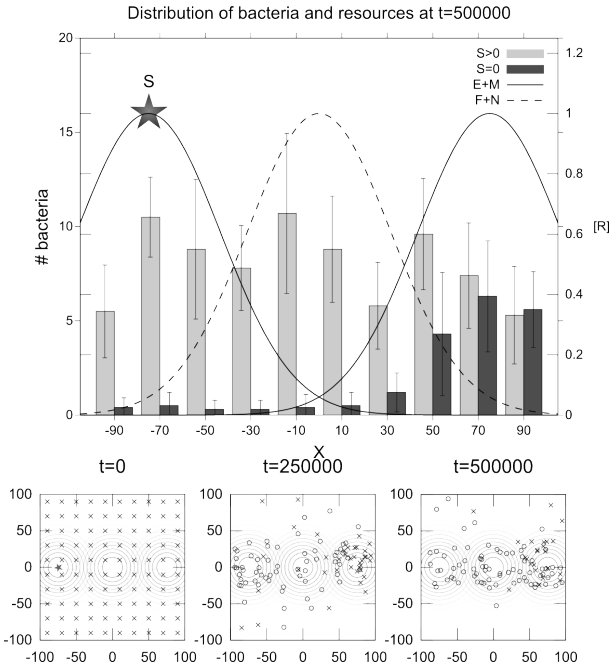


Figure 7: Experiment 2. Bacteria are initially attracted to sources of $M + E$, but those that encounter the metabolic-path-opening reactant S, automatically become also attracted to new resources $N + F$.

uated *online* (i.e., in ontongenetic time and in relatively short timescales) and behavior can be modulated accordingly.

2. Organisms can **adapt not only to the presence of specific chemicals but also to other environmental conditions** (e.g., temperature) that might influence metabolism.

3. Organisms can **adapt not only to changes in the environment, but to changes in their own metabolic organization** by modulating their behavior accordingly.

4. Organisms can **integrate** information from the environment and from within, which means that **behavioral and metabolic processes of adaptation can feed back to each other**.

As a consequence, organisms can adapt (respond appropriately) to various environmental and internal chemical compounds and conditions that were never previously experienced by the individual nor even by any of its ancestors. Note that the system will be attracted to any compound or condition that increases metabolic rate and will be repelled by those that decrease or inhibit metabolism. However, this does not rule out potential cases of maladaptation such as parasitic interactions that override the behavioral mechanism or interactions that increase the short-term rate of production of C but damage metabolism in the long-term by e.g., destroying the membrane.

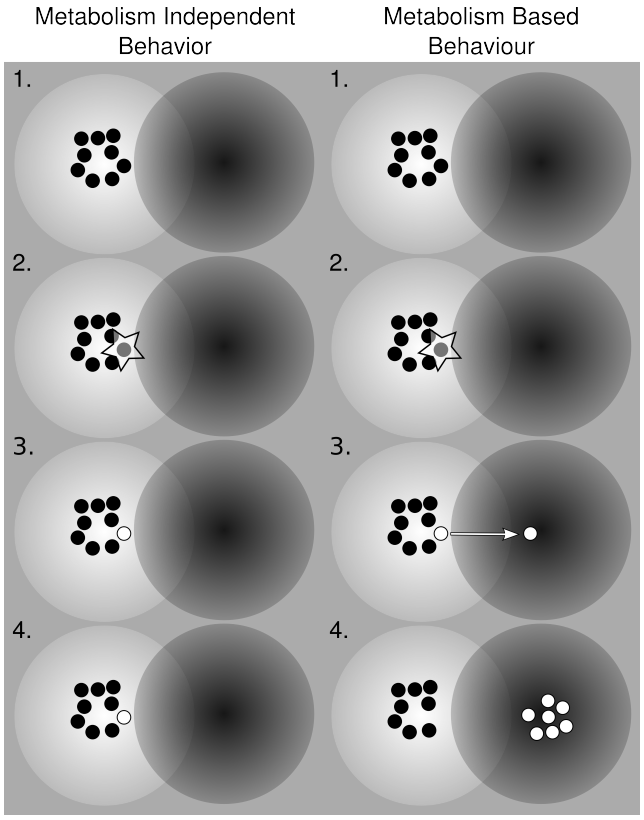


Figure 8: Metabolism-independent and metabolism-dependent responses to a change in organization (represented by a star in frame 2) that allows them to consume a new resource (dark circle).

Behavioral metabolution, the very idea

Not only does metabolism-based behavior unveil a powerful form of adaptation in ontogenetic time, but it also exposes an interesting evolutionary potential. Figure 8 illustrates the case of a mutation (genetic or otherwise inheritable) on metabolic pathways that permits one bacterium to exploit and metabolize a new environmental resource. Metabolism-independent chemotactic agents (left) will remain in place and the benefits of the mutation will pass unnoticed; unless there is an unlikely coincident mutation that makes transmembrane receptors sensitive to the new metabolic source *and* generates attraction to it. Genetic drift dictates that, most probably, such a *potentially* beneficial mutation will be lost since it has no beneficial effect on the bacterium. Metabolism-based chemotactic agents (right), contrarily, will immediately and automatically be attracted to the new resource (for it benefits metabolism) if they are exposed to it. They will benefit from the mutation by in-

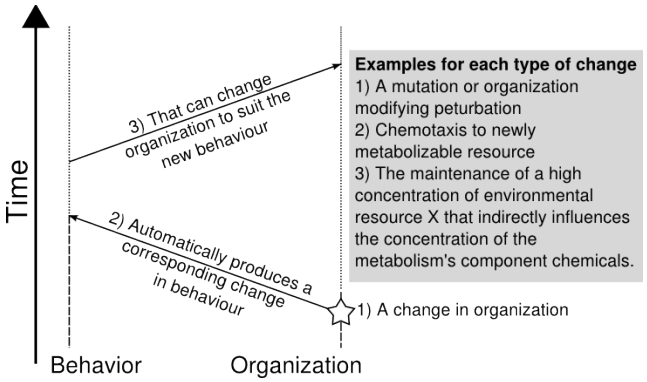


Figure 9: A cycle of mechanisms contributing to adaptation.

corporating a new metabolizable resource into their organization; the mutation will be retained and a new population could emerge in the new resource-rich environment, leading potentially to speciation.

The model presented in this article was inspired on bacterial chemotaxis. But the underlying principles can be easily generalized to a wider context:

1. Behavior modulated by metabolism can produce an online automatic adaptation to change. This change could be *external* (in the sense of an environmental change), or *internal* in that the behaving system has itself changed. Internal change could include genetic mutations or simply perturbations that damage or enhance the behaving system in some way.

2. Automatic, online adaptation to phenomena never experienced before, neither by the individual, nor its ancestors can make otherwise neutral mutations (such as the new ability to consume a resource) more likely to be beneficial mutations (through e.g., moving towards the new resource). It also facilitates speciation events through rapid separation of a newly capable individual from its previous population (discussed above).

3. Behavior can significantly influence metabolism during lifetime. This change can be caused by a persistent behavior (e.g., seeking out of a reactant) or through a random behavioral encounter with a reactant that is incorporated into the auto-catalytic network. In this way, behavior can provide an important source of variation of available chemical compounds, or simply significantly influence the local concentration of reactants.

These type of interactions between behavior, metabolism and evolution we have termed *Behavioral Metabolution*. We can see the cycle of influence in Figure 9, where a change to the organization of an agent causes it to automatically behave differently, in a way appropriate to its change in organization. The new behavior brings the system to a new environment where new mutations (or old mutations) and/or new environmental conditions might be beneficial for metabolism, or as demonstrated in Experiment 1, can produce a new (possibly improved) behavioral mechanism. In this way, a push-me/pull-you dynamic interplay can be established between changes in behavior and changes in metabolism, influencing evolutionary processes in ways that remain mostly unexplored.

The goal of the above experiments is not to provide *evidence* for this phenomenon but to show the very *possibility* and some potential aspects of it. Further extensions of the present work could include an open artificial chemistry with moving protocellular systems that could be used to determine whether the presence of self-movement largely increases the probability of chemical-evolutionary adaptation.

Behavioral metabolution as proto-evolution

It is at the very early stages of life when the coupling between metabolism and behavior could have played a particularly powerful role by instantiating, on its own (and without the presence of a genetic code or even without reproduction!), a form of (proto-)evolution.

Assuming an origins-of-life scenario where membrane compartments or oil-droplets enclose proto-metabolic reaction networks undergoing natural selection (Shenhav et al., 2005; Fernando and Rowe, 2008; Shapiro, 2007) it is evident how any tendency to move (even randomly) would become beneficial to such systems: local metabolic resources would soon be consumed and random movement would lessen competition for local resources. Any bias of random movement towards metabolically more beneficial environments would rapidly be selected. Since the selective-stopping chemotactic strategy has been shown to be easily evolvable (Goldstein and Soyer, 2008) it seems that it would, sooner or later, appear and be metabolism-based (since early metabolic networks would tend to be highly integrated and simple—certainly not with the degree of specialization required for metabolism-independent modes of chemotaxis).

Admittedly, we have implemented an abstract version of a sophisticated flagellar movement, which is highly unlikely to be found at any early stage of evolution. However, at such early stages movement could be implemented on a wide variety of metabolism-controllable ways. For instance, simple reaction-diffusion spots have been shown to be capable of movement (Krischer and Mikhailov, 1994), and more recent work on convection cells (Toyota et al., 2009) also provides an example of potential early prebiotic life-like self-movement. In addition, changes in membrane properties could operate selectively on environmental currents; or, control of protocell buoyancy could lead to upward and downward selective movement. Finally, in its most simplified form, movement could be completely random and provided by environmental factors; to accomplish behavioral metabo-

lution, it would suffice (in this extremely simple form) for the protocell to be capable of influencing the permeability of its membrane.

In any of its possible instantiations, what remains central to the idea of behavioral metabolution (and its relevance to early forms of life) is the potential of the coupling between metabolism and behavior to explore and select the chemical space that is available for metabolic organization (and its behavioral control). In addition, differences between the behavioral trajectories of protocells could lead to differences in their metabolic and behavioral organization, potentially causing speciation and new ecological relationships (e.g., one species consuming another's waste products). An example of a "speciation-like" effect of behavioral metabolution might be to consider irreversible effects on metabolic organization caused by behavioral patterns. Thus, for instance, if the agent continuously moves towards certain types of environments where resources of a certain redundant metabolic pathway are not available it might lose its capacity to metabolize such resources. A variation of experiment 2 could explore this phenomenon by making S act like C (i.e., act as a flagellar rotation modulator), so that agents without C are still viable in environments with $F + N$; without the presence of $E + M$, C could eventually disappear and the agent will lose its capacity to metabolize $E + M$ again.

Conclusion

Despite the central role that both metabolism and adaptive behavior play in artificial life and theoretical biology, very little attention has been paid to the interplay between the two, especially at the ontogenetic and evolutionary scales. When behavior is not controlled by a subsystem that maximizes some function (generally external to the subsystem itself, in the form of selected adaptations or satisfaction of internal "needs") but is, instead, directly modulated by metabolism, then a wide range of adaptive phenomena come to the surface. We have shown, through a model of metabolism-based chemotaxis, how changes to metabolic pathways can qualitatively improve behavioral strategies (e.g., from a selective-stopping to a gradient-climbing strategy; experiment 1) and how behavior might serve to explore and fixate new metabolic pathways (experiment 2). These two examples may be used to reveal the deep role that the behavior-metabolism interplay could have played in evolution: by permitting the behavioral *exploration* of the chemical space available for metabolism, by allowing the behaviorally driven selective and repetitive *exposure* to such chemical compounds and their subsequent incorporation into metabolism and, finally, by the potential behavioral improvements that changes in metabolism could produce. We coined the term "behavioral metabolution" to refer to these phenomena where variations on metabolic dynamics (genetic mutations, creation of new chemical species, etc.) feed back into behavioral changes that, in turn, affect the environmental conditions that feed metabolism.

Different forms of metabolism-behavior coupling could have bootstrapped or driven the evolution of early (pre-genetic) life and could be currently instantiating forms of non-genetic inheritance or genetic assimilation of phenotypic plasticity. We hope to have shown that incorporating this type of connection between behavior and metabolism opens up a promising line of artificial life research where the long term (evolutionary) consequences of interactions between behavior, system organisation and environment and can be systematically studied in simulation.

Acknowledgements X.B. is funded by Programa Nacional de Movilidad de Recursos Humanos del MEC, Plan I-D+I 2008-2011, Spain.

References

Adler, J. (1969). Chemoreceptors in bacteria. *Science*, 166(3913):1588–1597.

Alexandre, G. and Zhulin, I. B. (2001). More than one way to sense chemicals. *J. Bacteriol.*, 183:p. 4681–4686.

Ashby, W. R. (1952). *Design for a brain*. J. Wiley.

Bray, D., Levin, M., and Lipkow, K. (2007). The chemotactic behavior of Computer-Based surrogate bacteria. *Current Biology*, 17(1):12–19.

Egbert, M. D. and Di Paolo, E. (2009). Integrating autopoiesis and behavior: An exploration in computational chemo-ethology. *Adaptive Behavior*, 17(5):387–401.

Egbert, M. D., Di Paolo, E., and Barandiaran, X. E. (2009). Chemo-ethology of a adaptive protocell: Sensor-less sensitivity to implicit viability conditions. In *Proc. of the 10th European Conf. on Artificial Life*, Budapest. Springer Verlag.

Fernando, C. and Rowe, J. (2008). The origin of autonomous agents by natural selection. *BioSystems*, 91(2):355–373.

Ganti, T. (1975). Organization of chemical reactions into dividing and metabolizing units: the chemotons. *Bio Systems*, 7(1):15.

Goldstein, R. A. and Soyer, O. S. (2008). Evolution of taxis responses in virtual bacteria: Non-Adaptive dynamics. *PLoS Comput. Biol.*, 4(5):e1000084.

Kauffman, S. A. and Farmer, J. D. (1986). Autocatalytic sets of proteins. *Origins Life Evol. Biosphere*, 16(3):446–447.

Krischer, K. and Mikhailov, A. (1994). Bifurcation to traveling spots in Reaction-Diffusion systems. *Phys. Rev. Lett.*, 73(23):3165.

Morowitz, H. J. (1999). A theory of biochemical organization, metabolic pathways, and evolution. *Complexity*, 4(6):39–53.

Rasmussen, S., Bedau, M. A., Chen, L., Deamer, D., Krakauer, D. C., Packard, N. H., and Stadler, P. F., editors (2008). *Protocells: Bridging Nonliving and Living Matter*. The MIT Press.

Shapiro, R. (2007). A simpler origin for life. *Scientific American Magazine*, 296(6):46–53.

Shenhav, B., Solomon, A., Lancet, D., and Kafri, R. (2005). Early systems biology and prebiotic networks. *Lecture Notes in Computer Science*, 3380:14–27.

Suzuki, K. and Ikegami, T. (2009). Shapes and self-movement in protocell systems. *Artif. Life*, 15(1):59–70.

Toyota, T., Maru, N., Hanczyc, M. M., Ikegami, T., and Sugawara, T. (2009). Self-Propelled oil droplets consuming Fuel surfactant. *J. Am. Chem. Soc.*, 131(14):5012–5013.

On the Semantic Capacity of (Bio-)Chemical Systems

Dennis Görlich[1,2] and Peter Dittrich[1,2]

[1]Bio Systems Analysis Group, Institute of Computer Science,
Friedrich Schiller University Jena, Ernst-Abbe-Platz 2, D-07743, Jena, Germany
[2]Jena Centre for Bioinformatics, JCB
dennis.goerlich@uni-jena.de, peter.dittrich@uni-jena.de

Extended Abstract

Can we objectively distinguish chemical system that are able to process meaningful information from those that are not suitable for information processing? In this talk we present a formal method to asses the semantic capacity of a chemical reaction network.

The basic idea is to measure how easy it is to implement an organic molecular code with this network. Inspired by Barbieri (2008), we define a molecular organic code with respect to a given reaction network as a mapping between two sets of molecular species called signs and meanings, respectively, such that (a) this mapping can be realized by a third set of molecular species, the codemaker and (b) there exists alternative sets of molecular species, i.e., alternative codemakers, implying different mappings between the same two sets of signals and meanings (Görlich and Dittrich, in press).For an example see figure . We define the semantic capacity of a reaction network by simply counting the number of different codes. We analyzed models of real chemical systems (Martian atmosphere chemistry and various combustion chemistries), bio-chemical systems (gene expression, gene translation, and phosphorylation signaling cascades), as well as random networks and artificial chemistries. We found that different chemical systems posses different semantic capacities. Basically no semantic capacity was found in the atmosphere chemistry of Mars and all combustion chemistries, i.e., with these chemistries, organic codes cannot be implemented. Whereas the bio-chemical systems posses very high semantic capacities, with (hypothetically) increasing capacity from metabolic networks, signalling networks, to gene regulatory networks. andom networks have a much lower semantic capacity than biological networks like regulatory networks or the genetic code network. Random networks show only organic codes for very specific parameters, for example a random network with 15 species and an optimal density of reactions (i.e., 30) has on average 2.7 code pairs whereas a gene regulatory network of the same size has 9 code pairs. This can be explained by the fact that it is hard to achieve at the same time a high number of closures and a large pool of pathways to select from. Note that for a code pair at least ten different closed sets are necessary.

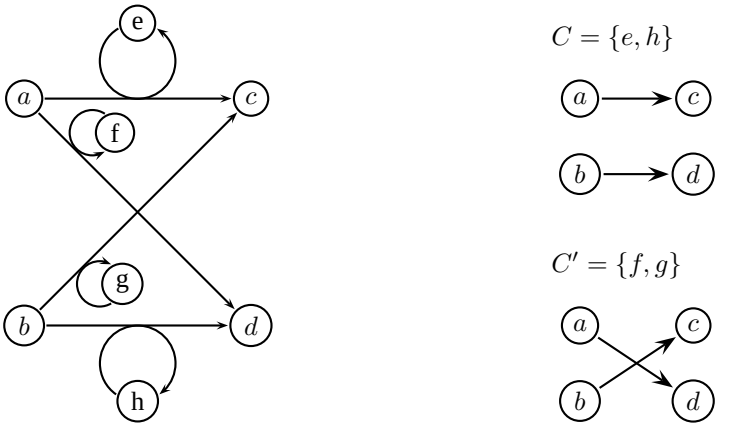
Our definition provides neither a necessary nor sufficient criteria for information processing, however our results indicate that it can be applied to evaluate the information processing capabilities of a chemical system on an algebraic level and may thus be a useful tool to understand the origin and evolution of meaningful information, e.g., at the origin of life.

Acknowledgement: We acknowledge financial support by the Jena School of Microbial Communication (JSMC) and the German Research Foundation (DFG).

References

Barbieri, M. (2008). Biosemiotics: a new understanding of life. *Naturwissenschaften*, 95(7):577–599.

Görlich, D. and Dittrich, P. (in press). Identifying Molecular Organic Codes in Reaction Networks. To appear in *Proceedings of 10th European Conference, ECAL 2009, Budapest, Hungary, September 13-16, 2009, LNCS, Springer Berlin*.



(a) Network view

(b) Code view

Figure 1: (a) Illustration of a reaction network motif that can realize a molecular organic code. The network $\langle \mathcal{M}, \mathcal{R} \rangle$ consists of species $\mathcal{M} = \{a, b, c, d, e, f, g, h\}$ and reaction rules $\mathcal{R} = \{a + e \rightarrow e + c, \ldots\}$. (b) Illustration of the two possible mappings between binary sets of species. In this example, we can obtain a molecular organic code by choosing $S = \{a, b\}$ and $M = \{c, d\}$ as signs and meanings, respectively. The sets $C = \{e, h\}$ is a codemaker with $C' = \{f, g\}$ the respective alternative codemaker. Note that the arrows in (a) denote reactions and the arrows in (b) denote mappings.

Mode Switching and Collective Behavior in Chemical Oil Droplets

Naoto Horibe[1], Kei Kobayashi[1], Martin Hanczyc[2] and Takashi Ikegami[1]

[1]University of Tokyo
[2]University of Southern Denmark
nhoribee@08.alumni.u-tokyo.ac.jp

Extended Abstract

We have designed a series of chemical experiments to investigate the emergence of spontaneous self-movement in a simple chemical system. More specifically we have followed the dynamic motile behavior of oil droplets consisting of oleic anhydride in an aqueous environment. The droplets can move by creating an internal convection flow, which enforces a break in symmetry and organizes droplet movement. The droplets can exhibit several different styles of motion depending on their age, size and the pH condition. The dynamics of single droplets on a glass plate show a transition from the anomalous diffusion to a directional motion then to a more complex vibrating motion by radically modulating its boundary shape. When many droplets are present, they aggregate and physically contact each other. We often observe that the internal convection flow of those droplets synchronize, i.e. the directions of flow become parallel to each other like magnetic spin systems. These discoveries illustrate that coupling a chemical reaction (hydrolysis of the anhydride) to a physical body (the oil droplet) can result in an instability that affects both convective flow patterns and overall shape, and therefore the agents and their collective behavior.

In order to clarify how droplet 'behavior' changes with controlled parameters of the system, we analyzed the system for micro scale flow patterns using microscopy and for macro scale behavior using image analysis and droplet tracking tools. First, the shape of the droplet changes at a certain point as we increase the size from a few micrometers to a few centimeters, and accordingly the motion pattern changes from the quasi Brownian to directional movement to a vibrating mode. We have characterized those tendencies by measuring the stop/go intervals and the auto correlation functions. A shape change in such a system has great importance since deformations will create new interfacial surfaces where dynamic phenomena may occur. Second, when droplets come together, their internal convection flow is re-configured resulting in a collective motion. When the droplets use up their chemical energy (reaction on their surfaces), the collective behavior will disappear. Therefore the collapse and genesis of collective behavior is the evidence of the active moving droplets.

We tried to replicate those phenomena with the numerical procedure (coupling the Navier-Stokes equation with a chemical reaction). When the initial size exceeds a certain limit, the numerical procedure fails to produce physically correct values. The droplet breaks up into pieces. Thus the breakup of the numerical procedure may correspond to the shape transition. Therefore the system is challenging for both experimental and numerical studies and at the conference we will focus on how single droplet mode switching may reflect the important parameters that will allow different behaviors to emerge from such a simple chemical system. Also when multiple droplets are present, the same signals that organize the movement of a single droplet may be used to organize and coordinate the behavior of several droplets. Collective behavior can begin to be understood following the simple physico-chemical processes described here.

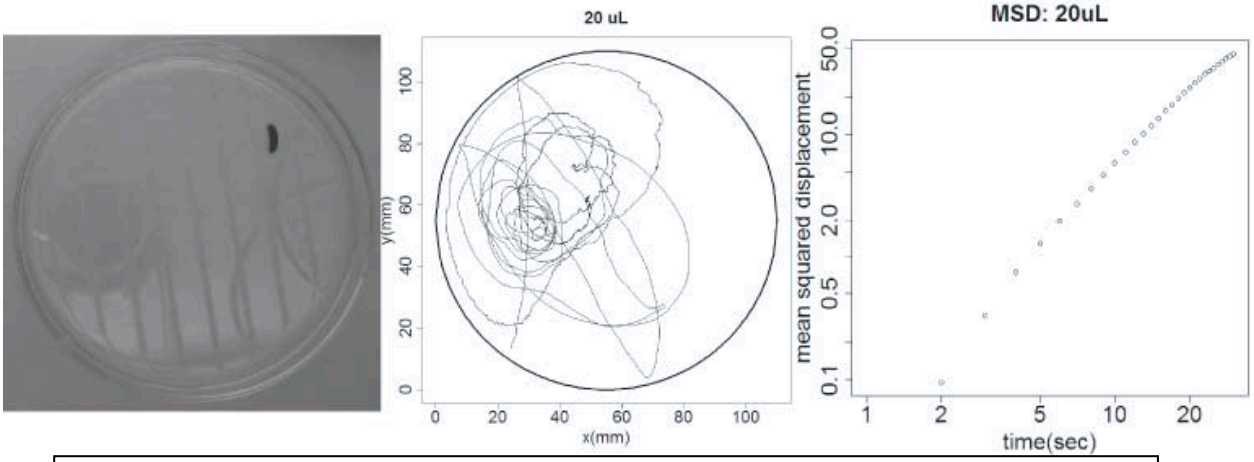


Fig. Droplet in a glass plate, trajectory of a droplet and autocorrelation function of a trajectory.

Early Evolution of Memory Usage in Digital Organisms

Laura M. Grabowski[1], David M. Bryson[2], Fred C. Dyer[2], Charles Ofria[2], and Robert T. Pennock[2]

[1]University of Texas-Pan American, Edinburg, TX 78539
[2]Michigan State University, East Lansing, MI 48864
grabowskilm@utpa.edu

Abstract

We investigate the evolution of memory usage in environments where information about past experience is required for optimal decision making. For this study, we use digital organisms, which are self-replicating computer programs that are subject to mutations and natural selection. We place the digital organisms in a range of experimental environments: simple ones where environmental cues indicate that a specific action should be taken (*e.g.*, turn left to find food) as well as slightly more complex ones where cues refer to prior experience (*e.g.*, repeat the action indicated by the previous cue). We demonstrate that flexible behaviors evolve in each of these environments, often leading to clever survival strategies. Additionally, memory usage evolves only when it provides a significant advantage and organisms will often employ surprisingly successful strategies that do not use memory. However, the most powerful strategies we found all made effective use of memory.

Introduction

Organisms must be able to respond to their environment to maximize their chances of survival. They must be able to vary their reactions based on differences in time, place, or circumstance. Evolution has produced many mechanisms that allow such flexible responses, including simple reflexive behavioral routines, such as the response of bacteria like *Escherichia coli* (*E. coli*) to move toward food, or innate behavioral preferences and patterns, as observed in many insects (Dukas and Bernays, 2000). In well-defined, stable circumstances, a repertoire of innate, fixed behaviors may be sufficient to allow organisms to be successful. However, when circumstances can vary due to dependencies on time, place, previous experiences or environmental changes, then more dynamic and flexible behavioral mechanisms are needed. In such cases, memory and learning may allow individuals to more effectively adjust behavior according to the local world state (Dukas, 2008).

How do environment, memory, and learning interact in an evolutionary context? This question is of great interest to both biologists and computer scientists who study the evolution of intelligence. We present early results in our exploration of this interplay in the context of the evolution of navigation. Our experimental environments are inspired by maze-learning experiments with honey bees (described below). By using these types of environments, we maintained a strong connection between our experiments and their biological motivation, and we were able to probe specific issues relating to the evolution of memory use. Situated at the intersection of biology and computer science, our approach aims to provide insight for both disciplines.

Motivation from insect navigation

Insects are ideal subjects for the study of navigation behaviors. Ants, bees, and other insects use an array of innate strategies to navigate, including *landmark tracking*, where the insect refers to a visual marker (Graham et al., 2003), and *path integration* (Müller and Wehner, 1988), which is the continual internal monitoring of distance and direction relative to a reference location (*e.g.*, the nest). Studies of maze learning in insects are of particular interest, since many bees and ants often follow fixed routes from the nest to a foraging site (Collett et al., 2003). In learning a maze, an insect is learning to follow a well-defined path (Collett et al., 1993). Bees have been trained to fly through mazes of varying complexity. Studies by Collett and colleagues (Collett and Baron, 1995; Collett et al., 1993) used small mazes to investigate bees' ability to learn motor or sensorimotor sequences. One study (Collett et al., 1993) forced bees to fly along prescribed routes and through obstacles in a large box and concluded that bees can remember sensory and motor information that allows them to reproduce a complex route.

A study by Zhang and colleagues (1996) demonstrated that honey bees could use specific visual cues to learn to fly through structurally complex mazes. Another study (Zhang et al., 2000) probed whether bees learn and recognize structural regularity in the mazes. For these experiments, bees were trained and tested in four different types of mazes: constant-turn, where turns are always in the same direction; zig-zag, where each turn alternates direction; irregular, which has no apparent pattern of turns; and variable irregular, where bees had to learn several irregular mazes at the same time. The bees performed best in constant-turn mazes, somewhat poorer in zig-zag mazes, still worse in irregular

mazes, and poorest of all in variable irregular mazes. The authors concluded that the bees' performance in the various configurations depends on the structural regularity of the mazes, and the ease with which the bees can recognize and learn that regularity.

Computational approaches

Evolutionary robotics has dealt extensively with several facets of evolving memory and learning. One aspect is phenotypic plasticity, the ability of a genotype to express differently in different environments. Nolfi et al. (1994) studied this topic by evolving neural network "brains" for virtual robots in environments that alternated between light and dark. Individuals that evolved under these conditions were able to tune their behavior appropriately for both kinds of environments, adapting within an individual "lifetime" to environmental changes.

Evolution and learning employ different mechanisms and occur at differing time scales making their interaction, and, indeed, the evolution *of* learning, a topic of intense study (Nolfi and Floreano, 2002). A study by Floreano and Urzelai (2000) is a strong example of the latter. They evolved neural networks with local synaptic plasticity and compared them to fixed-weight networks in a two-step task. The networks evolved to turn on a light and then move to a grey square. The results showed that local learning rules helped networks alter functionality quickly, facilitating moving from one task to the other. Blynel and Floreano (2003) explored the ability of continuous time recurrent neural networks (CTRNNs) to solve reinforcement learning problems in the context of T-Maze and double T-Maze navigation tasks, where the robot had to find and "remember" the location of a reward zone. The learning in this case occurred without modification of synapse strengths, coming about instead from internal network dynamics.

Methods

Avida: Overview

Digital evolution (Adami et al., 2000) is a form of evolutionary computation in which a population of self-replicating computer programs, or "digital organisms," is placed in a computational environment where they compete and mutate. Digital evolution can be used both for understanding biological processes and for applying insights from biology to computational problems. The Avida software system (Lenski et al., 2003; Ofria and Wilke, 2004) is a widely used platform for digital evolution. Avida provides a separate instance of real evolution useful for experimental studies (Pennock, 2007).

The "world" in which evolution takes place in Avida is a discrete two dimensional grid containing a population of digital organisms (or "Avidians"), with at most one Avidian per grid cell. The individual organism consists of its "genome," which is a circular list of assembly language-like instructions, and its virtual CPU. The CPU contains three general purpose registers, several heads, and two stacks. The instructions in the organism's genome execute by acting on the components of the virtual CPU, and execution of instructions incurs a cost in virtual CPU cycles. An Avida organism accomplishes all tasks (*e.g.*, replication and movement) by executing Avida instructions.

An Avida organism replicates by copying its genome into a block of memory that will be its offspring's genome. The copying process is sometimes imperfect, leading to differences between the genomes of parent and offspring. These differences are mutations, and may occur as a substitution, insertion or deletion of an instruction. The Avida instruction set is robust to mutations, so that any program will be syntactically correct even when mutations occur (Ofria et al., 2002). Upon replication, an organism's offspring is placed in a random grid cell, terminating any organism that previously occupied that cell. Thus, organisms in the population compete for the limited space in the set of grid cells, and organisms that replicate more quickly will have a greater number of descendants. An organism can increase its metabolic rate (the relative speed it executes instructions) by performing user-specified tasks. We measure the fitness of an organism as its metabolic rate divided by the number of CPU cycles it requires to replicate.

Experimental environments

Each Avidian was placed in an environment containing a path (inspired by the maze-learning experiments discussed earlier (Zhang et al., 1996, 1999)) that it could gain nutrients by following. To follow a path, an organism must sense cues in the environment that tell it how to stay on the path, and react appropriately to those cues. In some cases, this task necessitated evolving the ability to store and reuse experience. Sensing and movement in the virtual grids were accomplished by executing experiment-specific Avida instructions. The movement instruction moves the organism into the grid cell that it is currently facing. Movement occurs only one step at a time. In the virtual environments of the current study, each organism has its own virtual grid, so organisms do not interact during movement. Orientation changes require additional instructions, one for turning right 45 degrees and another for turning left 45 degrees. Organisms had to combine the different instructions—sensing, movement, and orientation—in order to successfully follow more complex paths.

An organism must navigate its environment to find sparsely distributed "food". Movement requires energy, so each step depletes the organism's energy store. When an organism encounters food, the food gives it more energy than the amount lost through movement. Locations that are off the path are "empty", containing no food. When an organism moves into an empty location, the organism loses a small amount of energy, without regaining any energy.

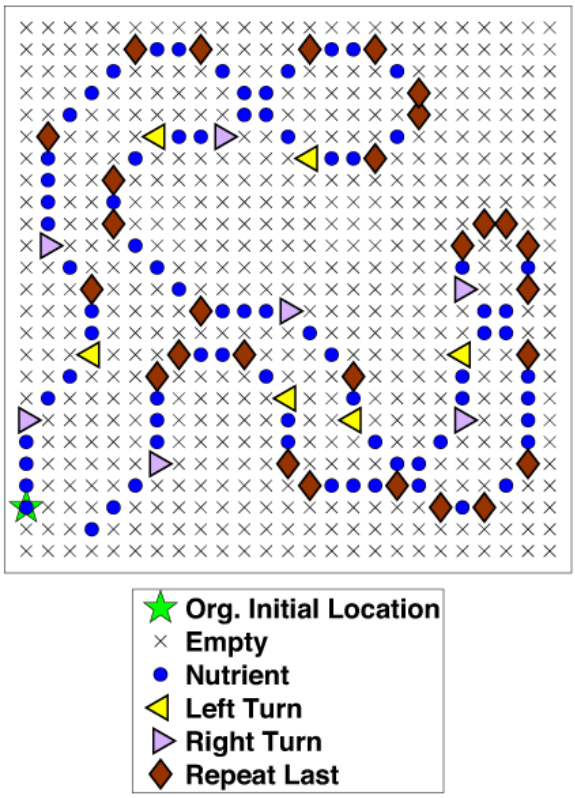


Figure 1: Example experimental environment, using all cues.

Movements into empty locations are detrimental to the organism: continued energy depletion will impair the organism's ability to replicate. Organisms that move along the food-rich path build up their energy, and are able to execute at an accelerated rate. Each environment contained some combination of the following cues (*e.g.*, Figure 1):

1. **Nutrient:** A cue that indicates the path, and provides energy (the "food" on the path).
2. **Directional cue:** A cue that indicates to turn either right or left (45 degrees in the specified direction) to remain on the path. Directional cues also act as a nutrient.
3. **Repeat-last:** A special directional cue to repeat the last turn direction, and acting as a nutrient.
4. **Empty:** A cue that indicates cells that are off of the path. The net loss of energy from a step into an empty cell equals the net gain of energy from a nutrient.

All paths used only 45-degree turns, so that a turn could be accomplished with a single, unmodified Avida instruction.

An organism that travels the entire path without a misstep receives the maximum possible bonus. The bonus is based on the count of unique path cells that the organism encountered less the total count of movements into cells that are off the path, without allowing the value to become negative. Organisms were not penalized for taking extra steps on the path. Conceptually, the path cells are analogous to food patches. The organism consumes most of the food in the patch the first time it moves into a path cell. Subsequent visits to a previously visited location supply only enough food to offset the energy lost in moving to the location. On the other hand, empty cells are always empty, and movement always requires energy. Each step into an empty location results in a net loss of energy, because the organism cannot replenish its energy stores at that location. We used the value of the count of path cells traversed to determine the organism's metabolic rate bonus. Our approach delivered an exponential reward, doubling the organism's metabolic rate bonus for each step on the path that is not counteracted by a step off the path into an empty cell.

Experiments and results

We conducted experiments using multiple environment types. Each environment type placed different memory use and decision-making demands on the organisms. In all cases, an organism could sense the contents of a cell by using a sense instruction; each cue (nutrient, right turn, left turn, repeat last, empty) had a unique sensed value. The sense instruction provided the sensory information from the environment, but the organism had to decide what, if anything, to do with that information.

Environment 1: Evolving reflex actions. This environment type contained turns in a single direction (*i.e.*, one path instance contained only right turns, while another path instance had only left turns; see Figure 3 below). The single-direction paths had a spiral shape and contained three cues: nutrient, empty, and only one type of directional cue (right or left). This environment presented organisms with all information required to make turn decisions at the time and place that it was needed.

It is reasonable to believe that reflexive responses evolved before learning (Todd and Miller, 1990), and these types of responses are well known as the basis for conditioning (Rescorla, 1988). From a practical standpoint, if an organism cannot evolve to perform an action correctly when it always should, it will never be able to effectively decide to act selectively.

Environment 2: Evolving volatile memory. In the first environment type, the organisms could sense a directional cue at each turn; a right turn and a left turn have different sensed values. In that setup, past cues never had to be stored in order to make an informed decision about the current action. In the second set of experiments, the organism can remain on the path only if it remembers the most recent turn

direction. In this environment, if a turn is in the same direction as the preceding turn, the sense value is different from the sense values of a right turn and a left turn. This new cue signals an organism to "repeat the last turn direction". This arrangement of information along the path means that an Avidian must be able to change the remembered sense cue value an arbitrary number of times in its lifetime, and at irregular intervals. Thus, this memory is *volatile* as opposed to the unchanging reflex memory needed for the first experimental environment. The arrangement of cues in the second environment type necessitates flexible use of information from an increasingly complex environment. An organism must remember a binary value (turn right or turn left), or one bit of information in information theory terms.

To provide environmental variation and discourage the evolution of brute-force solutions, organisms were presented (at random) with one of four different paths of each environment type during the course of evolution. Thus, any individual organism had a 0.25 probability of being born into the same environment as its parent.

For each experimental environment, we ran 50 replicate populations capped at 3600 organisms for 250,000 updates (a unit of time in Avida), or a median of approximately 33,000 generations. Each experiment seeded the population with an organism capable only of replication. This simple self-replicator ancestor's genome consists of 100 instructions, comprising a short copy loop and a large number of no-operation instructions. Any other instructions and capabilities can appear through mutations. All experiments used a 0.085 genomic mutation rate for a length-100 organism (a 0.0075 copy-mutation probability per copied instruction, and insertion and deletion mutation probabilities of 0.05 per divide) (Ofria and Wilke, 2004).

Results and discussion

To evaluate the success of different experimental treatments, we used both quantitative performance measures and behavioral tests of evolved organisms. For the quantitative measures of performance, we examined fitness and task quality over time. These values are tracked and recorded during the course of an Avida experiment. For behavioral tests, we traced execution and trajectory of evolved organisms on different path configurations, including paths that were never experienced during the course of evolution.

We use task quality to measure how well an organism performs in a given environment. For this study, task quality measures the fraction of the path an organism traversed, less any movement into empty cells; an organism that traversed the full path without moving into any empty squares would have a task quality of 1.0. Because overall metabolic rate for these experiments was associated solely with the path traversal task, task quality and fitness track closely. The overall performance of a population is shown by the average task quality for that population; the maximum task quality quantifies the performance of the best-performing organisms from each population, and the Average Maximum Task Quality (AMTQ) averages this population maximum task quality over all 50 replicate experiments of each environment type.

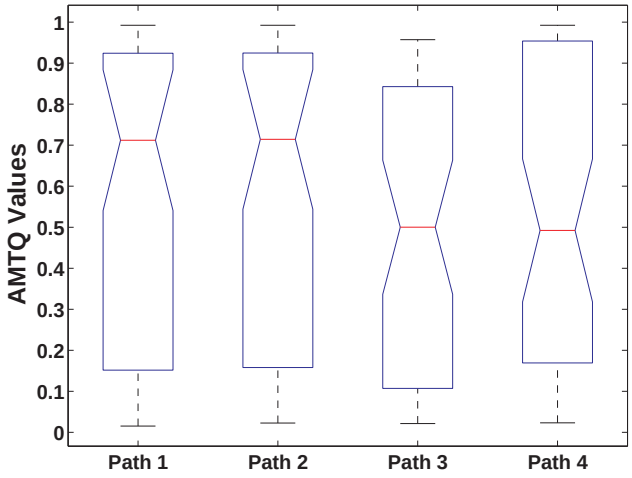
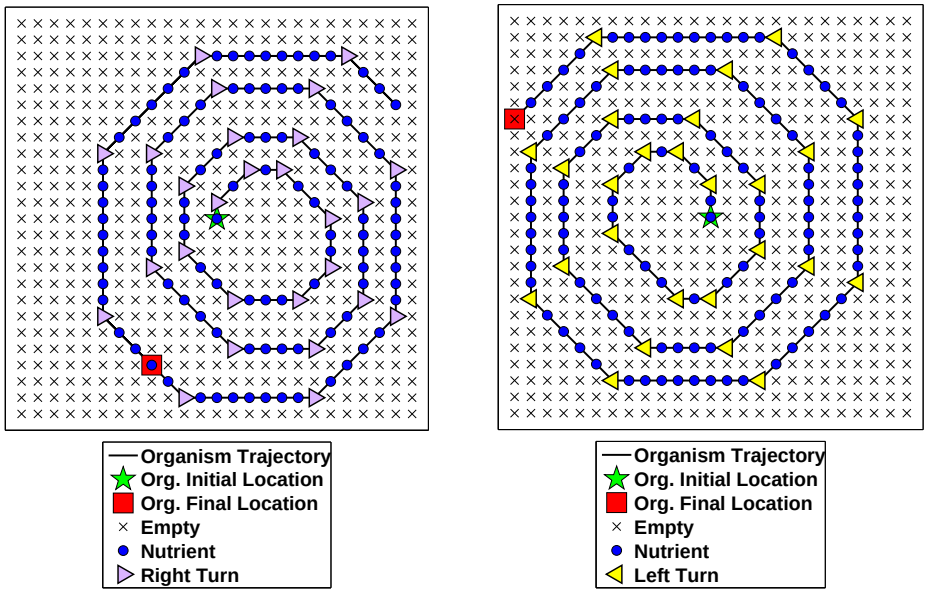


Figure 2: Distribution of average maximum task quality (AMTQ), individual Experiment 1 paths. Paths 1 and 2 are right-turn-only paths, Paths 3 and 4 are left-turn-only paths. There is no significant difference in the AMTQ distributions for each path (Kruskal-Wallis Test, $p = 0.287$).

To test the behavior of evolved organisms, we ran execution traces for selected final dominant genotypes (most abundant genotype at the end of an evolution experiment) in different environments. With each environment, we tested organisms (1) on the same virtual grids that the organisms experienced during evolution, to observe their behavior in those "ancestral" environments, and (2) in novel environments, *i.e.*, paths that no organism experienced during evolution, to demonstrate the generality of the evolved solutions, or uncover solutions that had been tuned specifically to the ancestral environments.

Evolving reflex actions. Figure 2 shows the distributions of AMTQ values for each of the four single-direction paths. There was no significant difference between the AMTQ distributions for each path, as measured by the AMTQ at the end of evolution (Kruskal-Wallis Test, $p = 0.287$). Figure 3 shows trajectories of the final dominant with the highest ending metabolic rate among all 50 replicate single-direction path experiments, on a right-turn-only path (Figure 3a) and on a left-turn-only path (Figure 3b). The organism's trajectories on the other two evolutionary environment paths are qualitatively identical to those shown. The organism's evolved strategy performed well in both turn environments. The organism did some "backtracking" on the right-turn grid, *i.e.*, it turned around and retraced some of its steps on the path. This behavior did not reduce the organism's task



(a) Right-turn Path (b) Left-turn Path

Figure 3: Trajectories of an example evolved organism from Experiment 1 on paths that were experienced during evolution ("ancestral" paths).

quality as the calculation does not penalize an organism for multiple traversals of a path cell. The risk of such behavior is that the organism wastes CPU cycles, thus reducing fitness, although this particular organism still evolved to be the most fit individual in its population. This organism was able to navigate the entire right-turn path without entering any empty cells. The organism also successfully followed the left-turn-only path, stopping after it encountered one empty cell.

To understand this organism's algorithm, we analyzed its execution while traversing each of these two paths. Most of the path-following and replication code of this organism's genome is organized into two modules. The first module, "Module 1A," is mostly concerned with moving on a right-turn path while the second module, "Module 1B," focuses on left-turn paths and contains a copy loop. These code sections are both executed, regardless of whether the organism is on a right-turn or left-turn path, but the behavior that the modules produce differs according to the path type. In general, Module 1A is a "counting" routine. When the organism is on a right-turn path, Module 1A counts the organism's steps. On a left-turn path, Module 1A counts the number of rotations the organism executes. Module 1B allows the organism to travel to the end of a left-turn path and then replicate. When the organism is on a right-turn path, the organism uses Module 1B to "backtrack" on the path, retracing some of its steps, while it finishes its replication process.

Evolving volatile memory. The irregular path environment was more challenging than the environments of Experiment 1. The AMTQ for these experiments shows a weaker performance than in the other environment. The difference in AMTQ at the end of 250,000 updates was significantly different in the irregular path experiments compared to the other environment (Kruskal-Wallis Test, $p < 0.05$). There was, however, no significant difference in the performance on each path, measured by the AMTQ at the end of evolution (Kruskal-Wallis Test, $p = 0.238$). Figure 4 shows the

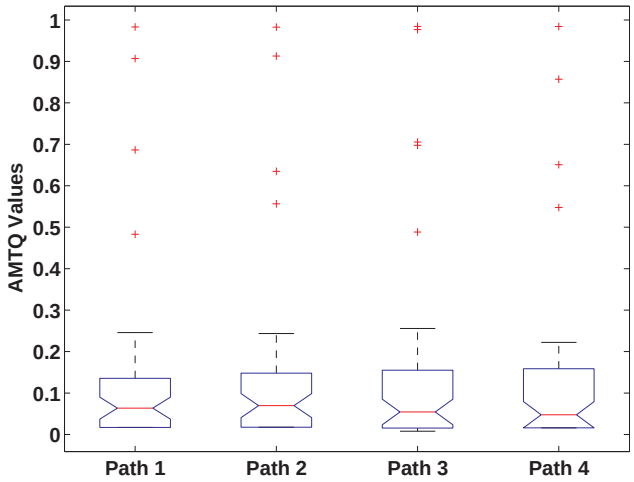
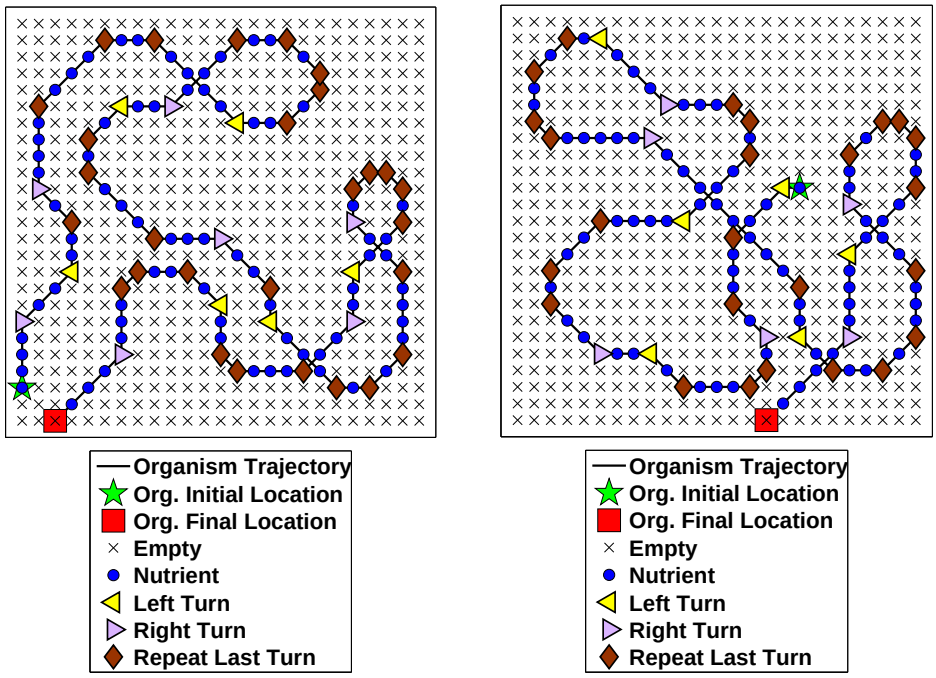


Figure 4: Distribution of average maximum task quality (AMTQ), individual Experiment 2 paths. There is no significant difference in the AMTQ distributions for each path (Kruskal-Wallis Test, $p = 0.238$).



(a) Example Trajectory, Path 1

(b) Example Trajectory, Path 2

Figure 5: Trajectories of an evolved organism from Experiment 2 irregular path experiments. In both (a) and (b), the organism stops moving after encountering one empty cell.

distributions of AMTQ values for each of the four ancestral irregular paths.

Despite the generally inferior performance of the evolved populations in this environment, some highly effective strategies evolved. Figure 5 shows the trajectories of the final dominant organism from the population with the highest AMTQ at the end of the 250,000 update evolution run. This organism has an excellent solution for following these paths, stopping after taking one step off the end of the path into an empty cell. The evolved algorithm is equally effective on novel paths, as shown in Figure 6.

The execution of this organism's genome is somewhat complicated, and shows an impressive degree of flexibility. In general, this organism operates by moving its execution to different parts of its genome based on the sensed environmental cue. The organism accomplishes all of its path-following with two loops, one for moving through left-turn path sections, "Module 2A," and the other for moving through right-turn path segments, "Module 2B." Unlike the other organisms that we have examined in detail, this organism has well-defined functional and structural modularity for handling right-turn and left-turn path sections. Module 2A appears before Module 2B in the organism's genome. Module 2A can perform an arbitrary number of consecutive left turns, and any number of forward steps. Using Module 2B, the organism can maneuver through right-turn path sections. Module 2B functions with arbitrary numbers of forward steps and repeated right turns. If a left turn cue is sensed, Module 2B terminates and execution jumps to the beginning of the genome, eventually reaching Module 2A again. If an empty cell is sensed while execution is in Module 2B, the module terminates and execution continues with the instructions after the module. In addition to the movement modules, the organism has a tight copy loop near the end of its genome that accomplishes almost all the copying for the organism's replication.

There are two features of this organism that are particularly interesting. The first is the organization of the genome. The sections of the genome that do the bulk of the work for this organism—the two movement modules and the copy loop—are functionally and spatially modular. For all three of these loops, very little happens within them apart from the main function of the loop. The loops are also spatially modular: they are located in different sections of the genome. Example organisms from the preceding experiments also demonstrate structural modularity, but their functional modularity is generally less defined. The second feature of special interest is the flexibility of execution flow between code modules. The execution flow enables the organism to cleverly handle all the contingencies of the environment. For example, even though Module 2A (left-turn module) is encountered first in the sequential execution of the genome, if a right turn is encountered first, the flow moves easily through Module 2A and into Module 2B (right-turn module). The

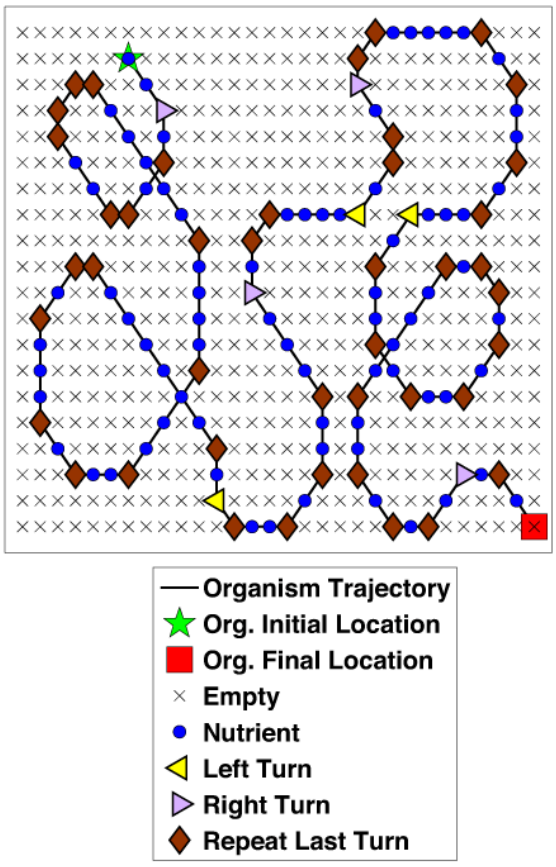


Figure 6: Trajectory of an evolved organism from Experiment 2 irregular path experiments, traversing a novel path.

algorithm evolved to deftly maneuver along the paths, using environmental cue information to alter its execution.

By analyzing the execution of evolved genomes from both environment types, we found that memory use involved both the organization of the genome and volatile states of the organisms' virtual CPUs. The organization of the genomes provided functional modularity, while different environmental information created different states of the virtual CPU that lead to differential behavior based on the current state in the environment. The resulting behaviors formed a simple set of behavioral repertoires that could be used flexibly in response to environmental stimuli.

Conclusions and Future Work

Through these results, we illustrate that memory and flexible behavior can evolve in simple environments. Evolution capitalizes on both environmental change and regularity to construct these solutions. The experiments presented here suggest, not surprisingly, that it is more difficult to evolve volatile memory than to maintain "evolutionary memory" (reflexes).

Results such as those we present here may inform investigation in both biology and computer science. Insights into the evolution of behavioral characteristics of natural organisms must rely on studies of extant species, since the fossil record provides little information about an animal's behavior. Our results may help provide additional insights by allowing detailed analysis of the evolutionary transitions that led to intelligent behavior. Those insights can, in turn, be used in the context of computer science to produce artificial systems that exhibit the behavioral flexibility of natural systems. The current work is an early step in this direction.

Natural evolution produced many impressive navigation abilities in animals. These capabilities are made up of interwoven strategies, which are themselves made up of simpler underlying mechanisms. Memory is undoubtedly one such underlying mechanism. We witnessed memory evolve even when not required in the single-direction path experiments; the "step-counter" organism based part of its strategy on tracking its progress along its path. This organism possesses a simple odometry mechanism, like those found in many animal navigation systems. This same organism was also able to count its rotations to orient itself in the correct direction. Self-referential compasses are another component of animal navigation. The results from our study hold promise of future insights into questions surrounding the evolution of navigation. For example, the environments used in the current study can be adjusted so that organisms need to explore the environment to find resources, and then return to their initial location as efficiently as possible. This situation sets up investigating the evolution of path integration. There is a rich collection of evidence of this ability in many animals, and different models of the mechanism have been presented (*e.g.*, Mittelstaedt (1985), Müller and Wehner (1988), Hartmann and Wehner (1995)). How evolution produced such a capability is, however, an open question. Some interesting work has explored this issue, such as Vickerstaff and DiPaolo (2005), who used a genetic algorithm approach to evolve neural network models of path integration. Experiments such as those in the current work have the potential to contribute to that discussion, by allowing detailed examination of both the evolution and the evolved algorithms that are not possible in network based approaches.

The path-following environments can be used to study the evolution of associative memory, the process by which animals learn about cause-and-effect relationships between events and then behave appropriately (Rescorla, 1988; Shettleworth, 1998). We can simulate the arbitrary stimulus, important for associative learning, by generating random numbers for signpost cues each time a particular path is assigned to an organism, changing the values for the organism's offspring. For true associative memory, the organisms should be able to associate arbitrary features of their surroundings with their desired goal. We plan to vary the relationship between the cue and the target, so the cue might be prompting a turn in the paths, or it might indicate that the food source is a

certain distance ahead, regardless of what else the organisms have seen in the interim.

The experimental results that we present here demonstrate the evolutionary origin of simple intelligence and behavioral flexibility. Organisms from these experiments were capable of gathering information from the environment, storing that information, and using the information for decisions. Moreover, organisms that succeeded in the irregular path environments were able to use a past individual life experience to guide future decision-making.

Acknowledgments

We would like to thank Philip McKinley, Wesley Elsberry, Jeff Clune, Michael Vo, Erica Rettig, and other members of the MSU Digital Evolution Laboratory for valuable comments and discussion. This work was supported by a grant from the DARPA FunBio program, NSF grant CCF-0643952, and a grant from the Cambridge Templeton Consortium,"Emerging Intelligence: Contingency, Convergence and Constraints in the Evolution of Intelligent Behavior."

References

Adami, C., Ofria, C. A., and Collier, T. C. (2000). Evolution of biological complexity. *Proceedings of the National Academy of Science*, 97:4463–4468.

Blynel, J. and Floreano, D. (2003). Exploring the T-maze: evolving learning-like robot behaviors using CTRNNs. In Cagnoni, S., editor, *Applications of Evolutionary Computing*, Lecture Notes in Computer Science, pages 593–604. Springer, Berlin.

Collett, T. S. and Baron, J. (1995). Learnt sensori-motor mappings in honeybees: interpolation and its possible relevance to navigation. *Journal of Comparative Physiology A*, 177:287–298.

Collett, T. S., Fry, S. N., and Wehner, R. (1993). Sequence learning by honeybees. *Journal of Comparative Physiology A*, 172:693–706.

Collett, T. S., Graham, P., and Durier, V. (2003). Route learning by insects. *Current Opinion in Neurobiology*, 13:718–725.

Dukas, R. (2008). Evolutionary biology of insect learning. *Annual Review: Entomology*, 53:145–160.

Dukas, R. and Bernays, E. A. (2000). Learning improves growth rate in grasshoppers. *Proceedings of the National Academy of Science*, 97(6):2637–2640.

Floreano, D. and Urzelai, J. (2000). Evolutionary robots with online self-organization and behavioral fitness. *Neural Networks*, 13:431–443.

Graham, P., Fauria, K., and Collett, T. S. (2003). The influence of beacon-aiming on the routes of wood ants. *Journal of Experimental Biology*, 206:535–541.

Hartmann, G. and Wehner, R. (1995). The ant's path integration system: a neural architecture. *Biological Cybernetics*, 73:483–497.

Lenski, R. E., Ofria, C., Pennock, R. T., and Adami, C. (2003). The evolutionary origin of complex features. *Nature*, 423:139–144.

Mittelstaedt, H. (1985). Analytical cybernetics of spider navigation. In Barth, F. G., editor, *Neurobiology of arachnids*, pages 298–316. Springer, Berlin.

Müller, M. and Wehner, R. (1988). Path integration in desert ants, *Cataglyphis fortis*. *Proceedings of the National Academy of Sciences of the United States of America*, 85:5287–5290.

Nolfi, S. and Floreano, D. (2002). Synthesis of autonomous robots through evolution. *Trends in Cognitive Sciences*, 6(1):31–37.

Nolfi, S., Miglino, O., and Parisi, D. (1994). Phenotypic plasticity in evolving neural networks. In *Proceedings of the PerAc '94 Conference*, pages 146–157, Los Alamitos, CA. IEEE, IEEE Computer Society Press.

Ofria, C., Adami, C., and Collier, T. C. (2002). Design of evolvable computer languages. *IEEE Transactions in Evolutionary Computation*, 17:528–532.

Ofria, C. and Wilke, C. O. (2004). Avida: a software platform for research in computational evolutionary biology. *Artificial Life*, 10(2):191–229.

Pennock, R. T. (2007). Models, simulations, instantiations, and evidence: the case of digital evolution. *Journal of Experimental and Theoretical Artificial Intelligence*, 19(1):29–42.

Rescorla, R. A. (1988). Behavioral studies of Pavlovian conditioning. *Annual Review of Neuroscience*, 11:329–352.

Shettleworth, S. J. (1998). *Cognition, evolution, and behavior*. Oxford University Press, New York.

Todd, P. M. and Miller, G. F. (1990). Exploring adaptive agency II: simulating the evolution of associative learning. In *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior on*, pages 306–315. MIT Press, Cambridge, MA, USA.

Vickerstaff, R. and Di Paolo, E. A. (2005). Evolving neural models of path integration. *Journal of Experimental Biology*, 208:3349–3366.

Zhang, S. W., Bartsch, K., and Srinivasan, M. V. (1996). Maze learning by honeybees. *Neurobiology of Learning and Memory*, 66:267–282.

Zhang, S. W., Lehrer, M., and Srinivasan, M. V. (1999). Honeybee memory: navigation by associative grouping and recall of visual stimuli. *Neurobiology of Learning and Memory*, 72:180–201.

Zhang, S. W., Mizutani, A., and Srinivasan, M. V. (2000). Maze navigation by honeybees: learning path regularity. *Learning and Memory*, 7:363–374.

Self-organized Segregation Effect on Self-Assembling Robots

Aubery Marchel Tientcheu Ngouabeu [1,2], Shuhei Miyashita [2], Rudolf M. Füchslin [2,3], Kohei Nakajima[2], Maurice Göldi [2], and Rolf Pfeifer [2]
[1] Technical University Munich
[2] Artificial Intelligence Laboratory, University of Zurich
[3] European Centre For Living Technology, Venice, Italy
aubery.tientcheu@mytum.de

Abstract

Complex systems involving many interacting components being out of equilibrium often organize into patterns. Understanding the underlying principles that govern such systems might lead to a deeper insight into living systems and the development of new applications in robotics. In this contribution, we investigate water-based self-assembling modules, exhibiting a segregation effect under some particular conditions. The system consists of vibrating (active) and non vibrating (passive) circular modules floating on the surface of the water. The segregation happens as a result of a depletion-like force, which is of purely entropic nature and is based on the characteristics of the modules (active or passive). We focus especially on the dynamics of the process with respect to the energy and the entropy. Some applications of the designed system are also discussed.

INTRODUCTION

Self-organization is one way by which nature builds artefacts at various scales. Nature offers diverse examples: the formation of molecular crystals [9], the folding of polypeptide chains into proteins [17], the folding of protein into their functional form [20], the cell's spontaneous organization into tissues [18], bacteria into colonies [10] [6], the formation of swarms (flock of bird or school of fish [23]) at a higher level, are commonly achieved in a distributed manner, where there is no central control system.

In the industry, as the aimed size of products decreases, people have started to recognize the advantages of self-organization in general and self-assembly in particular – which is typically approached in a bottom-up fashion. The potential capability as an alternative to replace traditional manipulating methods by self-assembly has been brought to attention. Standard manipulators have shown some limitations in the manipulation of nano-scaled components and there is a need for alternative methods with the miniaturization in the nanotechnology industry. *Nanogen Inc* employs electric field-mediated self-assembly to bring together DNA nanocomponents for electronic and diagnostic devices [13]. *Alien technology Corporation* uses self-assembly techniques like shape recognition or fluid transport to fabricate micro-scaled RFID tags [8][28].

One collective behavior that can emerge as result of local interaction is segregation, that is a spatial sorting method, where a group of objects occupies a continuous area of the environment which is not occupied by members of any other group. Segregation plays a key role in the food and drug processing industry. In particular, when shaking foods made of particles or granular material of different sizes, segregation effects occurs and the underlying mechanism is known as the *Brazil nut effect* or the *muesli effect* [24]. This spontaneous ordering goes against one's intuition that objects get mixed when merged in random directions and was described by Barker and Grimson in this way: "During the periods when shaking loosens the packing, individual small particles can move into voids beneath large particles and so prevent them from returning to their previous positions. It is far less probable that several small particles will move together so as to create a void that can be occupied by a single large particle. The net effect is that the smaller particles occupy the lower positions during the active part of the shaking process and then become trapped there when the grains fix into a new arrangement." [3]. A similar phenomenon takes place in the industrial production of drugs, thereby yielding considerable risks for patients (who are assumed to consume homogeneous mixings).

Many self-assembly and self-organizing systems have been suggested using different approaches, several of them inspired by biology. The best known example in this domain is probably the Reynolds flocks of birds [23], where different agents generate a flocking behavior by means of simple rules: collision avoidance, speed and heading matching and maintaining a close distance to the neighbor flock mates. The collision avoidance enables the agents to avoid colliding with each other; the second rule enables the agents to match their speed with their neighbors speed, whereas the third rule enables them to maintain a close distance to the neighboring birds. Reynolds simulations of the flock of bird show that these local interactions produce a global behavior similar to the flocks of birds we observe in the nature. Reynolds work doesn't only provide a tool to understand how the real flocks of birds achieve their global behavior but also help to de-

sign machines with formation control capabilities. Whitesides *et al.* assessed dynamic self-assembly would be one of the key challenges in building self-assembly systems [26] and in understanding life. Their suggestion relies on the fact that the most living systems are dynamic and understanding dynamic self-assembly would probably also leads to understanding life. Pfeifer *et al.* proposed a new approached in the design of robotics systems in general and living systems in particular. They suggested a synthetic approach taking morphological aspects into account [22].

There are three basic issues with this picture: (1) although little is known about the underlying assembly process, the fact that many living systems adopt similar mechanisms hints at common design principles suggesting that simplified models (such as the one presented in this paper) might be helpful in understanding the process; (2) even for a small cells, there are too many possible intermediates to allow a complete description of the assembly process with three independent stages [10]; and (3) a generalized scheme to avoid a substantial degree of incorrect assembly has to exist.

To date a few self-reconfigurable modular robots relying on stochastic self-assembly have been built [4][7]. White *et al.* studied two systems in which the modules binding preferences are coded in a program executed by an on-board microcontroller, and thus can easily reconfigure the structure [25]. The modules are initially unpowered and passive, but once they bind to a seed module connected to a power supply, they become active. Griffith *et al.* studied a system of template-replicating modules [12]. They used modules of the same type, which are programmable and can store distinct states. The system demonstrated the self-replication of a five modules polymer. Each module executed a finite-state machine. Klavins *et al.* examined the problems of designing a grammar that causes modules to assemble into desired products, of predicting the time complexity of such processes, and of predicting (and optimizing) the yield of such processes [15]. Emergent self-propulsion mechanisms were investigated by Ishiguro *et al.* [14]. In Ant-inspired robotics, the interest in self-organization has been driven by the observations of the same phenomena in ant colonies, in particular the brood sorting by *Temnothorax* [11]. *Wilson et al.* [27] created an algorithm to realize two colors annular sorting which used differential pull-back distances for different object types. By discriminating between three puck types, the robots could drop the first type of object on colliding with another puck, drop the second object type after pulling back a short distance and drop the third puck type after pulling back a further distance.

The Tribolon platform developed in our group is an example of a system using the morphology, which means the form and the shape of the involved components to get self-propelled robots to self-assemble [19]. Previouly, we carried out several experiments with circular sector shaped modules that can assemble to a single module. To overcome the restraint that the system has some difficulties to possess global information, the designer is supposed to consider the characteristics of the system and design new in/out scheme and apply an adequate controlling method to the robots. If the units move around by other means (e.g., by exploiting surface tension or by taking advantage of Brownian motion), the system is stochastically self-reconfigurable implying variable reconfiguration times and uncertainties in the knowledge of the units location (the location is known exactly only when the unit docks to the main structure). The advantages of this form of reconfiguration are at least two-folds: it can be extended to small scales, and it alleviates local power requirements.

In this paper, we show how segregation effects can be achieved on our platform. An important part of our modelling is the introduction of passive and active modules. We will see how these two types of particles successfully segregate and describe the dynamics of the segregation behaviour by discussing the center of mass of each cluster and the entropy of the system.

THE EXPERIMENTAL SETUP

The Model

The term self-assembly implies that the elements or parts involved assemble in a spontaneous manner without external intervention or control. Taking this into account, we chose to produce a set of modules with the same shape that swarm on water.

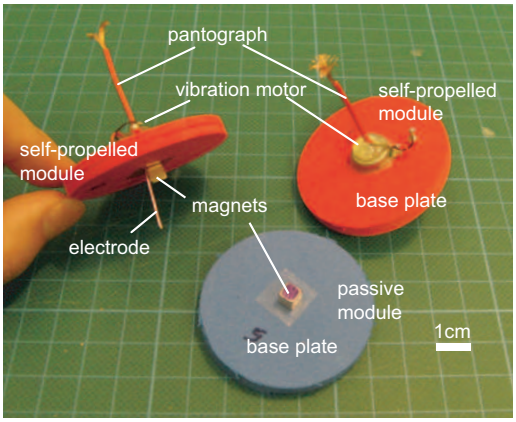


Figure 1: (a) Self-propelled and passive modules. Each module weighs $2.8\,g$ and has a footprint of $12.25\,cm^2$.

To conduct the experiments, we used the Tribolon platform [19] consisting of centimeter-sized modules floating on the water surface. All the modules are equipped with a permanent magnet attached at the bottom and aligned in a way so that they repel each other (north is always pointing up). Some of the modules are, in addition to the permanent magnet, also equipped with a vibration motor. In this paper,

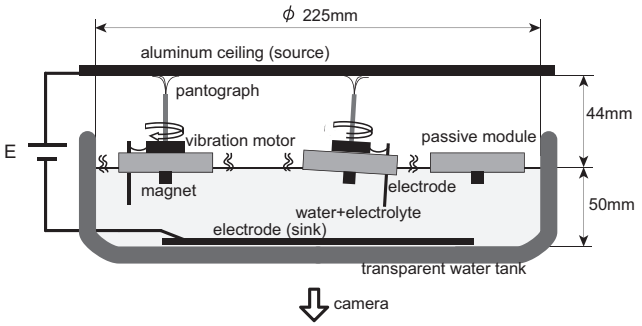


Figure 2: Illustration of the experimental environment with three modules.

we will denote a module provided with a vibration motor as vibrating or active module and a module only provided with a permanent magnet as passive module.

The vibrating modules are equipped with a flat coreless vibration motor (T.P.C DC MOTOR FM34F, $12000 \sim 14000\,rpm\,(2.5-3.5\,Volts)$) on the top of the base plate to allow self-propulsion, and all the modules with a single cubic permanent magnet (flux density $1.3\,T$, $5\times5\times5\,mm^3$, we decided that a single module should contain only one magnet) at the bottom for attractive/repulsive interactions (Fig. 2). This allowed the modules to jiggle and move around in their environment. A pantographic mechanism was used to supply the vibration motor with energy. When an electrical potential was applied to the ceiling plate (see Fig. 2), current flowed through the pantograph to the vibration motor was applied to the ceiling plate, current returning to ground via electrodes immersed in the conductive water.

Due to this setup, all modules receive the same constant power and they are be lightweight ($2.8\,g$ each), which would not be the case if batteries were used.

The Interaction Mechanism

Long-range interactions between two modules depend only on the force between the magnets on the tiles. We consider the magnets as dipoles with a magnetic moment $\boldsymbol{m}$.

The magnetic potential $\phi_j(\boldsymbol{r})$ at a position $\boldsymbol{r}$ due to the magnetic moment $\boldsymbol{m}_j$ is given by

$$\phi_j(\boldsymbol{r}) = \frac{\mu_0}{4\pi}\frac{\boldsymbol{m}_j\cdot\hat{\boldsymbol{r}}}{r^2} \tag{1}$$

where $\mu_0 = 4\pi\times10^{-7}Tm/A$ is the permeability of free space, and $\hat{\boldsymbol{r}} \equiv \boldsymbol{r}/|\boldsymbol{r}|$ assuming that $|\boldsymbol{r}| = r$ is much larger than the size of the magnet. The magnetic flux of the dipole is then given by

$$\boldsymbol{B}_j = -\nabla\,\phi_j \tag{2}$$

and the magnetic potential energy U_{ij} acquired by a second dipole $\boldsymbol{m}_i$ placed in the field of $\boldsymbol{m}_j$ is given by

$$U_{ij} = -\boldsymbol{m}_i\cdot\boldsymbol{B}_j. \tag{3}$$

Then, the force between the two dipoles is found by differentiating (3) with respect to $\boldsymbol{r}$.

$$\boldsymbol{F}_{ij} = (\boldsymbol{m_i}\cdot\nabla)\boldsymbol{B}_j \tag{4}$$

$$\boldsymbol{\tau}_{ij} = \boldsymbol{m}_i\times\boldsymbol{B}_j \tag{5}$$

We can determine the total potential energy of the system as

$$U_{total} = \frac{1}{2}\sum_{i,j\;i\neq j} U_{ij}. \tag{6}$$

Finally, we normalize the energy as $U'_{total} \equiv U_{total}/(\frac{\mu_0}{4\pi}m^2)$. The long range interaction described above is identical for each type of modules, since identical magnets were used. However, the short range interaction, i.e. the final alignment, is dominated by the non-linear dynamics and will be explain later in this paper.

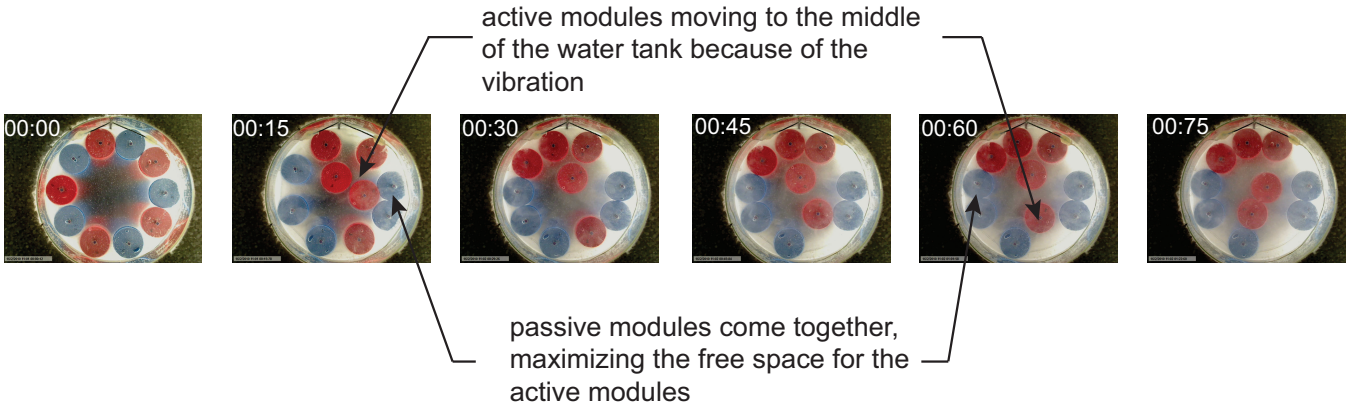


Figure 3: The experimental results in time sequence. The frames are captured every 15 seconds

THE EXPERIMENTAL RESULTS

The initial condition

In the following part, we investigate how designed system achieves a global segregation effect. Our experimental setup consists of ten modules, where five red colored modules are ”passive” and the remaining blue colored modules are ”active”, meaning the vibration motors are implemented. We conducted 15 trials for the statistical analysis (see section). In Fig. 3, we show a representative result in time sequence of the obtained segregation behavior. The initial starting condition was set as depicted in Fig. 3 (00:00), in which all the modules were symmetrically aligned in a circular form alternately, such that the passive and the vibrating modules have equal chances in the segregation process. This configuration also allows us to make a statistical analysis with similar starting conditions. The duration time for the experiment was set to 90 seconds.

Global Observations

In order to perform the analysis, fifteen experiments were conducted and the trajectories (positions) of all the modules were tracked using the open source tracking software ”Tracker Video Analysis and Modeling Tool” [5].

Our observation is that the red active modules tend to assemble together and go apart from the blue passive modules, such that two different modules clusters can be spatially distinguished; the first cluster contains only the active modules and the second cluster the passive modules (see Fig. 3 (00:75)).

In the following sections, we investigate the segregation behavior using statistical methods, by calculating the potential energy, the entropy and the centroids distance of the two clusters. The reader should notice that the calculated values for the entropy, the potential energy and the centroids are mean values over the fifteen experimental trials. The error bars represent the standard deviation of uncertainty within the fifteen experimental trials.

Potential Energy Transition

The magnetic potential energy of the system is defined in Eq. 6. We calculate the total magnetic potential energy of the system and show the obtained result in Fig. 4 presents the obtained result as function of the time.

Due to the characteristics of the system, non-equilibrium system, the value keeps changing. Suppose we have all passive modules, the system is supposed to reach to the state where modules are equally distributed and fixed.

The Centroid Distance

In this section, we investigate the cluster formation by computing the centroid of the system of the two clusters.

The centroid $(X, Y) = (\frac{1}{N}\sum_{i=1}^{N}(x_i), \frac{1}{N}\sum_{i=1}^{N}(y_i))$ of a group (or cluster) of modules is the center of mass of the modules, where N is the number of modules in the modules

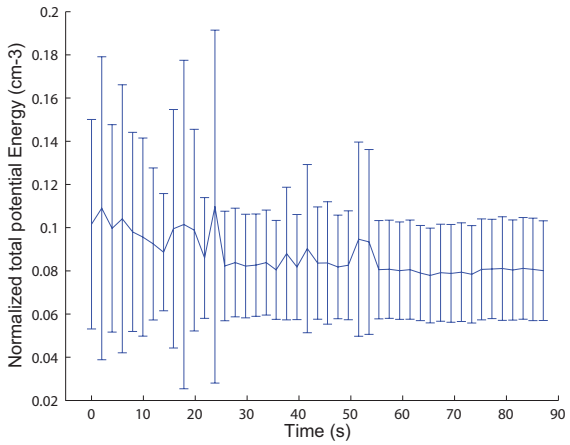


Figure 4: Total Energy of the system.

group, x_i and y_i are the positions of the i-th component of the considered group, respectively. We calculated the time evolution of the difference between the two modules groups (the passive modules on one side and the active modules on the second side and depicted in Fig. 5.

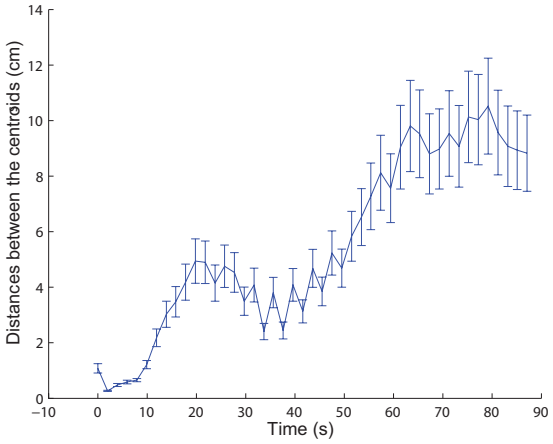


Figure 5: Time evolution of the distance to between the center of mass of the two clusters (N = 15).

As depicted in Fig. 5, there is an increase in the distance between the centroids of the passive and the vibrating modules. This corresponds to the formation of two clusters of modules with a final mean distance between the two clusters of approximately 10 centimeters. Given that the diameter of the arena (or tank as you wish) is 22.5 centimers, this corresponds to the 50% of the whole area.

Entropy

The definition of entropy differs in scientific fields, depending on to what one applies. Thermodynamics entropy (to

heat), statistical mechanics entropy (to object), and information entropy (to event) are probably the three best known entropies in science. In self-assembly, systems that cannot presume some specific physical amounts, such as quantity of heat, employ information entropy for the measurement of their "randomness".

Balch proposed a novel definition of entropy (position order) that can be applied for the measurement of multi-components distributions (or quantitative metric of diversity) [2]. He uses H from Shannon's theory

$$H(h) = -\sum_{i=1}^{N} p_i(h) \log_2(p_i(h)) \quad (7)$$

where p_i is the number of modules in the $i-th$ cluster ($i \in N$) divided by the total number of modules. A component belongs to a cluster if the distance is within the length of h ($||\vec{r_i} - \vec{r_j}|| < h$; $\vec{r_i}$ is the position of the i-th component). He then integrates $H(h)$ over all possible h, and defines it as entropy, namely:

$$S = \int_0^{\infty} H(h)dh. \quad (8)$$

The definition describes the randomness of modules well. Note that in this definition, the entropy may decreases over time. In physics, an entropic force acting in a system is a macroscopic force whose properties are primarily determined not by the character of a particular underlying microscopic force (such as electromagnetism), but by the whole system's statistical tendency to increase its entropy. We examined the entropy of the system as derived as in Eq. 8. Fig. 6 shows the time evolution of the entropy of the system.

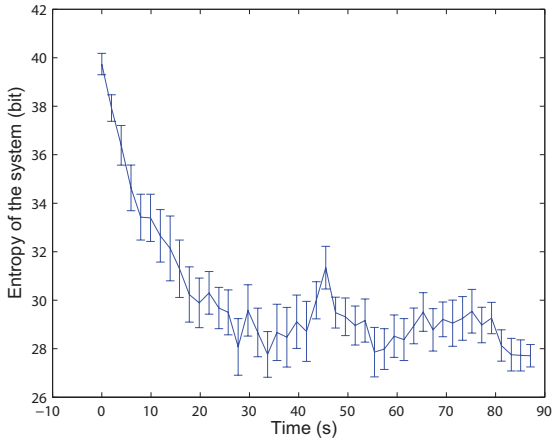


Figure 6: The Transition of Entropy.

As we can observe, the entropy of the system is decreasing as time progresses, which represents the convergence of the system to more ordered configurations. This corresponds to the cluster formation described of the previous section.

DISCUSSIONS

Depletion Effect

In this section, we speculate the main cause of the segregation effect. Fig. 7 illustrates the exclusive regions of modules, where different module have difficulty in lying in the area around another module due mainly to the magnetic repulsive forces. When the passive modules are closed to the wall, the excluded area for the passive modules and the wall overlap (shaded region) and this causes the reduction of the total excluded area. Now the extra area is left for the vibrating modules. As shown here, the overlap is larger when the passive modules are placed next to the curved portion

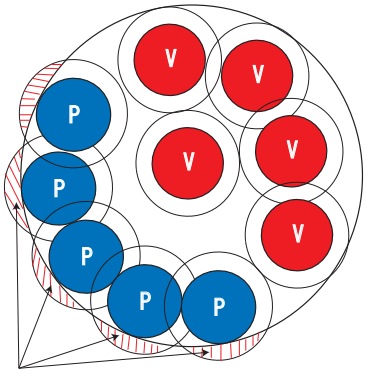


Figure 7: Illustration of the excluded area of the passive modules.

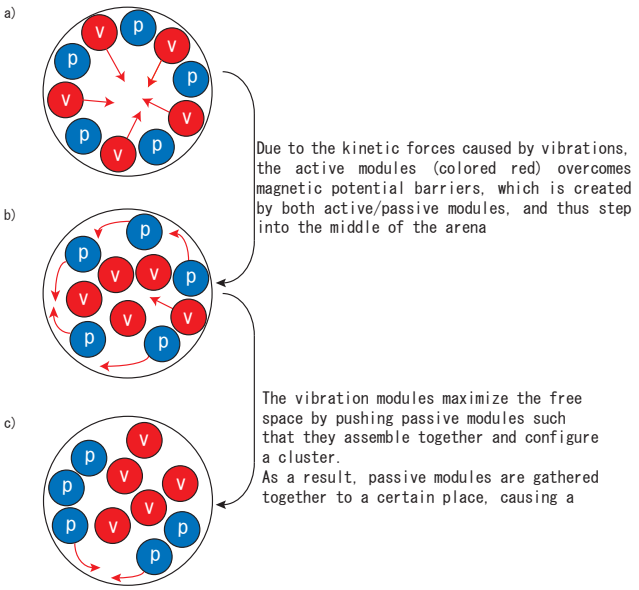


Figure 8: Explanation of the transitions in the experiments

of the wall compared to it being in the middle of the water tank. In the experiments, the vibration motion acts as an effective short range repelling potential, which results in the observed separation of the passive modules, and in consequence an effective attraction between the passive modules. In nature, depletion effects, which is also called exclusion effect are observed at all length scales; especially at the molecular scale, it can be described from a statistical mechanics point of view as a minimization of free energy.

The careful observation of the segregation process is described in Fig. 8. At the initial stage (Fig. 8 a), the vibrating modules tend to go to the middle of the water tank, due to the vibration. In a further step, the vibrating modules maximize their free space by pushing the passive modules to one side of the water tank (Fig. 8 b). The free-space reaches its maximum when all the passive modules are close together and there is no blank space between them. The passive modules move towards the wall (as illustrated in Fig. 7). In that way, the free area available to the vibrating modules is larger if a large module is placed next to the curved surface of the wall, than if it is in the middle of the water tank.

A similar segregation effect is observed in granular mixtures and is known in physics as depletion effect. The segregation criteria can be the size, the shape, the mass or some frictional coefficients and can be caused by several mechanisms, including vibration, percolation, convection and tumbling [16] [21]. The force created by the vibrating modules, which pushes the passive modules together and increases the space available for the vibrating modules, is called depletion force. This force, which is purely entropic in origin has been predicted by Asakura and Oosawa [1] and confirmed since then by several experiments. Other work on both experiments and simulations were conducted using passive modules mostly of different sizes and have shown, that a similar segregation can be produced by shaking mixtures of different sizes vertically ([24]). This underlying effect is called the *Brazil nut effect* and big particles, seem to move to the top, while smaller particles move to the bottom.

Properties of the system

The particularity of our experiments is that it is conducted at the centimeter size, , and not to mention, which helps to observe and investigate the phenomena directly using simple observation tools (i.e. visual tracking for example) compared to the experiments at smaller scales. Furthermore, our experiments were conducted in two dimension utilizing also vibrating modules; there is no microcontroller, no sensing, we only exploit the dynamic interaction between the modules to achieve the segregation. This way of proceeding is unusual in distributed system's robotics, where one mostly use distributed algorithms and local rules to reach global patterns.

The advandtage of distributed systems and the potential applications

Realizing controlled global segregation behavior of distributed modules offers various applications; here we highlight self-healing capabilities. A system containing a large amount of locally interacting (and cooperating) microcomponents offers considerable problems with respect to maintenance (removing of damaged components as well as recharging). If proper functioning is correlated with segregation behavior, non-functional modules may tend autonomously to the edge of the container where they can be replaced or recharged. Conceptually, this means that at least parts of the control of the maintenance process are embodied in the system. Future production processes may rely on swarms of agents, probably of different morphology and function. Tunable segregation mechanisms offer a potential for inducing a variety of different patterns of the agents under consideration, yielding an additional option for programming swarm based production processes.

Finally, studies of the type presented here may shed light on, in an industrial context, highly relevant class of segregation processes in mixtures of objects of different morphology. Examples are e.g. the Brazil nut effect, but also various types of sieving processes (in which the basically passive granules take up energy from a shaking table in a way that depends on their respective morphology).

CONCLUSIONS AND FUTURE WORK

We proposed a stochastic self-assembly system in which a segregation effect emerges as a result of local non-linear interactions between the modules of the system. The system involves passive and active vibrating modules, that randomly move on water in a purely distributed way. By analyzing fifteen experimental trials with statistical methods on a real setup, we have shown the expected segregation behavior, in which passive and active modules induced formed groups, hence causing a segregation behavior. We believe that understanding dynamic self-assembly will play a key role in the development of small-scaled modular robots and will offer new opportunities to deepen both the realization and the theoretical understanding of self-assembly systems. Furthermore, some of the principles discovered especially concerning the dependence of self-organization on the dynamic interaction between the modules might lead to a better understanding of similar processes found in natural systems and of life in general.

ACKNOWLEDGMENTS

This research is partially supported by the Swiss National Science Foundation project #200020-118117/1.

References

Asakura, O. (1954). On interaction between two bodies immersed in a solution of macromolecules. *The Journal Of Chemical Physics*, 22:1255.

Balch, T. (2000). Hierarchic social entropy: An information theoretic measure of robot group diversity. *Autonomous Robots*, 8:209–237.

Barker, G. and Grimson, M. (1990). The physics of muesli. *New Scientist*, 126:37–40.

Bowden, N., Terfort, A., Carbeck, J., and Whitesides, G. M. (1997). Self-assembly of mesoscale objects into ordered two-dimensional arrays. *Science*, 276:233–235.

Brown, D. (2009). Tracker video analysis and modeling tool. http://www.cabrillo.edu/ dbrown/tracker/.

Budrene EO, B. H. (1995). Dynamics of formation of symmetrical patterns by chemotactic. *Nature*, 376(6535):49–53.

Castano, A., Behar, A., and Will, P. M. (2002). The conro modules for reconfigurable robots. *IEEE/ASME Trans. on Mechatronics*, 7(4):403–409.

Corporation, A. T. (1994). Alien technology website. http://www.alientechnology.com/products.

Desiraju, G. (1989). *Crystal Engineering: The Design of Organic Solids*. Elsevier.

Elena O. Budrene, H. C. B. (1991). Complex patterns formed by motile cells of escherichia coli. *Nature*, 349:630 – 633.

Franks, Nigel R., S.-F. A. B. S. S. R. M. C. (2004). Brood sorting by ants: Two phases and differential diffusion. *Elsevier B.V.*, 68:1095–1106.

Griffith, S., Goldwater, D., and Jacobson, J. (2005). Robotics: Self-replication from random parts. *Nature*, 437:636.

Heller Michael J.; Cable, Jeffrey M.; Esener, S. C. (2003). Methods for the electronic assembly and fabrication of devices. *US Patent 6652808*.

Ishiguro, A., Shimizu, M., and Kawakatsu, T. (2006). A modular robot that exhibits amoebic locomotion. *Robotics and Autonomous Systems*, 54:641–650.

Klavins, E. (2007). Programmable self-assembly. *IEEE Control System Magazine*, 27:43–56.

Kudrolli, A. (2004). Size separation in vibrated granular matter. *Reports on Progress in Physics*, 67(3):209.

Maginn, S. J. (1991). Crystal engineering: the design of organic solids by g. r. desiraju. *Applied crystallography online*, 24:265–265.

Markwald, K. J. A. N. V. M. R. and Forgacs, G. (2004). Engineering biological structures of prescribed shape using self-assembling multicellular systems. *Proc. Natl. Acad. Sci. USA*, 101:2864–2869.

Miyashita, S., Kessler, M., and Lungarella, M. (2008). How morphology affects self-assembly in a stochastic modular robot. In *IEEE International Conference on Robotics and Automation*.

Neidle, S. (1999). *Oxford Handbook of Nucleic Acid Structure*. Oxford University Press.

Ottino, J. M. and Khakhar, D. V. (2000). Mixing and segregation of granular materials. *Annual Review of Fluid Mechanics*, 32:55–91.

Pfeifer, R., Lungarella, M., and Iida, F. (2007). Self-organization, embodiment, and biologically inspired robotics. *Science*, 318:1088–1093.

Reynolds, C. W. (1987). Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics*, 21:25–34.

Rosato, A., Strandburg, K. J., Prinz, F., and Swendsen, R. H. (1987). Why the brazil nuts are on top: Size segregation of particulate matter by shaking. *Phys. Rev. Lett.*, 58(10):1038–1040.

White, P., Kopanski, K., and Lipson, H. (2004). Stochastic self-reconfigurable cellular robotics. In *Proc. Int. Conf. on Robotics and Automation*, volume 3, pages 2888–2893.

Whitesides, G. M. and Grzybowski, B. (2002). Self-assembly at all scales. *Science*, 295:2418–2421.

Wilson M., Melhuish C., S.-F. A. S. S. (2002). Multi-object segregation: ant-like brood sorting using minimalism robots. *Proc. Seventh International Conf. on the Simulation of Adaptive Behaviour, Edinburgh, UK*, page 369370.

Yeh, H. and Smith, J. (1994). Fluidic self-assembly for the integration of gaas light-emitting diodes on si substrates. *Photonics Technology Letters, IEEE*, 6(6):706 –708.

Early Nervous Systems: Theoretical Background and a Preliminary Model of Neuronal Processes

Ot de Wiljes[1], Ronald A.J. van Elburg[1], Michael Biehl[2] and Fred Keijzer[3]

[1]Department of Artificial Intelligence, University of Groningen, The Netherlands
[2]Johann Bernouille Institute for Mathematics and Computer Science, University of Groningen, The Netherlands
[3]Department of Theoretical Philosophy, University of Groningen, The Netherlands
f.a.keijzer@rug.nl

Extended Abstract

The evolution of the earliest nervous systems remains seriously under-researched. Within this small field, the focus has so far been mostly on the evolution of nerve cells, nervous system centralization and biomolecular precursors of nerve cells (Lichtneckert & Reichert, 2007). Another line of research concerns the geological and molecular evidence on ecological and morphological changes that may have contributed to the development of nervous systems in Precambrian life (Dzik, 2005; Peterson et al., 2005).
An important open question is how the very first nervous systems might have worked as a behavior producing system. The classic assumption, dating back to Parker's (1919), is that nerve cells evolved to connect pre-existing sensors and effectors, a proposal that was strongly influenced by Sherrington's exposition of the reflex-organization in vertebrates. Nervous systems are here a connecting device that gradually became more complex by adding feedback loops and cognitive extensions (Braitenberg, 1984).
However, this standard interpretation does not combine easily with other findings within this field. For example, many authors (e.g. Pantin, Passano, Horridge, Pavans de Ceccaty) claim that reflexes are a secondary development on top of a more primitive arrangement. The most basic examples of nervous systems are loosely connected nerve nets – skin brains (Holland, 2003) – spread out over the body without fast and specialized connections between specific sensors and effectors. A long neglected suggestion, going back to Pantin (1956), is that early nerve nets contributed foremost to the organization of patterns of muscle contractions in large multicellular animals. Coordinated muscle contractions allowed large animals to move about when earlier mechanisms, like ciliary crawling, became too inefficient. Under this interpretation, the key innovative function of early nervous systems is primarily to generate larger-scaled effectors rather than connecting sensors to some pre-existing 'effector'.

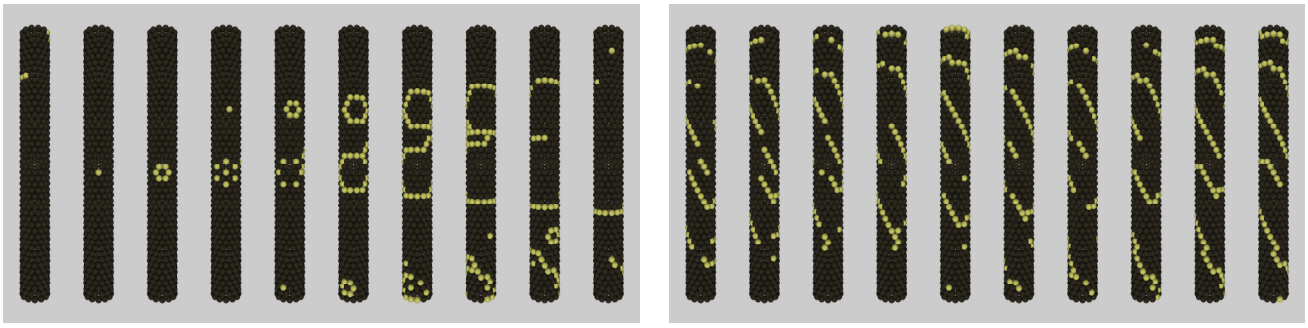


Figure: Emergent patterns on a simulated skin brain. Left: a simulation where every cell is connected to all six neighbors. Right: a simulation where every cell is connected to three, out of six, neighbors, forcing the spontaneous patterns to travel from bottom to top.

Our model investigates the transition from a non-neural conductive epithelium (Mackie, 1970) to a basic nerve net. A basic tube-like animal structure is approximated as a single sheet of cells that are both contractile and electrically conductive. Epithelial conduction produces spontaneous electrical activity on the bodily surface. We modelled the transition to nerve nets by varying three parameters: (a) Increasing the number of cells mimics increasing body-size. (b) Directionality of signalling, representing the evolution of synapses, makes cells in the model signal only in specific directions. (c) Formation and elongation of cell processes,

representing the early evolution of axons and dendrites, allows cells to signal to non-neighbouring cells without influencing cells in between. The two last parameters represent key-aspects of neurons and the model provides a platform to investigate how these parameters modify global activity patterns at different body-sizes. The findings are relevant for a better understanding of the basic operation of nervous systems, early nervous system evolution and the problems encountered in the field of soft robotics.

References

Braitenberg, V. (1984). *Vehicles: Experiments in Synthetic Psychology*. MIT Press, Cambridge, MA:.

Dzik, J. (2005). Behavioral and anatomical unity of the earliest burrowing animals and the cause of the "Cambrian explosion". *Paleobiology,* 31(3): 503-521.

Holland, N. (2003). Early central nervous system evolution: An era of skin brains? *Nature Reviews Neuroscience*, 4(8): 617-627.

Lichtneckert, R. & Reichert, H. (2007). Origin and evolution of the first nervous system. In Kaas, J.H., Striedter, G.F. and Rubinstein, J.L.R., editors, *Evolution of Nervous Systems, Vol. 1: Theories, Development, Invertebrates* , pages 289-315. Elsevier, Amsterdam.

Mackie, G.O. (1970). Neuroid conduction and the evolution of conducting tissues. *Quarterly Review of Biology*, 45: 319-332.

Pantin, C.F.A. (1956). The origin of the nervous system. *Pubblicazioni della Stazione Zoologica di Napoli,* 28: 171-181.

Parker, G. (1919). *The Elementary Nervous System*. Lippincott: Philadelphia.

Peterson, K. J., McPeek, M.A., & Evans, D.A.D. (2005). Tempo and mode of early animal evolution: Inferences from rocks, Hox, and molecular clocks. *Paleobiology*, 31(2): 36-55.

Solving Mazes using an Artificial Developmental Neuron

Gul Muhammad Khan[1] and Julian F. Miller[2]

[1]NWFP UET Peshawar, Pakistan, gk502@nwfpuet.edu.pk
[2]University of York, UK
jfm7@ohm.york.ac.uk

Abstract

An agent controlled by a single computational neuron is used to solve maze problems. The neuron has activity and time-dependent computational and topological structure. The behaviour of a neuron is controlled by a collection of seven evolved programs that are loosely analogous to aspects of biological neuron (dendrites, soma, axons, synapses, electrical and developmental behaviour). The programs are represented using Cartesian Genetic Programming. Our aim is to show that it is possible to evolve programs that develop a *single* neuron so that it is able to learn how to solve maze problems purely by experience.

Introduction

Although many techniques have been introduced to develop Artificial Neural Networks (ANNs) using genetic programming, we found no evidence that an attempt has been made to develop the *functional* model of real neurons with biological morphology. We have attempted to do this by devising an abstraction of real neurons which captures many important features. Various studies have shown that "dendritic trees enhance computational power" (Koch and Segev (2000)). Neurons communicate through synapses which are not merely the point of connection between neurons (Kandel et al. (2000)). They can change the strength and shape of the signal over various time scales. We have taken the view that the time dependent and environmentally sensitive variation of morphology and many other processes of real neurons is very important and richer models are required that incorporate these features. In our model a neuron consists of a soma, dendrites, axons with branches and dynamic synapses and synaptic communication. Neurite branches can grow, shrink, self-prune, or produce new branches. This allows it to arrive at a network whose structure and complexity is related to properties of the learning problem.

Our aim is to find a set of computational functions that encode neural structures with an ability to learn through experience. Such neural structure would be very different from conventional ANN models as they are self-training and constantly adjust themselves over time in response to external environmental signals. In addition they could grow new networks of connections when the problem domain required it.

From our studies of neuroscience, we have identified seven essential computational functions that need to be included in a model of a neuron and its communication mechanisms. From this analysis we decided what kind of data these functions should work with and how they should interact, however we cannot design the functions themselves. So we turned to a well established and efficient form of Genetic Programming called Cartesian Genetic Programming (CGP) (Miller and Thomson (2000)).

We have tested the learning capability of this developmental system on maze problems. A maze is a complex tour puzzle with a number of passages and obstacles (impenetrable barriers). It has a starting point and an end point. The job of the agent is to find a route from starting point to the end point. The agent starts with a limited energy that increases and decreases as a result of interaction with the paths and the obstacles in the maze environment. We show that the agent is able to solve the maze a number of times in a single life cycle. The agents start a maze with a single neuron having random structure. However, the branching structure of the neuron can grow and shrink during the game environment.

In previously work, we evaluated the effectiveness of this approach on a classic AI problem called wumpus world (Khan et al. (2007)). There we used a number of neurons to solve the wumpus world. We have also tested the network of CGP neurons for playing Checkers (Khan and Miller (2009)). We found that the agents improved with experience and exhibited a range of intelligent behaviours. In this paper we have turned our attention toward a single neuron. The motivation for this was to explore the capability of a single neuron in this model.

Biology of Neuron

Neurons are the main cells responsible for information processing in the brain. They are different from other cells in the body not only in term of functionality, but also in biophysical structure (Kandel et al. (2000)). They have different shapes and structures depending on their location in the

brain, but the basic structure of neurons is always the same. They have three main parts.

- Dendrites (Inputs): Receive information from other neurons and transfer it to the cell body. They have the form of a tree structure, with branches close to the cell body.

- Axons (Outputs): Transfer the information to other neurons by the propagation of a spike or action potential. Axons usually branch away from the cell body and make synapses (connections) onto the dendrites and cell bodies of other neurons.

- Cell body (Processing area or Function): This is the main processing part of neuron. It receives all the information from dendrite branches connected to it in the form of electrical disturbances and converts it into action potentials, which are then transferred through axon to other neurons. It also controls the development of neurons and branches.

Neural modeling

A number of techniques are used for simulation of neural development either in the form of construction algorithms or biologically-inspired growth processes. One approach aims to reproduce the geometrical properties of real neurons and does not consider the actual biological processes responsible for neural growth that could be used in an electrophysiology simulator (Stiefel and Sejnowski (2007)). Lindenmayer-System have been used to invent the procedure for modeling plant branching structures (Lindenmayer (1968)) and later has been successfully applied to develop neural morphologies (Ascoli et al. (2001)). A number of other methods such as probabilistic branching models (Kliemann (1987)), Markov models (Samsonovich and Ascoli (2005)) and Monte Carlo processes (da Fontoura Costa and Coelho (2005)) are also proposed as construction algorithm for neural development. Although these methods produce interesting neuronal shapes, they do not provide any insight into the fundamental growth mechanisms for neuronal growth. Growth models on the other hand provide the biological mechanisms responsible for generation of neuronal morphology. A number of interesting agent-based simulations are produced that highlights various aspects of biological development, such as cell proliferation (Al-Musa et al. (1999)), polarization (Samuels et al. (1996)), neurite extension (Kiddie et al. (2005)), growth cone steering (Krottje and van Ooyen (2007)) synapse formation (Stepanyants et al. (2008)) and axon guidance and map formation (de Gennes (2007)).

Although these methods introduce various interesting techniques to model the neuronal growth which is the early stage of development of brain, they have not consider the signal processing aspects and its effect on the growth during interaction with the world via sensory mechanisms. We introduce the method of evolving the functions that are responsiple for neuronal growth, signalling and synapse formation during the lifetime of the agent as explained in later sections.

Computational Development

In biology, multicellular organisms are built through developmental process from 'relatively simple' gene structures. The same technique could be used in computational development to produce complex systems from simpler systems that are capable of learning and adapting (Stanley and Miikkulainen (2003)).

Quartz and Sejnowski proposed a powerful manifesto for the importance of dynamic neural growth mechanisms in cognitive development (Quartz and Sejnowski (1997)). Marcus emphasized the importance of growing neural structures using a developmental approach (Marcus (2001)).

Parisi and Nolfi suggested that if neural networks are viewed in the biological context of artificial life, they should be accompanied by genotypes which are part of a population and inherited from parents to offspring (Parisi and Nolfi (2001)). They have used a growing encoding scheme to evolve the architecture and the connection strengths of neural networks. The network consists of a collection of artificial neurons distributed in 2D space with growing and branching axons. The genetic code inside them specifies the instructions for axonal growth and branching in neurons.

Cangelosi proposed a neural development model, which starts with a single cell that undergoes a process of cell division and migration until a collection of neurons arranged in 2D space is developed (Cangelosi et al. (1994)). At the end, neurons grow their axons to produce connection among each other until a neural network is developed. The rules for cell division and migration are stored in genotype, for a related approach see (Dalaert and Beer (1994)). Gruau also proposed a similar method (Gruau (1994)). The genotype used in Gruau's model is in the form of a binary tree structure as in GP (Koza (1992)).

Rust and Adams have used a developmental model coupled with a genetic algorithm to evolve parameters that grow into artificial neurons with biologically-realistic morphologies (Rust et al. (2000)). Jakobi created an impressive artificial genome regulatory network, where genes code for proteins and proteins activate (or suppress) genes (Jakobi (1995)). The proteins define neurons with excitatory or inhibitory dendrites. The individual cell divides and moves due to protein interactions causing a complete multicellular network to develop. Federici presented an indirect encoding scheme for development of a neuro-controller and compared it with a direct scheme (Federici (2005)). He implemented the system on a Khepera robot and tested it using direct and indirect encoding schemes, finding that the latter reached high fitness faster.

Downing favors a higher abstraction level in neural de-

velopment to avoid the complexities of axonal and dendritic growth while maintaining key aspects of cell signaling, competition and cooperation of neural topologies in nature (Downing (2007)). He tested it on a simple movement control problem known as *starfish*. The task for the k-limbed animate is to move away from its starting point as far as possible in a limited time, producing encouraging preliminary results.

One of the major difficulties in abstracting neuroscience is that one can lose the essential aspects required to make a powerful learning system. However the evidence of importance of time-dependent morphological processes in learning is highly compelling and we have thus included many of these aspects in a model of an artificial neuron.

The Neuron Model

This section describes the Cartesian Genetic Programming (CGP) and details the structure and processing inside the CGP Neuron and the way inputs and outputs are interfaced with it.

Cartesian Genetic Programming (CGP)

CGP is a well established and effective form of Genetic Programming. It represents programs by directed acyclic graphs (Miller and Thomson (2000)). The genotype is a fixed length list of integers, which encode the function of nodes and the connections of a directed graph. Nodes can take their inputs from either the output of any previous node or from a program input (terminal). The phenotype is obtained by following the connected nodes from the program outputs to the inputs. The function nodes used here are variants of binary if-statements known as 2 to 1 multiplexers (Miller et al. (2000)).

In CGP an evolutionary strategy of the form $1 + \lambda$, with λ set to 4 is often used (Miller et al. (2000)). The parent, or elite, is preserved unaltered, whilst the offspring are generated by mutation of the parent. If two or more chromosomes achieve the highest fitness then *newest* (genetically) is always chosen. We have used this algorithm in the work we report here.

Health, Resistance, Weight and Statefactor

Four variables are incorporated into the CGP Neuron, representing either fundamental properties of the neuron (*health*, *resistance*, *weight*) or as an aid to computational efficiency (*statefactor*). The values of these variables are adjusted by the CGP programs.

The *health* variable is used to govern replication and/or death of dendritic and axonal connections. The *resistance* variable controls growth and/or shrinkage of dendrites and axons. The *weight* is used in calculating the potentials in the network. Each soma has only two variables: *health* and *weight*. The *statefactor* is used as a parameter to reduce computational burden, by keeping neuron and branches inactive for a number of cycles. Only when the *statefactor* is zero are the neuron and branches are considered to be active and their corresponding program is run. *Statefactor* is affected indirectly by CGP programs.

Inputs, Outputs and Information Processing inside CGP Neuron

The signal is transferred to and taken from this neuron using virtual axon and dendrite branches by making synaptic connections.

The signal from the environment is applied to CGP neuron using five virtual input axo-synaptic connections. Five virtual output dendrite branches are used to decide the movement of the agent. The virtual axo-synaptic branches are allowed to not only transfer signals to the dendrite branches of processing neuron (CGP Neuron) but also to the output virtual dendrite branches which decide the movement of the agent. The CGP Neuron transfers signals to the virtual output dendrite branches using the program encoded in the axo-synaptic chromosome.

Information processing in the CGP Neuron starts by selecting the list of dendrites and running the electrical dendrite branch program. The updated signals from dendrites are averaged and applied to the soma program along with the soma potential. The soma program is executed to get the final value of soma potential, which decides whether a neuron should fire an action potential or not. If soma fires, an action potential is transferred in forward direction using axo-synaptic branch programs.

Functionality of CGP Neuron

The CGP Neuron is placed at a random location in a two dimensional spatial neural grid (as shown in figure 1). It is initially allocated a random number of dendrites, dendrite branches, one axon and a random number of axon branches. Neurons receive information through dendrite branches, and transfer information through axon branches to neighbouring dendrite branches. The branches may grow or shrink and move from one neural grid location to another. They can produce new branches and can disappear. Axon branches transfer information only to dendrite branches in their proximity. Electrical potential is used for internal processing of neurons and communication between neuron and is represented by an integer (32 bit).

Neural functionality is divided into three major categories: electrical processing, life cycle and weight processing. These categories are described in detail below.

Electrical Processing The electrical processing part is responsible for signal processing inside neuron and communication between neurons. It consists of dendrite branch, soma, and axo-synaptic branch electrical chromosomes.

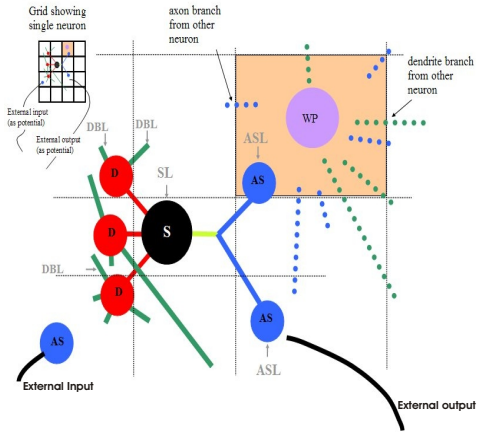


Figure 1: On the top left a neural grid is shown containing a single neuron. The rest of the figure is an exploded view of the neuron is given. Electrical processing parts: dendrite (D), soma (S) and axo-synapse branch (AS) are shown as part of neuron. Developmental programs responsible for the *life-cycle* of neural components are also shown (shown in grey). These are dendrite branch life (DBL), soma life (SL) and axo-synaptic branch life (ASL). The weight processing program (WP) is used to adjusts synaptic and dendritic weights.

The dendrite program D, handles the interaction of dendrite branches belonging to a dendrite. It take active dendrite branch potentials and soma potential as input and updates their values. The *Statefactor* is decreased if the update in potential is large and vice versa.

If any of the branches are active (statefactor equal to zero), their life cycle program (DBL) is run, otherwise D continues processing the other dendrites.

The soma program S, determines the final value of soma potential after receiving signals from all the dendrites. The processed potential of the soma is then compared with the threshold potential of the soma, and a decision is made whether to fire an action potential or not. If it fires, it is kept inactive (refractory) for a few cycles by changing its *statefactor*, the soma life cycle chromosome (SL) is run, and the firing potential is sent to the other neurons by running the AS programs in axon branches.

AS updates neighbouring dendrite branch potentials and the axo-synaptic potential. The *statefactor* of the axosynaptic branch is also updated. If the axo-synaptic branch is active its life cycle program (ASL) is executed.

After this the weight processing program (WP) is run which updates the *Weights* of neighbouring (branches sharing same neural grid square) branches.

Life Cycle of Neuron This part is responsible for replication, death, growth and migration of neurite branches. It consists of three life cycle chromosomes responsible for the neurites development. The two branch chromosomes update *Resistance* and *Health* of the branch. Change in *Resistance* of a neurite branch is used to decide whether it will grow, shrink, or stay at its current location. The updated value of neurite branch *Health* decides whether to produce offspring, to die, or remain as it was with an updated *Health* value. If the updated *Health* is above a certain threshold it is allowed to produce offspring and if below certain threshold, it is removed from the neurite. Producing offspring results in a new branch at the same neural grid square connected to the same neurite (axon or dendrite). The soma life cycle chromosome produces updated values of *Health* and *Weight* of the soma as output.

Maze

A maze is a term used for complex and confusing series of pathways. It is an important subject for autonomous robot navigation and route optimization (Tani (1996); Blynel and Floreano (2003)). The idea is to teach an agent to navigate through an unknown environment and find the optimal route without having prior knowledge. A simplified version of this problem can be simulated by using a random two-dimensional synthetic maze. The pathways and obstacles in a maze are fixed.

Experimental Setup

In our experiments an agent is provided with CGP Neuron as its computational network. The job of the agent is to find routes from a starting point toward an end point of a maze as many times as it can in a single life cycle. We have used a 2D maze representation for this experiment as shown in figure 2. The 2D Maze representation is explored in a number of scenarios (Werbos and Pang (1996); Ilin et al. (2007)). We have represented the maze as a rectangular array of squares with obstacles and pathways (As shown in the figure 2). A square containing an obstacle cannot be occupied. Movement is possible up or down on squares on the outside columns. Movement is either left or right on rows, unless there is a pathway, in which case downward motion is possible. This is inspired by the clustering approach used to improve learning capabilities of an agent (Mannor et al. (2004)). We used different sizes of mazes to test the ability of the agent. The location of the obstacles, pathways and exit are chosen randomly for different experimental scenarios.

Energy of Agent The agent is assigned a quantity called energy, which has an initial value of 50 units. If an agent attempts to penetrate an obstacle its energy level is reduced by 5 units. If it encounters a pathway and moves to a row closer to the exit, its energy level is increased by 10 units. If it moves a row further away from the maze exit, its energy is reduced by 10 units. This is done to enhance the learning capability of agent by giving it a reward signal. If the agent

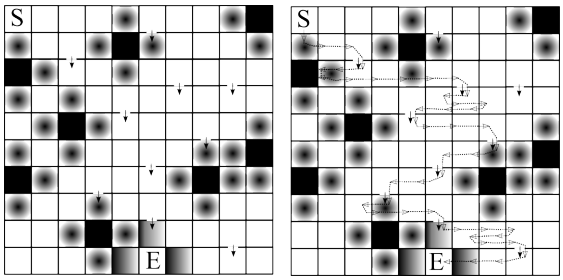


Figure 2: The left figure shows a 10x10 maze with impenetrable obstacles (black), downward pathways (arrows), start (S) and exit point (E), and their corresponding signals. On the neighbouring squares of an obstacle (north, south, east and west) and the exit there is a signal detectable by the agent indicating whether the agent is on a square neigbouring an obstacle (radial shading) or exit(linear shading). The figure on the right shows the path of an evolved agent.

reaches the exit, its energy level is increased by 50 units and it is placed back at the starting point and allowed to solve the maze again. Finally, if the agent arrives home, without having reached the exit, the agent is terminated. For each single move, the agent's energy level is reduced by 1 unit, so if the agent just oscillates in the environment and does not move around and acquire energy through solving tasks, it will run out of energy and die.

Fitness Calculation The fitness value, which is used in the evolutionary scheme, is accumulated while the agent's energy is greater than zero as follows:

- For each move, increase fitness by one. This is done, to encourage the agents to have 'brain' that remains active and does not die.
- Each time the agent reaches the exit, its fitness is increased by 100 units.

Inputs to neuron The maximum allowed neural potential is $M = 2^{32}-1$. The agent's input axo-synapses can perceive input potentials, I, depending on the circumstances in the following way. Note that the agent can perceives *only one* signal on a maze square, even if there are more than one.

- $I = 0$ default.
- $I = M/60$ finds a pathway to a row closer to exit.
- $I = M/120$ tries to land on obstacle.
- $I = M/200$ on exit square.
- $I = M/100$ adjoining square north of an obstacle.
- $I = M/110$ adjoining square east of an obstacle.
- $I = M/130$ adjoining square south of an obstacle.
- $I = M/140$ adjoining square west of an obstacle.
- $I = M/180$ approaches exit from north direction
- $I = M/190$ approaches exit from east direction
- $I = M/210$ approaches exit from south direction
- $I = M/220$ approaches exit from west direction
- $I = M/255$ home square (starting point)

Agent movement and termination When the experiment starts, the agent takes its input from the starting point (on the top left corner as shown in figure 2). This input is applied to the computational network (CGP Neuron) of the agent using input axo-synapses. The network is then run for five cycles (one step). During this process it updates the potentials of the output dendrite branches. After the step is complete the updated potentials of all output dendrite branches are noted and averaged. The value of this average potential decides the direction of movement for the agent. If there is more than one direction the potential is divided into as many ranges as possible movements. For instance if two possible directions of movement exist, then it will take one direction if the potential is less than $(M/2)$ and the other if greater. The same process is then repeated for the next maze square. The agent is terminated if either its energy level becomes zero or if it returns home.

CGP Neuron Setup The various parameters of CGP neuron are chosen as follows. The neuron's branches are confined to 3x3 CGPN neural grid. Inputs and outputs to the network are located at five different random squares. The maximum number of dendrites is 5. The maximum branch *statefactor* is 7. The maximum soma *statefactor* is 3. The mutation rate is 2%. The maximum number of nodes per chromosome is 100. Maximum number of dendrite and axon branches are hundred and twenty respectively. These parameters have not been optimized and have largely been chosen as they work reasonably well and do not incur a prohibitive computational cost.

Difficulty of the problem

It is important to appreciate how difficult this problem is. The agents starts with a single neuron with random connections. Evolution must find a series of programs that build a computational neural structure that is stable (not lose all branches etc.). Secondly, it must find a way of processing infrequent environmental signals (pathway, blocks, exit, home etc) and understand their meaning (beneficial and deleterious). Thirdly, it must navigate in this environment using some form of memory. Fourthly, it must confer goal-driven behaviour on the agent. The agent performance is determined by its capability to solve the maze as many times as it can during a single life cycle.

The maze environment we produced is much more complex than the traditional mazes, as the agent in this environment can only sense the signal from the maze square it is occupying, not from neighbouring squares. So in order to solve the maze the agent must develop a memory of each step it makes and the direction of movement, and use this memory to find a route toward the exit. As the structure and weights of branches changes at runtime while solving the maze, the learned information is stored both in weights and the structure of the neuron. The *capability to learn* and transformation of learned information into memory in the form of update in weights and structure is stored in genotype.

Results and Analysis

Figure 3 shows a number of mazes in first column. Fitness improvement during evolution is shown in the second column. The third column in figure 3 shows the energy variation of the best maze solving agent. The small continuous drop in energy is due to an agent losing its energy after every step. Large decreases occur through encounters with an obstacle or going away from the exit by following the pathway in opposite direction. Small increases shows the result of following the pathway and moving toward the exit and large increases happen when the agent finds the exit. The fourth and the last column shows the variation in neuron branching structure over the agent lifetime, while it is solving the maze.

The agent is able to solve the maze four to five times during a single life cycle in all the cases as shown in the second column of figure 3. During this process the structure of the neuron also changes in terms of the number of dendrite and axon branches. The fourth column of the figure 3 shows that although agents start with a minimal structure they soon achieve a structure that is most advantageous.

In traditional methods that train an agent to solve the maze and find a path, the network characteristics are fixed once it is trained to solve the maze. So if they are allowed to start the maze again they would always follow the same path. As the CGP Neuron continues to change its architecture and parameter values it also continues to explore different paths on future runs. This makes it possible for it to obtain (or forget!) a global optimum route. The networks is not trained to stabilize on a fixed structure, that it does so, seems to be because it has found a suitable structure for the desired task. The best architecture does not necessarily have to have the most neurite branches. This is evident from the varied characteristics in the last column of figure 3.

It is interesting to note that as the task become bigger and bigger the structure of the neuron grows in response to it. This is evident from the last column of the figure 3. For an 8x8 maze (first and second maze) the agent structure grows and stabilizes on a fairly small structure whereas for a 10x10 maze (3rd, 4th and 5th mazes) the number of dendrite and axon branches grows into a fairly large structure (the maximum allowed value is 100 in this case). Further investigation reveals that as the route toward the exit becomes more and more complex, the network structure become richer in terms of branches. This is evident from the second 10x10 maze (4th row) where the number of blocking paths are 10 (with each obstacle providing four walls in all the four directions, 40 walls), and number of pathways are 20. Ten on the sides (first and last column) with possibility to move in both upward and downward directions and ten that are only open toward the exit in downward direction). In this case the agent was able to solve the maze three times, as is evident from the rises in the energy level diagram. However, it dies on the fourth run when it tried to escape through the starting point. In next case, when we have reduced the number of obstacles to six (24 walls) while keeping the number of pathways the same as shown in the in fourth row of figure 3. This time the agent was able to solve the maze four times and its axon branch structure is improved during its run but the dendrite structure is stabilized on a low value. The final maze is a variant of 10x10 maze in third row with similar characteristics. In 8x8 mazes when the environment is simple, the agent was able to solve the maze a number of times even though it stabilized on a fairly small branch structure. This strongly suggests that the complexity of the CGP Neuron structure increases with increase in the complexity of the task environment.

Conclusion

We have described a neuron-inspired developmental approach to construct a new kind of computational neural architectures which has the potential to learn through experience. We found that the neural structure controlling the agents grows and changes in response to their behaviour, interactions with the environment, and allow them to learn and exhibit intelligent behaviour. We found that the network complexifies itself in response to the environmental complexity. The eventual aim is to see if it is possible to evolve a network that can learn by experience.

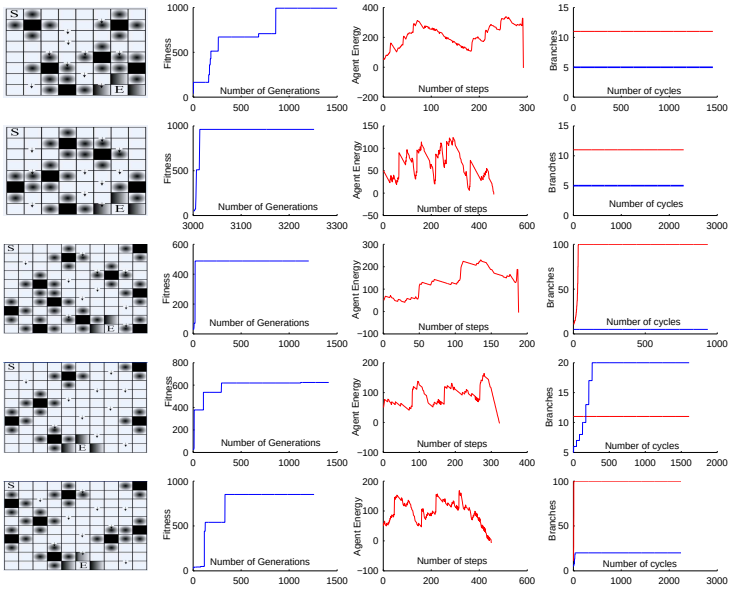


Figure 3: Mazes, Fitness, Best Run and Variation in Branch Structure

References

Al-Musa, S., Abu Fara, D., Badwan, A., Ryder, E., Bullard, L., Hone, J., Olmstead, J., and Ward, M. (1999). Graphical simulation of early development of the cerebral cortex. *Computer Methods and Programs in Biomedicine*, 59(2).

Ascoli, G. A., Krichmar, J. L., Scorcioni, R., Nasuto, S. J., and Senft, S. L. (2001). Computer generation and quantitative morphometric analysis of virtual neurons. *Anat. Embryol.*, 204.

Blynel, J. and Floreano, D. (2003). Exploring the t-maze: Evolving learning-like robot behaviors using ctrnns. In *EvoWorkshops*, pages 593–604. Springer Berlin / Heidelberg.

Cangelosi, A., Nolfi, S., and Parisi, D. (1994). Cell division and migration in a 'genotype' for neural networks. *Network-Computation in Neural Systems*, 5:497–515.

da Fontoura Costa, L. and Coelho, R. C. (2005). Growth-driven percolations: the dynamics of connectivity in neuronal systems. *Eur. Phys. J. B Condens Matter Complex Syst.*, 47.

Dalaert, F. and Beer, R. (1994). Towards an evolvable model of development for autonomous agent synthesis. In *Brooks, R. and Maes, P. eds. Proceedings of the Fourth Conference on Artificial Life*. MIT Press.

de Gennes, P.-G. (2007). Collective neuronal growth and self organization of axons. In *Proc. Natl. Acad. Sci. U.S.A.*, page 49044906.

Downing, K. L. (2007). Supplementing evolutionary developmental systems with abstract models of neurogenesis. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 990–996.

Federici, D. (2005). Evolving developing spiking neural net-

works. In *Proceedings of CEC 2005 IEEE Congress on Evolutionary Computation*, pages 543–550.

Gruau, F. (1994). Automatic definition of modular neural networks. *Adaptive Behaviour*, 3:151–183.

Ilin, R., Kozma, R., and Werbos, P. (2007). Efficient learning in cellular simultaneous recurrent neural network the case of maze navigation problem. In *IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, pages 324–329. IEEE Press.

Jakobi, N. (1995). *Harnessing Morphogenesis, Cognitive Science Research Paper 423, COGS*. University of Sussex.

Kandel, E. R., Schwartz, J. H., and Jessell, T. (2000). *Principles of Neural Science, 4rth Edition*. McGraw-Hill.

Khan, G., Miller, J., and Halliday, D. (2007). Coevolution of intelligent agents using cartesian genetic programming. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 269 – 276.

Khan, G. M. and Miller, J. F. (2009). Evolution of cartesian genetic program capable of learning. In *Proceedings of Genetic and Evolutionary Computation Conference (GECCO'09)*, pages 707–714. ACM.

Kiddie, G., McLean, D., Ooyen, A. V., and Graham, B. (2005). Development,dynamics and pathology of neuronal networks: from molecules to functional circuits, progress in brain research 147. *In Biologically Plausible Models of Neurite Outgrowth*.

Kliemann, W. (1987). A stochastic dynamical model for the characterization of the geometrical structure of dendritic processes. *Bull. Math. Biol.*, 49.

Koch, C. and Segev, I. (2000). The role of single neurons in information processing. *Nature Neuroscience Supplement*, 3:1171–1177.

Koza, J. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural selection*. MIT Press.

Krottje, J. K. and van Ooyen, A. (2007). A mathematical framework for modeling axon guidance. *Bull. Math. Biol.*, 69.

Lindenmayer, A. (1968). Mathematical models for cellular interactions in development. parts 1 and 2. *J. Theor. Biol.*, 18.

Mannor, S., Menache, I., Hoze, A., and Klein, U. (2004). Dynamic abstraction in reinforcement learning via clustering. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 71.

Marcus, G. F. (2001). Plasticity and nativism: towards a resolution of an apparent paradox. pages 368–382.

Miller, J. F. and Thomson, P. (2000). Cartesian genetic programming. In *Proc. of the 3rd European Conf. on Genetic Programming*, volume 1802, pages 121–132.

Miller, J. F., Vassilev, V. K., and Job, D. (2000). Principles in the evolutionary design of digital circuits-part i. *genetic programming*. volume 1:1/2, pages 7–35.

Parisi, D. and Nolfi, S. (2001). *Development in Neural Networks. In Patel, M., Honovar, V and Balakrishnan, K.eds. Advances in the Evolutionary Synthesis of Intelligent Agents*. MIT Press.

Quartz, S. and Sejnowski, T. (1997). The neural basis of cognitive development: A constructivist manifesto. *Behav. Brain. Sci*, 20:537–556.

Rust, A., Adams, R., and H., B. (2000). Evolutionary neural topiary: Growing and sculpting artificial neurons to order. In *Proc. of the 7th Int. Conf. on the Simulation and synthesis of Living Systems (ALife VII)*, pages 146–150. MIT Press.

Samsonovich, A. V. and Ascoli, G. A. (2005). Statistical determinants of dendritic morphology in hippocampal pyramidal neurons: a hidden markov model. *Hippocampus*, 15.

Samuels, D. C., Hentschel, H. G., and Fine, A. (1996). The origin of neuronal polarization: a model of axon formation. philos. trans. r. soc. lond., b. *Biol. Sci.*, 351.

Stanley, K. O. and Miikkulainen, R. (2003). A taxonomy for artificial embryogeny. *Artificial Life*, 9(2):93–130.

Stepanyants, A., Hirsch, J. A., Martinez, L. M., Kisvrday, Z. F., Ferecsk, A. S., and Chklovskii, D. B. (2008). Local potential connectivity in cat primary visual cortex. *Cereb. Cortex.*, 18.

Stiefel, K. M. and Sejnowski, T. J. (2007). In biologically plausible models of neurite outgrowth mapping function onto neuronal morphology. *J. Neurophysiol*, 98.

Tani, J. (1996). Model-based learning for mobile robot navigation from the dynamical systems perspective. *IEEE Trans. on Systems, Man, and Cybernetics*, 26:421–436.

Werbos, P. and Pang, X. (1996). Neural network design for j function approximation in dynamic programming. *Math'l Modeling and Scientific Comp.*, 2.

Self-Organization of a Virtual Multicellular Organism by Adding a Shape Model in the Cellular Potts Model

Sébastien Tripodi, Pascal Ballet and Vincent Rodin

European University of Brittany - UEB UBO, EA 3883-LISyC 20 av Le Gorgeu CS 93837 29238 Brest Cedex 3, France
sebastien.tripodi@univ-brest.fr

Abstract

The Cellular Potts Model (CPM) is a cellular automaton (CA) allowing to model the morphogenesis of living cells. It characterizes a cell by its volume, surface and type. The CPM has already been used to simulate several models of cell self-organization. However, the cell shape is under-constraint $i.e.$ it does not implies a unique shape. We propose a definition and an implementation of the cell shape in the CPM, that can target a unique shape. The results of our simulations show that this target shape can structure and maintain the cellular tissue since the beginning of its growth and during its life.

I Introduction

The Cellular Potts Model (CPM) is a cellular automaton (CA) made by Glazier and Graner (Graner and Glazier, 1992). It has been often used to model and simulate phenomena occurring in the morphogenesis and embriogenisis. (Cickovski et al., 2005; Marée, 2000). The CPM is an extension of the Potts Model developed by Potts in 1952 which generalizes the Ising Model as described in (Wu, 1982). The dynamics of these models are based on a minimization of energy. In the discrete case, the CPM consists of a grid where a set of cells fills each site of the grid. The entities of the system are called cells and are characterized by a volume, surface and type. They are in interaction via contact energies and restricted access to grid sites.

The first model used to illustrate the CPM is the cell sorting. It shows how simple local interactions allow self-organization of the biological cells. At the cellular automata level the self-organization has already been done in more abstract phenomena like the Game of Life developed by John Conway (Gardner, 1970) or the Langton's Ant (Langton, 1984).

Since this first model several extensions of CPM have been done (Anderson et al., 2007). However, the cell shape is not defined in a more specific way. Indeed, in the basic CPM, the shape is characterized only by a target volume and surface. So several shapes can verify a same target volume and surface. In this paper we propose to add an energy that allows the cells to emerge towards a unique and defined shape. This energy comes from a set of springs which provides the cell a elastic shape .

We use the cell shape to structure the shape tissue via the cell self-organization. To test and show the characteristics of the cell shape we simulate a model which comes from an extended CPM. This model allows the cell to self-align and to build a coherent cellular tissue $i.e$ with a recognizable shape and a dynamical tissue renewal.

This paper is organized as follows. A formalization of the CPM is given in section II. In section III we describe the MorphoPotts which represents a cell defined in the CPM to which we add the elastic shape in section IV and other cell behaviors. Using the MorphoPotts, in section V, we simulate a model of tissue formation from which a stability of the cellular tissue and a dynamical tissue renewal emerge. Finally, we conclude in section VI.

II Presentation of the CPM

In this part we recall the formalism of CPM explained in (Graner and Glazier, 1992; Glazier and Graner, 1993). The first part describes the necessary notations to the comprehension of this paper. The second part describes the strong notions of this formalism (see Figure 1), $i.e.$ the state of the system and the transition function thanks to the transition probability, the energy function and the neighborhood function.

Notation. A grid is denoted by Sx and a site of this grid is denoted by (i, j). The value of a site (i, j) is denoted by $sx_{i,j}$. A cell is denoted by C_σ^t with $\sigma \in [1, N]$ where N is the number of cells and t the type of cell. The number 0 is reserved for the medium. A cell C_σ^t has a target volume (resp. surface) $V\sigma_t$ (resp. $S\sigma_t$) and current volume $V\sigma$ (resp. $S\sigma$). The target volumes and target surfaces of the cell are the volumes and surfaces to which the cell tends. The contact energies are recorded in a matrix T such that $T_{\sigma,\sigma'}$ (resp. $T_{t,t'}$) is the contact energy between the cell C_σ^t and the cell $C_{\sigma'}^{t'}$ (resp. between the cells of type t and t').

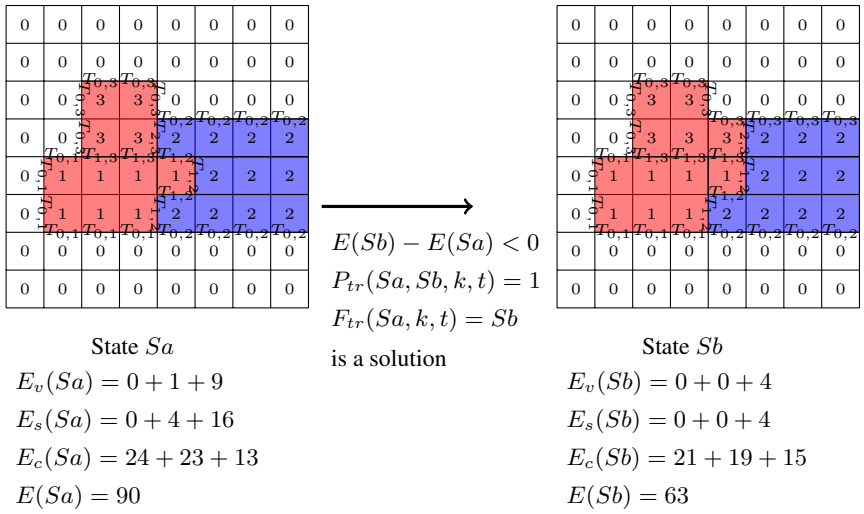


Figure 1: Example of a transition in the CPM. A state Sx is a grid 8x8 where to each site (i, j) we associate a value $sx_{i,j}$. So we have four cells ($\sigma \in [0, 3]$): one cell for the medium (C_0^m), two cells of red type (C_1^r, C_3^r) and one cell of blue type (C_2^b). The cells of the red type have the following characteristics: $V_{target} = 7$, $S_{target} = 12$, and the cell of blue type the following characteristics: $V_{target} = 10$, $S_{target} = 11$. In the state Sa the cell C_1^r (resp. C_2^b, C_3^r) has a volume $V1 = 7$ (resp. $V2 = 11$, $V3 = 4$) and a surface $S1 = 12$ (resp. $S2 = 13$, $S3 = 8$). In the state Sb the cell C_1^r (resp. C_2^b, C_3^r) has a volume $V1 = 7$ (resp. $V2 = 10$, $V3 = 5$) and a surface $S1 = 12$ (resp. $S2 = 11$, $S3 = 10$). The cell for the medium does not have volume and surface constraints. The matrix (symmetric) of contact energy (given) is defined as: $T_{0,1} = T_{0,3} = 2$, $T_{0,2} = 1$, $T_{1,2} = T_{2,3} = 3$, $T_{1,3} = 0$. Since the cell C_1^r and the cell C_3^r are of the same type $T_{1,?} = T_{?,3}$.

State of the System. The CPM is composed of a grid Sx[1] of D dimensions (here $D = 2$). Each site (i, j) is filled by a particle of cell C_σ^t, *i.e.* the value $sx_{i,j}$ of site (i, j) in the state Sx is equal to σ. So a cell C_σ^t is equal to $\{(i, j) \in Sx | sx_{i,j} = \sigma\}$ the set of sites whose value is σ.
Finally a state of system is a grid Sx where each $sx_{i,j}$ is equal to an integer $\sigma \in [0, N]$.

Transition Function. Let $F_{tr}(Sa, k, t) = Sb$ the transition function of the CPM between the State Sa and Sb according to k and t. Let Sc be the state Sa where the value of a site has been replaced by the value of a neighbor site. If the probability of transition P_{tr} between the states Sa and Sc is accepted, then $Sb = Sc$, otherwise $Sb = Sa$.

$$\begin{aligned} F_{tr}(Sa, k, t, p) = Sb \Leftrightarrow \exists (i', j') \in neighbor(i, j)(\\ (sc_{i,j} = sa_{i',j'}) \wedge \\ (Sc - sc_{i,j} = Sa - sa_{i,j}) \wedge (p = rand(]0, 1])) \wedge \\ (p \leq P_{tr}(Sa, Sc, k, t) \Rightarrow Sb = Sc) \wedge \\ (p > P_{tr}(Sa, Sc, k, t) \Rightarrow Sb = Sa)) \end{aligned}$$

where $rand(E)$ returns a random element of the set of E, $neighbor(i, j)$ is the set of neighbor sites of (i, j) and P_{tr} the probability of transition.
We can observe that only one site of the grid can change and since several sites can be candidates to change, the dynamics is asynchronous and non-deterministic.

Probability of Transition. The Probability of transition used is the Monte Carlo probability following a temperature t. Let $P_{tr}(Sa, Sb, k, t) = p$, the probability of transition between the states Sa and Sb according to k and t.

$$\begin{aligned} P_{tr}(Sa, Sb, k, t) = p \Leftrightarrow \\ t > 0 \wedge (E(Sb) - E(Sa)) \leq 0 \Rightarrow p = 1 \\ t > 0 \wedge (E(Sb) - E(Sa)) > 0 \Rightarrow \\ p = \exp\left((E(Sb) - E(Sa))/kt\right) \\ t = 0 \wedge (E(Sb) - E(Sa)) < 0 \Rightarrow p = 1 \\ t = 0 \wedge (E(Sb) - E(Sa)) = 0 \Rightarrow p = 0.5 \\ t = 0 \wedge (E(Sb) - E(Sa)) > 0 \Rightarrow p = 0 \end{aligned}$$

where $E(S)$ is the function of energy.
This probability promotes the transitions which lead to a lower energy state.

Energy Function. Let $E(S) = e$ the energy function of the state S. This function characterizes the state of the system. In the CPM, a basic function depends on the volume and surface of each cell and on the contact energies between two cells. $E(S)$ can be defined as:
$E(S) = \lambda_c * E_c(S) + \lambda_v * E_v(S) + \lambda_s * E_s(S)$ with
$E_c(S) =$

$$\sum_{(i,j) \in S} \sum_{(i',j') \in neighbors(i,j)} {}^2T_{s_{i,j}, s_{i',j'}} * (1 - \delta_{s_{i,j}, s_{i',j'}})$$

where λ_c, λ_v, λ_s are constants, $T_{x,x'}$ is a matrix of contact

[1] Here the environment is discrete but the continuous case is also defined (Glazier and Graner, 1993).

[2] In our simulations the neighbors are the nearest on a 3D square lattice.

energy between the type t and t' respectively of C_x^t and $C_{x'}^{t'}$. If $x = x'$ then $\delta_{x,x'} = 1$ otherwise 0.

$$E_v(S) = \sum_{\sigma \in [1,N]} (V\sigma_t - V\sigma)^2$$

$$E_s(S) = \sum_{\sigma \in [1,N]} (S\sigma_t - S\sigma)^2.$$

III MorphoPotts

To model biological phenomena in a more realistic way, we have proposed in (Tripodi et al., 2010) an multi-agent approach of CPM and a cell called MorphoPotts. The MorphoPotts is an extension of the cell defined in the CPM by adding the following behaviors: secretion and consumption of molecules, transformation of molecules into energy, migration on a gradient of molecules, cell division and cell differentiation. The MorphoPotts is very close to MorphoBlock (Ballet et al., 2009) compared to secretion of molecules and the migration under a gradient of molecules. But the core of MorphoBlock is a pixel whereas the core of MorphoPotts is a cell defined in the CPM. At CPM level, the closest work to MorphoPotts is probably Compucell3D (Cickovski et al., 2007), a software which implements the CPM and other behaviors. In this section, we describe fristly the MorphoPotts, and secondly a step of simulation of CPM-MorphoPotts couple.

Description of MorphoPotts

A MorphoPotts C_σ^t is based on the properties of the cell defined in section II, but it also has an internal energy E. This energy results from the consumption of molecules found in the environment. The MorphoPotts can perceive and modify the environment beyond their neighborhood boundaries defined in section II.

The behaviors of the MorphoPotts are described in Table 1. We assume that the secretion creates a gradient because the diffusion of molecules is faster than cell migration and the secretion is continuous. For the same reasons we assume that the consumption of molecules creates a "well" ($i.e.$ inverse effect of secretion). In this paper, the energy of the MorphoPotts is used as a criterion for MorphoPotts division and MorphoPotts death.

Step of Simulation

The step of the simulation which combines the CPM and the MorphoPotts is following:

1. Let i equals to 0 and n equal to the membrane size of all MorphoPotts.
2. While i is lower than n
   (a) One transition function of the CPM is applied.
   (b) If the criterion of division of the chosen MorphoPotts during the transition is verified, they divide.
   (c) i is incremented by 1
3. All MorphoPotts execute their method of maintenance.
4. All MorphoPotts execute their method of secretion.
5. All MorphoPotts (the scheduling is random to delete the artefacts) execute their method of consumption.
6. If the internal energy of the cells is lower than 0, they die.

The step of simulation can, for each cell, modify each membrane site before calling to methods of maintenance, secretion, consumption and death. This allows to synchronize every MorphoPotts and so to delete some artefacts due to asynchronicity of the CPM. Indeed, in reality, the cells move at the same time and not one after another.

Proposition of a cell shape energy

In the previous section we have built a model of cell called MorphoPotts. However, the cell shape is not strongly defined. A volume and a surface do not entirely characterize a geometric shape. The goal of this section is to constraint the cell to keep a certain rigidity of the shape. The cell shape is an important feature. It can lead to different functions and properties, $i.g.$ the spherical shape of red blood cells adapts perfectly to their role in transport from the bloodstream, the spindle-shaped muscle cells allows them to contact and realizes a close fit between them, thus facilitating the simultaneous contraction of muscle tissue.
Several propositions have already been done to target the cell shape, like cell elongation (Merks et al., 2006), but to our knowledge, none can target all forms. The idea is to give an elastic shape to the cell. For this we add a set of springs to the cell like described in Figure 2. In this section, we describe fristly the formalism , and secondly the implementation.

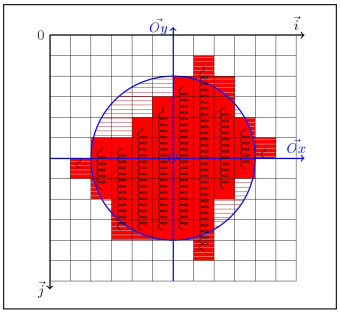


Figure 2: Example of elastic shape. We have one red cell C_2^r with an elastic shape where the distribution $sL0_p^s$ of springs R_p^s is given by the function of a circle of center O and radius 4, represented by the blue circle. The energy of this elastic shape is the sum of distance power 2 between the sites with the lines and the circle blue. The sites with white lines are sites of extension and the sites with red lines are sites of compression.

Behavior	Description
Secret a gradient of arg molecules Y	If the site (i,j) contains n molecules Y then, after the secretion, the site will contain a number of molecules Y equal to the integer closest to $n + \frac{arg}{\sqrt{(i-gx)^2+(i-gy)^2}}$ where (gx, gy) is the center of gravity of MorphoPotts.
Consume a gradient of arg molecules Y	If the site (gx, gy) contains n molecules Y and the site (i,j) contains n', the number of molecules Y in (i,j) is modified such that the new value is 0 if $n' < \frac{min(n,arg)}{\sqrt{(i-gx)^2+(j-gy)^2}}$ otherwise the new value is the integer closest to $n' - \frac{min(n,arg)}{\sqrt{(i-gx)^2+(j-gy)^2}}$
Migrate to the molecules	The energy function of the CPM is modified by adding a new energy $E_{migr} = -arg * \sum_{(i,j)\in M_\sigma} nbMolecules((i,j),Y)$ where $nbMolecules((i,j),Y)$ is the number of molecules Y on the site (i,j).
Transform the consumed molecules in energy	In this paper for each consumed molecule the energy is incremented by 1.
Differentiate	The probability that the MorphoPotts changes its types is equal to $\frac{arg}{\sum_{Y'\neq Y} arg'}$ where arg is the probabilty associated to the type Y cell.
Divide	A MorphoPotts can divide in two axes (vertical or horizontal). A new MorphoPotts is created according to the probability of differentiation. The energy of the new MorphoPotts is equal to E' and the energy of the old MorphoPotts is equal to E (internal energy of the MorphoPotts) minus E' minus $cost$ the cost of the MorphoPotts division.
Maintain	The energy of the MorphoPotts is decremented by arg, representing the costs of the maintenance.
Die	The MorphoPotts dies if its internal energy is equal to 0. The death means that the MorphoPotts looses all its abilities and it does not generate energy in the CPM.

Table 1: Abilities of the MorphoPotts

Formalisation of the elastic cell

To constraint the cell to keep a 3D shape in the CPM formalism, we define in this section a function of energy $E\sigma_{sp}$. $E\sigma_{sp}$ is null if the shape is reached by the Cell C_σ. $E\sigma_{sp}$ is the sum of energies provided by the springs given to the cell. The energy of one spring R at the position p, p' (the position of these extremities) for a cell C_σ is defined like:

$$\sum_{s_a=\sigma} \begin{cases} 1/2 * k * dist(a,R)^2 & \text{if this spring is the closest to site } a \text{ according to criterion } C(R,a) \text{ and } dist(a,R) = min(|\overrightarrow{ap}|, |\overrightarrow{ap'}|) \\ 0 & \text{otherwise.} \end{cases}$$

where k is the constant force of the spring.
The disposition of the springs depends on the model and several shapes can be given to one cell. In this paper the springs are parallel. For this:

- we add a Cartesian coordinate system (O, Ox, Oy, Oz) where O is a point in the grid. The axis Oy gives the direction of the springs.

- we add a set of springs perpendicular to the plan defined by the axes Ox and Oz, i.e the springs $R\sigma_p^s$ where $s \in \{+1, -1\}$ whose two extremities are in position (p_x, p_y, p_z) and $(p_x, s * L0_p^s + p_y, p_z)$, $L0_p^s$ being the rest length of spring. The distribution of $R\sigma_p^s$ and the length $L0_p^s$ depend on the desired shape (see Figure 2).

To compute $E\sigma_{sp}$ we define in this paper the following criterion C:
"$R\sigma_p^s$ is the closest spring to the site (i,j,l) if a spring $R\sigma_{p_x,y',p_z}^s$ such that $dist((i,j,l), R\sigma_p^s) > dist((i,j,l), R\sigma_{p_x,y',p_z}^s)$ does not exist"
So $E\sigma_{sp}$ in this paper is defined like:

$$E\sigma_{sp} = 1/2 \sum_{R\sigma_p^s} \sum_{s_a=\sigma \wedge C(R\sigma_p^s,a)} k_p^s * dist(a, R\sigma_p^s)^2$$

Implementation of the elastic cell

The implementation of the elastic cell can be done by the computation of the intersection between a cell and a line (the axis of the springs). A naive implementation could be to browse all sites of the cell and to build the set of sites which are crossed by the spring. The problem is that it will take too long simulation time.

In one simulation step of the CPM, only one site value $s_{i,j,l}$ changes, modifying the cells $C_\sigma, C_{\sigma'}$. So we have:
$\Delta E\sigma_{sp} = 1/2 * ($
$z(j, L0_p^s) * k_p^s * dist((i,j,l), R\sigma_p^s)^2 -$
$z(j, L0_{p'}^{s'}) * k_{p'}^{s'} * dist((i,j,l), R\sigma_{p'}^{s'})^2)$
C_σ is the cell which increases, C'_σ is the cell which decreases and (i,j,l) the site added or deleted. $C(R\sigma_p^s, (i,j,l))$ and $C(R\sigma_{p'}^{s'}, (i,j,l))$ are verified.
$z(j, L0_p^s) = 1$ if $p_j \leq j \leq s * L0_p^s + p_j$ (compression) otherwise -1 (extension).

Also to compute $\Delta_{E\sigma_{sp}}$, we store in a table for each site p of the shape, the static following informations: $z(p_j, R_p^s)$,

$L0_p^s$ and k_p^s, in the coordinate system of the shape.
So $\Delta_{E\sigma_{sp}}$ returns to compute one translation and one rotation (to find the position of the changed site in the coordinate system of the shape) and an access to the table. The cost is constant and does not significantly modify the simulation time.

Rotation and Translation of the elastic shape

We saw in the previous part that the definition of the cell shape uses a target shape. However, the shape is located at a specific coordinate. This causes the cell does not move in the environment. In this part we show how we consider the rotation and the translation of the shape according to the adding or deleting sites

Rotation of the elastic shape This part describes how the shape turns in the environment. For example, if a cell is attracted to a direction due to a gradient of molecules, the sites which are closed to the source of the gradient have a higher probability to be added to the cell. This behavior can turn the cell in the direction of the gradient.

We construct a function named $rotation(m, p, C_\sigma)$ which returns a vector of angle. The size of this vector is equal to the number of dimension. The angle corresponds to the rotation of the cell (C_σ) shape after adding the site s_p if $m = +$ or the deleting of the site s_p if $m = -$. The shape rotation is made by the rotation of its coordinate system compared with the coordinate system of the environment.

Here, $rotation(m, p, C_\sigma) = \alpha/V_\sigma * (arcant2(p_y, p_x), arcant2(p_z, p_y), arcant2(p_x, p_z))$.
This function means that the rotation angle is the angle between the axis Oy, the origin and the point p in the coordinate system of the shape. The angle value is normalized by the volume of the cell and the value is increased or decreased by α.

Translation of the elastic shape We construct a function $translation(C_\sigma)$ which returns a vector. This vector is used to translate the shape after adding or deleting a site of the cell C_σ.

Here, $translation(C_\sigma) = \beta * \Delta\vec{G}_\sigma$ where ΔG_σ is the variation of the gravity center of the cell C_σ during a simulation step of the CPM. β can favour or not the translation of the shape.

Rotation and Translation in the simulation step The rotation and translation of the shape is possible because environmental or internal conditions can add or delete sites of the cell in specific directions. However if the translation and the rotation are made at each step of the simulation, an undesirable $perpetuum\ mobile$ is possible.

Indeed, if the translation is realized towards a direction, the sites in this direction will be added to the cell that implies a new translation in this same direction and etc ... The translation and rotation are not done when the transition is accepted thanks to the energy provided by the springs, $i.e.$ when the variation of the energy is negative. The shape has to be reached before doing a new translation or rotation.

IV Validation of the elastic shape

To validate and show the interest of the elastic shape we test 2 models of MorphoPotts. The first model proposes to test the energy of the shape without cell translation and rotation, the second to test the cell translation and rotation by simulating the formation of a tissue via cell self-organization.

Example of the elastic shape

In this part, we test the elastic shape. For this, thanks to our tool we can draw a 3D shape and automatically store the informations described in section IV (see figure 3(a)).

The model used for the simulation consist of 4 MorphoPotts: one MorphoPotts to model the exterior medium to the cells and three MorphoPotts to test the same shape. The coordinate system of the shape of the middle MorphoPotts is rotated by $\pi/2$ on the axis $0x$ (see Figure 3(a)). A vertical section of the shape is given in Figure 3(a). The visible springs on the horizontal axis have the parameters $k = 10$. The springs of length null, complete the horizontal axis with $k = 10^6$ to avoid a growth of the cell along this axis. In this model, the parameter α (resp. β) of the rotation (resp. translation) is $null$. We just test the target shape. No contact, volume and surface energy are taken into account in this Model.

The results of the simulation are given in the Figure 3. The Figure 3(a) shows the initial state. The Figure 3(b) is a picture of the shape being built. The Figure 3(c) shows the MorphoPotts having reached the target shape and also validate our implementation of the elastic shape.

V Cell Self-organization

In this section we present a simulation of a model which test both the translation and rotation of the shape, and the cell self-organization to build a coherent tissue (a recognizable shape and a dynamical tissue renewal). After a description of model, we discuss the parameters before showing the results of the simulations.

Presentation of the model To show the interest and the properties of the rotation and the translation of the shape, we made a model allowing to simulate the generation and the life of a cellular tissue. This model consists of three type MorphoPotts:

- the first type of MorphoPotts models the exterior medium.
- the second type of MorphoPotts produces molecules in the medium.
- the third type of MorphoPotts consumes the produced molecules by the second type and divides. This type has a elastic shape and is used to build the tissue.

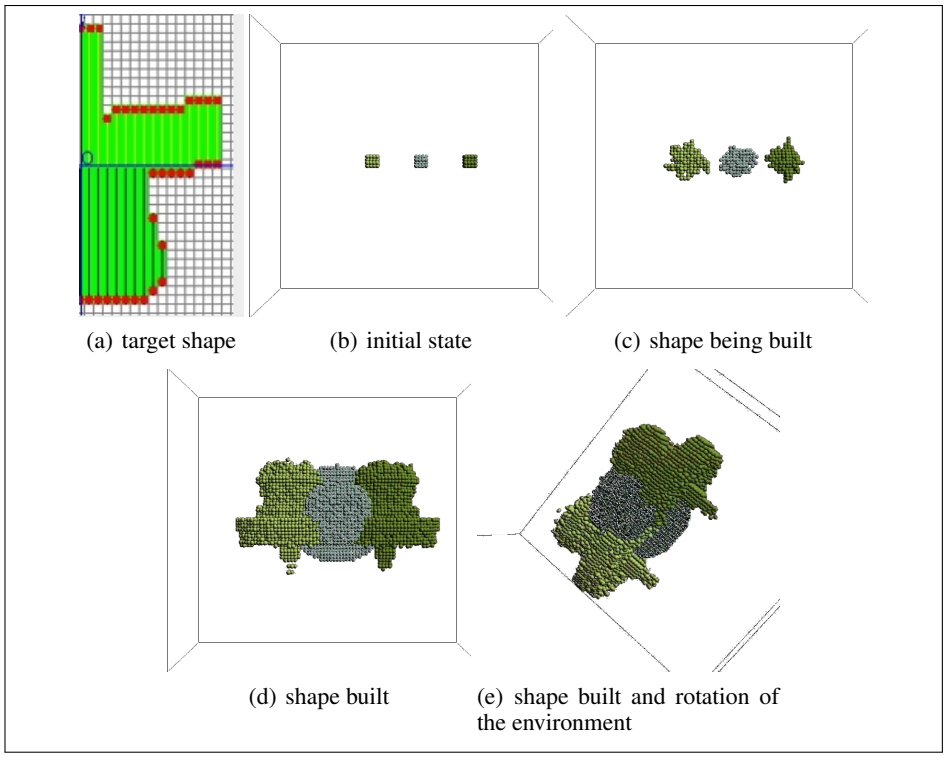


Figure 3: Example of elastic shape

The interactions between the MorphoPotts are:

- a direct interaction. A negative energy of contact (means that the MorphoPotts which stay together do not use energy) is set between the MorphoPotts of type 3 . A positive energy of contact is set between the MorphoPotts of type 3 and 2
- an indirect interaction. The MorphoPotts of type 2 provides molecules to MorphoPotts of type 3. If the MorphoPotts of type 3 does not found the molecules, it dies.

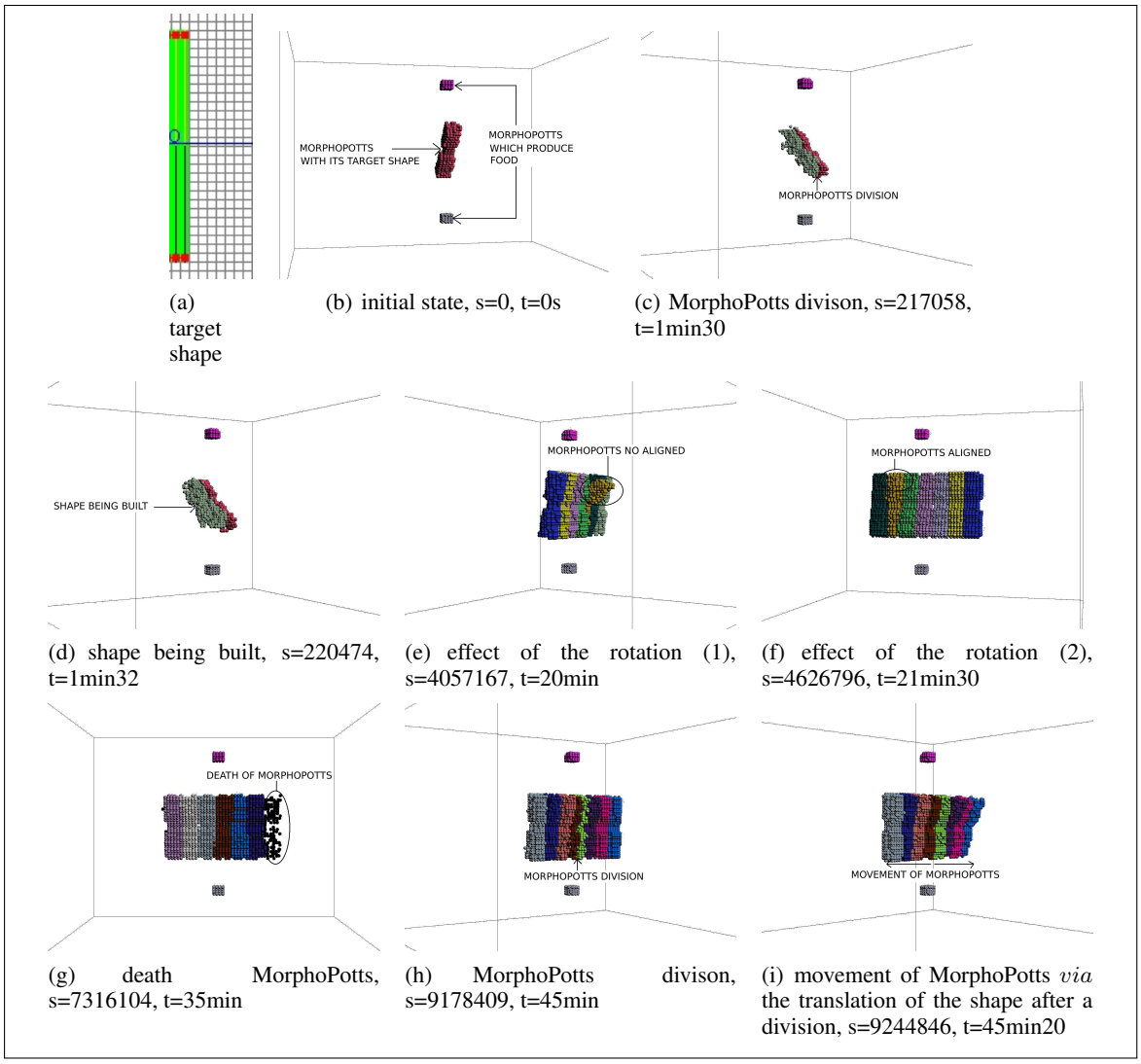
We show with this model that the cell shape and the contact energy can structure the cellular tissue. The competition of the MorphoPotts to consume the molecules allows a finite growth of cellular tissue like described in (Laforge et al., 2005) and a dynamical tissue renewal.

Parameters analysis We have defined 4 types of MorphoPotts. The parameters of these MorphoPotts are given in Table 1.

The energies of contact verify that $5 * T_{1,3} + T_{3,3} < 0$. When two MorphoPotts of type 3 are in contact thanks to the adding of a site, $\Delta E_c < 0$. The adding of this site is favored by energies of contact.
The concentration of the molecule 1 (produced by MorphoPotts of type 2) decreases with the distace from the source. If the MorphoPotts of type 3 are at a too long distance from a MorphoPotts of type 2, they have not enough molecules to survive (higher than 52 pixels).
The MorphoPotts of type 3 can divide if its energy is higher than 20000 (experimental value).

The shape described in Figure 4(a) is given to the MorphoPotts of type 3. The volume and the surface are each equal to 328,64. So the target volume and surface can fill the shape. 21 extra sites have to be added to the MorphoPotts of type 3 to verify the target volume and surface. The visible springs in Figure 4(a) on the horizontal axis have the parameters $k = 10^7$ to force the MorphoPotts to reach its shape. The springs of length null complete the horizontal axis with $k = 10^5$ to avoid a growth of the cell along this axis. In this model, the parameter α (resp. β) of the rotation (resp. the translation) is 10 (resp. 75). The rotation and the translation are possible only on the axis Oz because we model the construction of a cellular tissue along one direction. The α and β have been calibrated by dichotomy.

The parameters kt of the CPM is equal to 1, so the probability of transition is equal to $e^{-\Delta E}$. The transitions with $\Delta E > 0$ have a weak chance to be accepted. The constant λ_c (resp. λ_v, λ_s) is equal to 1 (resp. 10000, 10000). These constant values allow the MorphoPotts not to oversize their target volume.



(a) target shape

(b) initial state, s=0, t=0s

(c) MorphoPotts divison, s=217058, t=1min30

(d) shape being built, s=220474, t=1min32

(e) effect of the rotation (1), s=4057167, t=20min

(f) effect of the rotation (2), s=4626796, t=21min30

(g) death MorphoPotts, s=7316104, t=35min

(h) MorphoPotts divison, s=9178409, t=45min

(i) movement of MorphoPotts *via* the translation of the shape after a division, s=9244846, t=45min20

Figure 4: Cell Self-organization. This simulation shows how the cell shape can structure and maintain the cellular tissue since the beginning of its growth and during its life. t is the time of simulation and s is the number of CPM steps. The pc used used for this simulation is a Pentium Quad 2.8Ghz and the language is JAVA.

type	target volume	target surface	Energy of Contact	Secretion	Consumption	Division	Maintenance
1 (exterior medium)	_	_	$T_{1,3}$=100	_	_	_	_
2 (producer of molecules)	_	_	_	secr(310000,1)	_	_	_
3 (producer of molecules)	350	350	$T_{3,1}$=100 $T_{3,3}$=-10000	_	cons(1000,1)	div({E>20000, E/2,0}, 3)	main(600)

Table 2: MorphoPotts Parameters. The symbol _ means that the parameter is not taken into account. $cons(1000, 1)$ ($resp.$ secr(310000,1)) means that the MorphoPotts consumes ($resp.$ produced) a gradient, 1000 ($resp.$ 310000) molecules of type 1 in the center. div({E>20000, E/2,0}, 3) means that if the internal energy of the MorphoPotts is higher than 20000, it divides and gives half of its energy to newly born MorphoPotts and the cost of the division is null. main(600) means that the maintenance cost 600.

Discussion of the results The Figure 4 shows the results of the simulation[3]. The initial state (see Figure 4(b)) consists of one MorphoPotts of type 3 being attained its shape. The MorphoPotts of type 2, which produce the molecules, are also present. The MorphoPotts of type 1, which models the exterior medium, is invisible and occupies the empty environment. The environment is a 3D matrix 100x100x100.

Between Figure 4(b) and 4(c) the MorphoPotts of type 3 consumes enough molecules to have an energy allowing its division (on the axis Oy) in Figure 4(c). In Figure 4(d) the shape of the MorphoPotts of type 3 is being built. In the same time the two MorphoPotts of type 3 self-align thanks to the energies of contact. In figure 4(e) and 4(f) we observe the effects of the rotation of the shape. A MorphoPotts is not aligned with the other, the energy of contact favors the sites which are in contact with the other MorphoPotts to be added. So the shape is rotated in this direction. In figure 4(g), the MorphoPotts on right in the figure is too far (a distance higher than 52), and dies. This keeps a finite width of the cellular tissue. Figure 4(e) and 4(f) show the effects of the translation of the shape. After a MorphoPotts division at the center of the tissue, the MorphoPotts are compressed. This implies a translation of the MorphoPotts towards the exterior of the tissue.

The rotation of the shape and the energy of contact allow a self-alignment of the MorphoPotts. The translation of the shape and the competition between the MorphoPotts allow a finite growth of cellular tissue. During the simulation, the MorphoPotts divide at the center of tissue, move towards the exteriors and die at the extremities of the tissue. The shape of the tissue emerges thanks to the shape of the MorphoPotts.

VI Conclusion

We have defined a virtual cell called MorphoPotts. This MorphoPotts is based on the cell defined in the Cellular Potts Model. The MorphoPotts keeps the properties of this cell and the cell behaviors that have been added. In the CPM, the cell shape is represented only by a target volume and surface. We have proposed and implemented a target shape. Therefore, a set of springs is given to the MorphoPotts to build the shape. These springs provide an energy which is used to build a new function of energy in the CPM.

We have tested the target shape in two simulations. The first one shows that it is possible, with this target shape, to give a complex form to the MorphoPotts. The second simulation shows that this target shape allows to structure the cellular tissue. Combined with the energy of contact, the target shape allows the MorphoPotts to self-align. By adding the notion of the internal energy, available in the notion of the MorphoPotts, the second simulation shows that the MorphoPotts self-organize to form a cellular tissue. This tissue has a recognizable shape and a dynamical tissue renewal.

[3]The video of this simulation is available at
http://pagesperso.univ-brest.fr/~tripodi/private/ALIFE12/

Acknowledgements

We thank the Region Bretagne for its financial contributions.

References

Anderson, A., Chaplain, M., and Rejniak, K. (2007). *Single-Cell-Based Models in Biology and Medicine*. Birkhauser.

Ballet, P., Tripodi, S., and Rodin, V. (2009). Morphoblock programming: a way to model and simulate morphogenesis of multicellular organisms. *Journal of Biological Physics and Chemistry ISSN 1512-0856*, 9(1):37–44.

Cickovski, T., Aras, K., Swat, M., Merks, R., Glimm, T., Hentschel, H., Alber, M., Glazier, J., Newman, S., and Izaguirre, J. (2007). From Genes to Organisms Via the Cell: A Problem-Solving Environment for Multicellular Development. *Computing in Science & Engineering*, 9(4):50–60.

Cickovski, T., Huang, C., Chaturvedi, R., Glimm, T., Hentschel, H., Alber, M., Glazier, J., Newman, S., and Izaguirre, J. (2005). A framework for three-dimensional simulation of morphogenesis. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 2(4):273–288.

Gardner, M. (1970). Mathematical games: The fantastic combinations of John Conway's new solitaire game 'Life'. *Scientific American*, 223(4):120–123.

Glazier, J. and Graner, F. (1993). Simulation of the differential adhesion driven rearrangement of biological cells. *Physical Review E*, 47(3):2128–2154.

Graner, F. and Glazier, J. (1992). Simulation of biological cell sorting using a two-dimensional extended Potts model. *Physical Review Letters*, 69(13):2013–2016.

Laforge, B., Guez, D., Martinez, M., and Kupiec, J. (2005). Modeling embryogenesis and cancer: an approach based on an equilibrium between the autostabilization of stochastic gene expression and the interdependence of cells for proliferation. *Progress in biophysics and molecular biology*, 89(1):93–120.

Langton, C. (1984). Self-reproduction in cellular automata. *Physica D: Nonlinear Phenomena*, 10(1-2):135–144.

Marée, S. (2000). *From Pattern Formation to Morphogenesis*. PhD thesis, Utrecht University.

Merks, R., Brodsky, S., Goligorksy, M., Newman, S., and Glazier, J. (2006). Cell elongation is key to in silico replication of in vitro vasculogenesis and subsequent remodeling. *Developmental biology*, 289(1):44–54.

Tripodi, S., Ballet, P., and Rodin, V. (2010). *Computational energetic model of morphogenesis based on multi-agent Cellular Potts Model. Book chapter in Advances in Computational Biology*. Advances in Experimental Medicine and Biology. research book series Springer (accepted).

Wu, F. Y. (1982). The potts model. *Rev. Mod. Phys.*, 54(1):235–268.

Body/Brain Co-Evolution in Soft Robots

John Rieffel[1] and Barry Trimmer[2]

[1]Union College, Schenectady, NY 12308
[2]Tufts University, Medford, MA 02155
rieffelj@union.edu

Extended Abstract

Autonomous Robots have achieved considerable results in a wide variety of domains, from the depths of the ocean to the surface of Mars, and yet many vital locations, particularly collapsed buildings and mines, remain largely inaccessible. In light of recent natural disasters in Haiti and Chile, there is a compelling need for more versatile and robust search and rescue robots. Imagine, for instance, a machine that can squeeze through holes, climb up walls, and flow around obstacles. Though it may sound like the domain of science fiction, modern advances in materials such as silk polymers (Huang et al., 2007) and nanocomposites (Capadona et al., 2008) such a "soft robot" is becoming an increasing possibility.

By soft, we mean an ability to significantly deform and alter shape at a much higher level of detail than discrete "modular" snake-like robots (such as Yim's Polybot Yim et al. (2000) and Rus's Molecubes (Kotay et al., 1998)). In fact the degree of deformability demanded of truly soft robots requires that they contain no rigid parts at all. Unfortunately, the incredible flexibility and deformability demanded of soft robotics carry with them considerable complexity.

There are two significant and coupled challenges to the creation of soft robots: no one knows how to design soft robots, and no one knows how to control them. These challenges arise from the complex dynamics intrinsic softness. Soft and deformable bodies can possess near-infinite degrees of freedom, and elastic pre-stresses mean that any local perturbation causes a redistribution of forces throughout the structure. As a consequence, there are no established principles or purely analytical approaches to the problem of soft mechanical design and control To make matters worse, the biomechanics of soft animals are too complex and too inscrutable to provide much useful insight.

Consider what might seem like a relatively simple completely soft animal: *Manduca sexta*, the tobacco hornworm. The caterpillar achieves remarkable control and flexibility despite the fact that each of its segments contains relatively few motoneurons (one, or maximally two per muscle, with approximately 70 muscles per segment), and no inhibitory motor units (Levine and Truman, 1985). It is conjectured that the complex and coupled dynamics caused by the interaction of hydrostatics, an elastic body wall, and nonlinear muscular behavior, are all harnessed and exploited by the organism (Trimmer, 2007).

This relationship between morphology and control in biology is a richly studied and fascinating topic. Recent research on the tendinous network of the human hand indicate that the system performs "anatomical computation". It is conjectured that "outsourcing" the computation into the mechanics of the structure allows related neural pathways to devote their resources to higher level tasks (Valero-Cuevas et al., 2007). Similar phenomena have been shown in the physiology of wallabies (Biewener et al., 2004) and cockroaches (Ahn and R.J.Full, 2002). Pfeifer and Paul (2006) coined the term "morphological computation" to describe this class of effect. Blickhan (2007) has similarly used the phrase "intelligence by mechanics".

Biological morphological computation has served as inspiration for robotic control in several recent works. Iida and Pfeifer (2006) explored how the body dynamics of a quadraped robot can be exploited for sensing. Watanabe *et al* (2003) demonstrated how inducing long distance mechanical coupling in a snake robot improves its ability to learning a crawling motion. All of these systems, however, involved relatively rigid robotic platforms, and relatively well understood mechanics and dynamics.

An outstanding challenge, therefore, lies in discovering how to inject the properties of this "morphological computation" into soft robots. Classically, engineers design complex robotic systems and only later try to find a controller capable of operating it. However, this approach has difficulty scaling – it is entirely possible to design a robot too complex to

reasonably control. Of course, biology doesn't first "discover" an animal's body, and only later its brain, rather, much like the proverbial chicken and egg, both evolve in tandem. Inspired by those biological processes, modern approaches to the Evolutionary Design of robots by co-evolving morphology and control (Pollack et al., 1999; Sims, 1994).

In this work we show how the chicken-and-egg problem of soft robotic design and control can be addressed via body/brain co-evolution. A co-evolutionary algorithm operating within the PhysX physics simulator simultaneously searches for soft robot muscle attachment points (morphology) along with for firing patterns for those muscles(gaits) capable of making those bodies move. More specifically, two parallel populations are evolved: fitness of the population of gaits relies upon the current best evolved body plan, and fitness of the population of body plans relies upon the best evolved gait. By evolving these two properties contingently and in lock-step, our algorithm is able to produce effective, and sometimes surprising, soft bodied gaits. One particularly interesting outcome is the emergence of antagonistically-placed muscle groups as an effective feature, whereas intuition would suggest that body wall elasticity obviates such a need. This "discovered" design feature was then fed back into physical prototypes of a soft robot, leading to improved real-world performance.

References

Ahn, A. and R.J.Full (2002). A motor and a brake: two leg extensor muscles acting at the same joint manage energy differently in a running insect. *Journal of Experimental Biology*, 205.

Biewener, A., McGowan, C., Card, G., and Baudinette, R. (2004). Dynamics of leg muscle function in tammar wallabies (m. eugenii) during level versus incline hopping. *Journal of Experimental Biology*, 207.

Blickhan, R., Seyfarth, A., Geyer, H., Grimmer, S., Wagner, H., and Gnther, M. (2007). Intelligence by mechanics. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1850):199–220.

Capadona, J., Shanmuganathan, K., Tyler, D. J., and Rowan, S. (2008). Stimuli-responsive polymer nanocomposites inspired by the sea cucumber dermis. *Science*, 319(7).

Huang, J., Foo, C. W. P., and Kaplan, D. (2007). Biosynthesis and applications of silk-like and collagen-like proteins. *Polymer Reviews*.

Kotay, K., Rus, D., Vona, M., and McGray, C. (1998). The self-reconfiguring robotic molecule. In *IEEE International Conference on Robotics and Automation*.

Levine, R. and Truman, J. W. (1985). Dendritic reorganization of abdominal motoneurons during metamorphosis of the moth, manduca sexta. *Journal of Neuroscience*, 5:2424–2431.

Pfeifer, R. and Bongard, J. C. (2006). *How the Body Shapes the Way We Think: A New View of Intelligence (Bradford Books)*. The MIT Press.

Pollack, J., Lipson, H., Funes, P., Ficici, S., and Hornby, G. (1999). Coevolutionary robotics. In *EH '99: Proceedings of the 1st NASA/DOD workshop on Evolvable Hardware*, page 208, Washington, DC, USA. IEEE Computer Society.

Sims, K. (1994). Evolving 3d morphology and behavior by competition. In Brooks, R. and Maes, P., editors, *Artificial Life IV Proceedings*, pages 28–39. MIT Press.

Trimmer, B. (2007). New challengees in biorobotics: incorporating soft tissue into control systems. In *IEEE International Conference on Robotics and Automation*.

Udawatta, L., Watanabe, K., Izumi, K., and Kiguchi, K. (2003). Control of underactuated manipulators using fuzzy logic based switching controller. *J. Intell. Robotics Syst.*, 38(2):155–173.

Valero-Cuevas, F., Yi, J., Brown, D., McNamara, R., Paul, C., and Lipson, H. (2007). The tendon network of the fingers performs anatomical computation at a macroscopic scale. *IEEE Trans Biomed Eng.*, 54:1161–6.

Yim, M., Duff, D., and Roufas, K. (2000). Polybot: a modular reconfigurable robot. In *IEEE International Conference on Robotics*.

Artificial Chemistries

RBN-World: The Hunt for a Rich AChem

Adam Faulconbridge, Susan Stepney, Julian F. Miller and Leo Caves

YCCSA, University of York, Heslington, YO10 5DD, UK
asf500@yccsa.org

Abstract

An Artificial Chemistry (AChem) is a set of components and interactions that result in a composable system. Ideally, the system is rich, and results in rich higher-order emergent properties. We present a methodology for discovering interesting AChems through a series of tests that probe elementary low-level properties. In doing so, we assume that these elementary properties are a necessary, but not sufficient, basis for higher-order emergent properties, such as autocatalytic sets and hypercycles. The test strategy is applied to RBN-World, a sub-symbolic chemistry. This results in identifying a number of new and interesting RBN-World chemistries that appear richer than our original parameterisation.

Introduction

One approach towards the goal of Artificial Life (ALife) has been Artificial Chemistry (AChem), particularly for the origins of life. Unlike many ALife approaches, life-like properties are not explicitly designed in, but emerge from the dynamics of the system. AChems have been applied in other contexts [16, 12] however here we focus on their role as approach to the study of composable systems capable of exhibiting rich higher-order emergent behaviour.

In its most basic form, an AChem is a collection of molecules and reactions that describe transformations between groups of molecules, and an algorithm which determines how the reactions are applied over time [2]. There are a large number of possible AChem designs (relating to the nature of the components, interactions and reactions) each with a potentially large parameter space. Moreover, some examples of emergent systems (Boids [14], Conway's Game of Life [8], etc) only exhibit emergence at a small subset of possible parameters. This motivates the need to develop strategies to search the parameter spaces of AChems to find those regions that exhibit rich emergence.

Here we describe a set of tests suitable for any AChem and apply those tests to filtering 200 alternatives of an AChem — RBN-World [7].

Desired high-level properties

Determining how to evaluate different AChems is a difficult task. The overall goal when developing an AChem for ALife is an emergent system capable of open-ended evolution. The metric for this is unclear; some suggestions include Chemical Organization Theory [1] and Granger causality [15]; however, searching for interesting chemistries using metrics such as these would not be computationally tractable over the large search space of alternative chemistries. Several mid-level properties have been previously suggested as important in the emergence of rich evolutionary characteristics; in the context of artificial chemistry, three of particular relevance are *autocatalytic sets* [11], *hypercycles* [4, 6, 5] and *heteropolymers* or co-polymers [13]. Desirable characteristics of artificial chemistries have been suggested before [17] however, these are design specifications rather than emergent properties.

Autocatalytic Sets An autocatalyst is a molecular species that catalyses its own production. Autocatalytic sets are two or more molecular species where one or more reactions producing each member of the set is catalysed by itself or another member of the set [11]. The members of an autocatalytic set may be, but do not have to be, autocatalysts themselves. In addition, autocatalytic sets may overlap with individual molecular species belonging to more than one set. Autocatalytic sets are thought to be important to the emergence of life because of their characteristic growth; as long as substrate is available, the members of an autocatalytic set will continue to increase in concentration.

Hypercycles Hypercycles are a collection of coupled self-replicative units and thought to be important as a higher-order organization [4, 6, 5] — many biochemical metabolic processes are hypercycles for example.

Heteropolymers Polymers are molecules composed of repeated subunits. Heteropolymers are molecules composed of non-identical subunits, such as DNA or proteins which both have a repeating backbone structure with different side-groups attached to it. The important feature of heteropolymers is their capacity for information storage encoded into the ordering of the subunits.

Desired low-level properties

Searching for autocatalytic sets, hypercycles and/or heteropolymers would be an useful step towards finding artificial chemistries with sufficiently rich emergent properties. However, this is still too computationally intensive to be useful as an initial step. We suggest that the space of possible chemistries can be first reduced by selecting for specific features thought to be required by the higher goal; towards that end, the features being examined should be low-level and computationally tractable.

In order to hunt for rich AChems, we specify tests for low-level properties that we believe are necessary (but possibly insufficient) 'stepping stones' to higher-order emergent behaviour. The tests can be structured, and as a result chemistries that fail the lowest-level tests are not considered for the intermediate tests thus allowing subsequent searches to focus on interesting subspaces.

Synthesis is the formation of bonds is the lowest level property possible; however it is important not only that synthesis can occur in an AChem, but also that too much synthesis does not occur. If every molecule can bond with every other molecule, the chemistry is trivial and will not support rich dynamic higher-level properties.

Self-Synthesis is bonding between two identical atoms or molecules. As with synthesis, this is important for the formation of larger molecular structures but also should be able to occur between any two identical atoms/molecules.

Decomposition should also be possible, but not universal, within the AChem. Without the breakdown of larger molecules, many conceivable mechanisms for higher-level properties become impossible and the system may reach a steady state once all raw materials have been consumed.

Substitution is a potential emergent behaviour given that a particular AChem exhibits synthesis and decomposition. While arguably not important in itself, substitution represents the potential for relationships between more than one or two molecules.

Catalysis is another property of interest. We define catalysis as a series of reactions that do not consume the catalyst, yet the overall reaction would be slower (or not occur at all) without it.

RBN-World: Overview

RBN-World [7] is an AChem framework combining random Boolean networks (RBNs) [9, 10, 3] via bonding sites.

RBNs consist of n nodes synchronously updated in discrete timesteps. Each node in the RBN has a Boolean state, inputs from k nodes, and a Boolean function that maps the state of inputs to an updated state at the next timestep. The state of an RBN is the collection of states of all its nodes. All RBNs have cyclic behaviour, returning to a previous state after sufficient number (usually small) of timesteps.

To use RBNs in a chemistry some modifications have been made — we refer to the modified RBNs as bRBNs (bonding random Boolean networks). Important aspects of these are:

Atoms Within each RBN, there are one or more *bonding sites* (b); these are additional nodes that provide inputs to ordinary nodes. Bonding sites do not have any inputs, instead their state is determined by whether they are "bonded" or "unbonded".

Bonds A *bond* links two bRBNs, and there can be multiple bonds between the same pair of bRBNs. Each bond requires one "unbonded" site within each of the bRBN pair to become "bonded", and each "bonded" site is associated with only one bond.

Bonds are formed as a consequence of *reactions* when specific criteria are met. If a bond is not formed by a reaction, it is attempted again with any higher-level structures (e.g. molecules) that the pair of bRBNs are part of. This iteration of attempting bonding and re-trying for higher-level structures continues until either a bond is formed or there are no more higher structures.

Molecules bRBNs that are linked by bonds can be expressed as a composite bRBN. The composite bRBN's inputs and functions are the component bRBNs with inputs from "bonded" sites are replaced with direct inputs from the other "bonded" node. Non-composed bRBNs are *RBN-atoms*, and a composite bRBN is a *RBN-molecule*. A composite bRBN that is part of a larger composite structure is a *functional group* (by analogy with functional groups in chemistry, such as the amine group). RBN-molecules undergo reactions and form bonds in the same manner as RBN-atoms to make further higher-level composite structures. Note that an internal RBN node can be in different Boolean states at different levels of the structural hierarchy.

Bonding Consequences Forming a bond has two direct consequences:

1. The process of bonding changes a bonding site in each linked bRBN from "unbonded" to "bonded". This changes one input to one node, which can potentially lead to a change in the dynamic behaviour of the Boolean network.
2. The bRBNs linked by the bond form a new higher-level composite bRBN. If one of the participants of the bond was already a component in another bRBN, then the composite structures are combined into a larger composite bRBN.

In addition to the direct consequences, there are potential indirect consequences as well. The formation of a bond may

change the dynamics of either bRBN, which may cause the bonding requirements to be violated. When bonding criteria are no longer valid, bonds *break* and the associated bonding sites reverts to "unbonded". This also alters any higher-level composite structures, collapsing them if they are not distinct from their lower-level components.

Due to the combinatoric nature of Boolean networks, there are a vast number of possible bRBN-atoms. However, only a small subset will lead to the emergence of sufficiently rich properties – most of the chemistry that underpins life is consists of a restricted number of elements: Carbon, Hydrogen, Nitrogen, and Oxygen. Finding analogues of such highly composable elements (and implicitly their interactions) in a particular AChem is our task here.

RBN-World: Alternative Chemistries

During the development of RBN-World, it became clear that a number of modelling decisions had to be made based on limited information; for example, the size of the bRBNs and bonding criteria between them. Also, pragmatically, a number of choices and assumptions were made without explicit consideration of alternatives. These choices may have impact on the emergent properties of the AChem.

To investigate the alternative chemistries, some of the choices have been explicitly defined in order to determine their effect upon the resulting AChem. It is worth noting that the decisions around which alternatives to study have themselves been made based on limited information from preliminary experiments and exploratory ideas.

Four different categories of alternatives have been identified with multiple options within those categories. As well as these separate alternatives, combinations of alternatives from different categories can also be investigated.

Bonding Property

One of the novel aspects of RBN-World is the use of properties of the underlying dynamical system to determine bonding. However, it is not clear which property would be most suitable and what effect different properties might have. Several alternatives are considered here, each with distinct distributions. See tables 1 and 2 for summary and example.

Cyclelength (c) is the number of different states the bRBN passes through between repeats. Cyclelength has a large but bounded asymmetric discrete distribution of values, with a median of approximately $\sqrt{n}$ for small values of k [9].

Flashing counts how many Boolean nodes change state during the cycle. RBNs typically have a 'frozen core' of static Boolean nodes, and flashing is the inverse of this. This can expressed as follows; let a state of 'true' have a value of 1 and a state of 'false' have a value of -1; N be the set of nodes in the bRBN; $s_{i,j}$ be the state of the i^{th} node at the j^{th} state of the repeating cycle. Then:

$$N_{i_{\text{flashing}}} = \begin{cases} 1 & \text{if } \left| \sum_{j=1}^{c} s_{i,j} \right| \neq c \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$N_{\text{flashing}} = \sum_{i \in N} N_{i_{\text{flashing}}} \quad (2)$$

Flashes is the total number of Boolean node state changes over the cycle. As at least one node must change state at each step around the cycle, this is related to the cyclelength and the flashing property. This can be expressed as:

$$N_{\text{flashes}} = \frac{1}{2} \sum_{i \in N} \sum_{j=1}^{c} \left| s_{i,j} - s_{i,j-1} \right| \quad (3)$$

Total is the sum of all Boolean node values at all time steps over the cycle. This is a property of the states of the bRBN rather than its dynamics and is related to the cyclelength property and the number of Boolean nodes.

$$N_{\text{tot}} = \sum_{i \in N} \sum_{j=1}^{c} s_{i,j} \quad (4)$$

Magnitude is the larger out of the total number of Boolean nodes at all time steps over the cycle that are in the 'true' state compared with the number that are in the 'false' state.

$$N_{\text{mag}_\text{T}} = \frac{1}{2} \sum_{i \in N} \sum_{j=1}^{c} (1 + s_{i,j}) \quad (5)$$

$$N_{\text{mag}_\text{F}} = \frac{1}{2} \sum_{i \in N} \sum_{j=1}^{c} (1 - s_{i,j}) \quad (6)$$

$$N_{\text{mag}} = \max\{N_{\text{mag}_\text{T}}, N_{\text{mag}_\text{F}}\} \quad (7)$$

Proportion is the proportion of nodes in state 'true' averaged over both cyclelength and number of Boolean nodes.

$$N_{\text{prop}} = \frac{N_{\text{mag}_\text{T}}}{n \times c} \quad (8)$$

Bonding Criteria

In addition to the bonding property, the bonding rule requires a comparison between the properties of two bRBNs for some criteria to be met. There are multiple possibilities to conduct this comparison, and this is another area for exploration.

Equal is the simplest bonding criteria; form a bond where the value of bonding property is *equal* within 0.1% of the maximum possible range of values to allow for numerical error). This can be expressed as:

$$\frac{p(N_i) - p_{\text{min}}}{p_{\text{max}} - p_{\text{min}}} - \frac{p(N_j) - p_{\text{min}}}{p_{\text{max}} - p_{\text{min}}} = 0 \pm 0.001 \quad (9)$$

Measurement	Minimum	Maximum	Description
Cyclelength	1	2^n	Count of steps on cycle
Flashing	0	n	Count of nodes that change state
Flashes	0	$n \times c$	Count of changes of node states over
Total	$-n \times c$	$n \times c$	Sum of node states over cycle
Proportion	0	1	Proportion of node steps with a value of True on cycle
Magnitude	1	$n \times c$	Maximum count of node states with False/True on cycle

Table 1: Alternative bRBN bonding criteria properties. n is the number of nodes within the bRBN, c is the cyclelength of the bRBN.

	bRBN node			
	A	B	C	D
Cycle steps	F	T	F	F
	F	F	F	F
	T	T	F	F
	T	T	T	F
	T	F	T	F
	F	T	T	F

$c = 6$

$N_{\text{mag}} = 14$

$N_{\text{flashing}} = 1 + 1 + 1 + 0 = 3$

$N_{\text{mag}_\text{T}} = 3 + 4 + 3 + 0 = 10$

$N_{\text{flashes}} = \frac{4 + 8 + 4 + 0}{2} = 8$

$N_{\text{mag}_\text{F}} = 3 + 2 + 3 + 6 = 14$

$N_{\text{tot}} = 0 + 2 + 0 + -6 = -4$

$N_{\text{prop}} = \frac{10}{4 \times 6} = 0.417$

Table 2: Example bonding properties for a $n = 4$ bRBN. Although only one would be used for any specific AChem, here they are all displayed. The table indicates the states of the bRBN nodes at each sequential step on the cycle.

where N_i and N_j are the bRBNs involved in the bond, $p(x)$ is a function to calculate the bonding property of bRBN x, and p_{min} & p_{max} are the minimum and maximum possible bonding property values.

Similar is a relaxation of the equal criteria — i.e. within 5% of the maximum possible range of values.

$$\frac{p(N_i) - p_{\text{min}}}{p_{\text{max}} - p_{\text{min}}} - \frac{p(N_j) - p_{\text{min}}}{p_{\text{max}} - p_{\text{min}}} \leq 0.05 \tag{10}$$

Different is the inversion of similar.

$$\frac{p(N_i) - p_{\text{min}}}{p_{\text{max}} - p_{\text{min}}} - \frac{p(N_j) - p_{\text{min}}}{p_{\text{max}} - p_{\text{min}}} \geq 0.05 \tag{11}$$

Sum one (applicable only to proportion) allows the formation of bonds where the proportion property of the interacting molecules total to one (± 0.001 allowing for numerical error).

$$p(N_i) + p(N_j) = 1 \pm 0.001 \tag{12}$$

Sum Zero (applicable only to total) requires that the total property of the bRBNs sum to a value of zero (± 0.001).

$$p(N_i) + p(N_j) = 0 \pm 0.001 \tag{13}$$

Sum One and Sum Zero are applicable only to proportion and total bonding properties respectively as these are the only bonding properties that can meet these bonding criteria.

n	k	Bonding Property	Bonding Criteria
5	2	Equal	Cyclelength
10	3	Similar	Flashing
15		Difference	Flashes
20			Total
25			Magnitude
			Proportion
		Sum One	Proportion
		Sum Zero	Total

Table 3: Features of the 200 alternative AChems tested. Every chemistry must have one feature from each column. Horizontal lines cannot be crossed within the table when moving from one column to the next. For example, 5 – 2 – Equal – Cyclelength is valid, 20 – 2 – Sum One – Proportion is valid, but 5 – 2 – Sum One – Flashes is not valid.

Sizes of bRBNs

The number of nodes (n) within each bRBN-atom must be chosen. A range of values at intervals was investigated ($n \in \{5, 10, 15, 20, 25\}$ with the potential to expand this range if there appears to be a directional trend).

The size of a bRBN does not have much impact on the chemistry directly. However, it does alter the distribution of the bonding properties, and their responses to bond formation, which in turn affects the propensity for different types of reactions.

Connectivity of bRBNs

Previous work on RBNs [10] has shown that the number of inputs (k) each node has can have an impact on their properties. There is also an interplay with the Boolean function

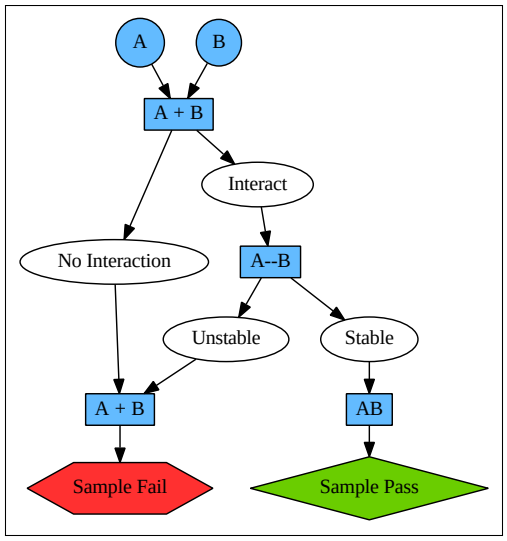


Figure 1: Schematic depiction of one sample for the 'synthesis' test its possible outcomes, and how those outcomes are interpreted as 'pass' or 'fail' for that sample. A & B are two sample atoms. A '+' symbol denotes separate atoms and '–' indicates a potential bond formation between two atoms. Adjacent atoms (e.g. AB) indicates that a bond has formed.

assigned to each node; certain functions can result in one or more inputs having no affect on the state of the node (canalisation) and more different Boolean functions are possible with more inputs.

As an initial assessment, we consider alternatives of two- and three-input bRBNs ($k \in \{2, 3\}$). In theory, any positive integer value equal to or less than the total number of nodes could be used. However, these are values known to be on the 'edge-of-chaos' — higher values are chaotic and lower values are statuc.

Combinations of Alternatives

The alternatives discussed above each change different, but potentially interlinked, aspects of the AChem. Different combinations of alternatives can be used, though some are mutually exclusive. Table 3 shows the possible combinations; in total there are 200 different AChems to be considered, each of which may have potentially different and interesting features.

Previous work [7] used $n = 10$ $k = 2$ with 'cyclelength' as the bonding property and 'equal' for the bonding criterion as an arbitrary initial choice from the 200 alternatives

Method

As discussed previously, there are a large number of potential alternative chemistries, and each of those has a very large number of potential elemental bRBNs.

Due to the vast number of possible bRBNs, exhaustively testing multiple chemistries is not feasible. Therefore, a random sampling approach is taken. In order for a chemistry to be have the potential for sufficiently rich properties, it is important that at the desired low-level behaviours are seen at least once. However, it is also important that the behaviours are not omnipresent — consider the synthesis test for example (described below); if every interaction resulted in the formation of a stable bond, it would rapidly coalesce into a single molecule and would therefore not exhibit sufficiently rich properties.

We do not seek to find the optimal subset of bRBNs in the optimal AChem; we are simply looking to remove those alternative AChems unlikely to exhibit sufficiently rich emergent properties.

Desired Behaviours

As well as the alternative chemistries, the tests for required low-level behaviours must also be defined. There is a natural structuring of prerequisites within the behaviours – decomposition can only occur if synthesis occurs for example. This can be used to increase the efficiency of the sampling.

Synthesis Synthesis is the lowest-level behaviour possible in an atom-based AChem. A pair of atoms is randomly sampled, the two atoms interact, and the outcome is recorded. RBN-World has a two-stage bonding process, and the bonding criteria must be met both at the start of the interaction and after bonding. If a stable bond can be formed, then the sample passes; if not, the sample fails (figure 1).

Self-synthesis The self-synthesis test the synthesis test between two copies of the same element. If a stable bond can be formed, then the sample passes; if not, the sample fails.

Decomposition This is the breaking of bonds, potentially leading to a molecule separating into two (or more) smaller molecules. In RBN-world this is triggered by an interaction between an bRBN molecule and another bRBN. In the decomposition test, samples of three atoms are taken and the first two attempt to form a stable bond. If they cannot form a stable bond, then that sample is ignored for determining pass/fail; this is a test for decomposition, not for synthesis. Once a stable molecule has been formed, it interacts with the third sample. This can have several possible outcomes; no interaction, formation of a larger molecule, or breakdown into two or three separate molecules. If it results in the bond between the first two sampled bRBNs breaking, then it is recorded as a pass; other outcomes are classed as fail (figure 2).

Substitution Similar to decomposition, the substitution test involves an interaction with a molecule that leads to replacement of part of the molecule with the reacting bRBN. The process is the same as the decomposition test, but the only valid outcome is a direct replacement of the second sampled bRBN with the third sampled bRBN, i.e. AC+B in figure 2.

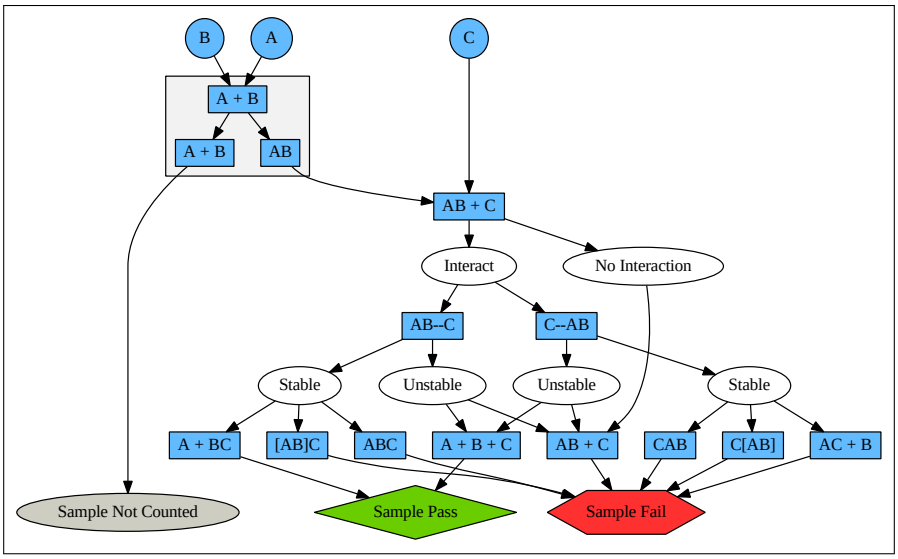


Figure 2: Schematic depiction of the 'decomposition' test. A requirement of the decomposition test is that synthesis must have first occurred, this part of the schematic is indicated in the highlighted subgraph (details removed for brevity).

Catalysis This is the highest-level property investigated here. Unlike the other desired properties, catalysis can take many forms. Any of the other tests could be repeated requiring the presence of a catalyst. For simplicity, we focus on catalysis of synthesis reactions.

The test proceeds as follows: as before, a sample of three bRBNs is taken and the first two attempt to form a stable bond. However, unlike decomposition or substitution tests, this time it is important that a stable bond does not form. If a bond does form, then the sample is not counted for pass/fail.

After that initial bond formation stage, the third bRBN in the sample attempts to form a bond with the first; this is analogues to interacting with a catalyst to form a temporary intermediate. If this does not form a stable bond, then again the sample is not counted for pass/fail.

The final step is to test that the second bRBN from the sample can substitute for the third bRBN. If this is the case, then the third bRBN has acted as a catalyst for the formation of the bond between the first and the second bRBN that would not occur directly (figure 3).

Results

The outcomes of testing the described alternative chemistries with 10,000 randomly generated samples of bRBNs is summarized in table 4 (testing took approx. 2 days on a 24 quad-CPU cluster). With each test a number of alternative AChems are ruled out; the chemistries that pass all tests are listed table 5.

Less than 5% of alternative chemistries pass all the tests. The n & k categories of alternatives have little or no influence on the low-level properties of the chemistry. The anomaly is $n = 25$, $k = 3$ with bonding property 'total' and a comparison of 'sum zero'; however, this may be due to sample size. Closer examination of this case shows that of $10,000$ samples in the decomposition test, $9,677$ were not counted (as the did not form a molecule that could break down) and none of the remaining 323 samples passed. In comparison, the $n = 20$ equivalent AChem where $9,382$ were not counted and 43 of the remaining 618 samples passed.

Test	# of AChems tested	# of AChems where all samples Passed or Not Counted	# of AChems where all samples Failed or Not Counted	# of AChems remaining
Synthesis	200	10	7	183
Self-Synthesis	183	110	53	20
Decomposition	183	0	6	177
Substitution	177	0	18	159
Catalysis	177	0	39	138

Table 4: Results from testing 10,000 samples from each of 200 alternative chemistries for low-level emergent behaviours. The prerequisite for decomposition and self-synthesis tests is synthesis. The prerequisite for substitution and catalysis tests is decomposition. See text for details.

For the property and comparison alternatives, only those using 'proportion' as property and 'sum one' as the criterion or those using 'total' as the property and 'sum zero' as the criterion pass all tests. Whilst alternatives should be kept in mind, we now have evidence that these are options are more likely to be capable of rich emergent properties. As various

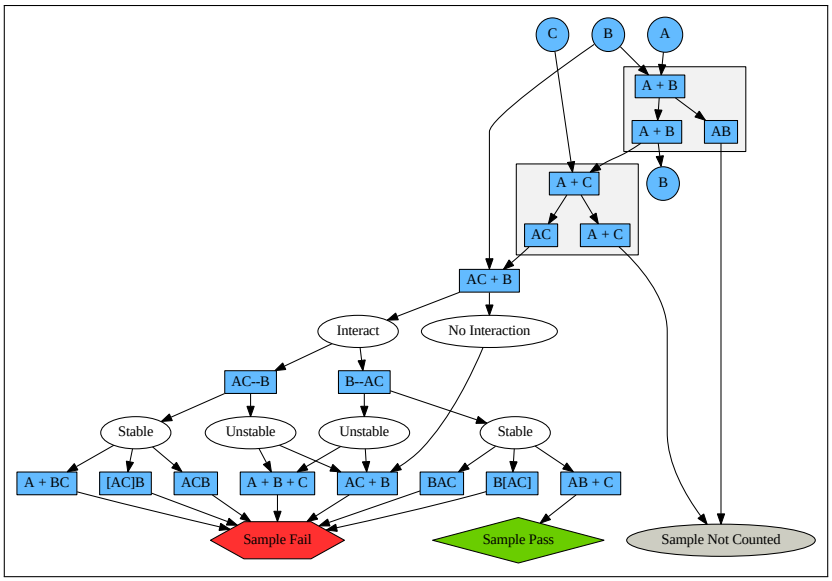


Figure 3: Schematic depiction of the 'catalysis' test. Requirements of the catalysis test are that A and B must *not* synthesise, and that C and A must synthesise; these are indicated by the highlighed subgraphs (details removed for brevity).

n	k	Measurement	Comparison
5	2	Proportion	Sum One
10	2	Proportion	Sum One
15	2	Proportion	Sum One
20	2	Proportion	Sum One
25	2	Proportion	Sum One
5	3	Proportion	Sum One
10	3	Proportion	Sum One
15	3	Proportion	Sum One
20	3	Proportion	Sum One
25	3	Proportion	Sum One
5	2	Total	Sum Zero
10	2	Total	Sum Zero
15	2	Total	Sum Zero
20	2	Total	Sum Zero
25	2	Total	Sum Zero
5	3	Total	Sum Zero
10	3	Total	Sum Zero
15	3	Total	Sum Zero
20	3	Total	Sum Zero

Table 5: The 19 alternative AChems that exhibit variation across all 5 low-level emergent behaviours tested.

different values of n and k were tested and did not affect which chemistries passed the tests, these values can be chosen based on other concerns, such computational tractability. One potential issue is that this work has only samples from atomic constituents; it is not guaranteed that molecular structures will also exhibit these behaviours. While various values of n were tested, molecular bRBNs of many atoms may not behave as an equivalent large bRBN atom due to the constrictions from reciprocal bonding sites between atoms.

Conclusions

We have presented simple tests of an AChem that can be used to restrict the design space to non-trivial chemistries. This is important, as for many AChems there are a large number of alternatives that should be considered – for RBN-World we have only examined a small fraction of possible alternatives. It has also been shown that our initial arbitrary choice of parameters did not pass these tests [7]. This is an important consideration as the processes that lead to the design of an AChem are typically opaque to the community.

A filtering metric provides a useful testing approach that does not require computationally expensive and/or exhaustive testing of molecules and/or reactions. It is also interesting to see that some AChems fail because all tested samples interactions failed, but some chemistries fail because all tested sample interactions passed; the presence of variation is a requirement for emergent properties.

Future work

Two specific alternative parameterisations of RBN-World have been identified as containing interesting atoms; future work can now be focused onto searching for specific small sets of elements within these chemistries that give rise to the high-level desired properties discussed earlier — autocatalytic sets, hypercycles and heteropolymers. These have not been tested for in the experiments described here due to the small samples from each chemistry that were being examined.

In addition, the low-level tests will be refined further. One example is that here only atoms were tested and there is no guarantee that these properties are also applicable for larger structures. As we can now remove the trivial, uninterest-

ing cases, computational effort will be concentrated on those non-trivial cases, in the hunt for rich AChems.

Acknowledgements

Thanks to Jess Wardman for constructive feedback.
This work was funded by BBSRC.

References

[1] Dittrich, P. and di Fenizio, P. S. (2007). Chemical organisation theory. *Bulletin of mathematical biology*, 69(4):1199–1231.

[2] Dittrich, P., Ziegler, J., and Banzhaf, W. (2001). Artificial chemistries-a review. *Artificial Life*, 7(3):225–275.

[3] Drossel, B. (2007). Random Boolean Networks. *Cybernetics and Systems*, 12(1):103–121.

[4] Eigen, M. and Schuster, P. (1977). The Hypercycle: Part A. *Naturwissenschaften*, 64(11):541–565.

[5] Eigen, M. and Schuster, P. (1978a). The Hypercycle: Part B. *Naturwissenschaften*, 65(1):7–41.

[6] Eigen, M. and Schuster, P. (1978b). The Hypercycle: Part C. *Naturwissenschaften*, 65(7):341–369.

[7] Faulconbridge, A., Stepney, S., Miller, J. F., and Caves, L. (2009). RBN-World: A Sub-Symbolic Artificial Chemistry. In *Proceedings of the tenth European Conference on Artificial Life*. Springer LNCS5777.

[8] Gardner, M. (1970). The fantastic combinations of John Conway's new solitaire game "life". *Scientific American*, 223:6–10.

[9] Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of theoretical biology*, 22(3):437–467.

[10] Kauffman, S. A. (1993). *The Origins of Order: Self-Organization and Selection in Evolution*.

[11] Kauffman, S. A. and Farmer, J. D. (1986). Autocatalytic sets of proteins. *Origins of Life and Evolution of the Biosphere*, 16(3-4):446–447.

[12] Meyer, T., Yamamoto, L., and Tschudin, C. (2008). An Artificial Chemistry for Networking. *Bio-Inspired Computing and Communication*, 5151:45–57.

[13] Odian, G. (2004). *Principles of Polymerization*. John Wiley & Sons, Inc., Hoboken, NJ, USA.

[14] Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*.

[15] Seth, A. (2010). Measuring Autonomy and Emergence via Granger Causality. *Artificial Life*, 18(Early Access):118.

[16] Straatman, B., White, R., and Banzhaf, W. (2008). An Artificial Chemistry-based Model of Economies. *Artificial Life*, 11:592.

[17] Suzuki, H., Ono, N., and Yuta, K. (2003). Several necessary conditions for the evolution of complex forms of life in an artificial environment. *Artificial Life*, 9(2):153–174.

Automatically Moving Between Levels in Artificial Chemistries

Adam Nellis and Susan Stepney

York Centre for Complex Systems Analysis, University of York, UK, Y010 5DD
adam@cs.york.ac.uk www.yccsa.org

Abstract

We introduce multi-level Artificial Chemistries as a way of tackling difficult problems in the evolution of complexity. We present two algorithms for moving between levels of abstraction in a multi-level Artificial Chemistry. (1) Moving upwards from a low-level description to a high-level description involves making approximations. We discuss these, and provide an algorithm to perform the approximations. (2) Moving downwards is more problematic. We discuss the issues involved in moving down, including conservation of mass. We present an algorithm to generate constraints that any low-level implementation of the system must satisfy. These constraints can be used to: obtain information about the system; automatically generate a low-level implementation of the system; guide a search for suitable low-level implementations of the system.

Introduction

Artificial Chemistries (AChems) can be explored from a computational viewpoint, for example, as tools for implementing evolutionary algorithms [9] and controlling robots [6]. They can also be used to model biological systems [10] such as replication [12] and membrane formation [13]. These varied applications of AChems lead to varied ways of defining them, and consequently to AChems defined on different levels of abstraction, with different properties. However, one common feature among AChems is that they are defined on only one level. Some problems, relevant to both computation and biology, span two or more levels of abstraction (for example, any of the 'major transitions in evolution' [14]). If AChems are to tackle these problems, they must span multiple levels of abstraction.

Previous authors have observed that biological systems contain components on different levels [3], but the purpose of multi-level AChems is to produce two different models of the same system, from two different levels. Work has been done on Course-Grained Molecular Dynamics [1] and Dissipative Particle Dynamics [11], which move from the very low level simulations of Molecular Dynamics, upwards to a slightly higher level that is more computationally tractable for larger molecules and longer timescales. But these systems still only operate on one level. Currently there is no well-defined way for the AChem itself to move between levels of abstraction. We discuss the issues involved in moving between levels of abstraction, and present two algorithms to aid movement up and down levels of abstraction in AChems.

Traditionally, people use computers to do the 'work' of running the AChem, and themselves do the 'meta-work' of deciding at which level to run. But what if computers could do this 'meta-work'? A system that could automatically decide which level to model at could attempt to tackle some of the difficult modelling challenges that span multiple levels, such as the 'major transitions in evolution'. Here we discuss both moving downwards from a higher level to a lower level and moving upwards from a lower level to a higher level.

The higher level is an *approximation* of the lower level. The lower level contains more information than the higher level, and so moving downwards requires adding this information into the system. When moving downwards, we do not know how the lower level is implemented. We only know how it must behave when viewed from a high level. So we cannot map directly from a high-level description to 'the correct' low-level description. In this paper, we map to a set of *constraints* that any low-level implementation must satisfy. These constraints describe how the low-level components of the system combine to form high-level structures.

The constraints could then be used to guide an implementation of the lower level. For some low-level implementations, these constraints correspond almost directly to an implementation (with possibly some arbitrary choices to be made). For more involved low-level descriptions, these constraints can be used to search for low-level implementations.

When moving up from a low-level description to a high-level description, an approximation must be made. The purpose of having a high-level description of a system is that there is too much information in the low-level description, and a summary of this information is desired. The high-level description approximates this information in a meaningful way. We must decide precisely how to approximate the system and how much to approximate it. An algorithm is presented for performing this approximation, and the issues surrounding approximation are discussed.

What is an Artificial Chemistry?

AChems are agent-based systems where the agents are analogues of chemicals participating in reactions. There are different types of AChem [4] with varying levels of complexity. The simplest are defined by finite lists of chemical types and the reactions they can participate in. More sophisticated AChems define chemicals containing some internal structure or properties. This makes it possible to describe an infinite number of different chemicals using a finite number of properties [10]. The reactions in these systems do not need to be explicitly listed; they are defined implicitly by the structure and properties of the chemicals, and specific reactions can be computed as and when they are needed.

When defining reactions implicitly, the possibility exists for open chemistries [7]. In an open chemistry, the possible chemical species that can exist need not be pre-specified. Although many different chemical species are possible, only a small number of them exist at any one time. A particular instance of the chemistry occupies a sub-space of the space of all possible chemical species. As an open chemistry runs, it changes the sub-space that it occupies.

If an AChem is to be used to evolve a network of chemicals and reactions, an open chemistry is required. Additionally, the chemistry should also be *evolvable*: the chemical species should change (via mutation) in a structured way that evolution can use to move through the space of possible chemical species. Most changes should have only a small effect (so a mutated chemical can perform the same reactions as its parent, but maybe faster or slower), but some changes should have a large effect (occasionally a mutated chemical can perform a new reaction, or lose the ability to perform an existing reaction).

One way of making evolvable chemistries is to use subsymbolic chemistries [5], where chemicals have two levels of description. On the higher level, the system is an open AChem with chemical species containing structure and rules that define their reactions. On the lower level, a chemical is composed of parts that interact to give rise to properties that entail the rules on the higher level. The lower level could be a complex system such as a random boolean network [5], it could be another AChem (for example a simple, closed chemistry), or it could be a computer programming language [8]. AChems that work on two or more levels have the potential to possess properties such as evolvability.

What are levels?

There is no 'correct' level at which to design AChems, as it depends on the particular problem being solved. This includes whether the purpose of using the AChem is to simulate a system from actual chemistry (or biochemistry), or to use the AChem as a computational tool, exploiting its properties to create a computational system (or to study a computational system). But there are some problems that involve crossing levels. For example, actual chemistry has gone through events crossing levels at different times during the evolution of life (the 'major transitions in evolution' [14]), for example: naked replicating molecules becoming encased in compartments and replicating as populations; RNA acting as both genes and enzymes, changing to use DNA as genes and proteins as enzymes; and the evolution of multi-cellular organisms from single-celled organisms. These kinds of problem may be interesting to systems biologists wanting to better understand what happened in real chemistry/biology. They may also be interesting to people wanting to use AChems for computational purposes, as they are examples of natural systems increasing their own complexity, something that current artificial systems find difficult to achieve.

All of these problems involve concepts at two (or more) levels. Choosing the most appropriate level at which to model is not easy. Addressing these problems (from the point of view of either biology or computation) involves one of two options: either modelling and simulating the whole system from the lower level, and enduring the computational burden that this entails; or modelling the system on both levels simultaneously, switching between the two levels in a multi-level chemistry. To automate the second option requires a well defined way of moving between the levels.

Going downwards

The concept of multi-level chemistries can provide new ways of thinking about high-level, symbolic, chemistries (lists of chemicals as symbols, and their reactions). Any symbolic chemistry describes a system at a certain level.

For systems that are models of the real world, there is always a lower level of description that the system could be described on (until we reach the level of our understanding of particle physics). Also, for real world systems, some information about this lower level is always known (we know that organisms are composed of cells, which are composed of molecules, and so on.)

For artificial systems, however, the implementation of any level is arbitrary (and is often chosen to make the program execute efficiently). So when describing an artificial system in terms of a lower level, there are arbitrary implementation choices to be made, some of which are constrained by the higher level. Looking for these constraints can give insight and information about the higher level, and resolve some of the seemingly arbitrary design choices for the lower level. These kinds of insight can also be gained about real systems as well as computational ones.

The high-level entities are symbols. On the lower level, each of these high level symbols is expressed as a collection of lower-level components. For example, the decomposition of hydrogen peroxide into water and oxygen can be written as:

$$2\ \text{hydrogen-peroxide} \rightarrow 2\ \text{water} + \text{oxygen} \qquad (1)$$

But the same equation can be written in terms of the lower level of atoms, instead of in terms of the higher level of molecules:

$$2\,H_2O_2 \rightarrow 2\,H_2O + O_2 \quad (2)$$

Here, 'hydrogen-peroxide' is a symbol on the higher level, that is expressed as two 'H' components and two 'O' components on the lower level. Note also the constraint: 'oxygen' and 'hydrogen-peroxide' are different symbols on the higher level, but they share common components on the lower level: 'hydrogen-peroxide' has the same components as 'oxygen', along with some other components.

On the high level, information about the system is contained in the reaction equations. On the low level, it is contained in the structure of the chemicals (how their components are arranged). So the task of describing a high-level system on a lower level is about *moving information from reaction equations to chemical structures.*

This movement can be performed by humans looking at reaction equations and diagrams. But as the lists of equations become longer and the number of different symbols increases, the problem becomes harder and more tedious to solve. Also, if evolutionary algorithms are to evolve symbolic systems, then this problem needs to be solved hundreds of times for each generation of the evolutionary algorithm. This is why it is useful to have an algorithm for automatically performing this process.

Conservation of mass

The above reasoning relied on the assumption that 'mass' is conserved in the high-level reaction equations: if $\alpha+\beta \rightarrow \gamma$, then all the low-level components making up α and β are present in γ, and γ contains no new components that have not come from α or β.

This is not a difficult condition to fulfill on the high level, as new symbols can be introduced to account for any mass gained or lost in a reaction. For example, if $\alpha + \beta \rightarrow \gamma$, but mass is lost ($\gamma$ does not contain all of the components of α and β), then $\alpha + \beta \rightarrow \gamma$ can be replaced by $\alpha + \beta \rightarrow \xi + \gamma$, where the symbol ξ does not appear anywhere else in the system. ξ represents the mass that is lost in the reaction. Likewise, if mass is gained in the reaction (γ contains a component that does not come from α or β), then $\alpha+\beta+\zeta \rightarrow \gamma$ can be used, where ζ represents the mass gained in the reaction. These two patterns can be applied to any reaction. If they are applied at the same time, they can represent reactions in which some components are lost and some are gained.

Given a high-level system of reaction equations that conserve mass, we can deduce constraints on how the high-level symbols are composed of low-level components. We can also put constraints on the possible masses that the symbols can have. Technically, we deduce a partial order on the masses of the symbols, with constraints of the form: 'χ has more mass than ψ'. We can also use this to work out if a system conserves mass or not, so we do not need to know beforehand. If we encounter a contradiction when building the partial order, we have proved the system does not conserve mass. If we can build the partial order with no contradictions, then we have proved the system does conserve mass.

Multiple meanings

Some high-level reaction equations can have more than one interpretation on the lower level. These can be disambiguated by modifying the reaction equations to include intermediate steps. Different disambiguations lead to different low-level constraints for the same high-level system.

3 chemicals or fewer — unambiguous reactions

There are five kinds of reaction equation that have only one interpretation on the lower level: they involve three molecules or fewer.

1. nothing $\rightarrow \alpha$ (influx)
2. $\alpha \rightarrow$ nothing (outflux, or decay)
3. $\alpha \rightarrow \beta$ (isomerisation)
4. $\alpha + \beta \rightarrow \gamma$ (composition or association)
5. $\gamma \rightarrow \alpha + \beta$ (decomposition or dissociation)

Reaction types (1) and (2) give no information about the lower level (other than saying "α is a symbol that exists"), so are ignored in later analysis.

4 chemicals

Four chemicals participating in a reaction can have more than one interpretation on the lower level.

3 $\rightarrow$ 1 reactions. Reactions of the form $\alpha+\beta+\gamma \rightarrow \delta$ imply that chemical δ is a composite of chemicals α, β and γ. The ambiguity lies in the order in which α, β and γ combine to form δ. Because the probability of three molecules reacting with each other at the same instant is negligibly small, two of α, β and γ must react first, the other one reacting with the intermediate complex, ξ. There are three possibilities:

$$\alpha + \beta \rightarrow \xi\,; \quad \xi + \gamma \rightarrow \delta \quad (3)$$
$$\alpha + \gamma \rightarrow \xi\,; \quad \xi + \beta \rightarrow \delta \quad (4)$$
$$\beta + \gamma \rightarrow \xi\,; \quad \xi + \alpha \rightarrow \delta \quad (5)$$

If $\alpha + \beta + \gamma \rightarrow \delta$ were the only reaction in the system, then these three disambiguations would be equivalent. But if α, β and γ participate in other reactions, then the order in which they combine to form δ could have implications on the lower level.

$1 \rightarrow 3$ reactions. Similarly to the $3 \rightarrow 1$ reactions, reactions of the form $\alpha \rightarrow \beta+\gamma+\delta$ can also have multiple interpretations. The chemical α must be composed of chemicals β, γ and δ, and so it must be composed of their low-level components, held together in a certain structure. It must release one of chemicals β, γ and δ first, which implies the existence of an intermediate chemical, ξ, that is the combination of two of β, γ and δ. There are three possibilities:

$$\alpha \rightarrow \beta + \xi\,; \quad \xi \rightarrow \gamma + \delta \tag{6}$$
$$\alpha \rightarrow \gamma + \xi\,; \quad \xi \rightarrow \beta + \delta \tag{7}$$
$$\alpha \rightarrow \delta + \xi\,; \quad \xi \rightarrow \beta + \gamma \tag{8}$$

$2 \rightarrow 2$ reactions. Reactions of the form $\alpha+\beta \rightarrow \gamma+\delta$ can have multiple interpretations, but these interpretations are of a different kind from those above. The earlier interpretations are about the order in which three chemicals come together to form a complex (or come apart from a complex). The interpretations for $\alpha+\beta \rightarrow \gamma+\delta$ reactions concern symbols being transformed into other symbols, which corresponds, on the lower level, to chemicals undergoing isomerisations. There are three possibilities for how this isomerisation can occur:

1. α is an isomer of γ; and β is an isomer of δ.
2. α is an isomer of δ; and β is an isomer of γ.
3. Both α and β contain some components of γ and δ.

Depending on precisely how the lower level will be implemented, point (3) may or may not be possible.

For the purpose of reducing every ambiguous reaction to unambiguous reactions, the reaction $\alpha + \beta \rightarrow \gamma + \delta$ can be replaced with the two reactions:

$$\alpha + \beta \rightarrow \xi\,; \quad \xi \rightarrow \gamma + \delta \tag{9}$$

This again introduces an intermediate complex, ξ. Replacing the equation in this way does not remove the underlying ambiguity. We must make another disambiguation by choosing one of the three cases above.

More than 4 chemicals

In the same way that reactions involving four chemicals can be reduced to unambiguous reactions involving three chemicals or fewer, reactions with more than 4 chemicals can be reduced to unambiguous reactions by the repeated application of the above reductions.

Disambiguation

The first step in the analysis of a high-level system is to pre-process the reactions, reducing them to unambiguous reactions. This involves making choices about how to decompose ambiguous reactions, as described above. If only one ambiguous reaction needs to be decomposed, then the choice made is somewhat arbitrary. But if multiple choices need to be made, then there is the possibility that choices can affect each other.

There is no way in which choices can be incompatible with each other: any set of choices will always lead to a valid disambiguation, and every disambiguation can always be reversed (by removing the intermediates) to return to the same set of ambiguous equations. However, different disambiguations of the same equations can differ in the number of intermediates introduced. If two reactions need to be disambiguated, then this will introduce two new intermediate symbols (one for each reaction). These intermediates are different symbols on the high level, but if there is extra information in the system about the reactants and products of the ambiguous reactions, then it may be possible to relate the intermediates on the lower level, seeing them as isomers of each other (i.e. realising they are composed of the same components). If, however, the equations were disambiguated using different choices, then it might not be possible to relate the intermediates on the lower level. This can also carry over to some of the non-intermediate symbols as well. One disambiguation may make it possible to infer that two non-intermediate symbols are isomers of each other, but a different disambiguation may not make it possible to infer this.

Note that this is not a mistake in the disambiguation process: it is a choice that must be made about how to interpret the high-level equations. If an equation is ambiguous about how one reaction happens, then this ambiguity can carry over to other parts of the system. If application-specific information is available about how ambiguous equations should be disambiguated, then they can be disambiguated by hand before running the analysis. Or if the equations are being generated by a computer program, then this program can be instructed to produce unambiguous equations of the correct form. If it is not known which way the equations would be best disambiguated, then any disambiguation will give a valid representation of the equations. If there is reason to believe that one representation will be better than the others, but it is not known which, then all disambiguations can be enumerated. The analysis can be run on all disambiguations and the results compared to see if multiple representations are possible. If the most compact representation is desired (i.e. the representation that sees the greatest number of symbols as isomers of each other), then this can be found by comparing the different representations. The fact that multiple representations are possible via different disambiguations, highlights the fact that the lower level contains more information than the higher level. Thus we can not map directly to a low-level description from a high-level description; we can only obtain constraints on the lower level.

Algorithm

Once the high-level set of reaction equations has been disambiguated, they can be reasoned about to obtain constraints on the low-level implementation of the system. This reasoning will give us:

- L : a list of low-level components.
- H : a list of the high-level symbols and how they are composed of low-level components.
- I : a list which high-level symbols are isomers of each other.
- P : a partial order on the masses of the components and symbols.

The low-level components here represent constraints on how the lower level must be implemented. These components are no more than the high-level symbols that do not need to be broken down into other symbols. (If a high-level symbol does not need to be broken down on the lower-level, it does not mean that it must not be broken down; it just means there is no information in the high-level reaction equations requiring it to be broken down.)

A set of high-level reaction equations can be thought of as an implicit description of how some symbols in the system are composed of other symbols. The purpose of this algorithm is to make this implicit description explicit. This uses a form of unification [2]. The word 'unification' has a specific meaning in Computer Science (that applies here), but it can be thought of more generally as a way of taking information that is implicit and spread out; making it explicit and bringing it into one place. In this situation, the information is implicitly spread throughout the high-level reaction equations. We are bringing it into an explicit description of how the high-level symbols are composed of low-level components. Off-the-shelf unification algorithms are not suited to this particular situation, as here there is only one function (composition), and it is commutative. So we have designed a special-purpose unification algorithm (algorithms 1 and 2) to exploit the structure of this problem.

Algorithm 1 — set-up

Before we can perform the unification, we need some equations to unify. These will be of the form $\alpha = \beta + \gamma$, representing the fact that the high-level symbol α is composed of the same low-level components as a β symbol combined with a γ symbol. These equations are stored in the data structure D. After the pre-processing steps of disambiguation and removal of influx and outflux reactions, we have isomerisation, composition and decomposition reactions. Algorithm 1 processes these reactions, putting their information into the data structures D, I and P. The decomposition reactions are added as-is into D; the composition reactions are reversed, and added to D. The isomerisation reactions do not need to be put into D, instead their information can be put directly into I. As each equation is added, its information about the partial order on the masses is added to P. When every equation has been processed, the unification can begin. We check the partial order to see if the system conserves mass, and stop now if it does not (because the unification would fail). If there is a contradiction in the partial order, then the high-level system does not conserve mass. If there is not a contradiction then this does not necessarily mean that the system does conserve mass; there is another conservation of mass check during the unification.

Algorithm 1 The first half of the 'downwards' algorithm: Setting up the decompositions of symbols.

```
P := ∅ {partial order on the masses}
I := ∅ {high-level symbols that are isomers}
D := ∅ {decompositions being unified}
for all reaction in high-level-reactions do
  if reaction is α → β {isomerisation} then
    add isomer 'α = β' to I
    add order relation 'α = β' to P
  else if reaction is α + β → γ {composition}
       or γ → α + β {decomposition} then
    add decomposition 'γ = α + β' to D
    add order relations 'α < γ' and 'β < γ' to P
  if there is a contradiction in P then
    return failure: the system does not conserve mass
```

Algorithm 2 — unification

After set-up stage, the data structure D is filled with the equations to unify. Algorithm 2 performs this unification and completes the 'downwards' algorithm. D contains a list of equations of the form $\omega = \chi + \psi$, where ω, χ and ψ are symbols from the high-level system (or intermediates generated by disambiguation). The equation $\omega = \chi + \psi$ means that the symbol ω is composed of the same low-level components as the symbols χ and ψ. But D could contain another equation: $\omega = \tau + \upsilon$. These two equations both describe how ω is composed, and need to be considered together during this step of the algorithm. During this step we iterate through the equations in D, grouping together all equations describing the same symbol (e.g. ω). So in a typical iteration we might consider the decompositions $d = d_1 = d_2$, where d is ω, d_1 is $\chi + \psi$ and d_2 is $\tau + \upsilon$. So the notation $d = d_1 = d_2$ means that we are considering the two equations, $\omega = \chi + \psi$ and $\omega = \tau + \upsilon$.

For each of these sets of decompositions, we apply one of five operations (in order) to simplify the equations. This process is iterated until no equations remain. Then the equations have been unified and the process is complete. WHen simplifying the equations, we may find a way to partially decompose a symbol but not know its full decomposition yet. This information is stored in the temporary variable PA,

which is like H but stores partial information about decompositions. The operations that we perform are:

1. If a symbol has only one decomposition ($d = d_1$) then this symbol has been fully decomposed. We add this decomposition to H, including any partial decomposition already done to d.

2. If there are common symbols in the decompositions of a symbol ($d = \chi + \psi = \tau + \psi$) then we cancel these and add them to the partial decomposition of d.

3. If any decompositions of a symbol contain only one symbol themselves, then we can cancel them. We remove all but one of these decompositions from D, and add into I the fact that these symbols are all isomers of each other. We also update the partial order, P, with the fact that these symbols all have the same mass (and we check the partial order for contradictions).

4. Because different symbols can be isomers of each other, we replace all instances of these isomers with a common identifier so they can be cancelled from the equations by operation 2.

5. If none of the above operations can be performed, then we search the decompositions for the first symbol that we know how to decompose (it has an entry in H). We replace this symbol in its equation by its decomposition. So if $\omega = \psi + \chi = \tau + \upsilon$ and $\tau = \rho + \sigma$ then we end up with $\omega = \psi + \chi = \rho + \sigma + \upsilon$.

After all the equations have been unified, the set of low-level components, L, can be read off as those high-level symbols that can not be decomposed (are not in H) and are not isomers of a different high-level symbol (are not in I).

Algorithm 2 The second half of the 'downwards' algorithm: Unifying the decompositions

```
L := ∅ {low-level components}
H := ∅ {high-level symbols to be broken down}
PA := ∅ {partial decompositions}
while D is not empty do
  for all d = d_1 = d_2 = ··· = d_n in D do
    if n = 1 then
      add decomposition 'd = PA(d) ∪ d_1' to H
      remove decomposition 'd = d_1' from D
    else if common symbols in d_1 = d_2 = ··· = d_n
    then
      cancel the common symbols
      add the common symbols to PA(d)
    else if more than one of d_1 = d_2 = ··· = d_n are
    length 1 then
      s_1 = s_2 = ··· = s_m are these decompositions
      for all unique pairs s_i, s_j do
        add isomer 's_i = s_j' to I
        add order relation 's_i = s_j' to P
        if there is a contradiction in P then
          return failure: system not conserve mass
      remove all but one of s_1 = ··· = s_m from D
    else if at least one of d_1 = d_2 = ··· = d_n contains a
    chemical in S then
      for all matching chemicals c do
        replace c with its common identifier from I
    else
      find the first d_i in d_1 = d_2 = ··· = d_n with a
      match in H
      replace 'd = d_i' in D with 'd = H(d_i)'
L := {all high-level symbols} \ (H ∪ I)
return success: L, H, I, P
```

Going upwards

Moving up from a low-level system to a high-level system is more straightforward than moving down.

The precise implementation details of the lower level system do not matter for the process of moving up to the higher level. However the low-level system is implemented, it will consist of components that interact with each other and join together to form structures. (For example, two hydrogen atoms and one oxygen atom may join to form a water molecule structure.) These structures are symbols on the higher level. The reactions on the higher level summarise the low-level mechanisms by which these structures interact. To produce a high-level description of a low-level system, two things are needed: (1) a list of high-level symbols; and (2) a list of reactions involving these symbols. The symbols represent the structures formed by the low-level components and the reactions represent the dynamics happening on the lower level. Algorithm 3 gives the pseudocode of an algorithm to do this.

To produce a list of symbols, it is necessary to simulate the low-level system and observe the structures that form. The length of time the system is observed for has an impact on the structures observed. If very involved structures could form within the system but they take longer to form than the system is simulated for, then they will not be observed. Likewise if some structures form quickly but rarely, they may not be observed if the system is not simulated for long enough. This highlights the fact that the high-level system is an approximation of the low-level system, capturing those structures that form within a certain timescale.

There is another timescale associated with the observation of the low-level system. When observing structures within the system, a short timescale must also be chosen. Because the low-level components are constantly interacting with each other, a complicated structure goes through intermediate stages in its formation. These intermediate stages may not be appropriate to represent in the high-level system: the only thing required may be the resulting structure. The

Algorithm 3 Going upwards from a low-level description to a high-level description of a system.

```
S := ∅ {set of high-level symbols}
R := ∅ {set of high-level reactions}
while long timescale has not expired do
  while short timescale has not expired do
    simulate low-level system
  observe low-level system
  for all new structures not seen before do
    create a new symbol for this structure
    add this new symbol to S
  for all structures, S, at the start of this timescale do
    if S was in a reaction during this timescale then
      A := { structures that S reacted with }
      B := { products remaining after these reactions }
      create a new symbolic reaction: A → B
      add this reaction to R
```

separate, simple operations happening on the low level are combined into one complicated operation on the high level. This again highlights the fact that the high-level system is an approximation of the low-level system. A complicated structure that forms through intermediate stages on the lower level springs into being in one step on the higher level.

Observation of the low-level system gives a list of reaction equations as well as a list of symbols. The short timescale is used to approximate a series of intermediate structures by one symbol: the end product of the series. This approximation gives a reaction equation. Whatever structures were present in the area of interest at the start of the short timescale are the reactants in the reaction equation, and whatever structures were left over after the short timescale are the products of the reaction equation. Thus the observation of symbols also gives a list of reaction equations. For a new symbol to be observed, there must have been a process taking place by which the symbol was formed. This process is observed and approximated by the short timescale. This gives a new symbol (or symbols), and a reaction creating the symbol(s). Repeating this observation of the low-level system for the whole duration of the long timescale gives a list of high-level symbols and a list of reaction equations. This is a high-level description of the system.

Conclusions and future work

Building AChems on multiple levels provides more flexibility than using just one level. It may provide a way of approaching difficult problems in the evolution of complexity, such as the 'major transitions in evolution' [14]. This paper presents some initial thoughts about moving between levels, and some algorithms that allow systems to automatically move between levels.

An algorithm is presented for moving down from a high-level description of a system to a lower level of description. Conservation of mass is needed in the high-level system in order to infer information about the low-level system. The algorithm can be used to determine if a system conserves mass or not. If the system does conserve mass, then the analysis can be performed. If it does not, then the algorithm can be used to determine precisely which parts of the system do not conserve mass. Since a high-level description is an approximation of a low-level system, this algorithm generates a set of constraints that any low-level implementation of the system must satisfy. Depending on the precise way in which the low-level system is implemented, this either provides a way of generating an implementation, or it provides a criterion that can be used to search for an implementation.

We will use this algorithm to investigate different low-level implementations of AChems. We have developed some implementations where the constraints generated by this algorithm map directly into low-level descriptions. We also have some sub-symbolic representations [5] where these constraints can be used to search for sub-symbolic chemistries that fulfil the high-level description.

Some high-level systems are ambiguous as to how they operate on the low-level. This algorithm can be used on different disambiguations of the high-level system to give information about the system. We will build a tool to show which parts of the system are most ambiguous, and which are most constrained on the lower level. This information may be helpful, particularly in guiding algorithms that are searching for low-level implementations.

The algorithm introduces intermediate chemicals into the system to disambiguate reactions. A consequence is that reactions happening in one step in the high-level system can take multiple steps to happen in the low-level system. Intermediates can interact with other parts of the system, disrupting the reaction. Things not possible in the high-level system become possible by moving to a lower level of description. So some richness is added into the system by a low-level description, which may be useful to other processes that are exploiting the AChem. For example, the extra richness can provide more ways in which to evolve the reactions.

There is a further part to the 'downwards' algorithm, which we will develop. As well as knowing how high-level symbols are composed of low-level components, it would be useful to know precisely how these low-level components are connected together. If we consider the low-level components connected to each other by 'binding sites', then we can work this out. Each binding site has an affinity to each other binding site, and each component can have many binding sites. Components binding to each other can cause new binding sites on the components to become available, and existing ones to become unavailable. After running the presented 'downwards' algorithm, we have enough information to work out how many binding sites each low-level component needs to have, and which sites must be able to bind with which others. If the high-level reaction equations come with

reaction rates, then in principle we could carry these rates through the algorithm and work out values for the affinities on the binding sites (although this would require knowing how the kinetics will be simulated on the lower level). The concept of binding sites shows further richness gained by using a low-level description of a system. Rather than listing which components must come together to form which high-level symbols, we just list which binding sites each component must possess. The high-level symbols come out from the low-level system as a consequence of the binding sites possessed by each component in the system. Creating a new component only involves creating an arrangement of binding sites (with affinities to sites already in the system). Adding a new component to an existing system changes the high-level structures the system can form.

An algorithm is also presented for moving up from a low-level description to a high-level description. A high-level description is described as an approximation of the low-level description, and this approximation is made precise by the description of two different timescales that constitute this approximation. A short timescale is used to approximate the interactions and intermediate structures on the lower level into symbols and reactions on the higher level. A long timescale is chosen to give a period over which the low-level system will be observed, and only those events occurring within this time period will be approximated.

We will link the two algorithms presented here. One way to do this is with a heuristic search algorithm operating on two different levels. A search algorithm (such as an evolutionary algorithm) is used to search for an AChem to solve a particular problem. A common issue encountered when designing heuristic search algorithms is which problem representation to choose. This issue can be somewhat avoided by representing solutions to the problem as two-level AChems. The search algorithm can search through different high-level representations of the AChem until it becomes stuck in a local optimum. It can then switch to the low-level representation and search in this representation for a time (perhaps until it becomes stuck in another local optimum). Now, it can move back to the high-level representation. When it does this it will not only find itself in a different part of the high-level search space, but it may find itself in a different high-level search space altogether. Because the low-level representation can easily create new high-level symbols, moving down to the low-level description and running the search will change the symbols that exist on the high-level, and change the relationships of existing symbols to each other. Likewise, running a search on the high level and moving back down to the low level has the potential to change the type of low-level representation that will be generated by the 'downwards' algorithm. This searching on two levels effectively co-evolves two different problem representations. It is just one way in which the two tools provided by this paper can be used.

Acknowledgements

The authors would like to thank the other members of the Plazzmid project at the University of York for valuable comments and ideas throughout the work: Ed Clark, Simon Hickinbotham, Peter Young, Tim Clarke and Mungo Pay.

This work is part of the Plazzmid project, funded by EPSRC grant EP/F031033/1

References

[1] Anton Arkhipov, Peter L. Freddolino, and Klaus Schulten. Stability and dynamics of virus capsids described by coarse-grained modeling. *Structure*, 14(12):1767–1777, 2006.

[2] F. Baader and W. Snyder. Unification theory. In J. A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume 1, pages 447–533. Elsevier Science, 2001.

[3] M. Conrad. Cross-scale interactions in biomolecular information processing. *Biosystems*, 35(2-3):157–160, 1995.

[4] P. Dittrich, J. Ziegler, and W. Banzhaf. Artificial chemistries–a review. *Artificial Life*, 7(3):225–275, 2001.

[5] Adam Faulconbridge, Susan Stepney, Julian Miller, and Leo Caves. RBNWorld: Sub-symbolic artificial chemistry. In *ECAL 2009, Budapest, Hungary, September 2009*. LNCS. Springer, September 2009.

[6] Verena Fischer and Simon Hickinbotham. A metabolic subsumption architecture for cooperative control of the e-puck. In *To appear in: International Workshop on Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*. Springer, 2010.

[7] W. Fontana. Algorithmic chemistry. *Artificial Life II*, pages 159–210, 1992.

[8] Simon Hickinbotham, Edward Clark, Susan Stepney, Tim Clarke, Adam Nellis, Mungo Pay, and Peter Young. Molecular microprograms. In *ECAL 2009, Budapest, Hungary, September 2009*. LNCS. Springer, September 2009.

[9] Simon Hickinbotham, Edward Clark, Susan Stepney, Tim Clarke, and Peter Young. Gene regulation in a particle metabolome. In *CEC 2009, Trondheim, Norway, May 2009*. IEEE Press, May 2009.

[10] W. S. Hlavacek, J. R. Faeder, M. L. Blinov, R. G. Posner, M. Hucka, and W. Fontana. Rules for modeling signal-transduction systems. *Science's STKE : signal transduction knowledge environment*, 2006(344), 2006.

[11] P. J. Hoogerbrugge and J. M. V. A. Koelman. Simulating microscopic hydrodynamic phenomena with dissipative particle dynamics. *EPL (Europhysics Letters)*, 19(3):155–160, 1992.

[12] Tim J. Hutton. Evolvable self-replicating molecules in an artificial chemistry. *Artificial Life*, 8(4):341–356, October 2002.

[13] Duraid Madina, Naoaki Ono, and Takashi Ikegami. Cellular evolution in a 3D lattice artificial chemistry. In *Advances in Artificial Life*, pages 59–68. 2003.

[14] John Maynard Smith and Eörs Szathmáry. *The major transitions in evolution*. Perseus Books, 1999.

Catalytic Search in Dynamic Environments

Lidia Yamamoto[1] and Wolfgang Banzhaf[2]

[1]Computer Science Department, University of Basel, Switzerland
[2]Memorial University of Newfoundland, St. John's, NL, Canada
Lidia.Yamamoto@unibas.ch, banzhaf@cs.mun.ca

Abstract

Catalytic Search is an optimization algorithm inspired by random catalytic reaction networks and their pre-evolutionary dynamics. It runs within an Artificial Chemistry in which reactions can be reversible, and replication is not taken for granted. In previous work one of us had shown that although inherently slower than Evolutionary Algorithms, Catalytic Search is able to solve simple problems while naturally maintaining diversity in the population. This is a useful property when the environment may change.

In this paper, we compare the performance of Catalytic Search and a Genetic Algorithm in a dynamic environment represented by a periodically changing objective function. We investigate the impact of parameters such as temperature, inflow/outflow rate, and amount of enzymes. We show that Catalytic Search is generally more stable in the face of changes, although still slower in achieving the absolute best fitness. Our results also offer some indications on how catalytic search could either degenerate into random search, or progress towards evolutionary search, although the latter transition has not been fully demonstrated yet.

Introduction

Artificial chemistries have been used to understand the origin of evolution from a pre-evolutionary, random initial state (Fontana and Buss (1994); Dittrich and Banzhaf (1998)), to devise bottom-up chemical computing algorithms for emergent computation (Banzhaf et al. (1996); Dittrich (2005)), and to build new optimization algorithms (Banzhaf (1990); Kanada (1995); Weeks and Stepney (2005)), among other usages. The motivation for the present work lies at the intersection of these three application domains. We are interested in exploring the emergent computation properties of artificial chemistries for the construction of beamed search schemes able to optimize solutions to user-defined problems. Instead of a top-down, pre-designed optimization algorithm, optimization could be regarded as a computation task to emerge from the bottom up, as an outcome of molecule interactions. In this context, it is worth determining the conditions for the emergence of optimization, of which evolution is only one example.

Bagley and Farmer (1991) showed that primitive metabolisms called *autocatalytic metabolisms* can emerge in an artificial chemistry where polymers undergo reversible polymerization reactions. One of the conditions for the emergence of such metabolisms is to drive the system out of equilibrium by a constant inflow of molecules from the food set, accompanied by a non-selective dilution flow. In this case, some reactions may be boosted by *catalytic focusing*: starting from a random soup of molecules, the system ends up focusing most of its activity and mass into a few types of molecules, self-organizing into autocatalytic reaction networks that consume food molecules to produce longer polymers. The molecules taking part in this autocatalytic core can be regarded as primitive metabolisms.

In previous work, Yamamoto (2010) proposed *catalytic search*, an optimization scheme inspired by catalytic focusing. Catalytic search is based on a pre-evolutionary chemistry (Nowak and Ohtsuki (2008)), where reactions might be reversible, and replication is not taken for granted. The reaction energy functions are assigned such that reactions towards fitter products are favored. The selective pressure in catalytic search comes from the differences in reaction rates for different molecules in the reactor. These differences can be amplified selectively by enzymes: some reactions can be accelerated by enzymes that decrease the activation energy barrier necessary for them to occur. Due to the absence of direct replication, he performance of such scheme lies between that of a random search, and that of an evolutionary algorithm. In spite of such apparent weakness, catalytic search and related chemical schemes have many interesting properties, as pointed out by Weeks and Stepney (2005): the potential to undo wrong computations or to decompose bad solutions through reversible reactions; the ability to steer the reaction flow towards the production of good products by shifting the equilibrium distribution of molecules; a certain robustness to noisy fitness feedback; and the prevention of premature convergence through a natural tendency to generate and maintain diversity in the population. This paper focuses on the latter property.

Catalytic Search

In this section we summarize the catalytic search algorithm by Yamamoto (2010), and introduce our own modifications: an improvement of the original enzyme matching scheme, and its adaptation to run continuously in dynamic environments.

Catalytic search works as follows: initially, a random soup of monomers (letters from an alphabet Σ) is generated. These monomers later concatenate into polymers (strings of symbols from Σ). Each polymer represents a candidate solution to the problem to be solved. At every time step, two molecules (monomers or polymers) are chosen for collision. They react with a probability k, which is also the *kinetic coefficient* of the reaction. If they react, a *crossover* of the two molecules is produced, and the two resulting molecules are injected into the soup. The educts are consumed in the process. The collision is elastic with probability $(1 - k)$.

A crossover reaction can be written as follows:

$$A + B \underset{k_r}{\overset{k_f}{\rightleftharpoons}} C + D \tag{1}$$

where A, B, C and D are strings from an alphabet Σ, k_f is the coefficient of the forward reaction, and k_r is the coefficient of the reverse reaction. An example for strings from $\Sigma = \{a, b, c, d\}$ is:

$$abdba + ccbdd \underset{k_r}{\overset{k_f}{\rightleftharpoons}} abdbdd + ccba \tag{2}$$

Crossover is a mass-conserving operation, i.e. it conserves the total number of symbols before and after the reaction. Concatenation occurs as a special case of crossover where the crossover points are the beginning and end of each string, respectively.

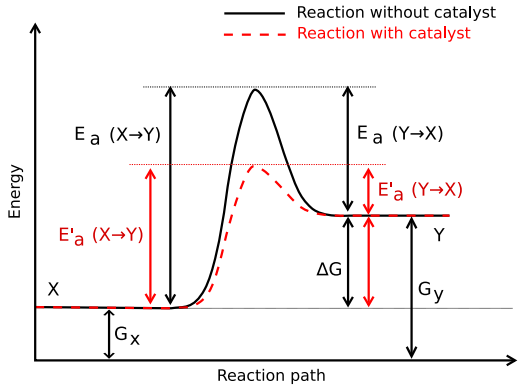


Figure 1: Potential energy changes during catalysed and uncatalyzed chemical reactions. From Yamamoto (2010).

Once the molecules have collided, the reaction only occurs if the molecules have sufficient kinetic energy in order to overcome the *activation energy barrier* (E_a) needed for the reaction. A *catalyst* is a substance that participates in a chemical reaction by accelerating it without being consumed in the process. Its effect is to lower the reaction's activation energy peak, thereby accelerating the reaction, while leaving the initial and final states unchanged. This acceleration comes from the fact that the coefficient k decreases exponentially with the activation energy, following the *Arrhenius equation* from chemistry:

$$k = Ae^{-\frac{E_a}{RT}} \tag{3}$$

where A is the so-called *pre-exponential factor* of the reaction, E_a is its *activation energy*, T is the absolute temperature, and R is a constant.

Figure 1 shows the energy diagram for a typical reversible reaction, where the effect of catalysis is highlighted with a red dotted line. The difference in potential energy before and after the reaction is given by ΔG:

$$\Delta G = G_p - G_e \tag{4}$$

where G_e is the potential energy of the educts, and G_p that of the products. In Figure 1, $G_e = G_X$, $G_p = G_Y$, and $\Delta G > 0$ if the reaction moves from left to right (i.e. in the direction from X to Y, the forward reaction); in the direction of the reverse reaction (from Y to X), we have $G_e = G_Y$, $G_p = G_X$, and $\Delta G < 0$. In this figure, the reverse direction is favored since it leads to more stable products (i.e. $\Delta G < 0$), while the forward direction is unfavored ($\Delta G > 0$). The reverse direction sees a lower activation energy than the forward direction ($E_a(Y \rightarrow X) < E_a(X \rightarrow Y)$) therefore it will be faster on average. Catalysis further reduces this barrier, accelerating the reaction in both directions ($E'_a(Y \rightarrow X) < E_a(Y \rightarrow X)$ and $E'_a(X \rightarrow Y) < E_a(X \rightarrow Y)$).

In order to steer the system towards the production of fitter solutions, in catalytic search the potential energy of a molecule is mapped to the fitness of the solution that it represents. The fitness function must be designed such that lower values indicate better fitness, for instance, a shorter distance to the optimum, or a lower cost of the solution. The educt and product energies are calculated as the sum of the fitness of the molecules involved:

$$G_e = f(A) + f(B) \tag{5}$$

$$G_p = f(C) + f(D) \tag{6}$$

where $f(i)$ is the fitness of solution i. In this way, fitter solutions have a lower potential energy and are therefore more stable. The production of fitter solutions (i.e. with lower potential energy) is favored ($\Delta G < 0$), whereas the production of poorer solutions is unfavored ($\Delta G > 0$), which is the desired effect.

The activation energy for a reaction is further mapped to the estimated computation cost of producing a solution: solutions that are more difficult to compute must overcome a

higher energy barrier, and therefore will be less likely to occur. This leads to a form of double-objective optimization scheme that seeks to optimize the fitness of the solution as well as the efficiency of its computation; these two objectives can be balanced via a proper choice of energy functions.

An increase in activation energy ΔE_a corresponding to the cost of the operation is then added on top of the highest potential energy G. ΔE_a corresponds to the portion $E_a(Y \rightarrow X)$ in Figure 1.

The activation energies of the forward and reverse reactions, E_{af} and E_{ar} respectively, are:

$$\text{if} \quad \Delta G \leq 0 \quad \begin{cases} E_{af} = \Delta E_a \\ E_{ar} = \Delta E_a - \Delta G \end{cases} \tag{7}$$

$$\text{if} \quad \Delta G > 0 \quad \begin{cases} E_{af} = \Delta E_a + \Delta G \\ E_{ar} = \Delta E_a \end{cases} \tag{8}$$

The coefficients k_f and k_r follow a simplified form of the Arrhenius equation:

$$k_f = e^{-\alpha E_{af}/T} \tag{9}$$

$$k_r = e^{-\alpha E_{ar}/T} \tag{10}$$

where α is a configuration parameter of the algorithm (currently set to $\alpha = 1$), and T is the temperature of the reactor.

This scheme is able to steer the flow of production of candidate solutions towards better ones, without explicit replication, and without an explicit memory of which molecules produced good solutions. The search process is guided by the differences in reaction rates to move from one pair of candidate solutions to another.

Enzymes

The energy-based reaction steering scheme described above is further complemented with an enzymatic step: reactions may be catalysed by enzymes that decrease the needed activation energy. In nature, enzymes act on both forward and reverse sides of the reaction, therefore the equilibrium concentrations do not change. In contrast, the enzymes used in catalytic search only facilitate the forward reaction in the direction of fitter products.

Enzymes are kept in a separate pool. When two molecules collide, if the reaction results in $\Delta G < 0$, i.e. in better fit products, then an enzyme might be created for this reaction, with a probability p_c proportional to the amount of improvement $|\Delta G|$. The next time similar molecules collide, the enzyme can be used to facilitate their reaction, by lowering the corresponding ΔE_a.

In the original catalytic search scheme only exact match between enzyme and substrates was supported. In this paper, we extend the matching scheme such that enzymes bind to their substrates with a certain affinity, proportional to how well their strings match. With this scheme, an enzyme may accelerate similar reactions, and a reaction may benefit from the combined catalytic effect of similar enzymes. For this purpose, we have modified the format of the enzymes in the original catalytic search scheme in order to take into account the strength of matching between enzyme and substrates. In our scheme, enzymes are built from chemical reactions as follows. A generic crossover reaction between two educt strings s_1 and s_2 can be written as:

$$s_{1a}s_{1b} + s_{2a}s_{2b} \rightarrow s_{1a}s_{2b} + s_{2a}s_{1b} \tag{11}$$

where s_{ij} are the substrings in s_i separated by the chosen crossover points. An enzyme for this reaction is a string of the form: "$s_{1a}|s_{1b}|s_{2a}|s_{2b}$", with the vertical bar "|" indicating the crossover points. The enzyme uniquely identifies the reaction, and can therefore be used to represent it in molecular form, constituting a memory of past successful reactions.

We use the similarity between the enzyme and the concatenated substrates as the affinity metric. The similarity is the number of matching positions in the alignment between the two strings. For the example of Reaction (2), the corresponding perfectly matching enzyme is "$abd|ba|cc|bdd$". If another reaction between similar strings with similar crossover point happens, say, one described by enzyme "$abb|a|cc|bd$", then the similarity between the two corresponding enzymes is high (10 over a maximum of 11 in this example), leading to a higher catalytic enhancement. The similarity is further normalized by the length of the smallest of the two strings, such that shorter polymers also have a chance to get catalysis. More exactly, the binding strength function between two strings s_1 and s_2 is defined as $bind(s_1, s_2) = similarity(s_1, s_2)/min(length(s_1), length(s_2))$.

Once two molecules collide and their crossover points are decided, a small number of enzymes (subset B) are drawn at random from the enzyme pool, and their matching strengths are calculated with respect to the perfect enzyme c for the reaction. The contributions of all enzymes are added up together: $s_c = \sum_{b \in B} bind(b, c)$. The sum of the strengths is then used to calculate the reduction in activation energy contributed by the enzymes. If $s_c \geq 1$, the new activation energy becomes:

$$\Delta E'_a = \frac{\Delta E_a}{s_c} \tag{12}$$

else ΔE_a remains unchanged.

In order to make sure that the enzyme pool is periodically refreshed and does not grow unbounded, enzymes are subject to a non-selective dilution flow beyond the maximum capacity of the enzyme pool, C_{max}.

We have further modified the algorithm to run continuously, not stopping when a solution is found, in order to run it in dynamic environments. The updated algorithm is shown in Algorithm 1.

Algorithm 1 Catalytic Search Algorithm

S: multiset of candidate solutions
C: pool of enzymes (catalysts)
C_{max}: maximum capacity of C
initialization:
$S =$ random soup of N monomers $m \in \Sigma$
$C = \emptyset$
while $true$ **do**
  expel two random molecules e_1 and e_2 out of S
  $(i_1, i_2) =$ random crossover points within e_1 and e_2
  $(p_1, p_2) \leftarrow$ crossover(e_1, e_2, i_1, i_2)
  $G_e =$ fitness$(e_1)+$ fitness(e_2)
  $G_p =$ fitness$(p_1)+$ fitness(p_2)
  $\Delta G = G_p - G_e$
  $E_a = (|e_1| + |e_2|)/2$
  if $\Delta G > 0$ **then**
    $E_a \leftarrow E_a + \Delta G$
  else if $\Delta G < 0$ **then**
    $c =$ enzyme(e_1, e_2, i_1, i_2)
    $B \leftarrow$ draw n_c enzymes from C
    $s_c = \sum_{b \in B}$ bind(b, c)
    if $s_c \geq 1$ **then**
      $E_a \leftarrow E_a / s_c$
    end if
    $p_c = |\Delta G| / G_e$
    add another instance of c to C with probability p_c
    while $|C| > C_{max}$ **do**
      destroy a random catalyst from C
    end while
  end if
  $k_f = e^{-\alpha E_a / T}$
  if dice(k_f) **then**
    inject new products p_1 and p_2 into S
  else
    inject educts e_1 and e_2 back to S
  end if
end while

Catalytic search steers the flow of chemical reactions by acting primarily on the rate coefficients rather than on the concentrations. Therefore it has a natural tendency to keep a diversity of molecules in the reactor, some of which are rarely used because of a slow reaction speed, but nevertheless stay present at some concentration. These molecules could become useful in the future, for instance when the environment changes. This provides a simple way to keep a pool of alternative solutions in the population, and to switch to different solutions by preferentially choosing different reaction pathways to construct alternative solutions using the elements in the pool. In this paper we perform experiments to support this claim.

Genetic Algorithm in a Chemistry

For comparison purposes, a Genetic Algorithm (GA) is implemented within a similar artificial chemistry. This GA was briefly introduced in (Yamamoto (2010)). Here we describe it in more detail for completeness. It is a variation of a Steady-State Genetic Algorithm (SSGA) based on tournament selection. SSGA is a non-generational evolutionary algorithm in which at each time step, individuals are selected for evaluation and reproduction, without a synchronized generational loop (see Lozano et al. (2008) for a survey).

The initial population in the "chemical GA" is also a collection of monomers, as in catalytic search. At every iteration, r individuals (the tournament size) are chosen at random and placed in a "catalyst pocket" C. The two best individuals (winners of the tournament) produce $r-2$ children by crossover and mutation. These children replaced the other $r-2$ individuals who had lost the tournament. The full algorithm is shown in Algorithm 2.

Note that in contrast with catalytic search, the GA is not mass-conserving: the new individuals might have completely different sizes from those they replaced. This is done in order to keep the chemical version of the GA as close as possible to a normal GA.

Algorithm 2 Steady State Genetic Algorithm in a Chemistry

S: multiset of candidate solutions
r: tournament size
p_c: crossover probability
p_m: mutation probability
initialization:
$S =$ random soup of N monomers $m \in \Sigma$
while $true$ **do**
  C: set of tournament members
  expel r random molecules out of S and inject them into C
  expel the two fittest molecules e_1 and e_2 out of C
  for $i = 1$ to $r/2 - 1$ **do**
    if dice(p_c) **then**
      $(p_1, p_2) \leftarrow$ crossover(e_1, e_2)
    else
      $p_1 = e_1, p_2 = e_2$
      if dice(p_m) **then**
        $p_1 \leftarrow$ mutate(p_1)
      end if
      if dice(p_m) **then**
        $p_2 \leftarrow$ mutate(p_2)
      end if
    end if
    inject p_1 and p_2 into S
  end for
end while

Experiments

Yamamoto (2010) compared catalytic search, GA and a random search to solve instances of the OneMax problem, extended to arbitrary target strings from a given alphabet Σ. The OneMax problem consists in maximizing the number of ones in a binary string, which is a special case of finding a hidden sentence $s \in \Sigma+$, made of a sequence of letters from Σ. This problem is known to be very easy to optimize, facilitating the comparison of the algorithms under ideal situations.

Yamamoto (2010) had already shown that catalytic search is able to solve simple problems, but in a slower manner than a GA. She had also shown that while catalytic search moves steadily towards the goal, a purely random search not only does not find the optimum but also diverges.

In this paper we focus on comparing catalytic search and GA under a changing environment, simulated by a target objective that is periodically modified. Furthermore, we investigate the influence of several parameters on the behavior of catalytic search, namely, the size of the enzyme pool, the amount of inflow/outflow, and the temperature. Two instances of the hidden sentence problem are used: one with binary strings with a target of all ones (OneMax), and another with an alphabetic sentence. They are shown in Table 1, where "id" is the identifier of the instance (subsequently labeled as "case 1" and "case 2" on the plots), and s_s is the size of the search space for each instance, when considering only sentences of length up to $|s|$.

id	Σ	$\|\Sigma\|$	target sentence s	$\|s\|$	s_s
1	01	2	1111111111111111	16	131070
2	a-z	26	catalyticsearch	15	1.744e+21

Table 1: Problem instances used

The ΔE_a cost function is set to the average length of the reacting strings, as in (Yamamoto (2010)). Fixed parameters set to default values are shown in Table 2.

size of the initial population of monomers	$N_0 = 100$
number of enzymes drawn from the enzyme pool for each catalysed reaction	$\|B\| = 10$
GA tournament size	$r = 4$

Table 2: Fixed parameter values

Results

We measure the obtained fitness and the ability to maintain diversity in the presence of changes. For catalytic search, we investigate the impact of the amount of inflow/outflow, the temperature and the size of the enzyme pool. Diversity is measured using a multiset diversity metric (Mattiussi et al. (2004)). It measures the fraction of unique elements (molecules) over the total size of the multiset (population size).

The target string changes 3 times during a run, at $t = 25, 50, 75$ (in units of 100 iterations). The target string is modified simply by applying the same mutation operator used in GA, with a given mutation probability per symbol of μ_t. All the results shown reflect the average of 10 runs.

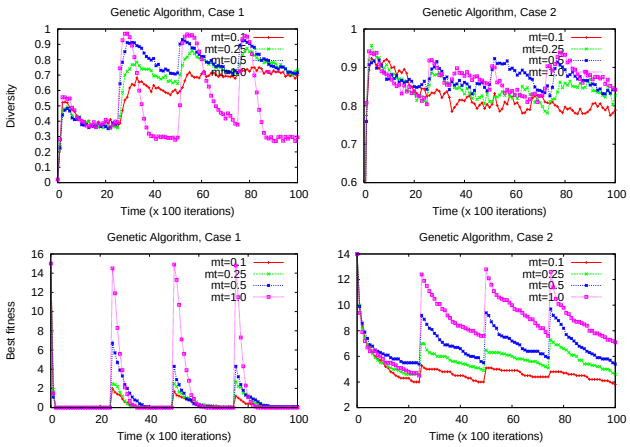


Figure 2: Average diversity and average best fitness for the genetic algorithm with changing target strings.

First of all, we compare GA and catalytic search for target mutation values μ_t varying from 0.1 to 1.0, representing slight to severe environmental changes.

Figure 2 shows the behavior of the GA under this scenario. As expected, bigger changes (represented by a higher μ_t) disturb the optimization process to a greater extent. For case 1, the amount of worsening in fitness corresponds roughly to the amount of mutation added. For example, for $\mu_t = 1.0$ (the target string changes entirely) the search restarts from scratch, with the best fitness jumping to 100% of its initial value at $t = 0$. For $\mu_t = 0.1$ (the target string changes slightly) the best fitness jumps to around 10% of its initial value, and so on. For case 2, the fitness also presents the characteristic sawtooth, but the recovery after changes is slower due to the higher difficulty of the problem.

The diversity of the population in GA displays a curious behavior under higher target mutation values. This is especially visible on case 1: soon after the target changes, the diversity jumps nearly to the maximum, and then decreases as the system approaches the optimum. The latter decrease in diversity is a well-known phenomenon in evolutionary computation, however the spontaneous jumps seem more surprising.

Figure 3 (left) shows the behavior of catalytic search under the same situation, for the case of no catalysis (empty enzyme pool), no inflow/outflow, and temperature $T = 1$. Naturally, the GA is much faster than the catalytic search at finding the optimum, which is an expected outcome. A

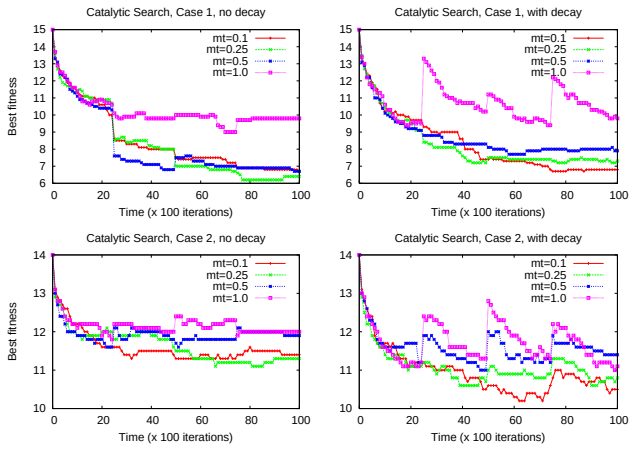


Figure 3: Average best fitness for catalytic search, with/without inflow/outflow.

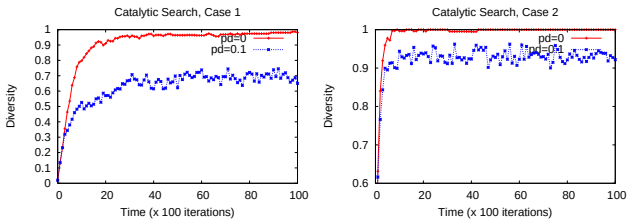


Figure 4: Average diversity for catalytic search, with and without inflow/outflow.

more surprising result is that the behavior of catalytic search is qualitatively distinct from the GA: a small amount of target mutation does not seem to affect the system so clearly as it does for GA: sometimes, it even seems to help the search, such as around $t = 25$ for case 1 and $\mu_t \leq 0.5$.

Figure 3 (right) shows what happens when we introduce a small amount of inflow/outflow. This is represented by a decay parameter $p_d = 0.1$, meaning that at every iteration, with probability p_d, a negative tournament with size r is executed: $r = 4$ individuals are extracted at random from the population; their fitness is evaluated, and the one with the worst fitness (the loser of the tournament) is destroyed. It is then replaced by its length in new randomly generated monomers. In this way we ensure a mass-conserving inflow/outflow mechanism that combined with a negative selection mechanism makes sure that worse individuals are replaced with a higher probability. Here two types of behavior can be distinguished:

- for high target changes ($\mu_t \geq 0.5$) the behavior is qualitatively different from that with no inflow: it looks more like a GA (the fitness jumps when the target changes) although quantitatively (in terms of absolute fitness values) it still cannot optimize as fast as GA.
- for low target changes ($\mu_t \leq 0.25$) the behavior looks like the catalytic search with no inflow/outflow.

Increasing p_d does not seem to help: it floods the system with new monomers that cannot be consumed on time, and also causes the search to become more random.

Figure 4 compares the diversity of the population for catalytic search with and without inflow/outflow, for both cases studied. In contrast to the GA, the diversity in catalytic search is unaffected by the mutation of the target string. All mutation values produced similar figures, so we chose to plot only the results for $\mu_t = 0.5$.

At the beginning, the population is made entirely of monomers, therefore the diversity is at most $|\Sigma|/N$, i.e. 0.02 for case 1 and 0.26 for case 2, for $N = 100$. It then increases progressively as new solutions are built by concatenating monomers. The fact that the diversity is close to the maximum for the case of no inflow/outflow ($p_d = 0$ on Fig. 4) means that in this situation, every individual in the population is nearly unique; there is no visible catalytic effect fostering the production of selected individuals.

For the case with inflow/outflow ($p_d = 0.1$ on Fig. 4) a lower diversity is observed. This is explained by the constant inflow of new monomers: since the size of the alphabet is small compared to the population, the monomer population necessarily contains a lot of copies of the same molecule. This is more evident for case 1, which uses a binary alphabet. There, the inflow causes the diversity to decrease much more prominently than in case 2.

Catalysis is expected to decrease diversity, by focusing the mass of the system into fewer species when the system is out of equilibrium. This phenomenon has not been observed in our system: the plots for $C_{max} = 100$ and $C_{max} = 1000$ closely resemble Fig. 4. This result indicates that the way catalysis is implemented in this system is not sufficient to modify the concentration pattern significantly when out of equilibrium, and focus most of the mass of the system into fewer, selected species. Catalysis does have a moderate effect on the performance, as will be shown in Figures 5 and 6. However, this effect is probably achieved primarily by accelerating a few reactions selectively by increasing their kinetic coefficients, and not by a significant concentration change. Even if faster, the enzymatic reactions do not succeed to focus sufficient mass, since the amount of possible reactions is not restricted: random crossover points are chosen at every time step, leading to different outcomes. This issue deserves further investigation. Actually, it is not straightforward to design an artificial chemistry to exhibit the focusing effect reported by Bagley and Farmer (1991), and it is even more difficult to cause it to spontaneously produce autocatalytic networks, which could later lead to the emergence of a GA-like scheme. On the other hand, the fact that catalytic search is able to keep diversity under a wide variety of conditions is a good property worth exploring.

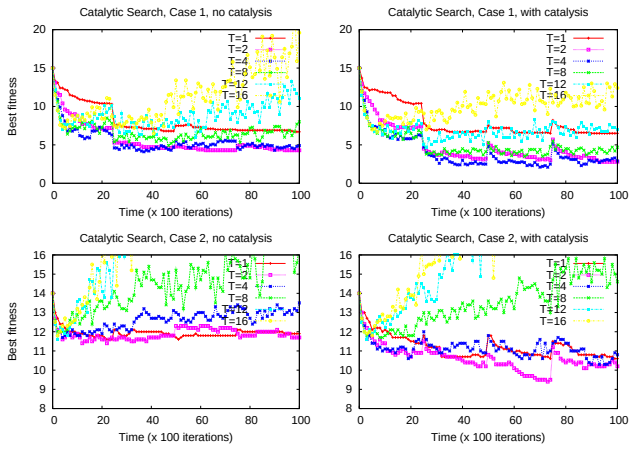


Figure 5: Influence of temperature and catalysis, no inflow/outflow

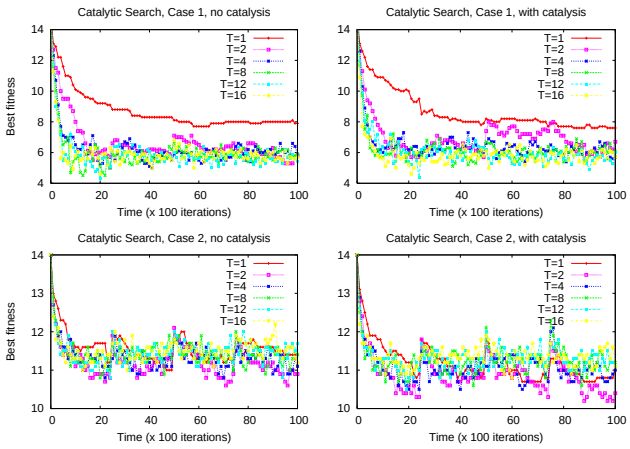


Figure 6: Influence of temperature and catalysis, with inflow/outflow

We now look at the influence of the temperature and of the amount of enzymes in the catalyst pool. We take $\mu_t = 0.5$ as an example (other values of μ_t produced similar results). The temperature makes all reactions faster, non-selectively, while the enzymes selectively speed up a few matching reactions. Figure 5 compares the best fitness of catalytic search for varying temperatures, with and without explicit catalysis, and no inflow/outflow ($p_d = 0$). We first look at the results without catalysis (left side). For case 1, increasing the temperature to moderate values improves the search: the optimum temperature is around $2 \leq T \leq 4$. For case 2, increasing the temperature does not seem to help: the best fitness does not improve. This can be explained by the fact that the energy barrier for case 1 might be too high, excessively penalizing the longer solutions necessary to solve this problem. Case 2 suffers from the same problem, but has a much larger search space, so merely increasing the temperature, a global parameter affecting all individuals, is not sufficient to improve the search.

Very high temperatures (for example, $T \geq 12$ for case 1, $T \geq 8$ for case 2, Fig. 5 (left), without catalysis) introduce more noise in the system, which becomes closer to a random search and hence tends to diverge.

Figure 5 (right) shows the effect of catalysts, for a total capacity of the catalyst pool set to $C_{max} = 1000$ enzymes. Catalysts help to improve the search and sometimes also help to stabilize the system: for lower temperatures, the system with catalysts moves faster towards the optimum; for higher temperatures, sometimes the catalysts prevent the search from becoming random, as for $T = 12$ in case 1.

When combining catalysis with inflow/outflow ($C_{max} = 1000$ and $p_d = 0.1$) the effect of catalysis becomes barely noticeable (Figure 6). This could be due to the fact that individuals that could be recognized by the enzymes are then selected for destruction, while new individuals for which there are no ready-made catalysts are created at a higher rate. Figure 6 also shows that the temperature has little impact on the performance (except for case 1 for $T = 1$ vs. other values of T). More importantly, the system with inflow/outflow no longer tends to diverge to a random search when the temperature increases, which is a positive aspect. The sawtooth pattern reminding us of GA appears here again, as in Figure 3 (right).

Related Work

This work was inspired mainly by Bagley and Farmer (1991), Banzhaf (1990), Kanada (1995), and Weeks and Stepney (2005).

Farmer et al. (1986) identify a critical probability of catalysis, near which the spontaneous emergence of self-sustaining autocatalytic networks becomes highly probable. Bagley and Farmer (1991) then show the spontaneous emergence of autocatalytic metabolisms, together with further conditions for their emergence. However, their results were based on a random assignment of catalytic efficiencies. Methods still lack for designing a proper structure-to-function mapping in a string-based chemistry, that would lead to a critical catalysis probability in the range needed for such emergent phenomenon to occur and persist. Hintze and Adami (2008) showed the evolution of metabolisms using a string-based chemistry with binding affinity and specificity. However, their design already assumes a whole cell structure with interacting genes and proteins.

Suzuki et al. (2003) enumerate minimal conditions for the evolution of artificial life forms, however they do so in a qualitative way. The quantitative conditions for the emergence of life subsystems (including metabolism) in an artificial environment are still not entirely understood, and methods for designing emergent algorithms based on these principles are still lacking. Designing algorithms inspired by such thin border between life and inanimate chemistry could help to understand such conditions and to devise correspond-

ing methods in an iterative way.

The Molecular Travelling Salesman by Banzhaf (1990) is an optimization algorithm based on an artificial chemistry in which molecules representing candidate solutions are processed by machines that float in the reactor. These machines perform variation and selection, and are therefore closer to our version of GA in a chemistry.

In the Chemical Casting Model (CCM) by Kanada (1995), reaction rules modify and select molecules (candidate solutions) such as to drive the system towards a more ordered state (with lower entropy) in which molecules encode better solutions. The fitness mapping in CCM is similar to catalytic search: CCM seeks to maximize order by minimizing entropy (which is a macroscopic quantity), whereas catalytic search seeks to improve the fitness by moving towards lower energy levels at the microscopic level.

In the Artificial Catalysed Reaction Networks by Weeks and Stepney (2005), molecules encode partial solutions that are constructed via reversible polymerization reactions. Fitter products are rewarded by catalyzing their own production, therefore each molecule is potentially an autocatalyst, in contrast to our work where autocatalysis is not assumed.

A lot of work has been done on improving evolutionary computation for dynamic environments (see Jin and Branke (2005)). However, the potential of pre-evolutionary schemes in such context remains to be explored.

Summary and Conclusions

Our results reveal interesting aspects and point to many issues to be investigated. First of all, the behavior of catalytic search in the presence of changes is qualitatively different from that of an evolutionary algorithm. Evolution is capable of fast optimization, but is also more severely affected by changes. Catalytic search, on the other hand, is slower but also less sensitive to changes, and able to maintain a diverse pool of individuals in the population.

The behavior of catalytic search can be steered by parameters: a higher temperature, for instance, can cause the system to degenerate into a random search. Such degradation can be slowed down by the presence of catalysts, which have a stabilizing effect provided that the amount of inflow/outflow is very small or none.

Perhaps the most interesting phenomenon that could be expected from such a system would be a spontaneous transition to an autocatalytic or collectively autocatalytic stage, which could become a bridge towards a further transition to an evolutionary stage. So far however, we were not able to demonstrate such transitions in an emergent way. One of the major improvements needed in the current system is to ensure a larger impact of catalysts, in order to exhibit the focusing phenomenon that could enable such transitions to occur spontaneously. This would require a carefully designed structure-to-function mapping reflecting the required catalysis probabilities. It would also require a more efficient stochastic collision algorithm able to take into account a large number of possible reactions with rates differing by several orders of magnitude. Another major improvement needed is to make the system more tolerant to a continuous inflow/outflow, which is one of the primary conditions necessary for catalytic focusing to succeed.

References

Bagley, R. J. and Farmer, J. (1991). Spontaneous Emergence of a Metabolism. In *Artificial Life II*, pages 93–140. Addison-Wesley.

Banzhaf, W. (1990). The "Molecular" Traveling Salesman. *Biological Cybernetics*, 64:7–14.

Banzhaf, W., Dittrich, P., and Rauhe, H. (1996). Emergent Computation by Catalytic Reactions. *Nanotechnology*, 7:307–314.

Dittrich, P. (2005). Chemical Computing. In *Unconventional Programming Paradigms (UPP 2004), Springer LNCS 3566*, pages 19–32.

Dittrich, P. and Banzhaf, W. (1998). Self-Evolution in a Constructive Binary String System. *Artificial Life*, 4:203–220.

Farmer, J. D., Kauffman, S. A., and Packard, N. H. (1986). Autocatalytic replication of polymers. *Physica D*, 2(1-3):50–67.

Fontana, W. and Buss, L. W. (1994). 'The Arrival of the Fittest': Toward a Theory of Biological Organization. *Bulletin of Mathematical Biology*, 56:1–64.

Hintze, A. and Adami, C. (2008). Evolution of complex modular biological networks. *PLoS Comput Biol*, 4(2):e23.

Jin, Y. and Branke, J. (2005). Evolutionary Optimization in Uncertain Environments - A Survey. *IEEE Transactions on Evolutionary Computation*, 9(3):303–317.

Kanada, Y. (1995). Combinatorial Problem Solving Using Randomized Dynamic Composition of Production Rules. In *IEEE International Conference on Evolutionary Computation*, pages 467–472.

Lozano, M., Herrera, F., and Cano, J. R. (2008). Replacement Strategies to Preserve Useful Diversity in Steady-State Genetic Algorithms. *Information Sciences*, 178(23):4421–4433.

Mattiussi, C., Waibel, M., and Floreano, D. (2004). Measures of Diversity for Populations and Distances Between Individuals with Highly Reorganizable Genomes. *Evolutionary Computation*, 12(4):495–515.

Nowak, M. A. and Ohtsuki, H. (2008). Prevolutionary Dynamics and the Origin of Evolution. *PNAS*, 105(39).

Suzuki, H., Ono, N., and Yuta, K. (2003). Several necessary conditions for the evolution of complex forms of lif e in an artificial environment. *Artificial Life*, 9(2):153–174.

Weeks, A. and Stepney, S. (2005). Artificial Catalysed Reaction Networks for Search. In *ECAL Workshop on Artificial Chemistry*.

Yamamoto, L. (2010). Evaluation of a Catalytic Search Algorithm. In *Proc. 4th International Workshop on Nature Inspired Cooperative Strategies for Optimization (NICSO)*, Granada, Spain.

Rock-Paper-Scissors Dynamics in a Digital Ecology

Philip Gerlee[1] and Torbjörn Lundh[2]

[1]Center for Models of Life, Niels Bohr Institute, Blegdamsvej 17, 2100, Copenhagen, Denmark, gerlee@nbi.dk
[2]Mathematical Sciences, Chalmers University of Technology and
University of Gothenburg, SE-41296, Göteborg, Sweden, torbjorn.lundh@chalmers.se

Abstract

In this paper we present an Alife-platform named *Urdar* aimed at investigating dynamics of ecosystems where species engage in cross-feeding, i.e. where metabolites are passed from one species to the next in a process of sequential degradation. This type of interactions are commonly found in microbial ecosystems such as bacterial consortia degrading complex compounds. We have studied this phenomenon from an abstract point of view by considering artificial organisms which metabolise binary strings from a shared environment. The organisms are represented as simple cellular automaton rules and the analogue of energy in the system is an approximation of the Shannon entropy of the binary strings. Only organisms which increase the entropy of the transformed strings are allowed to replicate. We find that the system exhibits a large degree of biodiversity and a non-stationary species distribution, especially during low rates of energy inflow, and that the time spent in each species configuration exhibits a broad distribution. Investigating the interaction between different species in the system by invasion experiments we observe that co-existence is a common feature and that some triplets of species exhibit intransitive, i.e. rock-paper-scissors like, interactions.

Introduction

The origin and maintenance of biodiversity has been a long standing question among ecologists (Hutchinson, 1959). One of the simplest ecological system where biodiversity emerges, and is stably maintained, is in populations of *E. coli* growing in a homogeneous environment limited by a single resource, usually glucose. The diversity is facilitated by cross-feeding (syntrophy), where one strain partially degrades the limiting resource into a secondary metabolite which is then utilised by a second strain. This phenomenon was first observed by Helling et al. (1987) and has since been reported to occur in other systems such as methanogenic environments (Stams, 1994), bacteria engaging in nitrification (Costa et al., 2006) and degradation of xenobiotic compunds (Dejonghe et al., 2003; Katsuyama et al., 2009).

The evolution of cross-feeding has been investigated by Pfeiffer and Bonhoeffer (2004) using a theoretical model, and their results showed that cross-feeding naturally emerges under the assumption that ATP production is maximised while the total concentrations of enzymes and intermediates are minimised. Further they showed that the evolution of cross-feeding depends on the dilution rate in the chemostat, and that a stable polymorphism is more likely to emerge at low dilution rates.

A different approach was taken by Doebeli (2002) who investigated the emergence of cross-feeding in the framework of adaptive dynamics. In this case the conditions for evolutionary branching and the appearance of cross-feeding are that there is a trade-off between uptake efficiency of the primary and seconday metabolites, and that this trade-off function has a positive curvature. The model also makes the correct prediction that cross-feeding is less likely to occur in serial batch culture, in which the primary resource is not replenished (Rozen and Lenski, 2000). This highlights the necessity of the secondary metabolite being present for an extended period of time for cross-feeding to evolve.

In this study we present a recent Alife-platform (Gerlee and Lundh, 2010) aimed at investigating the evolution of cross-feeding, but not in the context of a specific biological system, but instead we extract and analyse the general principles governing systems where cross-feeding might emerge. In its abstract nature the model will be more akin to an artificial chemistry (Dittrich et al., 2001), but with the difference that we make a distinction between the agents subject to an evolutionary process and resources which they consume for reproduction. The aim of this paper is to describe the new platform, present some new results, and discuss future investigations and possible extensions of the system.

The model

To explain the motivation behind the plaform *Urdar*, let us consider the following thought experiment: a population of different species of bacteria inhabit a petri dish continually supplied with a given nutrient. The bacteria only partially metabolise the nutrient, which is added at a certain rate, so other bacteria might extract energy from the "left-overs" of this successive degradation. Assume that this experiment is

carried out for a long period of time, so that species that do well will increase their share of the total population. Since we can imagine that different strains of bacteria have variations to their metabolism, we have that if a single species dominates the population, a certain type of left-overs will be abundant in the free pool of metabolites. Hence that would lead to higher number of offspring of a species that is specialised on extracting energy from that kind of left-overs.

Please note that the model we will present is not specific to bacteria, but could represent any ecosystem where resources are consecutively degraded by several species, creating a network of interdependence. We set up such an experiment using artificial organisms or agents that are capable of successive degradation (transformation) of metabolites from which they extract energy used for self-maintenance and reproduction.

In our model we will use binary strings as the "foodstuff", and we will view the metabolic process as the degradation of ordered strings into strings with a higher degree of disorder. More specifically, let R be a pool of resources (or metabolites) $\{\mathrm{r}_i\}$ where each r_i is a binary string of length L, as for example $r_i = 00101\ldots01110$. Let A be the population $\{a_j\}$ of agents (or organisms), where each agent a_j is represented by a function that transforms binary strings into new binary strings, $a_j : R \to R$. We can view this mapping as a "metabolic digestion" of the string being transformed. More precisely the agents in A transforms resource strings from R in the following way

$$r_i^{\text{new}} = a_j(r_i^{\text{old}}).$$

Let now a positive function E on the binary strings in R represent the "energy state" of such a string. If the agent a_j is able to extract energy from the resource string r_i, we have that $E(r_i^{\text{new}}) < E(r_i^{\text{old}})$, and the amount of energy extracted is given by

$$\Delta E_j = E(r_i^{\text{old}}) - E(r_i^{\text{new}}).$$

The evolutionary dynamics are then introduced by a possible replication of the agent a_j to a daughter agent whenever $\Delta E_j > 0$. Replication in the current model is asexual and offspring have just a single parent organism. The offspring is mutated with probability μ, and replaces another agent in the population, thus keeping the population size constant. The constant population size can be thought of as either being imposed by a space constraint, or by the carrying capacity of an additional nutrient required for biomass synthesis (assuming that the evolutionary dynamics related to this trait occurs on a much slower time-scale). The probability for a reproduction to take place is an increasing function of ΔE_j with zero probability if $\Delta E_j \leq 0$. Hence a successful type of agent, is one which is able to effectively extract energy from the binary resource strings in R, and the content of R in turn depends on which agents constitute the population. In order to feed the system with energy, strings in the resource pool R are continually being replaced with new high-energetic strings at a rate γ, representing a flow of energy into the system. A schematic of the modelling framework is shown in fig. 1, which illustrates how binary strings are metabolised by the organisms and flow through the system.

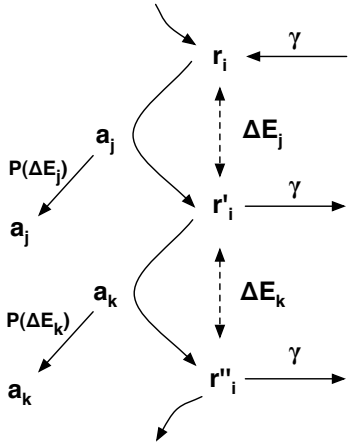


Figure 1: A schematic view of the model. The agents in the model digest binary strings by applying CA-rules, transforming r to r'. To each such metabolic step we can associate a difference in energy ΔE (visualised with dotted lines). The reproduction of each agent depends on how much it can decrease the energy of the binary string and occurs with probability $P(\Delta E)$ (represented by the arrows on the left hand side). The binary strings exist in a common pool which they enter (and leave) at a rate γ, as shown by the arrows on the right hand side.

The frame-work described so far is quite general, and we will in the following describe the particular choices we have made in the current study. Firstly, the agents a_j are chosen to be nearest-neighbour one-dimensional elementary cellular automata (CA), one of the simplest notions of digital algorithms. The reason for that particular choice in *Urdar* is that such functions are well studied in the literature starting from the work of Wolfram (1983). They are simple, but still shows a surprisingly wide range of complexity. The second choice we made was using an approximated Shannon Entropy as the energy function E, which gives an estimate of the amount of disorder a binary string contains (Shannon, 1948), associating a low entropy (low level of disorder) with a high "energy" state of the string, i.e. we set $E = 1 - s$. To motivate such a choice, one can see organismal metabolism as degradation of ordered structures into less ordered configurations. Entropy is a measure of such disorder. This

viewpoint is both common and well established:

> "Thus the device by which an agent maintains stationary at a fairly high level of orderliness (= fairly low level of entropy) really consists in continually sucking orderliness from its environment." (Schrödinger, 1944)

One could of course make use of a more sophisticated artificial chemistry by assigning higher energy, and hence fitness, if an organism is able to transform strings into certain patterns, instead of just increasing the entropy; but in our effort for simplicity and a more open-ended fitness function we have chosen the current set up.

Finally, the probability for agent a_j to reproduce, as a function of the energy it extracts from a binary string, is given by

$$P(\Delta E) = \begin{cases} \frac{1-\exp(-\Delta E/\beta)}{1-\exp(-\beta)}, & \text{if } \Delta E > 0 \\ 0, & \text{if } \Delta E \leq 0. \end{cases} \quad (1)$$

where β is a positive parameter indicating the level of competitive pressure among the agents. When β tends to zero, selection is weak as any $\Delta E > 0$ gives a probability of reproduction very close to unity, while for larger β selection is stronger as the magnitude of ΔE is more important for determining the value of $P(\Delta E)$ and hence the reproductive success of the organisms.

An example of applying CA-rules to binary strings is shown in fig. 2, where three rules, i.e. three different species, digest a string with a low entropy to binary strings with successively increasing entropy. This is the type of interactions we can expect in the model, in particular at low γ when the strings are replenished at a low rate. This figure also illustrates the fact that the CA-rules in general make small changes to the food string during digestion. In fact there is no CA-rule which can, in a single metabolic step, increase the entropy of a fairly ordered string to the maximum attainable entropy. This is similar to individual metabolic reactions in real organisms which generally only change the free energy of the metabolites a small amount, while the metabolism as a whole is responsible for the major difference in free energy between the nutrients taken up by the organism and the waste products being excreted. This fact also suggests that *Urdar* can be viewed as a model of the early stages of life on earth when the metabolic repertoire of organisms was much smaller and cross-feeding was possibly more prominent.

Note that in the current set up, the mapping between the genotype and phenotype of the agents is one-to-one, where the genotype corresponds to the integer value representing the rule (ranging from 0 to 255), and the phenotype simply is the action of the rule on the strings which are metabolised. All organisms implementing the same CA-rule are consequently referred to as belonging to the same species. In the current set up, we have chosen not to explicitly model self-replication in order to keep things simple. In future extensions of the model both sex and self-replication can be included.

The implementation of the model

To conclude the model description, let us sum up the main features of the model[1]. The dynamics, depicted schematically in fig. 1, in the model during one update can be described in the following way:

1. Each agent in the population picks randomly a resource string r_j from the well mixed resource pool R and transform it accordingly to its CA-rule and then puts the transformed string back into the resource pool.
2. The efficiency of the "metabolic process" just occurred is evaluated by measuring the energy difference ΔE of the string before and after the "digestion/transformation". This is done by drawing a random number x uniformly between 0 and 1, and if $P(\Delta E) > x$ the agent reproduces.
3. With probability μ the offspring will be mutated uniformly to another CA-rule.
4. In order to keep energy flowing into the system, after all agents have been updated, a fraction γ of the strings are replaced with high energy binary strings.

The replacement rate γ can be seen as a flow rate of energy into the system. If that rate is high, there will be less interaction through cross-feeding among the agents in A, as strings are flushed out at high rate, but if on the other hand γ is set to zero, the whole process will slow down to a halt, as only a finite amount of energy can be extracted from each food string. The strings introduced into the system are random binary strings, however with a low entropy (high degree of order). The new strings are constructed by at each position adding a 1 with probability p_0 and a 0 with the complementary probability $1-p_0$. The Shannon entropy of such strings is given by

$$s_0 = p_0 \log_2 \frac{1}{p_0} + (1-p_0) \log_2 \frac{1}{1-p_0}, \quad (2)$$

where $\log_2$ is the logarithm with base 2, i.e. $2^{\log_2 x} = x$. By setting p_0 close to unity we can create strings which, although being random, have a low entropy. In order not to bias the resource pool to strings which are dominated by ones, at an equal rate we add strings which have the probabilities reversed, i.e. are dominated by zeros instead.

The parameters

We here briefly recapitulate the main parameters of the system and their significance.

[1] An online version of the platform is available at: http://www.math.chalmers.se/∼torbjrn/Urdar/urdar.html

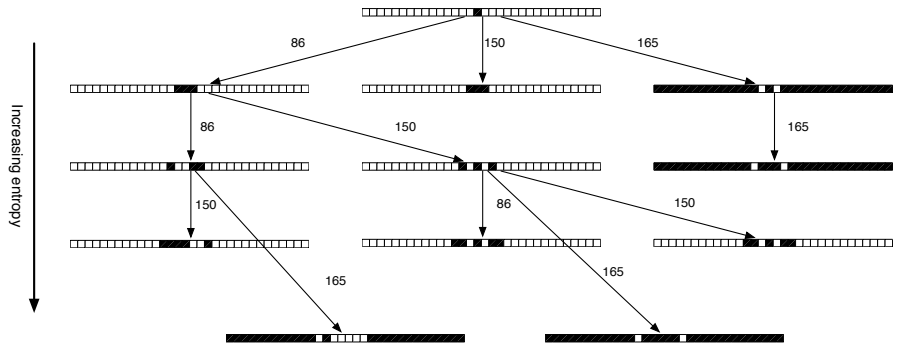


Figure 2: The transformation of binary food strings by three different species (i.e. CA-rules). Only transformations that increase the entropy are shown and they have been truncated at a metabolic depth of four. The number of possible transformations is greater for the three rules together than for a single isolated rule suggesting the possible advantage of cross-feeding among the species in the model.

γ is the inflow rate of new high energetic binary resource strings into the pool R. After each update, i.e. after all agents have digested a resource string, the probability for each resource string in the pool to be replaced by a new fresh one, keeping the total number of resource strings constant, is γ. Here we will typically set $\gamma \in [0.003, 0.3]$.

μ is the mutation probability during reproduction, where an agent is uniformly changed to another of the 256 CA-rules. We will use $\mu = 0.01$ as a default value of the mutation rate.

β is the level of selective pressure, as it determines the importance of ΔE in calculating the reproductive rate, see eq. (1). The default value of in the current study is $\beta = 0.1$.

The population size is set to $N_A = 1024$, and the number of binary strings in the resource pool is $N_R = 5N_A = 5120$. The size of the binary strings is set to $L = 100$, and level of order in the inflowing strings is $p_0 = 0.95$, which gives, through eq. (2), an initial energy of $E_0 = 1 - s_0 \approx 0.8$. The initial condition of each simulation is a uniform distribution of species, i.e. 1024/256 = 4 organisms of each species, and a resource pool consisting of strings with the initial energy E_0.

Results

All ecosystem on earth are driven by energy entering the system either in the form of sunlight or in some chemical form such as for example glucose or ironsulphide. Similarly the dynamics in *Urdar* are driven by the flow of food strings with a high energy into the system, and if γ is set to zero the dynamics will eventually grind to a halt when all possible energy has been extracted from the resource pool, i.e. no new agents will be generated. The rate of energy supply is known to be of great importance to real ecosystems (Waide et al., 1999), and it is therefore of interest to analyse how the dynamics in our system depend on the flow rate of energy γ.

The most straight forward way of characterising the dynamics is to look at the time evolution of the species distribution. This is shown in fig. 3 for two different values of the flow rate, in (a) $\gamma = 0.3$ while in (b) $\gamma = 0.003$. The striking difference between these two simulations implies the interesting statement that the number of co-existing species in the low flow case is considerably higher. Hence one might say that a relative supply shortage encourages species diversification and cooperation. This relation is investigated in detail in Gerlee and Lundh (2010) and we will here focus on ecosystem stability and species interactions.

Ecosystem stability

These plots also show that at low flow rates the species distribution does not settle in a steady state but seems to fluctuate with different species dominating the ecosystem at different times. This shows that the dynamics of the system does not converge to a fixed-point, but instead obeys oscillatory or even chaotic dynamics. If the mutation rate is set to zero the dynamics settle on a species distribution with a diversity which still depends on the flow rate. However, the distribution is stable over time, which suggests that the small mutation rate is what drives the intermittent dynamics.

We can visualise the dynamics more easily if instead of viewing the frequency of all species in a 2-d plot as in fig. 3, pick a reference state $\mathbf{F^0} = (f_0^0, f_1^0, ..., f_{255}^0)$, and plot the L_1-distance from the reference state as a function of time, i.e.

$$\Delta F(t) = \sum_{i=0}^{255} |f_i^0 - f_i(t)|, \tag{3}$$

where $f_i(t)$ is the fraction of the agents belonging to species i (i.e. performing the elementary CA-rule i) at time t. An

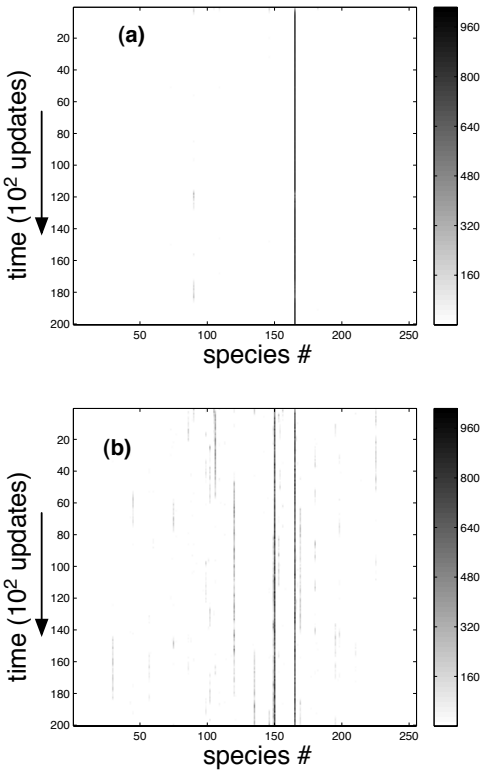


Figure 3: The time evolution of the species distribution for (a) $\gamma = 0.3$ and (b) $\gamma = 0.003$.

example of such a plot is shown in fig. 4, which illustrates the same simulation as in fig. 3b, where the reference state was picked as the final state of the system at $t = 2 \times 10^4$. From this point of view we can clearly see how the system exhibits long periods of stasis and seems to jump between different states corresponding to specific species configurations; as in the so called punctuated equilibria introduced in Eldregde and Gould (1972). This can be compared to different epochs in the history of the ecosystem, and is thus comparable to paleontological data, which we will return to in the discussion. The time spent in these states seems to vary heavily and in order to quantify this we measured the waiting time distribution, i.e. the probability of the species distribution remaining in the same state a time T. The mutations present in the system, together with the relatively small population size, introduces fluctuations into the system, and in order to avoid these the projected time series $\Delta F(t)$ was binned into 20 equal sized bins (as shown in fig. 4).

From this discretised data we calculated the cumulative probability $P(x > T)$ of finding the system in the same bin for at least T time steps. This was calculated from 50 different simulations each lasting $t_{max} = 2 \times 10^4$ times steps for $\gamma = 0.3$, 0.03 and 0.003. The result is shown in fig. 5, where the curves corresponding to the lower flow

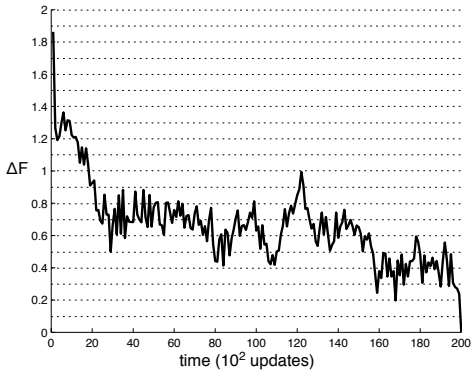


Figure 4: The species distribution shown in fig. 3b projected down to a one-dimensional state using (3). The dotted horizontal lines indicates the bins used for calculating the waiting times shown in fig. 5 below. The reference state $\mathbf{F^0}$ was picked as the final state of the system at $t = 2 \times 10^4$.

rates appear approximately as straight lines in a loglog-plot. This suggests that the waiting time scales as a power-law, and a linear regression showed that $P(T) \sim T^{-\alpha}$, where $\alpha \approx 2.6$ and 3.5 for $\gamma = 0.03$ and 0.003 respectively. On the other hand, the curve corresponding to $\gamma = 0.3$ is closer to a straight line in semilog-plot (see inset), and from this we found that $P(T) \sim e^{-\varepsilon T}$, where $\varepsilon \approx 0.04$. The exact slope of the curves naturally depends on the number of bins (a smaller bin size gives steeper curves), but the difference between the functional forms of the curves is robust. Please note that the waiting time for a random walk is exponential, which gives an indication of the difference in dynamics between the high and low flow rate.

Pair-wise species interactions

A natural question that arises is what kind of underlying dynamics gives rise to these transition patterns. If there existed for a fixed flow rate a single dominant species among the 256 possible then we would expect the evolutionary dynamics to converge to a species distribution and remain there. This is clearly not the case, at least not for the lower flow rates, which suggests that more complicated dynamics than simply the selection for the best metaboliser occurs in the system.

This is in fact obvious if we return to the schematic of the model and also realise that different species have varying capacity to metabolise different strings. The fitness of a species depends on its ability to extract energy from the strings in the resource pool, but the composition of the resource pool in turn depends on what species are present in the ecosystem. This means that the fitness of a species depends on state of the entire ecosystem and will therefore change as the system evolves.

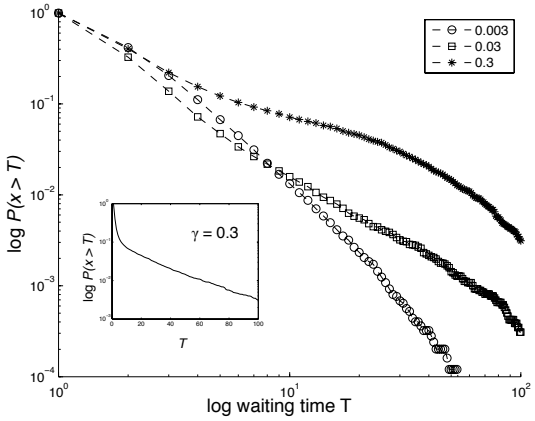


Figure 5: The cumulative distribution of waiting times plotted in a loglog-diagram for three different values of γ. For low flow rates the waiting times appear to scale as a power-law, while for high flow it seems to follow an exponential distribution as indicated in the inset where the graph follows approximately a straight line over a long period in the semilog-diagram.

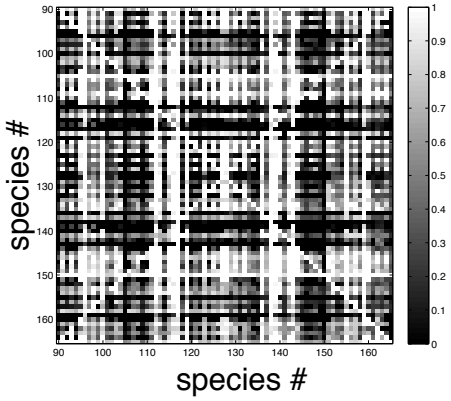


Figure 6: Excerpt of the matrix C describing the pair-wise species interactions in the system. White and black correspond to complete dominance, while any shade of grey corresponds to co-existence.

The simplest possible way to analyse the species interactions is to simulate the dynamics when only a pair of species are present and the mutation rate is set to zero. This of course neglects higher-order interactions, between conglomerates of species, which might influence the dynamics, but at least it represents a starting point for a deeper understanding of the system. We probed these species interactions by initialising the system with a 9:1 ratio in the abundance of a pair of species and then ran the simulation (without mutations) for 1000 time steps or until only one of the species remained. At the end of the simulation we recorded the abundance of the species and stored the frequency of the initially abundant species in a matrix C. Element c_{ij} thus holds the equilibrium frequency of species i when the initial ratio between $i : j$ was $1 : 9$. This experiment was carried out for all possible pairs of species in the range 90-164 of which there are $74 \times 74 = 5476$, and an excerpt of the resulting matrix is shown in fig. 6. Here white and black correspond to complete dominance, while any shade in between signifies stable co-existence between the species.

A striking feature is that co-existence seems to be a common mode of interaction. This emphasises what was discussed before, namely that the replication rate of species depends on the totality of species present (including itself) in the ecosystem. In the case of co-existing species, the increase in abundance is balanced by a reduction in reproduction rate, a phenomenon known as negative frequency-dependent selection (Huisman and Weissing, 1999), and when the replication rate of both species is balanced a steady-state is attained.

The interaction matrix in most cases satisfies $c_{ij} + c_{ji} = 1$, which means that the equilibrium concentration of the species is independent of the initial condition, but there are some interesting exceptions from this rule. Firstly we have the anti-diagonal of the matrix where $c_{ij} + c_{ji} \approx 2$, and this is due to the underlying symmetry of the cellular automaton rules. The pairs on the anti-diagonal are in fact rules that are inverses of each other when viewed in binary representation. For example rule 145 = 10010001_2 and its anti-diagonal partner is rule $255 - 145 = 110 = 01101110_2$. When these rules are applied to a generic binary string the output strings they yield are inverses of each other, which by symmetry of the entropy function imply that they have the same entropy. This means that the two rules, when competing in isolation, are neutral and the only evolutionary force acting on the system is random drift. The consequence of this is that the initially dominant rule is more likely to win and therefore we observe $c_{ij} \approx c_{ji} \approx 1$ (or visually a white line) on the anti-diagonal. Note that this does not imply that the two species are identical in their competition with other rules, and this has some important consequences for the dynamics of the model.

Secondly we have the cases where $1 < c_{ij} + c_{ji} < 2$, which indicates that the initial condition in fact influences the equilibrium concentration. Upon further inspection we found that the dynamics of these pair-wise interactions contain two stable fixed-points, as opposed to one which is the case in all other interactions. Typically the only fixed-point lies either, in the case of co-existence, in the interior of the phase space at $(i, j) = (c, 1 - c)$, for the equilibrium concentration c, which satisfies $0 < c < 1$, or in the case of dominance at (0,1). In the above mentioned cases both an interior and a boundary fixed-point are present, and this implies that the dynamics can converge either to co-existence or dominance depending on the initial

frequencies of the species.

Rock-Paper-Scissors

The presence of co-existence in the pair-wise experiments gives a reasonable explanation of the large degree of co-existence in the full simulation (cf. fig. 3), but it does not explain why the species configuration never settles into a steady state. The lack of stability must be an inherent in the species configuration itself, and one possible explanation is that the property of being able to invade another species is not transitive. By this we mean that if a_i invades a_j, and a_j invades a_k, then it is not necessarily so that a_i invades a_k. If on the contrary a_k invades a_i we have what is called an intransitive cycle, similar to the Rock-Paper-Scissors game.

In order to investigate this possibility we searched the matrix C for species triplets which satisfy the above condition, and found 59 unique triplets (containing 44 different species) which satisfied the condition of intransitivity. A suitable way to illustrate this is with a network where the species are represented as nodes and a directed link connects node A and B if species A can invade species B. This is shown in fig. 7, and in this figure the intransitive relations appear as directed triangles of which there are plenty. For clarity we have only included species involved in at least one intransitive interaction. The network consists of 4 connected components suggesting a certain degree of modularity, which could allow for independent competition occurring simultaneously in the well-stirred environment. Further analysis showed that all except two triplets exhibited the double fixed-point property discussed above, and thus exhibit a weaker form of intransitivity. The two fully intransitive triplets where given by (120,145,158) and (120,131,158) and are highlighted in fig. 7. Mathematical analysis has suggested that RPS-dynamics can give rise to oscillatory behaviour due to the cyclic replacement of the species (Laird and Schamp, 2009). We investigated this possibility by performing experiments where the three members of an intransitive cycle were present in equal proportion in the initial population and the system was run without any mutations. We did however not observe oscillatory behaviour, but instead the dynamics converged on either a pair of species co-existing (and one species going extinct) or one single species dominating the system. This discrepancy from the analytical result is most likely due to a difference in the rates of replacement of the species, which in the analytical treatment is set to be equal for all interactions. This deviation from theory has also been observed in a bacterial system exhibiting RPS-dynamics (Kerr et al., 2002).

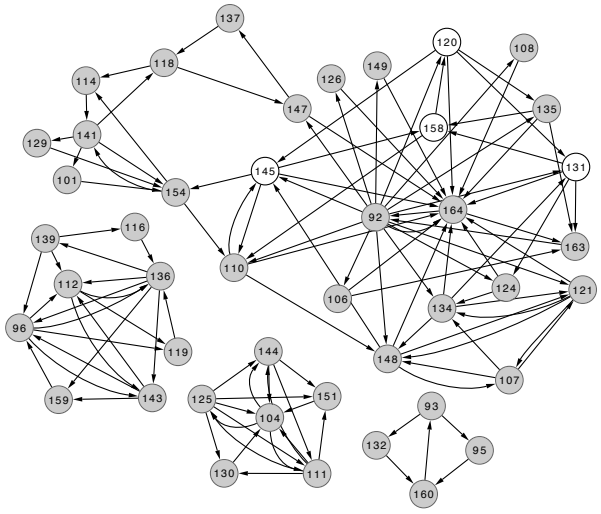


Figure 7: Network illustrating the intransitive species interactions. An edge points from node a to b if species a wins over b in a pair-wise invasion experiment, i.e. $C_{ab} > 0.75$. Intransitive triples are seen as cyclic triangles in the network. The species involved in fully intransitive competition (not involving multiple fixed-points) are highlighted.

Discussion

In this paper we have presented an Alife-platform *Urdar*, based on the mechanism of cross-feeding, which is observed in many microbial ecologies. The components of the platform are fairly simple consisting of elementary CA rules that transform binary strings. Similar systems have been analysed by for example Dittrich et al. (2001) and Ikegami and Hashimoto (1995). The former considered a matrix multiplication chemistry, where binary strings could act both as agents and substrate, and in which stable autocatalytic cycles emerged. In the latter a different formalism was applied, where agents defined as Turing machines acted on tapes represented as binary strings. What these systems did not include was a notion of energy necessary for replication, which is a central feature of *Urdar*.

This energy is obtained by increasing the entropy (disorder) of binary food strings. Despite of its simplicity the system exhibits surprising features such as a high degree of species diversity, non-stationary dynamics, and periods of stasis with broad distribution of waiting times.

The latter have also been observed in other evolutionary models such as Bak and Sneppen (1993) and Sole and Manrubia (1996), and relates to the punctuated equilibrium hypothesis put forward by Eldregde and Gould (1972). In the original conception of the hypothesis it was believed that geographic separation was a necessary condition. Our results show that long periods of stasis of stasis can appear in cross-feeding ecosystem that lacks any spatial component, and where the dynamics are driven by the mutual dependence between the species.

The above mentioned features are all driven by the cross-feeding interactions between the species and are more pronounced at low flow rates of high energy strings into the

system. One way to study these interactions is to perform pair-wise invasion experiments captured in the matrix C (see fig. 6), which reveal that co-existence is quite common in the system. Studying this matrix we also found intransitive relations between three different species similar to the Rock-Paper-Scissors game. This type of interactions are commonly found in real ecosystems, and are know to promote biodiversity (Kerr et al., 2002; Laird and Schamp, 2009), suggesting a source of the observed non-stationarity in our system.

However, preliminary results indicate that removing the 44 species involved in intransitive relations from the ecosystem (and prohibiting mutations to them) does not reduce species diversity nor increases ecosystem stability. This suggest that higher-order interactions not captured by the pair-wise invasion experiments are responsible for the inherent instability of *Urdar*.

Future work

The experiments presented in this article only scratch the surface of this surprisingly complex ecosystem, and whole host of interesting questions remain to study. One obvious question that remains unanswered regards the underlying mechanism driving the above mentioned non-stationarity. One could also investigate the dynamics from a different point of view by making use of the metabolic history of all food strings (i.e. the list of species each string has been metabolised by). This makes it possible to map out which species engage in cross-feeding, and from this information generate a network of ecological interactions. Another possibility is to examine to which extent the process of evolution maximises productivity from an ecosystem point of view, i.e. how well does the evolved species composition do compared to an optimal species composition which maximises productivity (for a given flow rate). Further, the model could also be extended to include features present in real biological systems, such as a distinction between the genotype and phenotype of the organisms and a spatial dimension which would impact the nature of the species interactions.

Aknowledgement

This work was in part supported by Adlerbertska Forskningsstiftelsen.

References

Bak, P. and Sneppen, K. (1993). Punctuated equilibrium and criticality in a simple model of evolution. *Phys. Rev. Lett.*, 71:4083–4086.

Costa, E., Perez, J., and Kreft, J.-U. (2006). Why is metabolic labour divided in nitrification? *Trends in Microbiology*, 14:213–219.

Dejonghe, W., Berteloot, E., Goris, J., Boon, N., Crul, K., Maertens, S., Hfte, M., Vos, P. D., Verstraete, W., and Top, E. (2003). Synergistic degradation of linuron by a bacterial consortium and isolation of a single linuron-degrading variovorax strain. *Appl Environ Microbiol*, 69(3):1532–1541.

Dittrich, P., Ziegler, J., and Banzhaf, W. (2001). Artificial chemistries - a review. *Artificial Life*, 7(3):225–275.

Doebeli, M. (2002). A model for the evolutionary dynamics of cross-feeding polymorphisms in microorganisms. *Population Ecology*, 44:59–70.

Eldregde, N. and Gould, S. (1972). *Models in Paleobiology*, chapter Punctuated equilibria: an alternative to phyletic gradualism, pages 82–115. Freeman Cooper.

Gerlee, P. and Lundh, T. (2010). Productivity and diversity in a cross-feeding population of artificial organisms. *Evolution, In Press*.

Helling, R. B., Vargas, C. N., and Adams, J. (1987). Evolution of escherichia coli during growth in a constant environment. *Genetics*, 116:349–358.

Huisman, J. and Weissing, F. J. (1999). Biodiversity of plankton by species oscillations and chaos. *Nature*, 402(6760):407–410.

Hutchinson, G. (1959). Homage to santa rosalia or why are there so many kinds of animals? *American Naturalist*, 93:145–159.

Ikegami, T. and Hashimoto, T. (1995). Active mutation in self-reproducing networks of machines and tapes. *Artif Life*, 2(3):305–318.

Katsuyama, C., Nakaoka, S., Takeuchi, Y., Tago, K., Hayatsu, M., and Kato, K. (2009). Complementary cooperation between two syntrophic bacteria in pesticide degradation. *Journal of Theoretical Biology*, 256(4):644–654.

Kerr, B., Riley, M. A., Feldman, M. W., and Bohannan, B. J. M. (2002). Local dispersal promotes biodiversity in a real-life game of rock-paper-scissors. *Nature*, 418(6894):171–174.

Laird, R. A. and Schamp, B. S. (2009). Species coexistence, intransitivity, and topological variation in competitive tournaments. *J Theor Biol*, 256(1):90–95.

Pfeiffer, T. and Bonhoeffer, S. (2004). Evolution of cross-feeding in microbial populations. *Am Nat*, 163:E126–E135.

Rozen, D. and Lenski, R. (2000). Long-term experimental evolution in escherichia coli. viii. dynamics of a balanced polymorphism. *Am. Nat.*, 155:24–35.

Schrödinger, E. (1944). *What is Life?* Cambridge University Press.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27.

Sole, R. and Manrubia, S. (1996). Extinction and self-organized criticality in a model of large-scale evolution. *Phys. Rev. E*, 54:R42–45.

Stams, A. J. (1994). Metabolic interactions between anaerobic bacteria in methanogenic environments. *Antonie Van Leeuwenhoek*, 66:271–294.

Waide, R., Willig, M., Steiner, C., G., M., Gough, L., Dodson, S., Juday, G., and Parmenter, R. (1999). The relationship between productivity and species richness. *Annu. Rev. Ecol. Syst.*, 30:257–300.

Wolfram, S. (1983). Statistical mechanics of cellular automata. *Rev. Modern Phys.*, 55(3):601–644.

Complex Networks

Optimizing Spatially Embedded Networks for Synchronization

Markus Brede[1]

[1]CSIRO Centre for Complex Systems Science, Canberra, ACT 2602, Australia
Markus.Brede@csiro.au

Abstract

In this paper we consider the problem of organizing networks of spatially embedded oscillators to maximize the propensity for synchronization for limited availability of wire, needed to realize the physical connections between the oscillators. We consider two extensions of previous work (Brede, 2010b): (i) oscillators that can flexibly arrange in space during the optimization process and (ii) a generalization to weighted networks. In the first case, we discuss the emergence of spatially and relationally modular network organizations, while in the second case the emphasis of our analysis is on link heterogeneity and the particular organization of strong and weak links that facilitates synchronization in space.

Introduction

Probably starting with Huygens observation of synchronized motions among nearby pendula clocks, synchronization phenomena have long attracted much interest among physicists. Synchronization is ubiquitous in the biological (Winfree, 1980) and in the engineered world (Blekhman, 1988): fireflies that flash in unison, cardiac pacemaker cells, rhythms in the brain or power stations and laser systems are just a few examples (Arenas et al., 2008; Manrubia et al., 2004). All of these systems are distributed coupled systems that can be described by complex networks. Recent findings, that many such networks have highly non-trivial topologies have given rise to a wave of studies about synchronization on complex networks.

One overriding question in this research has been to identify characteristics of network topology that are correlated with enhanced or poor synchronization characteristics. Even though such a statistical characterization has caveats (Atay et al., 2006), important findings have resulted, which allow a rough "rule of thumb" characterization of a networks' propensity for synchronization. Many factors that influence synchronization have been identified: homogeneous network topologies such that every node receives the same strength of an 'in-signal' (Donetti et al., 2005; Motter et al., 2005; Hwang et al., 2005; Chavez et al., 2005; Nishikawa and Motter, 2006b; Brede, 2010c; Nishikawa and Motter, 2010), an 'entangled' structure that does not allow for separate communities of nodes (Donetti et al., 2005), short pathlengths (Watts and Strogatz, 1998) and disassortative degree mixing are just a few examples. Even though optimal network topologies have thus been well-classified, understanding the role of constraints on the network topology and the varying trade-offs between the mentioned characteristics still pose a challenging problem.

One natural source of constraints on network organization is the spatial embedding typical to almost all application systems. The biological fitness (or in an engineering context, a system's optimality) is then not only determined by it's synchronization properties, but also by cost factors associated with requirements to realize the physical connections in space that are needed to establish the coupling. If one considers a system without the spatial embedding, this synchronization cost is related to the number (and possibly weight) of links. In fact, for this case it has been shown that optimal synchronization can be achieved for minimized cost (Nishikawa and Motter, 2006a). However, for spatially embedded networks the cost to establish linkages is a combination of the number and length of links: It can be seen as the length of a wire needed to realize the network links in space.

The problem of optimal synchronization in space has recently been addressed (Brede, 2010b), finding that over a large range of parameters synchrony-optimal networks are small worlds with power law distributed link length. The more severe spatial constraints, the steeper the decay of the power law describing the link length distribution. For several reasons this is an important finding: small worlds with power law distributed link length have been found in neurological networks (Schüz and Braitenberg, 2002), the (physical) internet (Yook et al., 2002) or networks of wire in electronic circuits (Zarkesh-Ha et al., 2000) – all systems where synchronization plays a role. Moreover, random walks on such particular small worlds establish fractal movements patterns in the underlying space, which could have relevance for optimal search (or foraging) patterns (Viswanathan et al., 1999).

Optimization and evolutionary algorithms have a natural place in this research, since they allow for the numerical construction of networks with enhanced synchronization characteristics. Further, apart from the scientific problem setting, many of the biological systems, like the brain, where synchronization plays a role, are systems that have evolved to their current state over a long period of time. Synchronization very likely has played a role in their evolution, such that one can imagine an algorithm that optimizes as networked system for enhanced synchronization as a model to mimick this evolution process.

In this paper we discuss two natural extensions of the abovementioned study Brede (2010b) and investigate whether the power law distributions that classify optimal networks persist in these more general situations as well. First, after a short description of the framework and methodes we employ, we consider optimal synchronization in systems where the nodes are not fixed in space, but are free to change their relative arrangement during the optimization for synchrony-enhancement. Second, in the next following section, we consider the case of synchrony-optimal weighted networks in space. The paper concludes with a section that summarizes our results and puts them into a more general context.

The Model

We investigate identical synchronization in systems of N coupled oscillators, the collective dynamics of which is given by

$$\dot{x_i} = f(x_i) + \sigma \sum_j A_{ij}(g(x_j) - g(x_i)). \quad (1)$$

In the above equation, the function f describes the dynamics of the individual oscillators (without coupling), the matrix A_{ij} is the adjacency matrix of the coupling network, σ the coupling strength and the function g characterizes the so-called 'inner coupling', i.e. defines how the oscillators influence each other. The equation can be rewritten as $\dot{x_i} = f(x_i) + \sigma \sum_j L_{ij} g(x_j)$, which introduces the graph Laplacian matrix belonging to the adjacency matrix A via $L = I - A$, where $I = (\delta_{ij})$ is the identity matrix. It is important to note that in all scenarios considered in this paper A is symmetrical and has only positive entries, such that all eigenvalues of L are real and nonnegative. Without loss of generality we will further restrict the study to connected networks. In this case L has exactly one zero eigenvalue and one can label the eigenvalues of L in ascending order $0 = \lambda_1 < \lambda_2 < ... < \lambda_N$.

A big step forward in understanding identical synchronization in the system (1) is due to Pecora and Carroll (Pecora and Carroll, 1998), who analyzed the stability of the synchronized state $\dot{x} = f(x)$. In (Pecora and Carroll, 1998) they were able to show that for a large class of oscillators f and inner couplings g, the stability of the fully synchronized state is determined by the eigenratio $e = \lambda_N / \lambda_2$. Importantly, the eigenratio analysis abstracts from the details of the underlying dynamics (i.e. the function f) and allows an analysis of the influence of the connection architecture (given by the adjacency matrix A of the coupling network) for a general class of dynamics. Essentially, a network has a superior propensity to synchronize when the spread of the eigenvalues is as small as possible – or e close (or identical) to one.

The spatial component of the model is introduced by allocating nodes spatial locations $l_i > 0$ in a one-dimensional space with periodic boundary conditions. Then, if $l_{\max} = \max_i(l_i)$, a spatial distance metric can be defined via $d(i,j) = \min(|l_i - l_j|, l_{\max} - |l_i - l_j|)$ and the amount of 'wire' needed to connect the nodes in space according to a network A is

$$W = \sum_{i<j} A_{ij} d(i,j). \quad (2)$$

As already suggested in (Brede, 2010b), spatial constraints on the network evolution can be considered via the optimization of the synchronization properties of the network for limited amount of wire W. Alternatively, a more elegant framework can be the minimization of an energy-like goal function that combines considerations of synchronization properties with the minimization of the amount of wire used via

$$E = \beta W + (1 - \beta) e, \quad (3)$$

where the trade-off parameter $0 \leq \beta \leq 1$ weighs the importance of wire minimization versus that of enhanced synchronization during network evolution. Compare also (Mathias and Gopal, 2001; Sole and Ferrer i Cancho, 2003; Brede, 2008) for other studies where a similar framework has been used in different contexts.

Importantly, if $\beta = 1$ the goal function is solely determined by the amount of wire. The minimum of E then corresponds to a network configuration in which only spatial nearest neighbours are connected – a configuration which is known to have very poor synchronization properties. On the other hand, when $\beta = 0$ considerations of wire and the underlying space play no role in the minimization of E. This case corresponds to (Donetti et al., 2005) (and apart from (Brede, 2010b) all other studies of optimal identical synchronization on networks, cf. (Arenas et al., 2008)). Note, that if one decreases β towards $\beta = 1$, the 'severity' of spatial constraints in the network evolution process can be tunedx.

We approach the problem of minimizing (3) via a numerical optimization scheme using simulated annealing. The scheme consists of a series of rewiring suggestions, which

are accepted if they improve the fitness or energy (3) of the network configuration. Prototypically, even though step 2 is modified according to the slightly more general problem definitions in section III and IV, we employ the following scheme:

1. Start with an Erdös-Rényi random graph with exactly L links and distribute oscillators uniformly in space at locations $l_i = i$, $i = 0, ..., N-1$. Calculate the fitness of the first network configuration.

2. Rewire one or several links. Calculate the resulting network fitness E' of the modified configuration according to Eq. (3) and accept if $E' < E$ or with probability $p \propto \exp(-\nu(E - E'))$ otherwise. The inverse temperature ν of the annealing procedure is gradually reduced as the optimization progresses.

3. Terminate the algorithm if no large improvement in E was achieved during a certain number of iterations.

Because for larger networks the optimization procedure did not result in a unique optimal configuration (and due to the inherent difficulty of making sure a numerical optimization approach actually achieved a global optimum) we typically constructed around $R = 100$ optimal network configurations by the algorithm. In both situations considered in more detail in the following sections, all the near-optimal networks proved to be structurally very similar, which underlines that the findings we will discuss below are robust. The structural similarity of the constructed networks also gives support to the approach to optimize linear combinations of the quantities of interest rather than to construct the full Pareto front in a multi-objective optimization approach.

Optimal synchronization with flexible node locations

In the previous study (Brede, 2010b), optimal synchronization was considered for the case of spatial networks with nodes that have fixed locations in space. Here, we extend the framework and consider nodes that can arrange freely in space during the optimization procedure. However, without further constraint this would clearly imply that all nodes drift to one location, thus allowing for complete connectivity without cost of wire. To prevent this and to study which arrangement of nodes is optimal, we introduce a further constraint, requiring that the average spatial distance of the nodes remains the same during the optimization, i.e. that

$$D = \frac{2}{N(N-1)} \sum_{i<j} d(i,j) = \text{const.} \tag{4}$$

Accordingly, we then modify step 2. of the optimization procedure of the previous section, in which we now also include suggestions for location changes of nodes $l_i \rightarrow$

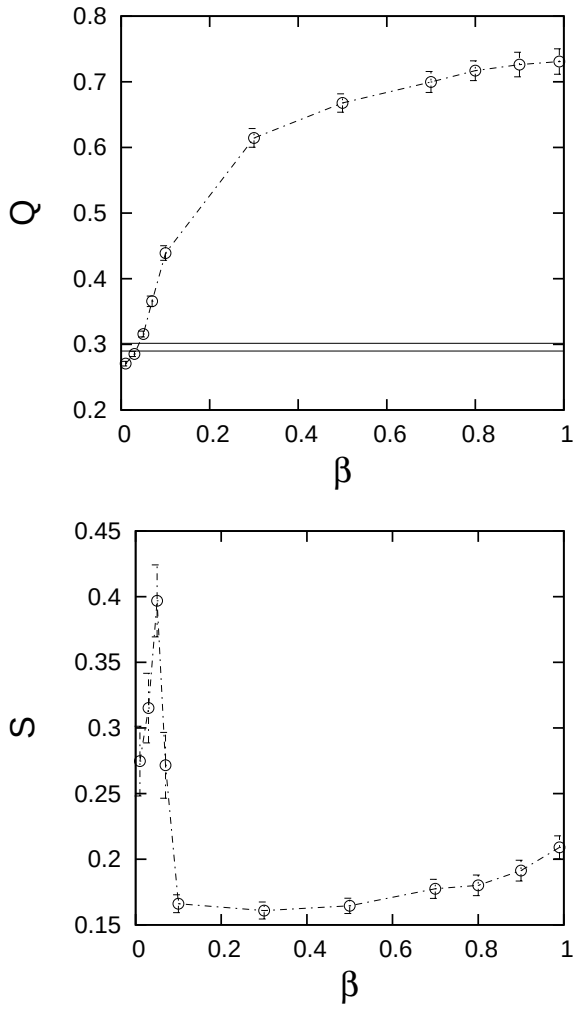


Figure 2: Dependence of the relational (top) and spatial modularity (bottom) of the evolved networks and spatial arrangements on the trade-off parameter β: For reference, the horizontal lines indicate the range the respective quantities would assume for a an Erdös-Rényi random graph whose nodes are uniformly distributed in space. In the plot of the spatial modularity the lines are omitted for scaling reasons, one has $S < 10^{-3}$ in that case. All data are for networks of $N = 100$ nodes with $L = 400$ links and are averaged over 100 different initial configurations for each β.

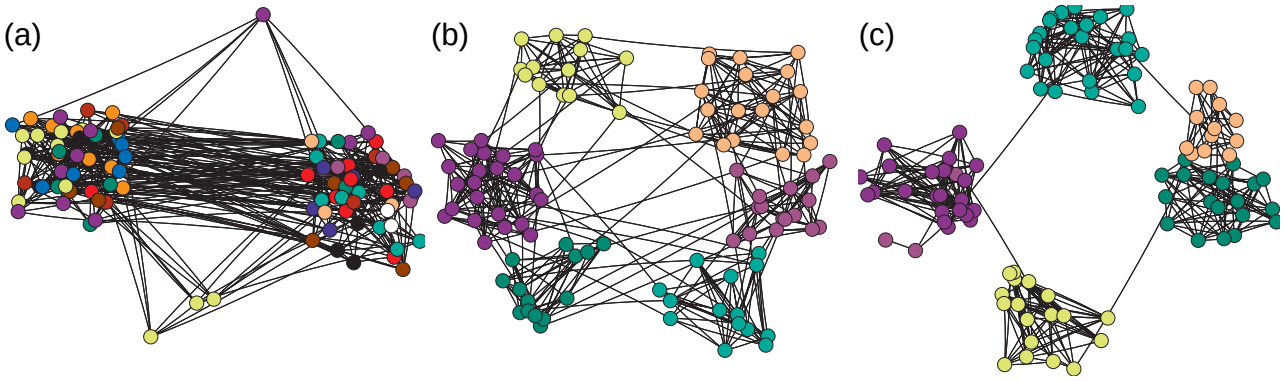


Figure 1: Examples of evolved networks for different trade-offs between cost of wire and desireability for superior synchronization: (a) $\beta = 0.01$ (very low cost of wire), (b) $\beta = 0.5$ (balanced costs for wire and synchronization), and (c) $\beta = .01$ (very high cost of wire). The networks are of size $N = 100$ and contain $L = 400$ links. In the figure vertices have been colored according to the modules they belong to (modularities are $Q = .26$ for (a) and $Q = 0.71$ and $Q = 0.78$ for (b) and (c)). The spatial locations roughly correspond to the evolved spatial locations of the nodes during the optimization, however a random number was added to make vertices distinguishable.

$l_i + \Delta l_i$. After such a location change suggestion, D' of the modified configuration is calculated and all locations l_i, $i = 1, ..., N$ are scaled by D/D', i.e. we set $l_i \rightarrow D/D' l_i$, to ensure $D = const.$ during the optimization.

Figure 1 gives some illustrations of example networks constructed by optimizing the energy (3) for three different scenarios: (a) very low cost of wire, (b) balanced cost of wire and desirability of superior synchronization and (c) expensive wire. The figures already illustrate a number of differences in network organization to the results reported in (Brede, 2010b). First, it becomes apparent that two distinct classes of link lengths can be identified: short links and long links. The gap between these two types of links depends on the trade-off parameter β – it is large when wire is very costly or very inexpensive and relatively small when the cost of wire and synchronization needs are balanced. Second, depending on β, the network organization can become distinctly modular. Third, it becomes apparent that the spatial locations of nodes become distinctly clustered, such that the nodes either crowd at two (for the case of low β) or more (for intermediate and large β) spatial locations.

Modularity is an important property of many real-world networks, see, e.g. (Girvan and Newman, 2004). It denotes the fact that networks are organized into communities of nodes that are more strongly connected to each other than to the rest of the network. A widely accepted measure to quantify network modularity has been introduced in (Girvan and Newman, 2004)

$$Q = \sum_m [L_m/L - (d_m/2L)^2]. \quad (5)$$

In eq. (5) the index m runs over all network communities, L_m denotes the number of links within a module, d_m the sum of all degrees of nodes in module m and $L = \sum_{i<j} A_{ij}$ the overall number of links in the network. Several algorithms to identify modules in networks have been suggested. Because the networks that we evolved above are relatively small, we use extremal stochastic optimization (Duch and Arenas, 2005) to calculate Q and identify modules. As an example of results of the module identification see figure 1a-c, where we have identified modules by the colors of the nodes. The respective values of the modularity measure Q are given in the caption of the figure.

For an analysis of the spatial modularity of the evolved networks we have analyzed the correlation function $G(x)$ that gives the density of nodes at distance x from an average node. A plot of G for different trade-off parameters allows the distinction between two scenarios (see also figure 1): (i) $G(x)$ is u-shaped with two peaks at $x = 0$ and $x = l_{\max}/2$ and a flat trough in between which clearly corresponds to an arrangement of nodes into two clusters separated by the maximum distance and (ii) $G(x)$ has one sharp peak at $x = 0$ which corresponds to an arrangement into several spatial clusters. A more thorough investigation of the link length distributions and widths of the peaks of the correlation function G suggests a cut-off of around $\Delta x = 0.01$, links with length $l \leq \Delta x$ being classified as 'short' and links with lengths $l > \Delta x$ being 'long'. Then, one can define a spatial cluster as the maximum number of nodes with distances less than Δx or

$$S = \int_0^{\Delta x} G(x) dx. \quad (6)$$

Thus, our spatial modularity measure is the average fraction of nodes in one spatial '0.01-cluster'.

In the top and bottom panel of figure 2 we present a more detailed analysis of the modularities of the evolved

networks. A short glance at the spatial modularity reveals two typical structural regimes that are separated by a sharp transition at around $\beta_c = 0.1$. Below the transition for $\beta < \beta_c$ the evolved networks show a clear two spatial cluster regime, both clusters being separated by the maximum spatial distance. A comparison of the respective network modularity to that of random networks (indicated by the two lines in Fig. 2) shows that the network modularity is suppressed in this regime. The spatial clustering is thus not correlated with a corresponding network modularity. As an aside, it is also clear that $S \to \Delta x l_{\max}/(2N)$ (uniform distribution of nodes in space) as $\beta \to 0$. From this argument one understands that the spatial modularity peaks and then declines again in the $\beta < \beta_c$-regime.

Above the transition, with $S > 0.15$ the networks are still strongly spatially clustered, albeit the spatial clustering is strongly reduced in comparison to the $\beta > \beta_c$ case. This indicates the presence of multiple ($\approx 1/S$) smaller spatial clusters. In contrast to the case of $\beta < \beta_c$, the spatial clustering goes hand in hand with strong network modularity. Closer investigation reveals that membership of nodes to spatial modules is correlated with membership in network modules, which already hints to a mechanism of module formation. Clearly, in terms of wire it is 'cheap' to connect near-by nodes. Thus, there is a positive feedback mechanism: near-by nodes are likely to move closer to each other. This makes it cheaper to connect them and fosters the establishment of connections to other near-by nodes, thus facilitating network modularity. Network modularity in turn causes more spatial clustering since moving nodes of the same module spatially closer to each other further reduces the cost of wire.

Hence, allowing for flexible node locations during the network evolution leads to the formation of very different optimal network organizations than described in (Brede, 2010b), namely a very clear two mode structure of the link length distribution compared to the presence of all length scales leading to the power laws observed in (Brede, 2010b). Interestingly, the additional degree of freedom leads to the emergence of spatial clustering and modular network organization, and, associated with it, separate time-scales of synchronization processes (Arenas et al., 2006). For a more detailed discussion of these networks the reader is referred to (Brede, 2010d).

Optimal synchronization in weighted networks

In this section we are interested in weighted synchrony-optimal networks in space. In order to understand the influence of weights and spatial arrangement separately, like (Brede, 2010b) we consider nodes at fixed spatial locations $l_i = i$, $i = 0, ..., N-1$ that do not change position during the optimization. However, links A_{ij} are now not restricted to binary values, but can assume any weight $A_{ij} \geq A_{\min}$. The lower cut-off $A_{\min}$ was introduced for reasons of limited computation time and limited numerical precision in the eigenvalue calculations.

A larger coupling A_{ij} between two nodes allows for better synchronization between the nodes i and j. However, larger coupling also requires more wire and thus implies a larger cost for the physical connection of the nodes in space. A reasonable assumption is that the connection strength between two nodes is proportional to the thickness of the wire to connect the nodes. Hence, assuming a wire of constant density, the cost C of the wire is proportional to its length in space and the connection strength, such that $C_{ij} = d(i,j)A_{ij}$. One may also think of more general formulations for the cost function like $C_{ij} = d_{ij}h(A_{ij})$, which we leave for future work.

If one considers optimal weighted networks in the framework of the stability analysis of the synchronized state which leads to the eigenratio $e = \lambda_N/\lambda_2$ as a measure for synchronization, it is important to note that for any coupling network with Laplacian matrix L one has $e(kL) = e(L)$ for any scaling factor $k > 0$. Also, one has $e = 1$ for the fully connected graph with $L^F_{ij} = 1$ for $i \neq j$ and $L^F_{ii} = -N + 1$. Thus, for any coupling network configuration in space with $E(\beta) > 1 - \beta$ one can always choose a small enough factor k, such that the fully connected graph with link weights k has a smaller energy. As one easily realizes, however, this is a consequence of the different scaling of both contributing factors to Eq. (3). A more adequate problem definition that avoids this scaling issue is to introduce a scaled cost of wire via

$$C_{ij} = \frac{L \sum_{i<j} d(i,j)A_{ij}}{\sum_{i<j} A_{ij}}, \tag{7}$$

where $L = \sum_{i<j} H(A_{ij})$ and $H(x) = 1$ if $x > A_{\min}$ and $H(x) = 0$ otherwise. In Eq. (7) every link contributes to the cost with its weight relative to the average weight of links $\overline{w} = \sum_{i<j} A_{ij}/L$. One can then substitute (7) into Eq. (3) and obtains

$$E(\beta) = \beta \sum_{ij} C_{ij} + (1-\beta)e. \tag{8}$$

In fact, since $\overline{w} = 1$ for binary networks the definition (8) coincides with (3) for this case.

To construct optimal weighted networks in space we modify step 2. of the network evolution algorithm outlined before, by now considering weight transfers between links in the network, i.e. for randomly selected i, j with $A_{ij} \geq \epsilon$ and randomly selected k, l we suggest a reconfiguration $A_{ij} \to A_{ij} - \epsilon$ and $A_{kl} \to A_{kl} + \epsilon$. The suggested amount of the weight transfers ϵ is randomly selected from the interval $[A_{\min}, s]$. Best performance of the algorithm could be achieved when one starts with $s \approx 2\overline{w}$ and then decreases s linearly during the optimization.

Figure 3 displays some illustrations of typical optimal weighted networks for various trade-off parameters β. As

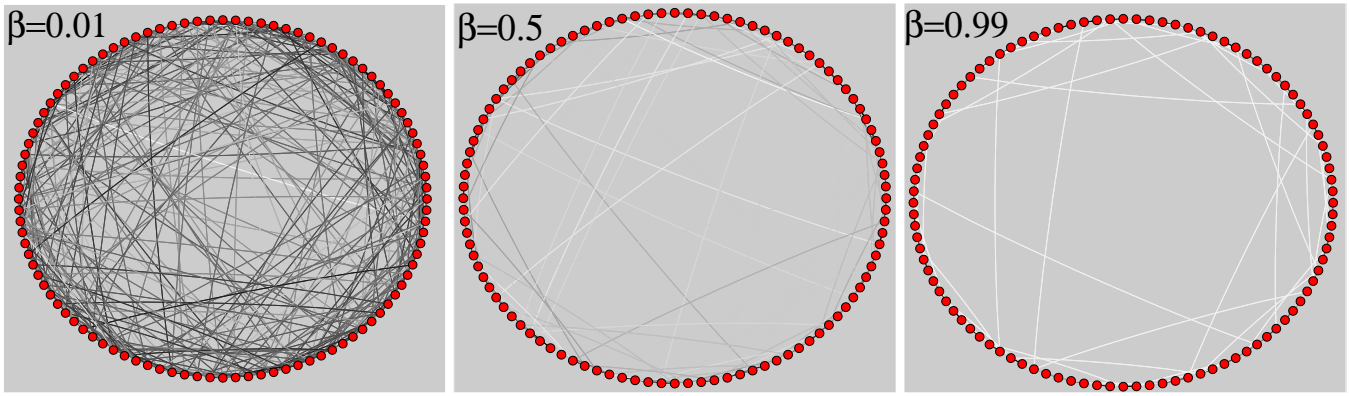


Figure 3: Illustration of some example weighted optimized networks for trade-off parameters $\beta = 0.01, 0.50$ and 0.99. The shade of grey of the links corresponds to link weight, weak links are white and strong links black. The background is shaded in grey to demonstrate the presence of very weak long links.

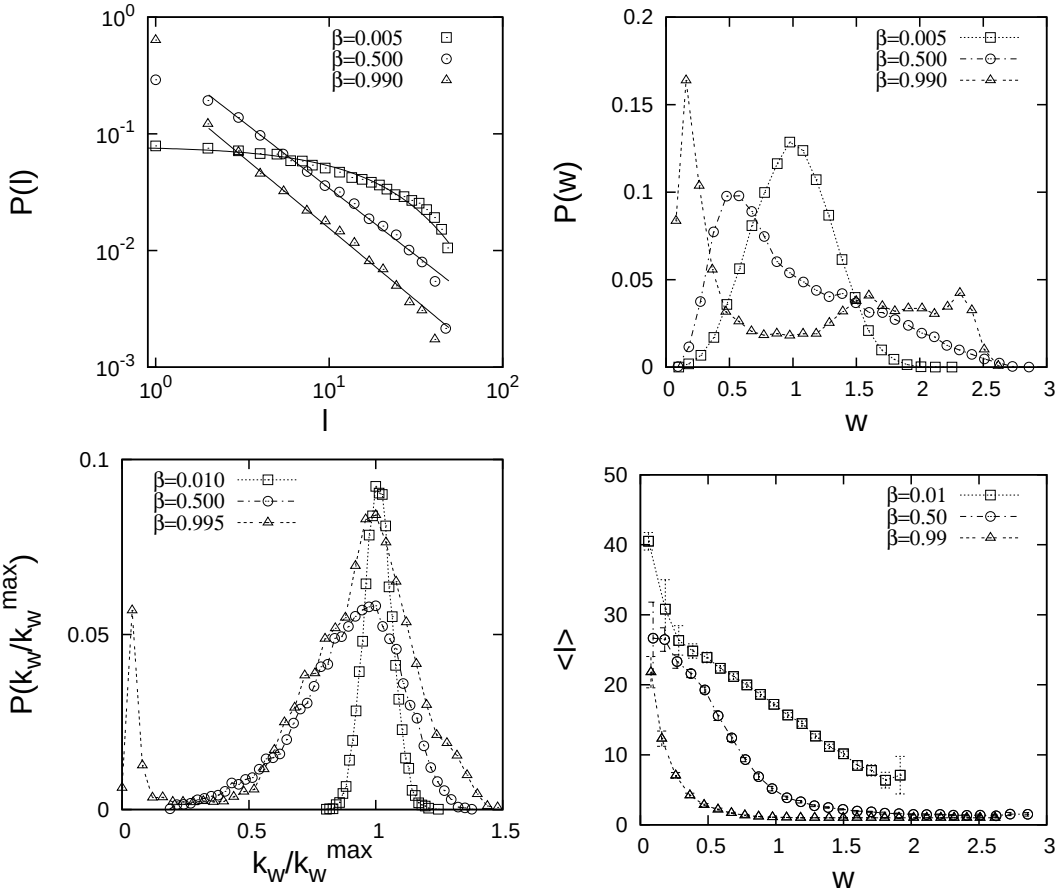


Figure 4: Network statistics for typical optimal weighted networks for small ($\beta = 0.005$), intermediate ($\beta = 0.5$) and large ($\beta = 0.99$) trade-off parameter: (top left) distribution of link lengths, (top right) distribution of link weights, (bottom left) distribution of weighted degrees (normalized by the respective maxima of the distributions for different β) and (bottom right) dependence of link length on link weight. The data have been averaged over 100 optimized networks of size $N = 100$.

one would expect, links are relatively dense and strong for low β and become increasingly scarcer and weaker when β is increased. Importantly, however, a careful inspection of the figures reveals that in all situations strong and weak links are present: strong links typically connecting spatially close nodes whereas weak links establish long-range connections.

For a more detailed investigation we constructed ensembles of 100 optimized networks of $N = 100$ nodes for various trade-off parameters β. To understand the pecularities of a given network, comparisons to suitable randomized null models are necessary. For the case of spatially embedded networks of interest here, a possible null model are (connected) networks with the same spatial constraint, i.e. random weighted networks that use the same amount of wire than the original network. Such networks can easily be constructed by randomly shifting small amounts of wire density between links (and to link vacancies), which are accepted as long as they (i) leave the network connected and (ii) leave the amount of wire used constant within a certain tolerance interval.

As a reference point for comparison below it is of interest to understand the architecture of such randomized networks. As connections between nodes are random, they have binomial degree distributions. The distribution of link weights is centred around a mean with steeply decaying tails towards much larger or much smaller weights. Further, link length distributions are exponential and, by construction, link weight is independent of link length.

In Figure 4 some network statistics for typical situations for low, intermediate and large β are displayed. The top left panel gives the distribution of link lengths in the optimized networks. Even though the system size is relatively small, it is apparent that the optimal link arrangements for strong and intermediate spatial constraint are characterized by power law tails $P(l) \propto l^{-\alpha}$ in the link length distribution. Best fits yield $\alpha = 1.23 \pm 0.02$ for $\beta = 0.99$ and $\alpha = 1.15 \pm 0.02$ for $\beta = 0.5$ and the organization is thus clearly distinct from the random null model.

The decay of the tails with growing link size becomes steeper, the more emphasis is put on link economy. In contrast, when spatial contraints play only a minor role for $\beta = 0.005$, the link length distribution is fitted well by an exponential function. This function, however, declines more strongly for large link length than expected from the null model.

Networks in the power law regime are very sparse and not very far from being tree-like, such that the power laws in the link length distributions appear consistent with a hierarchical organization in space (Brede, 2010a). The exponents of the power laws, however, are distinctly smaller than $\alpha = 2$ which has been found to be the optimal arrangement for discrete networks that optimize a trade-off between cost of wire and network distance.

Of interest is also the distribution of weights, cf. figure 4 (top right). Whereas this distribution is only slightly skewed for small β, increasing the cost of wire leads to increasingly more asymmetrical skewed distributions. Finally, for very large β, the distributions become bimodal – strongly spatially constrained synchrony-optimal networks are thus comprised of clearly distinct strong and weak links. How is the arrangement of these links? The answer is already suggested by the network illustrations in figure 3. A more thorough statistical analysis is provided in figure 4 (bottom right), in which we plot the dependence of the average length of a link on its weight. For all situations investigated, low, intermediate and strong spatial constraints, a clear picture emerges. Strong links typically connect spatially close nodes whereas weak links establish long-range connections.

The increasing skewness of the link weight distributions with increasing spatial constraints is also reflected in the distributions of (weighted) degrees $k_{\mathrm{W}}(i) = \sum_j A_{ij}$, cf. figure 4 (bottom left). When spatial constraints only play a small role, the distribution of weighted degrees is very narrow and almost bell-shaped, as one would expect from previous studies of unconstrained networks which have highlighted the important role of in-signal homogeneity for superior synchronization (Motter et al., 2005). With increasing influence of spatial constraints, however, the distribution becomes more and more skewed towards lower degrees and finally becomes bimodal at around $\beta = 0.95$.

Summary and conclusions

In this paper we have explored three scenarios for synchrony-optimal undirected networks subject to a tuneable degree of spatial constraints, which are parametrised by a cost-of-wire parameter β. We started with the model of (Brede, 2010b) of unweighted networks connecting nodes with fixed spatial locations. In this case, over a wide range of constraints β, such networks are characterized by link size arrangements that obey a power law $P(l) \propto l^{-\alpha}$ with an exponent α that becomes larger the stronger the influence of spatial constraints on network formation.

Next, in the same model, we explored optimal network configurations that arise when nodes are free to change their relative arrangement in space during the optimization. Two regimes of optimal networks separated by a sharp transition at some critical trade-off parameter β_c can be distinguished. For low $\beta < \beta_c$, nodes are found to cluster into two spatial groups separated by the maximum spatial distance. Above the critical β, nodes arrange themselves into multiple spatial clusters that conincide with network modules. These findings are of interest, since they point out that network arrangements normally not associated with superior synchronization can become optimal, when spatial constraints are important.

In the third part of the paper we have gone back to the scenario of (Brede, 2010b), but now considered weighted undirected networks. The results essentially corroborate

that spatially constrained synchrony-optimal networks are characterized by power-law link length distributions. However, as the networks are weighted, when spatial constraints are important, a clear separation of strong and weak links emerges as well. We typically find that strong links connect spatially close nodes, whereas weak links establish remote connections.

Acknowledgements

This research was undertaken on the NCI National Facility in Canberra, Australia, which is supported by the Australian Commonwealth Government.

References

Arenas, A., Díaz-Guilera, A., Kurths, J., Moreno, Y., and Zhou, C. (2008). Synchronization in complex networks. *Physics Reports*, 469:93.

Arenas, A., Diaz-Guilera, A., and Perez-Vicente, C. J. (2006). Synchronization reveals topological scales in complex networks. *Physical Review Letters*, 96:114102.

Atay, F. M., Biyikoglu, T., and Jost, J. (2006). Network synchronization: Spectral versus statistical properties. *Physica D*, 224:35–41.

Blekhman, I. (1988). *Synchronization in Science and Technology*. ASME Press, New York.

Brede, M. (2008). Locals vs. global synchronization in networks of non-identical kuramoto oscillators. *European Physics Journal B*, 62:87.

Brede, M. (2010a). Coordinated and uncoordinated optimization of networks. *Physical Review E*, 81:066104.

Brede, M. (2010b). Optimal synchronization in space. *Physical Review E*, 81:025202(R).

Brede, M. (2010c). Optimal synchronization in strongly connected directed networks. *European Physics Journal B*, 74:217.

Brede, M. (2010d). Small worlds in space: Synchronization, spatial and relational modularity. *Europhysics Letters (in press), available online: arXiv:1006.2894v1*.

Chavez, M., Hwang, D.-U., Amann, A., Hentschel, H. G. E., and Boccaletti, S. (2005). Synchronization is enhanced in weighted complex networks. *Physical Review Letters*, 94:218701.

Donetti, L., Hurtado, P. I., and Muñoz, M. A. (2005). Entangled networks, synchronization, and optimal network topology. *Physical Review Letters*, 95:188701.

Duch, J. and Arenas, A. (2005). Community detection in complex networks using extremal optimization. *Physical Review E*, 72:027104.

Girvan, M. and Newman, M. E. J. (2004). Finding and evaluating community structure in networks. *Physical Review E*, 69:026113.

Hwang, D.-U., Chavez, M., Amann, A., and Boccaletti, S. (2005). Synchronization in complex networks with age ordering. *Physical Review Letters*, 94:138701.

Manrubia, S. C., Mikhailov, A. S., and Zanette, D. H. (2004). *Emergence of Dynamical Order. Synchronization Phenomena in Complex Systems*. World Scientific, Singapore.

Mathias, N. and Gopal, V. (2001). Small worlds: How and why. *Physical Review E*, 63:021117.

Motter, A. E., Zhou, C. S., and Kurths, J. (2005). Enhancing complex-network synchronization. *Europhysics Letters*, 69:334–340.

Nishikawa, T. and Motter, A. E. (2006a). Maximum performance at minimum cost in network synchronization. *Physica D*, 224:77–89.

Nishikawa, T. and Motter, A. E. (2006b). Synchronization is optimal in non-diagonalizable networks. *Physical Review E*, 73:065106.

Nishikawa, T. and Motter, A. E. (2010). Resolving the network synchronization landscape: compensatory structures, quantization, and the positive effect of negative interactions. pages e–print arxiv:0909:2874v1.

Pecora, L. M. and Carroll, T. L. (1998). Master stability functions for synchronized coupled systems. *Physical Review Letters*, 80:2109.

Schüz, A. and Braitenberg, V. (2002). The human cortical white matter: Quantitative aspects of cortico-cortical long-range connectivity. In Schüz, A. and Miller, R., editors, *Cortical Areas: Unity and Diversity*, pages 377–385. Taylor & Francis, London.

Sole, R. V. and Ferrer i Cancho, R. (2003). Optimization in complex networks. In *Statistical Mechanics of Complex Networks,Lecture notes in Physics*, pages 114–125. Springer, Berlin.

Viswanathan, G., Buldyrev, S. V., Havlin, S., da Luz, M. G., Raposo, E., and Stanley, H. E. (1999). Optimizing the success of random searches. *Nature*, 401:911–914.

Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393:440–442.

Winfree, A. T. (1980). *The Geometry of Biological Time*. Springer-Verlag, New York.

Yook, S.-H., Jeong, H., and Barabási, A.-L. (2002). Modelling the internet's large-scale topology. *Proc. Natl. Acad. Sci. U.S.A.*, 99:13382.

Zarkesh-Ha, P., Davis, J. A., and D., M. J. (2000). Prediction of net-length distribution for global interconnects in a heterogeneous system-on-a-chip. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 8:649.

Modular Random Boolean Networks

Rodrigo Poblanno-Balp[1,2] and Carlos Gershenson[2,1]

[1] Centro de Ciencias de la Complejidad, C3, México
[2] Instituto de Investigaciones Matemáticas Aplicadas y en Sistemas, UNAM, México
balpeaux@gmail.com, cgg@unam.mx

Extended Abstract

Modularity plays an important role in evolution, for even unicellular organisms have separable functional systems (Wagner et al., 2007) which are relatively autonomous. Modularity allows for changes to occur within modules without propagating to other regions and the combination of modules to explore new functions (Espinoza-Soto and Wagner, 2010).

Random Boolean networks (RBNs) (Kauffman, 1993; Gershenson, 2004) have been a very popular model of genetic regulatory networks for several decades, where the state of N nodes is regulated by the state of K neighbour nodes using randomly generated Boolean lookup tables.

However, most studies consider homogeneous or normal topological connectivities between nodes. Aldana (2003) already studied the effect of a scale-free topology on the dynamics of RBNs. In this work, we study the effect of modularity on the dynamics of RBNs, which has been missing from most RBN studies, in spite of its prevalence in natural systems.

We define a modular RBN (MRBN) as a set of M modules connected by L "weak" links. Each module is a RBN with N nodes and K connections between the N nodes within the module. The total number of nodes N_{TOT} is given by $N \cdot M$, while the total number of connections T is given by $M \cdot (K \cdot N + L)$. The average connections per node K_{TOT} is $\frac{N_{\text{TOT}}}{T}$.

Our preliminary results suggest that, for a broad range of values of K_{TOT}, modularity induces complex dynamics, i.e. closer to the transition between the ordered and dynamic phases, also dubbed "the edge of chaos" (Kauffman, 1993). In terms of sensitivity to initial conditions, trajectories in the state space tend to converge in the ordered phase and to diverge in the chaotic phase. For regular RBNs, it is well known that the transition lies at $K = 2$ (Gershenson, 2004). At this point, trajectories neither converge nor diverge. This represents a balance where information can be stored (chaotic phase is too dynamic) as well as modified (ordered phase is too static). However, this behaviour is observed only in a small region of possibilities in unstructured, regular RBNs. Modularity broadens this region considerably, reducing the sensitivity to initial conditions for values of $K_{\text{TOT}} > 2$. Keeping N_{TOT} constant, the number of attractors grows as M grows, although the lengths of these attractors tend to decrease. The highest percentage of states in attractors is given when $N = M$.

We defend that modularity plays an important role in RBNs, as it constrains the topology in such a way that damage is not fully spread across the modular network. Thus, modularity reduces chaos and is desirable for evolvability. It is clear that there is a considerable dynamical difference between modular and regular topologies. Since most studies of RBNs have been made with regular topologies, their results have to be reconsidered in the light of the new evidence, given the fact that real genetic regulatory networks are modular (Segal et al., 2003; Callebaut and Rasskin-Gutman, 2005; Schlosser and Wagner, 2004).

References

Aldana, M. (2003). Boolean dynamics of networks with scale-free topology. *Physica D*, 185:45–66.

Callebaut, W. and Rasskin-Gutman, D. (2005). *Modularity: Understanding the Development and Evolution of Natural Complex Systems*. MIT Press.

Espinoza-Soto, C. and Wagner, A. (2010). Specialization can drive the evolution of modularity. *PLoS Computational Biology*, 6 issue 3.

Gershenson, C. (2004). Introduction to random Boolean networks. In Bedau, M., Husbands, P., Hutton, T., Kumar, S., and Suzuki, H., editors, *Workshop and Tutorial Proceedings, Ninth International Conference on the Simulation and Synthesis of Living Systems (ALife IX)*, pages 160–173, Boston, MA.

Kauffman, S. A. (1993). *The Origins of Order*. Oxford University Press.

Schlosser, G. and Wagner, G. P. (2004). *Modularity in Development and Evolution*. The University of Chicago Press.

Segal, E., Shapira, M., Regev, A., Pe'er, D., Botstein, D., Koller, D., and Friedman, N. (2003). Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics*, 34:166–176.

Wagner, G., Pavlicev, M., and Cheverud, J. M. (2007). The road to modularity. *Nature Reviews*, 8.

A Fisher Information Study of Phase Transitions in Random Boolean Networks

X. Rosalind Wang[1], Joseph T. Lizier[1,2] and Mikhail Prokopenko[1]

[1]CSIRO Information and Communications Technology Centre, PO Box 76, Epping, NSW 1710, Australia
[2]School of Information Technologies, The University of Sydney, NSW 2006, Australia
Rosalind.Wang@csiro.au

Abstract

We study the order-chaos phase transition in random Boolean networks (RBNs), which have been used as models of gene regulatory networks. In particular we seek to characterise the phase diagram in information-theoretic terms, focussing on the effect of the control parameters (activity level and connectivity). Fisher information, which measures how much system dynamics reveal about its parameters, offers a natural interpretation of the phase diagram in RBNs. We report that this measure is maximised near the critical state in the order-chaos phase transitions in RBNs, since this is the region where the system is most sensitive to its parameters. Furthermore, we use this study of RBNs to clarify the relationship between Shannon and Fisher information measures.

Introduction

Random Boolean Networks (RBNs) (Kauffman, 1993) have typically been used by Artificial Life researchers as discrete dynamical network models (e.g., models of Gene Regulatory Networks) with a large sample space available. In particular, RBNs exhibit a well-known phase transition from ordered to chaotic dynamics, with respect to average connectivity or activity level.

Recently, there have been several attempts to study the order-chaos phase transitions of RBNs using information theory.[1] Ribeiro et al. (2008) measured mutual information within random node pairs as a function of connectivity in the network, finding a maximum near the critical point. Rämö et al. (2007) measured the uncertainty (entropy) in the size of perturbation avalanches as a function of an order parameter, and also found a maximum near the critical point. Lizier et al. (2008a) studied the information storage and transfer components of the computation conducted by each node in RBNs. The authors found maxima of these computational quantities just inside the ordered and chaotic sides of the critical point respectively.

While all of these studies provide useful findings regarding the nature of the phase transition, none provide a generic measure that can directly, reliably, and information-theoretically locate the critical point in other systems. For example, the study of perturbation avalanches in (Rämö et al., 2007) is not applicable to systems in which we cannot interfere. The measure of pairwise mutual information (Ribeiro et al., 2008) can be imagined to be maximised for trivial short-periodic behaviour as well as complex behaviour at critical point. And while our previous work (Lizier et al., 2008a) certainly characterises how the RBNs' computation is made up of both information storage and transfer, none of the measures of computation examined were maximised precisely at the critical point in finite-sized systems. In this study we aim to provide a preliminary analysis (in the context of RBNs) of a phase diagram in information-theoretic terms, aiming for the analysis to be generically applicable to other phase transitions. The search for generic tools motivates our study and we use information theory that allows us to analyse and compare critical behaviours across different domains.

Phase transitions are often related to symmetry breaking and self-organisation (Polani, 2007). For instance, Jetschke (1989) defines a system as undergoing a self-organising transition if the symmetry group of its dynamics changes to a less symmetrical one (e.g., a subgroup of the original symmetry group). An example may be given by a ferromagnetic system undergoing a second-order phase transition: (i) in the high-temperature phase the system has no net magnetisation, is 'disordered' and has a complete rotational symmetry (isotropy); (ii) at low temperature, the system becomes 'ordered', and the net magnetisation defines a preferred direction in space (anisotropy), breaking rotational symmetry. The low-temperature ordered phase is therefore *less* symmetrical and can be fully described by an order parameter — the magnetisation vector (Parwani, 2001).

In explaining non-equilibrium structures that spontaneously self-organise in nature, Synergetics (Haken, 1983) — a theory of pattern formation in complex systems — also employs order parameters. When energy or matter flows into a system typically describable by many variables, it may move far from equilibrium, approach a threshold (that can

[1]We note the study of entropy and mutual information between node inputs and outputs by Oosawa and Savageau (2002), though this study did not consider the phase transition in RBNs.

be defined in terms of some control parameters, e.g., the strength of interactions within the system, or the correlation length), and undergo a phase transition. At this stage, the behaviour of the overall system can be described by only a few order parameters (degrees of freedom) that characterise newly formed patterns. In other words, the system becomes low-dimensional as some dominant variables "enslave" others, making the whole system act in synchrony. By varying control parameters (e.g., the strength of interactions within the system) one may trigger phase transitions.

At this stage we would like to highlight the role of (Shannon) information: "a macroscopic description allows an enormous compression of information so that we are no more concerned with the individual microscopic data but rather with global properties" (Haken, 2006). A canonical example is a laser: a beam of coherent light created out of the chaotic movement of particles. Rather than using a large amount of information describing the states of individual atoms, only a single quantity (e.g., the phase of the total light field) is needed, achieving compression of information. Hence, a consensus is reached among the individual parts of the system, indicated by the compression of information, and only one or a few variables have to be guided or controlled (Prokopenko, 2009). In addition, in a vicinity of phase transitions, the information of the order parameters changes dramatically whereas the information of the enslaved modes does not (Haken, 2006).

These insights suggest the use of Fisher information (Fisher, 1922), which measures the amount of information that an observable random variable carries about an unknown parameter. Intuitively, if this unknown parameter can be estimated well using the observable random variable, then Fisher information carried by these observations with respect to this parameter must be high. Otherwise, if the parameter cannot be well-estimated using the observations, the corresponding Fisher information must be low. The application of Fisher information to measure the information that system dynamics contain about control parameters during a phase transition is quite natural. One could expect this quantity to be maximised near the critical point where system dynamics are most sensitive to control parameters.

Our main goal then is to obtain a phase diagram of RBNs in information-theoretic terms using Fisher information. Furthermore, since some studies of Fisher information discuss its connections to (derivatives of) Shannon information, we intend to clarify the relationship between Shannon and Fisher information, using RBNs.

We begin this paper with overviews of RBNs, Fisher information and Shannon information. This is followed by a discussion of how to apply Fisher information to RBNs. We then present the phase diagram of RBNs in terms of Fisher information about the control parameter, demonstrating that this quantity is indeed maximised near the critical point in the order-chaos phase transition in RBNs. Finally, we provide quantitative clarification regarding the relationship between Fisher and Shannon information measures using RBNs as an example.

Random Boolean Networks

Random Boolean Networks is a class of generic discrete dynamical network models. They are particularly important in artificial life, since they were proposed as models of gene regulatory networks by Kauffman (1993). See also Gershenson (2004a) for another thorough introduction to RBNs.

An RBN consists of N nodes in a directed *network*. The nodes take *boolean* state values, and update their state values in time as a function of the state values of the nodes from which they have incoming links. The network topology (i.e. the adjacency matrix) is determined at *random*, subject to whether the in-degree for each node is constant or stochastically determined given an average in-degree $\overline{K}$ (giving a Poisson distribution). It is also possible to bias the network structure, e.g., toward scale-free degree distribution (Aldana, 2003). Given the topology, the deterministic boolean function or lookup table by which each node computes its next state from its neighbours is also decided at *random* for each node, subject to a probability r of producing outputs of "1" (the *bias*). Note that r close to 1 or 0 gives low activity, whereas r close to 0.5 gives the highest activity for any $\overline{K}$. The nodes here are heterogeneous agents: there is no spatial pattern to the network structure (indeed there is no inherent concept of locality), nor do the nodes have the same update functions. (Though, of course either of these can arise at random). Importantly, the network structure and update functions for each node are held static in time ("quenched"). In classical RBNs (CRBNs), the nodes all update their states synchronously.[2]

The synchronous nature of CRBNs, their boolean states and deterministic update functions give rise to a global state space for the network as a whole with deterministic transient trajectories ultimately leading to either fixed or periodic attractors in finite-sized networks (Wuensche, 1997). Effectively, the transient is the period in which the network is *computing* its steady state attractor.

RBNs are known to exhibit three distinct phases of dynamics, depending on their parameters: ordered, chaotic and critical. At relatively low connectivity (i.e., low degree K) or activity (i.e., r close to 0 or 1), the network is in an ordered phase, characterised by high regularity of states and strong convergence of similar global states in state space. Alternatively, at relatively high connectivity and activity, the network is in a chaotic phase, characterised by low regularity

[2]While there has been some debate about the best updating scheme to model GRNs (Darabos et al., 2007), the relevant phase transitions are known to exist in all updating schemes, and their properties depend more on the network size than on the updating scheme (Gershenson, 2004b). As such, the use of CRBNs is justified for ensemble studies such as ours (Gershenson, 2004c).

of states and divergence of similar global states. In the critical phase (the *edge of chaos* (Langton, 1990)), there is percolation in nodes remaining static or updating their values, and uncertainty in the convergence or divergence of similar macro states. This phase transition is typically quantified using a measure of sensitivity to initial conditions, or damage spreading. Following Gershenson (2004c), we take a random initial state A of the network, invert the value of a single node to produce state B, then run both A and B for many time steps (enough to reach an attractor is most appropriate). We then use the Hamming distance:

$$D(A,B) = \frac{1}{N}\sum_{i=1}^{N}|a_i - b_i|, \quad (1)$$

between A and B at their initial and final states to obtain a convergence/divergence parameter δ:

$$\delta = D(A,B)_{t\to\infty} - D(A,B)_{t=0}. \quad (2)$$

(Note $D(A,B)_{t=0} = 1/N$). Finding $\delta < 0$, implies the convergence of similar initial states, while $\delta > 0$ implies their divergence.[3] For fixed r, the critical value of $\overline{K}$ between the ordered and chaotic phases is (Derrida and Pomeau, 1986):

$$K_c = \frac{1}{2r(1-r)}. \quad (3)$$

For finite-sized networks the standard deviation of δ peaks slightly inside the chaotic regime, indicating the widest diversity of networks for those parameters (Gershenson, 2004b). Indeed, the standard deviation is used as a guide to the relative regions of dynamics in finite-sized networks by Rämö et al. (2007), and the indicated shift of the critical point towards the chaotic regime at these finite sizes is reflected by other measures, e.g. (Ribeiro et al., 2008).

Much has been speculated on the possibility that gene regulatory and other biological networks function in (or evolve to) the critical regime (see Gershenson (2004a)). It has been suggested that computation occurs more naturally with the balance of order and chaos there (Langton, 1990), possibly with information storage, propagation and processing capabilities maximised (Kauffman, 1993). Indeed, our previous work has indicated that both information storage and coherent (single-source) information transfer are maximised near the critical state, just within the ordered and chaotic regimes respectively (Lizier et al., 2008a). Because of the importance of the critical state, identifying its precise location is a crucial task, particularly in other systems where analytical solutions are not possible. We look to information theory to address this question.

[3]Typically an order parameter is 1 in the extreme ordered phase, and 0 in the extreme disordered phase. Here, δ is a proxy to this, with negative values representing the ordered phase and positive values representing the chaotic phase.

Fisher Information

Information theory (MacKay, 2003) is an increasingly popular framework for the study of complex systems and their phase transitions (Prokopenko et al., 2009). In part, this is because complex systems can be viewed as distributed computing systems, and information theory is a natural way to study computation, e.g. Lizier et al. (2008b). Information theory is applicable to any system, provided that one can define probability distribution functions for its states. This is a particularly important characteristic since it means that information-theoretic insights can be directly compared across different system types. It is for these reasons that we seek an information-theoretic characterisation of the phase transition in RBNs.

Fisher information (Fisher, 1922) is a way of measuring the amount of information that an observable random variable X has about an unknown parameter θ, upon which the likelihood function of θ depends. Let $p(x|\theta)$ be the likelihood function of θ given the observations x. Then, Fisher information can be written as:

$$F(\theta) = \int_x \left(\frac{\partial \ln(p(x|\theta))}{\partial\theta}\right)^2 p(x|\theta)dx, \quad (4)$$

where $\ln(p(x|\theta))$ is the log-likelihood of θ given x. Thus, Fisher information is not a function of a particular observation, since the random variable X has been averaged out.

Fisher information can be reduced to:

$$F(\theta) = -\int_x \left(\frac{\partial^2 \ln(p(x|\theta))}{\partial\theta^2}\right) p(x|\theta)dx, \quad (5)$$

if $\ln(p(x|\theta))$ is twice differentiable with respect to θ and if the regularity condition:

$$\int \frac{\partial^2}{\partial\theta^2} p(x|\theta)dx = 0 \quad (6)$$

holds. In this paper we use Equation 4, since the regularity condition (Equation 6) does not necessarily hold.

The discrete form of Fisher information is:

$$F(\theta) = \sum_{x_j} p_{x_j}\left(\frac{\Delta \ln(p_{x_j})}{\Delta\theta}\right)^2, \quad (7)$$

where $\Delta\ln(p_{x_j}) = \ln(p'_{x_j}) - \ln(p_{x_j})$ and $p_{x_j} = p(x_j|\theta)$, $p'_{x_j} = p(x_j|\theta+\Delta\theta)$. In this case, $p(x)$ is a discrete probability distribution function, such that $x \in \{x_1, \ldots, x_D\}$, where D is the number of states for the variable X. For example, for a boolean network, $x \in \{0,1\}$.

Fisher information has been extensively used in many fields of science. Frank (2009) argued that Fisher information may be used as the intrinsic metric of natural selection and evolutionary dynamics. Brunel and Nadal (1998) showed that in the context of neural coding, the mutual information between stimuli applied to neurons and neuronal

activity can be characterised by Fisher information. In computer science, Ganguli et al. (2008) studied short term memory in discrete time neural networks by using a criterion based on Fisher information.

We are interested in two aspects of Fisher information. Firstly, it is a measure of the ability to estimate a parameter, making it an important aspect of parameter estimation in statistics (Frieden, 1998). Secondly, it is related to the fundamental quantity of information theory, Shannon information that measures system's uncertainty.

Shannon Information

Shannon Information (Shannon, 1948) was originally developed for reliable transmission of information from a source X to a receiver Y over noisy communication channels. Put simply, it addresses the question of "how can we achieve perfect communication over an imperfect, noisy communication channel?" (MacKay, 2003). When dealing with outcomes of imperfect probabilistic processes, it is useful to define the information content of an outcome x, which has the probability $P(x)$, as $\log_2 \frac{1}{P(x)}$. Crucially, improbable outcomes convey more information than probable outcomes. Given a probability distribution P over the outcomes $x \in \mathcal{X}$ (a discrete random variable X representing the process, and defined by the probabilities $P(x) \equiv P(X = x)$ given for all $x \in \mathcal{X}$), the average Shannon information content of an outcome is determined by

$$H(X) = -\sum_{x \in \mathcal{X}} P(x) \log_2 P(x), \tag{8}$$

We note the information is measured in bits, and henceforth omit the logarithm base 2. This quantity is known as *(information) entropy*, and may be contrasted with Fisher information in Equation 7.

Intuitively, Shannon information measures the amount of freedom of choice (or the degree of randomness) contained in the process — a process with many possible outcomes has high entropy. This measure has some unique properties that make it specifically suitable for measuring "how much 'choice' is involved in the selection of the event or of how uncertain we are of the outcome?" (Shannon, 1948). In answering this question, Shannon suggested the entropy function $-k \sum_{i=1}^{n} P(x_i) \log P(x_i)$, where a positive constant k represents a unit of measure.

In this paper we consider the entropy defined in terms of the probability distribution of the states of each node with respect to some parameter θ.[4] Here the probabilities $p(x_j^i|\theta)$ are defined for each possible state x_j^i for each node i (given θ), and Shannon entropy

$$H(X^i|\theta) = -\sum_{x_j} p(x_j^i|\theta) \log p(x_j^i|\theta)$$

is subsequently also defined for each node i given θ, measuring the diversity of system's states. Then this quantity is averaged across the network given θ,

$$H(\theta)_{RBN} \triangleq \langle H(X^i|\theta) \rangle_i. \tag{9}$$

Fisher Information for RBNs

We aim to study Fisher information $F(r)$ in RBNs as a function of the probability r of each node producing an output of "1". When changing r, the total number of 1s and 0s in the logic tables (which each node uses to compute its next state from its neighbours) would change. So when we calculate $p(x|r)$ and $p(x|r + \Delta r)$ for each r, some nodes in the network with $\theta = r + \Delta r$ would have different logic tables. Therefore, we will produce two sets of results when calculating Fisher information: one where we take into account all the nodes in the network, and one where we ignore those nodes that have their logic table changed. This will allow us to see whether the changes in dynamics are mostly constrained to the nodes whose logic tables have changed, or whether the alterations to their logic genuinely cause changes to the dynamics of the *whole* network and allow insights into r from across the network.

To find Fisher information for the networks, Equation 7 is used since the RBN has nodes with discrete states 0 and 1. If we applied this equation to the RBN as a whole, the likelihood function $p(\mathbf{x}|r)$ is a joint distribution over all nodes $\mathbf{X}$ in the network. This means that for an RBN of 100 nodes, there are 2^{100} possible joint states, which makes a calculation of Fisher information for the joint state of the RBN impractical. Furthermore, since the RBN is not a directed acyclic graph, and its nodes are not independent and identically distributed (i.i.d.), we can not write the likelihood function as a product of the individual nodes. An alternative would be to apply Equation 7 to the single node states x, computing the $p(x|r)$ by combining observations of all nodes in the RBN. This is undesirable though, since the nodes are heterogeneous agents with very different dynamics. Instead, we chose to study the average Fisher information of the individual nodes:

$$F(r)_{RBN} \triangleq \langle F_i(r) \rangle \tag{10}$$

where $F_i(r)$ is the Fisher information of the i-th node of the RBN calculated using Equation 7.

We model the RBNs using enhancements to Gershenson's RBNLab software (http://rbn.sourceforge.net). When approximating an infinitely-sized network with a finite one, the risk is to run the dynamics for too many time steps and reach a periodic or fixed attractor (inevitable for finite-sized

[4] We note the alternative view used elsewhere of information in networks as that contained in the degree distribution amongst nodes (Solé and Valverde, 2004; Bianconi, 2008; Piraveenan et al., 2009).

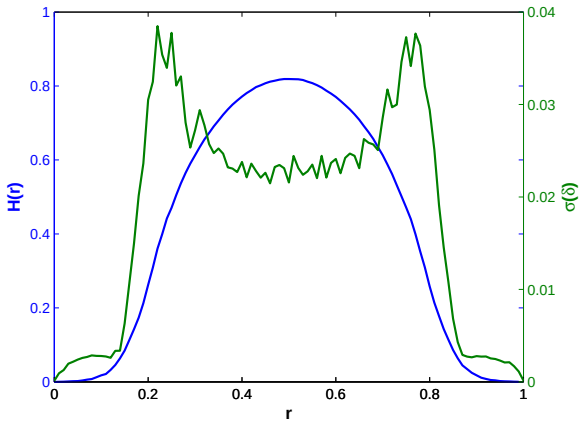


Figure 1: (blue) Average Shannon information $H(r)$ and (green) standard deviation of the convergence/divergence parameter δ versus the bias of the network r. The RBNs here have network size $N = 250$, and average network connectivity $\overline{K} = 4.0$.

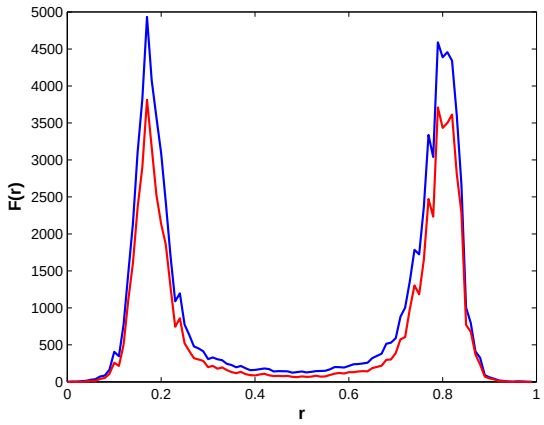


Figure 2: Average Fisher information $F(r)$ versus the bias of the network, r, for networks of size $N = 250$ and average connectivity $\overline{K} = 4.0$. The blue curve shows the Fisher information if we take into account all the nodes in the network, the red curve shows the Fisher information if we ignore those nodes whose logic table has changed due to the change in parameter r.

RBNs). In order to avoid this, for each simulation run starting from an initial randomised state, we ignore a short initial transient of 30 steps to allow the network to settle into the main phase of the computation, and then stop the computation after 400 time steps.

In order to properly sample the dynamics of each node in each RBN and generate enough data for the information theoretic calculations, many repeat runs from random initial states are required for each network (250 were used).

We thus calculate $p(x^i|r)$ of each node i in a given RBN over all the repeat runs. This likelihood of each node is used to calculate the Fisher information at node i, thus giving us the average Fisher information of the network, $F(r) = \langle F(r)_{RBN} \rangle$. Similarly, we averaged the entropy measurements $H(r) = \langle H(r)_{RBN} \rangle$ over the network realisations for each r.

It should be noted, that for many nodes, it often happens that p_x and/or $p'_x = 0$ because a node may exhibit either all 0s or all 1s, especially when r of the network is heavily biased towards 0 or 1. In these cases, if $p_{x_j} = 0$, we set the corresponding individual terms in Equation 7 $p_{x_j} \left(\frac{\Delta \ln(p_{x_j})}{\Delta r} \right)^2 = 0$, where j is the state of the node i. If $p'_{x_j} = 0$, we write the respective terms as (Frank, 2009):

$$p_{x_j} \left(\frac{\Delta \ln(p_{x_j})}{\Delta r} \right)^2 = \frac{1}{p_{x_j}} (p'_{x_j} - p_{x_j})^2,$$

yielding: $p_{x_j} \left(\frac{\Delta \ln(p_{x_j})}{\Delta r} \right)^2 = p_{x_j}$.

Results and Discussions

We focus on RBNs with $N = 250$ nodes and average connectivity of $\overline{K} = 4.0$, while altering the bias in the network r. $\overline{K} = 4.0$ was chosen because, with it held constant, RBNs at low and high values of r exhibit ordered behaviour and RBNs at mid-range values of r exhibit chaotic behaviour.

Figure 1 shows two baseline measures for studying the phase transition. The green curve shows the standard deviation of the convergence/divergence parameter δ as it changes over r. As discussed earlier, this is a typical parameter used to study this phase transition, and the standard deviation is known to reflect the shift of the edge of chaos in finite-sized networks. We can see that there are two separate peaks in this curve, representing the edge of chaos for this finite-sized RBN. This is expected, since the probability distribution function is symmetrical about $r = 0.5$, where there is no bias between choosing a 0 or a 1. These two peaks occur at $r = 0.22$ and 0.77, which as expected are 'inside' the theoretical edge of chaos of an infinite-sized RBN at $r = 0.147$ and 0.853 as found using Equation 3. The blue curve shows the average Shannon information $H(r)$ through this phase transition. $H(r)$ exhibits a bell shaped curve with maximum at $r = 0.5$; this is as expected since the level of activity in the network should be maximum when there is no bias. This result aligns with the previous study of the entropy through the phase transition in RBNs as a function of $\overline{K}$ while holding r constant (Lizier et al., 2008a) .

Now, we examine the phase transition with respect to r using Fisher information $F(r)$. Figure 2 shows the average $F(r)$ calculated in two scenarios: the blue curve shows $F(r)$

when we take into account observations from all nodes in the network, the red curve shows $F(r)$ when we ignore observations from those nodes that have their logic table changed from $p(x|r)$ to $p(x|r + \Delta r)$.

It can be seen from this plot that $F(r)$ has two peaks almost mirrored about $r = 0.5$. These peaks occur approximately at the phase transition between the chaotic phase and the ordered phase for RBN with $\overline{K} = 4.0$ as shown in Figure 2, while $F(r)$ away from the phase transition r has values at least one order of magnitude smaller than the peaks. This indicates that close to the phase transition, there is a large increase in the information in the state distribution of the nodes about the parameter r. On the other hand, deep inside the ordered and chaotic phases, the state distribution of the nodes indicates little about r, other than that the network is in one of these phases.

Certainly the blue curve for $F(r)$ is consistently higher. This curve includes Fisher information from the nodes whose logic tables were changed, and these nodes obviously carry a significant amount of information about the r parameter. Crucially though, there is little difference between the two curves, and both have peaks at $r = 0.17$ and $r = 0.79$. Were the curves identical, this would imply the amount of information about r in the changed nodes did not differ from that in the unchanged nodes, and the average $F(r)$ was not affected. A small quantitative difference indicates that the nodes with changed logic tables retain more information about r. Nevertheless, the information diffuses through the whole network, making the curves quite similar here.

Some studies on Fisher information discuss the relationship between Fisher and Shannon information. Frank (2009) proposed the interpretation that Fisher information is equivalent to the acceleration of Shannon information, i.e. the second derivative of $H(X|\theta)$ with respect to θ. This was shown under the assumption that the outer (or averaging) term $p(x|\theta)$ holds constant while differentiating $H(X|\theta)$, thus differentiating $\log p(x|\theta)$ only. The equivalence between Fisher and acceleration of Shannon information also requires that the regularity condition in Equation 6 holds. However, this is not always the case, and here we now describe our observation of more similarity between Fisher information and *first* derivative of Shannon information.

Figure 3 shows the derivatives of Shannon information $H(r)$ versus network bias r for RBNs with average connectivity of $\overline{K} = 4.0$: the square of the first derivative of Shannon information, $(\frac{d}{dr}H)^2$, is shown in blue and the second derivative, $\frac{d^2}{dr^2}H$, is shown in green. In comparison with Figure 2, we can see that Fisher information for RBNs is more qualitatively similar in shape to the square of rate of change of Shannon information than the acceleration of Shannon information. However, there is a difference in their orders of magnitude, an explanation for which is presented in the Appendix. In general, this is because in finding $F(\theta)$, we first differentiate and then square and average the values, while for $(\frac{dH}{d\theta})^2$ we average and then differentiate and square the values. Furthermore, the peaks for $(\frac{dH}{d\theta})^2$ occur at $r = 0.21$ and $r = 0.79$, coinciding with the Fisher information peaks shown in Figure 2. This shows that for RBNs, the regularity condition of Equation 6 does not hold, and Fisher information is not equivalent to the acceleration of Shannon entropy.

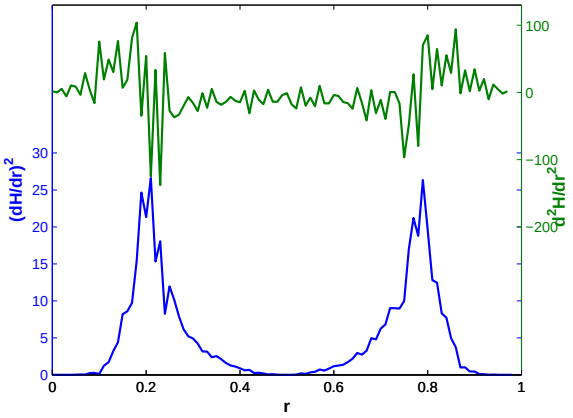


Figure 3: Derivatives of Shannon Information, $H(r)$, for the same networks as Figure 2 ($\overline{K} = 4.0$). (blue) First derivative of H squared. (green) Second derivative of H.

Let r_{max} denote the maximum Fisher information that occurs with respect to r for fixed $\overline{K}$. Formally, r_{max} for every $\overline{K}$ is set to the global maxima of $F(r)$ in two regions: $0 \leq r \leq 0.5$ and $0.5 \leq r \leq 1$. For example, r_{max} correspond to the peaks shown in Figure 2. We now examine the values of r_{max} as a function of $\overline{K}$. To reiterate, each $F(r)$ is an average of Fisher information $F(r)_{RBN}$ over 250 networks, yielding r_{max} values for both regions. Repeating the experiment 10 times with different 250 networks allows us to average these r_{max} values over 10 runs. Figure 4 shows the plot of r_{max} versus $\overline{K}$ for $\overline{K} = 2.0$ to $\overline{K} = 10.0$. The blue curve shows the r_{max} computed over all nodes in the network, and the red curve corresponds to the case when those nodes that have their logic table changed were ignored. As we can see from the figure, there is very little difference between the two r_{max} curves. In alignment with the findings for Figure 2, we see that the changes to the logic tables of a few nodes genuinely cause the effect of changes in r to diffuse throughout the network.

The green dashed curve in Figure 4 shows the theoretical critical phase (edge of chaos) of the RBNs, generated using Equation 3. We can see from the figure that the phase diagram obtained by maximising Fisher information generally follows the same shape, but is bounded by the theoretical curve for critical K_c versus r. This is because the theoretical curve corresponds to an RBN with an infinite size, while the phase diagram based on the maximum Fisher information is

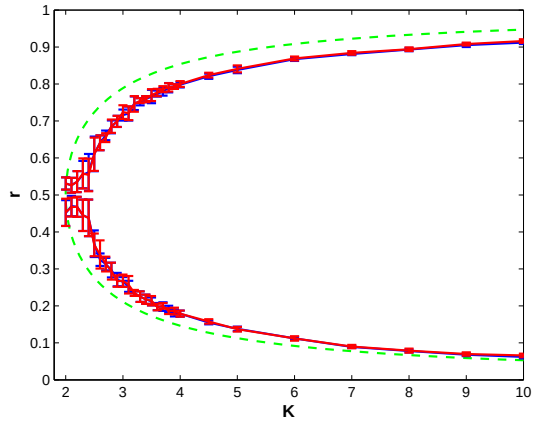


Figure 4: Phase diagram of r_{max} where the maximum Fisher information, $F(r)$, occurs with respect to r for fixed $\overline{K}$, as a function of $\overline{K}$. Blue: when all the nodes in the network were taken into account; Red: when those nodes whose logic table has changed due to the change in parameter r were ignored. The error bars on the curves show the standard deviation of r_{max}. The green dashed line is the theoretical curve for critical K_c versus r.

for a finite size RBN. As pointed out previously, for finite-size networks the critical point is known to shift towards the theoretically chaotic region, and the maximum Fisher information certainly reflects this.

Indeed, these finite-size effects also partly explain why the loci of the divergent maxima of Fisher information do not meet as $K \to 2$. For $r = 0.5$, the phase transition with respect to K shifts towards the chaotic regime at around $K \approx 2.5$ in these finite size RBNs rather than the theoretical 2.0. Our experimental curve(s) should converge/diverge at around $K \approx 2.5$. The fact that they do not converge is an artifact of our explicit search for two maximum values of $F(r)$ for $0 \leq r \leq 0.5$ and $0.5 \leq r \leq 1$.

Conclusion

In this paper, we contrasted Fisher information and Shannon information in the context of Random Boolean Networks (RBNs). RBNs are known to exhibit three distinct phases of dynamics, depending on their parameters: ordered, chaotic and critical, and we analysed the phase diagram of RBN dynamics interpreted in information-theoretic terms.

Both the activity level r and average connectivity K play the role of control parameters, and the phase diagram is obtained by plotting (K, r) points that separate the ordered and chaotic phases. If δ was used as a proxy to an order parameter, the critical (K, r) points are those where δ changes sign. Information-theoretically, Shannon information $H(r)$ which measures (globally) the diversity of RBN's states given the parameter r, is minimal in the ordered phase and maximal in the chaotic phase. However, it does not identify the precise location of the critical points. On the other hand, Fisher information about the control parameters has maxima at the critical (K, r) points. This is because $F(r)$ measures (locally) the amount of information that RBN dynamics carry about the parameter r, and these dynamics are most sensitive to the control parameter near the critical point.

Our analysis showed that an information-theoretic interpretation of the phase diagram (K with respect to r) reveals expected phases (ordered, chaotic and critical) as well as symmetry breaking (slightly obscured by finite-size effects). In addition, the comparison between Fisher information $F(r)$ and a square of a first derivative of Shannon information $H(r)$ uncovered their strong qualitative similarity, albeit separated by an order of magnitude. The analysis shed more light on connections between Fisher information and (derivatives of) Shannon information, and provided a means for further rigorous information-theoretic studies of phase transitions in complex networks.

Acknowledgements

We like to thank the High Performance Computing and Communications Centre (http://www.hpccc.gov.au/) for the use of their supercomputer clusters in performing the experiments for this paper.

Appendix

It can be seen from Figure 2 and 3 that the magnitude of the $F(r)$ is much higher than $(\frac{dH}{dr})^2$, in fact, the peak for $F(r)$ is approximately 200 times that of $(\frac{dH}{dr})^2$'s peak. This is due to the order of averaging and differential in the two calculations. To illustrate this, let us take one simple example, where the variable x has two states $\{0, 1\}$ the probabilities of which depend on some parameter θ:

$$\begin{aligned} \text{Let}: \quad & p(0|\theta) = 0.5 && p(1|\theta) = 0.5 \\ & p(0|\theta + \Delta\theta) = 0.3 && p(1|\theta + \Delta\theta) = 0.7 \\ & \Delta\theta = 0.01 \end{aligned}$$

Now, using Equation 8, we can find the Shannon information:

$$H(X|\theta) = -(0.5 \log_2 0.5 + 0.5 \log_2 0.5) = 1,$$
$$H(X|\theta + \Delta\theta) = -(0.3 \log_2 0.3 + 0.7 \log_2 0.7) = 0.8843.$$

Thus, the first derivative squared in this case is:

$$\left(\frac{dH(X|\theta)}{d\theta}\right)^2 = \left(\frac{H(X|\theta + \Delta\theta) - H(X|\theta)}{\Delta\theta}\right)^2 = 133.86.$$

Using Equation 7, we can find the Fisher information:

$$F(\theta) = 0.5 \left(\frac{\ln 0.3 - \ln 0.5}{\Delta\theta}\right)^2 + 0.5 \left(\frac{\ln 0.7 - \ln 0.5}{\Delta\theta}\right)^2$$
$$= \frac{0.5(-0.5108)^2 + 0.5(0.33647)^2}{(0.01)^2} = 6965.$$

Here, we can see that $F(\theta)$ is 50 times larger than $(\frac{dH}{d\theta})^2$. This shows that while at the first glance, the values of $F(\theta)$ and $(\frac{dH}{d\theta})^2$ should be similar, there is actually one to two orders of magnitudes difference between them.

References

Aldana, M. (2003). Boolean dynamics of networks with scale-free topology. *Physica D*, 185(1):45–66.

Bianconi, G. (2008). The entropy of randomized network ensembles. *Europhysics Letters*, 81:28005.

Brunel, N. and Nadal, J.-P. (1998). Mutual information, Fisher information and population coding. *Neural Computation*, 10(7):1731–57.

Darabos, C., Giacobini, M., and Tomassini, M. (2007). Semi-synchronous activation in scale-free boolean networks. In Almeida e Costa, F., Rocha, L. M., Costa, E., Harvey, I., and Coutinho, A., editors, *Proceedings of the 9th European Conference on Artificial Life (ECAL 2007), Lisbon, Portugal*, volume 4648 of *Lecture Notes in Artificial Intelligence*, pages 976–985, Berlin / Heidelberg. Springer.

Derrida, B. and Pomeau, Y. (1986). Random networks of automata: a simple annealed approximation. *Europhysics Letters*, 1(2):45–49.

Fisher, R. A. (1922). On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society, A*, 220:309–368.

Frank, S. A. (2009). Natural selection maximizes Fisher information. *Journal of Evolutionary Biology*, 22:231–244.

Frieden, B. R. (1998). *Physics from Fisher Information*. Cambridge University Press, Cambridge, United Kingdom.

Ganguli, S., Huh, D., and Sompolinsky, H. (2008). Memory traces in dynamical systems. *Proceedings of the National Academy of Sciences*, 105(48):18970–18975.

Gershenson, C. (2004a). Introduction to random boolean networks. In Bedau, M., Husbands, P., Hutton, T., Kumar, S., and Suzuki, H., editors, *Proceedings of the Workshops and Tutorials of the Ninth International Conference on the Simulation and Synthesis of Living Systems (ALife IX), Boston, USA*, pages 160–173.

Gershenson, C. (2004b). Phase transitions in random boolean networks with different updating schemes. arXiv:nlin/0311008.

Gershenson, C. (2004c). Updating schemes in random boolean networks: Do they really matter? In Pollack, J., Bedau, M., Husbands, P., Ikegami, T., and Watson, R. A., editors, *Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems (ALife IX), Boston, USA*, pages 238–243, Cambridge, USA. MIT Press.

Haken, H. (1983). *Synergetics, an Introduction: Nonequilibrium Phase Transitions and Self-Organization in Physics, Chemistry, and Biology*. Springer-Verlag, 3rd rev. enl. ed., New York.

Haken, H. (2006). *Information and Self-Organization: A Macroscopic Approach to Complex Systems*. Springer-Verlag, Berlin Heidelberg.

Jetschke, G. (1989). *Mathematik der Selbstorganisation*. Vieweg, Braunschweig.

Kauffman, S. A. (1993). *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, New York.

Langton, C. G. (1990). Computation at the edge of chaos: phase transitions and emergent computation. *Physica D*, 42(1-3):12–37.

Lizier, J. T., Prokopenko, M., and Zomaya, A. Y. (2008a). The information dynamics of phase transitions in random boolean networks. In Bullock, S., Noble, J., Watson, R., and Bedau, M. A., editors, *Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems (ALife XI), Winchester, UK*, pages 374–381, Cambridge, MA. MIT Press.

Lizier, J. T., Prokopenko, M., and Zomaya, A. Y. (2008b). Local information transfer as a spatiotemporal filter for complex systems. *Physical Review E*, 77(2):026110.

MacKay, D. J. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, Cambridge.

Oosawa, C. and Savageau, M. A. (2002). Effects of alternative connectivity on behavior of randomly constructed boolean networks. *Physica D*, 170(2):143–161.

Parwani, R. R. (2001). Complexity: A web-book. http://staff.science.nus.edu.sg/~parwani/complexity.html.

Piraveenan, M., Prokopenko, M., and Zomaya, A. Y. (2009). Assortativeness and information in scale-free networks. *European Physical Journal B*, 67(3):291–300.

Polani, D. (2007). Foundations and formalizations of self-organization. In Prokopenko, M., editor, *Advances in Applied Self-organizing Systems*, pages 19–37. Springer, London.

Prokopenko, M. (2009). Book review: Information and self-organization: A macroscopic approach to complex systems (H. Haken). *Artificial Life*, 15(3):377–383.

Prokopenko, M., Boschetti, F., and Ryan, A. J. (2009). An information-theoretic primer on complexity, self-organization, and emergence. *Complexity*, 15(1):11–28.

Rämö, P., Kauffman, S., Kesseli, J., and Yli-Harja, O. (2007). Measures for information propagation in boolean networks. *Physica D*, 227(1):100–104.

Ribeiro, A. S., Kauffman, S. A., Lloyd-Price, J., Samuelsson, B., and Socolar, J. E. S. (2008). Mutual information in random boolean models of regulatory networks. *Physical Review E*, 77(1):011901–10.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, July, October.

Solé, R. V. and Valverde, S. (2004). Information theory of complex networks: Onevolution and architectural constraints. In Ben-Naim, E., Frauenfelder, H., and Toroczkai, Z., editors, *Complex Networks*, volume 650 of *Lecture Notes in Physics*, pages 189–207. Springer, Berlin / Heidelberg.

Wuensche, A. (1997). *Attractor Basins of Discrete Networks*. PhD thesis, The University of Sussex.

Evolutionary Selection of Network Structure and Function

Larry Yaeger[1], Olaf Sporns[2], Steven Williams[1], Xin Shuai[1] and Sean Dougherty[3]

[1]School of Informatics and [2]Department of Psychological and Brain Sciences
Indiana University, Bloomington, IN 47408
[3]Open source contributor
larryy_at_indiana_dot_edu

Abstract

We explore the relationship between evolved neural network structure and function, by applying graph theoretical tools to the analysis of the topology of artificial neural networks known to exhibit evolutionary increases in dynamical neural complexity. Our results suggest a synergistic convergence between network structures emerging due to physical constraints, such as wiring length and brain volume, and optimal network topologies evolved purely for function in the absence of physical constraints. We observe increases in clustering coefficients in concert with decreases in path lengths that together produce a driven evolutionary bias towards small-world networks relative to comparable networks in a passive null model. These small-world biases are exhibited during the same periods that evolution actively selects for increasing neural complexity (also during which the model's agents are behaviorally adapting to their environment), thus strengthening the association between small-world network structures and complex neural dynamics.

Introduction

Dynamical processes in networks are unavoidably influenced by the networks' underlying topologies. As the study of networks has come to pervade all of science, a need has arisen to understand this relationship between the anatomical structure of networks and the dynamical functions they carry out (Strogatz, 2001).

Small-world properties have been shown (Watts and Strogatz, 1998) to characterize many networks of interest, including biological nervous systems. Small-world networks of Hodgkin-Huxley neurons have been shown (Lago-Fernández et al., 2000) to provide the best features of both random networks (fast system response) and regular networks (coherent oscillations). Small-world-ness has also been shown (Sporns et al., 2000) to be highly correlated with dynamical complexity in artificial neural networks evolved specifically for complexity. In the biological realm, cortical connection matrices for macaque visual cortex and rat cortex have been shown (Sporns et al., 2000) to exhibit both small world anatomical properties and high dynamical complexity.

It has been argued that physical constraints—evolutionary pressures to reduce overall wiring length (Mitchison, 1991; Cherniak, 1995) and to maximize connectivity while minimizing volume (Murre and Engelhardt, 1995)—might explain key aspects of biological brain connectivity. But it is unlikely that evolutionary pressure on wiring alone is responsible for the detailed patterns of connectivity seen in biological brains (Sporns et al., 2000). Thus one is led to ask how natural selection would act upon the topological characteristics of nervous systems in the absence of physical constraints, and whether such functional evolutionary pressures are opposed to, independent of, or aligned with physical evolutionary pressures.

In previous work using the Polyworld artificial life system (Yaeger, 1994) we have shown that when agents whose behaviors are controlled by a genetically prescribed artificial neural network are subject to natural selection, the networks' dynamical neural complexity increases over evolutionary time (Yaeger and Sporns, 2006), the networks' complexity will be actively selected for by evolution (Yaeger et al., 2008), and periods of neural complexity growth correspond to periods of behavioral adaptation of the agents to their environment (Yaeger, 2009).

We now seek to understand the underlying network topologies that give rise to this evolved functional complexity. Preliminary results for several graph theoretical metrics from one simulation suggested (Lizier et al., 2009) that evolutionary trends in Polyworld mirrored those in biological neural networks (and successfully related anatomical networks to inferred functional networks). We will more fully characterize those evolutionary trends, determine their robustness and statistical significance, quantify the small-world-ness of those trends, and confirm the role of natural selection (as opposed to random drift, in a "driven" vs. "passive" sense (McShea, 1996)) in the shaping of those trends. This allows us to characterize the relationship between evolutionary pressures on brain structure due to functional optimization vs. physical constraints.

Tools and Techniques

Polyworld

Polyworld is an ecosystem model in which the agents are controlled by artificial neural networks using a firing rate neuron model performing Hebbian learning at the synapses. The wiring diagrams of these networks are the primary subject of evolution in the system, through a genetic encoding

of a generative model of network architectures. This genetic encoding describes the network topology in terms of an arbitrary number of neural groups, containing arbitrary numbers of excitatory and inhibitory neurons, wired together with genetically determined connection densities, ordered-ness of connections, and learning rates. By eschewing any particular model of ontogenetic development, Polyworld avoids the biases inherent in such a model choice. Further, instead of evolving specific network topologies, Polyworld forces evolution to select for useful statistics of neural connectivity.

Vision, current energy level, and a randomly firing neuron are the inputs to the network. A suite of primitive behaviors (move, turn, eat, mate, attack, light, focus) are the outputs. All agent actions consume energy, which must be replenished by consuming food from the environment, or by killing and eating other agents. Normally there are per-neuron and per-synapse energy costs, but these have been eliminated for this study so as not to impose any pseudo-physical constraints on network topology. Survival and reproduction, variation and selection, are the only driving forces, so Polyworld acts as a model of natural selection, with no fitness function, rather than in the manner of a genetic algorithm (though that is possible, if desired).

In these experiments Polyworld is used to produce paired runs in which an initial, normal "driven" run is followed by a "passive", null-model run. (The terms driven and passive are used in the sense proposed by McShea (1996).) In the passive run, agents cannot reproduce or die on their own; rather, pairs are chosen for reproduction at random and individuals are killed at random to match the birth and death events of the original driven run, thus removing the effects of selection, while retaining population statistics and levels of genetic variation that are equivalent to those in the driven run. This allows the direct comparison of driven vs. passive, natural-selection vs. random-walk evolutionary trajectories. See (Yaeger et al., 2008; Yaeger, 2009) for more details.

The activation of every neuron at every time step for every agent is recorded to disk as simulations progress, as is the neural architecture of every agent. Thus we are able to study both the structure and the function of the evolved neural networks, under conditions in which either natural selection or increasing variance due to a random walk are holding sway.

The Polyworld source code and data analysis tools are available at http://sourceforge.net/projects/polyworld/ and instructions for installing and building Polyworld are at http://beanblossom.in.us/larryy/BuildingPolyworld.html.

Complexity

Though other measures of complexity are being investigated, our primary tool for analyzing neural dynamics is an information theoretic measure of neural complexity, originally proposed by Tononi et al. (1994), introduced in a simplified and more computationally tractable form in (Tononi et al., 1998), and explored computationally in (Sporns et al., 2000; Lungarella et al., 2005). Referred to throughout as just "complexity" (aka "TSE complexity", for the initials of its inventors), the measure captures a trade-off between integration (cooperation) and segregation (specialization) at multiple scales in any system of random variables, such as the temporal traces of one of our agents' neural activations. Maximally complex networks exhibit a high degree of both integration and segregation at multiple scales. The simplified version of TSE complexity we use is given by:

$$C(X) = H(X) - \sum_{x_i \in X} H(x_i|X - x_i) \qquad (1)$$

where $H(X)$ is the entropy of the entire system and the $H(x_i|X - x_i)$ terms are the conditional entropy of each of the variables x_i given the entropy of the rest of the system.

Graph Theoretical Metrics

For current purposes we are interested primarily in three graph theoretical metrics. Two of them—clustering coefficient and characteristic path length—were used by Watts and Strogatz (1998) to define and characterize small-world networks. The third is a quantitative means of characterizing the degree of small-world-ness exhibited by a network introduced by Humphries et al. (2006). Throughout we will talk about our neural networks as graphs, which can be described by the number of nodes (aka vertices or neurons) and the number of links (aka edges or synapses) that connect them.

Clustering coefficient (CC) is a local measure of cliquishness in a graph, and characterizes the degree to which a node's neighbors are likely to be neighbors of each other (where "neighbor" means a link exists between the nodes). In friend networks this would be the degree to which friends of a common friend are likely to be friends of each other. It is defined at each node as the fraction of possible links between neighbors that are actually present in the graph, and defined for the entire network as the average of this fraction over all nodes in the graph.

Characteristic path length (CPL), also called average shortest path length, is a global measure of the average separation between all node pairs in a graph—an estimate of how far it is from any one node to another. The average distance to all other nodes is calculated for each node, and then averaged over all nodes.

Watts and Strogatz (1998) identified small-world networks by their combination of high clustering and low path length. By contrast, though regular lattice networks also exhibit high clustering, they typically have high path lengths, since any given node must traverse all intervening nodes and links to reach a distant node. And while random graphs tend to have low path lengths, since any given node is only a few random hops away, they usually exhibit low clustering.

Small-world index (SWI) is a quantitative measure of small-world-ness introduced by Humphries et al. (2006). To calculate the SWI of a graph, one computes CC and CPL

for the actual graph, plus CC and CPL for a corresponding random graph (or ensemble of random graphs as done here), and compares the ratios of actual to random measurements, as follows:

$$\gamma = CC/\langle CC_r \rangle \quad (2)$$

$$\lambda = CPL/\langle CPL_r \rangle \quad (3)$$

$$s = \gamma/\lambda \quad (4)$$

where $\langle CC_r \rangle$ and $\langle CPL_r \rangle$ are the ensemble averages of CC and CPL over some number of random graphs having the same number of nodes and edges as the original graph, and s is the desired SWI.[1] SWI captures the degree to which clustering and path length in the actual, original graph vary, in the appropriate directions, from the values seen in comparable random graphs. The more small-world a network is, the greater its SWI will be above 1.0.

These metrics are most frequently applied to undirected graphs (a given edge connects in both directions), often with binary edges (either present or not). However, neural networks importantly have both weighted and directed edges. Fortunately these metrics extend straightforwardly to support the analysis of weighted, directed (WD) graphs, but their application to such networks has been less well characterized than for binary, undirected (BU) graphs and, indeed, there turn out to be some issues applying them to WD graphs. (Such as a greater prevalence of disconnected nodes in WD graphs.) Accordingly, we analyzed our networks treating them both as BU and WD graphs.

Neural network edge weights are also signed—positive for excitatory connections, negative for inhibitory connections. Unfortunately, few graph theoretical metrics extend well to signed graphs. So for these analyses we have made the less than desirable, but simple and common, approximation of using the absolute values of the network weights on the graph edges.

The fact that one of our key metrics, path length, is based on distances between nodes, yet our neural networks have weights, not distances, associated with their connections, presents another small conundrum. We again take the simplest, most common approach, and invert the weights to provide a distance measure. Thus a strong weight, which produces a strong influence, after inversion corresponds to a short distance. So nodes that strongly influence each other are seen as close neighbors, while nodes that only weakly influence each other are seen as distant neighbors, and nodes that do not directly affect each other at all (have zero weight) are infinitely far apart (though they may be reachable indirectly, through other nodes and links). For our other fundamental metric, clustering coefficient, we use the original neural network weights on the edges.

[1]Humphries used a single random graph corresponding to each original graph, but there is sufficient variance in CC and CPL amongst graphs with the same numbers of nodes and links that we have chosen to use ensemble averages instead.

A question also arises as to which neural network nodes to include in the graph being analyzed. One obvious answer is all of them. However, the sensory nodes have an unusual constraint—zero in-degree (no incoming connections)—and their activations are purely determined by what the agent senses in its environment rather than anything that happens within the neural network. Another answer, then, is the non-sensory neurons; i.e., all internal and output/behavioral neurons. In our complexity work we have referred to this set of non-sensory neurons as the "processing" neurons. Accordingly, we have carried out our graph theoretical analyses looking at both cases: all (A) neurons and processing (P) neurons.

Finally, especially early on in our simulations, some of the graphs are quite small and consist of multiple components (disconnected sub-graphs) and even contain disconnected neurons. It turns out that CPL behaves poorly and erratically in this situation. This is due to its treatment of inter-node distances between disconnected nodes as infinite. Thus path lengths are computed only within each disconnected subgraph and the metric can exhibit sudden large changes as subgraphs become connected or disconnected and shortest paths span much larger or smaller subsets of nodes.

A length metric proposed by Marchiori and Latora (2000), connectivity length (CL), uses inverted lengths to calculate the harmonic (rather than arithmetic) mean of average shortest path length, and better handles multiple components and disconnected nodes. However, by effectively including all those infinities (as zeroes), it can compress the distinctions between sparsely connected and disconnected graphs.

We therefore devised, and introduce here, a new length metric, normalized path length (NPL), that appears to be better behaved than either CPL or CL for the class of graphs we are analyzing, though it too has some quirks (a sensitivity to edge weights that makes it somewhat noisy in its WD form).

To calculate NPL, node pairs that have no path between them are assigned a maximum path length l_{max} defined as N/w_{max}, rather than infinity, where N is the number of nodes in the graph and w_{max} is the maximum possible synaptic weight in our neural networks. (For binary networks the greatest possible path length is $N-1$, hence this value of N is one that cannot occur by any means other than disconnection.) Inverting to convert weight to distance, we also define a minimum path length l_{min}, which is just $1/w_{max}$. We then proceed to compute CPL normally, limiting path length to the defined maximum, and normalize first by subtracting the minimum path length and then dividing by the difference between the maximimum and minimum path lengths. Thus, in terms of CPL, NPL may be written as follows:

$$NPL = (CPL^* - l_{min})/(l_{max} - l_{min}) \quad (5)$$

where CPL^* is a normally calculated CPL using l_{max} as the maximum possible distance between nodes. Or expressed in

terms of path lengths:

$$NPL = \frac{\frac{\sum_{\substack{i,j=1 \\ j \neq i}}^{N} min(l_{ij}, l_{max})}{N(N-1)} - l_{min}}{l_{max} - l_{min}} \quad (6)$$

where l_{ij} is the shortest path from node j to node i. NPL is guaranteed to lie between 0.0, for a fully connected graph, and 1.0, for a fully disconnected graph (a collection of nodes with no links between them), and has proven to be well behaved for graphs with multiple components and disconnected nodes (as well as for the more commonly analyzed strongly connected graphs).

Since none of our three length metrics is "perfect" and NPL is entirely new, wherever a length metric is calculated or used, we examine all three, and refer in general to simply path length. Thus for each metric we treat the graph as consisting of either the A neurons or the P neurons and we treat the graph edges as being either BU or WD, and for length metrics we look at each of CPL, CL, and NPL. Different neuron sets, graph types, and length metrics usually agree on common trends, but do sometimes provide different insights into the algorithms and architectures. Unfortunately, due to space constraints we cannot show all variations of all metrics. A complete set of plots of these metrics may be obtained as supplementary material here: http://informatics.indiana.edu/larryy/alife12_sup.zip. The abbreviations defined here (CC, CPL, CL, NPL, SWI, A, P, BU, WD) and another new metric (SWB) defined later are consistently applied in these plots as well as this paper.

All graph theoretical metrics were calculated using our new C++ implementation (bct-cpp) of the Brain Connectivity Toolbox (BCT) MATLAB module (Rubinov and Sporns, 2010). The original BCT may be found at http://www.brain-connectivity-toolbox.net/ and bct-cpp may be found at http://code.google.com/p/bct-cpp/.

Simulations and Data Acquisition

A set of 10 paired simulations, differing only in initial random number seeds, were run in driven and passive modes; i.e., 20 simulations in all. Each simulation ran for 30,000 time steps. Temporal traces of neural activations and structural descriptions of neural anatomies were recorded for all agents. Agents were assigned to temporal bins corresponding to 1,000 time steps, according to the time of their death.

This type of binning was necessary for our complexity studies, since an agent's neural complexity can only be accurately computed after the completion of its neural activation time series—its death. We have retained this binning in our graph theoretical analysis so we can directly compare structural and functional results.

Complexity and graph theoretical metrics were calculated for each agent and averaged to produce a population mean (and standard deviation) in each temporal bin, for each driven and passive run. In addition, for each agent's actual neural network, 10 graphs with an identical node count, edge count, and distribution of weights were generated randomly, and the means of the graph theoretical measures for these networks were used to characterize the structure of a random graph corresponding to each actual graph.

Results and Discussion

Given that we know complexity increases over evolutionary time in Polyworld and is, in fact, actively selected for by evolution under certain conditions, our intention is to develop a better understanding of the structural characteristics that give rise to these complex network dynamics. To this end we start by examining clustering coefficient.

The various neuron sets and graph types tell much the same story for clustering coefficient, as represented by the P,WD results in Figure 1. Initially CC is actively selected for by evolution, as evidenced by the more rapid rate of increase in the driven runs than in the passive runs. But once a "good enough" solution emerges and spreads throughout the population, CC in the passive populations surpasses that in the driven populations. The period during which there exists a statistically significant bias for high CC in the driven runs is from about t=1000 to t=11000. This mimics but extends the trend previously observed in neural complexity (Yaeger et al., 2008), as complexity's period of statistically significant differences lasted only from about t=1000 to t=4000, and passive complexity caught up to driven complexity by about t=7000. The period of behavioral adaptation is approximately t=1000 to t=7000 (Yaeger, 2009).

A traditional means of looking for meaningful graph structure is to compare suitable graph theoretical metrics computed for one's actual graphs to the same metrics calculated for comparable random graphs. We examined driven vs. random and passive vs. random CC, but do not include the results here due to space considerations. CC was substantially and statistically significantly greater in the actual evolved graphs than in the corresponding random graphs. Curiously, this difference was observed in passive vs. random as well as driven vs. random graphs, which we take as a warning that there is a bias present in our genetic encoding mechanism towards at least some degree of clustering. Given that the encoding expresses connectivity between groups of neurons, this seems reasonable. This result suggests that the differences we observe between driven and passive results may be lower than one might find with a completely unbiased encoding scheme. It also means we are probably better off focusing on driven vs. passive results than driven vs. random results, since the passive runs represent a more appropriate and tightly constrained null model than do the random graphs.

Turning to path length, the stories told by NPL and CL are very similar to each other and to that told by CC and

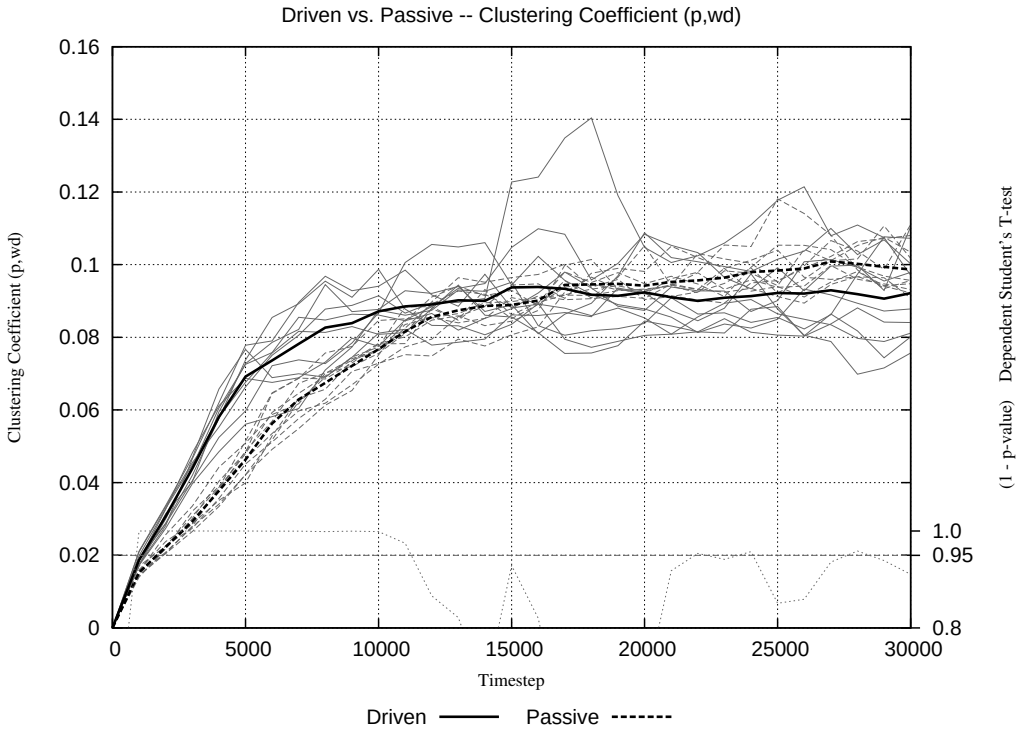


Figure 1: Driven *vs.* passive clustering coefficient as a function of time. Light solid lines show mean population CC for each driven run. Light dashed lines show mean population CC for each passive run. Heavy lines show meta-means of all ten runs for the corresponding line style. Light dotted line at bottom shows dependent $1-$p-value for a Student's T-test with typical $p > 0.05$ statistical significance indicated by the horizontal line at $p = 0.95$.

complexity. CPL is less consistent, due to its previously discussed shortcomings, showing generally the same trends, but without much statistical significance in both WD analyses, large and greatly extended statistical significance in the P,BU analysis, and a result much like the other length metrics in the A,BU analysis. Figure 2, though somewhat noisy, shows the typical trends in path length, using NPL. Path length initially drops much more rapidly in the driven runs than it does in the passive runs, but as that "good enough" solution becomes weakly stabilized in the driven runs, path length in the passive runs drops below that in the driven runs. In fact, path length in the passive runs drops nearly to the level seen in random graphs (not shown). The initial period of driven vs. passive statistical significance is from about t=1000 to t=7000, again corresponding well to the period of complexity growth and behavioral adaptation.

Thus we have seen that during the period of growth in the complexity of the agents' neural dynamics there is a corresponding, statistically significant growth in clustering coefficient and reduction in path length. High clustering coefficient and low path length are the defining characteristics of a small-world network. So our results are suggestive of a selective pressure towards small-world networks, and provide support for a correlation between small-world structure and complex function.

To investigate this trend towards small-world-ness, we turned to the small world index proposed by Humphries et al. (2006). As it was originally formulated, SWI is based on comparing CC and CPL in actual graphs vs. random graphs. However, given the problems previously discussed in applying CPL to our small, sparse, multi-component graphs with disconnected nodes, the standard version of SWI proved to be uninformative, displaying little consistency amongst the different neuron sets and graph types we analyzed and with sufficient noise to render some results uninterpretable. So we developed alternative formulations of SWI, using our better behaved length metrics, CL and NPL. Curiously, some of the inconsistencies were present in these formulations as well.

We could have cherry-picked an SWI result based on NPL for the A neuron set and BU graph type that looks very much like we expected, with a statistically significantly higher growth rate in SWI for the driven runs compared to the passive runs. However, the P,WD version of this metric, even using NPL, actually reverses the roles of driven and passive (in a clear, although not significant fashion). We believe that the small and weakly connected character of our early nets are contributing to these difficulties, which explains why the problems are most exacerbated in the nets with the most limited set of connections (P,WD), but are not entirely satisfied

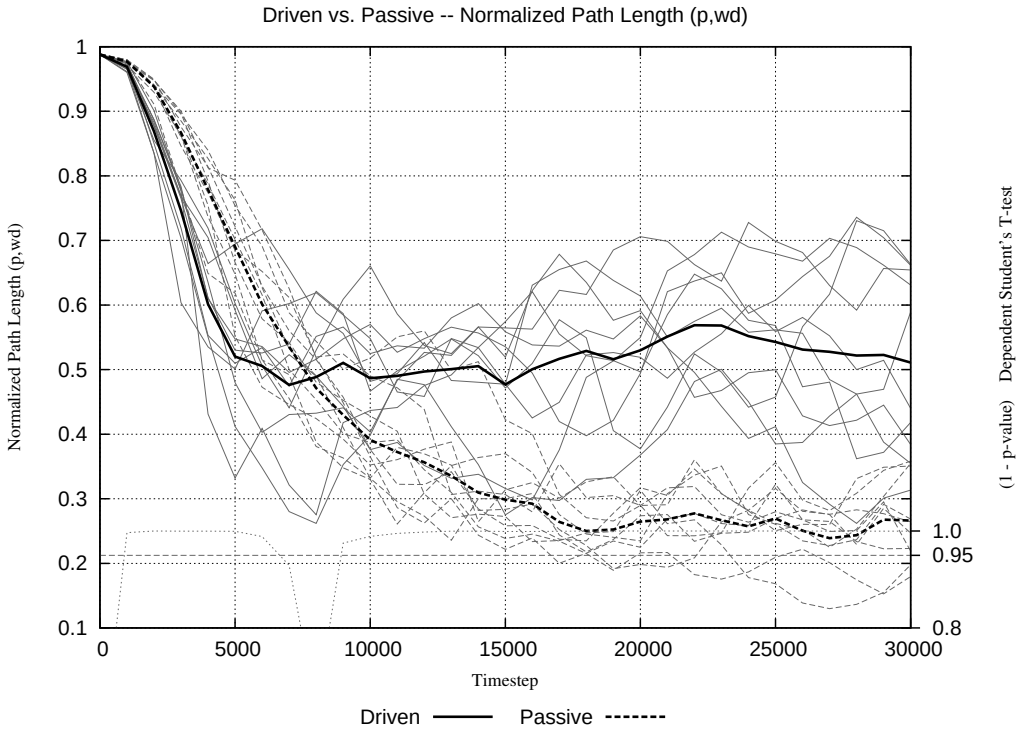


Figure 2: Driven *vs.* passive normalized path length as a function of time. Light solid lines show mean population NPL for each driven run. Light dashed lines show mean population NPL for each passive run. Heavy lines show meta-means of all ten runs for the corresponding line style. Light dotted line at bottom shows dependent $1-$p-value for a Student's T-test with typical $p > 0.05$ statistical significance indicated by the horizontal line at $p = 0.95$.

with any of the explanations we have devised so far and feel this needs further investigation, which is why none of these results are included here (though they are all present in the supplemental materials).

The actual numerical values of all these different versions of SWI are greater than 1.0 for the driven runs, ranging from 1.5 to as much as 32.0, depending on the specific data and specific form of the metric, and the values are generally (though not always) greater for the driven runs than they are for the passive runs. So all we can really take away from the SWI analysis is that the evolved nets are small-world nets.

Given the difficulties and inconsistencies with SWI, we sought to define a metric that would more directly capture and quantify the apparent bias towards high clustering and short path lengths evidenced in all of the raw clustering and path length data. To this end we have defined a new "small-world bias" (SWB) metric that takes its form from Humphries et al's SWI, but directly compares driven to passive—instead of actual to random—clustering and length metrics:

$$\gamma = \langle CC_{driven} \rangle \,/\, \langle CC_{passive} \rangle \tag{7}$$

$$\lambda = \langle L_{driven} \rangle \,/\, \langle L_{passive} \rangle \tag{8}$$

$$SWB = \gamma / \lambda \tag{9}$$

where L can be any suitable length metric (such as CPL, CL, or NPL). The ensemble averages are taken over the usual population of agents expiring during a given temporal epoch. The numerator captures the degree to which a driven run favors high clustering relative to a passive run. The denominator captures the degree to which a driven run favors low path length relative to a passive run. Accordingly, when SWB exceeds 1.0, the driven run is at least slightly biased towards small world network characteristics relative to a passive run. It is not actually possible (because driven and passive graph sizes are different), but if one could calculate Humphries et al. (2006)'s SWI using the same random-graph basis for corresponding terms in SWI_{driven} and $SWI_{passive}$, then take their ratio, all the random-graph terms would cancel out and what one would be left with is SWB.

The precise numerical values and periods of bias vary, but the resultant trends in SWB were remarkably consistent for both sets of neurons (A and P), all graph types (BU and WD), and all length metrics (CPL, CL, and NPL). Figure 3 shows the results for SWB based on connectivity length for the processing neurons treated as weighted, directed graphs. There is a strong (> 1.5) bias towards small-world-ness from about t=2000 to t=7000, corresponding to the previously observed, statistically significant growth in neural complexity and behavioral adaptation to the environment.

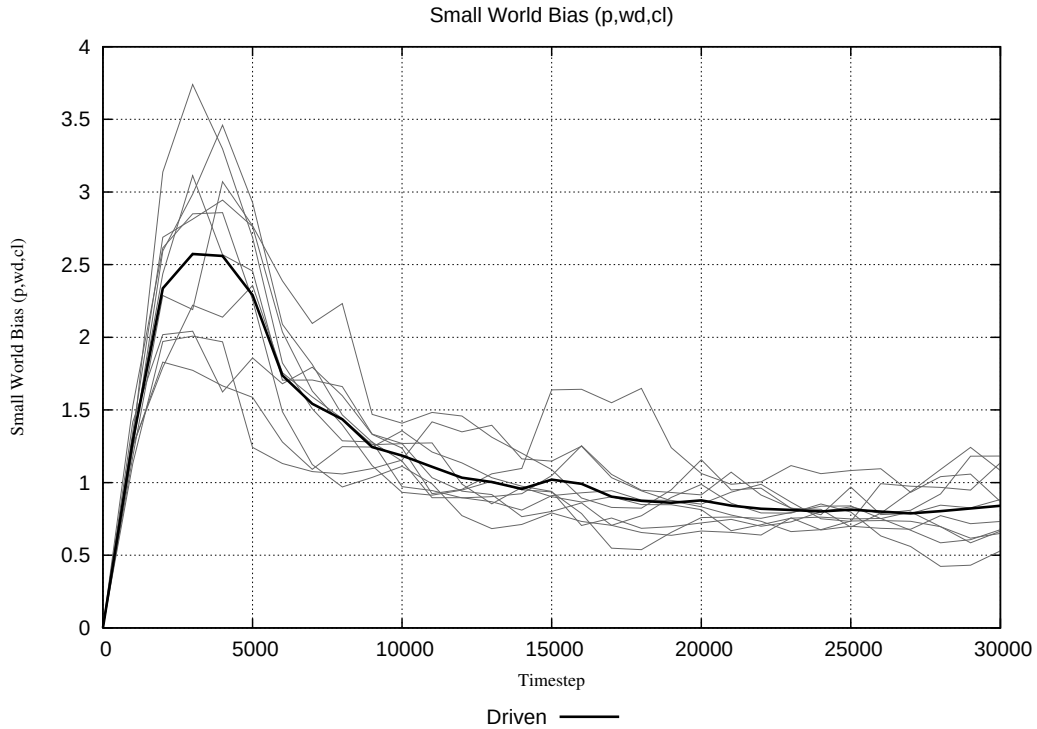


Figure 3: Small-world bias as a function of time. Where SWB is > 1.0, the driven run is exhibiting a bias towards small-world networks relative to the passive run.

Conclusions

We have shown strong, reproducible evolutionary biases towards high clustering coefficients, short path lengths, and small-world-ness in driven runs subject to natural selection relative to passive runs in which natural selection is disabled. These structural, graph theoretical trends correspond to previously observed evolutionary trends in the dynamical complexity of neural function and behavioral adaptation of agents to their environment. These observations thus strengthen the association between small-world-ness and complexity.

Short path lengths contribute to increased "integration" of neural function throughout the brain. Clustering can contribute to and is often evidence of increased "segregation" of specialized neural functions in the brain. It is this combination of increasing integration and segregation that produces the measured increases in dynamical neural complexity (Tononi et al., 1994).

Our work demonstrates that even in the absence of physical constraints on wiring length and brain volume, evolution selects for small-world networks in order to enhance brain function. The resulting networks thus combine the predominantly local connectivity imposed by physical volume constraints (Murre and Engelhardt, 1995) with the short path lengths necessary to satisfy fast response time requirements (Lago-Fernández et al., 2000), despite a lack of physical constraints in their evolution. We suggest that humans (and all biological organisms with even modestly complex nervous systems) are the fortunate beneficiaries of these convergent and synergistic physical and functional constraints. Rather than physical constraints acting to limit brain function, our evidence suggests that physical constraints work in concert with evolutionary pressures to select neural topologies that foster more complex, adaptive behaviors.

Future Directions

There is one instance in which increases in clustering coefficient are not correlated with increasing neural segregation and complexity, which is progression towards a single large cluster. Since we do see correlated increases in neural complexity our clustering increases cannot be the result of network topologies approaching a single large cluster, however in the future we intend to look into modularity metrics that more directly address community structure. Our expectations are that structural modularity and functional complexity will be positively correlated. However, preliminary inconsistent and contradictory results have led to the realization that standard measures of modularity, such as those due to Newman (2006) and Blondel et al. (2008), are not well suited to the types of networks generated early in our simulations and we believe values of these metrics are artificially elevated for such graphs. Further research is required to either develop better ways to characterize community structure in these networks or determine suitable subsets of these

graphs to which the standard modularity metrics may reasonably be applied, perhaps only after having evolved beyond certain minimum size and connectivity constraints.

We further hope to identify more discriminating structural metrics, that will be reliably predictive of functional complexity. We also seek to improve upon our current technique of ignoring (by taking absolute values) what is likely to be a crucial distinction between the positive and negative weights associated with excitatory and inhibitory connections. One particular direction we intend to explore may address both aims at once, which is distributions of signed motifs. Network motifs, such as those advanced by Milo et al. (2008) and related to small-world properties and complexity by Sporns and Kötter (2004), are typically treated as unsigned, though there has been some discussion of small subsets of signed motifs in genetic transcription and other biological networks (Alon, 2007). Work by Kashtan and Alon (2005) demonstrates that modularity and motif distributions are sometimes correlated, but not uniquely so. We speculate that motif distributions may be more discriminating and predictive of functional complexity than modularity or the other metrics we have examined to date. We also expect that extending the standard 13 unsigned motifs to a corresponding 204 signed motifs will provide much greater discrimination, as well as greater relevance to signed neural networks.

Acknowledgements

Thanks to Mikail Rubinov for his enlightening discussions on graph theory algorithms. Thanks to Santosh Manicka for his coding efforts on bct-cpp.

References

Alon, U. (2007). Network motifs: theory and experimental approaches. *Nat Rev Genet*, 8:450–461.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *J. Stat. Mech.*, page P10008.

Cherniak, C. (1995). Neural component placement. *Trends in Neurosciences*, 18:522–527.

Humphries, M. D., Gurney, K., and Prescott, T. J. (2006). The brainstem reticular formation is a small-world, not scale-free, network. *Proc. R. Soc. B*, 273:503–511.

Kashtan, N. and Alon, U. (2005). Spontaneous evolution of modularity and network motifs. *Proc. Natl. Acad. Sci. U. S. A.*

Lago-Fernández, L. F., Huerta, R., Corbacho, F., and Sigüenza, J. A. (2000). Fast Response and Temporal Coherent Oscillations in Small-World Networks. *Phys. Rev. Lett.*, 84:2758–2761.

Lizier, J. T., Piraveenan, M., Dany, P., Prokopenko, M., and Yaeger, L. S. (2009). Functional and Structural Topologies in Evolved Neural Networks. In Kampis, G., Szathmry, E., Fernando, C., Jelasity, M., Jordn, F., Lrincz, A., and Scheuring, I., editors, *Advances in Artificial Life: Proceedings of the Tenth European Conference on Artificial Life (ECAL2009)*. Springer Verlag, Heidelberg.

Lungarella, M., Pegors, T., Bulwinkle, D., and Sporns, O. (2005). Methods for quantifying the information structure of sensory and motor data. *Neuroinformatics*, 3(3):243–262.

Marchiori, M. and Latora, V. (2000). Harmony in the Small-World. *Physica A*, 285(3-4):539–546.

McShea, D. W. (1996). Metazoan complexity and evolution: Is there a trend? *Evolution*, 50:477–492.

Milo, R., Itzkovitz, S., Kashtan, N., Levitt, R., Shen-Orr, S., Ayzenshtat, I., Sheffer, M., and Alon, U. (2008). Superfamilies of Evolved and Designed Networks. *Science*, 303:1538–1542.

Mitchison, G. (1991). Neuronal branching patterns and the economy of cortical wiring. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 245(1313):151–158.

Murre, J. M. J. and Engelhardt, D. P. F. (1995). The connectivity of the brain: multi-level quantitative analysis. *Biological Cybernetics*, 73:529–545.

Newman, M. E. J. (2006). Modularity and community structure in networks. *Proc. Natl. Acad. Sci. U. S. A.*, 103:8577–8582.

Rubinov, M. and Sporns, O. (2010). Complex network measures of brain connectivity: Uses and interpretations. *NeuroImage*, In Press, Corrected Proof:–.

Sporns, O. and Kötter, R. (2004). Motifs in brain networks. *PLoS Biol*, 2(11):e369.

Sporns, O., Tononi, G., and Edelman, G. (2000). Theoretical Neuroanatomy: Relating Anatomical and Functional Connectivity in Graphs and Cortical Connection Matrices. *Cerebral Cortex*, 10:127–141.

Strogatz, S. H. (2001). Exploring complex networks. *Nature*, 410:268–276.

Tononi, G., Edelman, G., and Sporns, O. (1998). Complexity and coherency: integrating information in the brain. *Trends in Cognitive Sciences*, 2(12):474–484.

Tononi, G., Sporns, O., and Edelman, G. (1994). A measure for brain complexity: Relating functional segregation and integration in the nervous system. *Proc. Nat. Acad. Sci.*, 91:5033–5037.

Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442.

Yaeger, L. S. (1994). Computational Genetics, Physiology, Metabolism, Neural Systems, Learning, Vision, and Behavior or Polyworld: Life in a New Context. In Langton, C. G., editor, *Proceedings of the Artificial Life III Conference*, pages 263–298. Addison-Wesley, Reading, MA.

Yaeger, L. S. (2009). How evolution guides complexity. *HFSP*, 3(5):328–339.

Yaeger, L. S., Griffith, V., and Sporns, O. (2008). Passive and Driven Trends in the Evolution of Complexity. In Bullock, S., Noble, J., Watson, R., and Bedau, M. A., editors, *Artificial Life XI: Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, pages 725–732. MIT Press, Cambridge, MA.

Yaeger, L. S. and Sporns, O. (2006). Evolution of Neural Structure and Complexity in a Computational Ecology. In Rocha, L., Yaeger, L. S., Bedau, M., Floreano, D., Goldstone, R., and Vespignani, A., editors, *Artificial Life X: Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, pages 330–336. MIT Press (Bradford Books), Cambridge, MA.

Stability in Flux: Group Dynamics in Evolving Networks

Nicholas Geard[1], John Bryden[2], Sebastian Funk[2], Vincent Jansen[2],and Seth Bullock[1]

[1]School of Electronics and Computer Science, University of Southampton
Southampton, SO17 1BJ, United Kingdom
[2]School of Biological Sciences, Royal Holloway University of London
Egham, TW20 0EX, United Kingdom
nlg@ecs.soton.ac.uk

Extended Abstract

From Facebook groups and online gaming clans, to social movements and terrorist cells, groups of individuals aligned by interest, values or background are of increasing interest to social network researchers (Snow et al., 1980; Zheleva et al., 2009). In particular, understanding the structural and dynamic factors that influence the evolution of these groups remains an open challenge (Palla et al., 2007; Geard and Bullock, 2008, 2010). Why do some groups persist and succeed, while others fail to do so?

Three features characterise real social networks. They are inherently dynamic: explaining the structure of social networks requires us to understand how this structure is created, modified and maintained. They are co-evolutionary, exhibiting a reflexive relationship between topology and state. For example, individuals often interact preferentially with others who are similar to themselves, thus state affects topology; at the same time, neighbouring individuals tend to influence one another and hence become more similar, thus topology affects state (Gross and Blasius, 2008). Finally, interactions between individuals are not distributed uniformly across a network: rather, we can detect community structure, in which subsets of individuals are more densely linked to each other than to the rest of the population (Newman, 2006).

Analysis of telephone and collaboration data by Palla et al. (2007) has demonstrated some of the ways in which social groups evolve over time, but there is more to be done in understanding the multi-level relationship between individual and group dynamics. Here, we address two questions: How do stable macro-level structures and behaviours emerge and persist as a consequence of simple micro-level processes? How can we characterise the dynamics of meso-level structures such as groups and communities?

We introduce a simple model of a co-evolving network in which the state of an individual represents the group to which it is currently (and exclusively) affiliated. Four processes govern network evolution: individuals can create new groups, influence neighbours to switch affiliation to their group, replace an out-group edge with an in-group edge, or replace edges at random.

Using this model, we explore the parameter space defined by the relative rates of each process, revealing a region in which networks exhibit connected community structure reminiscent of observed social networks (Figure 1). We demonstrate how macro-level properties of the network (e.g., state and degree distribution, modularity, clustering coefficient and path length) stabilise, while underlying micro- and meso-level properties remain dynamic; that is, individuals continue to update their neighbours and states, and groups are born, grow, shrink and die.

Finally, we report findings on the behaviour of groups: at equilibrium, there is a stable rank-distribution of group sizes; however, the identities of the groups occupying each rank change over time. Furthermore, the distribution of group lifespans is bimodal, reflecting two possible group trajectories: After being introduced into a population, a group either thrives, or struggles. Interestingly, the probability of these two events appears to be almost entirely stochastic, and appears to be independent of factors that one might expect play a role, such as the location of group foundation.

While our model is undoubtedly simple, we believe it provides a useful baseline for further studies, and a helpful tool for understanding the multi-level dynamic interactions that underlie the complex behaviour of more complicated models.

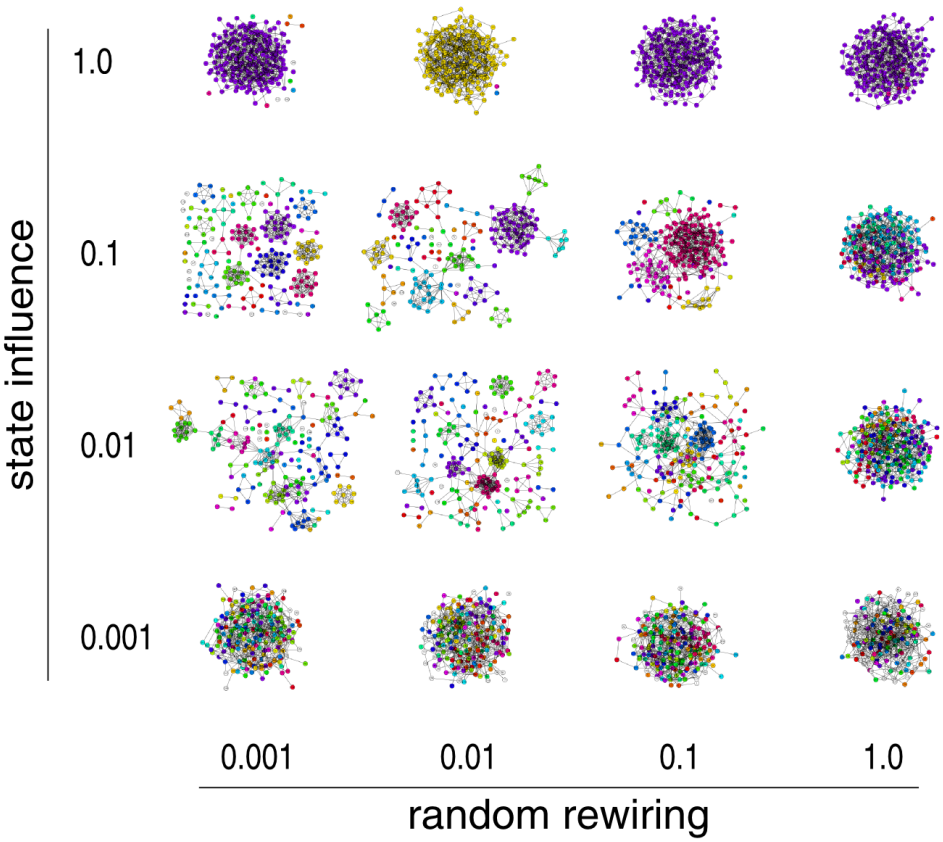


Figure 1: A slice through model parameter space, showing sample networks that result from different rates of state influence (y-axis) and random rewiring (x-axis), given fixed rates of group rewiring (1.0) and state innovation (0.001). When state influence is very high (top row), a single group spreads to dominate the population. In contrast, when state influence is very low (bottom row), groups grow very slowly, if at all, and many small groups coexist. When random rewiring is very high (right column), little structure emerges in the population. Lower levels of random rewiring enable the emergence of topological communities focused around shared state. These communities either disconnect completely, fragmenting the population and inhibiting the flow of individuals between groups, or remain connected (the central region). Note that these networks are static snapshots: while aggregate network properties stabilise, local properties such as the pattern of social ties and distribution of groups continue to evolve dynamically.

References

Geard, N. and Bullock, S. (2008). Group formation and social evolution: a computational model. In *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*.

Geard, N. and Bullock, S. (2010). Competition and the dynamics of group affiliation. *Advances in Complex Systems*, In Press.

Gross, T. and Blasius, B. (2008). Adaptive coevolutionary networks: a review. *J R Soc Interface*, 5(20):259–271.

Newman, M. E. J. (2006). Modularity and community structure in networks. *Proc Natl Acad Sci USA*, 103(23):8577–8582.

Palla, G., Barabási, A.-L., and Vicsek, T. (2007). Quantifying social group evolution. *Nature*, 446:664–667.

Snow, D. A., Zurcher, L. A., and Ekland-Olson, S. (1980). Social networks and social movements: A microstructural approach to differential recruitment. *American Sociological Review*, 45(5):787–801.

Zheleva, E., Sharara, H., and Getoor, L. (2009). Co-evolution of social and affiliation networks. In *The 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Network Measures of Ecosystem Complexity

Alan Dorin and Kevin B. Korb

Faculty of Information Technology,
Monash University, Clayton, Australia
[alan.dorin, kevin.korb] @ infotech.monash.edu.au

Abstract

We argue that the networks that can be constructed to represent ecosystems may inform us about the open-endedness of the evolutionary systems that underlie their dynamics. By adopting this approach we circumvent problems that arise from looking for open-endedness at the level of the organism, the more usual approach. We then examine various measures of ecosystem (niche web) complexity and propose a new information-theoretic approach, *Shannon Web Complexity*. We compare its behaviour to that of the more common measures in ecology, in the light of common intuitions about complexity over a set of test networks and real ecosystem trophic webs. We show that our measure better accommodates intuitions about the complexity of these networks.

Introduction

The search for open-ended evolutionary simulations is compelling and has driven a sub-community of Artificial Life researchers to join philosophers and theoretical biologists in pondering the manner in which biological evolution is open-ended. This has resulted in various simulation environments that attempt to replicate the behaviour of real ecosystems (e.g. see the review in (Dorin, Korb et al. 2008)) and open problems such as the call to, "Create a formal framework for synthesizing dynamical hierarchies at all scales" (Bedau, McCaskill et al. 2000).

To achieve the goal of open-ended evolutionary software, we must first unambiguously identify open-ended complexity increase when we see it – we require a measure. Typically, as we show below, the search has focused on the increasing complexity of *organisms*, their structure and behaviour. For reasons we outline, we believe this to be wrong-headed and the source of much confusion. Instead, we propose to measure the complexity of the ecosystems of which organisms are a part, and to show that these *do* increase in complexity over evolutionary time periods. We achieve this by looking at ecosystem networks.

Ecosystem networks

Biological evolution operates within ecosystems on changing populations that define for themselves new ways of accumulating and consuming energy and matter to be employed for reproduction. Through feedback loops, organisms construct their own niches, passively and actively organising their environment, modifying the selection pressures acting on themselves, their progeny, and their cohabiters (Odling-Smee, Laland et al. 2003). The moulding of self-selection pressures by a population shifts the constraints within which future generations are introduced. Ecosystems can be described by a variety of networks linking these biotic and abiotic physical, chemical and behavioural relationships. We, like many ecologists, focus our attention on such networks as a way of understanding the global properties of the systems they represent (Watts and Strogatz 1998; Barrat and Weigt 2000; Dunne, Williams et al. 2002; Proulx, Promislow et al. 2005; Blüthgen, Fründ et al. 2008).

We examine several techniques employed in the ecological and other literature for measuring the properties of ecosystem food webs and networks, describing also the *Shannon web complexity* based on information theory (Boulton and Wallace 1969). We then assess how these measures stack up against one another and against our intuitions about the complexity of ecosystem networks in a set of examples.

Open-Ended Complexity Increase

A common opinion about evolution has been that it swims against the tide of entropy and in particular that evolution over time constructs more and more complex organisms (e.g., see (Bronowski 1970)). This idea of creative complexity increase equates at its most extreme, to the view that evolution is progressing from bacteria to invertebrates and thence to vertebrates and mammals and, finally, to the pinnacle of life forms, us.[1] Such a view of Progress, however, ignores some quite basic features of evolution. For example, that the bacteria being "progressed from" still exist today and, indeed, have exactly as long an evolutionary history as we do, since we all have common ancestry. So, progress can hardly be characterized by endurance. Instead, progress has been recast as complexity, and complexity itself has been cast in terms favorable to ourselves; for example, as owning complex neural organizations — an account that fails to address the vast majority of earth's life (Maynard-Smith and Szathmáry

[1] For a skeptical review of this consensus opinion, identifying culprits, see McShea, D. W. (1991). "Complexity and evolution: what everybody knows." *Biology and Philosophy* **6**(3): 303-324.

1995). An infatuation with ourselves is also behind the "C-value paradox" — our chromosomes appear no more complex than those of other mammals and less complex than those of some plants. The two long-standing antagonists Dawkins and Gould have together, and quite rightly, castigated this view as human chauvinism in an exchange in Evolution (Dawkins, 1997; Gould, 1997). Gould preferred to see in every attempt to characterize complexity and attribute its increase to evolutionary processes this hidden agenda of congratulating ourselves on our own unique wonderfulness. Dawkins, on the other hand, considers the evolutionary increase in complexity to be not just compatible with evolution, but intrinsic to it. Evolution climbs "Mount Improbable" (Dawkins, 1996). Dawkins' line of defence for ongoing complexity increase is to suggest that, whereas adaptive processes responding to the abiotic environment may just track meandering changes in the climate, coevolutionary processes acting between species work to develop coadaptations in trajectories that can be regarded as progressive in an engineering sense. Arms races lead to better weaponry and better defences, including better speed, flight, hearing and vision, for example.

In the Artificial Life literature, Bedau takes up the debate, offering his evolutionary activity statistics to assess whether or not an evolutionary system is evolving in an open-ended fashion (Bedau, Snyder et al. 1998) and an *Arrow of Complexity Hypothesis* that evolutionary systems show a systematic tendency to increase the complexity of organisms over time (Bedau 2006). Some Artificial Life researchers, notably (Ray 1990), have attempted to replicate this apparent evolutionary complexity increase in software, thus far, without any consensus of success, although some claim a limited success whilst improving Bedau et al's measures of open-endedness (Channon 2006). The fundamental problem we have with the activity statistics, however, is that whatever they measure is not what we want to measure: they make no attempt to assess the complexity of organisms or ecosystems, but only the volume of new, adaptive "components" within an evolutionary system.

An attempt to dismiss complexity increases in species' organisation and behaviour over evolutionary time periods invokes a metaphoric "passive diffusion" (McShea 1994) through species design space, rather than a directed drive towards greater complexity. While diffusion may well contribute to increases in species complexity, it is unlikely to explain it entirely (Korb and Dorin 2010). In any case, we prefer to sidestep the issue and focus on complexity at a higher level: in the organization of niches in the ecosystem. Niche web complexity is not subject to the diffusion effects cited by McShea and others; furthermore, it, and its correlate species biodiversity, relatively non-controversially have shown sustained increases over geological time. Indeed, it is arguable that niche web complexity exhibits an exponential trajectory over evolutionary time periods, which we call the *Arrow of Niche Complexity Hypothesis* (Korb and Dorin 2010): with complexity interpreted simply as the number of niches, this hypothesis states that any ecosystem acting beneath the ceiling of its capacity constraints whilst maintaining its stability will robustly tend to produce new niches, at an exponential growth rate – every species, without exception, creates multiple new niches by its waste products, its impact as an ecosystem engineer (the existence of its body as habitat, for instance), its availability as food for other organisms, and its removal of resources from the environment changing their relative abundance and distribution. Elsewhere we offer a simulation that demonstrates the effect of an exponentially increasing number of niches (Korb and Dorin 2009). Furthermore, the network of dependence of species (in niches) to other species (in other niches) also increases in complexity. In order to argue that the latter increases are exponential and, in general, to assess changes in niche web complexity, we require a principled way of measuring such complexity.

Complexity Measures for Ecosystems

We require a measure for the complexity of (virtual or real) ecosystems in order to assess whether or not our *Arrow of Niche Complexity* hypothesis holds true under some circumstances. This measure must correspond to our (educated) intuitions about what constitutes the complexity of a network (such as a food web). A few useful intuitions are listed next. We then present some measures of network properties that have been employed in the literature and our own suggestion.

Intuitions about network complexity

Intuition 1 (simple): A network with a regular, repeating structure is simple (e.g. a lattice or a fully-connected network).

Intuition 2 (simple): Networks with few links are simple (e.g. a single long chain or a fully disconnected network).

Intuition 3 (simple): A random network is simple (with a high probably; but since random processes can produce any structure, such a net will sometimes accidentally be complex!).

Intuition 4 (simple): Small world networks – those with low "degrees of separation" – are simple.

Intuition 5 (complex): A complex network has organisation (e.g. clusters, loops) at multiple scales.

Intuition 6 (complex): A complex network has organisation (clusters, loops) of multiple sizes.

Intuition 7 (complex): A bigger network is more complex than a small one.

These intuitions, while widely commented upon in the ecological literature, are not universal; nor are they unambiguous. For one thing, they only make sense with *ceteris paribus* clauses – other things remaining equal. And there are potential interactions between some of them. For example, Intuition 7 may be undermined by increasing the size of the network while simultaneously deleting arcs and bringing in Intuition 2. They work perhaps as heuristic guides to assessing networks and their complexity measures.

Intuitions 5 and 6 are likely to capture some aspects of the major transitions in evolution, which can lead to tightly organized groups of niches.

Intuitions 1 and 2 have an interesting joint consequence which we will make use of later: networks of low density are simple, but so too are networks of high density (fully connected or worse). We infer that there may well be some kind of "Goldilocks effect", i.e., that there is a maximum of network complexity achieved at some middle level of density, which tapers off when there is either too many or too few arcs. The same effect applies to Intuition 4 as well: very small worlds mean very high interconnection, while very large worlds imply very low interconnection.

Measures of network properties

Networks have been widely studied in biology and several measures have been used to inform us about their properties in general (Watts and Strogatz 1998; Dunne, Williams et al. 2002; Proulx, Promislow et al. 2005; Neutel, Heesterbeek et al. 2007; Blüthgen, Fründ et al. 2008). Here we list some of relevance,[2] gauging the extent to which each informs us about network complexity. We conclude this list by introducing our own proposal.

Number of nodes: n
Ceteris paribus, smaller networks are simpler networks (cf. Intuition 7).

Number of edges: e
Having fewer edges is another way in which networks can be smaller and therefore simpler (Intuition 2).

Density: $D = e / n^2$
Given that there are n^2 potential directed arcs in a network (where a node may have an arc directed back to itself), this is the frequency of arcs (relevant to Intuitions 1, 2, 4, 5 and 6).

Density-Mass: $D \times n = e / n$
This combines Intuitions 2 and 7. Given that denser networks are more complex (other things being held equal *and* up to a point of diminishing returns) and larger networks are more complex, it's reasonable to suppose that a measure of complexity might be proportional to both simultaneously, so we multiply the two measures.

Characteristic path length (CPL):

$$CPL = \sum_{j=1}^{n} \sum_{i=1}^{n} \frac{s_{ij}}{p}$$

where s_{ij} is the shortest path between nodes i and j (0 in case the shortest path is infinite) and p is the number of finite shortest paths between two nodes in the network (i.e. $p < n^2$ just in case some shortest paths are infinite). Thus, *CPL* is the average shortest path length ("degree of separation") between nodes. Low values would normally indicate a highly connected network, i.e., high edge density, or perhaps strategically placed edges allowing for shortcuts, corresponding to Intuitions 1 and 4.

[2] The literature contains many measures and variations. We focus on a few popular unweighted measures. Measures such as the *maximum omnivourous loop weight* (Neutel, Heesterbeek, et al. 2007) are useful in some ecological applications but obviously not applicable to networks with unweighted edges.

Clustering Coefficient (CC):

$$CC = \frac{1}{n} \sum_{i} \frac{S_i}{N_i}$$

where N_i is the number of i's neighbors and S_i is the number of shared neighbors, i.e., neighbors which are also neighbors of neighbors. This measures, on average, how "cliquey" the neighbors are across a network.
In a niche web a high clustering coefficient shows the presence of tightly coupled clusters of niche-dependencies. So, this is a partial indicator of the clusters and loops of Intuitions 5 and 6.

Shannon web complexity (SWC).
This is a new use of a prior information-theoretic complexity measure, measuring niche web complexity by the number of bits needed to efficiently encode a network with n nodes, where the web may be any directed graph between the nodes. The code should be Shannon efficient for specifying the network structure to a receiver. In this first version of SWC we make the simplifying assumption that the density of arcs in the network is uniform; i.e., the number of arcs in any two subgraphs of the same size is approximately the same. This assumption admittedly will be untrue for many networks, when the measure will no longer be Shannon efficient; however, SWC can be refined in the future to deal with such networks. As it stands, this measure will still be useful for a very large range of networks.

First, we need to identify (label, number) all the nodes. We can do this simply by specifying how many there are, i.e., coding the number n, assuming the labels will be 1, 2, …, n.

$$\log_2 n$$

Now we need to specify all arcs. We can do this in two steps. First we encode an estimate p of the probability that an arc exists between any two nodes; call this code length $M(p)$. Given knowledge of p, specifying an existing arc takes $-\log_2 p$ bits and specifying the absence of an arc takes $-\log_2 (1 - p)$ bits. The number of possible arcs (going in either direction between nodes) is n^2 (since nodes may be parents of themselves), so

$$p = e / n^2$$

where e is the number of arcs in the graph (i.e., this is the density measure from above).
Hence, we can identify the arc structure in the following number of bits:

$$e\,(-\log_2 p) + [n^2 - e]\,(-\log_2 (1 - p))$$

The first summand is the bit cost of specifying e arcs; the second is the bit cost of specifying all other potential arcs are missing. So, our final measure is:

$$M(p) + \log_2 n + e\,(-\log_2 p) + [n^2 - e]\,(-\log_2 (1 - p))$$

This has the reasonable Goldlilocks property above: a low density web is counted as simple; complexity increases as the number of arcs increase; but as the web becomes very dense – as, for example, an ecosystem turns into an indiscriminate

mush – it starts losing complexity. Maximal complexity is reached when $p \approx 0.5$. This measure is shown by (Boulton and Wallace 1969) to be effectively the same as the following adaptive code, which is simpler to compute (meaning, e.g., we don't actually have to measure $M(p)$):

$$\log_2 \frac{(n^2+1)!}{e!(n^2-e)!}$$

This measure doesn't respond directly to the Intuitions that loopiness implies complexity (5 and 6), however as the arc density goes from low towards 0.5, loopiness is inevitable. Loopiness is improbable at low arc densities, while in some way meaningless at very high arc densities.

Testing Our Measures

Sample graphs.

Figure 1 shows four test graphs C1… 4 that we have designed with a constant number of nodes but increasing number of edges to highlight the behaviour of the network measures.

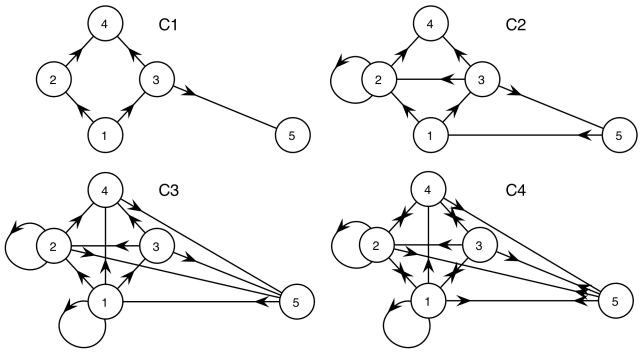


Figure 1. Graphs of five nodes with increasing number of edges.

Figure 2 is a set of networks showing successional stages of a subterranean food web redrawn from (Neutel, Heesterbeek et al. 2007). To the authors of that paper and this alike, these networks appear to be of increasing complexity[3]. In the following section we present the results of our measurement of the properties of these two sets of graphs.

[3] Sch(iermonnikoog) and Hul(shorsterzand) 1 are both successional stage 1 food webs. Hul 2, 2-3 and 4 are subsequent stages of development at the latter site. Nodes represent trophic groups detailed in the original paper.

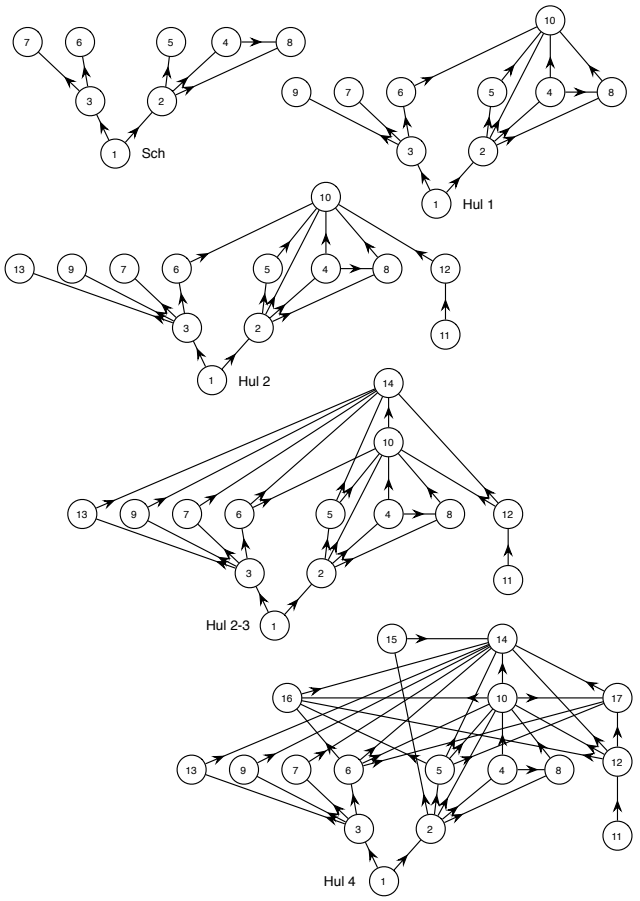


Figure 2. Graphs of subterranean food webs at progressing successional stages (Schiermonnikoog and Hulshorsterzand in the Netherlands).

Results

Figure 3 allows us to read the trends of the measures given above for graphs C1-4. Apart from SWC and CPL, all measures rise with the number of edges in the network. This certainly corresponds with naïve Intuition 2. But this suggests the measures are actually poor indicators of complexity as the sustained increase contradicts Intuitions 1 and 4 that as the network becomes more fully connected, it is becoming more homogeneous, less likely to have long loops and distinct clusters, and therefore *less* complex. In contrast, we see here that SWC and CPL both take the requisite dive after C3 (which has density ≈ 0.5) as the network connectedness climbs "too far".

Figures 4 and 5 show the CPL and SWC respectively, as applied to the webs of figure 2. The CPL drops in the middle stages, before rising once again. As the ratio of the number of arcs to number of nodes increases (i.e., the edge density increases), the chance of having differentiated sub-networks actually decreases – the network will become one large structure with many internally connecting arcs. Depending on how these edges are added, the characteristic path length may drop, as is the case here and as we saw above, in moving from our network C3 to C4. If more nodes are later added in such a

way as to add *lengthy* loops, then the CPL too will rise.
The SWC demonstrates a continued increase in complexity across the webs as we, and the ecologists, would wish.

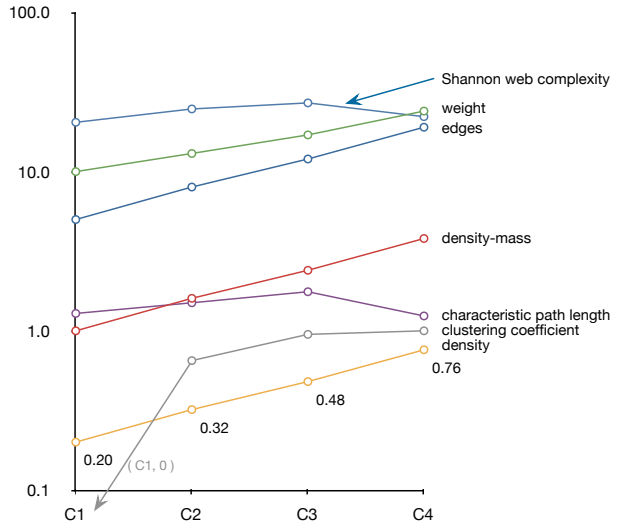


Figure 3. Trends of the various dimensionless measures across test graphs, C1-4 (vertical axis has a log scale).

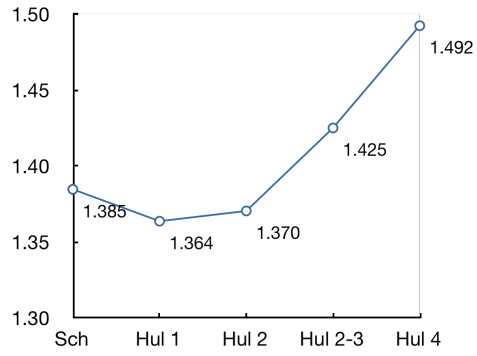


Figure 4. The drop in characteristic path length (CPL) of the food webs from Sch to Hul 1 and 2 is counter to our intuition about the webs' complexity.

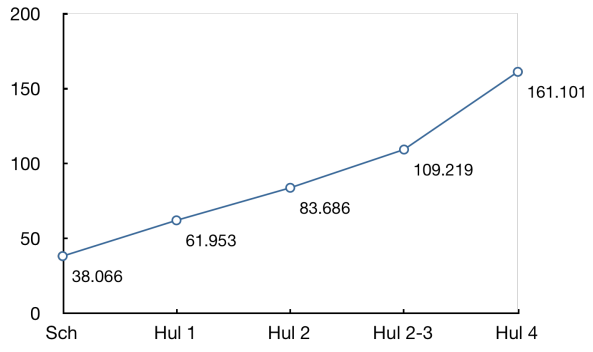


Figure 5. The increase in the Shannon web complexity (SWC) of the food webs matches our intuitions about their complexity.

Discussion and Future Work

Our proposed measure of niche web complexity is an improvement upon Bedau's evolutionary activity plots for identifying open-ended evolution – in particular, it measures the right thing, biological complexity, at at least one of the right levels of organisation, the niche web. Even this first SWC measure appears tricky to subvert; it is at least better than the measures actually employed in the ecological literature. In particular, we have shown that SWC corresponds to basic intuitions regarding complexity and, at least in our test cases, tells us more than its competitors in this regard. There are various options for improvement nevertheless. We can anticipate in the future looking at: the number of iterations required to reduce a non-planar graph to planarity by subtraction of maximal planar subgraphs; the standard deviations of shortest path lengths clustering coefficients across subgraphs; dropping the assumption of uniform arc densities in the SWC measure by compounding the SWCs of subgraphs.

Even before we extend our existing measures, we plan to apply them to the networks generated by various artificial-life ecosystems, especially our own (Korb and Dorin 2009) and those measured by others using their own statistics (e.g. (Ray 1990; Bedau, Snyder et al. 1998; Channon and Damper 2000)) to see what they may tell us about the simulations' open-endedness. Should they prove to support open-endedness, one significant hurdle must still be overcome — accommodating the "major transitions" of evolution (Maynard-Smith and Szathmáry 1995) that play a key role in the open-endedness of real evolution. Can these be replicated in simulation? Would our measures detect them if they did occur?

The major transitions such as the evolution of eukaryotes and the development of sexual reproduction, relate in part to changes in how information is passed between generations. Niche webs do not explicitly model such behaviour; however, another prominent feature of many of these transitions is the incorporation of one entity in the life cycle of another (e.g., bacteria in digestion or the development of mitochondria) or, again, the differentiation of subparts into specialising modules (e.g., new organs and tissues). These kinds of transitions have impacts on niche webs, either explicitly or implicitly, and will often show up in the ways in which subgraphs of niches are interrelated. So, while there are limitations to what examining niche webs can reveal about major transitions, there are also potential impacts of the transitions on niche webs that should not go unexamined.

Conclusions

Niche web complexity is a promising focus for understanding biological complexity growth and so for assessing also the complexity of Artificial Life simulations. While there is a long tradition in ecology of considering this kind of complexity, most of the literature uncritically adopts one or another measure on the basis of intuitive arguments. We have codified these intuitions, formalized a variety of measures corresponding to them, as well as an information-theoretic measure, and tested them using a range of networks. We think

the information-theoretic measure has considerable promise for assisting us in understanding biological complexity growth and, therefore, open-ended evolution.

Acknowledgements

The authors wish to thank Lloyd Allison for spotting bugs in an early version of our SWC and Mark Bedau and Tim Taylor for discussions on the issues addressed by this paper.

References

Barrat, A. and M. Weigt (2000). "On the properties of small-world network models." *European Phsycial Journal B* 13: 547-560.

Bedau, M. (2006). The evolution of complexity. *Symposium on the Making up of Organisms*, Ecole Normale Suprieure.

Bedau, M. A., J. S. McCaskill, et al. (2000). "Open Problems in Artificial Life." *Artificial Life* 6(4): 363-376.

Bedau, M. A., E. Snyder, et al. (1998). *A Classification of Long-Term Evolutionary Dynamics*. Artificial Life VI, MIT Press.

Blüthgen, N., J. Fründ, et al. (2008). "What do interaction network metrics tell us about specialization and biological traits." *Ecology* 89(12): 3387-3399.

Boulton, D. M. and C. S. Wallace (1969). "The information content of a multistate distribution." *Journal of Theoretical Biology* 23: 269-278.

Bronowski, J. (1970). "New concepts in the evolution of complexity." *Synthese* 21: 228-246.

Channon, A. (2006). "Unbounded evolutionary dynamics in a system of agents that actively process and transform their environment." *Genetic Programming and Evolvable Machines* **7**(3): 253-281.

Channon, A. D. and R. I. Damper (2000). "Towards the evolutionary emergence of increasingly complex advantageous behaviours." *International Journal of Systems Science* 31(7): 843 – 860.

Dorin, A., K. B. Korb, et al. (2008). Artificial-Life Ecosystems: What are they and what could they become? *Eleventh International Conference on Artificial Life*. S. Bullock, J. Noble, R. A. Watson and M. A. Bedau, MIT Press**:** 173-180.

Dunne, J. A., R. J. Williams, et al. (2002). "Food-web structure and network theory: The role of connectance and size." *Proceedings of the National Academy of Sciences* 99(20): 12917-12922.

Korb, K. B. and A. Dorin (2009). A simulation of niche construction, Bayesian Intelligence. 2009/2: 13.

Korb, K. B. and A. Dorin (2010). Evolution unbound: releasing the arrow of complexity. *Bayesian Intelligence Technical Report*, Bayesian Inteligence. 2010/2: 21.

Maynard-Smith, J. and E. Szathmáry (1995). *The Major Transitions in Evolution*. New York, W.H. Freeman.

McShea, D. W. (1991). "Complexity and evolution: what everybody knows." *Biology and Philosophy* 6(3): 303-324.

McShea, D. W. (1994). "Mechanisms of large-scale evolutionary trends." *Evolution* 48(6): 1747-1763.

Neutel, A.-M., J. A. P. Heesterbeek, et al. (2007). "Reconciling complexity with stability in naturally assembling food webs." *Nature* 449: 599-602.

Odling-Smee, F. J., K. N. Laland, et al. (2003). *Niche Construction, the neglected process in evolution*. Princeton, Princeton University Press.

Proulx, S. R., D. E. L. Promislow, et al. (2005). "Network thinking in ecology and evolution." *Trends in Ecology and Evolution* 20(6): 345-353.

Ray, T. S. (1990). *An approach to the synthesis of life*. Artificial Life II, Santa Fe, New Mexico, Addison Wesley.

Watts, D. J. and S. H. Strogatz (1998). "Collective dynamics of small-world networks." *Nature* 393: 440-442.

Classifying Complex Networks using Unbiased Local Assortativity

Mahendra Piraveenan[1,2], Mikhail Prokopenko[2] and Albert Y. Zomaya[1]

[1] Centre for Distributed and High Performance Computing, School of Information Technologies,
The University of Sydney, NSW 2006, Australia
[2] CSIRO Information and Communications Technologies Centre, PO Box 76, Epping, NSW 1710, Australia
mikhail.prokopenko@csiro.au

Abstract

Assortativity is a network-level measure which quantifies the tendency of nodes to mix with similar nodes in a network. Local assortativity has been introduced as a measure to analyse the contribution of individual nodes to network assortativity. In this paper we argue that there is a bias in the formulation of local assortativity which favours low-degree nodes. We show that, after the bias is removed, local assortativity of a node can be interpreted as a scaled difference between the average excess degree of the node neighbours and the expected excess degree of the network as a whole. Finally, we study the local assortativity profiles of a number of model and real world networks, demonstrating that four classes of complex networks exist: (i) assortative networks with disassortative hubs, (ii) assortative networks with assortative hubs, (iii) disassortative networks with disassortative hubs, and (iv) disassortative networks with assortative hubs.

Introduction

Many complex systems are amenable to be described as networks, with a given number of nodes and connecting edges. These include ecological systems, author collaborations, metabolism of biological species, and interaction of autonomous systems in the Internet, among others (Solé and Valverde, 2004; Albert and Barabasi, 2002; Albert et al., 1999; Newman, 2003; Faloutsos et al., 1999). It has been a recent trend to study common topological features of such networks. Network diameter, clustering coefficients, modularity and community structure, information content are some features analysed in recent literature in this regard (Faloutsos et al., 1999; Alon, 2007; Lizier et al., 2009; Prokopenko et al., 2009). One such measure which has been analysed extensively is assortativity (Solé and Valverde, 2004; Newman, 2002; Albert and Barabasi, 2002; Newman, 2003; Callaway et al., 2001; Palsson, 2006; Maslov and Sneppen, 2002; Zhou et al., 2008; Bagler and Sinha, 2007; Vázquez, 2003). Having originated in ecological and epidemiological literature (Albert and Barabasi, 2002), the term 'assortativity' refers to the correlation between the properties of adjacent network nodes.

While similarity between adjacent nodes can be measured in a number of ways, the property that is of interest to us is node degree. Based on degree-degree correlations, assortativity has been defined as a correlation function, and the level of assortative mixing has been measured quantitatively for a number of networks, including social, biological and technical networks (Solé and Valverde, 2004). The networks that have a positive correlation coefficient are called assortative: similar nodes tend to mix with each other in such networks. The networks characterised by a negative correlation coefficient are called disassortative: dissimilar nodes tend to connect predominantly in these networks. The precise local contribution of each node to the global level of assortative mixing can also be quantified (Piraveenan et al., 2008, 2009b, 2010). This quantity has been called "local assortativity". Local assortativity measures the local contribution of each node to the global correlation coefficient which is the network assortativity. Local assortativity profiles (as distributions of local assortativity over nodes' degrees) can also be constructed for various networks, and these profiles, in turn, can be used to classify networks (Piraveenan et al., 2008, 2009a). Two such classes of disassortative networks have been proposed in Piraveenan et al. (2008).

In this paper, we demonstrate that the formulation proposed for local assortativity in Piraveenan et al. (2008) has a bias, which favours low-degree nodes over hubs. This bias needs to be removed before networks can be analysed in terms of local assortativity. Therefore, our objective is twofold: (i) to propose an unbiased formulation of local assortativity, and (ii) to characterise classes of networks in terms of this unbiased formulation. After presenting the unbiased formulation for local assortativity, we show that the classification of disassortative real-world networks that was proposed in Piraveenan et al. (2008) still holds, and in addition, there are two similar classes among assortative networks as well. The unbiased formulation also provides a clearer interpretation of what it means for a node to be locally assortative.

Definitions and Terminology

We need to introduce a number of definitions before removing the bias from the formulation of local assortativity. Consider a network with N nodes and M links. Assortativity for

such a network has been defined as a correlation function (Newman, 2002), in terms of the network's excess degree distribution $q(k)$, and link distribution $e_{j,k}$. The excess degree is the number of remaining links encountered when one reaches a node by traversing a link. The link distribution of the network is the joint probability distribution of the excess degrees of the two nodes at either end of a randomly chosen link. The formal definition of network assortativity is given by:

$$r = \frac{1}{\sigma_q^2}\left[\sum_{jk} jk\left(e_{j,k} - q\left(j\right)q\left(k\right)\right)\right] \tag{1}$$

where $e_{j,k}$ is the link distribution of the network and σ_q is the standard deviation of the excess degree distribution of the network, $q(k)$.

Since the expectation of the distribution $q(k)$ is given by $\sum_k kq(k)$, the assortativity of a network can also be written as:

$$r = \frac{1}{\sigma_q^2}\left[\left(\sum_{jk} jke_{j,k}\right) - \mu_q^2\right] \tag{2}$$

where μ_q is the expectation of the distribution.

Local assortativity was motivated in Piraveenan et al. (2008) by calculating the contribution of each node to the above correlation coefficient. Therefore, the sum over all nodes is equal to network assortativity. Formally, local assortativity of a given node v was derived in Piraveenan et al. (2008) to be:

$$\rho_v = \frac{\alpha_v - \beta_v}{\sigma_q^2} = \frac{\left(j+1\right)\left(j\overline{k} - \mu_q^2\right)}{2M\sigma_q^2} \tag{3}$$

where j is the node's excess degree; $\overline{k}$ is the average excess degree of its neighbours, $\sigma_q \neq 0$; the contribution α_v of the node v to the first term in (2), that is, to the sum $\sum_{jk} jke_{j,k}$ is

$$\alpha_v = \left(j+1\right)\frac{j\overline{k}}{2M} \tag{4}$$

and the contribution β_v of the node v to the second term in (2), that is, to μ_q^2 is

$$\beta_v = \left(j+1\right)\frac{\mu_q^2}{2M} \tag{5}$$

It can be shown that local assortativity satisfies the summation property:

$$r = \sum_{v=1}^{N} \rho_v \tag{6}$$

In particular,

$$\sum_{jk} jke_{j,k} = \sum_{v=1}^{N} \alpha_v \quad \text{and} \quad \mu_q^2 = \sum_{v=1}^{N} \beta_v \tag{7}$$

While the component α_v captures the precise contribution of each node to the term $\sum_{jk} jke_{j,k}$, the component β_v represents the contribution of each node to the term μ_q^2 with an imprecise scaling. Specifically, the scaling factor $(j+1)/2M$ in (5) is the correct scaling factor for μ_q, rather than μ_q^2, and hence, β_v has a bias towards low-degree nodes (Piraveenan et al., 2010).

Unbiased local assortativity

The derivation of the correctly scaled (and hence, unbiased) contribution, $\hat{\beta}_v$, of a given node v to the term μ_q^2 is shown in Appendix A, yielding

$$\hat{\beta}_v = \left(j+1\right)\frac{j\mu_q}{2M} \tag{8}$$

where j is the node's excess degree, as before. Hence, the unbiased representation of local assortativity is given by

$$\hat{\rho}_v = \frac{\alpha_v - \hat{\beta}_v}{\sigma_q^2} = \frac{j\left(j+1\right)\left(\overline{k} - \mu_q\right)}{2M\sigma_q^2} \tag{9}$$

Let us compare the unbiased local assortativity $\hat{\rho}_v$ with that defined by (3). Specifically, the sign of the local assortativity (positive or negative) is determined by the difference between the average excess degree ($\overline{k}$) of the neighbours and the global average excess degree (μ_q). If the neighbours' average is higher, then the node is assortative. If the global average is higher, the node is disassortative. Therefore, the local assortativity can also be defined as a scaled difference between the average excess degree of the node's neighbours and the global average excess degree (the scale factor is proportional to the product of the node's degree and excess degree). In other words, a node tends to be locally assortative if it is surrounded by nodes with comparatively high degrees — hence, even though local assortativity is a property of a node, it is influenced by a node's 'locality', or neighbourhood.

The only difference between β_v defined by (5) and the unbiased $\hat{\beta}_v$ defined by (8) is that the network's mean μ_q, which is constant across nodes, is replaced by j, the node's excess degree. This means that there is a bias in the term (3) which favours low-degree nodes (with smaller j) and disfavours hubs (with larger j). In summary,

1. both the β_v proposed in Piraveenan et al. (2008) and $\hat{\beta}_v$ corrected in Piraveenan et al. (2010) adhere to summation rule $\sum \beta_v = \sum \hat{\beta}_v = \mu_q^2$.

2. $\hat{\beta}_v$ is higher for hubs and lower for low-degree nodes compared to β_v.

We will utilise average local assortativity plotted against degree. Average local assortativity $\overline{\rho}(d)$ can be calculated by averaging local assortativity quantities of all nodes with a

given degree d. For example, the difference between biased local assortativity profile $\rho(d)$ and unbiased local assortativity $\hat{\rho}(d)$ for H. pylori Protein Protein Interaction network is shown in the Appendix B.

We point out that local assortativity is a quantity that involves both degree and average (neighbour) degree, and as a result, the local assortativity profiles clearly differ from average degree profiles. In particular, an average degree profile always contains positive values that increase with the degree, while local assortativity profiles may contain both positive or negative values, increasing or decreasing with the degree.

Local assortativity in canonical networks

Regular lattice

For a lattice network each node has the same degree and excess degree, therefore the variance of the excess degree distribution is 0. Since there is only one type of nodes, the network is perfectly assortative ($r = 1$) and the local assortativity of all nodes is $1/N$, as shown in Figure 1.

Star network

A star graph is another extreme example of complex networks in terms of topology. In a pure star graph, any given link has a degree-one node at one end, with the excess degree zero. It can be shown that a star graph is perfectly disassortative ($r = -1$). Furthermore, any node in the star graph has either its excess degree as zero, or all of its neighbours' excess degrees as zero. It is easy to see that the term represented by equation (4) reduces to zero in all cases. Thus, the local assortativity reduces to

$$\rho = -\frac{j+1}{2M} \frac{\mu_q}{\sigma_q^2} \tag{10}$$

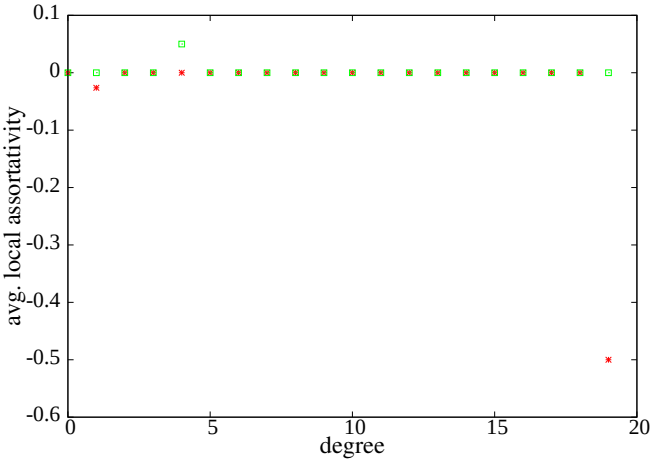


Figure 1: Local assortativity distribution, $\overline{\rho}(k)$ vs k, of a regular lattice with four nodes connecting to each node (squares), and of a star graph (stars). Network size in both cases is $N = 20$.

Figure 1 shows the local assortativity distribution for a pure star graph: the central node is much more locally-disassortative, as it connects with many dissimilar nodes, whereas the low-degree nodes are less locally-disassortative since they connect to only one dissimilar node.

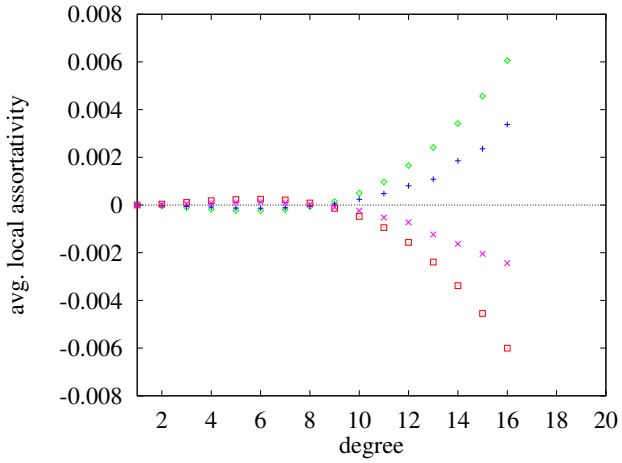


Figure 2: Local assortativity profile of scale-free networks ($N = 1000$ and $\gamma = 2.1$) with $r = 1.0$ ('◇'), $r = 0.5$ ('+') , $r = -0.5$ ('×') and $r = -1.0$ ('□').

Classification of networks using unbiased local assortativity profiles

In this section we aim to classify both model and real-world networks using the unbiased local assortativity. Since local assortativity is a property of a node, it is possible to construct local assortativity distributions of networks (Piraveenan et al., 2008).

We begin the analysis by constructing model Barabási–Albert scale-free networks (Albert and Barabasi, 2002) of various assortativity levels and observing their local assortativity profiles. Specifically, we use the Assortative Preferential Attachment method (APA) (Piraveenan et al., 2007) to control the level of assortativity. Some of the results are shown in Figure 2 for network size $N = 1000$ and power law exponent $\gamma = 2.1$.

We could observe from Figure 2 that the profiles are symmetric with respect to the degree axis when assortativity is varied from $r = 1.0$ to $r = -1.0$ while other network parameters are kept constant. We also note that (i) globally assortative networks have assortative hubs and disassortative low-degree nodes, and (ii) globally disassortative networks have disassortative hubs and assortative low-degree nodes. That is, the overall assortativity of the network is matched by that of the hubs. Thus, we are able classify the constructed model networks as either (i) assortative networks with assortative hubs, or (ii) disassortative networks with disassortative hubs. This is not surprising. However, one may ask whether there are also any disassortative networks with assortative hubs, as proposed in Piraveenan et al. (2008). To

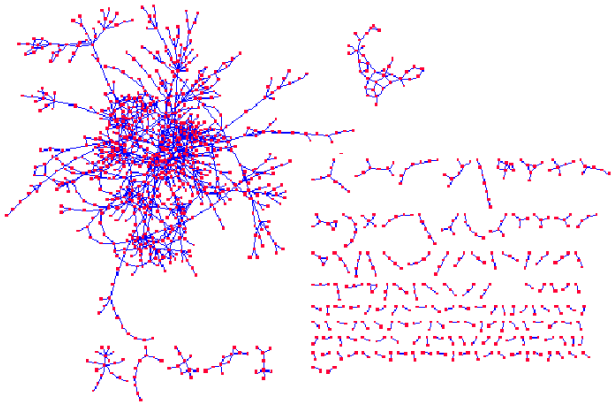


Figure 3: Example of an assortative network with assortative hubs. H. sapiens metabolic network; $N = 1288$, $\gamma \approx 2.32$, $r = 0.382$.

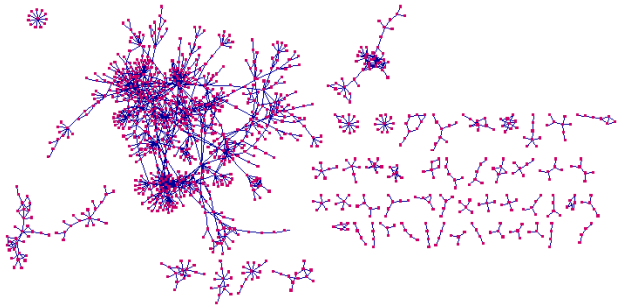


Figure 4: Example of an assortative network with disassortative hubs. H. sapiens Protein Protein Interaction network; $N = 1529$, $\gamma \approx 2.1$, $r = 0.075$.

answer this question, let us look at the model network given in Figure 5. This network is made up of a number of interconnected star-like subnetworks. Each subnetwork has a core of hubs that are densely connected to one another: this is the 'rich club phenomenon' (Zhou and Mondragón, 2004; Colizza et al., 2006). The rest of the subnetwork seems to have mostly disassortative connections. The subnetworks are then linked together with hub-to-hub connections, further reinforcing the rich-club phenomenon. The overall assortativity of the network is $r = -0.109$. However, as shown in Figure 9, the hubs are assortative. The embedded subnetworks pattern can be repeated on larger scales, retaining the assortative hubs with higher and higher degrees, while keeping the overall disassortativity. This example represents a third class, demonstrating that it is possible to have disassortative networks with assortative hubs.

The real-world networks we studied included most recent metabolic networks (KEGG database), citation networks, Protein-Protein Interaction (PPI) networks, food-webs, and Internet AS level networks among others. A list of the networks we analysed is shown in Table 1. We were able to

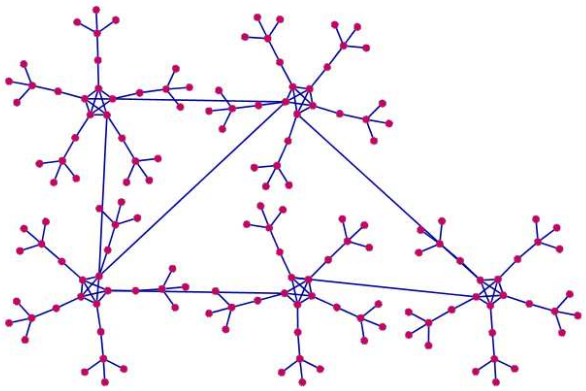


Figure 5: Example of a disassortative network with assortative hubs. A model network with $N = 150$, $r = -0.109$.

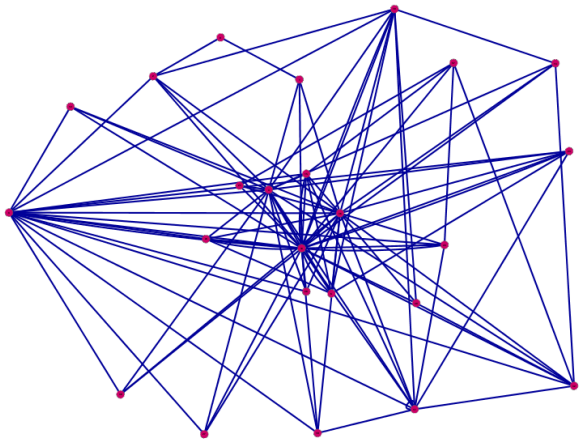


Figure 6: Example of a disassortative network with disassortative hubs. Crystal River D foodweb, $N = 24$, $r = -0.467$.

observe the following from our analysis.

Firstly, as in the case of model APA networks, some real-world assortative networks have assortative hubs (e.g., Figure 7; most other metabolic networks showed similar profiles). Also many real-world disassortative networks have disassortative hubs, e.g., one such food-web is shown in Figure 10. However, other assortative networks exhibit disassortative hubs, such as the PPI networks of H. sapiens shown in Figure 8. A number of other PPI networks displayed a similar profile. These networks represent the fourth class, namely the assortative networks with disassortative hubs.

Therefore, we can identify four classes of complex networks, namely: (i) assortative networks with assortative hubs, (ii) assortative networks with disassortative hubs, (iii) disassortative networks with disassortative hubs, (iv) disassortative networks with assortative hubs.

There are several examples of real-world networks for each of the first three cases, and we have shown representative examples in Figures 7, 8, and 10 respectively . We did

Network	assortativity r	class
Human metabolic (KEG, 2009)	0.382	assortative with assortative hubs
Chimpanzee metabolic (KEG, 2009)	0.398	assortative with assortative hubs
Rhesus monkey metabolic (KEG, 2009)	0.363	assortative with assortative hubs
Astro physics citation (Newman, 2009)	0.276	assortative with assortative hubs
Cond. mat. 2003 citation (Newman, 2009)	0.178	assortative with assortative hubs
Cond. mat. 2005 citation (Newman, 2009)	0.186	assortative with assortative hubs
Hep theory citation (Newman, 2009)	0.293	assortative with disassortative hubs
Net science citation (Newman, 2009)	0.46	assortative with disassortative hubs
H. sapiens PPI (PPI, 2009)	0.075	assortative with disassortative hubs
E. coli PPI (PPI, 2009)	0.056	assortative with disassortative hubs
Internet AS 1998 (CAI, 2009)	-0.198	disassortative with disassortative hubs
Internet AS 2008 (CAI, 2009)	-0.198	disassortative with disassortative hubs
Fruitfly PPI (PPI, 2009)	-0.21	disassortative with disassortative hubs
H. pylori PPI (PPI, 2009)	-0.235	disassortative with disassortative hubs
Mouse PPI (PPI, 2009)	-0.057	disassortative with disassortative hubs
Crystal River C (Batagelj and Mrvar, 2006)	-0.334	disassortative with disassortative hubs
Crystal River D (Batagelj and Mrvar, 2006)	-0.467	disassortative with disassortative hubs
Lower Chesapeake (Batagelj and Mrvar, 2006)	-0.391	disassortative with disassortative hubs
Scimet collaboration (Batagelj and Mrvar, 2006)	-0.03	disassortative with disassortative hubs
Smart grid collaboration (Batagelj and Mrvar, 2006)	-0.193	disassortative with disassortative hubs

Table 1: The networks studied and their classification.

not find any example of the fourth case among the networks we studied, however we have demonstrated that in theory such networks could exist, as shown in the profile in Figure 9, and real-world examples may yet be found as the range of networks studied is expanded.

We show the corresponding networks for each example in Figures 3, 4, 5, and 6 respectively. Note that the networks with assortative hubs and disassortative hubs are not always visually distinguishable, however, the local assortativity profiles are able to highlight an important topological difference in them.

While a detailed analysis of the classification results in the context of biological networks is out of scope for the paper, we briefly mention some possibilities. Assortative metabolic networks may have assortative hubs due to optimality in flux balance (Varma and Palsson, 1994): most metabolic reactions form chains ending with a regulatory decision in a hub, and the connections between hubs may optimise metabolic requirements for growth, utilising different pathways.

The hubs in food-webs could be disassortative because the separation between hubs plays an evolutionary role, maintaining sustainable food chains.

It is somewhat more complicated why the PPI networks that are assortative overall have disassortative hubs. On the one hand, many individual proteins may form a multi-protein complex, and some of the proteins can participate in the formation of a variety of different protein complexes. Such high-interacting proteins are likely to be locally assortative. On the other hand, the anticorrelation in the node degree of connected nodes, i.e., the tendency of highly interacting nodes to be connected to low-interacting ones, has been reported previously (Maslov and Sneppen, 2002; Spirin and Mirny, 2003). In particular, Maslov and Sneppen argued that "this effect decreases the likelihood of cross talk between different functional modules of the cell and increases the overall robustness of a network by localizing effects of deleterious perturbations" (Maslov and Sneppen, 2002). These two alternatives are related to the distinction between protein complexes and functional modules (Spirin and Mirny, 2003): protein complexes are groups of proteins that interact with each other at the same time and place, forming a single multimolecular machine, while functional modules consist of proteins that participate in a particular cellular process while binding each other at a different time and place. Disassortative hubs are likely to be the proteins within functional modules. In addition, one may point out that there are artefacts of the high-throughput methods used to discover the interactions that may lead to low interaction coverage of certain protein types and obscure local assortativity profiles (Shoemaker and Panchenko, 2007a,b).

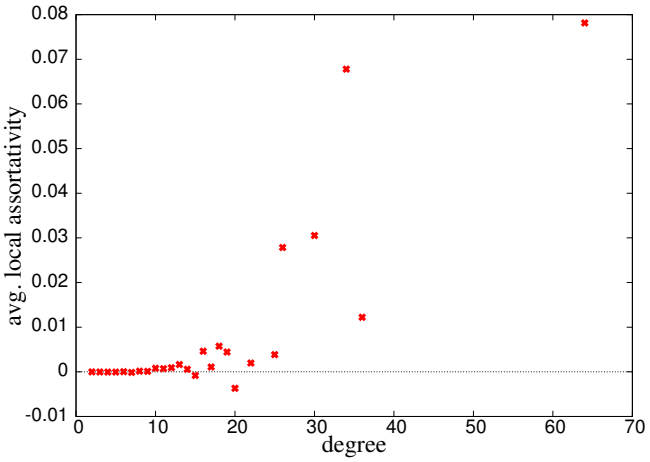


Figure 7: Local assortativity profile of H. sapiens metabolic network; $N = 1288$, $\gamma \approx 2.32$, $r = 0.382$.

Conclusions

We proposed an unbiased formulation for local assortativity in complex networks, and analysed the local assortativity profiles of some model and real-world networks in terms of this new formulation. We showed that a node's local assortativity is proportional to the difference between the average excess degree of its neighbours and the network's overall average excess degree. Specifically, a node is locally assortative if its neighbours have comparatively (i.e., compared with all nodes in the network) higher degrees. It is important to realise that the nodes with the highest local assortativity differ in general from the largest hubs (the nodes with the highest degrees).

Analyzing a range of model and real-world networks, we observed four classes of networks, namely: (i) assortative networks with assortative hubs, (ii) assortative networks with disassortative hubs, (iii) disassortative networks with disassortative hubs, and (iv) disassortative networks with assortative hubs. Real-world examples for the first three classes were identified, and a model network was constructed as an example for the fourth class.

The local assortativity profiles provide an additional quantitative tool for network analysis. These profiles highlight important topological differences in otherwise seemingly indistinguishable networks. This may help in studying diverse network properties and dynamics: e.g., (a) network growth may be modelled in such a way that the grown networks not only satisfy global characteristics, but also agree with required local assortativity profiles (Piraveenan et al., 2009b); (b) network robustness may be analysed in terms of an attack targeting the nodes with higher local assortativity; (c) motifs within networks can be studied via their average local assortativity, etc. One avenue for future work is to define local assortativity in directed networks, and apply this definition to directed biological networks, studying the role of the nodes with the highest local assortativity in regulatory processes (e.g., reaction cascades).

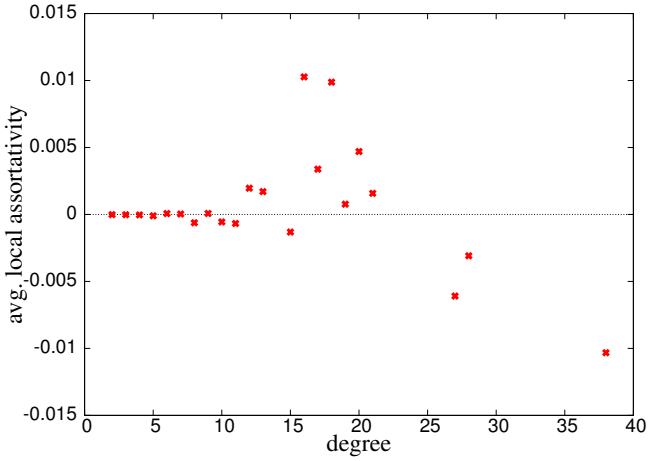


Figure 8: Local assortativity profile of H. sapiens Protein-Protein Interaction network; $N = 1529$, $\gamma \approx 2.1$, $r = 0.075$.

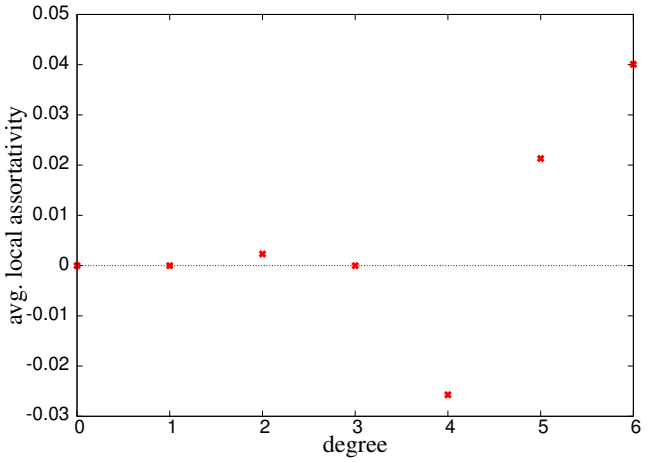


Figure 9: Local assortativity profile of the network shown in Figure 5; $N = 150$, $r = -0.109$.

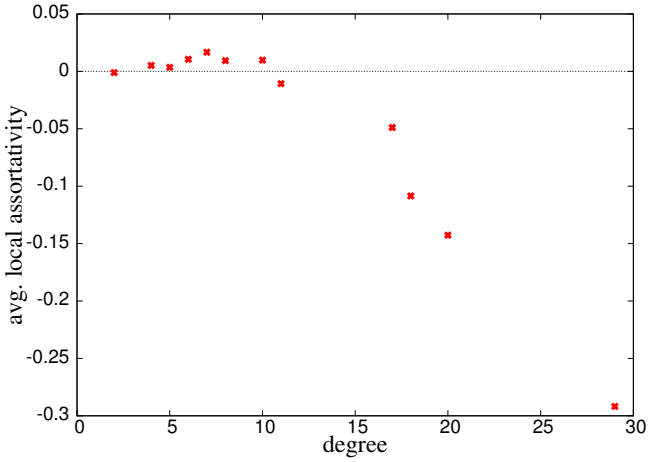


Figure 10: Local assortativity profile Chrystal River D food-web; $N = 24$, $r = -0.467$.

Acknowledgements

We thank Prof. G. Q. Zhang from the Institute of Computing Technology, Chinese Academy of Sciences for posing the question regarding the contribution of nodes to μ_q^2. We are also grateful to anonymous referees who provided useful suggestions on possible interpretations of the classification results in the context of biological networks.

Appendix A

To derive the contribution of each node to μ_q^2 we first look at the following equivalent definitions of μ_q:

$$\mu_q = \frac{1}{2M}\sum_{m=1}^{M} k_m \tag{11}$$

$$\mu_q = \frac{1}{2M}\sum_{v=1}^{N} k_v(1+k_v) \tag{12}$$

where k is excess degree, m is a given edge and v is a given node of the network. We are especially interested in the latter form (12) since it makes it obvious what each node contributes to the term μ_q. It follows that

$$\mu_q = \frac{1}{2M}\left(\sum_{v=1}^{N} k_v + \sum_{v=1}^{N} {k_v}^2\right) \tag{13}$$

yielding

$$\mu_q^2 = \frac{1}{4M^2}\left((\sum_{v=1}^{N} k_v)^2 + (\sum_{v=1}^{N} {k_v}^2)^2 + 2\sum_{v=1}^{N} k_v \sum_{v=1}^{N} {k_v}^2\right) \tag{14}$$

Now, let us consider a single node (without loss of generality, let it be the node 1 with excess degree k_1), and its contribution to each of the three summation terms in the expression above. Considering the first summation term, excess degree k_1 contributes to it as follows:

$${k_1}^2 + 2(k_1k_2 + k_1k_3 + ... + k_1k_N) \tag{15}$$

Among these, terms such as $2k_1k_j$ have to be 'divided' between node 1 and node j respectively. These are multiplication terms, and we assume that an equal division is appropriate. Therefore, the contribution of node 1 is:

$${k_1}^2 + (k_1k_2 + k_1k_3 + ... + k_1k_N) = k_1\sum_{j=1}^{N} k_j \tag{16}$$

Considering the second summation term in (14), we observe that the contribution of node 1 is ${k_1}^2\sum_{j=1}^{N} {k_j}^2$. Let us analyse the contribution of node 1 to the third summation term in (14). The third summation term is given by

$$2\sum_{i=1}^{N} k_i \sum_{j=1}^{N} {k_j}^2 = 2\left(k_1 + \sum_{i=2}^{N} k_i\right)\left(k_1^2 + \sum_{j=2}^{N} {k_j}^2\right) \tag{17}$$

where i, j are node indices. The contribution of node 1 to the third term is obtained by dividing terms such as $2k_1k_j$ between node 1 and node j respectively:

$$2k_1^3 + k_1^2\sum_{i=2}^{N} k_i + k_1\sum_{j=2}^{N} {k_j}^2 = k_1\sum_{j=1}^{N} {k_j}^2 + {k_1}^2\sum_{j=1}^{N} k_j \tag{18}$$

Therefore, the total contribution of node 1, β_1, to μ_q^2 is:

$$\beta_1 = \frac{k_1\sum_{j=1}^{N} k_j + {k_1}^2\sum_{j=1}^{N} {k_j}^2 + k_1\sum_{j=1}^{N} {k_j}^2 + {k_1}^2\sum_{j=1}^{N} k_j}{4M^2} \tag{19}$$

This can be further regrouped as

$$\beta_1 = \frac{k_1 + {k_1}^2}{4M^2}\left(\sum_{j=1}^{N} k_j + \sum_{j=1}^{N} {k_j}^2\right) \tag{20}$$

Using equation (13) for μ_q, this can be reduced to:

$$\beta_1 = \frac{k_1 + {k_1}^2}{2M}\mu_q \tag{21}$$

Hence, the contribution of a node v to μ_q^2 is given by:

$$\hat{\beta}_v = (j+1)\frac{j\mu_q}{2M} \tag{22}$$

where j is the excess degree of the node v. Thus, local assortativity is given by

$$\hat{\rho}_v = \frac{\alpha_v - \hat{\beta}_v}{\sigma_q^2} = \frac{j\,(j+1)\,(\overline{k} - \mu_q)}{2M\sigma_q^2} \tag{23}$$

Appendix B

The difference between the biased local assortativity profile $\rho(d)$, defined by (3), and the unbiased local assortativity $\hat{\rho}(d)$, defined by (9), for H. pylori Protein Protein Interaction network is shown in Figure 11. It is evident that $\hat{\rho}(d) < \rho(d)$ for the hubs, and more importantly, the hubs are now locally disassortative.

References

(2009). The Cooperative Association for Internet Data Analysis. http://www.caida.org/home/.

(2009). Database of Interacting Proteins. http://dip.doe-mbi.ucla.edu/dip/Main.cgi.

(2009). The Kyoto Encyclopaedia for Genes and Genomes. http://www.genome.jp/kegg/.

Albert, R., Jeong, H., and Barabasi, A.-L. (1999). Diameter of the world-wide web. *Nature*, 401:130–131.

Albert, T. R. and Barabasi, A.-L. (2002). Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74:47–97.

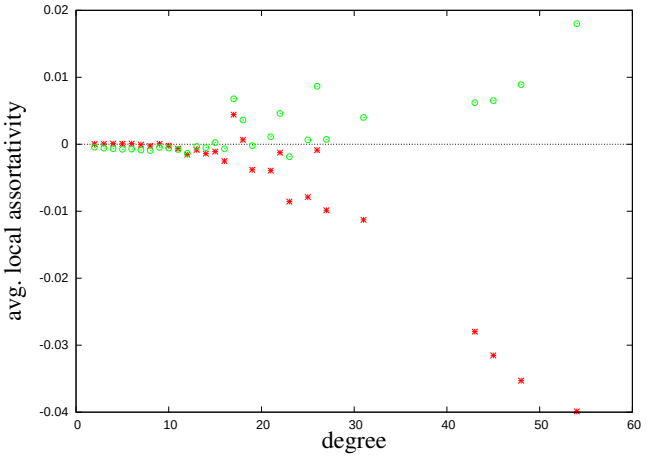


Figure 11: H. pylori Protein Protein Interaction network. Local assortativity profiles ρ ('$\circ$') and $\hat{\rho}$ ('$*$'); $N = 714$, $\gamma \approx 2.54$, $r = -0.235$.

Alon, U. (2007). *Introduction to Systems Biology: Design Principles of Biological Circuits*. Chapman and Hall, London.

Bagler, G. and Sinha, S. (2007). Assortative mixing in protein contact networks and protein folding kinetics. *Bioinformatics*, 23(14):1760–1767.

Batagelj, V. and Mrvar, A. (2006). Pajek datasets. http://vlado.fmf.uni-lj.si/pub/networks/data/.

Callaway, D. S., Hopcroft, J. E., Kleinberg, J. M., Newman, M. E., and Strogatz, S. H. (2001). Are randomly grown graphs really random? *Phys Rev E Stat Nonlin Soft Matter Phys*, 64(4 Pt 1).

Colizza, V., Flammini, A., Serrano, M. A., and Vespignani, A. (2006). Detecting rich-club ordering in complex networks. *Nature Physics*, 2:110–115.

Faloutsos, M., Faloutsos, P., and Faloutsos, C. (1999). On power-law relationships of the internet topology. *Computer Communications Review*, 29:251–262.

Lizier, J. T., Piraveenan, M., Pradhana, D., Prokopenko, M., and Yaeger, L. S. (2009). Functional and structural topologies in evolved neural networks. In *Advances in Artificial Life: Tenth European Conference on Artificial Life (ECAL '09)*, volume 5777-5778 of *LNCS/LNAI*. Springer.

Maslov, S. and Sneppen, K. (2002). Specificity and stability in topology of protein networks. *Science*, 296(5569):910–913.

Newman, M. E. (2002). Assortative mixing in networks. *Phys Rev Lett*, 89(20):208701.

Newman, M. E. (2003). Mixing patterns in networks. *Phys Rev E*, 67(2):026126.

Newman, M. E. (2009). Newman network datasets. http://www-personal.umich.edu/~mejn/netdata/.

Palsson, B. O. (2006). *Systems Biology: Properties of Reconstructed Networks*. Cambridge University Press, 1 edition.

Piraveenan, M., Prokopenko, M., and Zomaya, A. (2007). Information-cloning of scale-free networks. In e Costa, F. A., Rocha, L. M., Costa, E., and I. Harvey, A. C., editors, *Advances in Artificial Life: 9th European Conference on Artificial Life (ECAL-2007), Lisbon, Portugal*, volume 4648 of *Lecture Notes in Artificial Intelligence*, pages 925–935. Springer.

Piraveenan, M., Prokopenko, M., and Zomaya, A. Y. (2008). Local assortativeness in scale-free networks. *Europhysics Letters*, 84(2):28002.

Piraveenan, M., Prokopenko, M., and Zomaya, A. Y. (2009a). Assortativeness and information in scale-free networks. *European Physical Journal B*, 67(3):291–300.

Piraveenan, M., Prokopenko, M., and Zomaya, A. Y. (2009b). Local assortativity and growth of Internet. *European Physical Journal B*, 70(2):275–285.

Piraveenan, M., Prokopenko, M., and Zomaya, A. Y. (2010). Local assortativeness in scale-free networks — addendum. *Europhysics Letters*, 89(4):49901.

Prokopenko, M., Boschetti, F., and Ryan, A. J. (2009). An information-theoretic primer on complexity, self-organization, and emergence. *Complexity*, 15(1):11–28.

Shoemaker, B. A. and Panchenko, A. R. (2007a). Deciphering protein-protein interactions. part I. experimental techniques and databases.. *PLoS computational biology*, 3(3):e42+.

Shoemaker, B. A. and Panchenko, A. R. (2007b). Deciphering protein-protein interactions. part II. computational methods to predict protein and domain interaction partners. *PLoS computational biology*, 3(4):e43+.

Solé, R. V. and Valverde, S. (2004). Information theory of complex networks: on evolution and architectural constraints. In Ben-Naim, E., Frauenfelder, H., and Toroczkai, Z., editors, *Complex Networks*, volume 650 of *Lecture Notes in Physics*. Springer.

Spirin, V. and Mirny, L. A. (2003). Protein complexes and functional modules in molecular networks. *Proceedings of the National Academy of Sciences of the United States of America*, 100(21):12123–12128.

Varma, A. and Palsson, B. O. (1994). Stoichiometric flux balance models quantitatively predict growth and metabolic by-product secretion in wild-type escherichia coli w3110. *Applied and Environmental Microbiology*, 60(10):3724–3731.

Vázquez, A. (2003). Growing network with local rules: Preferential attachment, clustering hierarchy, and degree correlations. *Physical Review E*, 67(5):056104.

Zhou, J., Xu, X., Zhang, J., Sun, J., Small, M., and Lu, J. (2008). Generating an assortative network with a given degree distribution. *International Journal of Bifurcation and Chaos*, 18(11):3495–3502.

Zhou, S. and Mondragón, R. J. (2004). The rich-club phenomenon in the internet topology. *IEEE Comm. Lett.*, 8:180–182.

Network Complexity of Foodwebs

Russell Standish
School of Mathematics and Statistics, University of New South Wales

Abstract

In previous work, I have developed an information theoretic complexity measure of networks. When applied to several real world food webs, there is a distinct difference in complexity between the real food web, and randomised control networks obtained by shuffling the network links. One hypothesis is that this complexity surplus represents information captured by the evolutionary process that generated the network.

In this paper, I test this idea by applying the same complexity measure to several well-known artificial life models that exhibit ecological networks: Tierra, EcoLab and Webworld. Contrary to what was found in real networks, the artificial life generated foodwebs had little information difference between itself and randomly shuffled versions.

Introduction

In Standish (2005), I developed a method for computing the information complexity of a network. In Standish (2010a), I refined and generalised the method to overcome a problem with higher complexity values of empty and full networks relative to partially filled networks of the same degree, as well as taking account of link weights. Coupled with some new algorithms for computing automorphism group size, this network complexity measure is practical for networks of several thousand nodes.

In Standish (2010a), I studied several published datasets of natural networks, including a number of foodwebs available from the Pajek website, and the neural network of *C. elegans* (see Table 1). In most cases, these networks exhibited significantly heightened complexity values compared with those of control networks obtained by shuffling the links in a random fashion. This leads to the hypothesis that evolutionary processes tend to produce networks with a *complexity surplus* (Δ) compared with random assembly processes.

In this work, I apply the same methods to networks created by artificial life evolutionary systems, in particular the interaction network of Tierra (Ray, 1991) and the foodwebs of EcoLab (Standish, 1994) and Webworld (Caldarelli et al., 1998).

Complexity as Information

The notion of using information content as a complexity measure is fairly simple. In most cases, there is an obvious *prefix-free* representation language within which descriptions of the objects of interest can be encoded. There is also a classifier of descriptions that can determine if two descriptions correspond to the same object. This classifier is commonly called the *observer*, denoted $O(x)$.

To compute the complexity of some object x, count the number of equivalent descriptions $\omega(\ell, x)$ of length ℓ that map to the object x under the agreed classifier. Then the complexity of x is given in the limit as $\ell \to \infty$:

$$\mathcal{C}(x) = \lim_{\ell\to\infty} \ell \log N - \log \omega(\ell, x) \tag{1}$$

where N is the size of the alphabet used for the representation language.

Because the representation language is prefix-free, every description y in that language has a unique prefix of length $s(y)$. The classifier does not care what symbols appear after this unique prefix. Hence $\omega(\ell, O(y)) \geq N^{\ell - s(y)}$. As ℓ increases, ω must increase as fast, if not faster than N^ℓ, and do so monotonically. Therefore $C(O(y))$ decreases monotonically with ℓ, but is bounded below by 0. So equation (1) converges.

To use this formalism with networks, we need to fix two things: how to decide when two networks are identical, and a prefix-free representation language, which will be used to count the representations of a given network. In this context, ignoring any link weights, two networks are considered identical if the nodes of one can be placed over the nodes of the second one, such that the links correspond exactly. They are topologically identical. We ignore any labels on the nodes or links.

Network bitstring representation

To represent the network as a bitstring, we need to store the node count (n) and link count (l), as well as representation of the adjacency matrix. The initial part of the string has $w = \lceil \log_2 n \rceil$ '1' bits, followed by a single '0' stop bit. Following that are w bits representing the value of n in binary.

Dataset	nodes	links	$\mathcal{C}$	$e^{\langle \ln \mathcal{C}_{\mathrm{ER}} \rangle}$	$\Delta = \mathcal{C} - e^{\langle \ln \mathcal{C}_{\mathrm{ER}} \rangle}$	$\frac{\lvert \ln \mathcal{C} - \langle \ln \mathcal{C}_{\mathrm{ER}} \rangle \rvert}{\sigma_{\mathrm{ER}}}$
celegansneural	297	2345	442.7	251.6	191.1	29
celegansmetabolic	453	4050	25421.8	25387.2	34.6	∞
lesmis	77	508	199.7	114.2	85.4	24
adjnoun	112	850	3891	3890	0.98	∞
yeast	2112	4406	33500.6	30218.2	3282.4	113.0
baydry	128	2138	126.6	54.2	72.3	22
baywet	128	2107	128.3	51.0	77.3	20
cypdry	71	641	85.7	44.1	41.5	13
cypwet	71	632	87.4	42.3	45.0	14
gramdry	69	911	47.4	31.6	15.8	10
gramwet	69	912	54.5	32.7	21.8	12
Chesapeake	39	177	66.8	45.7	21.1	10.4
ChesLower	37	178	82.1	62.5	19.6	10.6
ChesMiddle	37	208	65.2	48.0	17.3	9.3
ChesUpper	37	215	81.8	60.7	21.1	10.2
CrystalC	24	126	31.1	24.2	6.9	6.4
CrystalD	24	100	31.3	24.2	7.0	6.2
Everglades	69	912	54.5	32.7	21.8	11.8
Florida	128	2107	128.4	51.0	77.3	20.1
Maspalomas	24	83	70.3	61.7	8.6	5.3
Michigan	39	219	47.6	33.7	14.0	9.5
Mondego	46	393	45.2	32.2	13.0	10.0
Narragan	35	219	58.2	39.6	18.6	11.0
Rhode	19	54	36.3	30.3	6.0	5.3
StMarks	54	354	110.8	73.6	37.2	16.0

Table 1: Complexity values of several freely available network datasets, as reported in Standish (2010a). For each network, the number of nodes and links are given, along with the computed complexity $\mathcal{C}$. In the fourth column, the original network is shuffled 1000 times, and the logarithm of the complexity is averaged ($\langle \ln \mathcal{C}_{\mathrm{ER}} \rangle$). The fifth column gives the difference between these two values, which represents the information content of the specific arrangement of links. The final column gives a measure of the significance of this difference in terms of the number of standard deviations ("sigmas") of the distribution of shuffled networks. In two examples, the distribution of shuffled networks had zero standard deviation, so ∞ appears in this column.

Knowing the value of n, the number of bits needed to represent l is $\lceil \log_2 L \rceil$, where $L = (n(n-1)/2)$ so l is stored in a field of that width.

For the final part of the string, the linkfield, we can represent the adjacency matrix such that a '1' bit in position $i(n-1)+j$-th represents a link from node i to j if $j < i$ or from i to $j+1$ if $j > i$, where nodes are numbered $0 \ldots n-1$, $i < n$ and $j < n-1$. However, this representation is not efficient — given l, there must be exactly l '1' bits in the linkfield, ie it is one of the permutations of l '1' bits and $L-l$ '0' bits. We can enumerate the $\begin{pmatrix} L \\ l \end{pmatrix}$ permutations, and choose the rank of our linkfield in the enumeration as the encoding of the linkfield. This is known as rank encoding (Myrvold and Ruskey, 2001). One of the effects of choosing this encoding is that both an empty and a full network have just one possible linkfield, so will have a rank encoding of 0, representable in 0 bits, as we already know whether a network is empty or full from the values of n and l. Hence, the full and empty networks are the simplest networks for given n and l.

Weighted links

Whilst the information contained in link weights might be significant in some circumstances (for instance the weights of a neural network can only be varied in a limited range without changing the overall qualitative behaviour of the network), of particular theoretical interest is to consider the weights as continuous parameters connecting one network structure with another. For instance if a network X has the same network structure as A, with b links of weight 1 with a network structure B and the remaining $a-b$ links of weight w, then we would like the network complexity of X to vary smoothly between that of A and B as w varies from 1 to 0. Görnerup and Crutchfield (2008) introduced a similar measure.

The most obvious way of defining this continuous complexity measure is to start with normalised weights $\sum_i w_i = 1$. Then arrange the links in weight order, and compute the complexity of networks with just those links of weights less than w. The final complexity value of a network $X = N \times L$, where N is the set of nodes, and L the set of links with associated weights $w_i, \exists i \in L$, is obtained by integrating:

$$\mathcal{C}(X = N \times L) = \int_0^1 \mathcal{C}(N \times \{i \in L : w_i < w\}) dw \quad (2)$$

Obviously, since the integrand is a stepped function, this is computed in practice by a sum of complexities of partial networks.

Counting the representations

In principle, one could compute the complexity of a network by enumerating all bitstrings for a given n and l, and counting the number of bitstrings that represent the target network. However, this algorithm is highly combinatoric, and only really feasible for small networks. However, the number of representations can also be computed by dividing the total number of possible renumberings of the nodes ($N!$) by the size of the automorphism group, for which several practical algorithms exist (McKay, 1981; Standish, 2010b; Darga et al., 2008). Even though each of these algorithms is NP-complete, in practice they tend to perform quite well for networks up to several thousands of nodes. Where each algorithm performs poorly, one of the other algorithms performs well, so a hybrid algorithm that runs each algorithm in parallel, and returning the result of the first algorithm to complete, performs extremely well.

ALife models

Tierra

Tierra (Ray, 1991) is a well known artificial life system in which self reproducing computer programs written in an assembly-like language are allowed to evolve. The programs, or *digital organisms* can interact with each via template matching operations, modelled loosely on the way proteins interact in real biological systems. A number of distinct strategies evolve, including parasitism, where organisms make use of another organism's code and hyperparasitism where an organism sets traps for parasites in order to steal their CPU resources. At any point in time in a Tierra run, there is an interaction network between the species present, which is the closest thing in the Tierra world to a foodweb.

Tierra is an aging platform, with the last release (v6.02) having been released more than six years ago. For this work, I used an even older release (5.0), for which I have had some experience in working with. Tierra was originally written in C for an environment where ints were 16 bits and long ints 32 bits. This posed a problem for using it on the current generation of 64 bit computers, where the word sizes are doubled. Some effort was needed to get the code 64 bit clean. Secondly a means of extracting the interaction network was needed. Whilst Tierra provided the concept of "watch bits", which recorded whether a digital organism had accessed another's genome or vice versa, it did not record which other genome was accessed. So I modified the template matching code to log the pair of genome labels that performed the template match to a file.

Having a record of interactions by genotype label, it is necessary to map the genotype to phenotype. In Tierra, the phenotype is the behaviour of the digital organism, and can be judged by running the organisms pairwise in a tournament, to see what effect each has on the other. The precise details for how this can be done is described in Standish (2003).

Having a record of interactions between phenotypes, and discarding self-self interactions, there are a number of ways of turning that record into a foodweb. The simplest way,

which I adopted, was sum the interactions between each pair of phenotypes over a sliding window of 100 million executed instructions, and doing this every 20 million executed instructions. This lead to time series of around 2000 foodwebs for each Tierra run.

In Tierra, parsimony pressure is controlled by the parameter SlicePow. CPU time is allocated proportional to genome size raised to SlicePow. If SlicePow is close to 0, then there is great evolutionary pressure for the organisms to get as small as possible to increase their replication rate. When it is one, this pressure is eliminated. In Standish (2004b), I found that a SlicePow of around 0.95 was optimal. If it were much higher, the organisms grow so large and so rapidly that they eventually occupy more than 50% of the soup. At which point they kill the soup at their next Mal (memory allocation) operation. In this work, I altered the implementation of Mal to fail if the request was more than than the soup size divided by minimum population save threshold (usually around 10). Organisms any larger than this will never appear in the Genebanker (Tierra's equivalent of the fossil record), as their population can never exceed the save threshold. This modification allows SlicePow = 1 runs to run for an extensive period of time without the soup dying.

EcoLab

EcoLab was introduced by the author as a simple model of an evolving ecosystem (Standish, 1994). The ecological dynamics is described by an n-dimensional generalised Lotka-Volterra equation:

$$\dot{n_i} = r_i n_i + \sum_j \beta_{ij} n_i n_j, \tag{3}$$

where n_i is the population density of species i, r_i its growth rate and β_{ij} the interaction matrix. Extinction is handled via a novel stochastic truncation algorithm, rather than the more usual threshold method. Speciation occurs by randomly mutating th ecological parameters (r_i and β_{ij}) of the parents, subject to the constraint that the system remain bounded (Standish, 2000).

The interaction matrix is a candidate foodweb, but has too much information. Its offdiagonal terms may be negative as well as positive, whereas for the complexity definition (2), we need the link weights to be positive. There are a number of ways of resolving this issue, such as ignoring the sign of the off-diagonal term (ie taking its absolute value), and antisymmetrising the matrix by subtracting its transpose, then using the sign of the offdiagonal term to determine the link direction.

For the purposes of this study, I chose to subtract just the negative β_{ij} terms from itself and its transpose term β_{ji}. This effects a maximal encoding of the interaction matrix information in the network structure, with link direction and weight encoding the direction and size of resource flow. The effect is as follows:

- Both β_{ij} and β_{ji} are positive (the *mutualist* case). Neither offdiagonal term changes, and the two nodes have links pointing in both directions, with weights given by the two offdiagonal terms.

- Both β_{ij} and β_{ji} are negative (the *competitive* case). The terms are swapped, and the signs changed to be positive. Again the two nodes have links pointing in both directions, but the link direction reflects the direction of resource flow.

- Both β_{ij} and β_{ji} are of opposite sign (the *predator-prey* or *parasitic* case). Only a single link exists between species i and j, whose weight is the summed absolute values of the offdiagonal terms, and whose link direction reflects the direction of resource flow.

Webworld

Webworld is another evolving ecology model, similar in some respects to EcoLab, introduced by Caldarelli et al. (1998), with some modifications described in Drossel et al. (2001). It features more realistic ecological interactions than does EcoLab, in that it tracks biomass resources. It too has an interaction matrix called a *functional response* in that model that could serve as a foodweb, which is converted to a directed weighted graph in the same way as the EcoLab interaction matrix. I used the Webworld implementation distributed with the *EcoLab* simulation platform Standish (2004a).

Results

Methods and materials

Tierra was run on a 512KB soup, with SlicePow set to 1, until the soup died, typically after some 5×10^{10} instructions have executed. Some variant runs were performed with SlicePow=0.95, and with different random number generators, but no difference in the outcome was observed.

The source code of Tierra 5.0 was modified in a few places, as described in the Tierra section of this paper. The final source code is available as tierra.5.0.D7.tar.gz from the *EcoLab* website hosted on SourceForge (http://ecolab.sf.net).

The genebanker output was processed by the eco-tierra.3.D13 code, also available from the *EcoLab* website, to produce a list of phenotype equivalents for each genotype. A function for processing the interaction log file generated by Tierra and producing a timeseries of foodweb graphs was added to Eco-tierra. The script for running this postprocessing step is process_ecollog.tcl.

The EcoLab model was adapted to convert the interaction matrix into a foodweb and log the foodweb to disk every 1000 time steps for later processing. The Webworld model

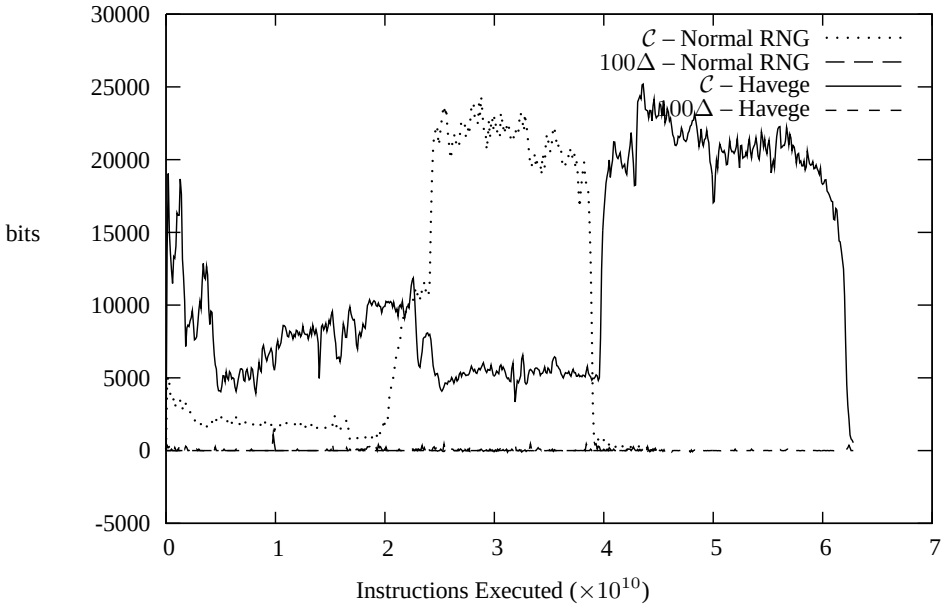


Figure 1: Complexity of the Tierran interaction network for SlicePow=0.95, and Δ, exaggerated by a factor of 100. Two different random number generators were used, Havege and the normal linear congruential generator supplied with Tierra.

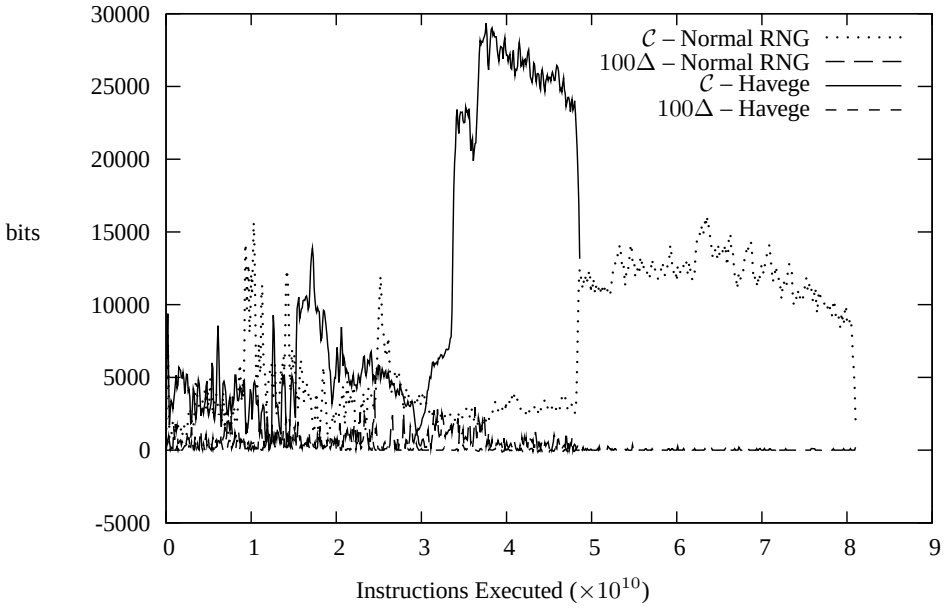


Figure 2: Complexity of the Tierran interaction network for SlicePow=1, and Δ, exaggerated by a factor of 100. Two different random number generators were used, Havege and the normal linear congruential generator supplied with Tierra.

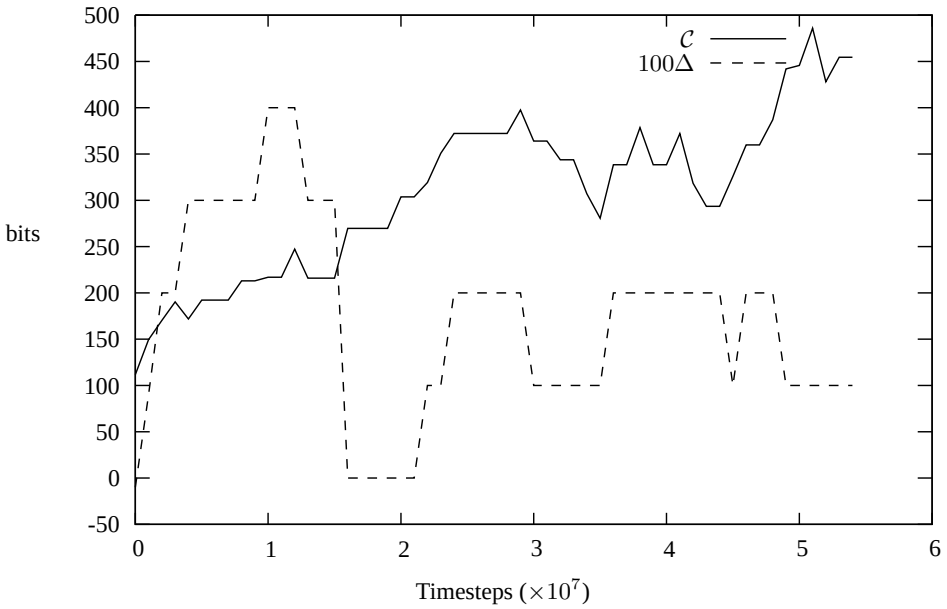


Figure 3: Complexity of EcoLab's foodweb, and Δ, exaggerated by a factor of 100, as described in the text.

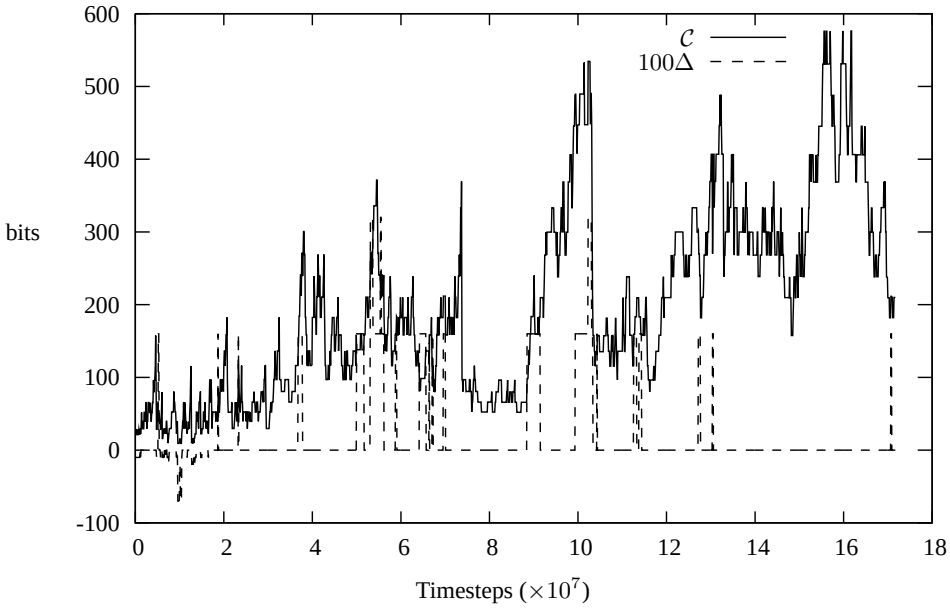


Figure 4: Complexity of Webworld's foodweb, and Δ, exaggerated by a factor of 100, as described in the text.

was adapted similarly. The model parameters were as documented in the included ecolab.tcl and webworld.tcl experiment files of the ecolab.4.D37 distribution, which is also available from the *EcoLab* website.

Finally, each foodweb, and 100 link-shuffled control versions were run through the network complexity algorithm (2). This is documented in the cmpERmodel.tcl script of ecolab.4.D37. The average and standard deviation of $\ln \mathcal{C}$ was calculated, rather than $\mathcal{C}$ directly, as the shuffled complexity values fitted a log-normal distribution better than a standard normal distribution. The difference between the measured complexity and $\exp\langle \ln \mathcal{C} \rangle$ (ie the geometric mean of the control network complexities) is what is reported as Δ in Figures 1–4.

Discussion

It can be seen from Figures 1–4, that none of the artificial life models studied generate substantially greater network complexities than do the control networks. By "substantially", I mean more than 10% of the total network complexity. The complexity difference that exists is nevertheless often statistically significant, albeit small (of the order of a few bits). By contrast, most of the 26 practical networks studied in Standish (2010a) exhibited substantially greater complexities than their controls, the exceptions being the David Copperfield adjective-noun adjacency dataset (0.98 bits), and the *C. elegans* metabolic network (which at 34.6 bits is about 0.1% of the total complexity).

The complete failure for several independent artificial evolutionary systems to be able to generate this complexity surplus weakens the case for the surplus as being due to operation of an evolutionary process. It is possible that this is another illustration of the difference between artificial evolutionary systems and natural evolutionary systems observed with Bedau-Packard statistics (Bedau et al., 1998). There is also the possibility that some systematic artifact skews the observational data towards more symmetric networks (which increases complexity values), however it seems implausible that networks collected by many different observers in many different fields should exhibit the same systematic error. More work needs to be done applying this complexity metric to both artificially evolved networks and observational data of naturally evolved networks to elucidate if this is artifact, or a real phenomenon.

Conclusion

In this work, I measured the network complexity of several artificially evolved foodwebs to see if I could reproduce the complexity surplus seen in empirical network data. In none of the artificial systems I studied was the complexity surplus substantial enough to be considered a real effect.

References

Bedau, M. A., Snyder, E., and Packard, N. H. (1998). A classification of long-term evolutionary dynamics. In Adami, C., Belew, R., Kitano, H., and Taylor, C., editors, *Artificial Life VI*, pages 228–237, Cambridge, Mass. MIT Press.

Caldarelli, G., Higgs, P. G., and McKane, A. J. (1998). Modelling coevolution in multispecies communities. *J. Theor, Biol.*, 193:345–358.

Darga, P. T., Sakallah, K. A., and Markov, I. L. (2008). Faster symmetry discovery using sparsity of symmetries. In *Proceedings of the 45st Design Automation Conference*, Anaheim, California.

Drossel, B., Higgs, P. G., and McKane, A. J. (2001). The influence of predator-prey population dynamics on the long-term evolution of food web structure. *J. Theor. Biol.*, 208:91–107.

Görnerup, O. and Crutchfield, J. P. (2008). Hierarchical self-organization in the finitary process soup. *Artificial Life*, 14:245–254.

McKay, B. D. (1981). Practical graph isomorphism. *Congressus Numerantium*, 30:45–87.

Myrvold, W. and Ruskey, F. (2001). Ranking and unranking permutations in linear time. *Information Processing Letters*, 79:281–284.

Ray, T. (1991). An approach to the synthesis of life. In Langton, C. G., Taylor, C., Farmer, J. D., and Rasmussen, S., editors, *Artificial Life II*, page 371. Addison-Wesley, Reading, Mass.

Standish, R. K. (1994). Population models with random embryologies as a paradigm for evolution. *Complexity International*, 2.

Standish, R. K. (2000). The role of innovation within economics. In Barnett, W., Chiarella, C., Keen, S., Marks, R., and Schnabl, H., editors, *Commerce, Complexity and Evolution*, volume 11 of *International Symposia in Economic Theory and Econometrics*, pages 61–79. Cambridge UP.

Standish, R. K. (2003). Open-ended artificial evolution. *International Journal of Computational Intelligence and Applications*, 3:167. arXiv:nlin.AO/0210027.

Standish, R. K. (2004a). Ecolab, Webworld and self-organisation. In Pollack et al., editors, *Artificial Life IX*, page 358, Cambridge, MA. MIT Press.

Standish, R. K. (2004b). The influence of parsimony and randomness on complexity growth in Tierra. In Bedau et al., editors, *ALife IX Workshop and Tutorial Proceedings*, pages 51–55. arXiv:nlin.AO/0604026.

Standish, R. K. (2005). Complexity of networks. In Abbass et al., editors, *Recent Advances in Artificial Life*, volume 3 of *Advances in Natural Computation*, pages 253–263, Singapore. World Scientific. arXiv:cs.IT/0508075.

Standish, R. K. (2010a). Complexity of networks (reprise). *Artificial Life*. submitted. arXiv: 0911.348.

Standish, R. K. (2010b). SuperNOVA: a novel algorithm for graph automorphism calculations. *Journal of Algorithms - Algorithms in Cognition, Informatics and Logic*. submitted, arXix: 0905.3927.

Identification of Functional Hubs through Metabolic Networks

M Beurton-Aimar[1], N. Parisey[2], F Vallée[1] and S. Colombié[3]

[1]UMR 5800 - LaBRI Laboratoire Bordelais de de Recherche en Informatique - Université de Bordeaux 1
[2]UMR BiO3P - Biologie des organismes et des populations appliquée la protection des plantes. INRA, le Rheu
[3]UMR 619 - Biologie du Fruit. INRA, Villenave d'Ornon
beurton@labri.fr

Abstract

Metabolic networks are described as a set of pathways, each pathway being a set of biochemical reactions, mainly enzymatic reactions. It is often considered that the global behavior of a metabolic network is characterized by the addition of behaviors of each pathway. But in fact, in such large networks it is difficult to predict the consequences of competition between several enzymes that react with the same molecule (metabolite) or, for example, how modification of the production of a specific molecule can influence, directly or not, another part of the network (Klamt and Stelling (2002)). Several works have shown that metabolic networks exhibit all characteristics of "small world" networks (Wagner and Fell (2001), Ravasz et al. (2002)). In this case, classical techniques from graph analysis domain can be used to find partitions or clusters in such networks. However in biological context, finding clusters must be related to biological functions and the analysis has to be driven by this concern to reveal functional links through the network. But these analyses from classical clustering use the network descriptions and do not take into account biological constraints on pathways. Tools based on linear algebra like elementary flux modes (Schuster et al. (1999) (or Extreme pathways Papin et al. (2002)) allow to select pathways through the network which satisfy constraints like the steady state of the system. In metabolism context, steady state is defined as a state where all the molecules produced by one reaction are consumed by another one, except external inputs or outputs. The obtained result is a set of unique and minimal reaction chains which are all solutions of the system. This set is often huge and gives a good appreciation of the network complexity. It is also considered as a measure of the network robustness to perturbations (Stelling et al. (2004)) and is suitable to identify if some reactions are always associated to another one even if they are not directly connected (path length between these two nodes longer than 1). We have used this tool to refine the description of 4 metabolic networks: 3 from mitochondria of different cell types (muscle, liver and yeast) and the last one from tomato fruit central metabolism. The elementary flux modes computings have identified from several thousands solutions for mitochondria networks to more than one hundred thousand for tomato fruit network. These results show the complexity level of interactions through the networks and obviously it is not possible for biologists to analyze them by hands (Pérès et al. (2006)). Building classification and identifying modular organization in the networks is an obvious requirement. We have applied clustering technique to identify reaction or molecule hubs and so to show new indirect links between distant parts of the networks. Evident hubs have been found like currency metabolites ATP, ADP ... but other belonging to the TCA cycle pathway like malate have been identified as good candidates for hub role whereas nothing in the primary network descriptions suggested that they are more implicated than another belonging to the TCA cycle. This result is consistent with analysis of the topology of E. Coli metabolism done by Zhao et al. (2007). These first results lead to build multi-layer description from metabolite hubs to small modules connections taking into account both information about feasible pathways and metabolites and reaction degree of connections.

References

Klamt, S. and Stelling, J. (2002). Combinatorial complexity of pathway anaysis in metabolic networks. *Mol Bio Rep*, 29:233–236.

Papin, J., Price, N., and Palsson, B. (2002). Extreme pathway lengths and reaction participation in genome-scale metabolic network. *Genome Re*, 12(12):1889–1900.

Pérès, S., Beurton-Aimar, M., and Mazat, J. (2006). Pathway classification of tca cycle. *IEE Proceedings Systems Biology*.

Ravasz, E., Somera, A., Mongru, D., Oltvai, Z., and Barabsi, A. (2002). Hierarchical organization of modularity in metabolic networks. *Science*, 297(5586):1551[5.

Schuster, S., Dandekar, T., and Fell, D. (1999). Detection of elementary modes in biochemical networks : A promising tool for pathway analysis and metabolic engineering. *Trend Biotechno.*, 17:53–60.

Stelling, J., Sauer, U., Doyle, F., and Doyle, J. (2004). Robustness of cellular functions. *Cell*, 118(6):675–685.

Wagner, A. and Fell, D. (2001). The small world inside large metabolic networks. *Proc. Roy. Soc. London*, Series B(68):1803–1810.

Zhao, J., Tao, L., Yu, H., Luo, J., Cao, Z., and Li, Y. (2007). Bow-tie topological features of metabolic networks and the functional significance. *Chinese Science Bulletin*, 52(8):1036–1045.

Computational Approach to the Gene Regulatory Network in the *Mus Musculus* Mouse Eye Development

Daniel Aguilar[1], Antonio Córdoba[1], M[a] Carmen Lemos[1], Fernando Casares[2], M[a] Ángeles Domínguez[2] and María J. Tavares[2]

[1]Departamento de Física de la Materia Condensada − Universidad de Sevilla
[2]Centro Andaluz de Biología del Desarrollo − CABD (CSIC-Universidad Pablo de Olavide)
daguilar@us.es

Extended Abstract

Gene regulatory networks set a second order approximation to genetics understanding, where the first order is the knowledge at the single gene activity level. With the increasing number of sequenced genomes, including human's, the time has come to investigate the interactions among myriads of genes that result into complex behaviours. The composition and unfolding of interactions among genes determine the activity of cells and, when is considered during development, the organogenesis. Hence the interest of building representative networks of gene expression and their temporal evolution, i.e. the structure as the network dynamics (Barabási (2005)), for certain development processes.

This paper shows research on the gene regulatory network that controls the early development of the mouse (*Mus musculus*) eye. The developmental stages chosen comprise the specification of the eye progenitor cells (E9: nine days post fertilization), the morphogenesis of the optic cup (E10.5) and the specification of the first neuronal precursors (E11.5). The reason for this choice of stages was two-fold: first, all subsequent stages are contingent upon these ones. And second, the complexity of cell types is reduced, so we can consider that the tissue we analyze is composed of basically one cell type. The gene network construction (see Figure 1) has been carried out from our gene transcription profiling experiments of murine eyes at the already mentioned embryonic stages and a wide bibliographic review for their interactions (see Rebay et al. (2005), Sansom et al. (2009) and Purcell et al. (2005)). The resulting network can be analysed through network theory, where genes are the network nodes and interactions are the network links (U. Alon (2006)).

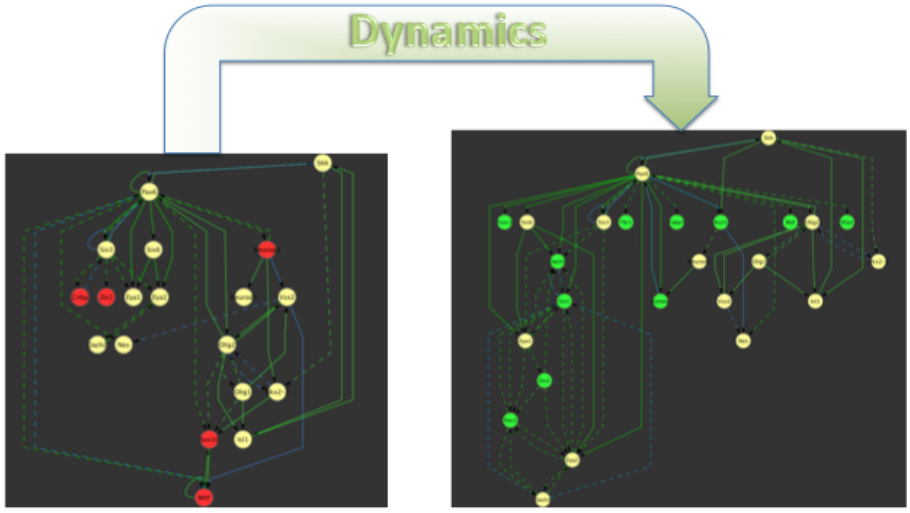


Figure 1: Visual transformation dynamics between E9 (left) to E10.5 (right) stages. Nodes: $Red = E9$, $Green = E10.5$, $Yellow = Common$; Links: $Green = Activator$, $Blue = Repressor$, $Solid = Functional\ interaction$, $Dashed = Protein - protein\ interaction$.

With the aim of determining a pathway through these links from E9 to E10.5, and then to E11.5, i.e. the process dynamics, a genetic algorithm (GA) has been developed (Mitchell (1999)). In this GA, each "chromosome" in the initial population consists of two parts. The first one involves the activity of the interactions among all nodes, i.e. activation, inhibition and non-interaction. The second part includes an activation/inhibition set of rules for the inputs into each gene. Each chromosome generates a dynamics to build a possible E10.5 stage starting from the well-known E9 stage (later E11.5 from found E10.5), where the input interactions for each node will determine its next state.

The GA fitness function is made of two suitably weighted addends: the first one, and the most important in the global computation, a distance between the experimental stage and the resulting from the GA; and the second one, a distance between the chromosome part formed by the genes interactions and the ones experimentally found.

It should be mentioned that certain experimental interactions may be lacking or be incorrect, so the interaction fitting must not be totally strict.

The results lead to a complete fitting for the gene activation states and to a good approximation for the links, and allow discovering some development dynamics. Further analysis, based on biological considerations, additional experiments and network pruning, will allow a final tuning to select the best network and dynamics for the early phases of eye development as a general model of organogenesis.

References

Albert-László Barabási (2005). Scale-Free Networks: A Decade and Beyond. *Science*, 325 (5939): 412.

Ilaria Rebay, Serena J. Silver and Tina L. Tootle (2005). New vision from Eyes absent: transcription factors as enzymes. *TRENDS in Genetics*, Vol.21 No.3: 163-171.

Stephen N. Sansom, Dean S. Griffiths, Andrea Faedo, Dirk-Jan Kleinjan, Youlin Ruan, James Smith, Veronica van Heyningen, John L. Rubenstein, Frederick J. Livesey (2009). The Level of the Transcription Factor Pax6 Is Essential for Controlling the Balance between Neural Stem Cell Self-Renewal and Neurogenesis. *PLOS Genetics*, 5(6): e1000511. doi:10.1371/journal.pgen.1000511.

Patricia Purcell, Guillermo Oliver, Graeme Mardon, Amy L. Donner, Richard L. Maasa (2005) Pax6-dependence of Six3, Eya1 and Dach1 expression during lens and nasal placode induction. *Gene Expression Patterns*, 6 (2005) 110118.

U. Alon (2006). An Introduction to Systems Biology: Design Principles of Biological Circuits. Chapman & Hall/CRC, 2006.

M. Mitchell (1999). An introduction to Genetic Algorithms. MIT Press, Cambridge, MA.

Evolving Gene Regulatory Networks for Real Time Control of Foraging Behaviours

Michał Joachimczak[1] and Borys Wróbel[1,2]
[1]Computational Biology Group, Institute of Oceanology, Polish Academy of Sciences
[2]Laboratory of Bioinformatics, Adam Mickiewicz University in Poznań, Poland
{mjoach,bwrobel}@iopan.gda.pl

Abstract

We use a genetic algorithm to obtain artificial gene regulatory networks (GRNs) controlling real time behaviour of artificial agents (animats) that gather food resources in a 2D environment. We build a system in which evolving GRNs are encoded in linear genomes. The encoding allows to determine which transcriptional factors (TFs) interact with which regulatory regions (promoters) to form a GRN. The sensory information is provided to an animat as externally driven concentration of selected TFs. Concentration of selected internally produced TFs is interpreted as signals for actuators. We first consider foraging for one food source and then scale the problem up to obtain animats that are able to switch between two types of food sources and avoid the poisonous one. We show that our system is highly evolvable, even though the genome encoding is very flexible (which results in a large search space) and though continuous product accumulation and degradation causes latencies in signal processing by the networks. We then discuss the topological properties of evolved networks and their evolutionary trajectories. Our results provide a first step toward a more ambitious goal of developing an artificial ecosystem in which multiple individuals will compete for food and mates.

Introduction

Gene regulatory networks (GRNs) are an underlying control mechanism of all living cells. Artificial gene regulatory networks are build either in order to understand how biological GRNs work or in the hope of engineering biologically-inspired systems that are, like biological systems, robust to environmental and mutational insults. Many GRN models have been proposed, and quite a few papers considered the properties of evolving GRNs (for recent examples see Kuo et al., 2006; Nicolau and Schoenauer, 2009). The model used in this work has been inspired by earlier work of Eggenberger (1997) and is similar to several models developed in recent years (e.g Andersen et al., 2009; Schramm et al., 2009). We have developed it originally for controlling development of 3-dimensional embryos with non-trivial morphologies or patterning (Joachimczak and Wróbel, 2008, 2009).

Models of multicellular development are of great interest in the field of Artificial Life, because they require considering at least two levels of biological organization: the level of molecules (genes, proteins, etc.) and the level of cells. Foraging behaviour also requires these two levels, and in this work we apply our model to control unicellular animats in an environment with a gradient of scents coming from food particles. The cells are provided with sensory information using externally driven concentrations of transcription factors. Such setup resembles chemotaxis of unicellular eukaryotic organisms which can detect gradients of substances with membrane receptors (Bagorda and Parent, 2008). However, small size of prokaryotic cells does not allow for signal to noise ratio high enough to do that, so prokaryotes evolved a different mechanism: bacterial chemotaxis is based on detecting concentration fluctuations over time and random changes in movement direction (see e.g. Alon, 2006).

What happens to the animat in our system depends not only on the GRN state but also on the laws of simulated simple Newtonian physics, and this can be exploited by evolution. The interplay between the GRN and the physical environment removes some of the computational burden from the GRN. This is analogous to the physics-GRN interplay used previously to guide developmental systems (e.g. Eggenberger, 2003; Joachimczak and Wróbel, 2009). Also, this is not the first time GRNs are used to control animat behaviour (see e.g. Bentley (2004); Taylor (2004); Quick et al. (2003) where obstacle avoidance, wall and light following were considered). Some previous papers considered the dynamical properties of GRNs in which product concentrations oscillate, decay within desired time frames or respond to noisy external signals (Kuo et al., 2004; Knabe et al., 2006).

In this paper, we will first provide a brief description of the regulatory model and the environment used in the experiments. Two experiments will be presented. In one only single food type was provided. In a more complex problem, two substances were present. One was poisonous until a certain number of particles of the other were consumed; at this point the roles were reversed. We end with a discussion of the topologies of evolved GRNs and of the evolutionary trajectories that lead to the solutions.

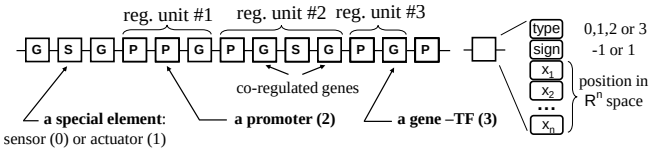


Figure 1: The genome and the structure of a single genetic element. Each element consists of a type field, a sign field, and a sequence of N real values used to determine affinity to other elements ($N = 2$ was used in this paper).

The model

Our model of the genome is designed to capture some of the most essential features of evolving regulatory networks. The GRN topology is encoded in a linear genome. Genes encode transcription factors (TFs). TFs bind to promoters of genes to regulate their expression. Any network topology can be encoded: there are no limits on the size of the network, number of connections or maximum number of connections per node. This is because our primary motivation is to build a model that allows to ask questions relevant to biology (where no such limits are imposed) rather the to solve a particular optimization problem (where enforcing them might decrease the search space).

Encoding a GRN in a linear genome

The genome is a list of genetic elements that fall into three classes: elements that code for products (called genes); regulatory elements (called promoters); special elements (that code for external inputs and outputs of the regulatory network). The genome is parsed sequentially, and regulatory units are formed whenever a series of promoter elements is followed by a series of genes. Special elements are assigned to input and output nodes at a later stage. In result, each regulatory unit is composed of one or several regulatory elements and one of several genetic elements coding for TFs. Regulatory units form the nodes in the regulatory graph. When the unit is expressed (active), all TFs that belong to it are produced at the same level. Fig. 1 provides an overview of the process, together with the structure of a single genetic element.

Each genetic element encodes N coordinates ($N = 2$ was used) and thus can be assigned to a point in R^2 space. When a TF lies close enough to a promoter in this space, a connection between the respective regulatory units is formed (a cut-off distance of 5 prevents full connectivity, Fig. 2). The abstract R^2 product-promoter space should not be confused with the 2D environment in which the animat is simulated. "Sign" fields of two elements allow to determine whether the weight of a connection is positive or negative (using multiplication). Because regulatory units can have multiple promoters and multiple genes, two nodes can be connected by several edges.

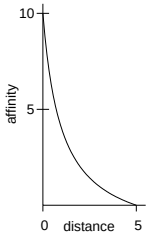


Figure 2: Translation of Euclidean distance between genetic elements into affinities (weights). Maximum weight of 10 and cut-off at the distance of 5 are used.

Genetic algorithm

Each evolutionary run was initiated with 300 genomes consisting of 5 randomly created regulatory units. Element coordinates were initiated using uniform distribution to draw a random direction and a random distance from (0,0). The population size was kept constant. Binary tournament selection (draw two individuals, keep the better one) was used.

Genetic operators in our system act on the level of genetic elements. Single element mutations can change element type, sign bit (changing all its connections from inhibitory to regulatory or vice versa) or coordinates (changing connection weights). Coordinates are changed by shifting the associated point in the abstract N-dimensional space in a random direction by a distance drawn from a Gaussian distribution. Duplications and deletions of multiple elements occur at random locations in the genome. When they occur, some points are created or removed in the abstract N-dimensional space. If $N \leq 3$, it is possible to visualize how the points move, appear and disappear. The duplication/deletion length is drawn from a geometric distribution, with equal probability of duplications and deletions. Since genetic elements cannot be created de novo and there is no recombination, all genetic elements in any individual can be traced back to the elements in one of the genomes that were present in the initial population.

GRN dynamics

During simulation of the network, regulation of a given regulatory unit (node of the graph) will result in the change in concentrations of TFs that belong to this unit. The rate of TF synthesis is a function of activation of all promoters belonging to the unit (inputs to the node). First, distances are converted to affinities using an exponential function shown in Fig. 2. Activation of each promoter is a sum of the concentration of all products binding to it weighted by their affinities. This sum (A) is used to derive product concentration (a value within $< 0, 1 >$) in the next simulation step using the equation:

$$\frac{dL}{dt} = \frac{2}{1 + e^{-(A-1)}} - L \qquad (1)$$

where the time step dt determines the simulation accuracy ($dt = 0.1$ was used), and current concentration (L) determines the intrinsic product degradation rate, so concentration increases only if the sigmoid function gives a value above L, and the degradation rate increases if the value is

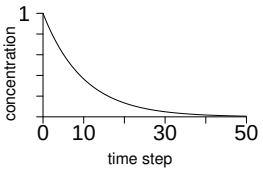


Figure 3: Time scale of exponential degradation of a transcription factor over time. All TF concentrations in the system are in the range $< 0, 1 >$.

negative. Fig. 3 illustrates the time scale of product degradation used in the system.

Animats and their environment

Animats are modelled as simple circular objects, equipped with two identical food sensors located towards the front and two actuators towards the back (Fig. 4). To evaluate the fitness of an animat, it is placed in the environment that is an open and continuous 2D space with randomly placed food particles. The animat and food particle coordinates are represented as real numbers.

Each food particle generates a field of scent. At each location in the environment, the scent coming from a food particle is directly proportional to the distance to this scent source. Fields from each food particle sum up, forming a scent map (see right panel of Fig. 6 for an example). Animat's sensors perceive the scent at its location in a non-directional manner, so the gradient information has to be extracted using two sensors and/or movement. When a food particle is consumed its field is removed from the map.

Sensors and actuators are assigned to special elements in the genome, which come in two subtypes: input and output. The scent perceived by a sensor determines the concentration of associated input product. In addition to inputs representing sensors, special product whose concentration is always at a maximum (1) can be used to initiate gene expression. The positions of input products in the R^2 product-promoter space determine how they are connected to the rest of the GRN. However, direct connection between the input products and the output is not permitted. The output element behaves essentially as a promoter in the system, but a better way of putting it is that it is a single promoter regulating expression of a single gene, and that the concentration of the corresponding product regulates the animat's actuators. The assignment of special elements to actual inputs and outputs in the system (sensors/actuators) is based on their order in the genome, superfluous special elements are ignored.

Actuators work as thrusters and animat motion is simulated using simple Newtonian physics. The thrust force is proportional to the concentration of a product associated with the output special element. The force is not directed toward the centre of the animat, so when the activation of actuators differ, the animat is caused to spin. However, the animat cannot turn on the spot: even when only one actuator is active, the animat moves in a loop rather than rotate in place. Switching the actuators off results in continued motion because of inertia, but the animat will be eventually

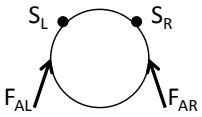


Figure 4: The placement of sensors of chemical signal (scent) and actuators on the animat.

brought to a stop due to fluid drag proportional to squared velocity. This drag also limits the maximum speed possible. To find a food particle it is thus not only necessary for the animat to properly orient itself but also to properly deal with inertia when taking turns.

Results

Designing a way to assess fitness in a chemotaxis problem

In preliminary experiments, we have assessed the fitness by measuring the energy level of an animat at the end of its lifetime divided by the maximum energy that could be obtained in a particular environment. The energy level was set to zero at the beginning of fitness evaluation. Each particle consumed by the animat increased the energy by 1.

We have noticed that if the genetic algorithm was constructed to minimize

$$f_{fitness} = 1 - \frac{energy}{energy_{max}} \tag{2}$$

the best animats would often show a suboptimal behaviour, circling towards the food (Fig. 5). The corresponding hill in the fitness landscape is very easy to find and climb, but difficult to escape from: simply circling around a map allows to find some food particles by chance and the behaviour can be further optimized by controlling the loop diameter with only a single actuator (tightening it when the scent gradient increases). To promote alternative solutions, an additional term was introduced in the fitness function. This term favours individuals that change the direction of the movement at least once during their lifetime. For such individuals $f_{fitness}$ was decreased by 10%. This helps to arrive at animats capable of controlling both actuators early during the course of evolution, even though circling behaviour remains a strong attractor for the genetic algorithm.

Using a map with fixed locations resulted in overfit individuals that simply followed trajectories optimized for a particular map. To prevent this, for each animat fitness was averaged for four maps with the same number of particles at random locations (so this average, f_{avg}, would differ slightly even for two identical genomes).

Designing sensor preprocessing for foraging behaviour

The only information about the environment made available to the animat is the state of two sensors S_L and S_R, corresponding to the concentration of the food scent in the location when the sensors would actually be at. To allow the information from the sensors to be processed by the

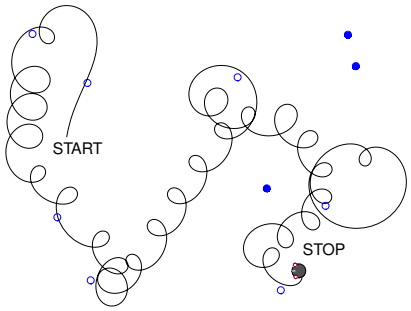


Figure 5: A common suboptimal solution in the fitness landscape: targeting the food particles by performing circular motion. Despite low average speed, it can be quite effective at targeting. Particles consumed during lifetime are drawn as empty circles.

GRN, some preprocessing of sensory information is necessary. This is because TF concentrations in the system are in the range $< 0, 1 >$ whereas the value of the scent field at a given location has no upper limit.

Our initial approach was to provide the GRN with concentrations of input products $S1$ and $S2$ that would correspond directly to the values of S_L and S_R but were restricted to $< 0, 1 >$ using sigmoid function. This, in principle, should have allowed for the emergence of simple controllers with sensors cross wired with actuators in the regulatory network, similar to Braitenberg vehicles.

However, such signal preprocessing resulted in very poor evolvability, for a very simple reason. The diameter of the animat is very small compared to the size of the environment, so both sensors perceive the scent at a very similar level. Unless the animat is very close to the food particle, the difference in signal levels would often be less than 1%. Although we were able to obtain some animats capable to climb the scent gradients, their overall performance was poor.

Much better results were obtained when a simple sigmoid function was used to derive the concentration of the input product $S1$:

$$S1 = \frac{1}{1 + e^{-\alpha(S_R - S_L)}} \tag{3}$$

where α controls the steepness of the function and was set so that it amplifies small differences between S_L and S_R. If S_L is equal to S_R, the $S1$ concentration is 0.5. The concentration approaches 1 or 0 depending on the difference between S_L and S_R.

Using just $S1$ was enough to evolve animats that quite efficiently search for one food source. However, we have observed that the animats turn too fast when close to the food sources and too slowly when far away. Information about the distance from sources is missing in $S1$, so to allow for better turn taking we have introduced a second input product ($S2$) which concentration depends on the perceived food scent at

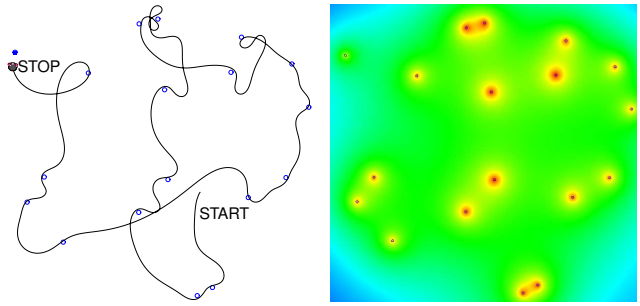


Figure 6: Left panel: best individual navigating the map with single type of food; Right panel: initial map of scent intensity that is locally perceived by animat sensors (normalized to span full colour range).

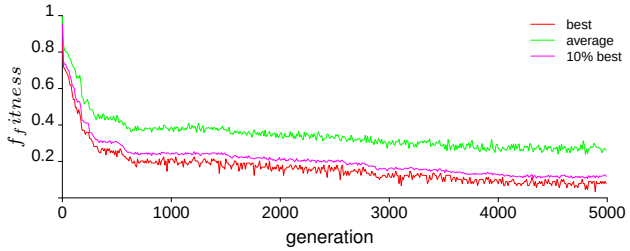


Figure 7: Fitness over generations for the problem with a single type of food source.

the animat location:

$$S2 = \frac{2}{1 + e^{-\beta(S_R + S_L)}} - 1 \tag{4}$$

where β similarly controls the steepness of the sigmoid.

Foraging for a single type of food

In the first experimental setting, maps were created by placing 20 food particles at random locations. Animat behaviour was simulated for 2000 time steps. The size of the map was such that typically about 300 time steps were required to cover the distance between the farthest food particles at maximum speed. Because about 50 steps are needed for TF degradation at the default rate (Fig. 3), latencies in information processing in the GRN quickly become an issue when there is a need to react fast.

Out of ten independent evolutionary runs of 5000 generations, seven resulted in very efficient solutions. The best animats had f_{avg} between 0.05 and 0.25, which means that around 70-90% of food particles were collected. In the remaining runs the algorithm got stuck in a solution with a circular motion and loop tightening when close to a food particle (such behaviour is shown in Fig. 5). Only about 30-40% food particles could be collected with this approach.

The behaviour of the best individual in ten runs is shown in Fig. 6 (left panel). Fairly good solutions were found quite early (Fig. 7; this could be observed also in the other runs).

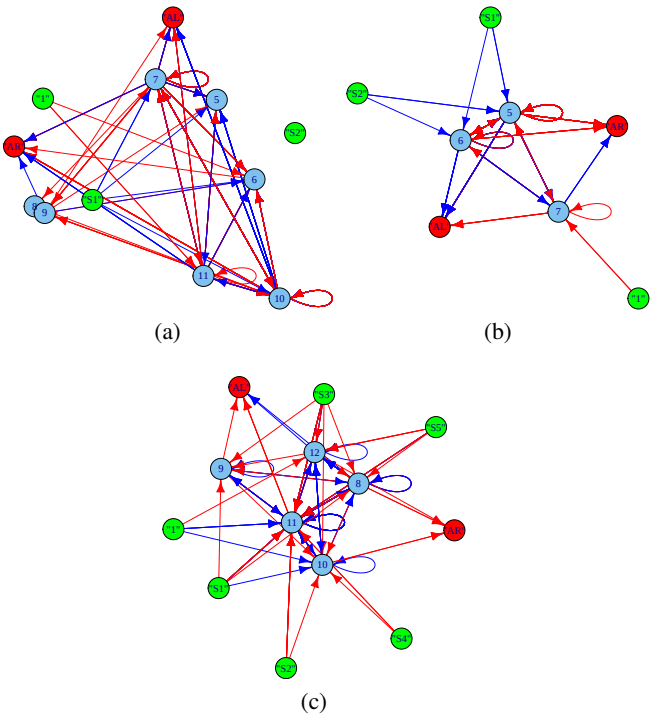


Figure 8: GRN topologies of animats foraging for one (a,b) or two (c) chemical substances; (b) shows the GRN of the best animat (generation 5000), and (a) its ancestor in generation 3000. Multiple links between nodes have been collapsed to one line.

In later generations, speed and targeting gradually improves, but even the best animats turn too widely (which later needs correction) and move only at about 60% of the maximum speed possible. However, this is an expected trade-off given the physical (inertia) and biochemical (latencies in product synthesis/degradation) constraints.

Analysis of the evolved regulatory network (Fig. 8b) shows a simple, largely symmetric topology with only three internal nodes. The best GRN uses both sensory inputs available: the directional information ($S1$) and the scent concentration at the animat location ($S2$). However, $S2$ is not critical for navigation, and in the best networks in other runs it was often disconnected. Indeed, going back from the best animat at generation 5000 to its ancestor at generation 3000 (Fig. 8a) shows that in the ancestral GRN $S2$ was not connected. Perhaps this is the primary reason why the ancestral animat is less efficient at gathering food particles.

In 2000 generations that separate these two animats, the network became less dense (see below) and the genome size roughly doubled. The number of deletions and duplications was similar, but the duplications were longer on average: 6.8 genetic elements for average duplication vs. 2.3 for deletion (despite lengths being drawn from the same distribution). It

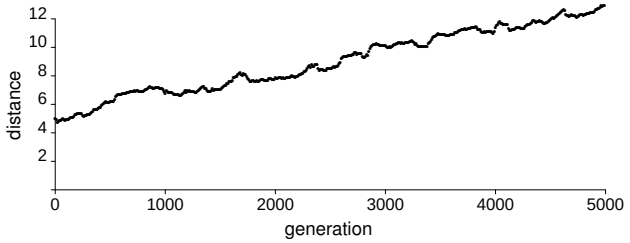


Figure 9: Measuring the spread of genetic elements over time: average distance from $(0, 0)$ for all genetic elements in each generation for the problem with single food type.

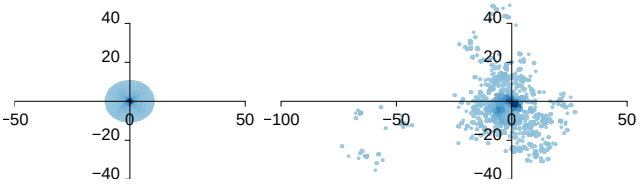


Figure 10: Distribution of genetic elements from all individuals in first generation (left) and last generation (right). Dots represent locations in R^2 space of all genetic elements in the gene pool.

is possible that this excess of duplications allows for some of the duplicated elements to take on new functions and perhaps to optimize the speed of information processing in the network. This requires changing the coordinates of points associated with the duplicated elements.

Many genetic elements in a particular genome are not important for GRN functionality and small mutations in their coordinates are neutral or almost so. This means that over time, points in product-promoter space spread away from each other, and because initial coordinates are drawn from a uniform distribution centred at 0, points spread away from the centre (Fig. 10 and Fig. 9). The unimportant points perform a random walk and slowly move beyond the interaction distance, which reduces the density of the network. This is a general property of element evolution in our system, but a similar process is at play in biological evolution: neutral mutations in duplicated genes or promoters eventually remove redundant connections in GRNs.

Foraging for two types of food

The chemotaxis problem can be made more difficult by introducing more types of food. Evolving animats that search for two types of food may be seen as a first step towards evolving even more complex behaviours, such as the ability to avoid obstacles or to search for mates, perhaps with separate modules in the network controlling different behaviours. The task was formulated so that consuming an appropriate food particle increases the energy by 1, wrong particle results in a decrease by 1. Poison changes to food and vice versa when energy reaches a certain value (5). When energy

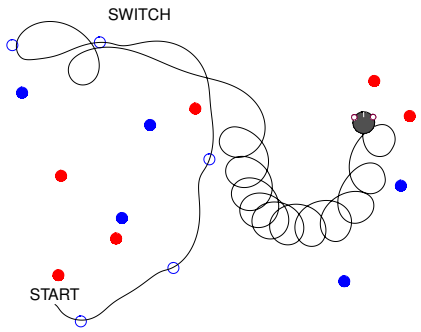


Figure 11: The path of the best individual from generation 2600 for the problem with two food sources. After seeking blue particles, the animat switches to circular motion strategy, similar to that observed in the previous experiment (Fig. 5). This behaviour is replaced later in evolution with direct targeting. Consumed particles are drawn as empty circles.

drops below zero the animat becomes immobile. 30 food particles of type one (blue) and 30 of type two (red) were placed in the environment, and this rather high density of particles was required so that poison avoidance could evolve (otherwise accidental consumption would be too rare to affect fitness).

To allow perception of two substances in the same fashion as for one, four special genetic elements were used as GRN input ($S1$ and $S2$ for the first type, and $S3$ and $S4$ for the second). To increase evolvability, one more element ($S5$) had to be introduced. The concentration of its product would be 0 until the energy reaches 5 for the first time, and 1 from then on, signalling that a behaviour switch is necessary. The best animats evolved before this mechanism was introduced would move slowly enough to collect only about 5 particles during their lifetime.

In this experimental setup ten independent evolutionary runs were performed, but with individual lifespan increased to 7000 time steps so that more particles could be collected. In three runs f_{avg} for the best individuals was between 0.19 and 0.26, which means that the animats extracted around 70% of energy available to them in the environment. The animats showed the desired behaviour: they first searched for blue particles and switched to search for red as soon as signal $S5$ was set to 1. In four runs the best animats would gather around 50% of energy by efficiently collecting blue particles, but then collected red using the circular motion approach (a manifestation of same attractor in the fitness landscape as seen on Fig. 5). The best animats in the remaining runs would gather only blue particles and then stop.

Fig. 12 shows the behaviour of the best animat in ten runs, its GRN has been presented in Fig. 8c. Information from all externally provided signals ($S1 - S5$) is used. This animat actively avoids wrong (red) food particles when searching for blue. However, after the behaviour switch, when it

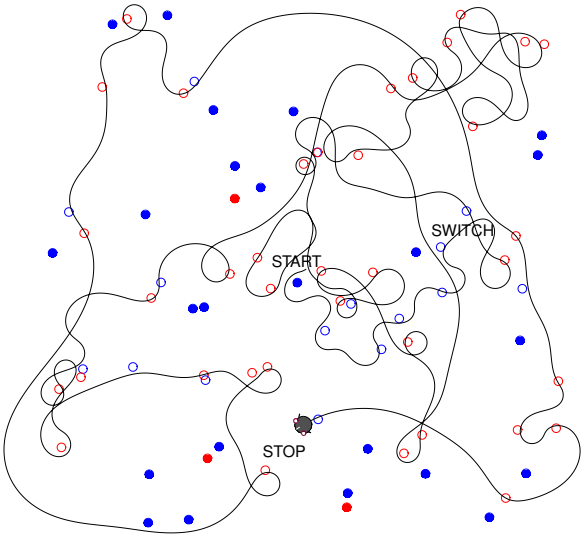


Figure 12: The path of the best individual from the final generation (5000) for the problem with two food sources. The switch in behaviour occurs after 5 blue particles are consumed. Particles consumed are marked as empty circles.[1]

actively seeks red particles, it will consume any blue particles that accidentally come its way. The difference in the avoidance behaviours likely stems from the fact that the evolutionary pressure to avoid red particles at the beginning is stronger: consuming them when low on energy will be lethal.

Fig. 13 shows that evolution of foraging for two types for food was less gradual than for one type (Fig. 7), though in some runs the plateaus were less pronounced; their lengths varied. The best individual from the first plateau (generation 2600) actively and efficiently searches for blue particles, and avoids the red, but uses the circular motion strategy after the food/poison switch (Fig. 11). This behaviour allows to gain energy because at this stage there is more red particles than blue. The best individual from generation 3100 (the second plateau) already seeks the red particles actively, but moves rather slowly. The third plateau in fitness is reached by improving the speed.

A large fitness improvement between generation 2900 and 3900 corresponds to an increase of genome size (Fig. 14). The duplications that lead to this increase tend to create new connections between existing nodes in the GRN rather than create new nodes. This is not surprising: duplication of genetic elements results more readily in a new product-promoter pair than in a new regulatory unit. However, it was rare for the duplications to occur before the onset of the episodes of fitness improvement. Rather, they tended to occur at the very end of these episodes or during the plateaus.

[1] Videos of animat behaviours are available at:
`http://www.evosys.org/alife12chemotaxis`

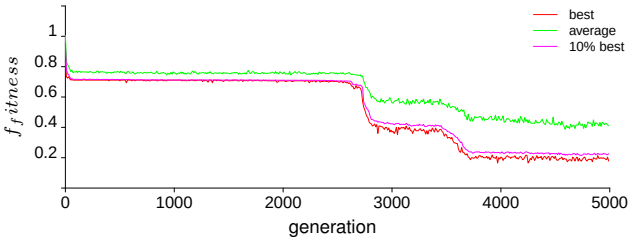


Figure 13: The fitness for the problem with two food types. Three stages corresponding to improved behaviour are seen.

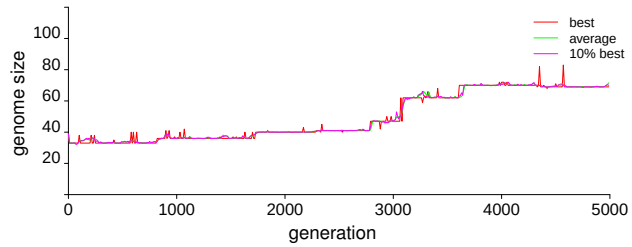


Figure 14: The genome size (the number of genetic elements) for the problem with two food types.

This suggests that even though the duplications may prepare the stage for the improvements, the episodes themselves are actually initiated when the elements acquire new functions, and the points in the promoter-product space need to move some distance before that can happen.

Discussion

The genetic algorithm used in this work did not include elitism nor recombination. Together with small population size and the fact that the fitness was evaluated using random scent maps would mean that the best genomes, subject to the Muller's ratchet, would not necessarily be maintained in the population. Even so, good solutions were obtained. Random genomes grew through duplications, with better and better fitness thanks to the divergence of duplicated elements. The evolvability was good enough to scale the system to a more complex foraging problem, in which several navigating behaviours are required. The best animat displayed 3 behaviours, activating them in a proper fashion: first seeking blue particles and avoiding red, and then seeking red particles after food/poison switch. Although preprocessing of sensory information was necessary to obtain good evolvability in the foraging tasks, all the information available to the animats came from the scent concentrations perceived at the locations of two sensors (Fig. 4).

Before this research platform could be used to address biologically relevant questions pertaining to the properties of evolving networks, a few issues need to be addressed. First of all, evolved networks are fairly small, even for the more complex problem. Secondly, to observe any emerging trend in properties with confidence, networks from multiple evolutionary histories will have to be analysed. This is because

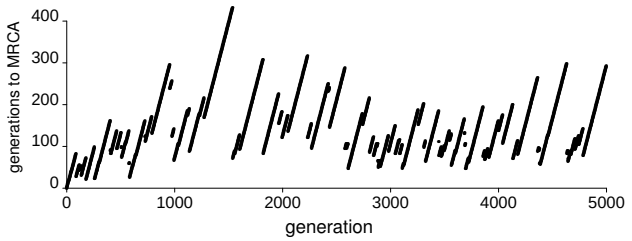


Figure 15: The number of generations to the most recent common ancestor (MRCA) for the entire population in each generation of the experiment with one food source. Average: 148.7; the value for the experiment with two sources was similar.

individuals in a single evolving population are not very divergent. For a given generation, all individuals have a common ancestor about 150 generations earlier (Fig. 15), so they represent a single successful lineage rather than multiple lineages evolving independently. To analyse general trends in properties, evolutionary runs will have to be repeated many times. Alternatively, such analysis will require constructing a system in which multiple lineages can co-exist.

Artificial GRNs have computational properties equivalent to recurrent neural networks. However, when compared with typical perceptron-based neural networks, GRNs have richer dynamics coming from product accumulation and degradation. This results in lower response time, but can allow e.g. to integrate noise or produce signals that change gradually. We provide a more in-depth discussion of evolvability of regulatory networks together with comparison to perceptron-like GRNs in a parallel paper (Joachimczak and Wróbel, 2010).

We have observed that animats in the final generation have usually low maximum TF concentrations, rarely above 0.3. This may stem from the evolutionary pressure to reduce the response time of the networks. In a system in which concentrations represent some continuous variables (such as the activity of a sensor or actuator), it is relative changes of concentrations that are important. Intrinsic TF degradation is exponential, so resulting relative changes do not depend on the concentration itself (Fig. 3). However, relative changes caused by regulation do depend on current concentration: a low concentration allows for a larger relative change, so keeping TF expression low permits to react faster to changing environmental signals. In biological systems lower concentrations would result in a decreased signal-to-noise ratio, but in our system GRNs there is no noise. The only thing that prevents using extremely low expression levels is the limit of maximum connection weight. It will be interesting to investigate if adding noise to gene expression will affect the properties of evolved networks and the way information is encoded in changing concentrations of TFs.

Our results demonstrate that a slightly simplified model previously employed for artificial embryogenesis (Joachim-

czak and Wróbel, 2009) can be used to obtain GRNs controlling real-time foraging behaviours of unicellular artificial organisms. In our future work, we plan to bring two problems together with the goal to build a system in which multicellular animats will develop from single cells and co-evolve competing for resources.

Acknowledgements

This work was supported by the Polish Ministry of Science and Education (project N519 384236). The computational resources used in this work were obtained thanks also to the support of the project N303 291234, the Tri-city Academic Computer Centre (TASK) and the Interdisciplinary Centre for Molecular and Mathematical Modelling (ICM, University of Warsaw; project G33-8).

References

Alon, U. (2006). *An Introduction to Systems Biology: Design Principles of Biological Circuits (Chapman & Hall/CRC Mathematical & Computational Biology)*. Chapman & Hall, 1 edition.

Andersen, T., Newman, R., and Otter, T. (2009). Shape homeostasis in virtual embryos. *Artif. Life*, 15(2):161–183.

Bagorda, A. and Parent, C. A. (2008). Eukaryotic chemotaxis at a glance. *J Cell Sci*, 121(16):2621–2624.

Bentley, P. J. (2004). Adaptive fractal gene regulatory networks for robot control. In *Workshop on Regeneration and Learning in Developmental Systems in the Genetic and Evolutionary Computation Conference (GECCO 2004)*.

Eggenberger, P. (1997). Evolving morphologies of simulated 3D organisms based on differential gene expression. In *Proceedings of the Fourth European Conference on Artificial Life*, pages 205–213, Cambridge, MA. MIT Press.

Eggenberger, P. (2003). Genome-physics interaction as a new concept to reduce the number of genetic parameters in artificial evolution. In *Congress on Evolutionary Computation, CEC '03*, volume 1, pages 191–198.

Joachimczak, M. and Wróbel, B. (2008). Evo-devo *in silico*: a model of a gene network regulating multicellular development in 3D space with artificial physics. In *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, pages 297–304. MIT Press, Cambridge, MA.

Joachimczak, M. and Wróbel, B. (2009). Evolution of the morphology and patterning of artificial embryos: scaling the tricolour problem to the third dimension. In *Proceedings of 10th European Conference on Artificial Life (ECAL 2009)*. Springer.

Joachimczak, M. and Wróbel, B. (2010). Processing signals with evolving artificial gene regulatory networks. In *Artificial Life XII: Proceedings of the Twelfth International Conference on the Simulation and Synthesis of Living Systems*. MIT Press, Cambridge, MA.

Knabe, J. F., Nehaniv, C. L., Schilstra, M. J., and Quick, T. (2006). Evolving biological clocks using genetic regulatory networks. In *Artificial Life X: Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, pages 15–21. MIT Press/Bradford Books.

Kuo, D., P., Banzhaf, W., and Leier, A. (2006). Network topology and the evolution of dynamics in an artificial genetic regulatory network model created by whole genome duplication and divergence. *BioSystems*, 85(3):177–200.

Kuo, D. P., Leier, A., and Banzhaf, W. (2004). Evolving dynamics in an artificial regulatory network model. In *Parallel Problem Solving from Nature - PPSN VIII*, volume 3242 of *Lecture Notes in Computer Science*, pages 571–580. Springer Berlin / Heidelberg.

Nicolau, M. and Schoenauer, M. (2009). On the evolution of scale-free topologies with a gene regulatory network model. *BioSystems*, 98(3):137–148.

Quick, T., Nehaniv, C. L., Dautenhahn, K., and Roberts, G. (2003). Evolving embodied genetic regulatory network-driven control systems. In *Advances in Artificial Life*, pages 266–277.

Schramm, L., Jin, Y., and Sendhoff, B. (2009). Emerged coupling of motor control and morphological development in evolution of multi-cellular animats. In Kampis, G. and Szathmáry, E., editors, *Proceedings of 10th European Conference on Artificial Life (ECAL 2009)*. Springer.

Taylor, T. (2004). A genetic regulatory network-inspired real-time controller for a group of underwater robots. In *Proceedings of the Eighth Conference on Intelligent Autonomous Systems (IAS-8)*, pages 403–412.

Theoretical and Computational Frameworks

Algebraic Representation and Modeling of Evolutionary Innovation and Adaptation in Biological Systems

Igor Balaz[1] and Dragutin T. Mihailovic[1]

[1]Faculty of Agriculture, University of Novi Sad, Dositej Obradovic Sq. 8,
Novi Sad, Serbia
ibalaz@polj.uns.ac.rs

Abstract

Living systems are equipped with the coding system which enables them to autonomously determine set of agents for performing all functional tasks. Since scope of their functioning for given environment is entirely dependant on internally given structure of the coding system they are able to evolve both new traits (evolutionary innovation) and optimize existent ones (evolutionary adaptation) by means of mutations and different mechanisms of genetic rearrangements. In this paper we give a generalized mathematical framework for presenting evolution in living systems in terms of category theory, comprising both innovation and adaptation. On that basis we construct a simple computational model, where as an example we performed evolution of randomly generated coding sequences and analyze appearance of interaction networks and their evolution, as well as evolution of the coding sequence itself. We also demonstrate that evolved networks have some properties of metabolism-like systems.

Introduction

Basic mechanism of evolution in biological world is well known. All organisms are equipped with some form of coding sequences (RNA or DNA) which serve as a blueprint for synthesis of RNA and/or proteins, which in turn perform all functional tasks (interaction with environment, transformation of elements, synthesis of all necessary systemic structures). Their functional role depends on their ability to assimilate a segment of environment with an appropriate set of internal operations to produce reactions. Therefore, some form of shared interface between organism and its environment should exist. At the same time, coding sequences are subject to changes through generations due to various external or internal factors, and these changes can be reflected on phenotypic traits of an organism. Usually, phenotype changes are only variations of a given trait, but sometimes organisms can attain completely new properties. These changes are reflected on the overall reproductive success of an organism, as a measure of evolutionary success, which is relative category and depends on three factors: genotype of that organism, properties of the given environment and other organisms in the same population. In more abstract terms, in a given universe, an organism occupy subset of that universe (called niche), and possible scope of organism's place is genetically determined.

Currently, full formal treatment of evolvability is not yet achieved. Evolutionary adaptation is addressed through evolutionary computation (e.g. De Jong, 2006) but innovation has been scarcely touched. One of the reasons for that may be the need for more general formal setting in order to fully capture possibilities of appearance of new structures or mechanism. Some efforts have been made, within domain of topology spaces (Stadler et al. 2001; Shpak and Wagner, 2000) where importance of introducing genotype-phenotype separation was highly emphasized. However, in those works focus was mainly on analysis of topological configuration of state space.

Our aim here is to show that generalized framework for creating an evolvable system (both innovative and adaptive) for the given universe can be described as generation of a set of free objects in n generators from the monoidal subcategory where objects of the mother category are collection of all possible words over the given alphabet which constitutes the set of generators of the universe. The process consists of subsequent creation of equivalence classes where equivalence relations are only implicitly determined, so that their exact action depends on structure of objects on which they are applied. For the sake of simplicity in our elaboration we will limit ourselves only to the domain of strings and lattices. Resulting objects are ordered pairs of strings, which can be considered as functions in the given universe, by creation of enriched category. Overall, starting monoidal subcategory can be interpreted as mutation search-space, where objects are coding strings (DNAs), their substrings are transformed to functions which operate on a given environment and give rise to appearance of the network of interactions (metabolism). Therefore, one object of the monoidal subcategory coupled with the set of all functions derived from its structure constitutes one genotype, while the phenotype is here simply equal to the metabolic network. Together, they constitute an organism.

In the next section we will develop described framework and will point out some general requirements of genotype-phenotype mapping, in order to be functionally evolvable.

After that we will concretize previous notions by generating computational model and demonstrate its functioning for randomly generated coding sequences placed in randomly generated environment. Finally, in conclusions we point out some possibilities for further development and application of the given framework.

Mathematical Framework

If we denote some living system as L and its environment as E, in analogy to biological world, we can define three basic premises important for our task:

1. interaction of two systems L and E is based on transformation of elements of E by action of elements of L, called functional elements;
2. in order to interact, functional elements and environment must share some of their properties and such shared subset of properties should be general enough to serve as a representative of the environment;
3. generation of functional elements is determined internally, by the system L, through existence of some coding element(s) on which a sequence of equivalence relations is applied.

Whatever mathematical representation we chose for L and E, in the most general sense both of them can be regarded as free objects generated over some alphabets, which are in turn members of category of sets, *Set*:

$$\mathbb{C} \underset{\mathbf{F}}{\overset{\mathbf{U}}{\rightleftarrows}} Set\,. \qquad (1)$$

where $\mathbf{F}$ is free functor, $\mathbf{U}$ is its right adjoint, forgetful functor, while $\mathbb{C}$ is category of all algebraic structures generated by $\mathbf{F}$. If we take some $X \in Set$, then $\mathbf{F}(X)$ is free object, while X can be defined as a set of generators of the $\mathbf{F}(X)$. In order to keep things as simple as possible, we will neglect all notions of dynamics of metabolism and existence of any control mechanism, which will demand definition of additional restrictions on chosen structures. Also, from the environment we expelled all other "organisms" and consider environment as an inert space without internal dynamics. Therefore, we will only define two alphabets, X_E and $X_L \subseteq X_E$, and corresponding monoidal categories generated by $\mathbf{F}$: $(M_L, \bullet, e)$ and $(M_E, \bullet, e)$ where M_L and M_E are categories of all strings generated by corresponding generators (objects are strings, mappings are inclusion maps), bifunctor $\bullet$ is binary operation of string concatenation, while e is identity element, in this case, empty string. Following our analogy with coding elements in living systems, $(M_L, \bullet, e)$ can be interpreted as universe of all possible coding strings while $(M_E, \bullet, e)$ can be regarded as universe of all possible structures in the environment. At this stage, $(M_L, \bullet, e)$ is simply a subcategory of $(M_E, \bullet, e)$ with no additional properties. However, applying premises we postulated at the beginning of this section, this simple monoidal category will be transformed into category of function over the given universe.

Since by premise 3, we demand that some structure should serve as the coding element we can choose any $d \in M_L$ and construct slice category M_L / d where objects are mappings in M_L with d as the codomain ($a \xrightarrow{\alpha} d, b \xrightarrow{\beta} d, \ldots$), while mappings are given by $f : (a \xrightarrow{\alpha} d) \to (b \xrightarrow{\beta} d)$ such that $\beta \circ f = \alpha, \alpha \circ f = \beta$. Therefore, M_L / d is category of all strings which are substrings of the string d. Description of structure of objects in categories M_L / d and $(M_L, \bullet, e)$ can be regarded only as a specialization, but these two categories are structurally different. Category M_L / d is not monoidal category since closeness under operation $\bullet$ is violated. Loosing its monoidal character, category M_L / d is also expelled from dynamics provided by bifunctor $\bullet$ which in practice means that by changing d, M_L / d should be reconstructed *de novo*. However, certain stability of M_L / d can be achieved if d is part of some equivalence class. Living systems provided such stability by existence of rewriting-like systems of gene expression where linear order of elements of chains is preserved while several points of reduction are performed (e.g. basis for DNA transcription are triplets, genetic code is degenerate and amino acids can be sorted into groups with similar chemical reactivity). In other words, process of gene expression generates additional structure on the DNA in the following manner. If we represent gene expression as the functor $\mathbf{G}$ which is full and faithful but not embedding, from category $(M_L, \bullet, e)$ to monoidal category $\mathbf{At}$ generated from some $X_A \subseteq X_E$, then category $\mathbf{At}$ will preserve internal string order but exact reconstruction of its source in $(M_L, \bullet, e)$ could not be realized. In other words, since $\mathbf{G}$ is bijective only on hom-sets, but is not injective on objects, its object function g have section such that for

$$a \underset{s}{\overset{g}{\rightleftarrows}} \mathbf{G}(a) \qquad (2)$$

equation $g \circ s = 1_{G(a)}$ is valid but $s \circ g = 1_a$ cannot hold (it does not have retraction). In that case objects of $(M_L, \bullet, e)$ are naturally, by functor $\mathbf{G}$, separated into disjoint union of n sets. If we denote set of objects of $(M_L, \bullet, e)$ as S_{M_L} then relation $R \subseteq S_{M_L} \times S_{M_L}$ naturally defined by g:

$$R = \{(d, d') \in S_{M_L} \times S_{M_L} \,|\, g(d) = g(d')\} \qquad (3)$$

is congruence relation, and by $[d]_{\mathbf{G}}$ we will denote equivalence class of elements of $(M_L, \bullet, e)$ with respect to functor $\mathbf{G}$. Therefore, in order to provide some degree of stability when facing with mutations, sufficient formal requirement is that expression mechanism from genotype to phenotype reduces number of elements along the process. However, structures created during expression should be able

to interact with environmental structures and perform some action upon them, in order to be functional. This notion leads us to our second basic premise: two systems should share some of their properties, thus creating an interface.

Formally, interface between two different objects can be introduced quite simply. It is enough to postulate segments of structures as visible to each other which is basically in focus of control theory and agent based systems. From algebraic perspective it raises an interesting problem of general mathematical properties of interfaced objects which should be fulfilled in order to be able to perform transformations, either mutual or governed by one of interacting systems. However, here we will omit that question and will take most simple approach by creating interface within the same group of mathematical structures, performed by the functor **G**.

As we implied in introduction, object part of the functor **G** decomposes objects of M_L into ordered pairs whose structure is determined by arrangement of attributes introduced by the functor **G**. It can be done in *n* steps, depending on the chosen model. If we follow analogy with the natural world, then we can construct 2-step process. The first one is defined simply by identifying all permutations of strings of fixed length created over the alphabet X_L, grouping them into m disjoint subsets, and declaring equivalence relation θ over each subset. Then, subsets are equivalence classes and strings $s \in S_{M_L}$ can be mapped into corresponding quotient strings s/θ.

Since by functor **G** each element of X_E is equipped with some attributes it raises following structure. Let us denote set of all words over the alphabet X_E as *T*, set of all attributes as *M*, and set of attribute values as *J*, then formal context is (T,M,J,I) where I is a ternary relation $I \subseteq T \times M \times J$ which unites objects with corresponding attributes. Further, if $O \subseteq T$ is set of all functional elements, $W \subseteq M$ is set of their attributes and $K \subseteq J$ is set of attribute values for W, such that $O' = \{w \in W \,|\, (\forall t \in O), tIw\}$ and $W' = \{t \in T \,|\, (\forall m \in W) tIw\}$, then concept of the context (T,M,J,I) is triple (O,W,K) where $O' = W$ and $W' = O$. Since for a given context a number of different concepts can be defined, we will denote set of all possible concepts as $\mathbf{B}(T,M,J,I)$. If we take $\subseteq$ as an order relation, then $(\mathbf{B}(T,M,J,I);\leq)$ is concept lattice where nodes are concepts of the given context. Finally, we will demand that set of attributes M is created as union of languages created over *n* alphabets. Elements of alphabets we will denote as generator attributes, and all other words will be called derived attributes. Reasons for that will be clear shortly.

Since elements of s/θ are also equipped with some attributes, they are characterized by specific $I_s \,|\, o \in s/\theta$ and are associated with the mapping $\tau : s/\theta \to (W,\leq)$, $((o_1,o_2,...,o_n) \to (\{w_1 \times j_1\}_1,\{w_2 \times j_2\}_2,...,\{w_n \times j_n\}_n))$. Clearly $(W,\leq)$ is not a chain anymore since there are no defined order relations among members of the set M by the mapping τ which preserves only order generated at the original coding string. However, following our analogy with natural systems, we can define some relations among attributes themselves. Keeping things as simple as possible we can for example take only one alphabet $D \subset M$ and define equivalence relation ϑ over all words of the *D*-language. Resulting posets $s/(\vartheta \circ \theta)$ now represents "folded" functional elements, where order of remaining attributes determine interface. Referring back to our second basic premise, we demand that in order to interact, functional elements should reduce environment on the basis of existence of shared properties. Here, it means that interface is formed on the basis of existence of some $V_p \times L \subseteq W \times K$ where $V_p \subseteq W, L \subseteq K$, and V_p is actually a subset of remaining generator attributes on the $s/(\vartheta \circ \theta)$. Our questions are: (i) what are the meaningful constraints for determining V_p within our framework, and (ii) what is the position of the concept generated by the $s/(\vartheta \circ \theta)$ within the $(\mathbf{B}(T,M,J,I);\leq)$. Since V_p is naturally designed to be a filter for representing environment we can postulate that it should be a part of majority of concepts of the $\mathbf{B}(T,M,J,I)$. Choosing some obscure attributes will promptly lead the system to evolutionary or functional dead end. Further, concept generated by the $s/(\vartheta \circ \theta)$ represents interface for that functional element and its upper bound is exclusively composed of concepts with derived attributes and represents place where environmental objects suitable for functional transformation can be found.

Finally, referring back to our first premise, $s/(\vartheta \circ \theta)$ should also govern determination of some function over the "visible" part of environment. Since determination of exact function is highly dependant on the chosen model, at this stage is only possible to point out general requirements which should be fulfilled in order to autonomously generate functions within given framework. Our strategy is to reconstruct possible relations from already generated structures, and then to group them into small number of isomorphic representatives, according to the structure of V_p. We will start with some basic notions.

Any finitary relation R can be defined as a couple $R = (D(R), C(R))$ where $D(R)$ is collection of nonempty sets $X_1,...,X_k$ which are called domains, while $C(R) \subseteq X_1 \times ... \times X_k$ can be denoted as the figure of *R*. Since, number of possible relations which can be constructed from the set A, equals $2^{|A|^2}$ (Robinson, 2003), our aim is to postulate some restriction rules and to find some route to grouping them together. Reconstructed relation should satisfy following axioms:

1. Function: for each element of $D(R)$, is assigned a unique element of $C(R)$;
2. Identity: for every object *a*, there exists relation $\mathrm{id}_a : a \to a$ such that for every relation $f : x \to y$, $\mathrm{id}_y \circ f = f = f \circ \mathrm{id}_x$;

3. Associativity: if f, g, h are relations, then $h \circ (g \circ f) = (h \circ g) \circ f$ should always hold;
4. Limit: if we have some set of relations of shape J, then a diagram of shape J is functor $F : J \to C$. A cone of the diagram is an object K of C together with family of morphisms $\varepsilon_x : K \to F(x)$ such that for every morphism $f : x \to y$, $F(f) \circ \varepsilon_x = \varepsilon_y$. A limit of the diagram is a cone (K, ε) such that for any other cone (L, ϕ) there exists a unique morphism $u : L \to K$ such that $\varepsilon_x \circ u = \phi_x$ for all x in J. Valid relations are only those that have limit.

By the first three axioms, we narrowed down universe of possible relations to structure preserving morphisms. In that sense set A of elements which constitute upper bound of the $s/(\vartheta \circ \theta)$ can be defined as domain of some function $f_{s/(\vartheta \circ \theta)}$ while its codomain should be subset of all concepts consisting V_p. By the last axiom we demand that reconstructed relation at least satisfy condition of being some of universal constructions in abstract algebra (product/coproduct, pushout/pullback...). As it is obvious we put only very elementary constraints, just in order to keep the system consistent. However, at the same time we come very near to our goal. Now, all possible functions can be generalized to some of few universal constructions applicable to chosen model. For example, if objects are strings, two most basic structure preserving operations are string separation and string concatenation. Limits of these universal operations are product and coproduct. Therefore, suitable codomains for the set A can only be such that operations of separation and concatenation are reconstructed. Exact structure of functions, of course depends on structure of attributes on a given functional element. As a final step, we should preserve stability of determined modes of action. It can be easily done by choosing any subalphabet of V_p attributes and mapping groups of words to some of possible universal constructions.

In summary, functor **G** maps strings of the category $(M_L, \bullet, e)$ to the monoidal category $\mathbf{At} = (A, \bullet, e)$ where objects are ordered pairs, composed of upper bound of the concept generated by the $s/(\vartheta \circ \theta)$ as the first member of the element, while the second member is determined again by equivalence relations generated by the V_p over the set *M*, morphisms are inclusion maps, bifunctor $\bullet$ is binary operation of object concatenation, while e is identity element, in this case, empty string. Due to its monoidal character, and structure of its objects, **At** can readily be used as generator of abstract metrics over the environment, represented by some category Γ, by replacing hom-sets from Γ with objects from **At**. Then Γ is category enriched over **At** (or **At**-category) such that for each pair of object $x, y \in ob(\Gamma)$, where $ob(\Gamma)$ is collection of objects of Γ, hom-set $\hom(x, y) \in \mathbf{At}$, with preserved identity, composition and associativity.

Computational Model

In order to demonstrate functioning of the framework described above we built the model of it using the individual-based approach: population of "cells" consists of individual coding strings glued with corresponding network of transformations of environmental elements, the environment is composed of *n* number of different strings and interaction of each cell with environment is computed individually. Additionally, process of transformation of codes to functions is inspired by natural process of gene expression and formally is composed of two approaches: reduction by imposing equivalence relations at different levels, on which are applied some elementary notions of relation theory. As a result, functions are created in recipe-like manner. It enables free application of mutations over the coding sequence, without designed constraints on allowed number of functional elements, or scope of their domains/codomains.

Elements of the alphabet O	**Corresponding triplets**
A	GCT, GCC, GCA, GCG
R	CGT, CGC, CGA, CGG, AGA, AGG, AAA, AAG
H	AAT, AAC, CAT, CAC, CAA, CAG
D	GAT, GAC, GAA, GAG, TCT, TCC, TCA, TCG, AGT, AGC, ACU, ACC, ACA, ACG
S	UGT, TGC
W	GGT, GGC, GGA, GGG, CCT, CCC, CCA, CCG, TGG
I	ATT, ATC, ATA, TTA, TTG, CTT, CTC, CTA, CTG, ATG, TTT, TTC
Y	TAT, TAC
V	GTT, GTC, GTA, GTG

Table 1: Rules of transformation of triplets from coding strings to elements of the alphabet O

Coding strings were generated randomly as words over the given alphabet $X = \{A, T, G, C\}$. There were no additional structures on coding strings; they were composed only as segments of symbols. Any additional structure on them can only be implicitly imposed, as a result of mappings applied to them. Separation into genes, and their expression into functional elements was designed as a composition of three mappings: identification of "proper" substrings (genes), their translation into strings of symbols equipped with attributes and folding into functions guided by order of attributes.

In analogy to the natural world, as a unit of reading of coding string we choose triplets (how changing complexity and strategy of reading influence dynamics of evolution will be presented elsewhere). Again, in analogy to the natural world, we determine rules of transformations of triplets into elements of the alphabet *O = {A,R,H,D,C,I,W,Y,V}* which is reduced version of list of amino acids. We analyzed their chemical properties and grouped similar ones into only one

representative. In order to keep their natural ratio, we assemble their coding triplets under representative groups (Table 1). As genes we identify those substrings which start with ATG tripled and end with TAG, TGA or TAA triplet. In order to optimize procedure, we neglected all sequences translated into strings shorter that 10 characters. At the same time, for members of the alphabet *O* we define formal context by introducing the set of many-valued attributes $M = \{C, H, I, K, \#\}$, and the set of attribute values $J = \{+, -, 0, 1, 2, 3, k+, k-\}$. Together they constitutes many valued context (O, M, J, I) where I is ternary relation $I \subseteq O \times M \times J$ which unites objects with corresponding attributes in accordance to the Table 2.

Translated strings to the alphabet *O* are "folded" in accordance to the attribute *H* such that all elements where H value is 1, constitute equivalence class. On that basis, reduced quotient string is formed and it will be regarded as an active place. After that, procedure for determination of domains of active place and mode of action is activated.

Having in mind that each function can be represented as subset of the set of ordered pairs, we determined functioning of folded strings creating following duple. First element, which determine domain of the function, is defined as a set of strings such that for all strings exists substring which is equal to the structure of the C-index in the single domain of the active place. Determination of the second element is based on the structure of the K-index at the domain. We simply determine mode of action as k- when there is prevalence of k- values and vice versa. Exact action is defined as string separation at the beginning of the matching region for the k-, and concatenation of all strings recognized by all domains at the single functional element, for k+. It is clear that action of k+ can only be performed if there are more than one domains. Therefore, second element of the duple is defined by performing K-derived action.

Environment is composed of *n* number of randomly generated binary strings composed only of C-attributes. Since our focus is possibility for evolution in principle, we did not determine any constraints regarding spatial distribution or concentration of substances, or their internal structure. When examining population of cells we suppose they share the same environment. It means that after each interaction, newly generated strings are placed into shared environment uniformly available to all cells. Interaction of cells with environment is performed by searching given environment for members of the domain of each functional element. If some environmental element is recognized, it is transformed into product(s) according to the K-derived action for given functional element. As a result, interaction networks are created. In order to analyze them, we used Python-based package, NetworkX (Hagberg, et al. 2008). As main indicators of evolution of networks we used number of connected components, and diameter of components. After completing interaction with environment, fitness value was calculated for each individual cell using formula: $fit = nd^{ac} + en$ where *nd* denotes total number of reactions which can be performed in given environment, *ac* is number of autocatalytic chains, while $en = ob * 100 / uk$ where *ob* is number of different molecules transformed by the cell in given environment and *uk* is total number of molecules in given environment. According to calculated fitness values, some cells were removed, while some were duplicated, keeping the number of cells in population constant. Detailed procedure depends of particular experiment performed. The rules for the evolution were chosen in order to represent essential mechanisms of natural evolution: selection, chance and ability to cope with the environment. Since in this model cells do not reproduce autonomously, which should be used as a measure of their fitness, we designed fitness so to reflect cell's ability to survive, in terms of number of reactions and percent of environmental objects which cell can recognize and transform.

Elements of *O*	Attributes				
	#	C	H	I	K
R	0	+	0	3	k+
H	0	+	0	2	k-
D	0	-	0	3	k-
Y	0	-	0	3	k+
W	0	0	0	0	0
A	0	0	1	1	0
V	0	0	1	2	0
I	0	0	1	3	0
S	1	0	0	0	0

Table 2: Attributes of elements of the alphabet *O*. C-index and I-index are taken in accordance to Whitford [2005] so that C-index represents unified charge and polarity values, while I-index is derived from Van der Waals index and normalized for a real number scale in a range [0-3]. H and K values are taken from Copeland [2000] so that H-index represents hydrophobicity distribution, while K values are normalized pKa values transformed to the mode of reactivity, where k- denotes string separation, while k+ means string concatenation. Index # is introduced as a separator of active place into domains.

Finally, next generation was created by mutating existent coding sequences of each individual. Mutation rate was set to 1 mutation cycle / 100 coding bases. Mutation cycle consists of three defined possibilities, randomly chosen at each cycle repetition: (1) point mutation – one base is randomly chosen and replaced by some other random base; (2) deletion – randomly chosen sequence up to 10 elements is removed; and (3) insertion – randomly generated sequence of the same length is inserted at randomly chosen place within coding sequence.

Performed Experiments and Simulation Results

We performed two kinds of experiments. In the first one we randomly generate population of 20 coding strings of variable length, which was randomly chosen from the interval 500-1000 elements. Environment was composed of 100 randomly generated different elements, where maximum length of generated strings was set to 10. All cells were placed into the same environment, so products generated by one cell were available to all other members of the population. After each generation fitness was calculated for each cell and the

population was accordingly separated into two halves: "least fit" and "most fit". Half of the cells from the first group were randomly chosen, eliminated and replaced by the same number of cells, randomly chosen from the second group.

In the second experiment our aim was to monitor evolution of a single cell lineage. Therefore, we randomly generated only one coding string of length randomly chosen from the interval 500-1000 elements. Environment was created as in the first experiment. In order to reduce complete randomness of the search space we performed "forced" evolution by creating generations according to the following procedure. After completing interaction with the environment, the string was multiplied into 10 copies and each of them was mutated. Interaction of mutants with environment was "virtual" in a sense that their interaction was performed separately of each other. Fitness for them was calculated like in the previous experiment and only one was randomly chosen from the "most fit" group to replace the original one.

Figure 1a, depicts growing of length of coding strings during the evolution, which is followed by increase in number of encoded functional elements at the same rate (results not shown). This is clearly governed by demands of the fitness value, where any increase in number of performed reaction is favorable. However, analysis of appeared interaction networks shows that underlying process of optimization also takes place. We searched each network for number of components where component is defined as its maximal connected subgraph where any two vertices are connected to each other by path. When number of components is 1, it means that the whole network is connected. Otherwise, network is divided into n disjointed subgraphs. For each component we also determined diameter defined as greatest distance between any pair of vertices. As it can be seen from Figure 1b, average number of components per cell in the population decreases with time and at n = 378, population became uniform in the sense that networks for all cells became fully connected. However, it takes additional 235 generations until population settled down to that value and remains uniform for next 200 generations. At the same time, diameter of the largest component slowly declined and for the last 200 generations oscillates around 9. Diameter is important characteristic which can indicate structural difference between non-biological scale-free networks, as opposed to metabolic networks (Jeong et al. 2000). For non-biological networks diameter increases logarithmically with the addition of new nodes (Barabasi and Albert, 1999) which, in this case would imply that increase in number of functional elements should lead to larger diameter of the corresponding interaction network. However, as Figure 2a depicts, diameter remains stable despite several-fold increase in number of nodes. Indirect confirmation can be seen in Figure 2b which show that after initial rapid increase in number of environmental elements suitable to transformation their number remains relatively stable, while total number elements is stabilized within first 30 generations. All of these facts indicate increased connectedness of substrates, leading to evolutionary stabilization of appeared interaction networks. Additionally, it clearly shows that networks evolved within described framework diverge from randomly generated ones.

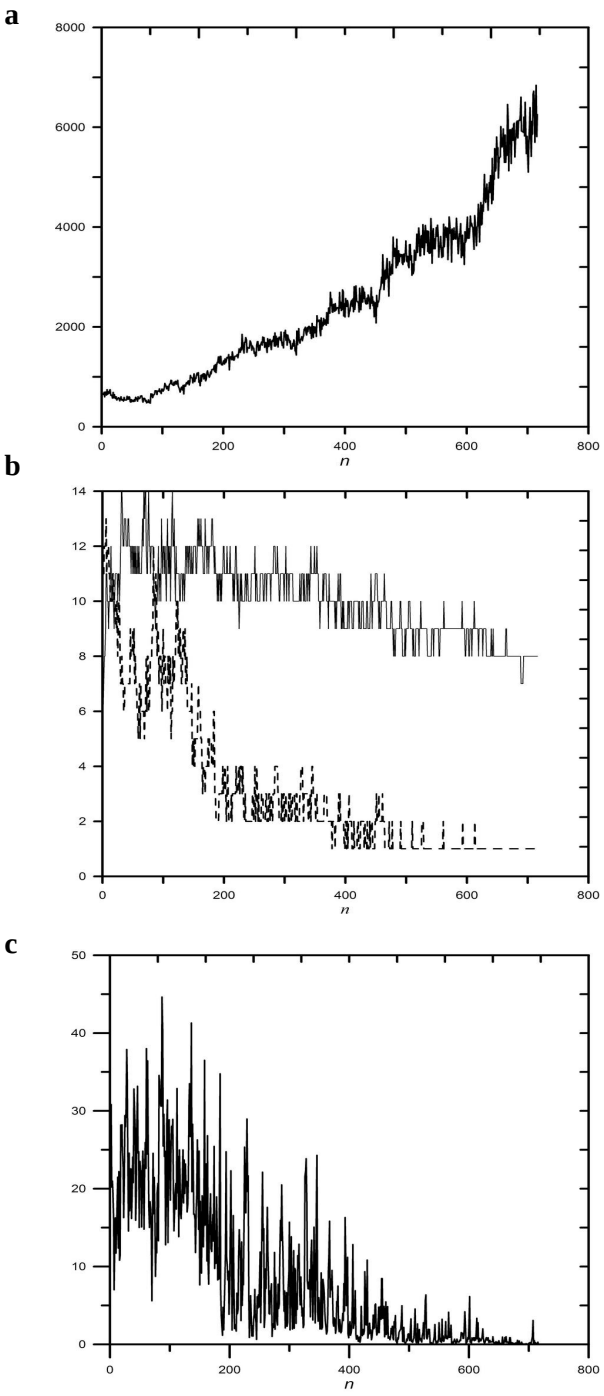


Figure 1. Results of the population experiment, where n is number of generations. Vertical axes represents: (a) average length of coding strings in population; (b) average number of connected components in interaction networks in population (dashed line) and average diameter of the largest component (solid line); (c) variance of number of connected components among cells in the population.

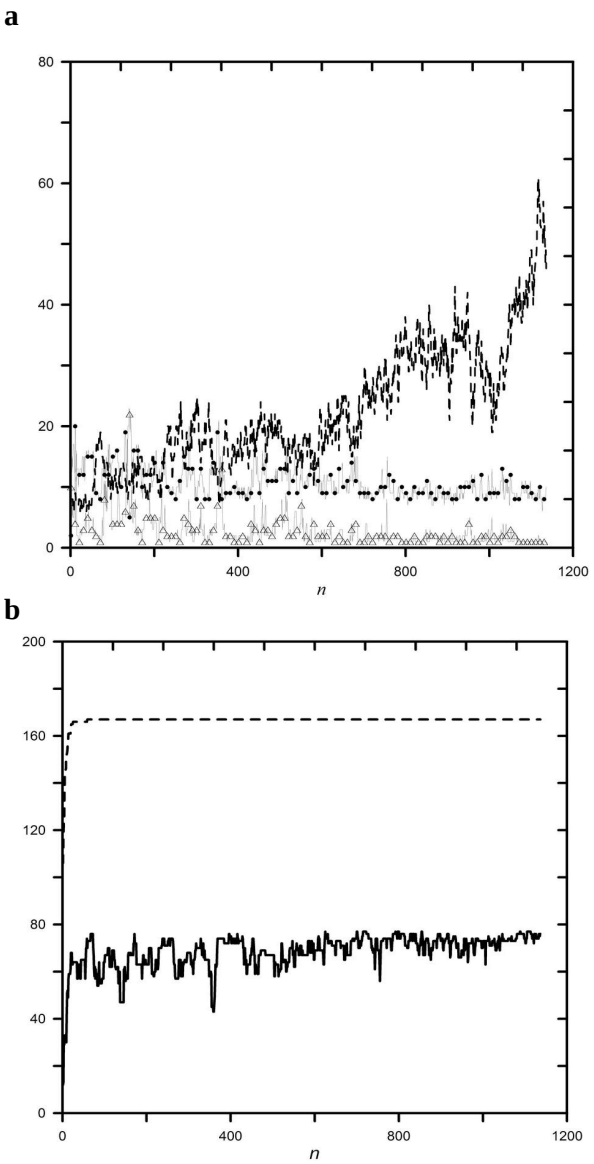


Figure 2. Results of the experiment where single cell lineage was followed through evolution; *n* is number of generations. (a) number of functional elements (dashed line), diameter of the network (circles), and number of components in the network (triangles); (b) total number of elements in the environment (dashed line) compared with number of environmental elements suitable to transformation by functional elements of the cell (solid line).

Another problem we investigated is the ability of the computational model to avoid being trapped in a fixed stable state. As it was pointed out by Conrad (1998), evolving system that gradually optimizes its traits can escape local stable state by increasing dimensionality of evolutionary search space. He termed such strategy as extradimensional bypass. In natural systems, the only mechanism to transform evolutionary search space is adding new observables by constructions of new sensors (Pattee, 1985). Each new sensor means opening possibility for functional existence of new observable in the environment which in turn means creation of additional state variable. Changing the set of state variables that characterize the system, at the same time means changing the structure of both: its functional space and its evolutionary search space. In order to examine the possibility of extradimensional bypass of our computational model, we first evolved one population of cells within one environment, and after 500 generations we replaced environment with the new one composed of 100 randomly generated different elements. Figure 3, depicts change of the fitness value along generations. A gap at n = 500 and relatively fast recovery indicate the ability of the model to extend its dimensionality when faced with new conditions. Therefore, settling into the stable state indicated by results shown in Figure 1, is directed by environmental fixedness that leads to evolutionary stagnation.

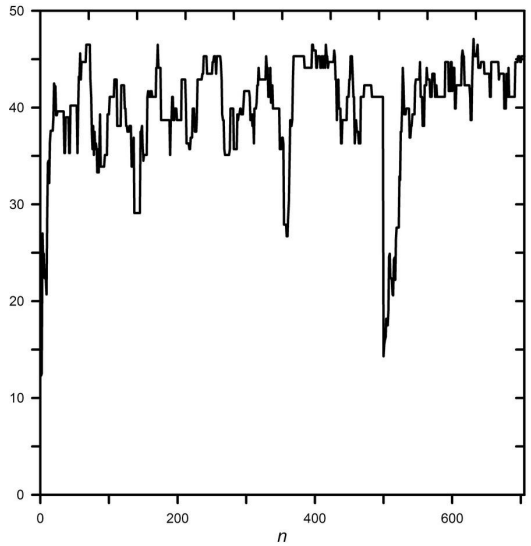


Figure 3. Results of the experiment where population of cells was successively evolved in two different environments; *n* is number of generations while vertical line shows fitness value. New environment is introduced after 500 generations.

Conclusions

We have created generalized mathematical framework for describing evolutionary systems in which appearance of phenotype (network generated by the interaction of the set of functional elements with the environment) is governed by expression of the coding sequence (genotype). Rules of expression were determined implicitly as successive determination and application of equivalence classes. In terms of universal algebra, described framework is actually process of freely creating algebraic structures. Therefore, depending on chosen rules for formation of equivalence classes, any mathematical object can be created. Comparison of such created objects can be performed by associating appropriate homomorphisms, which adds additional strength to the described framework. In the context of this paper it means that patterns of evolution can further be abstracted and analyzed

for different algebras, thus possibly revealing underlying mechanisms of evolutionary adaptation and innovation.

On the other hand, it is also a rich source of investigation of evolution within the single model. In this paper we confine ourselves only to pursuing analogy to existent natural world. However, dynamics of evolution can be easily investigated with different parameters: different modes of reading coding sequence, variations in chosen attributes and their characteristics, or cardinality of alphabets used.

Even within single model we described in this paper, some significant results are obtained. First, we show that initially created disjointed networks, during evolution tends to fall into single connected network which remains relatively stable under unchanged mutation pressure. Additionally, as opposed to random networks, diameter of evolved networks remains stable even when number of functional elements increase several-fold. Therefore, we think that they can be regarded as metabolism-like.

Finally, in order to enrich described system and raise it to the level of artificial cells, two additional aspects should be introduced into models derived from the framework. Firstly, in natural systems gene manipulation machinery is the product of that machinery and its evolution. Therefore, it would be necessary to allow interaction of obtained functions with the coding string in order to allow evolution of expression rules. Although the framework allows such reverse operations, we tried to keep presented models as simple as possible. Therefore, we didn't define any attributes over coding strings, thus keeping them out of the domain of derived functions. Secondly, notions of space (either metric or topological), quantity and control would add possibility for investigation of metabolism functioning, not just of its general structure, which is the case here.

Acknowledgements

This work was funded by the Serbian Ministry of Science and Technology under the project "Modeling and numerical simulations of complex physical systems", No. OI141035 for 2006-2010. We are also grateful to anonymous reviewers whose useful comments helped us to improve this manuscript.

References

Barabasi, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science* 286:509–512.

Conrad, M. (1998). Towards high evolvability dynamics. In Van de Vijver, G., Salthe, S., and Delpos, M., editors, *Evolutionary Systems*, pages 33-43. Kluwer Academic Publishers, Dordrecht, Holland

Copeland, R.A. (2000). *Enzymes-A Practical Introduction to Structure, Mechanism and Data Analysis*. Wiley-VCH, New York.

De Jong, K.A. (2006). *Evolutionary Computation: a unified approach*. MIT Press, Cambridge MA

Hagberg, A.A., Schult, D.A., and Swart, P.J. (2008). Exploring network structure, dynamics, and function using NetworkX. In Varoquaux, G., Vaught, T., and Millman, J., editors, *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pages 11–15. Pasadena, CA USA.

Jeong, H., Tombor, B., Albert, R., Oltvai, Z.N., and Barabasi, A.-L. (2000). The large-scale organization of metabolic networks. *Nature* 407:651–654.

Pattee, H.H. (1985). Universal principles of measurement and language function in evolving systems. In Casti, J. L., Karlqvist, A., editors, *Complexity, Language, and Life: Mathematical Approaches*, pages 268-281, Springer, Berlin

Robinson, D.J.S. (2003). *An Introduction to Abstract Algebra*. Walter de Gruyter, Berlin

Shpak, M., and Wagner, G.P. (2000). Asymmetry of Configuration Space Induced by Unequal Crossover: Implications for a Mathematical Theory of Evolutionary Innovation. *Artificial Life* 6:25–43.

Stadler, B.M.R., Stadler, P.F., Wagner, G.P., and Fontana, W. (2001). The Topology of the Possible: Formal Spaces Underlying patterns of Evolutionary Change. *J. theor. Biol.* 213:241–274.

Whitford, D. (2005). *Proteins – Structure and function*. Willey & Sons, West Sussex.

A Theoretical Study on Molecular Discreteness

Taichi Haruna[1,2]

[1]Department of Earth and Planetary Sciences, Graduate School of Science, Kobe University
1-1 Rokkodaicho, Nada, Kobe, 657-8501, Japan
[2]PRESTO, JST, 4-1-8 Honcho Kawaguchi, Saitama, Japan
tharuna@penguin.kobe-u.ac.jp

Extended Abstract

The molecular discreteness would be important in intracellular chemical reactions since the number of copies of molecules included in the reactions is small. In order to investigate the molecular discreteness systematically and theoretically, we proposed a scheme to bridge the chemical master equation (CME) and the chemical Fokker-Planck equation (CFPE) previously (Haruna, in press). CME is a discrete stochastic model and CFPE is a continuous stochastic model for chemically reacting systems. By making use of the well-known idea of approximating diffusion processes by birth-death processes (Gardiner, 2004), we constructed a family of master equations $\{\mathrm{M}_\epsilon\}_{0<\epsilon\leq 1}$ where the parameter ϵ can be considered as the degree of discreteness. This family of master equations $\{\mathrm{M}_\epsilon\}_{0<\epsilon\leq 1}$ bridges CME and CFPE in the following way: for $\epsilon = 1$ we recover CME and M_ϵ converges to CFPE as $\epsilon \to 0$. The basic idea of the construction of $\{\mathrm{M}_\epsilon\}_{0<\epsilon\leq 1}$ is as follows: in CFPE the time derivative of the probability distribution for the number of copies of molecules is the sum of the drift term and the diffusion term. Consequently we divide each reaction probability into two parts, one corresponding to the drift term and the other corresponding to the diffusion term, and introduce the parameter ϵ so that the first and the second jump moments for the number of copies of molecules (corresponding to the drift term and the diffusion term, respectively) are independent of ϵ. Our strategy here to investigate the molecular discreteness is not to study CME directly but to distinguish the properties of CME by putting CME into the family of master equations $\{\mathrm{M}_\epsilon\}_{0<\epsilon\leq 1}$ bridging CME and CFPE. In this presentation, we theoretically re-examine a transition phenomenon caused by the molecular discreteness in a simple set of autocatalytic reactions found by Togashi and Kaneko (2001) in terms of our scheme to bridge CME and CFPE. Togashi and Kaneko (2001) studied their autocatalytic reaction network consisting of four molecular species by computer simulation. Ohkubo et al. (2008) proposed a simplified version of the autocatalytic reaction network consisting of just two molecular species in which essentially the same transition phenomenon as that of Togashi and Kaneko (2001) occurs in order to explain the transition phenomenon analytically. Based on their simplified model, they showed that the transition phenomenon can also occur in the continuous stochastic model, i.e. in the Fokker-Planck equation formalism. However, they only considered the steady probability distribution. Our contribution to this problem is as follows: by combining generating function method and the large deviation theory for stationary time series, we succeeded to calculate stationary moments and correlation time for the autocatalytic network by Ohkubo et al. (2008) as functions of the degree of discreteness ϵ rigorously. We found that both stationary variance and correlation time decrease as $\epsilon \to 0$ due to an "imbalance effect" between the drift and the diffusion parts in the state in which the number of copies of one of the two molecular species is zero.

References

Gardiner, C. W. (2004). *Handbook of Stochastic Methods, Third Edition*. Springer, Berlin.

Haruna, T. (in press). Investigating the gap between discrete and continuous models of chemically reacting systems. To appear in the *Journal of Computer Chemistry, Japan*.

Ohkubo, J., Shnerb, N., Kessler, D. A. (2008). Transition phenomena induced by internal noise and quasi-absorbing state. *Journal of the Physical Society of Japan*, 77:044002.

Togashi, Y., Kaneko, K. (2001). Transitions induced by the discreteness of molecules in a small autocatalytic system. *Physical Review Letters*, 86:2459–2462.

Barrier Trees for Continuous Fitness Landscapes

Jacob Midtgaard-Olesen[1], Carsten Baldauf[2], Daniel Merkle[1]

[1]Deptartment of Mathematics and Computer Science
University of Southern Denmark, Denmark
{qilanto, daniel}@imada.sdu.dk

[2]Theory Department
Fritz Haber Institute of the Max Planck Society
Berlin, Germany
caba@paradocks.org

Extended Abstract

Local minima of a fitness landscape are separated by barriers. A barrier tree (Flamm et al., 2002) is a representation of a fitness landscape as a binary tree, where each leaf represents a local minimum; the barriers connecting the local minima are represented as the internal horizontal nodes of the barrier tree. To reflect the fitness values of barriers and minima, each node in the barrier tree is positioned relative to the height of the represented point in the fitness landscape.

Until now, barrier trees have been applied to discrete fitness landscapes. This contribution extends the concept to multi-dimensional continuous landscapes; a generalization that allows the use of the approach in various areas of life sciences. Methods for generating barrier trees for continuous fitness landscapes will be presented, ranging from a coarse grained view of the landscapes by converting them to discrete ones, to the use of heuristic approaches, where local minima are found via the Nelder-Mead simplex method, and the minima are then connected via biased random walks. Advantages and disadvantages of the approaches will be demonstrated and methods to compare generated trees will be explained.

In order to exemplify the power of the approach, the real-life problem of molecular docking will be treated. In molecular

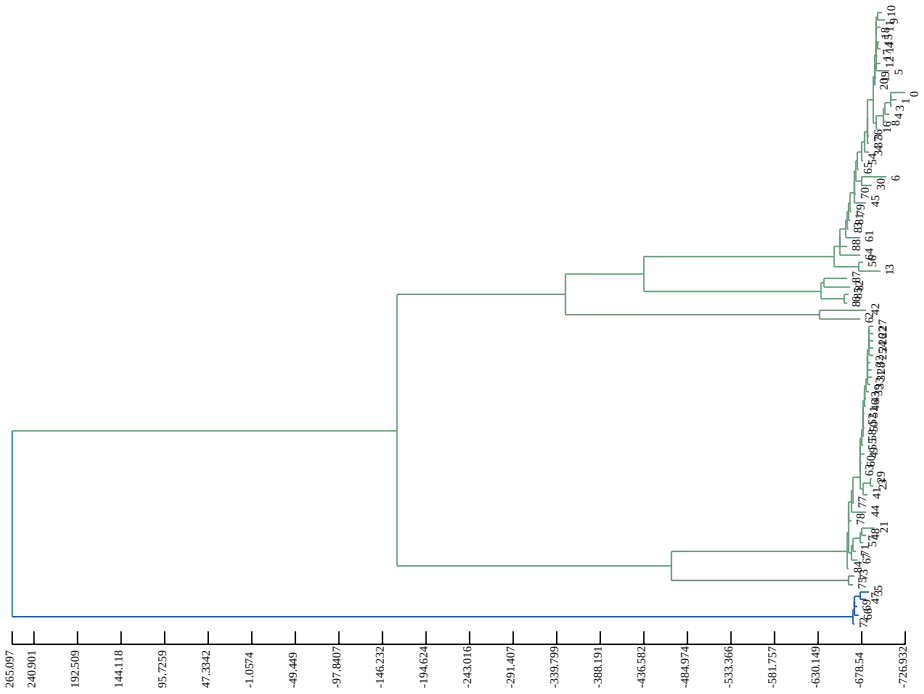


Figure 1: Barrier tree for docking of Buxaminol-E with AChE. Colored by Cartesian coordinates of the center of the ligand.

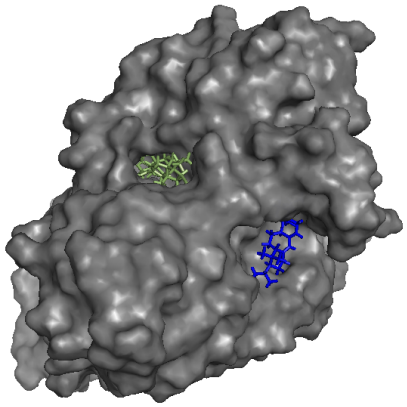


Figure 2: An illustration of the best nodes of the barrier tree from each subtree of the largest barrier. Id 35 (blue) from the lower subtree and id 0 (green) from the upper. The gray molecular surface represents the receptor AChE; the two ligands are illustrated in a stick view with the color of their respective subtree in Figure 1.

docking the interactions between small (ligand) and large (receptor) molecules are investigated in the search for the correct binding pose, which means in which ligand and receptor form a stable complex. Modeling the interaction between molecules is a complicated problem; the system's degrees of freedom include the position in Cartesian space, the orientation of the ligand, and internal flexibility of the ligand or of ligand and receptor. The ruggedness of the landscapes resulting from the different possible fitness functions makes sampling and optimization challenging. As the backbone for doing molecular docking landscape analysis, we make use of the fitness function from molecular docking software PARADOCKS (Meier et al., 2010). We used a test set with pharmaceutical relevance, a small library of known ligands and decoys of acetylcholinesterase.

Docking test illustrated in this abstract were done with acetylcholinesterase (AChE, Kryger et al. 1999) and Buxaminol-E, a natural occurring steroid isolated from Boxwood that is a known inhibitor of AChE (Thomson Scientific, 2001). First 10,000 local minima were located with the Nelder Mead method (Nelder and Mead, 1965), then removing any of those having a neighbor with a lower fitness value, neighbor again meaning within a certain step size range, and finally keeping only the 150 lowest points remaining. The barrier tree created is shown in Figure 1. The structure of the tree indicates that there are two groups of local minima separated by a high barrier, where the one group is again subdivided into smaller groups by smaller barriers.

With the barrier trees it can be seen how the search space of docking the ligand Buxaminol-E to the receptor AChE is structured. This is confirmed when we look at a figurative of the actual structure of the molecules. Figure 2 illustrates the difference in position for the ligands of the left and right subtree of the highest barrier of the barrier tree. The ligands are positioned in two distinct regions of space, which indicates two possible binding sites at the receptor.

References

C. Flamm, I.L. Hofacker P.F. Stadler, and M.T. Wolfinger. Barrier trees of degenerate landscapes. *Zeitschrift für Physikalische Chemie*, 216(2):1–19, **2002**.

Gitay Kryger, Israel Silman, and Joel L. Sussman. Structure of acetylcholinesterase complexed with e2020 (aricept): implications for the design of new anti-alzheimer drugs. *Structure*, 7:297 – 307, **1999**.

R. Meier, M. Pippel, F. Brandt, W. Sippl, and C. Baldauf. PARADOCKS - A framework for molecular docking with population-based metaheuristics. *J. Chem. Inf. Model.* 50:879 – 889, **2010**.

J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, **1965**.

World Drug Index Thomson Scientific: Philadelphia, PA, **2001**.

Binomics: Where Metagenomics meets the Binary World

Inman Harvey[1,2], Nicholas Tomko[1,2]

[1]CCNR, University of Sussex, Brighton, U.K.
[2]Evolutionary and Adaptive Systems Group, University of Sussex
inmanh@gmail.com, nick.tomko@gmail.com

Abstract

Artificial Life and Evolutionary Computation studies have until now failed to model the symbiotic evaluation methods and the extensive amounts of horizontal gene transfer that are starting to be recognized in recent Metagenomic approaches to understanding microbial populations. Examples can be seen, in Learning Classifier Systems, and the SANE algorithm, of symbiotic evaluations; the Microbial Genetic Algorithm (GA) introduced horizontal gene transfer. Here for the first time these two are brought together in the Binomic GA, which is shown to perform well in a series of trials. It is proposed that Binomics, defined as computational algorithms inspired by Metagenomic studies, forms a potentially fruitful field of study waiting to be investigated.

Introduction

For many years our conventional understanding of Darwinian evolution has been dominated by the idea of species of individuals, where those individuals favoured by selection become parents and pass on their genes to their offspring. Although selection takes place at the individual level, this vertical transmission of genetic material leads to an identifiable entity at the species level that has the capacity to adapt over time. Our models of artificial evolution, such as Genetic Algorithms (GAs), have typically followed this picture.

But in the last decade or so some biologists have started to realize that a significant part of evolution on this planet – in particular bacterial evolution – has important mismatches with this picture. There can be a significant amount of horizontal gene transmission between different individuals. As a result, much of their functionality can be passed on from their neighbours rather than inherited from parents. This makes the concept of a species in such circumstances rather looser than previously thought. Further, the fitness of a population of diverse bacteria floating in the sea may depend significantly on their local collective symbiotic functionality, rather than simply on the individual fitness of each.

Studies of the collective genetic properties of such a diverse population have come to be known as Metagenomics. Research into these natural processes has been driven by recent major advances in gene sequencing techniques. Analysis of Metagenomic results now needs new tools from complex systems theory, and already some people have started applying ideas from Artificial Life (AL) and Evolutionary Computation (EC). What has been conspicuously missing so far has been a movement of ideas in the other direction. This paper is primarily a position paper calling for new developments in AL and EC, as applied to synthetic problems, to be inspired by these new discoveries in the natural world. Drawing on biologists' use of the '-omics' suffix to refer to the collective properties of a totality, we propose *Binomics* as a new sub-field where ideas from Metagenomics are applied to applications in the binary computational world.

We start with a brief review of Metagenomics, and then a survey of those main techniques within AL that do already distil some relevant ideas. We focus on symbiotic evaluation, where individuals are evaluated collectively; specifically we look at Learning Classifier Systems (LCS) and the SANE algorithm for artificial neuro-evolution. Then we consider horizontal gene transfer, looking at the Microbial GA. We note that to date nobody seems to have combined symbiotic evaluations with horizontal gene transfer.

So we do just this with a proposal for a *Binomic Genetic Algorithm*. Although this is primarily a position paper, we can demonstrate its performance in a series of trials and compare with other evolutionary techniques. These are preliminary studies, but gratifyingly we can report that in these trials the Binomic GA outperformed the competitors by at least an order of magnitude. We suggest that this is a fruitful new area for further study, and discuss the types of applications where the particular properties of a Binomic GA could be beneficial.

Metagenomics

As a very recent field, most of the reporting on Metagenomics comes in specialised technical research papers. Useful overviews for a more general audience include Handelsman (2004), a report by the Committee on Metagenomics (2007), and Eisen (2007).

Previously our understanding of microbes has been based on studying rather few samples. In order to perform reproducible scientific experiments, well-defined species have been used, often with great care taken to culture them in the lab in isolation to ensure their purity. It is typically assumed that the test-tube is full of a single species that is genetically well-defined. It has been belatedly realized that such assumptions may not hold true in the real world.

In microbial communities there may often be large functional differences between close relatives; further, horizontal gene transmission means that many functions

(chemical cycles) typically performed by one species may be also performed by very different species. Microbes such as bacteria do not undergo sexual reproduction, but reproduce by binary fission. But they have a further method for exchanging genetic material, *bacterial conjugation*. Chunks of DNA, plasmids, can be transferred from one bacterium to the next when they are in direct contact with each other. Whereas the genomes of different humans vary by around 0.1%, different members of what may conventionally be termed a microbial species (or phylotype) can differ by up to 30%. It now makes conceptual sense – and technical developments make it possible – to perform shotgun sequencing of a whole bucketful of microbes taken from the Sargasso Sea (Venter et al., 2004) and consider the metagenomic sequence of the whole community, together with the functions that such a community collectively performs. Shotgun analysis involves breaking up the DNA randomly into small segments that are individually sequenced; then using computational methods, by seeking overlaps in these fragments, they are built up again into a complete sequence.

There are 10 times as many microbial cells in a human body than there are human cells; the human metagenome contains perhaps a hundred times more genes than the human genome (Qin et al. 2010). Many such bacteria are essential for our human well-being, and in turn they rely on us to provide them with an appropriate environment.

Comparisons: Metagenomics and AL, EC

Horizontal gene transfer rarely features in EC, though we give one example below with the Microbial GA. We can analyse the real world of bacteria floating in the sea in terms of two separate fitness criteria: internal (individual) and external (symbiotic). Firstly, each individual organism (given a sustaining environment) has to have the appropriately functioning internal mechanisms to *individually* survive. Secondly and collectively, their interactions -- the inputs and outputs of all such organisms -- must have an appropriate fit with their neighbours, so that they can *collectively* survive. In artificial evolution, we can choose to take the internal fitness criteria for granted and focus our attention solely on the external criteria, of fit to the environment. If we want to follow the Metagenomic metaphor, we shall be evolving individual entities whose value (as assessed by a fitness function) will depend on how they cooperate to tackle some task. Penn and Harvey (2004) demonstrated how ecosystem-level evolution can take place *without* genetic change in the component species, but here we want to focus on ecosystem-level evolution driven *by* genetic change.

We now discuss two areas of EC where relevant work has been done in the next sections on LCS and SANE.

Learning Classifier Systems

Learning Classifier Systems (LCS) were devised by John Holland (Holland 1976, Holland and Reitman, 1978) as a means of using a GA to do just this; for an introduction see Bull (2004). The *classifiers* are condition-action rules, typically expressed as a string of symbols, where the first part represents a template that expresses the conditions under which this classifier could match a possible input string; and the second part represents the output string of the classifier when the condition is met. Inputs to a classifier may come from the external task (e.g. they could come from sensors if this is a robot control task, or from a visual array if the task is pattern classifying), or come from other classifiers; outputs from a classifier could be to the external solution (e.g. strings interpreted as robot motor actions) or to other classifiers. Internal message-boards can be used for communication between the classifiers.

As Bull (2004) comments:

> It is important to note that the role of the GA in LCS is to create a cooperative set of rules which together solve the task. That is, unlike a traditional optimisation scenario, the search is not for a single fittest rule but a number of different types of rule which together give appropriate behaviour. The rule-base of an LCS has been described as an evolving ecology of rules - "each individual rule evolves in the context of the external environment and the other rules in the classifier system." [Forrest & Miller, 1991].

This raises a major issue in deciding how to assign a fitness to each rule, when this can only be evaluated in the context of a collective ecology. Two main approaches have been developed for LCS, named for the places where they were first proposed.

Pittsburgh LCS. In this approach each individual in the evolving population is a complete set of rules or classifiers. The rules play a role more similar to that of genes in an organism than being themselves independent organisms. In this way the problem of assigning value to each rule is avoided. The GA reproduces, with recombination and mutation, from the fitter rule sets.

Michigan LCS. In this approach the individuals in the population are the individual rules or classifiers themselves. During evolution, any of the individual rules can be operational, and this needs some arbitration mechanism to decide between them if some are matching in their input conditions but potentially conflicting in their outputs. Further complications arise from deciding how to allocate fitness to each rule that is actually operational, bearing in mind that only the collective can be evaluated. In some cases there may be a temporal element, in that the consequences of one specific condition-action rule may not be immediately apparent, but only become evident due to later knock-on consequences.

Many different methods have been proposed for tackling these issues, including auctions with specificity-based arbitration mechanisms to allow default hierarchies to form, and bucket-brigade algorithms for the temporal credit-assignment problem. This has resulted in many different flavours of Michigan LCS.

Implicit Niching in LCS

In a typical evolutionary algorithm such as a GA, we can expect selection to drive the population in the direction of genetic convergence, where it consists almost entirely of

copies, or near-copies, of the single fittest individual. But in the context of an LCS, where fitness will likely depend on the co-existence of several different individuals performing sub-functions of the whole task, such loss of diversity is undesirable. There is a need to find and maintain a diverse and cooperative set of classifiers. Some form of *niching* in the population is desirable. One approach to achieving this is through an island model, where distributed populations are separated into different demes.

Another approach is through *fitness sharing* (Goldberg and Richardson 1987), which requires some distance metric or similarity measure (either genotypic or phenotypic) between any two individuals. By using suitable methods to adjust the fitnesses of any individual according to how many other similar individuals there are nearby in this metric space, there is a tendency for the population to spread out over multiple peaks or niches in the fitness landscape; thus diversity is maintained. It can be shown that LCS models where fitness is shared amongst cooperating individuals can produce *implicit niching* (Horn et al. 1994), and this will be discussed further with the Binomic GA.

Comparisons: LCS and Metagenomics

We can relate the condition-action classifiers to the bacteria in the sea. The evaluation of the symbiotic functionality of groups of these does indeed reflect, in the context of artificial evolution, some aspects of what we observe in real world Metagenomics. The Michigan style of LCS does, at the expense of often complex auction and bucket-brigade schemes, manage the evaluation of individual 'organisms' (classifiers) that can only function effectively as part of a set. The evolutionary aspect is limited to the vertical genetic transfer between generations that is traditional with GAs.

Symbiotic Evaluations: SANE

There is a different perspective on evaluating different individuals on the basis of their group performance, taken by Moriarty and Miikkulainen (1996, 1999) in their proposal of the SANE algorithm. SANE stands for Symbiotic, Adaptive Neuro-Evolution, and this is one approach to evolving Artificial Neural Networks (ANNs). The motivation is described thus (Moriarty and Miikkulainen, 1999):

> SANE incorporates the idea of diversity into neuro-evolution. SANE evolves a population of neurons, where the fitness of each neuron is determined by how well it cooperates with other neurons in the population. To evolve a network capable of performing a task, the neurons must optimize different aspects of the network and form a mutualistic symbiotic relationship. Neurons will evolve into several *specializations* that search different areas of the solution space.

In an example implementation, they show a simple ANN with 2 layers of connection weights, from Input to Hidden neurons and from Hidden neurons to Outputs. They treat each Hidden neuron, together with its incoming and outgoing connections, as a member of the evolving population. Figure 1 shows how a complete network could be formed from e.g. 3 such Hidden neurons selected at random from the population. The network as a whole is evaluated on some required task, and the network's score is added to the fitness of each Hidden neuron that it contains. Thereafter, the selection, replication, crossover and mutation of members of the population is carried out by conventional GA methods.

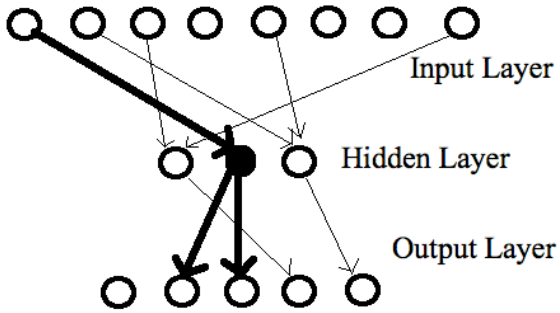


Figure 1. Each Hidden Layer neuron, with its associated incoming and outgoing connections (e.g. the highlighted central one with its links), is a member of the population. Here 3 such neurons combine to make a complete feedforward ANN.

Moriarty and Miikkulainen (1999) report that this implementation of SANE works well on such simple ANNs. They also comment that it is feasible to extend this approach to different neuron encodings, and to diverse network architectures including recurrency.

Comparisons between SANE and Metagenomics

Much as we did with the condition-action classifiers of LCS, we can relate the Hidden neurons (with incoming and outgoing connections) to the bacteria in the sea. Once again, these are only evaluated in the context of a group, which is why it has been called symbiotic (artificial) evolution. Implicit niching is again important. We can characterize this approach in much the same way as LCS, in that there are similarities in this symbiotic evaluation to some aspects of what we observe in real world Metagenomics; the evolutionary aspect is still restricted to the vertical gene transfer of conventional GAs.

Horizontal Gene Transfer: Microbial GA

Significant features of evolution that were under-recognised before Metagenomic studies included the symbiotic nature of functionality of groups of organisms, and the prevalence of horizontal gene transmission. In Genetic Algorithms, vertical genetic transmission has been very much the norm. An exception has been the Microbial GA (Harvey 2001, 2010 In Press) that we review here in a reprise of relevant sections of Harvey (2010). This is the result of stripping away as much as possible from a traditional GA, whilst maintaining the bare essentials of a population with Heredity, Variation and Selection. The Microbial GA uses Tournament Selection within a Steady State GA, hence we introduce these concepts first.

Steady State GAs

Traditionally GAs were first presented in generational form. This roughly corresponds to some natural species that has just one breeding season, say once a year, and after breeding the parents die out without a second chance. There are many natural species that do not have such constraints, with birth and death events happening asynchronously across the population. Hence the Steady State GA, which in its simplest form has as its basic event the replacement of just one individual from P by a single new one. One reason for using Steady State in a minimalist GA is that it allows for a very simple implementation of selection.

Tournament Selection

There are many problems with the traditional GA method of fitness-proportionate selection that are avoided by using some form of rank-based selection. In this, once all the members of the population have been evaluated, each fitness is rescaled on the basis of their relative ranking. A common choice made is to allocate (at least in principle) 2.0 reproductive units to the fittest, 1.0 units to the median, and 0.0 units to the least fit member, similarly scaling pro rata for intermediate rankings; this is linear rank selection. The probability of being a parent is now proportional to these rank-derived numbers, rather than to the original fitness scores.

It is possible to achieve equivalent results to this through tournament selection. If two members of the population are chosen at random, their fitnesses compared (the 'tournament'), and the Winner selected, then the probability of the Winner being any specific member of the population exactly matches the reproductive units allocated under linear rank selection.

Who to Breed, Who to Die?

Selection can be implemented in two very different ways; either is fine, as long as the end result is to bias the choice of those who contribute to future generations in favour of the fitter ones. The usual method in GAs is to focus the selection on who is to become a parent, whilst making an unbiased, unselective choice of who is to die. In the standard Generational GA, every member of the preceding generation is eliminated without any favouritism, so as to make way for the fresh generation reproduced from selected parents. In a Steady State GA, once a single new individual has been bred from selected parents, some other individual has to be removed so as to maintain a constant population size; this individual is often chosen at random, again unbiased.

Some people, however, will implement a method of biasing the choice of who is removed towards the less fit. It should be appreciated that this is a *second* form of selective pressure, that will compound with the selective pressure for fit parents and potentially make the combined selective pressure stronger than is wise. In fact, one can generate the same degree of selective pressure by biasing the culling choice towards the less fit (whilst selecting parents at random) as one gets by the conventional method of biasing the parental choice towards the more fit (whilst culling at random).

This leads to an unconventional, but effective, method of implementing Tournament Selection. For each birth/death cycle, generate one new offspring with random parentage; with a standard sexual GA, this means picking both parents at random, but it can similarly work with an asexual GA through picking a single parent at random. A single individual must be culled to be replaced by the new individual; by picking two at random, and culling the *Loser*, or least fit of the two, we have the requisite selection pressure.

Going further, we can consider a yet more unconventional method, that combines the random undirected parent-picking with the directed selection of who is to be culled. Pick two individuals at random to be parents, and generate a new offspring from them; then use the same two individuals for the tournament to select who is culled -- in other words the weaker parent is replaced by the offspring.

It turns out that this is easy to implement, and is effective. This is the underlying intuition behind the Microbial GA.

Microbial Sex: Horizontal Gene Transmission

We can reinterpret the Tournament described above, so as to somewhat resemble bacterial conjugation. If the two individuals picked at random to be parents are called A and B, whilst the offspring is called C, then we have described what happens as C replacing the weaker one of the parents, say B; B disappears and is replaced by C. If C is the product of sexual recombination between A and B, however, then ~50% of C's genetic material (give or take the odd mutation) is from A, ~50% from B. So what has happened is indistinguishable from B remaining in the population, but with ~50% of its original genetic material replaced by material copied and passed over from A. We can consider this as a rather excessive case of horizontal gene transfer from A (the fitter) to B (the weaker).

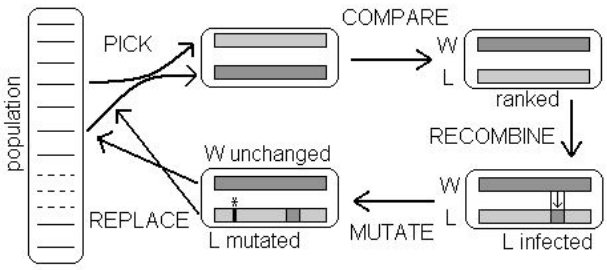


Figure 2. Sketch of the Microbial GA. The genotypes are represented as a pool of strings. One cycle of the GA is represented by the operations PICK (at random), COMPARE (their fitnesses to determine Winner = W, Loser = L, RECOMBINE (some proportion of Winner's genetic material 'infects' the Loser) and MUTATE (the revised version of Loser.

The Microbial GA in schematic form

We now have the basis for a radical, minimalist revision of the normal form of a GA, although functionally, in terms of Heredity, Variation and Selection, it is performing just the same job as the standard version. This is illustrated in Figure 2. Here the recombination is described in terms of 'infecting' the Loser with genetic material from the Winner, and we can note that this rate of infection can take different values. In bacterial conjugation it will typically be rather a low percentage that is replaced or supplemented; if instead we want to reproduce the typical effects of sexual reproduction, as indicated in the previous section, this rate should be ~50%. But in principle we may want, for different effects, to choose any value between 0% and 100%.

From a programming perspective, this cycle is very easy to implement efficiently. For each such tournament cycle, the Winner genotype can remain unchanged within the genotype-array, and the Loser genotype can be modified (by 'infection' and mutation) *in situ*. We can note that this cycle gives a version of 'elitism' for free: since the current fittest member of the population will win any tournament that it participates in, it will thus remain unchanged in the population -- until eventually overtaken by some new individual even fitter. Further, it allows us to implement an effective version of geographical clustering for a trivial amount of extra code.

Microbial GA: with a Trivial Geography

For some purposes we may not want a panmictic population, and instead constrain the operations of choosing tournament participants, and hence exchange of genetic material, to be within some local geographical distribution, perhaps within demes. This allows for more genetic diversity to be maintained across sub-populations. Spector and Klein (2005) note that a one-dimensional geography, as in Figure 3 where the population is considered to be on a (virtual) ring, can be as effective as higher dimensional versions. If we consider our array that contains the genotypes to be wrap-around, then we can implement this version by, for each tournament cycle: choose the first member A of the tournament at random from the whole population; then select the next member B at random from a deme, or sub-population that starts immediately after A in the array-order. The deme size D, <= P, is a parameter deciding just how local each tournament is.

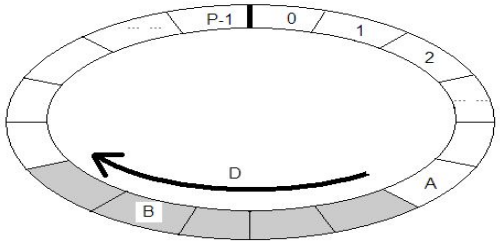


Figure 3. The population is geographically distributed on a ring, numbered from 0 to P-1. For a tournament, A is picked at random from the whole population; then B is picked at random from the deme (here of size D=5) immediately following A.

Comparisons: Microbial GA and Metagenomics

The Microbial GA is a deliberately minimalist version of a classical GA, but re-described in terms of horizontal gene transmission. The parameter that determines what proportion of genetic material is copied from Winner to Loser after each tournament can be varied according to need. Setting this at 50% gives the closest analogy to a classical GA, but other values may be of interest. Low 'rates of infection' may reflect typical values of gene transfer seen in real world Metagenomic studies; setting the rate to 100% would correspond to replication by fission of the Winner, since the Loser then becomes an identical copy. The addition of geographical demes could be tailored to correspond to any model of local interactions between, for example, bacteria swimming in the sea.

So this is a rare example of a GA with horizontal gene transmission. If we want to replicate in an evolutionary algorithm more of the essential properties that we see in Metagenomic studies of bacteria in a sea, then what is still missing is the aspect of assessing the fitness of each member of the population in some symbiotic or communal fashion.

Binomic GA

We now introduce a Binomic GA, that combines the symbiotic evaluation methodology of SANE with the horizontal gene transfer of the Microbial GA. We start with an outline of the general requirements, and then illustrate in the context of evolving Artificial Neural Networks.

General Requirements

We shall be evolving the equivalent of a Sargasso Sea (*Sea*) of individual organisms (*Orgs*). *Orgs* are not evaluated in isolation, but only as part of a randomly chosen subset of the *Sea*, a *Bucket*; such a *Bucket* may be drawn from a local area (or *Deme*) or from the whole of the *Sea* (Figure 4). The fitness function is used to evaluate a *Bucket* as a whole, and this fitness is passed on equally to all members of that *Bucket*. It is used to update the current fitness of each such *Org*, on the basis of New_Org_fit = R*Bucket_fit + (1.0-R)*Old_Org_fit. With an appropriate choice of R (0.0<R<1.0), the effective

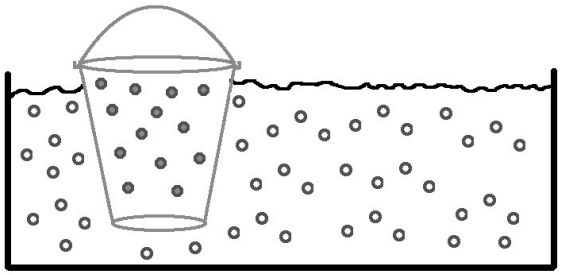


Figure 4. A Bucketful of Orgs is evaluated as a whole, and the resulting fitness assigned equally to all in that Bucket. Such a Bucket can be drawn locally from an area of the Sea, or (with a 'well-mixed' Sea) drawn at random.

fitness of each *Org* has any variance smoothed over several recent evaluations of different *Buckets* that it happens to have featured in, whilst still tracking any general changes in its environment.

Figure 5 sketches the Binomic GA. As with a Microbial GA, genetic changes in the *Sea* of *Orgs* are driven via a Tournament involving selecting two *Orgs* at random, comparing their currently stored fitnesses (as calculated via *Buckets*), and designating one as *W* = Winner, the other as *L* = Loser. *W* will remain unchanged, and *L* is the single *Org* that gets changed as a result of this tournament.

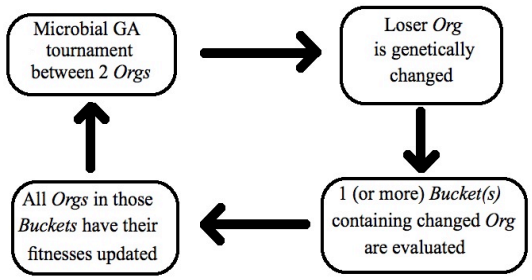


Figure 5. The Binomic GA.

With some small probability V (for vertical gene transmission) *L* is made an identical clone of *W* and also inherits its fitness, that is then updated as below. Otherwise (horizontal gene transmission) some proportion REC of the genes of *W* are copied over so as to replace those genes in *L*.

The tournament is completed by mutating *L*, and then evaluating a number (1 or more) of random *Buckets* that each contain *L*. In this way its inherited fitness is updated, along with the fitnesses of those other *Orgs* that happened to share those *Buckets*. The mutation will be a limited change in the *Org*, either retaining its functionality whilst making a small change in some parameter value (such as, in the case of ANNs, a neural network weight) or making a small change in functionality (such as, with ANNs, adding or deleting a connection between nodes). This should become clearer with a worked example that follows a brief explanation of niching.

Implicit Niching

To understand how implicit niching can occur in an algorithm like this, let us illustrate with a cartoon example. Suppose a population has 4 types of entity, *bread*, *butter*, *jam* and diverse *garbage*. The only collection that has any value is a *bread*+*butter*+*jam* sandwich. When fitnesses are allocated through assessing the value of a *Bucket* of such individuals, we can see that *garbage* would tend to decrease. But further, consider what happens if one of the useful components, e.g. *jam*, is in much shorter supply than the others. Then as a consequence of some *Buckets* containing *bread* and *butter* but no *jam* (and hence valueless), the relative fitness allocated to those individuals will decrease; whereas the relative fitness of *jam* (that will under these circumstances almost always complete a sandwich) increases. In this way, all these different component parts will tend towards similar proportional representation in the population as a whole.

Altruism and cheating. Any procedure that uses some form of group selection raises concerns about the possibility of cheating. If fitness is allocated collectively, why should an individual altruistically contribute to the common good, why not benefit from others' efforts whilst making no contribution itself? This potential pitfall is avoided by the use of *Buckets* allocating fitness within a temporary local subset of the whole *Sea*, even if that subset is taken at random from the whole well-mixed *Sea*. Restricting *Buckets* to (overlapping) local regions within a geographically distributed *Sea* provides yet more pressure to eliminate cheats and garbage.

Evolving ANNs with a Binomic GA

The SANE algorithm, discussed above, implemented the equivalent of *Orgs* as subsets of a 3-layer ANN, each one based on a single node in the middle (Hidden) layer with connections and weights to genetically specified Input or Output nodes. We can generalize this to ANNs of arbitrary topology (including recurrent networks such as CTRNNs) by first making each *Org* in principle equivalent to the whole fully-connected ANN; but then setting the majority of connections between nodes to zero, with a small subset of genetically specified non-zero weights. We can maintain, throughout evolution, the typical proportion of weights that are non-zero by monitoring the add-link and delete-link components of the mutation operator. Thus if, for instance, at any mutation each non-zero weight was mutated to zero with probability 9%, and each zero weight mutated to a non-zero value with probability 1%, we can expect the proportion of non-zero weights to stay around 10%. In this manner, each *Org* is a only a small part of the whole possible ANN, and may very well be functionless on its own through having no connected path from inputs to outputs.

When a *Bucket* is assembled, then this is treated as the full ANN with any specific weight on a connection calculated as the sum (an alternative method would be to use the mean) of all values for that connection as specified on all the constituent *Orgs*; a variant method with subtle differences would be to exclude any zero values in the calculation of such a mean.

Designing an Autoencoder ANN with a Binomic GA

As a working demonstration we chose to use the Binomic GA to evolve ANNs in the form of an autoencoder, as described below. This allows us to compare performance with other versions of evolutionary algorithms that we had developed for similar autoencoders in a separate study.

Such autoencoders (Hinton and Salakhutdinov, 2006) are ANNs with a feedforward succession of layers, potentially fully connected between each successive layer. When appropriate weights are found, it should reduce high-dimensional input data through a lower-dimensional Bottleneck layer and then recover the input pattern and replicate it at the final output layer. Between Input and Bottleneck there is a Hidden Layer, which should encode the input pattern into the Bottleneck; thereafter a further Hidden Layer should decode to the Output.

We used autoencoders of the form N-h-M-h-N (see Figure 6), where N is number of Inputs/Outputs, M is the size of the Bottleneck layer, and h is the size of each Hidden layer. In our

implementation, all inputs were either 1 or -1. The Hidden Layer transfer functions were hyperbolic tangents, whereas the Bottleneck transfer function was linear. The output layer transfer function was a discrete step function that mapped positive/negative values into +1/-1 respectively. For simplicity, no biases were used in any of the networks.

We report here initial results on evolving with the Binomic GA appropriate weights for such autoencoders of sizes 3-12-2-12-3 and 4-24-4-24-3. Evaluations of such networks tested every possible binary input pattern and assessed how many output patterns matched. We compared performance of the Binomic GA (BGA) with two versions of a straightforward Microbial GA (recombination or 'infection' rate 0.5) where each individual in the population was a complete autoencoder with the appropriate architecture and genotypes specifying all the weights. The Microbial GA versions differed in mutation method: either a single weight was mutated, or all weights were mutated together.

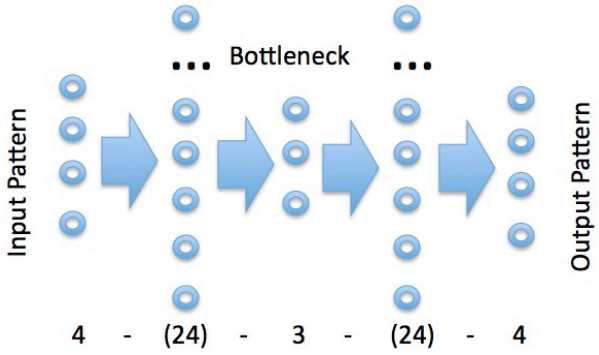


Figure 6. The task for this 4-24-3-24-3 autoencoder ANN is for the binary Input Pattern (here 4 bits) to be replicated at the final Output Layer, despite having passed through a narrower Bottleneck (here 3 nodes) in the middle.

Parameters used

We report on initial BGA experiments using a population or *Sea* of 50 *Orgs*, where each *Org* was a subset of the full autoencoder with (initially) 50% of the weights set to zero, the rest set to small random numbers with mean zero and standard deviation 0.1. Each *Bucket* took 25 *Orgs* at random from the *Sea*, and superimposed these on each other to form an autoencoder with weights on each connection equal to the sum of the respective weights on each *Org*. The fitness score of this *Bucket* was allocated equally to all of its component *Orgs*, their fitnesses updated with a smoothing factor R=0.1. No geographical demes were used.

Each tournament took two *Orgs* at random from the *Sea*, and determined Winner and Loser depending on their current fitnesses. The Loser was modified with a probability 0.5 of Vertical Gene Transmission (becoming a copy of the Winner), otherwise Horizontal Gene Transmission occurred (with 50% of the Winner's genes, or genetically specified weights, overwriting the corresponding Loser's genes). In order to maintain the proportions of zero/non-zero weights at around the initial 50/50 ratio, each non-zero weight in the Loser was deleted (set to zero) with probability (Number of non-zero weights)/(Number of weights) and conversely each zero weight was made non-zero, set to an initial small random value, with probability (Number of zero weights)/(Number of weights). Then a single non-zero weight of the Loser was mutated by adding a mutation, mean value 0.0, standard deviation 0.5 (the same mutation method as used with the single-weight-mutation Microbial GA).

Each time a tournament was completed, and the Loser thus modified, one *Bucket* containing the Loser was evaluated and all the *Orgs* within that *Bucket* had their fitnesses adjusted. This completes the Binomic GA cycle.

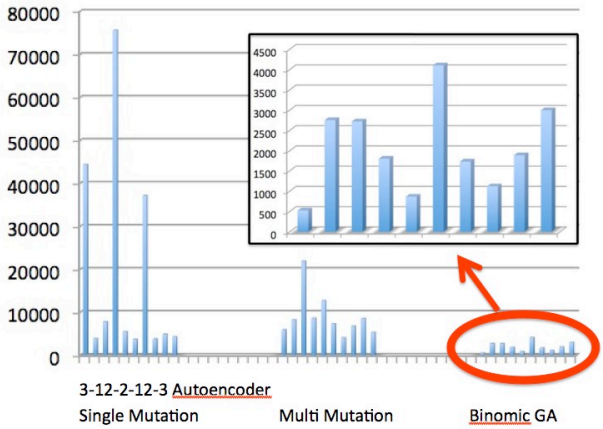


Figure 7. Number of evaluations needed to achieve a perfect score using 3 different GA methods (10 runs each) on the 3-12-2-12-3 autoencoder. The Microbial GA, with single weight mutation, took mean 19,092, std. dev. 24,932, maximum 75,623 evaluations; with multiple mutations 8,921, 5,118, 21,868 respectively. The Binomic GA took mean 2,052, std. dev. 1,098, maximum 4,105 evaluations, and is shown rescaled in the insert.

Experimental Results

For making comparisons, we take the significant factor to be the number of autoencoders that need evaluating before a perfect score is achieved. Runs were terminated if no success was achieved by a cutoff point. Each experiment was repeated 10 times; as is common with GAs, there was variance between runs; but there was a clear and striking pattern. The Binomic GA clearly outperformed its competitors.

We show in Figures 7 and 8 results for the 3-12-2-12-3 autoencoder, and the more difficult 4-24-3-24-4 autoencoder. In both cases the Binomic GA reliably generated perfect results, overall significantly faster than the competing methods, and with less variance. These are initial tests to demonstrate in principle that this method works, and it is gratifying to see the striking performance.

Discussion

GAs have been based on a traditional view of Darwinian evolution with individuals being evaluated for their fitness, and vertical gene transmission down the generations. Metagenomic studies have recently started to transform our

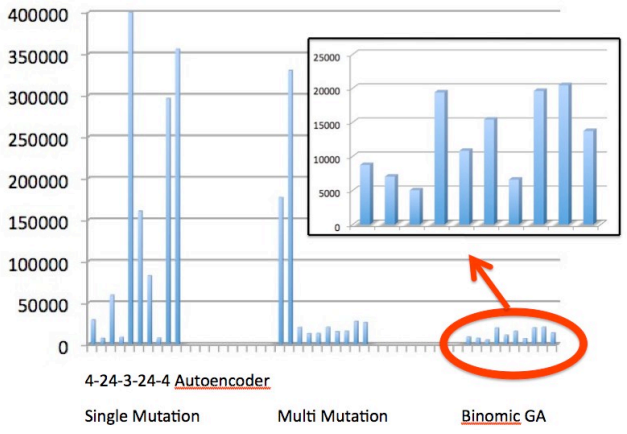


Figure 8. Results similarly shown for the 4-24-3-24-4 autoencoder. The Microbial GA, single weight mutation, took mean 140,608, std. dev. 154,047, maximum (cutoff without success) 400,000 evaluations; with multiple mutations 65,681, 105,408, 329,895 respectively. The Binomic GA took mean 12,681 evaluations, std. dev. 5,856, maximum 20,454.

view of evolution in the world of bacteria, which were amongst the earliest living entities and continue to play an enormous, often under-appreciated, role. We have highlighted the symbiotic nature of evaluations in communities of such real organisms, as emulated in part in the artificial world with LCS and SANE. We have shown how the horizontal gene transmission of bacteria is emulated in the Microbial GA. But as yet nobody appears to have combined these two aspects into applications in AL or EC.

This is primarily a position paper drawing attention to this lack of AL/EC work inspired by Metagenomics, despite significant traffic in the other direction. We propose a new sub-field of *Binomics* bringing these two ideas together as potentially fruitful in synthetic applications. The *Binomic GA* has been demonstrated to work well in preliminary tests, and this new approach opens up a whole range of new questions.

We need to investigate what parameter settings work well for what kind of problem. Does the autoencoder problem have some special property that is relevant? We note a potential relationship with neutral networks in the fitness landscape. The effects of varying *Bucket* size and the impact of drawing the *Buckets* locally within the *Sea* need to be studied. Taking account of this Metagenomic inspiration, we may expect that an appropriate application could be Evolutionary Computation that needs to be carried out online, with the evolving population actually carrying out its function in real time whilst adapting to environmental changes. One such example could be anti-virus (the computer variety of virus) software agents where a diverse population protects a system in real time, whilst reacting and adapting to new environmental threats.

Our preliminary work with the BGA leads us to believe that there is enormous scope for further developments. We hope this paper will stimulate interest in what has been until now a surprising gap in Artificial Life studies.

References

Bull, L., editor (2004). *Applications of Learning Classifier Systems*. Springer.

Committee on Metagenomics (2007). *The New Science of Metagenomics: Revealing the Secrets of Our Microbial Planet*. National Research Council, National Academies Press, Washington, DC. Downloadable from *www.nap.edu/catalog/11902.html*

Eisen, J. A. (2007). Environmental Shotgun Sequencing: Its Potential and Challenges for Studying the Hidden World of Microbes. *PLoS Biol.* 5(3): e82. Doi: 10.1371/journal.pbio.0050082.

Farmer, J. D. (1990). A Rosetta stone for connectionism. In Forrest, S., editor, *Emergent Computation: Proc. 9th Int. Conf. of the Center for Nonlinear Studies on Self-organizing, Collective, and Cooperative phenomena in Natural and Artificial Computing Networks,* pages 153-187. Amsterdam: North-Holland.

Forrest, S. and Miller, J. (1991). Emergent Behavior in Classifier Systems. *Physica D* 42: 213-217.

Goldberg, D. E., and Richardson, J. (1987). Genetic algorithms with sharing for multimodal function optimization. In J. Grefenstette (Ed.), *Proceedings of the Second International Conference on Genetic Algorithms*, pages 41-49. Hillsdale, NJ: Lawrence Erlbaum Associates.

Handelsman, J. (2004). Metagenomics: Application of Genomics to Uncultured Microorganisms. *Microbiology and Molecular Biology Reviews*, 68(4): 669-685.

Harvey, I. (2001). Artificial Evolution: a Continuing SAGA. In Gomi, T., editor, *Evolutionary Robotics: From Intelligent Robots to Artificial Life.* Springer-Verlag LNCS 2217.

Harvey, I. (2010 in Press). The Microbial Genetic Algorithm. In G. Kampis et al., editors, *Proc. of Tenth Eur. Conf. on Artificial Life.* Springer.

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786): 504-577.

Holland, J. H. (1976). Adaptation. In Rosen, R. and Snell, S. N., editors, *Progress in Theoretical Biology*, 4. Plenum.

Holland, J. H. and Reitman, J. H. (1978). Cognitive Systems Based in Adaptive Algorithms. In Waterman, D. and Hayes-Roth, F., editors, *Pattern-directed Inference Systems*. Academic Press.

Horn, J., Goldberg, D. E. and Deb, K. (1994). Implicit Niching in a learning classifier system: Nature's way. *Evolutionary Computation*, 2(1): 37-66.

Moriarty, D. E. and Miikkulainen, R. (1996). Efficient Reinforcement Learning through Symbiotic Evolution. *Machine Learning*, 22: 11-33.

Moriarty, D. E. and Miikkulainen, R. (1999). Learning Sequential Decision Tasks. In Honavar, V., Patel, M. and Balakrishnan, K., editors, *Advances in the Evolutionary Synthesis of Neural Systems.* MIT Press, Cambridge, MA.

Penn, A. and Harvey, I., (2004). The Role of Non-Genetic Change in the Heritability, Variation and Response to Selection of Artificially Selected Ecosystems. In Pollack, J., Bedau, M., Husbands, P., Ikegami, T., and Watson, R.A., editors, *Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems, ALIFE'9*, pages 352-357. MIT Press, Cambridge MA.

Qin, J., Li, R. et al. (2010). A human gut microbial gene catalogue established by metagenomic sequencing. *Nature*, 464: 59-67.

Smith, R. E. and Brown Cribbs, H. (1994). Is a Learning Classifier System a Type of Neural Network? *Evolutionary Computation*, 2(1): 19-36.

Spector, L. and Klein, J., (2005). Trivial Geography in Genetic Programming. In Yu, T., Riolo, R.L., Worzel, B., editors, *Genetic Programming Theory and Practice III*, pp. 109-124. Boston, MA: Kluwer Academic Publishers.

Venter, J. C., Remington, K. (and 21 further co-authors) (2004). Environmental genome shotgun sequencing of the Sargasso Sea. *Science* 304(5667): 66-74.

Using Meta-Genetic Algorithms to tune parameters of Genetic Algorithms to find lowest energy Molecular Conformers

Zoe Brain[1] and Matthew Addicoat [2]

[1]School of Computer Science, Australian National University
[2]Research School of Chemistry, Australian National University
zoe.brain@anu.edu.au

Abstract

Determining the electronic structure of long chain molecules is essential to the understanding of many biological processes, notably those involving molecular receptors in cells. Finding minimum energy conformers and thus electronic structure of long-chain molecules by exhaustive search quickly becomes infeasible as the chain length increases. Typically, resources required are proportional to the number of possible conformers (shapes), which scales as O(3^*L*) where *L* is the length. An optimized genetic algorithm that can determine the minimum energy conformer of an arbitrary long-chain molecule in a feasible time is described, using the tool, PyEvolve. The method is to first solve a generic problem for a long chain by exhaustive search, then by using the pre-determined results in a look-up table, to make use of a Meta-GA to optimize parameters of a simple GA through an evolutionary process to solve that same problem. By comparing the results using the tuned parameters obtained by this method with the results from exhaustive search on several molecules of comparable chain length we have obtained quantitative measurements of an increase in speed by a factor of three over standard parameter settings, and a factor of ten over exhaustive search.

Introduction

In computational chemistry, there is a requirement to determine minimal energy conformers (shapes) of molecules such as dipeptides using a high level of theory, in order to determine their molecular properties. Typically there are thousands of such possible shapes for any particular molecule, and the calculation of energy for each would take O(10e3) CPU-seconds at 500 GFlops for a relatively simple level of chemical theory, but O(10e7+) for successively more complex levels. Traditionally, the method has been to determine a good subset at one level of theory, then use these as candidates for the next level, then take a further, smaller subset at that level, and so on until the required level of theory had been reached.

Various levels of theory are used to determine energies. These vary from the semi-empirical AM1 (Austin Model 1)[1] and PM3 (Parametrized Model 3)[2][3] methods often used on such computationally intensive problems, through the higher level B3LYP (Becke, three-parameter, Lee-Yang-Parr) density functional [4][5] whose formal scaling is to the fourth power, the MP2 (Møller–Plesset 2nd order) method [6] which scales to the fifth power, the CCSD (Coupled-cluster including Single and Double excitations) model which scales to the sixth power, and with inclusion of iterative Triples i.e. CCSD(T) scales to the seventh power [7]. The computational resources required to determine the energies of all conformers of a general molecule are determined by the length *L* – and are typically O(3*L*). Beyond length 10, the problems become infeasible using B3LYP/6-31+g(d,p) and exhaustive search techniques[7]. An increase of efficiency of one order of magnitude would therefore allow either an increase in length of 2, or one higher level of theory, while consuming the same or less computational resources.

The initial molecule chosen for experimentation is the dipeptide carnosine (Figure 1), as the exhaustive search results were already available from previous work. Further molecules were examined later. The landscape of conformer energies for a dipeptide of length 8 such as carnosine correspond to an 8-dimensional manifold, with occasional gaps due to some molecular configurations resulting in infeasibly small inter-atomic distances.

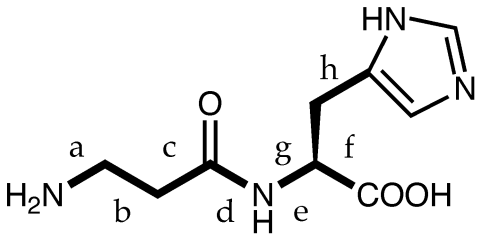


Figure 1. Carnosine

Carnosine (D-alanyl-L-histidine) is a dipeptide found in several human tissues, particularly skeletal muscle, heart tissue and the brain [9]. Its functions in each of these tissues is not well understood, but studies have shown that it possesses potent antioxidant properties, protects against neuronal cell death and that its zinc salt promotes the healing of peptic ulcers [10].

Carnosine may be considered to have 8 rotatable bonds (labeled a..h in Figure 1). Work by Izgorodina et al [11] . indicates that when starting from a previously optimized structure, 120 degree resolution is generally sufficient to map

sigma bonds. At this resolution, this yields 3^8 = 6561 possible conformers (shapes) of carnosine, some of which may be inaccessible as the combination of rotations places atoms at or near the same coordinates. In addition, it is known that in neutral polypeptides, rather than adopting its normal shape (Structure A, Figure 2), the carboxylic acid hydrogen may point away from the carboxylic acid group to form an intramolecular hydrogen bond with the amide oxygen atom (Structure B, Figure 2). As this does not correspond to a 120° rotation, these two structures are considered separately in this paper.

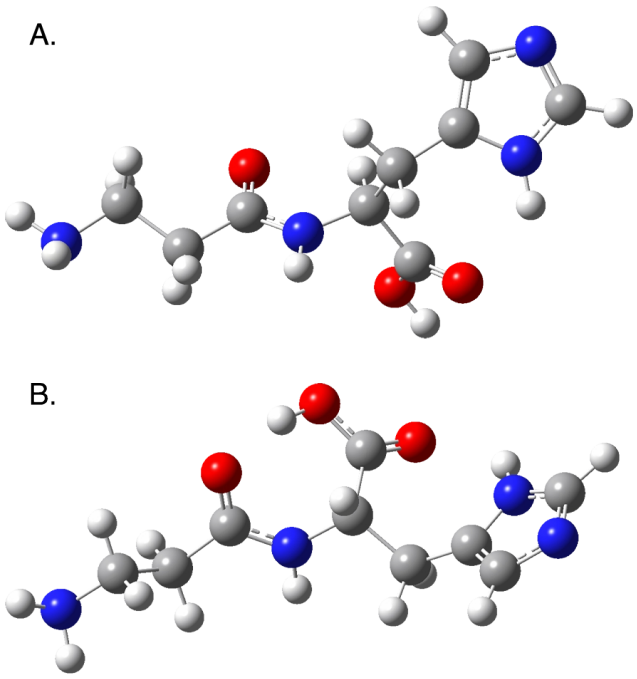


Figure 2 Optimized A and B Structures of Carnosine

Simple GAs were a plausible candidate for finding the minimum energy conformers, but the best parameter values to be used in them were unknown. While there is considerable heuristic knowledge about these values for particular problem domains, there has been very little systematic research investigating the interaction between the different parameters that are used to define GAs. Research has been mainly confined to modifying one or two of the parameter values, keeping all the others constant. Work has recently concentrated on optimizing a particular GA for a particular problem, injecting more and more domain-knowledge into the genetic representation, and making the GA more and more specialized. This has been found to be a very fruitful line of research, with large degrees of optimization having been achieved. Most knowledge we have is on the effect of varying population size and mutation rate parameters in isolation, with the rest having been assigned arbitrary values.[12][13]. Nannen's results [14] using 120 different combinations of Evolutionary Algorithm (EA) operators on 4 different generic problems using the generic information-theoretical metric of Shannon Entropy found the different components varied greatly in importance, but did not give practical optimum values for different classes of problem. Other methods used include statistical or theoretical analysis.[15][16]

The use of Meta-GAs to optimize parameters and thus tune GAs was first proposed by Grefenstette [17] and continued by Friesleben and Hartfelder [18] in 1993. de Laangraaf [19] showed that the performance of Meta-GA optimized simple GAs was at least comparable to those of adaptive ones. We therefore use Meta-GAs to optimize the simple GAs that calculate the lowest-energy conformers.

Aim

Our object was to provide computational chemists with little or no experience in the use of GAs with a "turnkey" method of determining minimum energy conformers of molecules.

What was needed was a set of default parameters to set the GA to to have a reasonably well optimized computational factory for generating candidate low-energy conformers.

As a single calculation for a dipeptide of length 8 using UB3LYP/6-31+g(d,p) takes approximately 25 minutes of CPU time at 0.5 TFLOPS on the National Computational Infrastructure in Australia, exhaustive search calculations beyond this level of theory become computationally infeasible; and even a single calculation of the energy of one length-8 conformer to the CCSD(T) level would take time on the limits of feasibility today. Previous computational studies by Diez [20] on carnosine have featured only two neutral or zwitterionic conformers, the only study to consider the full conformational landscape of carnosine was undertaken using the semi-empirical PM3 method [21].

Exhaustive Search Method

In order to produce the exhaustive search results, both A and B carnosine structures were constructed. Their geometries were optimized using the UB3LYP density functional and the 6-31+g(d,p) basis set [4][5]. All calculations were undertaken using the Gaussian09 suite [8].

The optimized structure was denoted **carnosine-a1b1c1d1e1f1g1h1** and had an energy of -796.150527 hartree. From this structure, internal coordinates for each conformer were generated. Single-point energy calculations, also using UB3LYP/6-31+g(d,p) were undertaken and the energies saved. The optimized geometries of both A and B structures are shown in Figure 2

For the "A" structure of carnosine, the optimized structure was only the second lowest energy structure (ΔE = 4.72 $kJmol^{-1}$). The global minimum corresponding to a1b2c1d1e1f1g1h1 differed by a single rotation and had an energy of -796.152327 hartree. 597 possible conformers were excluded due to infeasibly small interatomic distances. The optimized "B" structure was also similarly low in energy (ΔE 3.85 kJ mol^{-1}) but was only the third lowest in energy. Again, the **a1b2c1d1e1f1g1h1** conformer proved to be the global

minimum and the intervening conformer was the **a2b2c3d1e1f1g1h1** conformer (ΔE 2.44 kJ mol^{-1}). In this case 619 conformers were excluded based on interatomic distances.

The corresponding structures were also calculated using the HF/6-31g (Hartree-Fock)[22,23,24] model chemistry. This is computationally less intensive by a factor of 20. However, results differed significantly from those produced by the UB3LYP/6-31+g(d,p) model, indicating that this is a less desirable technique than using GAs for minimizing computational load. That we are close to the limits of computing feasibility is shown by the fact that this is the first time the conformational preference of the gas-phase structure of carnosine has been calculated to the UB3LYP/6-31+g(d,p) level of theory. The calculations for the conformers of carnosine-A took 2300 CPU hours on 2.93 GHz Intel Nehalem CPUs.

Meta-Genetic Algorithm Method

Table 1: Canonical Parameters of a Simple GA

Parameter	Values	Arguments
Crossover	OX Uniform Two Point One Point	Probability
Mutators	Swap	Probability
	Binary	Probability
	Gaussian	Probability Mean Standard Deviation Minimum Maximum
	Uniform	Probability
Parental Selection	Tournament	Tournament Size
	Uniform	
	Rank	
	RouletteWheel	
Survivor Selection	Elitism (True or False)	
Population Size	Positive binary	

A Meta-GA was used to tune the parameters of a simple GA, which in turn determined the energy of the conformers of the dipeptide carnosine. The parameters of a simple GA are shown in Table 1. Our Meta-GA genome was implemented as a 1D list of 12 integers between 1 and 1000, as shown in Table 2.

The problem of dealing with multiple mutually exclusive choices was dealt with by using "winner take all" probability densities. That is, given 3 possibilities, A, B and C, and if the PD of A was 500, B was 900, and C was 600, then the highest (B in this case) would always be chosen.

Table 2 - Genome Representation of GA Parameters

Parameter	Range	
Population Size	5-1000	1-1000 with a Floor of 5
Uniform X-Over Probability Density	1-1000	Exclusive with OX, 1-Pt, 2-Pt, None.
One-Point X-Over Probability Density	1-1000	
Two Point X-Over Probability Density	1-1000	
No X-Over Probability Density	1-1000	
Binary Mutator Probability	0.001-1.000	1-1000 thousandths
Swap Mutator Probability	0.001-1000	1-1000 thousandths
Roulette Selector Probability Density	1-1000	Exclusive with Tournament, Uniform, Rank
Tournament Selector Probability Density	1-1000	
Uniform Selector Probability Density	1-1000	
Rank Selector Probability Density	1-1000	

The swap mutator could be used on its own, or in addition to binary mutation. Tournament Size was left at the default value of 2. Mutator probability is defined in PyEvolve[25] as the proportion of the genome where a mutation is attempted, each mutation having a probability of the mutator probability. Thus a chromosome of length 4 and a mutator probability of 0.5 would have two of its genes selected randomly possibly mutated, each with a probability of 0.5. Elitism was enabled. No tuning of the Meta-Ga itself was attempted: the default parameters of the PyEvolve toolset were used. These are Parent Selector:Rank; Tournament Size:2; Swap:Enabled, Mutation rate:0.02, Population Size:80, Crossover:1 Point, Crossover rate: 0.5.

The Meta-GA termination condition was initially set to 20 generations, but later increased to 100 to confirm convergence. This Meta-GA was run 100 times yielding 100 different optimized parameter sets, with the corresponding fitness (number of computations required using that set) for each.

The termination condition for the conformer determining GA was when the the minimum energy conformer (known a-priori) had been found. The fitness for a GA with that set of parameter values was the number of computations required to obtain this minimum.

For the GA to determine minimum energy conformers, the genome was encoded as a simple vector of eight trinary numbers (**a,b,c...g,h**) each value corresponding to one of the three allowed positions of the corresponding bond.

In this case, the difference between Binary and Gaussian mutators were not examined – the differences between a flat distribution and a Gaussian distribution, are negligible in the range 1..3. OX crossover was not appropriate for this representation, so was not used.

To determine the energy for each conformer was just a matter of using a look-up table on the values for energy previously determined by the exhaustive search method. This enabled experimentation to be performed using significant quantities of evaluations. To calculate these values any other way would have taken many orders of magnitude more time. On the supercomputer network used in the experiment one such calculation took 20-30 CPU minutes.

After tuning of the GA parameters, each of the 100 fittest tuned GAs was run 100 times to gain some measure of reliability, as some of the associated parameters were probabilistic. Therefore the outcomes were stochastic not deterministic.

Initial experiments [26] only looked at population size, mutation rate, and 1-point crossover rate with Elitism enabled, the other values being set to the PyEvolve defaults (Parent Selector :Rank; Tournament Size:2; Swap:Enabled). These were applied to carnosine, and then to other molecules of comparable size to evaluate the general applicability of the technique.

Results

Exhaustive Search Calculations

Calculated energies represent the stabilization of the molecule compared to all of its constituent particles (nuclei, electrons) separated to infinity and thus are negative quantities. To use linear scaling within PyEvolve, positive raw scores are required, therefore the fitness of any given conformer is made equal to zero minus its energy and the normal chemical problem of minimization becomes a maximization problem within PyEvolve. Figure 3 shows the negative energy (0 - E) of the 5970 non-excluded carnosine A conformers, 1288 of these conformers have an energy within 0.05 a.u. of the global minimum. This energy range is shown expanded in Figure 4. In each figure, "Conformer ID" represents the encoded genome, minus the alpha characters (i.e. **a1b1c1d1e1f1g1h1** => 11111111) and listed in numeric order. Thus the vertical series of points apparent in Figure 4 represent sets of conformers where the first five bonds (a—e) are conserved.

Some general conclusions about conformer stability can be drawn from Figure 3. The very highest energies, clustered around -791 a.u. occur in three sets of three, corresponding to genomes of the form **a**[1-3]**b2c3d2e1f**[1-3]**g2h3.** These conformers all have the imidazole ring in extremely close proximity to the terminal NH2 group. The second cluster of high energy structures, having energies of approximately -793.055 a.u. also place the imidazole and NH_2 groups in close proximity. These conformers correspond to genomes of the form **a**[1-3]**b1c3d3e3f**[1-3]**g2h3** or **a**[1-3]**b3c2d3e3f**[1-3]**g2h3.**

In contrast, the lowest five energy structures all preserve the final 5 bits of the genome as their optimized (original) values – i.e. **d1e1f1g1h1**. These conformers span an energy range of 11.4 $kJmol^{-1}$, only just over the 10 $kJmol^{-1}$ range that is typically considered chemically relevant. The 10 lowest energy conformers of carnosine A are shown in Table 4. The conservation of this portion of the molecule is even more pronounced in the B structure of carnosine, where the 15 lowest energy structures all preserve the original histidine conformer, as shown in table 5.

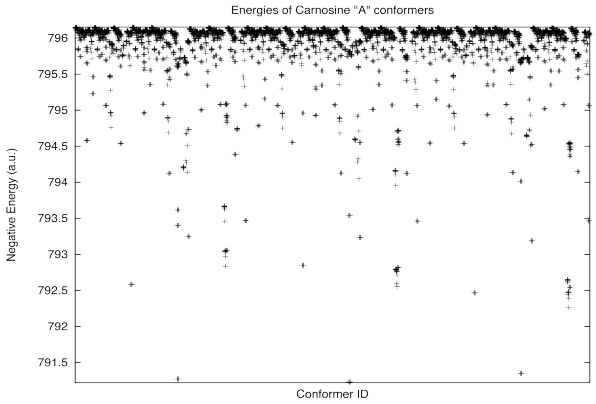


Figure 3 – Energies of All Carnosine A Conformers

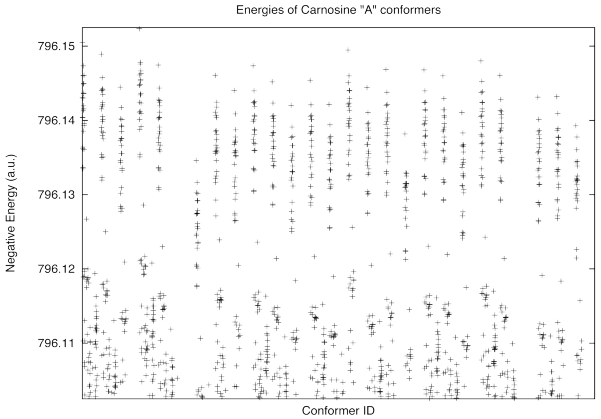


Figure 4. Energies below 796.1025 a.u. Scale shows 0-E

This greater conservation is expected, given the stabilization provided by the intramolecular hydrogen bond present in carnosine B, which effectively fixes bonds (d—g).

Table 2. 10 Lowest Energy Conformers of Carnosine A

E(UB3LYP/6-31+g(d,p))	Genome
-796.152327	a1b2c1d1e1f1g1h1
-796.150527	a1b1c1d1e1f1g1h1
-796.149450	a2b2c3d1e1f1g1h1
-796.148919	a1b1c2d1e1f1g1h1
-796.147980	a3b2c1d1e1f1g1h1
-796.147800	a1b2c1d1e1f2g3h1
-796.147574	a1b2c1d1e1f2g2h1
-796.147441	a1b2c2d1e1f1g1h1
-796.147369	a2b1c1d1e1f1g1h1
-796.147337	A1b1c1d1e1f2g2h1

Table 3. 15 Lowest Energy Conformers of Carnosine B

E(UB3LYP/6-31+g(d,p))	Genome
-796.157523	a1b2c1d1e1f1g1h1
-796.156592	a2b2c3d1e1f1g1h1
-796.156054	a1b1c1d1e1f1g1h1
-796.155025	a1b3c2d1e1f1g1h1
-796.155001	a1b1c2d1e1f1g1h1
-796.153013	a1b2c2d1e1f1g1h1
-796.152772	a3b2c1d1e1f1g1h1
-796.152660	a2b1c1d1e1f1g1h1
-796.152270	a3b1c1d1e1f1g1h1
-796.151997	a2b2c1d1e1f1g1h1
-796.151893	a2b3c2d1e1f1g1h1
-796.151554	a3b1c2d1e1f1g1h1
-796.151122	a2b1c2d1e1f1g1h1
-796.150818	a1b1c3d1e1f1g1h1
-796.150469	a3b2c2d1e1f1g1h1

Initial Experiments – Population Size, Mutation rate, 1-D Crossover rate

The top 5 tunings of the GA are shown in table 3. To verify the performance of the GA parameters, the top 5 sets were also used to determine the lowest energy conformer of the B structure of carnosine. Each set of parameters was used 100 times, results are shown in Table 4. All 5 GAs find the global minimum 100% of the time, the worst case required 1056 evaluations (16% of the 5942 conformers). The mean number of evaluations for all GAs was between 176 (2.7%) and 253 (3.9%).

Table 3. Results for Top 5 Parameter Sets Carnosine A

Init Rank	Pop size	Mut rate	XOvr rate	Min Evals	Max Evals	Mean Evals
1	6	0.238	0.156	12	888	218.52
2	2	0.225	0.005	12	1154	220.96
3	31	0.403	0.977	62	930	242.73
4	11	0.341	0.786	33	946	245.3
5	32	0.365	0.810	64	1056	256

Table 4. Results for Top 5 Parameter Sets Carnosine B

Init Rank	Pop size	Mut rate	XOvr rate	Min Evals	Max Evals	Mean Evals
1	6	0.238	0.156	12	1116	175.74
2	2	0.225	0.005	14	886	190.66
3	31	0.403	0.977	62	899	252.96
4	11	0.341	0.786	22	682	181.83
5	32	0.365	0.810	64	1056	253.44

The close agreement of the two sets of mean evaluation counts, both between the different parameter sets, and the different molecules, suggests that the estimates of performance are reliable, and applicable to a broad range of molecular species.

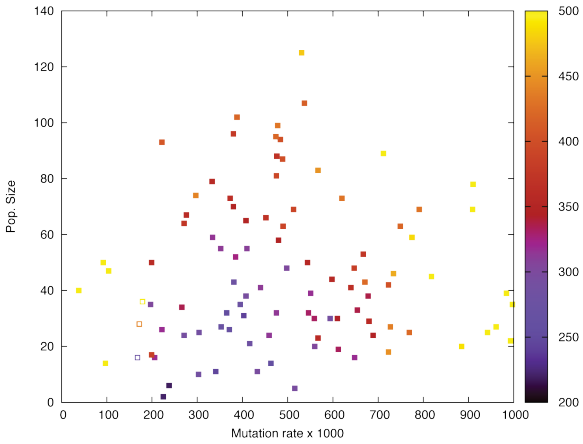


Figure 4. Computational Efficiency as a Function of Population Size and Mutation Rate for Carnosine-A Hollow squares denote parameter sets that did not always find the global optimum.

Figure 4 shows the computational requirements for each pair (p,m) of population size and mutation rate. Crossover rate was not found to affect the GA's fitness. All of the pairs generated a global optimum energy in all 100 runs (success rate 1.00) except for the three points marked as hollow squares.

Table 5. Partly Unsuccessful Parameters

Population Size	Mutation rate	Success Rate	Crossover Probability	Mean Number of Evaluations (Successful Runs only)
28	0.172	0.29	0.474	127.45
16	0.168	0.30	0.873	82.13
36	0.179	0.31	0.050	157.94

A variety of different molecules were downloaded from the Cambridge Structural Database [28]. Molecules were selected to be close to the largest size where exhaustive search was considered feasible (approximately 50 atoms) but to contain a wide variety of structural motifs (linear, branching, planar regions) and chemical functional groups. Using the same technique on a variety of other molecules suggested that the optimum parameters of population size and mutation rate

were valid in general for molecules of similar size. Three test molecules are illustrated in figures 5-7 and the corresponding results in figures 8-10. In figures 8-10, hollow squares denote parameter sets that did not always find the global optimum.

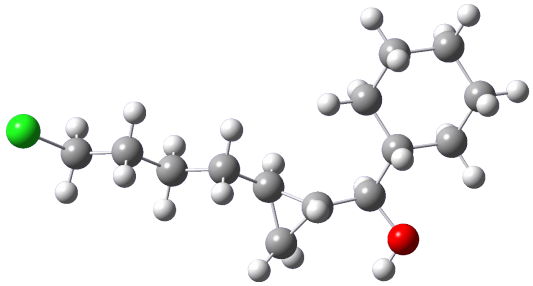


Figure 5 – Optimized Structure of Dawmoe

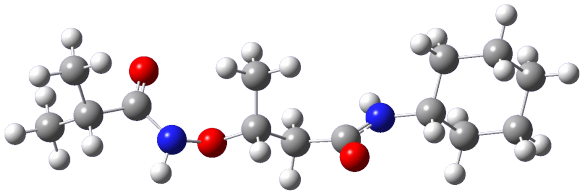


Figure 6 optimized Structure of Exuduy

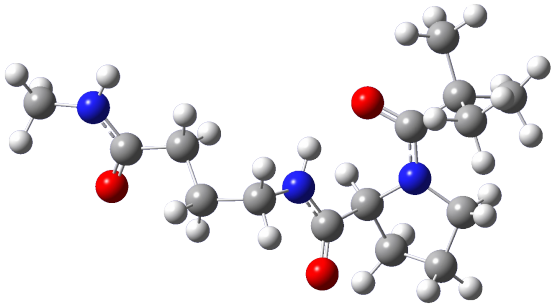


Figure 7 Optimized Structure of Ifevoe

Subsequent Experiments – Tuning all parameters

The use of the swap mutator was found to be strongly deleterious to the reliability of the GA, without any compensatory increase in efficiency. When Elitism and swap were both used, only 10 of the GAs found the global minimum 100% of the time. The use of elitism did not have a significant effect on reliability with only 11 GAs being 100% successful when swap was employed without elitism. 16 GAs were less than 20% reliable.

Elitism strongly affected the efficiency of the GA. Without employing swap, when elitism was employed, the mean minimum, maximum and mean number of evaluations were 147.82, 2800.10 and 794.99 respectively. Not employing elitism, raised these numbers to 157.25, 3457.90 and 991.76 respectively.

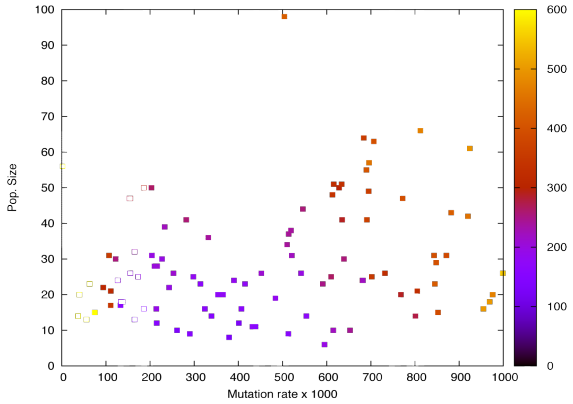


Figure 8. Computational Efficiency as a Function of Population Size and Mutation Rate for Dawmoe.

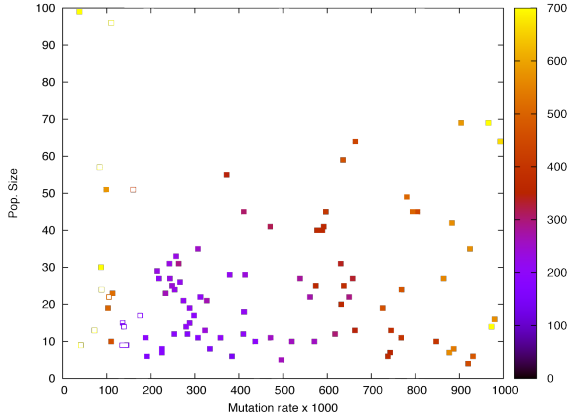


Figure 9. Computational Efficiency as a Function of Population Size and Mutation Rate for Exuduy.

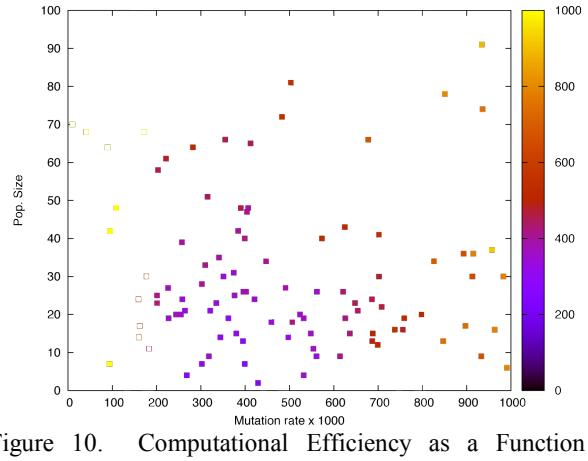


Figure 10. Computational Efficiency as a Function of Population Size and Mutation Rate for Ifevoe.

After running each parameter set 100 times, the 100 GA parameter sets were ranked according to their efficiency, such that the highest rank (100) has the lowest mean evaluation score, i.e. is the most fit parameter set. These rankings were then graphed against selector and crossover methods to determine which methods typically fared well, or conversely, which methods decreased the efficiency of the GA. These inverse rankings are shown in Figures 11 to 14 for the Tournament, Roulette, Uniform and Rank selectors respectively. Figure 15 compares all four selectors. The top 10 GA tunings were applied 100 times to A- and B-carnosine datasets and the mean evaluations are shown in Table 6. The results appear consistent, suggesting the general applicability of these tunings .

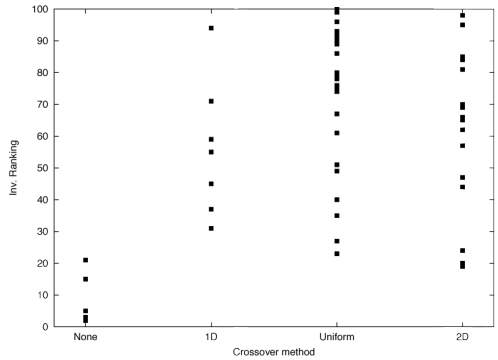


Figure 11 Inverse Rankings of Tournament Parent Selector for different Crossover Methods

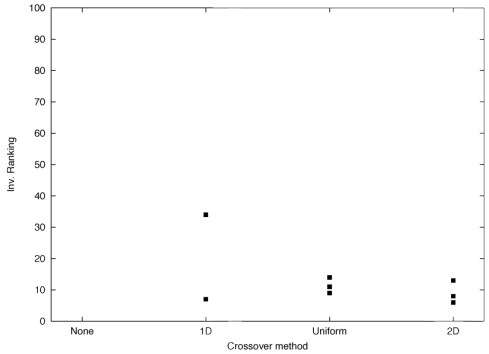


Figure 12 Inverse Rankings of Uniform Parent Selector for different Crossover Methods

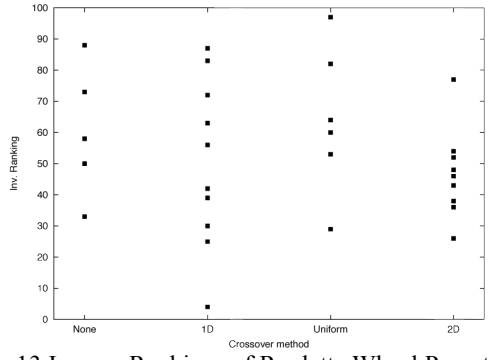


Figure 13 Inverse Rankings of Roulette Wheel Parent Selector for different Crossover Methods

Often it is not only the minimum energy conformer that is of chemical interest, but all conformers within a given energy range, say 10 kJmol-1. With this in mind, an alternate termination criterion was trialled, whereby the five lowest energy conformers were required to exist in the population. This fared very poorly, with success rates of only a few percent and the original termination criteria based on a single raw score was restored.

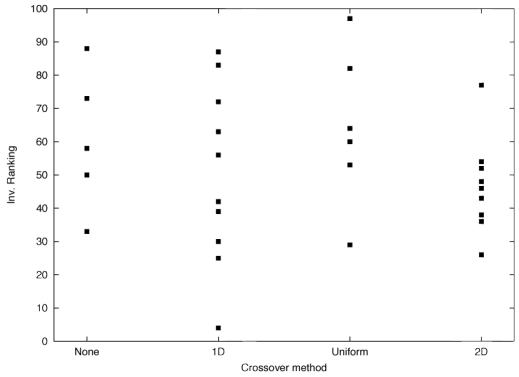


Figure 14 Inverse Rankings of Rank parent Selector for different Crossover Methods

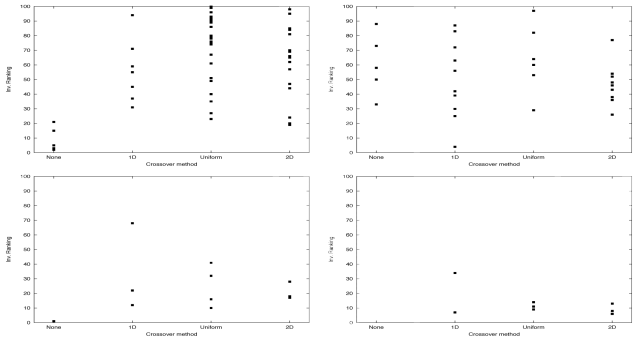


Figure 15 Comparison of all Parent Selectors for different Crossover Methods

Table 6. Performance of GAs with same parameter sets on different datasets

GA(ranked)	A-Carnosine Dataset	B-Carnosine Dataset
1	651.82	627.64
2	270.64	241.06
3	322.07	253.54
4	320.12	301.57
5	372.40	353.15
6	537.68	576.40
7	439.56	495.00
8	555.66	497.70
9	566.26	557.98
10	492.65	438.37

CONCLUSIONS

Unreliable parameter sets were found when mutation rate and population size were both low. This suggests that the algorithm degenerates into simple stairclimbing in those regions, and success depends on initial conditions. If the initial population contains values near the global optimum, performance is very good, but if not, the low mutation rate means that the result may be stuck in a local optimum.

Both Rank and Tournament parental selection far out-performed Uniform selection and Roulette-Wheel selection. Tournament selection plus uniform crossover appeared to be the most reliable. Tournament selection plus no crossover performed poorly : but Rank selection performed well with no crossover. Optimum population size was less than 100, and usually less than 50: optimum mutation rate was less 0.7 and usually less than 0.5. Examination of the populations revealed many duplicates of the lowest energy conformer. This suggests that incest-prevention is required in order to obtain results containing sets of near-optima for this method.

No improvement on default values was observed using optimized selection/crossover/mutation values for the best-performing mutation rate/population size combinations, except for replacing the swap mutator with the binary mutator. PyEvolve using default values, except for population size ~30, mutation rate of ~0.35, elitism and binary mutator as described in [26] is therefore recommended for use by computational chemists to locate global minimum conformers.

Without effective removal of duplicates or Incest prevention [27], a future implementation could work-around the problem of finding sets of lowest energy conformers by searching for the lowest energy conformer, then once that is identified, excluding it and looking for the next lowest energy conformer until the desired number of conformers were identified. A lookup table with the results of each energy calculation, means that later runs would undertake far fewer of these calculations.

Acknowledgments

This work was based on initial research funded in part by the Australian Government under the Auspices of the Co-Operative Research Centre for Advanced Automotive Systems (Auto-CRC). Supercomputer facilities were provided by the National Computing Infrastructure (NCI).

References

[1] Michael J. S. Dewar, Eve G. Zoebisch, Eamonn F. Healy, James J. P. Stewart (1985) Development and use of quantum mechanical molecular models. 76. AM1: a new general purpose quantum mechanical molecular model *Journal of the American Chemical Society* 1985 107 (13), 3902-3909

[2] James J. P. Stewart (1989) Optimization of parameters for semiempirical methods I. Method J*ournal of Computational Chemistry* Volume 10, Issue 2, Date: March 1989, Pages: 209-220

[3] James J. P. Stewart(1989) Optimization of parameters for semiempirical methods II. Applications *Journal of Computational Chemistry* Volume 10, Issue 2, Date: March 1989, Pages: 221-264

[4] Becke A.D. (1993), Density-functional thermochemistry. III. The role of exact exchange, *J. Chem. Phys*. 98 1993 5648-5652

[5] Stephens P.J. , Devlin F.J. , Chabalowski C.F. , Frisch, M.J. (1994) Ab initio calculation of vibrational absorption and circular dichroism spectra using density functional force fields, *J. Phys. Chem.* 98 1994 11623-11627.

[6] Møller C., Plesset M.S. (1934). "Note on an Approximation Treatment for Many-Electron Systems" (abstract). *Phys. Rev.* 46: 618–622.

[7] Cizek, J. (1966) On the correlation problem in atomic and molecular systems" *J. Chem. Phys*. 45, 4256 1966

[8] Frisch, M. J. et al (2004) Gaussian 09, Revision A.01, Gaussian, Inc., Wallingford CT, 2004.

[9] Quinn, PJ; Boldyrev, AA; Formazuyk(1992) VE *Mol Aspects Med* Vol. 13 379 1992

[10] Matsukura, T; Tanaka, H (2000) *Biochemistry (Moscow)* Vol. 65 961 2000

[11] Izgorodina E., Lin L., and Coote M.L. (2007) "Energy-Directed Tree Search: An Efficient Systematic Algorithm for Finding the Lowest Energy Conformation of Oligomeric Molecules", *Phys. Chem. Chem. Phys*., 2007, 9, 2507-2516

[12] Yu-an Zhang, Makoto Sakamoto, Hiroshi Furutani, (2008) "Effects of Population Size and Mutation Rate on Results of Genetic Algorithm," *Fourth International Conference on Natural Computation,* vol. 1, pp. 70-75

[13] Wolpert D. and Macready W (1997) "No free lunch theorems for optimisation". *IEEE Transactions on Evolutionary Computation*, 1(1):67-82, 1997

[14] Nannon V., Smit S.K., Eben A.E. (2008) "Costs and Benefits of Tuning Parameters of Evolutionary Algorithms" *Parallel Problem Solving from Nature* 2008

[15] Grefenstette J (1986), Optimization of control parameters for genetic algorithms, I*EEE Transactions on Systems, Man and Cybernetics,* v.16 n.1, p.122-128, Jan./Feb. 1986

[16] Nakama T. (2008), "Theoretical analysis of genetic algorithms in noisy environments based on a Markov Model". *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation* GECCO08 pp1001-1008.

[17] Smit, S. K. and Eiben, A. E (2009) Comparing parameter tuning methods for evolutionary algorithms *Proceedings of the Eleventh conference on Evolutionary Computation* CEC09 pp 399-406

[18] Friesleben, B., Hartfelder, M.(1993): "Optimization of Genetic Algorithms by Genetic Algorithms". In: Albrecht, R.F., Reeves, C.R., Steele, N.C. (eds.) *Artificial Neural Networks and Genetic Algorithms,* pp. 392–399. Springer, Heidelberg 1993

[19] De Landgraaf W.A. (2006), *Parameter Calibration Using Meta-Algorithms*, Master's Thesis, Artificial Intelligence Vrije Universiteit Amsterdam 2006

[20] Diez, R.P., Baran, E..J. (2003) *Journal of Molecular Structure – Theochem* Vol. 621(3) 245-251 2003

[21] Klyuev, S. A. (2006) *BIOFIZIKA* Vol. 51(4)669-672 2006

[22] Hartree D. R. (1928), Proc Cambridge Phil Soc 24 ,89,111,246

[23] Fock V(1930) , *Z Phyzik,* 61, 126

[24] Slater J.C(1930) , *Phys Rev*, 35, 210

[25] Perone C.S.(2009) PyEvolve a Python Open-Source Framework for Genetic Algorithms *ACM SIGEvolution* Vol 4 Issue 1 2009

[26] Addicoat, M.A, Brain Z.E. (in press) Using a Meta-GA for Parametric Optimization of Simple GAs in the Computational Chemistry Domain To appear in the *Proceedings of the Genetic and Evolutionary Computation Conference 2010* GECCO10

[27] Eshelman L.J., Schaffer J.D. (1997) Preventing premature convergence in genetic algorithms by preventing incest", in *Foundations of Genetic Algorithms 4*. pp 115-122 Eds R.K.Belew, M.D.Vose - Morgan Kaufman, San Francisco 1997

[28] Fletcher D.A., McMeeking, R.F., Parkin D, (1996) "The United Kingdom Chemical Database Service", J. *Chem. Inf. Comput. Sci*. 1996, 36, 746-749.

Formalising Harmony Seeking Rules of Morphogenesis

Tim Hoverd and Susan Stepney

Department of Computer Science, University of York, UK, YO10 5DD
tim.hoverd@cs.york.ac.uk susan@cs.york.ac.uk

Abstract

The 15 generative patterns of Alexander's "Nature of Order" are descriptions of architectural structures that are seen in both buildings and in the natural world. We are investigating various aspects of complex systems, including those relating to structural patterns that may underlie those systems. Here we describe some experiments to generate 2D structures that incorporate those patterns that Alexander describes as *Positive Space*, the voids that contribute to the overall pattern, and *Levels of Scale*, a gradation in the size of the pattern's components. We show some of the results, illustrating that these patterns can be achieved as emergent properties of simple placement algorithms with a generative component.

Introduction

Studies of morphogenesis in ALife are typically inspired by biological growth and development process. However, there are other systems that grow and develop, influencing and influenced by their environment: buildings and towns. Here we investigate using these processes as an alternative source of inspiration.

Alexander's *Generative Patterns* (Alexander, 2004) are a vision of the way that successful architectural forms can be seen as the product of the generative application of a small number of properties that are seen in those forms. They attempt to describe the way that an architectural whole, be it a house or a city, evolves as a consequence of its environment and use. For example, (Alexander, 2004) shows a diagram of ancient Rome and discusses how that particular configuration emerged from the human use and development of the city.

We are examining these *Generative Patterns* to investigate the way that such approaches work. Our long term goal is to apply these properties, or similar ones, to the generative development of the architecture of *complex systems*: systems whose complex behaviour *emerges* from the simple behaviour of a large number of elements. But first it is necessary to explore Alexander's patterns in more detail, and to be able to synthesise structures that satisfy his criteria.

Here we discuss Alexander's patterns, and show the results of a computer program that uses a number of different algorithms which attempt to generate structures that match two of his generative patterns.

The Nature of Order

The four volumes of *The Nature of Order* (Alexander, 2004) explore the notion of *Wholeness* in relation to architectural structures. *Wholeness* is Alexander's enigmatic term for the "quality without a name" that he identified earlier in (Alexander, 1979). In *The Nature of Order*, Alexander identifies 15 *generative properties* as the root characteristics of those architectural structures that form a satisfactory *whole*.

Alexander describes structures in terms of *centres*, each of which is "a zone of coherence in space". A centre is a region that is in some way *coherent* in the way it represents the space and its use. By *"coherence"* Alexander means that a centre is distinct from those around it and within it, but that in some way it contributes to the coherence of those other centres. Alexander refers to these as "centres" as they are "centres of influence, centres of action, centres of other centres" (Alexander, 2004, vol.1, p108). One particular reason for using the word "centre" is that he is trying to describe things that may have no specific boundary; a pond, for example, might include the pipes bringing in water, the rocks on its edge (Alexander, 2004, vol.1, p84). A centre is something noticeable about a structure; something that draws attention from neighbouring structures. Examples might be (Appleton, 1997) a row of tiles on a ceiling or floor, a hallway, a pond in the countryside, and—in the context of software development—what are known as "patterns" (Gamma et al., 1995).

The generative properties are used to describe a structure as a system of centres, and to show the ways that that structure can be further elaborated and extended, or *generated*, as a region is architecturally developed. Alexander sees this as a generative, developmental process, where the system of centres is progressively developed using the same set of generative processes which each application of these processes being dependent on the current structure. For example, Alexander (Alexander, 2004, vol.2, pp252–255) describes how the structure of St Mark's Square in Venice can

be described as the current end product of an evolutionary process. At each step of this process, Alexander identifies *latent centres* and shows how, in his view, new building supported and strengthened these centres.

Generative Properties

The 15 properties are described in (Alexander, 2004, vol.1) as:

Levels of Scale *"how a centre is made stronger (more coherent) by the smaller strong centres within it and the larger strong centres that surround it."*

Positive Space *"the way that a given centre must draw its strength, in part, from the strength of other centres immediately adjacent to it in space."*

Roughness *"the way that the field effect of a given centre draws its strength, necessarily, from irregularities in the sizes, shapes and arrangements of other nearby centres"*

Alternating Repetition *"the way in which centres are strengthened when they repeat, by the insertion of other centres between the repeating ones"*

Thick Boundary *"the way in which the field-like effect of a centre is strengthened by the creation of a ring-like centre, made of smaller centres which surround and intensify the first. [It] also unites the centre with the centres beyond it, thus strengthening it further"*

Good shape *"the way that the strength of a given centre depends on its actual shape and the way this effect requires that even the shape, its boundary, and the space around it are made up on strong centres."*

Local Symmetry *"the way that the intensity of a given centre is increased by the extent to which other smaller centres that it contains are themselves arranged in locally symmetrical groups"*

Contrast *"the way that a centre is strengthened by the sharpness of the distinction between its character and the character of surrounding centres"*

Gradient *"the way in which a centre is strengthened by a global series of different-sized centres which then **point** to the new centre and intensify its field effect"*

Deep Interlock and Ambiguity *"the way in which the intensity of a given centre can be increased when it is attached to nearby strong centres, through a third set of strong centres that ambiguously belong to both"*

Echoes *"the way that the strength of a given centre depends on similarities of angle and orientation and systems of centres forming characteristic angles thus forming larger centres, among the centres it contains"*

Simplicity and Inner Calm *"the way the strength of a centre depends on its simplicity - on the process of reducing the **number** of different centres which exist in it, while increasing the **strength** of these centres to make them weigh more"*

The Void *"the way that the intensity of every centre depends on the existence of a still place - an empty centre - somewhere in its field"*

Not Separateness *"the way the life and strength of a centre depends on the extent to which that centre is merged smoothly - sometimes even indistinguishably - with the centres that form its surroundings"*

Strong Centre *"defines the way that a strong centre requires a special field-like effect, created by other centres, as the primary source of its strength"*

These 15 separate properties address the same thing: the manner in which centres interact to increase the overall coherence of the space. Our long term objective is to examine how these properties, or analogous ones, might apply in the context of the evolutionary development of *complex systems* architectures. We start by examining two of these properties in more detail: *Positive Space* and *Levels of Scale*.

Positive Space

"Positive Space" is conventionally used to describe *"space that is occupied by a filled shape or a positive form"* (Wong, 1993). The positive space is the figure at the centre of attention; it is the part of the figure that the eye sees. In this sense positive space is in contrast with the negative space that surrounds the positive; it is the "figure" not the "ground".

Alexander describes the space between the artefacts of a built environment as ideally being *Positive Space*. This is in contrast with the conventional use of the term *negative space* where an artist "relies on the space that surrounds the subject to provide shape and meaning" (Bar, 2009).

For Alexander, *Positive Space* is that space which, although the space between other parts of a structure, itself contributes towards the "wholeness". That is, if the structure represents a coherent whole, then the space between the built artefacts is itself (also) positive, in that it contributes to the overall coherence rather than just being the (negative) space between those artefacts. So the figure *and* the ground are both positive, in a coherent whole.

An extreme example of this is the Escher wood-cut "Day and Night" (Escher, 1938): the space between flying geese is yet more geese, heading in the opposite direction. That is, the "space" has its own positive structure. The same relationship appears in non-spatial examples, too. For example, Tsur shows how the same concepts occur in areas such as music and poetry (Tsur, 2000).

Levels of Scale

Centres, the structural components of the architectural space, are made more "coherent" by the presence of both larger and smaller centres in the overall structure. A particular architectural space is overall more coherent if the various structures, and indeed the non-structures that are the *Positive Space* display a degree of gradation in their sizes. For example, a large structure placed next to a collection of smaller structures might represent an overall structure that was more "whole".

If the changes in scale are too extreme the centres would not be seen as increasing each other's coherence. Alexander shows how coherent structures often contain a number of levels of scale in the ratio of about 3:1 (Alexander, 2004, vol.1). The same ratio appears elsewhere; Salingaros shows levels of scale in the centres of a carpet design which appear in the ratio 3:1 over eight levels of scale (Salingaros, 1995).

BlobWorld: Exploring the properties

We first examine the properties of *Positive Space* and *Levels of Scale*. We do this in a very simplified simulation, of "blobs" (round or square) being placed in a 2D environment of previously placed blobs.

Our *BlobWorld* application generates simple diagrams that have greater or lesser degrees of these properties, dependent on various parameter values and the particular algorithms used. These algorithms are designed in such a way that, are far as possible, aspects of the desired properties *emerge* as a result of the generative processes, rather than being explicitly encoded.

Contingent Placement Algorithm

The first algorithm, *contingent placement*, attempts to produce emergent *Positive Space*. It attempts to place a blob at a given position; if it is obstructed by existing blobs, the new blob is moved along a randomly chosen direction until it is no longer obstructed. So the placement is contingent on the presence of pre-existing blobs. The algorithm is given in figure 1, in which:

blobShape is "round" or "square".

sizePDF is the probability distribution function (pdf) used to generate blob sizes (see later).

visProb is the probability of a blob being visible. Early versions of BlobWorld did not have this parameter and blobs were always visible on the diagram. The addition of "invisible" blobs (which are not visible but nevertheless affect the placement of other blobs) has a significant effect on the appearance of *Positive Space* in the resulting diagrams.

blobCount is the total number of blobs (both visible and invisible).

```
 1: blob[0] := new Blob(blobShape)
 2: blob[0].setSize(sizePDF)
 3: blob[0].setVis(boolean according to visProb)
 4: blob[0].setPosition(origin)
 5: blob[0].draw()
 6: for i = 1..blobCount-1 do
 7:     blob[i] := new Blob(blobShape)
 8:     blob[i].setSize(sizePDF)
 9:     blob[i].setVis(boolean according to visProb)
10:     blob[i].setPosition{blobs[0].getPosition()
                | blobs[i-1].getPosition()
                | blobs[random(0..i-1)].getPosition()}
11:     blob[i].setDirection(rand in 0 ... 360°)
12:     while not blob[i].isOverlapAcceptable(
              allowedOverlap) do
13:         blob[i].movePositionAlongDirection()
14:     end while
15:     blob[i].draw()
16: end for
```

Figure 1: Pseudo-code for the *contingent placement* algorithm

allowedOverlap determines how much a blob is allowed to overlap other blobs: when positive, blobs may overlap by an amount determined by the magnitude of this parameter; when zero blobs just touch; when negative, blobs have a small amount, determined by the magnitude of the parameter, of clear space around them.

setPosition takes one of three arguments: the centre of the initial blob, or the most recently placed blob, or a random blob, to start off the current blob. (In this paper, the initial blob position is always used.)

Every run creates a unique pattern of blobs which is highly dependent on the various parameters. Although the algorithm is simple, with appropriate parameter choices it is capable of generating patterns that display a significant degree of *Positive Space*. Three examples of generated patterns are shown in figure 2.

In most cases where **vis** = 1, (that is, where all blobs are always visible) the generated patterns show no significant degree of *Positive Space* (for example, figure 2a where the space is nothing more than a lack of blobs; it is ordinary "negative space").

The algorithm is more successful at generating *Positive Space* when some blobs are invisible (for example, figure 2b). The invisible blobs generate additional space, which enables the appearance of *Positive Space*. Figure 2b shows the effect of the *Positive Space* : in the left of these pictures, the observer gets a powerful impression of the space itself constraining, for example, the curve of blobs at the lower right corner. In many of the diagrams generated in this manner, the *Positive Space* does not exactly align with the invis-

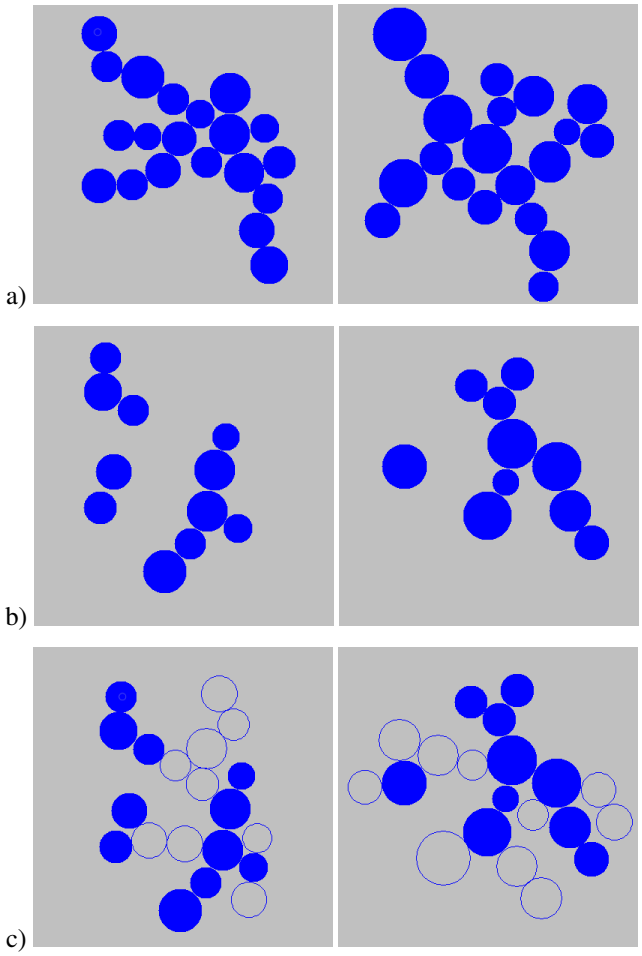


Figure 2: Results of the *contingent placement* algorithm with **blobCount** = 20, **sizePDF** = gaussian, **blobShape** = round, **allowedOverlap** = 0 : (a) **vis** = 1 ; (b) **vis** = 0.5 ; (c) as b, but with the position of the "invisible" blobs shown

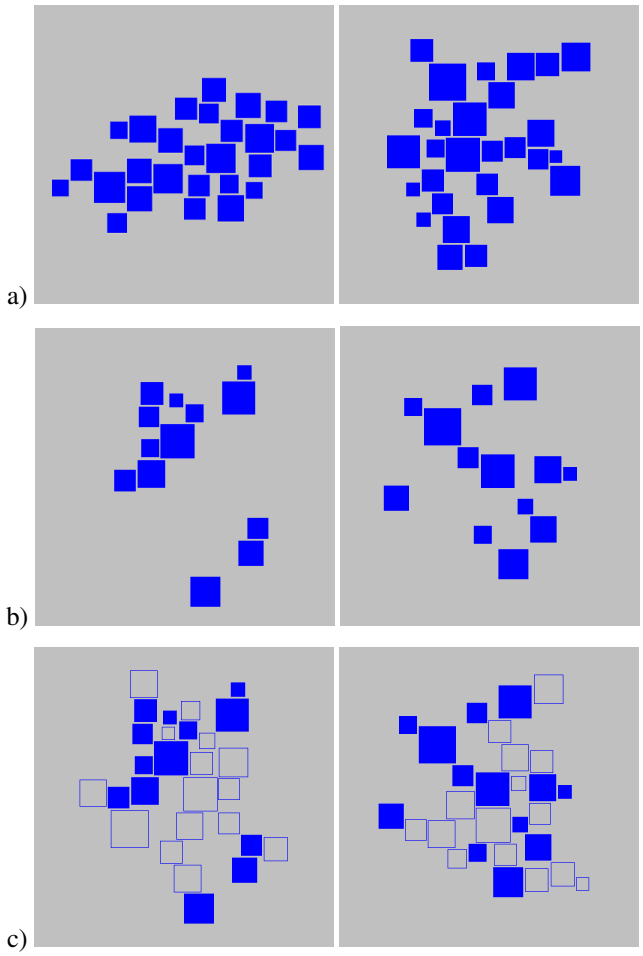


Figure 3: Results of the *contingent placement* algorithm with **blobCount** = 28, **sizePDF** = gaussian, **blobShape** = square, **allowedOverlap** < 0 : (a) **vis** = 1 ; (b) **vis** = 0.5 ; (c) as b, but with the position of the "invisible" blobs shown

ible blobs. That is, although the invisible blobs are in some way enabling the emergence of *Positive Space*, they are not themselves that space (figure 2c).

This successful generation of positive space is not dependent on using round blobs. The same effects are generated with square blobs (figure 3). Again, without the invisible blobs there is little sign of *Positive Space* (figure 3a), but when invisible blobs are introduced they create *Positive Space* (figure 3b).

With the square blobs, a further effect is visible. Here we have used a negative **allowedOverlap**, to separate the blobs from each other along their straight boundaries. Although the blobs are all perfectly aligned squares, an optical illusion makes some edges look slightly tilted or slightly bowed; this adds a degree of *Roughness* (another of Alexander's generative properties) to the picture.

Independent Placement Algorithm

In order to test whether *Positive Space* is manifested in any diagram that merely contains "invisible" blobs a second algorithm is also implemented by BlobWorld. This *independent placement* algorithm positions blobs not as a consequence of the positions of other blobs but as an initial step of the algorithm. In essence, the *contingent placement* algorithm positions blobs of a pre-determined size in a field of other blobs as the diagram evolves from a single blob. In contrast, the *independent placement* places blobs entirely independently of each other but then manipulates the *size* of all of the blobs until the diagram, as a whole, achieves the stated requirements for blob overlap.

The *independent placement* algorithm is described by the pseudo-code in figure 4 in which:

growthPDF is the pdf used to generate the growth rate of each blob (see later).

```
for i = 0..blobCount-1 do
   blob[i] := new Blob(blobShape)
   blob[i].setGrowthRate(growthPDF)
   blob[i].setSize(1)
   blob[i].setVis(boolean according to visProb)
   blob[i].setPosition(positionPDF)
   blob[i].unfreeze()
end for
while exists an unfrozen blob do
   for i = 0..blobCount-1 do
      if blob[i] is unfrozen then
         blob[i].setSize(
               blobs[i].getSize * blobs[i].getGrowthRate)
         blob[i].draw()
      end if
      if blob[i].overlapsOtherBlob(allowedOverlap) then
         blob[i].freeze()
      end if
   end for
end while
```

Figure 4: Pseudo-code for the *independent placement* algorithm

positionPDF is the pdf used to generate the initial position of each blob. (Here it is a uniform distribution across the drawing space.)

Examples of the *independent placement* algorithm are shown in figure 5. (One of the effects of the algorithm is that pairs of same-sized blobs occur often: if two nearby blobs have the same growth rate, they both grow at this same rate until they come into contact and become frozen.) Although the diagrams generated with this algorithm do contain space, it is not *Positive Space*. That is, space that is there does not contribute to the overall coherence of the pattern; essentially, it is merely a random collection of blobs of different sizes.

Positive Space appears in the results of the contingent placement algorithm only when the invisible blobs are allowed. However, invisible blobs do not result in *Positive Space* in the independent placement algorithm (figure 6). It is clear that the space does not have the same coherent influence as that seen in the results of the contingent placement algorithm.

The essential difference between the two algorithms is that the *contingent placement* algorithm places blobs in positions determined, to some extent, by the blobs that already exist. That is, it is essentially generative in nature. In contrast, the *independent placement* algorithm pre-determines the placement of the blobs. It naturally results in space within the pattern: the blobs cannot enlarge to fill the entire space given their fixed starting positions. But it does not generate *Positive Space*.

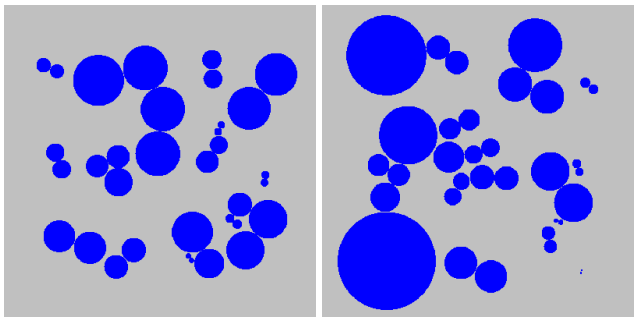


Figure 5: Typical results of the *independent placement* algorithm with **blobCount** = 34, **growthPDF** = gaussian, **blobShape** = round, **allowedOverlap** = 0 ; **vis** = 1

Levels of Scale Algorithm

With BlobWorld we can also start to explore the *Levels of Scale* property. As seen in the *placement* algorithms, the blob sizes are chosen according to a pdf; there a guassian (normal) distribution is used (with a user defined mean and standard deviation). This generates a range of sizes (figures 2, 3), resulting in some *Roughness*, but does not exhibit the 3:1 *Levels of Scale* property.

To investigate *Levels of Scale* we use bi-modal and tri-modal pdfs for size, where the mean (size) and occurrence likelihood (number) of blobs in the different modes have a fixed ratio of 3:1 (figure 7).

Figure 8 shows three blob figures generated using the bi-modal size distribution. The first and second examples show little evidence of the *Levels of Scale* property. The sizes follow the 3:1 distribution, but because that size has no effect on blob placement there is little evidence of any *coherence* in the size distributions spatially.

Our hypothesis is that to achieve the *Levels of Scale* property the various blob sizes would need to be arranged in such a way that changes in size are also, to some extent, reflected in their positions. Such an arrangement seldom appears in the context of either of the BlobWorld algorithms, as the blob sizes are either pre-determined, as in the contingent placement algorithm, or a consequence of the position of only the nearest other blob, as in the independent placement algorithm.

Occasionally, some degree of *Levels of Scale* is visible in BlobWorld patterns, for example in figure 8c in the two near-vertical "walls" at bottom centre, and in figure 9. This suggests that a small modification to the algorithm might well be capable of generated a suitable degree of *Levels of Scale*. This leads to our *generative size* algorithm.

Generative size algorithm

Experience with the *independent placement* and *contingent placement* algorithms shows that when blobs are positioned *generatively* then a diagram that demonstrates Alexander's

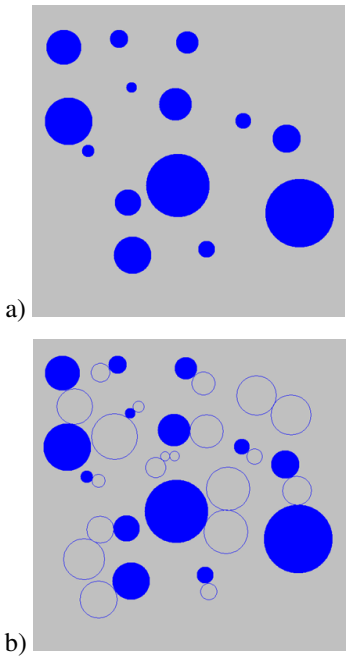


Figure 6: Typical results of the *independent placement* algorithm with **blobCount** = 34, **growthPDF** = gaussian, **blobShape** = round, **allowedOverlap** = 0 ; **vis** = 0.5 (a) invisible blobs not shown; (b) as a, but with the "invisible" blobs shown

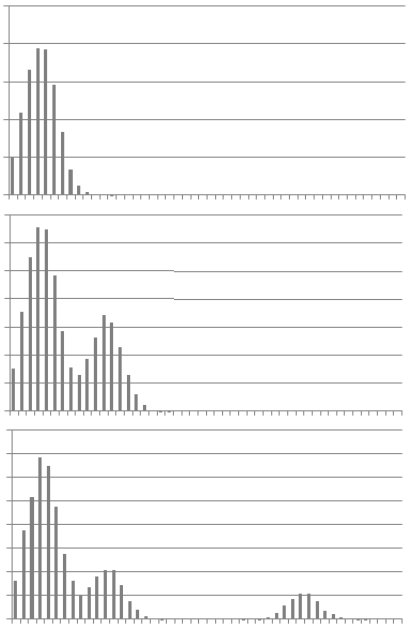


Figure 7: pdfs for investigating *Levels of Scale*. The x-axis is the blob size; the y-axis is the probability of that size: (a) single mode, gaussian distribution; (b) bi-modal, generating (approximately) three blobs of size 1 for every blob of size 3; (c) tri-modal generating (approximately) nine blobs of size 1 and three of size 3 for every blob of size 9

Positive Space property appears. That, when the diagram evolves from a small core in accordance then the result approximates a property that is observed in the end result of human-developed architecture.

However, the initial *contingent placement* algorithm is generative only with respect to the position of the blobs; their size is determined independently according to the pdfs discussed above.

A further algorithm exploits this observation by making both position and the size of the blobs the result of a generative process. It is essentially a simple modification of the *contingent placement* algorithm and the pseudo-code appears in figure 10 in which:

sizeRatio is the ratio is size between different "generations" of blob.

That is, as the algorithm is searching for a valid position for the blob it repetitively reduces the size of the blob in accordance with some predefined ratio. The effect of this is to make the size of each blob the result of a generative process which is influenced by the "environment" of each blob.

Results of executing this *generative size* algorithm are shown in figure 11. These diagrams are initially strongly reminscent of the diagrams Alexander shows as representative of the layout of cities and structures which are the result of long-term human development (Alexander, 2004): the blobs are positioned and sized in an generative manner that is a consequence of the positioning and sizing of pre-existing blobs as the diagram evolves. As can be seen from the diagrams in the figure the blobs are now showing evidence of the *Levels of Scale* property in that the blobs appear in a wide range of sizes but there are frequent clumps of similarly sized blobs.

Conclusions

The results of these initial BlobWorld experiments are encouraging. Our *contingent placement* algorithm is capable of generating diagrams that exhibit the *Positive Space* property. That the alternative *indepenedent placement* algorithm does not have this capability indicates that the effects observed are more than mere chance.

It is likely that this capability of the *contingent placement* algorithm is due to the combination of two aspects. Firstly, the invisible blobs generate spaces that do indeed have a positive aspect, in that they contain blobs; the space is more than mere empty space, there is actually something there: (invisible!) blobs. Secondly, the algorithm is to some degree generative, in that blobs are placed in positions that are strongly conditioned by the position of existing blobs. That is, the pattern does in fact *grow* towards its final configuration.

Conversely, the *independent placement* algorithm does it-

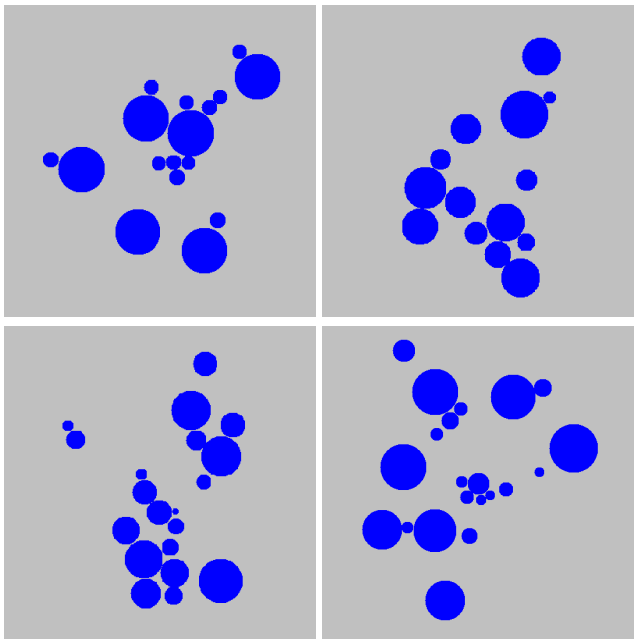


Figure 8: Attempts to generate *Levels of Scale*: contingent placement algorithm, bi-modal size distribution with a small standard deviation, **vis** = 0.5.

self naturally generate spaces. However, those spaces do not jostle directly against the blobs; the blobs jostle against each other. That is, the space is not *positive*, it is merely empty (negative) space.

Our attempts at generating the *Levels of Scale* property are also successful. The initial, somewhat explicit, attempt does not succeed in generating this property. However, the less explicit *generative size* algorithm shows that when blob size is made a direct consequence of the underlying generative process (that is when the size is a consequence of the evolution of the diagram) then the *Levels of Scale* property appears naturally in the resulting diagrams.

There is, therefore, a complex interaction of size and position taking place as the diagram evolves. Futher work is needed to establish the details of this interaction.

Future Work

This is the first step in a programme looking at Alexander's 15 generative properties. It is sufficiently successful to indicate immediately some further work, in particular on a generative algorithm that influences other properties. We have already remarked that a degree of *roughness* has emerged in the diagrams, as a consquence of optical effects and the inevitable quantisation of size and position due to the current algorithms.

What is obviously missing from the current work is some element of *measurement*. In particular, just because some diagrams appear to us to be more "whole" does not mean that

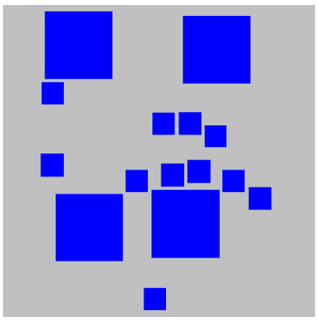


Figure 9: Attempt to generate *Levels of Scale* occasionally work: contingent placement algorithm, bi-modal size distribution, **vis** = 0.5.

```
blob[0] := new Blob(blobShape)
blob[0].setSize(sizePDF)
blob[0].setVis(boolean according to visProb)
blob[0].setPosition(origin)
blob[0].draw()
for i = 1..blobCount-1 do
   blob[i] := new Blob(blobShape)
   blob[i].setSize(sizePDF)
   blob[i].setVis(boolean according to visProb)
   blob[i].setPosition{blob[0].getPosition()
         | blob[i-1].getPosition()
         | blob[random(0..i-1)]. getPosition()}
   blob[i].setDirection(rand in 0 . . . 360°)
   while not blob[i].isOverlapAcceptable(
         allowedOverlap) do
      blob[i].movePositionAlongDirection()
      blob[i].reduceSize(sizeRatio)
   end while
   blob[i].draw()
end for
```

Figure 10: Pseudo-code for the *generative size* algorithm

that is objectively true. The *Nature of Order* includes some work, in particular the "bead game" (Gabriel, 1996), that shows that some aspects of the perception of "wholeness" are universal. We will address this by means of a scoring exercise in which a number of subjects will attempt to mark different blob patterns. We will compare these scores with the parameters used to generate the patterns.

What is at the moment more speculative, though, is the relevance this work could have for that of *complex systems* architectures. For example, if *Positive Space* is a particularly advantageous aspect of building structures, what does that imply for the complex systems that are the end target of this work? We will start with flocking behaviour models (Reynolds, 1987; Andrews et al., 2008), and draw an analogy between blobs and boids: for example, how might the presence of "invisible" boids affect the observed emergent

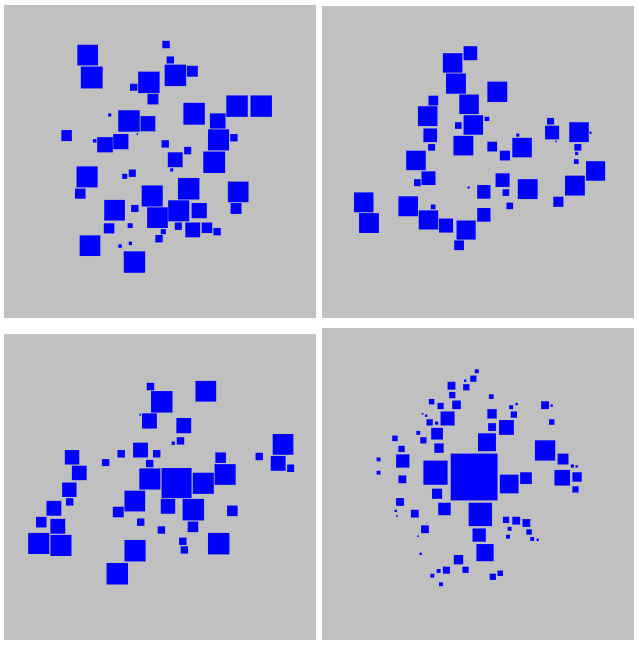


Figure 11: Typical results of the *generative size* algorithm with **blobCount** = 106, **sizePDF** = gaussian, **blobShape** = square, **allowedOverlap** = -3, **vis** = 0.5, **sizeRatio** = 1.4

flocking behaviour?

Additionally, the *Levels of Scale* property requires some form of inhomogeneous agents.

Positive Space indicates that the environment can play an imporant role in the development of the structure (recall that although "invisible blobs" are required to form *Positive Space* in our system, they are not coincident with it). This has led us to investigating the role of the environment in complex systems simulation, including taking an "environment-oriented" approach (Hoverd and Stepney, 2009) to modelling and implementation.

Alexander's properties are rooted in the consideration of structures in physical space. Design patterns (Gamma et al., 1995) are structures that exist in an abstract design space. The emergent properties of a complex system are structures that exist in the execution space of that system, or at least of a simulation of that system. We are investigating the extent to which the ideas explored in the Nature of Order might apply to these non-physical spaces.

Acknowledgments

We would like to thank Chris Alexander for many interesting and helpful discussions.

The work described here is part of the CoSMoS[1] project, funded by EPSRC grant EP/E053505/1 and a Microsoft Research Europe PhD studentship.

[1] http://www.cosmos-research.org.

References

Alexander, C. (1979). *The Timeless Way of Building*. Oxford University Press.

Alexander, C. (2004). *The Nature of Order, Volumes 1–4*. Center for Environmental Structure.

Andrews, P., Sampson, A., Bjørndalen, J., Stepney, S., Timmis, J., Warren, D., and Welch, P. (2008). Investigating patterns for the process-oriented modelling and simulation of space in complex systems. In *ALife XI*, pages 17–24. MIT Press.

Appleton, B. (1997). On the nature of the nature of order. http://www.cmcrossroads.com/bradapp/docs/NoNoO.html, last accessed 5 June 2010.

Bar, N. (2009). *Negative Space*. Mark Batty Publisher.

Escher, M. C. (1938). Day and Night. http://www.worldofescher.com/gallery/A11.html, last accessed 22 March 2010.

Gabriel, R. P. (1996). *Patterns of software: tales from the software community*. Oxford University Press.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Longman Publishing Co., Inc.

Hoverd, T. and Stepney, S. (2009). Environment orientation: an architecture for simulating complex systems. In *Proceedings of the 2009 Workshop on Complex Systems Modelling and Simulation*, pages 67–82. Luniver Press.

Reynolds, C. W. (1987). Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics*, 21(4):25–34.

Salingaros, N. (1995). In defense of Alexander. *HALI: The International Magazine of Antique Carpet and Textile Art*, 78:67–69.

Tsur, R. (2000). Metaphor and figure-ground relationship: Comparisons from poetry, music, and the visual arts. *PSYART: A Hyperlink Journal for the Psychological Study of the Arts*, (000201). http://www.clas.ufl.edu/ipsa/journal/2000_tsur03.shtml.

Wong, W. (1993). *Principles of Form and Design*. John Wiley and Sons.

Middle-out Modeling of Multiscale Morphodynamics

Walter de Back and Jörn Starruß

Center for High Performance Computing, Technische Universität Dresden, Germany
walter.deback@tu-dresden.de, joern.starruss@tu-dresden.de

Extended Abstract

How a fertilized egg develops into a multicellular organism remains one of the most challenging questions in biology. Novel techniques provides unprecedented high-resolution data on the spatiotemporal dynamics of the developing embryo. However, interpretation of these data requires both wet lab experiments and computational modeling (Oates et al., 2009). Here, we present a new modeling environment that is based on the following principles: Developmental systems (i) are multiscale systems, (ii) are morphodynamic, and (iii) require a middle-out modeling approach.

(i) Embryogenesis unfolds as a dynamic interplay of gene regulation, cellular signaling, differentiation, proliferation, and tissue mechanics. Developmental processes are coupled over multiple spatial and temporal scales and across structural levels. Understanding developmental processes implies unraveling how these scales are coupled.

(ii) Two main components of development can be distinguished: (a) induction, change of cell state and (b) morphogenesis, change in spatial distribution of cells. Although typically modeled as distinct processes, these mechanisms in fact occur concurrently and are causally interdependent (Salazar-Ciudad et al., 2003). Such 'morphodynamic' mechanisms enable a rich variety of tissues and provide correction mechanisms and robustness.

(iii) Restraining complexity in models of multiscale morphodynamics is essential to gain explanatory potential. Bottom-up approaches (from molecular kinetics pathways up) and top-down approaches (from tissue biophysics down), run into difficulties when attempting to encompass all relevant scales. The alternative is a middle-out strategy in which the cell is taken as a basic unit of modeling and only those molecular and tissue-level processes are included that are relevant to the phenomenon under investigation (Noble, 2002).

Similar to the popular CompuCell3D package (Cickovski et al., 2007), our modeling environment uses the well-known cellular Potts model (Glazier and Graner, 1993), reaction-diffusion solvers, a flexible plug-in architecture and an easy-to-use model description language. Several subtle yet crucial differences render our software pre-eminently suitable to model multiscale morphodynamics. Most prominently, the symbolic nature of description language enables the modeler to symbolically link all processes over spatiotemporal scales and structural levels without programming. This makes systematic exploration possible of the effects of multiscale and morphodynamic coupling. We demonstrate the conceptual and computational framework in the context of pattern formation models on neurogenic differentiation.

References

A. C. Oates et al. (2009). Quantitative approaches in developmental biology. *Nat Rev Genet*, 10(8): 517–530.

D. Noble (2002). Modeling the heart – from genes to cells to the whole organ. *Science*, 295(5560): 1678–1682.

I. Salazar-Ciudad et al. (2003). Mechanisms of pattern formation in development and evolution. *Development*, 130(10): 2027–2037.

J. Glazier and F. Graner (1993). Simulation of the differential adhesion driven rearrangement of biological cells. *Phys Rev E Stat Phys*, 47(3): 2128–2154.

T. Cickovski et al. (2007). From genes to organisms via the cell: a problem-solving environment for multicellular development. *Comp in Science and Eng*, 9: 50–60.

Accommodating Homeostatically Stable Dynamical Regimes to Cope with Different Environmental Conditions

Bruno A. Santos[1,2], Phil Husbands[1] and Tom Froese[1]

[1]Centre for Computational Neuroscience and Robotics, University of Sussex, Brighton, U.K.
[2]Computer Engineering Department, CEFET-MG, Belo Horizonte, Brazil.
brandre@gmail.com

Abstract

Does the dynamical regime in which a system engages when it is coping with a situation *A* change after adaptation to a new situation *B*? Is homeostatic instability a generic mechanism for flexible switching between dynamical regimes? We develop a model to approach these questions where a simulated agent that is stable and performing phototaxis has its vision field inverted so that it becomes unstable; instability activates synaptic plasticity changing the agent's simulated nervous system attractor landscape towards a configuration that accommodates stable dynamics under normal and inverted vision. Our results show that: 1) the dynamical regime in which the agent engages under normal vision changes after adaptation to inverted vision; 2) homeostatic instability is not necessary for switching between dynamical regimes. Additionally, during the dynamical system analyses we also show that: 3) qualitatively similar behaviours (phototaxis) can be generated by different dynamics; 4) the agent's simulated nervous system operates in transient dynamic towards an attractor that continuously move on the phase space; and 5) plasticity moves and reshapes the attractor landscape in order to accommodate a stable dynamical regimes to deal with inverted vision.

Introduction

The concept of homeostasis coined by Cannon (1932) refers to a condition in which coordinated physiological processes maintain certain variables within limits. Though this concept was introduced by Cannon, earlier work by Bernard (1927) had already identified regulatory systems in the organism's internal environment (*milieu interieur*). From these pioneering works, research in animal physiology studied homeostatic mechanisms controlling body temperature, heart rate, levels of blood sugar, breathing rate and others (see Cooper (2008) for a historical review). Recently, Turrigiano et al. (1998) observed that neurons also have a mechanism of homeostatic regulation which increases or decreases the strength of their synaptic inputs ensuring the maintenance of their firing rates within boundaries. She has also reported the presence of homeostatic regulations of activity in cortical networks (Turrigiano, 1999; Turrigiano and Nelson, 2004).

Rather than working directly with physiology, Ashby (1947, 1960) focused on more abstract dynamical system models of homeostasis in the context of adaptive behaviour. According to him, an animal behaviour is adaptive if it maintains essential variables within physiological limits. These variables are closely related to survival; they can be lethal (*e.g.* amount of oxygen in the blood), or only represent some approaching threat (*e.g.* heat on the skin). When essential variables cross certain boundaries a mechanism that changes the system configuration is activated until these variables return to homeostatic stable regions. The mechanism that pushes the variables back to their viable regions selects those configurations that not only recover stability at the current moment, but also leave the system stable in the presence of environmental conditions to which the system has previously adapted.

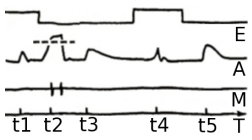


Figure 1: See text.
Adapted: Ashby (1960) p.116.

To illustrate the operation of this mechanism, consider an animal (A) interacting with its environment (E) (Fig. 1 represents the dynamic of A and E over time (T)). When the environment changes (at t2) the animal's dynamic becomes homeostatically unstable (the homeostatic boundary is represented by the dashed line). Due to instabilities the mechanism that changes the animal's organization is activated (downstrokes at M). The new organization found by M leaves the animal stable in the presence of both environmental conditions, as it is shown by the animal's dynamic (A) at t4 and t5.

Ashby also postulated that different environmental conditions can move the state of the system to different regions in phase space and at each region the system can have different dynamics. This is roughly illustrated by different dynamical regimes presented by the animal at t4 and t5. Summing up Ashby's main points in the context of our work, we can say that: an adaptive system interacting with its environment switches and engages in different dynamical regimes; when homeostatic instability increases the system reconfigures itself so that it: 1) accommodates a stable dynamical regime

that deals with the condition that triggered instability; and 2) maintains the stability of pre-existing dynamical regimes that deal with conditions previously adapted.

The homeostatic characteristics of a system do not impose constraints on the dynamics inside stable regions. As long as the state of the system is inside a homeostatic region, the system can be in an attractor or moving on a transient; it can also be monostable, bistable, multistable, or even without attractors inside stable regions. Thus, at the same time two types of stability can be measured in a system: homeostatic stability and Lyapunov stability[1].

Both stabilities are illustrated in Fig. 2. The axes X_1 and X_2 represent two generic variables; the dashed line is the homeostatic stable region; P_1 and P_2 are point attractors; continuous line around P_1 and P_2 define two regions on the phase space. On the border between these regions the system is Lyapunov unstable; outside the dashed line the system is homeostatically unstable. The point P_3 is homeostatically stable and Lyapunov unstable. Both types of stability are important to studying mechanisms of behavioural adaptation, but in this paper we focus exclusively on homeostatic stability.

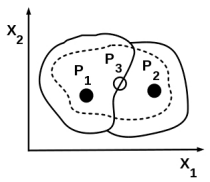


Figure 2: See text.

Given this brief introduction about homeostatic stability and adaptation, we present the questions we are tackling in this paper.

- Q1: Does the dynamical regime in which the system engages when it is coping with a situation *A* change after adaptation to a new situation *B*?

Using the illustration presented in Fig. 1 we can restate this question as: does the dynamical regime in which the animal engages when it is coping with the environmental condition presented at t1 change after adaptation to the new environmental condition presented at t2? We want to know the difference between the dynamics at t1 and t4, as the system has reorganized itself in order to accommodate a new stable dynamical regime to cope with the environmental condition presented at t2.

While the previous questions concerns the mechanism for adaptation, the second one approaches the mechanism for switching between dynamical regimes after adaptation.

- Q2: After adaptation, is homeostatic instability a generic mechanism for flexible switching between dynamical regimes?

Using the illustration presented in Fig. 1 we can restate this question as: is homeostatic instability a generic mechanism for flexible switching between dynamical regimes in which the animal engages at t4 and t5?

In order to approach these questions we develop a computational model based on a related model implemented by Di Paolo (2000). In his model, Di Paolo minimally replicated a psychological experiment carried out by Taylor (1962) where a human being adapts his behaviour to continuously wearing spectacles that distorts his vision field. Di Paolo replicated this experiment using an evolved simulated agent that performs phototaxis. During the agent's lifetime, he inverted the agent's vision field (switching right and left sensors) and studied the process of behavioural adaptation. The agent's mechanism of adaptation was implemented using homeostatic stability and synaptic plasticity[2].

Following Di Paolo we implement an agent performing phototaxis using homeostatic stability and synaptic plasticity. However, we replicate another experiment carried out by Taylor where a subject adapts his behaviour to intermittently (rather than continuously) wearing spectacles that distorts his vision field. Besides, in our model the inversion of the agent's vision field is done both during its lifetime and during evolution. Thus, our agent is evolved to adapt during its lifetime to inverted vision, differing from Di Paolo's agent which was evolved exclusively to perform phototaxis under normal vision.

The methodology to develop our computational model is based on four assumptions. The first three assumptions are grounded in Ashby's theory in the context of Turrigiano's empirical findings on homeostasis in neuronal networks, they are: 1) an agent behaviour is adaptive if it maintains its simulated neuronal network homeostatically stable; 2) changes in synapse strengths is a mechanism to recover homeostatic stability; and 3) a system conserves its condition of being adapted when synapse strengths are adjusted in such a way that homeostatic stability of neuronal networks is maintained in the presence of similar conditions that triggered instability in the past. The fourth assumption, which is supported by Ashby and Taylor[3], is that: 4) conditions to which the system is not adapted trigger homeostatic instability, that is, switching visual sensors triggers homeostatic instability in a not-yet-adapted simulated nervous system.

Details of the methodology are presented on the next section, followed by the Results where we study the dynamic of the system and show that: 1) the dynamical regime in which the agent engages under normal vision changes after adaptation to inverted vision; 2) homeostatic instability is not necessary for switching between dynamical regimes. Additionally, during the dynamical system analyses we also show that: 3) qualitatively similar behaviours (phototaxis)

[1] A fixed point x* is Lyapunov stable if all trajectories that start sufficiently close to x* remain close to it for all time. For a formal definition of Lyapunov stability see Strogatz (2000) p.141.

[2] For a theoretical discussion of Di Paolo's model see Di Paolo (2003).

[3] Taylor, in his experiment, uses Ashby's theory to explain the operation of the mechanism underlying the adaptive behaviour presented by the subject wearing distorted spectacles.

can be generated by different dynamics; 4) the agent's simulated nervous system operates in transient dynamic towards an attractor that continuously moves in the phase space; 5) plasticity moves and reshapes the attractor landscape in order to accommodate a stable dynamical regimes to deal with inverted vision.

Methods

This methodology follows, as much as possible, that one carried out by (Di Paolo, 2000). The main differences lie in the number of nodes used to implement the controller and in the evolutionary setup.

A genetic algorithm is used to evolve the parameters of our model. The range of each parameter, which defines the search space, is presented throughout the methodology together with the description of each variable.

Task. The task involves an agent that moves in a simulated environment and has to perform phototaxis on a sequence of light presentations (one by one) for 15000 secs. During its lifetime, the agent's right and left sensors are switched every 250 secs. The light is repositioned between 40 and 80 units away from the agent when either the sensors are switched or the agent spends 50 consecutive seconds close light (at a distance smaller than 10 unit).

Agent. The agent (Fig. 3) has a circular body of 8 units diameter, two diametrically opposed motors that receive a continuous signal in the range [-1,1] from the controller nodes (y_2 and y_3, respectively), and two light sensors separated by $120^\circ \pm 10^\circ$ whose output signal is given by $I_k = 1/\sqrt{d_k}$, where *k* represents each sensor, *d* is the distance from sensor *k* to the light source. $I_k = 0$ when the agent's body occludes the light and $I_k = 1$ if $d < 1$.

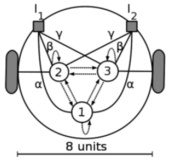


Figure 3: Agent.

Plastic controller. The agent's behaviour is controlled by a fully-connected, 3 nodes, continuous-time recurrent neural network (Eq. 1) (Beer, 1995).

$$\begin{aligned} \tau_i \dot{y}_i &= -y_i + \sum_{j=1}^{N} w_{ji} z_j + \sum_{k=1}^{M} s_{ki} I_k, \\ z_i &= \tfrac{1}{1+e^{-(y_i+b_i)}} \end{aligned} \tag{1}$$

where y is the state of each node which is integrated with time step of 0.1 using the Euler method , τ is its time constant (range [0.4,4], *N* is the number of CTRNN nodes (here 3); $w_{j,i}$ is the connection strength from the j^{th} to i^{th} node (range [-8,8]), z_j is the node output signal defined by a sigmoid function, b_j is a bias (range [-3,3]), *M* is the number of inputs (here 2); I_k is the sensory output signal, and s_{ki} is a constant that represents the sensory strength from the k^{ih} sensor to i^{th} node. The values for s_{ki} are: $s_{11} = s_{21} = \alpha$; $s_{12} = s_{23} = \beta$; $s_{13} = s_{22} = \gamma$, where α, β and γ are in the range [0.01,10] (see Fig. 3). Each connection between nodes ($w_{j,i}$) is adjusted by one out of four different homeostatic plastic rules (2). The rule used by each connection is defined by the genetic algorithm.

$$\begin{aligned} &R0 : \Delta w_{ji} = \delta\ \eta_{ji}\ p_i\ z_j\ z_i, \\ &R1 : \Delta w_{ji} = \delta\ \eta_{ji}\ p_i\ (z_j - z^o_{ji}) z_i, \\ &R2 : \Delta w_{ji} = \delta\ \eta_{ji}\ p_i\ (z_i - z^o_{ji}) z_j, \\ &R3 : \Delta w_{ji} = 0, \end{aligned} \tag{2}$$

where Δw_{ji} is the change in w_{ji}, δ is a linear damping function that constrains the weights between allowed values ([-8,8]), η_{ji} is the rate of change (range [-0.9,0.9], and p_i is the plastic facilitation defined by the function shown in the Fig. 4. Rule 0 is the Hebbian and anti-Hebbian rules (depending on p_i and n_{ji}); rules 1 and 2 potentiate or depress the connection depending on how presynaptic or postsynaptic node activity relates to a threshold z^o_{ji}. This threshold linearly depends on w_{ji} ($z^o_{ji} = 0$ if w_{ji}=-8 and $z^o_{ji} = 1$ if w_{ji}=8).

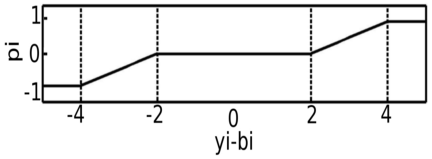


Figure 4: Local plasticity facilitation p_i. When the node activation minus its bias ($y_i - b_i$) is in the stable region ($[-2, 2]$) plasticity is not activated as $p_i = 0$. Out of this region p_i changes either positively or negatively according to the function.

Evolutionary setup. A total of 36 network parameters encoded in a genotype as a vector of real numbers in the range [0,1] are evolved using the microbial genetic algorithm (Harvey, 2001) and linearly scaled, at each trial, to their corresponding range. The genetic algorithm is setup as follows: population size (100); mutation rate (0.05); recombination (0.60); reflexive mutation; normal distribution for mutation $\left(\mu = 0, \sigma^2 = 0.1\right)$; and trials for each agent (8). At the end of the 8^{th} trial the worst fitness (out of 8) is used as the fitness of the agent.

The agent's lifetime is 15000 seconds and its sensors are inverted every 250 seconds. In total, sensors are inverted 60 times, where 30 times the agent is under normal vision and 30 under inverted vision. At each timeslot (250 secs) the fitness of the agent is measured according to Eq. 3:

$$F_t = \frac{F_b + F_s}{2} \tag{3}$$

where *t* is the timeslot (out of 60), F_b is the behavioural-fitness (Eq. 4) and F_s is the stability-fitness (Eq. 5).

$$F_b = \left(P + \left(1 - \frac{d_f}{d_i} \right) \right) \frac{R}{T} \tag{4}$$

where d_i and d_f are initial and final distances to the light source, respectively, and d_f is clipped at 0 when $d_f > d_i$; *P* is the number of times the agent approaches the light in the current timeslot (the agent can approach the light more than once as the light moves when the agent spends 50 seconds near it); *T* is the timeslot length (250 secs) and *R* (250 secs) is the required time given to the agent to approach a light source. During evolution, as *T=R* the agent should approach the light at least once in order to obtain $F_b = 1$. When the agent approaches the light more than the number of times required, F_b is clipped at 1.

$$F_s = \frac{1}{1 + e^{\left(\frac{u}{70} - 7\right)}} \tag{5}$$

where *u* is the number of times the nodes activate out of the stable region (at each Euler step, it can be incremented by 3 when the three nodes activate out of the stable region); the constants 70 and 7 define the shape of the function.

The total fitness of the agent is given by the weighted mean of the fitness at each timeslot.

$$F = \frac{1}{3K} \sum_{t=1}^{K} q_t;\ q_t = \begin{cases} F_t; & if\ v = 1 \wedge \forall\, t \\ 2(1 - F_t); & if\ v = -1 \wedge t \leq 30 \\ 2F_t; & if\ v = -1 \wedge t > 30 \end{cases} \tag{6}$$

where *K* is the number of timeslots (out of 60); F_t is defined in Eq. 3; *v* is the vision state (1 normal, -1 inverted). Under normal vision (*v*=1) the agent should get high fitness (F_t) during its whole lifetime ($\forall\, t$). Under inverted vision the agent should have low fitness (F_t) during the first 30 inversions ($t \leq 30$) and high fitness during the last 30. Hence: 1) the agent should perform phototaxis maintaining homeostatic stability under normal vision during the whole trial (30 timeslots); 2) the agent should be homeostatically unstable and not perform phototaxis when its vision field is inverted; and 3) over time, after a sequence of vision inversions (normal $\rightarrow$ inverted $\rightarrow$ normal $\rightarrow$ inverted, and so on), the agent should maintain stability and perform phototaxis under inverted vision (the last 30 timeslots).

After evolution the best agent of the population was selected and run 10000 in order to generate statistical measurements. The agent's lifetime was changed to 30000 secs and after 15000 secs of its lifetime its sensors were switched at a different frequency (as shown in Fig. 5-D).

Attractor landscape. In order to find the attractors of the controller while the agent is interacting with its environment, a snapshot of the system is taken at each Euler step of the agent's lifetime and the limit $\lim_{t\to\infty} \langle y_1(t), y_2(t), y_3(t) \rangle$ is numerically estimated. This snapshot consists of states of each CTRNN node (y_1,y_2,y_3), which are the initial conditions to find the limit; connection weights (w_{ji}); inputs (I_1 and I_2), which are maintained fixed during the numerical estimation; sensor strengths (s_{ki}), biases (b_i); and time constants (τ_i).

The limit is found using Euler integration with time step 0.1 and 900000 steps. When the system does not converge to a point attractor, the Euler integration runs for a further 100000 steps in order to capture at least some points of either the limit cycle or the strange attractor the system is assumed to be following.

Results

Evolution. The mean fitness of the population after evolution is 0.77 and the fitness of the best agent is 0.86. In Fig. 5-A and B (see caption) we present how the behavioural-fitness and stability-fitness of the best agent change during its lifetime.

Under normal vision the behavioural-fitness and the stability-fitness are maintained near 1 over the whole simulation. At the beginning of the agent's lifetime and under normal vision, the number of unstable activations is near 200. Despite these unstable activations the stability-fitness is still high due to the shape of the function defined in (5). Under inverted vision, the behavioural-fitness starts near 0 and linearly increases during the first 10000 secs; while the stability-fitness increases mainly between 5000 secs. and 10000 secs. These fitnesses increase at a different rate because while the activations of the nodes move towards the stable region, the behavioural-fitness increases; on the other hand, the stability-fitness only increases when the activations actually cross the boundaries (range [-2,2]), which starts after 5000 secs.

Behaviour. The distances from the agent to the light source before and after adaptation are presented in Fig. 6-A and B, respectively. After the first inversion (Fig. 6-A, $t = 251$ secs) the agent keeps turning around itself and only slightly moves towards the light until its sensors are switched back to the normal position (t=500 secs). After adaptation the agent approaches the light under both conditions.

Dynamics. The dynamical patterns in which the agent engages are represented in 6 dimensions (S1, S2: sensors; M1-M2: motors; y1, y2, y3: CTRNN nodes) by each pair of graphs in Fig 7 (see figure caption). From now on the dynamics of the CTRNN nodes presented in Fig 7-*A*, *B*, *C* and *D* will be referred as ρ_1, ρ_2, ρ_3 and ρ_4, respectively.

At the beginning of its lifetime, the agent engages in a homeostatic stable dynamical pattern (ρ_1) while performing phototaxis. Just after the first inversion the agent switches

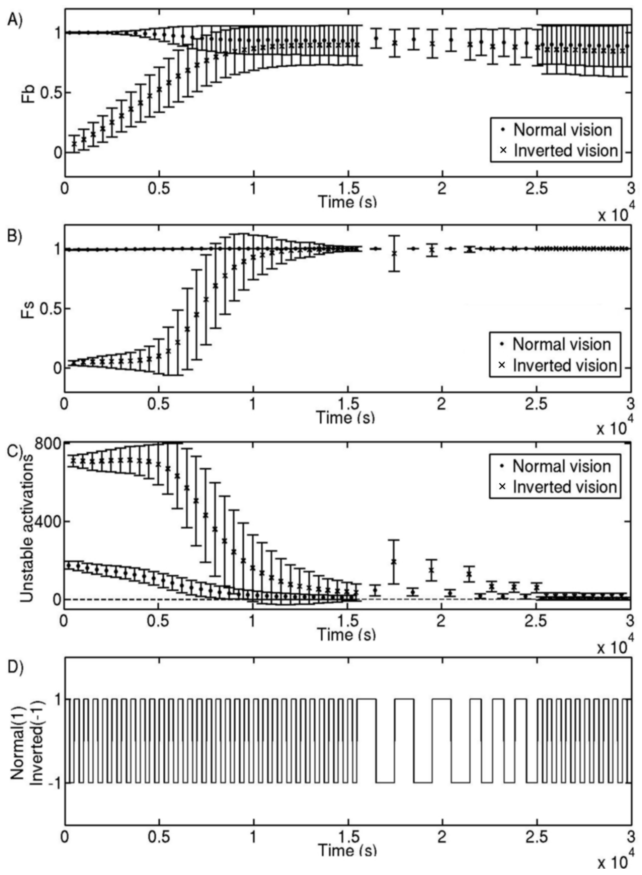


Figure 5: **A** and **B** show how behavioural-fitness and stability-fitness change over the agent's lifetime. Each point in those graphics represents the fitness for a specific timeslot. **C** depicts the number of node activations out of the stable region. **D** depicts the frequency of sensor switchings. These plots were generated running the best agent over 10000 trials. The vertical bars represent the standard deviation.

to the unstable ρ_2. After a sequence of inversions and plastic changes, the dynamical pattern instability under inverted vision decreases and changes from the unstable ρ_2 to the stable ρ_4. While instability under inverted vision decreases, the stability under normal vision is maintained (as shown in Fig 5-B); however, even while maintaining stability the dynamics under normal vision qualitatively changes from ρ_1 to ρ_3 as a side effect of adaptation to inverted vision.

While plasticity is activated during adaptation to inverted vision (from *t=250* to *t= 15000(s)*), the dynamical patterns under normal vision smoothly change from ρ_1 to ρ_3. In between these patterns there are other slightly different dynamical patterns and all of them generate phototatic behaviour (as shown by the behavioural-fitness - Fig 5-A). Besides the dynamical patterns under normal vision, ρ_4 under inverted vision also generates phototaxis. This shows that qualitatively the same behaviour can be generated by different dynamics.

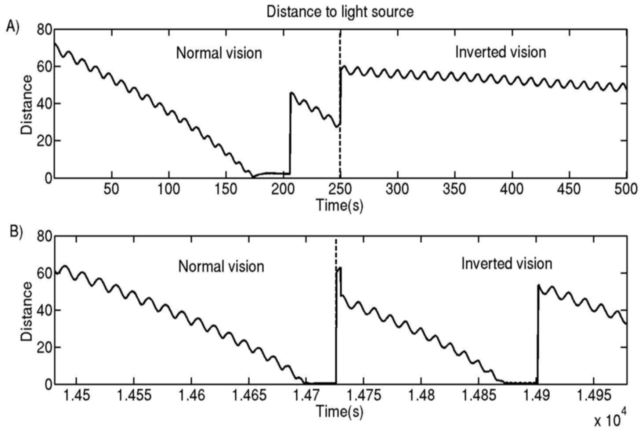


Figure 6: Distance from the agent to the light source before and after adaptation (**A** and **B** , respectively).

The dynamical patterns in which the agent engages are generated by an attractor that continuously moves in the phase space. This continuous movement of the attractor leaves the agent in a transient state while interacting with its environment (see Fig. 8). The transient dynamic is obtained because different sensor values define different set of parameters for the CTRNN equations which in turn gives different point attractors at each iteration. In other words, the agent's behaviour (movement in the environment) changes its sensor values which in turn moves the attractor in the phase space. The direction to which the attractor pulls the system generates new motor outputs that change the agent's position and consequently its sensor values. The resulting dynamical patterns involving the controller, body and environment generate the coordinated movement of the agent towards the light source.

While sensors values are changing and the rate of plastic changes is low, that is, when the agent is engaged in a stable dynamical pattern while interacting with its environment, the point attractor moves on a fixed 3D surface. At the beginning of the agent's lifetime this surface resembles a rectangle with attractors lying on its corners (see Fig.9 - gray dots). After adaptation, this surface moves to a different position and is reshaped (see Fig.9 - black dots). This new position and shape of the attractor landscape accommodates the stable dynamical patterns under normal and inverted vision, that is, both dynamical patterns ρ_3 and ρ_4 are generated by the same attractor landscape.

A quantitative difference between surfaces of attractors for each dynamical pattern (ρ_1, ρ_2, ρ_3 and ρ_4) is shown by the positions of clusters of attractors[4] (see Fig. 10-A1, B1,

[4]We used the K-means method (MacQueen, 1967) to identify clusters of attractors and their centroids.

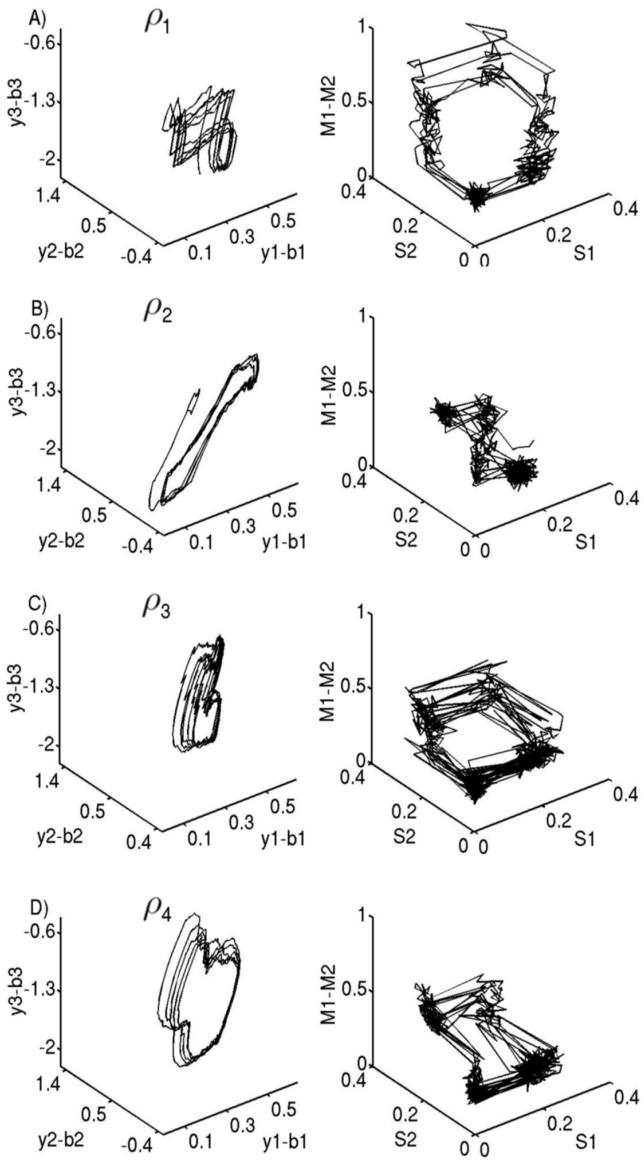


Figure 7: **A)** Stable dynamical pattern under normal vision before the first inversion; time: *85.7* to *137.0* secs; initial distance *di= 40.17*; final distance *df= 20.2*. **B)** Unstable dynamical pattern during the first inversion; time: *250.2* to *300.0* secs; *di= 57.75*; *df= 57.84*. **C)** Stable dynamical pattern under normal vision after adaptation; time: *14569.9* to *14650.2* secs; *di= 40.06*; *df=20.09*. **D)** Stable dynamical patterns under inverted vision after adaptation; time: *14749.7* to *14818.5* secs; *di= 40.02*; *df=20.01*.

C1, and D1). Comparing the centroid positions for ρ_1 and ρ_3 we see how the surface changed for normal vision after adaptation to inverted vision. Comparing the centroid positions for ρ_3 and ρ_4 we see that the surfaces after adaptation are qualitatively the same under normal and inverted vision. The new shape and position of the attractor surface

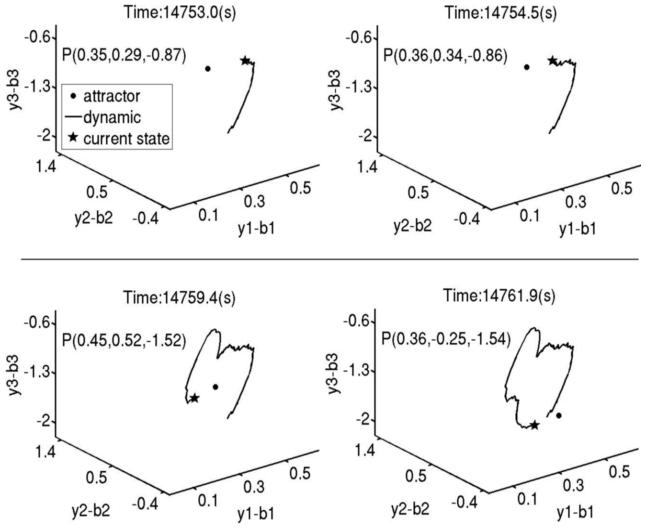


Figure 8: Four snapshots depicting the agent's transient internal dynamic while engaged in ρ_4. (time interval: [14753.0, 14761.9] secs.). P(y_1-b_1, y_2-b_2, y_3-b_3) indicates the attractor position.

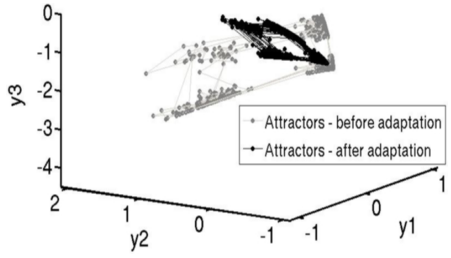


Figure 9: Surfaces defined by the movement of point attractors when the agent is doing phototaxis under normal vision before (gray) and after adaptation (black). Time intervals [*85.7*,*137.0*] and [*14569.9*, *14650.2*] secs, respectively.

after adaptation is caused by plastic changes that are activated when the system is homeostatic unstable.

Though the attractor surfaces are qualitatively the same after adaptation, the way the attractors move on the surface is different under normal and inverted vision. That is the reason why ρ_3 and ρ_4 are different (see Fig. 10-A2, B2, C2, and D2). While ρ_3 is generated by the movement of an attractor between the four clusters in the order $4 \rightarrow 3 \rightarrow 2 \rightarrow 1$, ρ_4 is generated by $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$.

Switching between dynamical regimes (*e.g.* switching from ρ_3 to ρ_4) does not require homeostatic instability. At the end of the agent's lifetime, after many plastic activations, the agent switches between the dynamical patterns without activation out of its viable region (see Fig. 11).

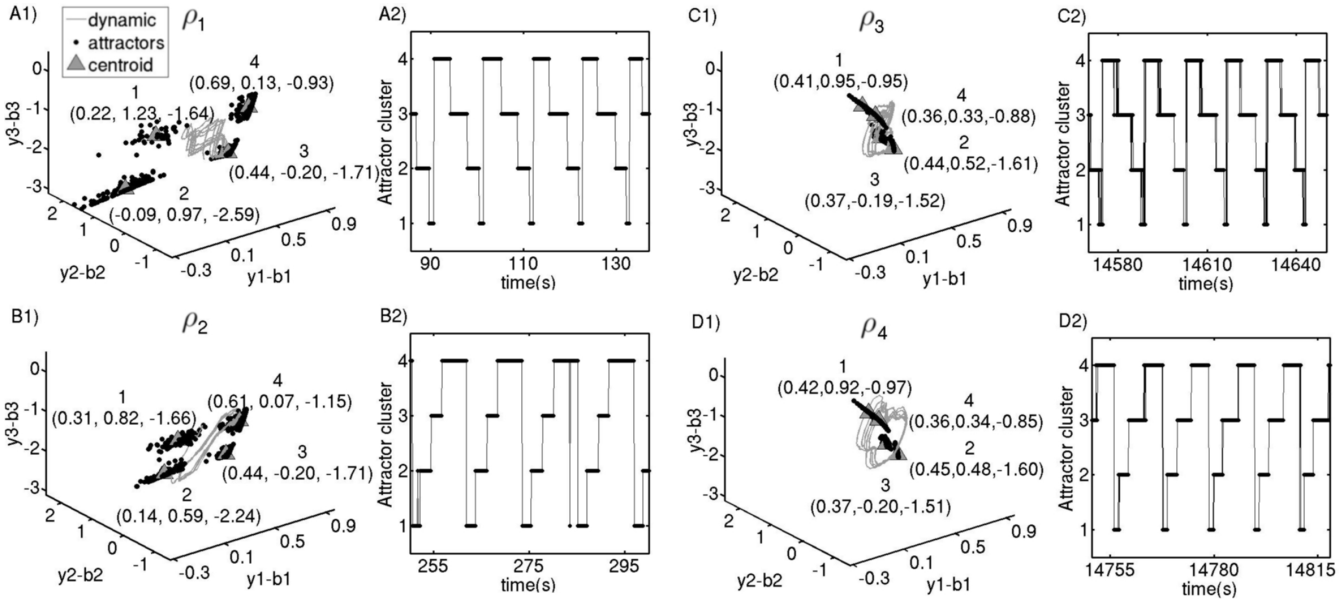


Figure 10: Phase space (*A1*, *B1*, *C1* and *D1*) depictions: dynamics of the internal nodes (gray lines); point attractors and attractor layout (black dots); cluster centroids (numbering from 1 to 4). Temporal sequence of the movement of attractors (*A2*, *B2*, *C2* and *D2*) shows how the attractors move between clusters over time. The time intervals to generate these graphs are the same as those in Fig 7

.

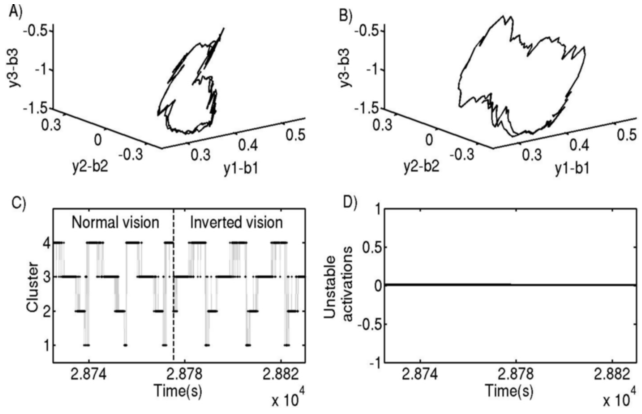


Figure 11: **A** and **B** depict dynamical patterns under normal and inverted vision, respectively. **C** depicts the difference between dynamic of attractors before and after inversion. **D** depicts the number of activations out of the homeostatic stable region.

Discussion

We would like to point out some of the important implications of this model. First, it has practical importance for the design of artificial neural network systems that can learn different behaviours. It is commonly believed that when a network system learns a new behaviour, the activation of neural plasticity will perturb the existing weight configuration of previously acquired behaviours and therefore will have a detrimental effect on the systems overall performance. One traditional way to address this so-called problem of neural interference is by taking inspiration from the modular computer architecture, namely by dividing the neural system into non-overlapping neuronal groups. However, here we have demonstrated that this kind of structural modularity is not the only way for one system to realize different styles of behaviour. Even a completely integrated system can achieve behavioural differentiation because the behaviours can be generated by different dynamical regimes on the phase space.

Accordingly, the current model also has important implications for our scientific understanding of the nervous system. It is a widely held belief in neuroscience that different cognitive functions map onto distinct regions of the brain, a belief reinforced by the advent of various brain imaging methods. This appeal to structural localizability may be valid to some extent. However, the model presented in this paper is a proof of concept that this is not the only way of realizing functional differentiation. Rather than focusing on anatomical divisions alone, it is also possible to take the nervous system as one integrated system which can realize a multiplicity of behaviours by transiting between different dynamical regimes.

Conclusion

We minimally replicated the psychological experiment described by Taylor based on assumptions drawn from Ashby's and Turrigiano's works. While Taylor's experiment shaped the desired behaviour, our assumptions constrained the dynamics of the mechanism underlying behaviour. Thus, the methodology to obtain the model we wanted to investigate incorporated restrictions on the task and on the agent's internal dynamic. Once the model was obtained we studied its dynamic in order to suggest answers to the questions Q1 and Q2 (detailed in the introduction).

In order to answer the question Q1, we showed that the dynamical regime in which the system engages under normal vision changes after adaptation to inverted vision (ρ_1 changed to ρ_3). As the system is relatively simple (only 3 CTRNN nodes) and fully-connected, even small reorganizations to accommodate new stable regimes are expected to affect pre-existing dynamics. Hence we can not generalize and say that pre-existing stable regimes always change when the system adapts to a new condition. More complex system, such as the brain, probably engages in independent dynamical regimes under different environmental conditions.

In order to answer the question Q2, we showed that homeostatic instability is not necessary for switching between dynamical regimes. This result contributes to research on brain dynamics as it complements the theoretical claim that Lyapunov instability is one generic mechanism for flexible switching among multiple attractive states; that is, for entering and exiting patterns of behaviour (Kelso, 1995). Indeed Ashby has already demonstrated that a system can switch between dynamical regimes without homeostatic instability. The difference is that, while Ashby uses the homeostat we use a more complex model where the homeostatic mechanism is intertwined with the mechanism that coordinates the movement of an agent that is continuously interacting with its environment. Thus, our investigation confirms Ashby's demonstration in a more complex environment and also complements Kelso's hypothesis about the importance of Lyapunov instability as a mechanism for switching dynamics.

We have also shown that qualitatively similar behaviours (phototaxis) can be generated by different dynamics; the agent's simulated nervous system operates in transient dynamics towards an attractor that continuously moves in the phase space; and plasticity moves and reshapes the attractor landscape in order to accommodate a stable dynamical regimes to deal with inverted vision.

Acknowledgements

Bruno A. Santos acknowledges financial support from Brazilian National Council of Research, CNPq. Thanks to Jose Fernandez-Leon, Lucas Wilkins, Matthew Egbert, Renan C. Moioli and Xabier Barandiaran for valuable discussions, and to the reviewers of an earlier version.

References

Ashby, W. R. (1947). The nervous system as physical machine: With special reference to the origin of adaptive behavior. *Mind, New Series*, 56(221):44–59.

Ashby, W. R. (1960). *Design for a brain : the origin of adaptive behaviour*. Chapman: London, 2nd edition.

Beer, R. D. (1995). On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior*, 3:469–509.

Bernard, C. (1927). *An introduction to the study of experimental medicine*. Macmillan, New York.

Cannon, W. B. (1932). *The wisdom of the body*. New York, NY, US: W W Norton & Co.

Cooper, S. J. (2008). From claude bernard to walter cannon. emergence of the concept of homeostasis. *Appetite*, 51:419–427.

Di Paolo, E. A. (2000). Homeostatic adaptation to inversion of the visual field and other sensorimotor disruptions. In J-A. Meyer, A. Berthoz, D. F. H. R. and Wilson, S. W., editors, *From Animals to Animals, Proc. of the Sixth International Conference on the Simulation of Adaptive Behavior, SAB'2000*, pages 440–449. MIT Press.

Di Paolo, E. A. (2003). *Dynamical Systems Approach to Embodiment and Sociality*, chapter Organismically-inspired robotics: homeostatic adaptation and natural teleology beyond the closed sensorimotor loop, pages 19–42. Advanced Knowledge International, Adelaide, Australia.

Harvey, I. (2001). Artificial evolution: A continuing saga. In Gomi, T., editor, *Evolutionary Robotics: From Intelligent Robots to Artificial Life - Proc. of 8th Intl. Symposium on Evolutionary Robotics*. Springer-Verlag Lecture Notes in Computer Science LNCS 2217.

Kelso, J. A. S. (1995). *Dynamic patterns: the self-organization of brain and behavior*. MIT Press.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Symposium on Math, Statistics and Probability*, pages 281–297. Berkeley, CA: University of California Press.

Strogatz, S. H. (2000). *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering*. Perseus Books.

Taylor, J. G. (1962). *The Behavioral Basis of Perception*. New Haven: Yale University Press.

Turrigiano, G. G. (1999). Homeostatic plasticity in neuronal networks: the more things change, the more they stay the same. *Trends in Neurosciences*, 22(5):221–227.

Turrigiano, G. G., Leslie, K. R., Desai, N. S., Rutherford, L. C., and Nelson, S. B. (1998). Activity-dependent scaling of quantal amplitude in neocortical neurons. *Nature*, 391:892–896.

Turrigiano, G. G. and Nelson, S. B. (2004). Homeostatic plasticity in the developing nervous system. *Nature Reviews Neuroscience*, 5:97–107.

Asynchronous Parallel Self-Replication Based on Logic Molecular Model

Katsuhiko Nakamura

School of Science and Engineering, Tokyo Denki University
Saitama-ken,350-0394 Japan.
nakamura@rd.dendai.ac.jp

Abstract

This paper discusses asynchronous parallel universal computation and self-replication based on a computation model, called a logic molecular model, or a parallel production system (PPS). The program in this model consists of extended Horn clause rules, which are used for forward deduction of unit clauses, called molecules, from unit clauses in working memory. All possible deductions in the system are asynchronously executed in parallel. This formalism is also effective in representing a broad class of speed-independent asynchronous computation and systems including parallel parsing and cellular automata. It is shown that for any PPS program P, there is a set of molecules that contains the coded program of P, which replicates itself by asynchronous parallel computation in time proportional to $\log n$, where n is the number of rules in P.

Introduction

The self-replication of complex systems is universal in biology, as cell division and propagation are essential to living organisms. Many biologists believe that the appearance of self-replicating molecules marked the origin of life. Several hypothetical models of the first self-replication have been presented and discussed in evolutionary biology (Dawkins, 2004). In information science, there have been several theoretical models of self-replication intended to clarify the principles and conditions of self-replication (Hutton, 2003; Sipper, 1998). Some of these models can be applied to artificial self-organization in complex systems including amorphous computing (Abelson et al., 2007) and molecular computing.

Von Neumann adopted a cellular automaton (CA) model of self-replication and presented a two-dimensional (2-D) 29-state CA with universal computation power and self-replicating processes in his last note titled "*Theory of Self-Reproducing Automata*" (von Neumann, 1966). A CA is essentially a parallel system used as a model of parallel computation. Transitions in von Neumann's CA, however, are serial and sequential because the universal computation and self-replication are based on a universal Turing machine. After von Neumann, lot of work focused on the parallel computation power of CAs and self-replication on CAs (Sipper, 1998). Nevertheless, there has been little work on parallel universal computation and parallel self-replication, not only using CAs but also with other computation models. Albert and Culik (1987) showed a 1-D CA with parallel universal computation power in the sense that the CA can simulate any 1-D CA in linear time. Nakamura (1997) showed a 1-D CA with parallel self-replication processes and parallel universal computation power in a similar sense. Nehaniv (2002) showed an asynchronous cellular automaton with a self-reproducing pattern known as "Langton's loop."

This paper proposes a parallel computation model, called a *logic molecular model*, or a *parallel production system* (PPS), and shows that this simple formalism is effective in modeling a broad class of asynchronous parallel computations and biological systems. This model is intended to be a simple and general basis not only for parallel universal computation such as universal Turing machines for serial computation but also for the modeling of self-replication in biological systems.

The logic molecular model proceeds as follows.

1. Every global state of the system is represented by a multiset of *molecules*, which are data tokens in working memory from the point of parallel computation.

2. A program in PPS is a set of production rules, or simply rules. The rules specify the interactions of the molecules by forward, data-driven deduction. Deduction by the rules is a kind of hyper-resolution (Robinson, 1992); each rule is described as an extended Horn clause rule and every molecule in the system as a unit clause.

3. All applicable deductions are asynchronously executed in parallel. Therefore, the computation needs to be speed-independent to reach a definite result in spite of the indefinite orders of the transitions of the elements.

Since the pioneering work of von Neumann, CAs have been used for modeling not only biological systems but also other complex systems. However, modeling using CAs has the following limitation.

- In a CA model, the arrangements of the cells and the interconnections among cells are strictly regular and fixed. This restriction prevents us not only from using CAs to model general parallel systems but also from applying CAs to parallel computers.

- Most standard CA models are synchronous systems. Synchronization generally simplifies the construction of deterministic systems. Nevertheless, it is a fundamentally accepted principle that asynchronous systems are generally faster than synchronous systems in large scale parallel systems, because the synchronization period is determined by the maximum delay in the system. As there are no specific synchronous biological systems, asynchronous systems are more appropriate for modeling self-replication.

There has been some researches into extensions of CAs. The Lindenmayer system (or L-system) (Lindenmayer, 1968) is an extended CA where every cell can propagate itself; the L-system is intended to model biological development. Nakamura (1981) showed that any synchronous d-D CA ($d = 1, 2, \cdots$) can be transformed into an asynchronous d-D CA while preserving its parallel computation power.

Recently, there have been several models other than CAs called biologically-motivated systems or natural computing. The chemical abstract machine (CHAM) (Berry and Boudol, 1992) and the GAMMA language (Baâtre and Métayer, 1993) based on *multiset transformation* have some properties similar to our model. In these formalisms as well as in the logic molecular model, every global state of the system is a multiset of data elements. In CHAM, the global state can be considered a solution of molecules that interact with each other. CHAM and GAMMA, in which no data element is deleted from the global states, are intended to provide a simple paradigm of parallel computation. They are not intended to describe speed-independent asynchronous parallel processes as does the logic molecular model.

The logic molecular model integrates several paradigms including logic programming, production systems, and functional data-flow programming. Some explanation of the relations between these paradigms is essential. Hyper-resolution (Robinson, 1992) is closely related to unit resolution (Chang, 1970) and, has been studied for bottom-up computation with large data sets including deductive databases. The current work is intended to use deduction in logic programming to represent a broad class of asynchronous parallel computation.

In contrast to logic programming, most production systems, such as OPS-5 (Cooper and Wogrin, 1988), mainly employ forward deduction. Although the purpose and the control mechanisms are essentially different, our computation model has some similarities to production systems: the unit clauses in the global state correspond to data tokens in the working memory, unification to pattern matching and the extended Horn clause rules to production rules for forward deduction.

The control of our computation model is closely related to that used in data-flow programs (Dennis, 1975), as the operations are evoked by data tokens. Our PPS programs are more general and more powerful than the data-flow programs because each rule represents a general pattern of symbolic operations based on unification and unit resolution.

This paper is organized as follows. The next section describes the basic model and its asynchronous transition. The rules and their application to data are defined by using the notions in logic programming. The transitions of the global states are based on asynchronous circuit theory. The third section describes the decomposition of general rules into simpler rules, and extensions of the basic rules so that we can use the models for parallel functional processes. The fourth section shows a PPS that simulates a 1-D bounded synchronous CA. This result is closely related to the synchronous-to-asynchronous transformation of CAs. The fifth section describes a universal computation by the PPS. Based on this universal computation, the sixth section shows several parallel self-replicating molecules and self-replicating programs. The final section gives brief concluding remarks.

The Basic Model and Parallel Derivation

We use basic notions of logic programming such as unification and most general unifier to describe the pattern matching and application of the rules.

Parallel Production Systems

We use the notations and syntax of standard Prolog for variables, terms, lists and operators. A *constant* is either a number or an identifier (an atom in Prolog) that starts with a lower-case character, and a variable starts with an upper-case character and the underscore "_". A *term* is either a constant, a variable, or a complex term of the form $f(t_1, \cdots, t_k)$, where f is an identifier (a function or predicate symbol), and each t_i is a term. An *atom* is a term of the form either $p(t_1, \cdots, t_k)$, or p when $k = 0$, where p is a predicate symbol, and each t_i is a term.

A *substitution* θ is a mapping from a set of variables to a set of terms. For any term t, an *instance* $t\theta$ is a term in which each variable X defined in θ is replaced by its value $\theta(X)$. For any terms s and t, we say that s and t are *variants* of each other, if t is an instance of s and t is an instance of s. A *unifier* for two terms s and t is a substitution θ, such that $s\theta = t\theta$. The unifier θ is the *most general unifier* (mgu), if for every other unifier σ of s and t, $s\sigma$ and $t\sigma$ are instances of $s\theta$ and $t\theta$, respectively.

A *parallel production system* (PPS) is defined by its program and its initial global states. The program is a set of *rules* of the form

$$B_1, \cdots, B_m \rightarrow C_1, \cdots, C_n, \quad m, n \geq 0,\ m + n \geq 1.$$

where each of B_i and C_j is either an atom or a variable. The variable in a rule is instantiated to an atom, when the rule is applied as will be described later. The global state, or the working memory, of the PPS is the multiset of unit clauses (or atoms) called *molecules*. The initial global set generally contains the input information.

A rule $R = (B_1, \cdots, B_m \rightarrow C_1, \cdots, C_n)$ is *applicable* to molecules $A_1, \cdots, A_m$ in a global state W, if and only if there is a most general unifier θ such that: $A_i\theta = B_i\theta$ for all $1 \leq i \leq m$. In this case, we write $W \Rightarrow W'$ for the *result* W' of the application defined by

$$W' = (W - \{B'_1\theta, \cdots, B'_m\theta\}) \cup \{C'_1\theta, \cdots, C'_n\theta\}.$$

The relation $\Rightarrow^*$ denotes the reflective and transitive closure of "$\Rightarrow$". For any initial global state W_0, every global state W with $W_0 \Rightarrow^* W$ is called a *derivable* global state of S.

The application of rule R to molecules $A_1, \cdots, A_m$ is equivalent to the simultaneous hyper-resolution of n Horn clause rules,

$$C_1 \leftarrow B_1, \cdots, B_m, \quad \cdots \quad C_n \leftarrow B_1, \cdots, B_m,$$

and the m unit clauses $A_1, \cdots, A_m$, except that these unit clauses are deleted from the global state. Hence, each resultant unit clause is a logical consequence of the unit clauses in the global state and the Horn clauses.

Asynchronous Transition and Speed-Independence

Asynchronous systems generally must be speed-independent to achieve definite computation results in spite of the indefinite order of operations. We represent asynchronous transition in PPSes by applying the terminology of asynchronous circuit theory (Muller and Burtky, 1959) as in defining asynchronous cellular automata (Nakamura, 1981). As several different terms are used for similar notions in term rewriting system (TRS) theories, we have added some comments on these terms in parentheses.

An *allowed sequence* in a PPS is a finite or infinite sequence $W_0, W_1, W_2, \cdots$ of the global states such that $W_i \Rightarrow W_{i+1}$ for $i = 0, 1, 2, \cdots$ and there is no subscript $i_0 \geq 0$ such that a rule is applicable to a subset of molecules in W_i for all $i \geq i_0$. (This notion corresponds to *fair computation* in TRS.) This condition states that all the delays in the application of the rules are arbitrary but finite.

The class G of global states in a PPS is partitioned into subclasses by the equivalence relation $W \Rightarrow^* W'$ and $W' \Rightarrow^* W$ for any $W, W' \in G$. The equivalence class ("strongly connected components" in TRS) is partially ordered by the relation $\Rightarrow^*$. A PPS S is *speed-independent*, if and only if for all allowed sequences $W_0, W_1, W_2, \cdots$ starting with an initial global state W_0, there is an integer j_0 such that all global states $W_j, j \geq j_0$ are in a common equivalence class. In a speed-independent system, if there is a finite allowed sequence $W_0, \cdots, W_t$, then all the allowed sequences starting with W_0 terminate with W_t, which we call the *terminal state*.

A PPS S is *race-free*, if and only if for any derivable global states W and W' such that a rule R is applicable to some molecules in W and $W \Rightarrow W'$, either W' has the result of the application of R, or R is still applicable to the same molecules in W'.

A PPS S has the *Church-Rosser* (*diamond*) property, if and only if for any derivable global states W, X and Y with $W \Rightarrow X$ and $W \Rightarrow Y$, there is a global state Z such that:

$$\begin{array}{ccc} W & \Longrightarrow & X \\ \Downarrow & & \Downarrow \\ Y & \Longrightarrow & Z. \end{array}$$

Proposition 1 *Any race-free PPS is Church-Rosser, and any Church-Rosser PPS is speed-independent. The converses of these relations do not hold.*

Proof It is obvious from the definitions that any race-free PPS is Church-Rosser. We omit the proof that any Church-Rosser PPS is speed-independent because it is similar to the corresponding propositions in asynchronous circuit theory (Muller and Burtky, 1959) and in the theory of TRS.

To prove that the converse does not hold, consider the PPS with program, $p, q \rightarrow s$; $s, r \rightarrow u$; $q, r \rightarrow t$; $p, t \rightarrow u$, and initial global state $\{p, q, r\}$. This system is Church-Rosser but not race-free. Consider another PPS with program, $p, q \rightarrow s$; $s, r \rightarrow u$; $p, q, r \rightarrow u$, and initial global state $\{p, q, r\}$. This system is speed-independent, but not Church-Rosser. □

Synchronous Transition

A *synchronous transition sequence* of a PPS is a subsequence $W_0, W_1, W_2, \cdots$ of an allowed sequence such that all applicable rules in W_i, and no other rule, have applied in W_{i+1} for $i = 0, 1, 2, \cdots$. In any race-free PPS, there is a unique synchronous transition sequence for any initial global state. The length of the synchronous transition sequence represents the number of steps, or time, of asynchronous computation where all applications of the rules require a constant time.

Example: Parallel Parsing of a CFL

The first example is parallel bottom-up parsing of the parenthesis languages, i.e, the set of strings with the same number of a's and b's such that no prefix contains more b's than a's. Each rule in the following program represents a production rule for a context free grammar, as in definite clause grammars (DCGs) (Imada and Nakamura, 2010).

[Parsing parenthesis language]

```
a(I,J), b(J,K)→ s(I,K,s(a,b)).
s(I,J,P),s(J,K,Q)→ s(I,K,s(P,Q)).
a(I,J),s(J,K,P),b(K,L)→ s(I,L,s(a,P,b)).
```

Suppose that the initial global state contains the following molecules representing the string $aababb$.

```
a(0,1).a(1,2).b(2,3).a(3,4).b(4,5).b(5,6).
```

The computation proceeds as follows and terminates with a molecule that has a term representing the derivation tree.

```
  a(0,1),a(1,2),b(2,3),a(3,4),b(4,5),b(5,6)
⇒ a(0,1),s(1,3,s(a,b)),a(3,4),b(4,5),b(5,6)
⇒ a(0,1),s(1,3,s(a,b)),s(3,5,s(a,b)),b(5,6)
⇒ a(0,1),s(1,5,s(s(a,b),s(a,b))),b(5,6)
⇒ s(0,6,s(a,s(1,5,s(s(a,b),s(a,b))),b))
```

This computation is speed-independent and terminates with a single molecule having the definite derivation tree for any initial state representing a string in the language. As the grammar is ambiguous, the computation with other initial global states, for example, those for parsing a string $ababab$, cannot be speed-independent. Nevertheless, parsing terminates with a final molecule containing one of the possible derivation trees.

Extensions of the Basic Model

This section describes transformations and extensions of the rules in the basic model. Transformed PPSes simulate the original PPSes in the following sense. A PPS S' *simulates* a PPS S, if and only if there is a computable function $c : U' \rightarrow U$, where each of U and U' is the class of global states of S and S', respectively, such that if for any allowed sequence $W_0, W_1, W_2, \cdots$ in S', the sequence $c(W_0), c(W_1), c(W_2), \cdots$ is an allowed sequence in S, provided that we ignore any repetitions.

Decomposition of Rules

Any rule $B_1, \cdots, B_m$ -> $C_1, \cdots, C_n$ with $m > 2$ and/or $n > 2$ can be decomposed into simpler rules with at most two atoms on each of the left and right hand sides. First we recursively transform a rule $B_1, \cdots, B_m \rightarrow C_1, \cdots, C_n$ with $m > 2$ into the following three rules.

$$B_1, \cdots, B_{m/2} \rightarrow r_1(X_1, \cdots, X_k),$$
$$B_{m/2+1}, \cdots, B_m \rightarrow r_2(X_1, \cdots, X_k),$$
$$r_1(X_1, \cdots, X_k), r_2(X_1, \cdots, X_k) \rightarrow C_1, \cdots, C_n$$

where r_1 and r_2 are unique predicate names and $X_1, \cdots, X_k$ is a list of all the variables in the rule. Secondly, we recursively decompose the rule of the form $B_1, B_2 \rightarrow C_1, \cdots, C_n$ with $n > 2$ into the three rules with unique predicate names q_1 and q_2:

$$B_1, B_2 \rightarrow q_1(X_1, \cdots, X_k), q_2(X_1, \cdots, X_k),$$
$$q_1(X_1, \cdots, X_k) \rightarrow C_1, \cdots, C_{n/2},$$
$$q_2(X_1, \cdots, X_k) \rightarrow C_{n/2+1}, \cdots, C_n.$$

Non-Deleting Molecules

We can extend the basic rule so that any molecule matching with an atom on the left side of the rule remains undeleted from a global state. Any molecule unifying the atom with the prefix operator $\star$, as in

$$B_1, \cdots, *B_i, \cdots, B_m \rightarrow C_1, \cdots, C_n,$$

is the *non-deleting molecule*, which is not deleted from the global states when this rule is applied. The asterisk can be prefixed in any atoms on the left-hand side. This rule can be replaced by the rule

$$B_1, \cdots, B_i, \cdots, B_m \rightarrow B_i, C_1, \cdots, C_n.$$

We can apply this transformation to any number of non-deleting molecules in the program. Note that any PPS in which all atoms are non-deleting is race-free.

Evaluable Predicates and Terms

We can extend the use of programs in PPS from pure logical deduction to functional computation by adding some functions to test conditions and evaluate the arithmetic expressions. For the first extension, atoms on the left side can be terms with "external predicates" to test conditions and converting data term. In this paper, we represent these atoms by deterministic Prolog goals with the prefix operator `#`, e.g., the term `#(X > Y+1)`, where operator `>` is the external predicate. These terms are evaluated after all necessary variables in the condition have been instantiated. We consider the term to be a non-deleting atom and the system to have an implicit model of the external predicate, a possibly infinite set of ground unit clauses.

For the second extension, we allow the rules to contain evaluable terms of arithmetic expressions with the prefix `$` in the atoms, e.g., `$(2.0*X+1.0)`. The arithmetic expression can be placed on both the left and right hand sides of a rule, and it is evaluated and replaced by its value when the rule is applied.

Simulating 1-D Cellular Automata

This section shows a PPS that simulates a 1-dimensional synchronous cellular automaton (1-D CA) and has an identical computational result. We suppose that the CA is bounded in the sense that the leftmost and rightmost cells are fixed and have the special boundary state $\natural$. The CA with three neighbors is defined by a set Q of cell states including $\natural$ and a local function $f : Q^3 \rightarrow Q$. We represent n cells by the numbers $1, 2, \cdots, n$ and each configuration at time i by $\natural q_1^i\ q_2^i \cdots q_n^i \natural$.

We construct a PPS P_Z simulating a 1-D CA Z as follows.

1. For all even time points $t \geq 0$, the state q_j^t of each cell $j, 2 \leq j \leq n-1$ is represented by three molecules $c(j, q_j^t), l(j, q_j^t), r(j, q_j^t)$, and the state of the leftmost

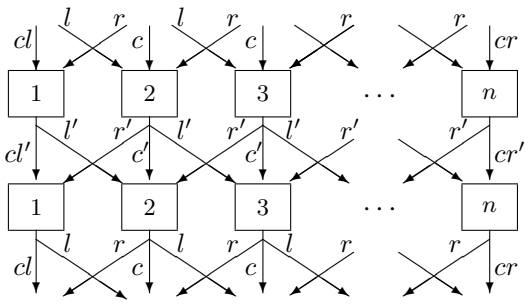


Figure 1: A hypothetical data flow diagram of PPS P_Z for simulating a 1-D CA.

and rightmost cells 1 and n by $cl(1, q_1^t), l(1, q_1^t)$, and $cr(n, q_n^t), r(n, q_n^t)$, respectively. For all odd time points t, the state q_j^t of each cell is represented similarly, except that the predicate symbols c, l and r are replaced by c', l' and r', respectively.

2. The initial global state is the set of the following molecules, which represents the initial configuration $\natural q_1^0\ q_2^0 \cdots q_n^0 \natural$.
   (a) $cl(1, q_1^0),\ l(1, q_1^0)$.
   (b) $l(j, q_j^0),\ c(j, q_j^0),\ r(j, q_j^0),\ 2 \le j \le n-1$.
   (c) $r(n, q_n^0),\ cr(n, q_n^0)$.

3. The program of P_Z is the set of the following rules, where $(V$ `is` $f(L, C, R))$ is a Prolog expression that unifies V with the value of the local function f.

[Program of P_Z for simulating 1-D CA]

```
cl(1,C),r(2,R),#(V is f(♮,C,R))→
                              cl′(1,V),l′(1,V).
c(J,C),#(2 ≤ J ≤ n−1),l($(J-1),L),
    r($(J+1),R),#(V is f(L,C,R))→
                              r′(J,V),c′(J,V),l′(J,V).
cr(n,C),l(n−1,R),#(V is f(L,C,♮))→
                              cl′(n, V),r′(n,V)).
cl′(1,C),r′(2,R),#(V is f(♮,C,R))→
                               cl(1, V),l(1,V).
c′(J, C),#(2 ≤ J ≤ n−1),l′($(J-1), L),
    r′($(J+1),R),#(V is f(L,C,R) →
                              r(J,V),c(J,V),l(J,V).
cr′(n,C),l′(n−1,R),#(U is f(L,C,♮))→
                               cl(n,V),r(n,V).
```

Fig. 1 shows a hypothetical data flow diagram for transitions in P_Z.

Proposition 2 *If the synchronous transition in the 1-D CA Z terminates at time t with the configuration $\natural q_1^t\ q_2^t \cdots q_n^t \natural = \natural q_1^{t+1}\ q_2^{t+1} \cdots\ q_n^{t+1} \natural$, then all the allowed sequences in the PPS P_Z fall into the final equivalence class, in which every global state represents this configuration.*

Proof (*Outline*) We can prove the following two lemmas by mathematical induction on the number of applications of the rules.

1. The proposition holds, if we restrict the allowed sequences to one that includes the synchronous transition sequence.
2. The PPS P_Z is race-free, i.e., all the applications of rules to two molecules and three molecules are not affected by the other operations.

These lemmas imply that the proposition is true for all allowed sequences by Proposition 1. □

We restrict the 1-D CA to the bounded CA in order to simplify the construction of P_Z. It is not difficult to extend the CA model to a more general one such that the boundaries expand with time.

Parallel Universal Computation

A universal program U for PPS is an interpreter such that for any program P, U inputs molecules for a coded program of P and (coded) data molecules D and outputs the molecules that are equivalent to the result of the computation of P for D. The universal program not only describes how the programs are computed, but also makes it possible to easily extend the language. Furthermore, using the universal program, PPS programs can generate programs to be executed later. In particular, the universal program for PPS provides an environment with fixed interaction rules, in which the codes of rules are active molecules that interact with the data molecules.

In this section, we show a universal program for race-free PPSes. We represent the internal code of a rule $B_1, \cdots, B_m \to C_1, \cdots, C_n$ without evaluable predicates by the molecule,

$$\texttt{rbc}([B_1, \cdots, B_m], [C_1, \cdots, C_n]),$$

where the list can be an empty list $[\,]$ when $m = 0$ or $n = 0$ and $[B_1]$ or $[C_1]$ are also written B_1 and C_1. For example, $\star\texttt{rbc}(B, C)$ and $\star\texttt{rbc}(B, [\,])$ are codes for $B \to C$ and $B \to$, respectively. We represent a rule having an atom $\#P$ with an evaluable predicate by

$$\texttt{rbpc}([B_1, \cdots, B_m], P, [C_1, \cdots, C_n]),$$

The following universal program uses list operations in Prolog to process sequences of atoms.

[Parallel universal program]

```
⋆rbc([B|L],CL),B  → rbc(L,CL).
rbc([],[]) →.
rbc([],[C|L]) → C, rbc([],L).
rbc([B|L],CL),B  → rbc(L,CL).

⋆rbpc([B|L],P,CL),B  → rbpc(L,P,CL).
rbpc([],P,[C|L]), #P → C, rbc([],L).
rbpc([B|L],P,CL),B  → rbpc(L,P,CL).
```

Consider that the global state contains a coded rule rbc($[B_1, \cdots, B_m]$,$[C_1, \cdots, C_n]$) and molecules $B'_1, \cdots, B'_m$. If B_i unifies with B'_i for each i by an mgu θ_i, the universal program generates molecules $(C_1, \cdots, C_n)\theta_1 \cdots \theta_m$. This process proceeds correctly in race-free computation of a PPS.

Self-Replication

In this section, we show not only small simple self-replicating sets of molecules, but also self-replication of coded programs composed of a number of *labelled* molecules, such that each replicated molecule is a variant of the original molecule except that the label is different from that of the original. By labeling groups of molecules, the global state can have two or more groups of equivalent coded programs working independently. We add a common label to either the first element of coded rules or the first argument of molecules in a group.

A set S of molecules is *self-replicating*, if and only if there are a simple "start command" molecule p and a set S' of molecules such that:

1. $S \cup \{p\} \Rightarrow^* S \cup S'$, and $S \cup S'$ is a terminal state; and
2. each member in S' is either a variant, or a variant with different label, of a member of S and vice versa, and hence, S' is also a self-replicating set.

In this section, we represent the coded rules using the more readable form ($[B_1, \cdots, B_m]$, #P $\rightarrow$ $[C_1, \cdots, C_n]$) in stead of the form rbpc($[B_1, \cdots, B_m]$, #P, $[C_1, \cdots, C_n]$).

Simple Self-Replicating Molecules

One common method of self-replicating programs is based on the doubling of a part within a program.

[Self-replication by doubling a term]

```
rep → p((p(R)→[(rep → p(R)),R])).
p(R)→ (rep → p(R)), R.
```

When the molecule rep is given, the first rule generates the molecule p((p(R)→[(rep→p(R)),R])). From this molecule, the second rule generates a pair of molecules, which is a variant of the coded program.

Self-Replicating Molecules with Labels

We can transform the simple self-replicating program above to a self-replicating set of molecules identified by a unique label.

Because of the restriction known as *single assignment rule* in logic programming, it is not straightforward to change part of a term without reconstructing the term. To assign the labels in the molecules to different labels, we use *mutable terms*, which are proposed to realize global variables in Prolog (Nakamura, 2009). We consider the mutable term as a variable with assignable values[1]. We represent mutable terms with a value v by $mt($v$)$, and suppose that its value can be changed to v' by evaluating the term alter($mt($v$), v'$).

The following rules constitute self-replicating molecules with label l.

[Self-replication with labels by doubling a term]

```
rep($mt(l),L1) → p($mt(l),L1,
    (p($mt(l),L1,R), #alter($mt(l),L2) →
    [(rep($mt(l),L2)→ p($mt(l),L2,R)),R])).

p($mt(l),L1,R),#alter($mt(l),L1)→
    (rep($mt(l),L2)→ p($mt(l),L2,R)),R.
```

For the starting molecule rep($mt(l),m), this program generates two molecules that are equivalent to the original program except that the mutable term $mt(l) is changed to $mt(m). We can repeat this self-replication process by giving the starting command rep($mt(m),n).

Self-Replication by Copying Molecules

Another common method for self-replicating programs is copying such that each part of the program alternately copies the other parts or a program code exists with the capability to inspect and copy itself (Laing, 1976; Ray, 1992; Hutton, 2003).

In the following self-replicating program, two coded rules copy each other.

[Self-replication by copying]

```
rep,*([rpl|B]→D)→([rpl|B]→D),rpl
rpl,*([rep|B]→D)→([rep|B]→D).
```

Note that the term ([rpl|B]→ D) on the left side of the first rule unifies with the second rule. For the starting command rep, the first rule generates a replicated coded rule of the second rule and the molecule rpl, which starts the second rule. The second rule generates a copy of the first rule.

There is also another type of self-replicating molecules that use copying.

[Parallel self-replication by copying]

```
rep → rpa,rpb.
rpa,*([rpb|B]→D)→([rpb|B]→D).

rpb,*([rpa|B]→D),*([rep|B1]→D1) →
     ([rpa|B]→D),([rep|B1]→D1).
```

For the starting command rep, the first rule generates two molecules rpa and rpb, which start the second and third rules, respectively. The second and third rules generate copies of the third and second rules. As this process can run in parallel, the second and third molecules can be used

[1]The mutable terms can be realized by using lists terminated by variables so that the last element E_k of the list $[E_1, \cdots, E_k|X]$ represents the value. This method is simple but not efficient.

```
%%%%%%%%%%%%%%%  Rule 1  %%%%%%%%%%%%%%%%%%%
 rep_j($mt(l),L)→
      rpa_j($mt(l),L),rpb_j($mt(l),L).

%%%%%%%%%%%%%%%  Rule 2  %%%%%%%%%%%%%%%%%%%
rpa_j($mt(l),L),
*([rpb_j($mt(l),L)|B],#P → D),
*([rule_{2j-1},$mt(l)|B1]→D1),
*([rule_{2j},$mt(l)|B2]→D2),
#alter($mt(l),L)
→
([rpb_j($mt(l),L1)|B],#alter($mt(l),L1)→D),
([rule_{2j-1},$mt(l)|B1]→D1),
([rule_{2j},$mt(l)|B2]→D2),
rep_{2j}($mt(l),L), rep_{2j+1}($mt(l),L).

%%%%%%%%%%%%%%%  Rule 3  %%%%%%%%%%%%%%%%%%%
rpb_j($mt(l),L),
*([rep_j($mt(l),L1)|B] →D),
*([rpa_j($mt(l),L1)|B1],#P →D1),
#alter($mt(l),L)
→
*([rep_j($mt(l),L1)|B],#alter($mt(l),L1)→D),
([rpa_j($mt(l),L1)|B1],#alter($mt(l),L1))→D1).
```

Figure 2: Three rules in module M_j for self-replication of the coded program.

simultaneously as rules and objects of the operation. Therefore, the second and third rules should be non-deleting to keep this PPS race-free.

Self-Replication of Coded Programs

Based on the self-replication by copying shown in the last subsection, we can transform a labelled PPS program to a self-replicating set of molecules.

Let P be any program of N rules. We suppose that each j-th rule in P is unified with the term `([rule`$_j$`,$mt(l)|B]→D)` with the initial label `l`. The transformed program is the union of $M_1, M_2, \cdots, M_{N/2}$ and P, where M_j is a module of the three rules in Fig. 3 for $1 \leq j \leq N/2$, and the second rule contains:

1. the term `([rule`$_{2j}$`,$mt(l)|B2]→D2)` in both sides of the rule, if and only if $2j \leq N$; and
2. the terms in the right hand side `rep`$_{2j}$`($mt(l),L)` and `rep`$_{2j+1}$`($mt(l),L)`, if and only if $j \leq N/2$.

For the starting command `repj($mt(l),m)`, each rule in M_j works as follows:

1. The first rule generates molecules `rpa`$_j$`($mt(l),m)` and `rpb`$_j$`($mt(l),m)`;
2. The second rule replicates the $(2j-1)$-th rule and the $2j$-th rule of P, if $2j \leq N$, and generates the molecules `rep`$_{2j}$`($mt(l),m)` and `rep`$_{2j+1}$`($mt(l),m)`, if $j \leq N/2$; and

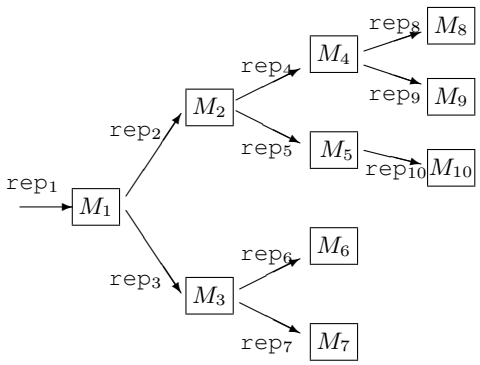


Figure 3: A data flow diagram for the self-replication of a coded program with 20 rules. Each of 10 modules replicates two program rules and three rules of the module itself.

3. The third rule replicates the first and second rules.

Fig.3 illustrates the data flow in the self-replication of program T for the case $N = 20$. Each modules M_j, with $1 \leq j \leq 4$ generates two molecules `rep`$_{2j}$`($mt(l),m)` and `rep`$_{2j+1}$`($mt(l),m)`, while M_5 generates only `rep`$_{10}$`($mt(l),m)`. Every module replicates two rules in P and three rules of the module.

The following proposition summarizes the discussion in this section.

Proposition 3 *For any PPS program P, we can construct a set T of labelled molecules such that*

1. $|T| \leq 2.5 \cdot |P|$.
2. *T contains a coded program equivalent to P.*
3. *T replicates itself in time $O(\log |P|)$ by race-free computation of T and the start command molecule: it generates all the modules each of which is a variant of the corresponding element in T with a different label .*

We can reduce the factor of 2.5 for the size $|T|$ to less than 2 by changing the module to copy three or more rules.

Concluding Remarks

In this paper, we discussed asynchronous parallel universal computation and self-replication based on a computation model, called the logic molecular model, or the parallel production system. The model is based on the parallel application of production rules, which is forward deduction based on extended Horn clauses.

We showed that for any PPS program, there is a set of molecules that contains the coded program, which replicates itself by asynchronous parallel computation in time proportional to $\log n$, where n is the number of rules in the program. This type of self-replication is important for a theoretical model of biological systems, in which the most

processes including self-replication seem asynchronous and parallel.

The essential features of the molecular model are summarized as follows.

- The PPS is a simple model for parallel functional computation as well as for parallel logical deduction.
- The programs in the PPS are compact, as the rules represent patterns of deductions and do not specify the order of the deductions. The PPS is effective for specifying several parallel computations, including parallel parsing, simulating 1-D CA and universal computation.
- As a universal Turing machine and universal programs for sequential computation, the parallel universal program suggests the generality and the computation power of the parallel computation model. By the universal program, the molecules are not only data tokens but also coded rules that can generate other molecules of coded rules. The coded rules are similar to enzymes in biological systems because these molecules control the interactions of other molecules.

We tested several PPS programs including parallel sorting using bitonic sort in addition to the example programs in this paper by using a serial interpreter of PPS in Prolog.

An interesting question regarding self-replication is the cost required to transform a coded program into a self-reproducing set of molecules. The transformed self-replicating coded program in the previous section requires extra $1.5N$ rules for a program with N rules. Reducing the number of rules in parallel self-reproducing programs is a topic for future work to address. Other future problems include:

- implementation of PPS in a concurrent environment;
- machine learning of self-replicating PPSes by extending methods of learning definite clause grammars (DCGs) (Imada and Nakamura, 2010); and
- application of this paper's approaches to amorphous computing (Abelson et al., 2007), to DNA and molecular computing and to chemical kinematics.

Acknowledgement

The author would like to thank Keita Imada for his help in preparing the manuscript. This work is partially supported by KAKENHI 21500148 and the Research Institute for Technology of Tokyo Denki University, Q09J-06.

References

Abelson, J., Beal, J., and Sussman, G. (2007). Amorphous computing. Computer Science and Artificial Intelligence Laboratory Technical report, MIT-CSAIL-TR-2007-030.

Albert, J. and Culik, K. (1987). A simple universal cellular automaton and its one-way and totalistic version. *Complex Systems*, 1:1–16.

Baâtre, J.-P. and Métayer, D. (1993). Programming by multiset transformation. *Comm. ACM*, 36:98–111.

Berry, G. and Boudol, G. (1992). The chemical abstract machine. *Theoretical Computer Science*, 96:217–248.

Chang, C. (1970). The unit proof and the input proof in theorem proving. *Jour. of ACM*, 17:689–707.

Cooper, T. and Wogrin, N. (1988). *Rule-Based Programming with OPS-5*. Morgan Kaufmann.

Dawkins, R. (2004). *The Ancestor's Tale: A Pilgrimage to the Dawn of Life*. Weidenfeld & Nicolson, London.

Dennis, J. (1975). First version of data flow procedure language. MIT/ LCS/ TM-61, MIT.

Hutton, T. (2003). Evolvable self-replicating molecules in an artificial chemistry. *Artificial Life*, 8:341–356.

Imada, K. and Nakamura, K. (2010). Search for semi-minimal rule sets in incremental learning of contextfree and definite clause grammars. *IEICE Trans.*, E93-D:1197–1204.

Laing, R. (1976). Automaton inspection. *Journal of Computer and System Sciences*, 13:172–183.

Lindenmayer, A. (1968). A mathematical model for cellular interaction in development i, filament with one-side inputs. *Jour. of Theoretical Biology*, 18:280–289.

Muller, D. and Burtky, S. (1959). A theory of asynchronous circuits. In *Proc. of International Symposium on the Theory of Switching 29*, pages 204–243. Annuals of the Computation Laboratory of Harvard University.

Nakamura, K. (1981). Synchronous to asynchronous transformation of polyautomata. *Jour. of Computer and System Sciences*, 23:22–37.

Nakamura, K. (1997). Parallel universal computation and self-reproduction in cellular spaces. *IEICE Trans.*, E80-D:547–552.

Nakamura, K. (2009). Proposal for global variable in prolog. ISO/IEC Draft PDTR 13211-X:2010 http://www.sju.edu/ jhodgson/wg17/Drafts/pdtr10.pdf.

Nehaniv, C. L. (2002). Self-reproduction in asynchronous cellular automata. *Proc. of NASA/DOD Conference on Evolvable Hardware (EH02)*, pages 201–209.

Ray, T. S. (1992). Evolution, ecology and optimization of digital organisms. Technicall Report, Santa Fe Institute 92-08-042.

Robinson, J. (1992). Logic and logic programming. *Comm. ACM*, 35:40–65.

Sipper, M. (1998). Fifty years of research on self-replication: An overview. *Artificial Life*, 4:237–257.

von Neumann, J. (1966). *Theory of Self-Reproducing Automata, (Edited and completed by A. W. Burks)*. University of Illinois Press, Urbana and London.

An Artificial Life View to the Collatz Problem

Hiroki Sayama

Collective Dynamics of Complex Systems Research Group / Department of Bioengineering
Binghamton University, State University of New York
sayama@binghamton.edu

Extended Abstract

The Collatz problem, also known as the $3x+1$ problem (Lagarias 1985), discusses the behavior of a series that starts with an arbitrary positive integer x_0 and develops according to the following rule:

$$x_{t+1} = \begin{cases} 3x_t + 1 & \text{if } x_t \text{ is odd} \\ x_t/2 & \text{if } x_t \text{ is even} \end{cases}$$

The Collatz conjecture asserts that this series always falls into a $4 \to 2 \to 1$ cycle regardless of x_0, which is believed to be true by many but has defied any formal proof for more than 70 years (Lagarias 2003; Lagarias 2006).

Here I propose a new perspective on the Collatz problem by considering it an ecological process of artificial organisms (1's in bit strings) and studying the spatio-temporal dynamics of their patterns. To make this approach easier, I ignore the second condition of the rule because it only right-shifts bit strings with no influence on their patterns. Ignoring it converts the series into a simpler iterative map with no ifs:

$$x_{t+1} = 3x_t + \mathrm{LSNB}(x_t)$$

Here $\mathrm{LSNB}(x)$ is the Least Significant Nonzero Bit of x (e.g., $\mathrm{LSNB}(172) = \mathrm{LSNB}(\textit{10101100}) = \textit{100} = 4$; *italics* are binary representations).

The above formula can be interpreted in ecological terms. A bit string of x_t represents the population distribution at time t, where 1's are living organisms and 0's are empty sites. $3x_t$ represents the replication of those organisms because it literally replicates each single bit (Fig. 1(a)). This causes leftward growth of the bit string as well as overcrowding of bits whose effects propagate leftward, depending on the carry rule. Also, $\mathrm{LSNB}(x_t)$ represents an external perturbation continuously introduced to the population, which causes extinction of the living organisms residing at the rightmost end, making the non-zero region of the bit string shrink from the right (Fig. 1(b)).

These interpretations suggest that the Collatz problem is about a competition between growth and extinction of the non-zero region in their speeds (Fig. 1(c)). The maximal speed of the leftward growth of the non-zero region is 2 bits/step, which can be sustained only if the population consists of a single 1, while its average speed is approximately $\log_2 3 \approx 1.58$ bits/step. In the meantime, there is no maximum regarding the speed of extinction of the non-zero region from the right. Assuming the equal probability of 0's and 1's in bit patterns, the average speed of extinction is analytically calculated to be 2 bits/step, which was confirmed by computer simulations. This indicates that the extinction from the right is "faster" than the population growth to the left, providing an ecological explanation of why the series always fall into a single-bit cycle.

Note that the above argument is still not a rigorous proof because it assumes stochasticity in bit patterns. The Artificial Life community could also contribute to this problem by attempting to design counter-examples to the conjecture. It may be possible to create, or even evolve, specific bit patterns that are able to "slow down" the extinction by continuously producing "barriers", which might be possible with very large initial conditions.

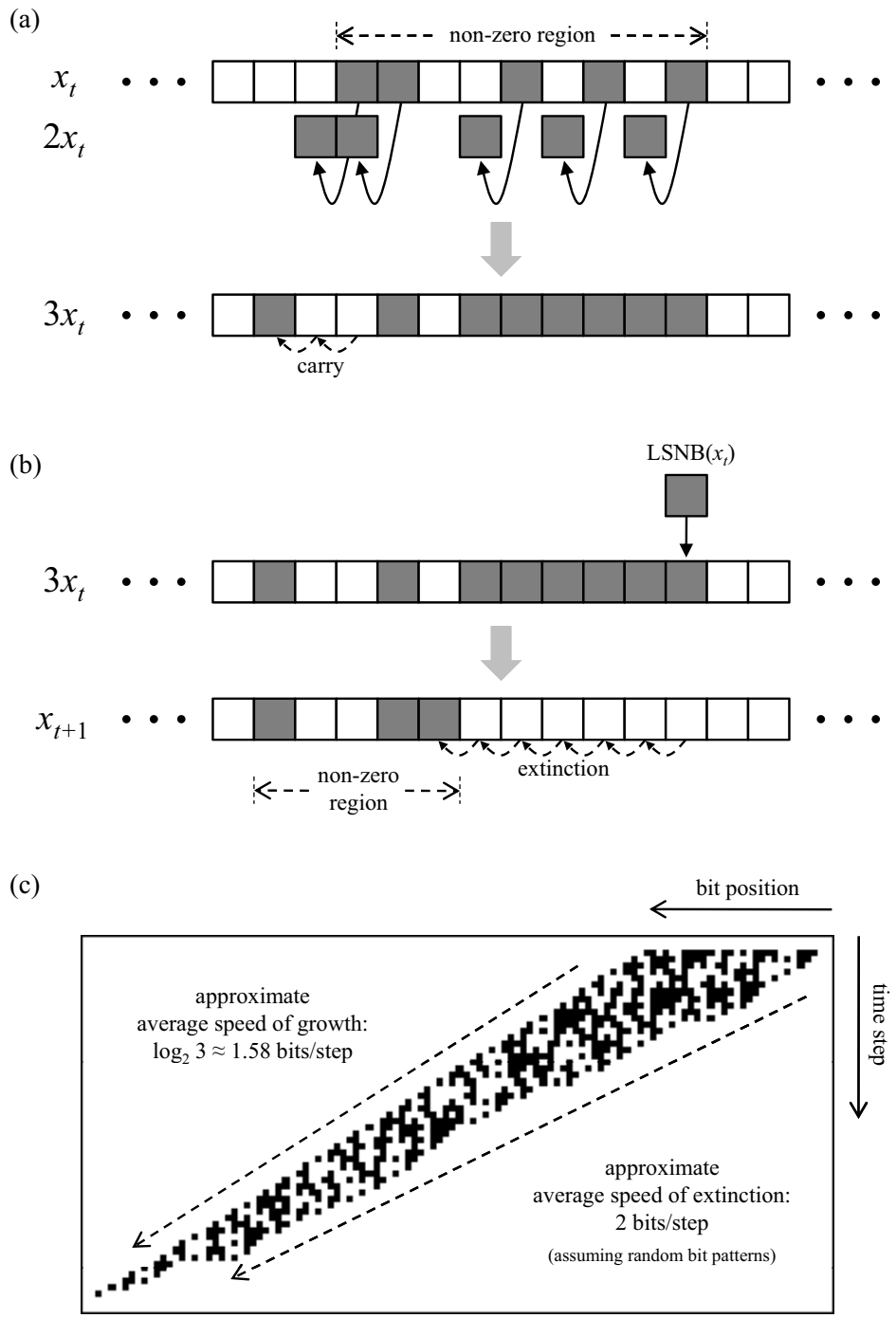


Figure 1: The Collatz problem as an ecological process of artificial organisms represented in bit strings. (a) Replication of 1's (gray cells) and growth of the non-zero region caused by $3x_t$. (b) Extinction from the right caused by LSNB(x_t). (c) Spatio-temporal dynamics of a sample series ($x_0 = 111111111$) visualized as bit patterns.

References

Lagarias, J. C. (1985). The $3x + 1$ problem and its generalizations. *American Mathematical Monthly*, 92(1):3–23.

Lagarias, J. C. (2003). The $3x + 1$ problem: An annotated bibliography (1963–1999). http://arxiv.org/abs/math/0309224.

Lagarias, J. C. (2006). The $3x + 1$ problem: An annotated bibliography, II (2000-). http://arxiv.org/abs/math/0608208.

Search for Computationally Universal Cellular Automata Guided by 1/f Noise

Shigeru Ninagawa

Kanazawa Institute of Technology
ninagawa@infor.kanazawa-it.ac.jp

Abstract

The Game of Life (LIFE) is one of the two-dimensional cellular automata (CA) and has a propagating pattern called “glider”. LIFE is able to emulate a conventional digital computer by considering a glider as a signal in a digital circuit. From the viewpoint of computability theory, LIFE is called computationally universal. Another distinguishing characteristic of LIFE is 1/f noise. The power spectrum calculated from the time evolution of cells starting from a random initial configuration exhibits 1/f characteristics (Ninagawa et al., 1998). Another example of CA exhibiting both computational universality and 1/f noise is found in elementary CA rule 110. Rule 110 was proved to be computationally universal (Cook, 2004) and exhibits 1/f noise (Ninagawa, 2008). These results suggest a relationship between computational universality and 1/f noise in CA. In this study we search two-dimensional three-state nine-neighbor CA rule space for a rule exhibiting a 1/f spectrum by means of genetic algorithms to find computationally universal rules.

The transition function of a CA is encoded into a 134 ternary digit string. Power spectrum is calculated from the discrete Fourier transform of a time series of states of a site and the power is summed up over all cells in the array. The fitness of a rule is given by the exponent estimated by the least squares fitting of the power spectrum divided by the residual sum of squares. The array consists of 100 * 100 sites and periodic boundary conditions are used. The array is started from a random initial configuration. We randomly generate initial rules whose value of lambda parameter is uniformly distributed between 1/135 and 90/135. We observed the evolution for 7200 and 8000 time steps. Since the calculation of the power spectrum needs a lot of computation time, we carry out a preliminary selection from initial rules to remove rules whose spectrum is far from a 1/f spectrum. In the preliminary selection the power spectrum of the evolution for 1024 time steps are calculated and we pick the rules with the exponent of the power spectrum equal to -0.3 or less. The selected rules are gathered as an initial population of 180 rules. 20 rules with the highest fitness are copied without modification to the next generation. The remaining 160 rules for the next generation are formed by uniform crossovers with a probability of 0.6 between pairs in the population chosen by roulette wheel selection. Every bit of the offspring from each crossover is mutated with a probability of 0.03.

Up to now we have performed the experiments for a total of 18789 generations in 80 runs in 7200 time steps and a total of 7881 generations in 100 runs in 8000 time steps. Although the search is in progress, we have found several rules with 1/f spectrum. Some of these rules exhibit stationary, periodic, and propagating patterns which are necessary for supporting universal computation.

This study was supported by a Grant-in-Aid for Scientific Research (C) (20500216) of JSPS and the ISM Cooperative Research Program (2010-ISM-CRP-0006).

References

Journal Article

Ninagawa, S., Yoneda, M., and Hirose, S. (1998). 1/f fluctuation in the Game of Life. *Physica D*, 118:49–52.
Cook, M. (2004). Universality in elementary cellular automata. *Complex Systems*, 15:1–40.
Ninagawa, S. (2008). Power spectral analysis of elementary cellular automata. *Complex Systems*, 17:399–411.

A Face-Encoding Grammar for the Generation of Tetrahedral-Mesh Soft Bodies

John Rieffel[1] and Schuyler Smith[1]

[1]Union College, Schenectady, NY 12308
rieffelj@union.edu

Abstract

Many of the most profound works of artificial life have emerged through the composition of physical simulation and generative representations. And yet, while physics engines are becoming more realistic, and generative representations are growing more powerful, they are still predominantly used to simulate *rigid* objects. The natural world and its organisms are, by contrast, *soft*, and full of much more interesting (and complex) interactions than those which can be faithfully reproduced by rigid body dynamics. In this work we describe and implement a grammatical encoding capable of generating large, complex, and multi-resolution *soft structures* which can be natively simulated by the state-of-the-art hardware-accelerated physics engines. The structures generated by the encoding exhibit all the benefits (structural modularity, large-scale co-ordinated change) of more conventional rigid-body generative encodings.

Introduction

The generative encoding of morphology embedded within physical simulation has a long and rich history in artificial life, tracing back to Karl Sim's seminal work on evolved virtual creatures (1994) and to Lindenmayer and Prusinkiewicz's L-system-based plants (1990). More recent notable contributions include the evolution of satellite antennae (Lohn et al., 2005), robots (Pollack et al., 2001), and tensegrity structures (Rieffel et al., 2009).

A unifying property of these contributions is that they all produce *rigid* objects. This is largely due to the limitations imposed by popular off-the-shelf physics engines, such as the Open Dynamics Engine (ODE) which, although capable of smoothly simulating the interaction of thousands of rigid bodies , lack the ability to effectively simulate softer materials such as cloth or rubber. Finite Element Analysis (FEA) and Computational Fluid Dynamics (CFD), are incredibly accurate, but too computationally intensive to be practical for Artificial Life purposes.

Of course, most biological organisms are quite soft, and the complex dynamical interactions which arise from this softness are beyond what can be realistically reproduced by simpler rigid body dynamics. Recently, off-the-shelf hardware-accelerated physics engines, such as NVidia's PhysX, have added to ability to simulate soft shapes, opening the door to a much more dynamic range of virtual creatures.

Taking full advantage of this functionality, however, requires a grammatical encoding capable of generating large, open-ended, and incredibly complex soft structures. In this paper we introduce a face-encoding grammar which operates upon tetrahedral meshes like the one shown in Figure 1. Meshes such as these are used to describe deformable objects in computational methods such as FEA, as well as in physics engines such as PhysX. By operating directly within the representational substrate of soft bodies (avoiding post-hoc methods such as generating a more generic CAD file and then computing a near-matching mesh) we avoid design bias and have a more nuanced control over the final product.

As we show, the face-encoding grammar we introduce is able to generate arbitrarily large, and incredibly complex tetrahedral meshes. Furthermore, like other grammatical encodings, our process exhibits implicit modularity and allows small changes in the underlying grammar to produce large-scale co-ordinated changes in the final product. The results of this paper open the door to a whole new dimension of the artificial life: *soft* virtual creatures, and *soft* robots.

Generative Encodings

Generative encodings come in a variety of styles: Artificial Ontogeny (Bongard and Pfeifer, 2003), Generative and Developmental Systems (Stanley, 2008), and Lindenmeyer Systems (L-Systems) (Prusinkiewicz and Lindenmayer, 1990)(to name a few), but all have a common set of features, and all offer a variety of advantages. Using the the biological processes of growth and development as inspiration, generative encodings grow large complex objects by applying a simple set of re-write rules to an initial "seed". In the case of L-Systems, the seed is a small starting string of characters, grammatical production rules determine the order of growth. Gene Regulatory Networks Bongard and Pfeifer (2003) model the interaction between transcription factors and gene expression, and can be used to grow both morphologies and neural networks.

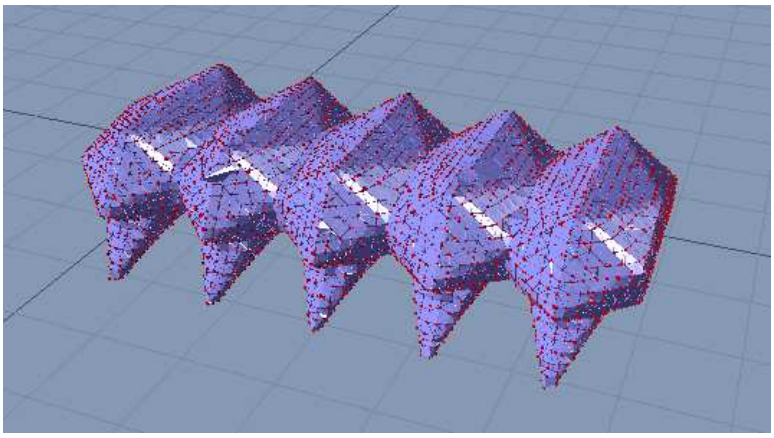


Figure 1: A large soft robot within the PhysX physics engine. Soft bodies are represented as tetrahedral meshes. This particular mesh was created in a top-down fashion: hand-designed by an engineer in CAD, and then manually converted into a tetrahedral mesh. This paper describes an alternative bottom-up approach: a grammar for automatically generating arbitrarily large and complex structured tetrahedral meshes.

Regardless of implementation, the benefits of generative representations, particularly in the context of Genetic Algorithms, stem from their ability to implicitly encode structural modularity and reuse, and the ability for small changes to the rule set to produce corresponding large-scale co-ordinated changes in the final result (Hornby and Pollack, 2001). As an example, when representing a table, unlike a direct encoding, a generative encoding is able to change the length of all four legs simultaneously.

Generative encodings are particularly popular in evolutionary design tasks, in which they are used to specify the structure (morphology) of objects and creatures. Karl Sims' early work (1994) on artificial life used a simple grammar to grow virtual creatures within a simulated environment, Lohn *et al* used L-Systems to design the satellite antennae (2005) and Hornby used a variety of L-System to develop the morphology of virtual robots (2001).

Physics Simulation

Generative encodings of morphology really come to life when they are embedded within realistic physical simulations. Karl Sim's virtual creatures were evaluated within a simple but quite effective quasi-static physical simulator (1994). Later work, such as Lipson's GOLEMs (2001) and Hornby's GenoBots (2003) also involved quasi-statics -static simulations. More recently, the advent of off-the-shelf physics engines such as the Open Dynamics Engine (ODE), has led to more dynamical simulations, such as Bongard's virtual creatures (2003) and Rieffel's tensegrity robots (2010).

Conventionally, the only means of simulating the dynamic behavior of *soft* objects was through computationally intensive tools such as Finite Element Analysis (FEA) and Computational Fluid Dynamics (CFD). While these methods are quite powerful, they are computationally intensive, and operate on small enough time scales (usually simulating only seconds at a time) as to make them impractical for common Artificial Life techniques such as evolutionary algorithms. Recently, however, following in the footsteps of modern advances in computer graphics (Jakobsen, 2001), commercial video-game physics engines, such as Intel's Havok, and NVidia's PhysX, have added the ability to simulate cloth as well as three-dimensional soft bodies. What makes these engines particularly appealing to the artificial life community is their ability to use General Purpose Computing on Graphics Processing Units (GPGPU) interfaces in order to achieve significant hardware acceleration of simulations – providing speedups of several orders of magnitude (Banzhaf and Harding, 2009).

A way of grammatically generating soft morphologies and testing them in simulation would be a valuable tool for further exploring these issues. The remainder of this paper describes one such implementation.

A Face-Encoding Grammar for Tetrahedral Meshes

Central to our approach is the use of tetrahedral meshes to represent soft bodies. While our examples below are within the context of NVidia's PhysX simulator, it is worth emphasizing that tetrahedra meshes are commonly used in other systems as well, such as Finite Element Analysis.

Figure 2 illustrates a single tetrahedron. The "softness" of a material within PhysX can be changed by varying a set of constraints placed upon the tetrahedron. The first constraint treats each edge of the tetrahedron as a spring-and-damper system, which resists both stretching and compression. A second constraint attempts to maintain each tetrahedra at

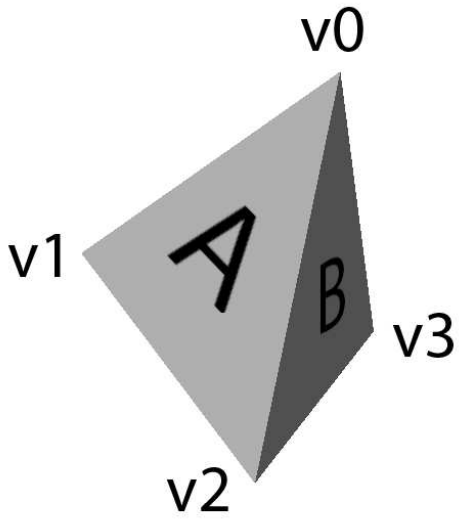


Figure 2: Soft bodies in PhysX are built out of tetrahedral meshes. Each tetrahedron is defined by four vertices and four corresponding faces

a constant volume. Changing the value these parameters changes the softness of the tetrahedron. These tetrahedra are then woven into a larger "mesh", in which neighboring tetrahedra are connected at their common vertices. By uniformly varying the parameters of the tetrahedral mesh, PhysX can simulate a wide range of soft materials, from rubbery Jell-O to semi-rigid plastics.

Since there are no known grammatical encodings which operate upon tetrahedral meshes, we will create our own. We use as inspiration the Map L-Systems, a special form of L-system whose rewrite rules operate upon the edges of 2-D graphs (Luke and Spector, 1996). Map L-Systems have been used to grow both 3-D surfaces(Hemberg and O'Reilly, 2004) and large tensegrity structures Rieffel et al. (2009).

Drawing an analogy between the edges of a graph (in 2-D) and the faces of a tetrahedron (3-D) our *face-encoding* grammar operates upon tetrahedral faces in much the same way that a Map L-system operates upon graph edges.

Assuming that each face of a tetrahedron can be given a label, there are three obvious operations which you can perform upon the faces of a tetrahedron, as illustrated in Figure 3. We will assume that operators can only be applied to *exposed* faces – that is, those which are not shared by two tetrahedra.

$A \rightarrow relabel(B)$ will replace a face labeled 'A' with a new face labeled 'B'

$A \rightarrow grow\{BCD\}$ replaces a face labeled 'A' with a new tetrahedron, labeling the new exposed faces as 'B','C', and 'D'.

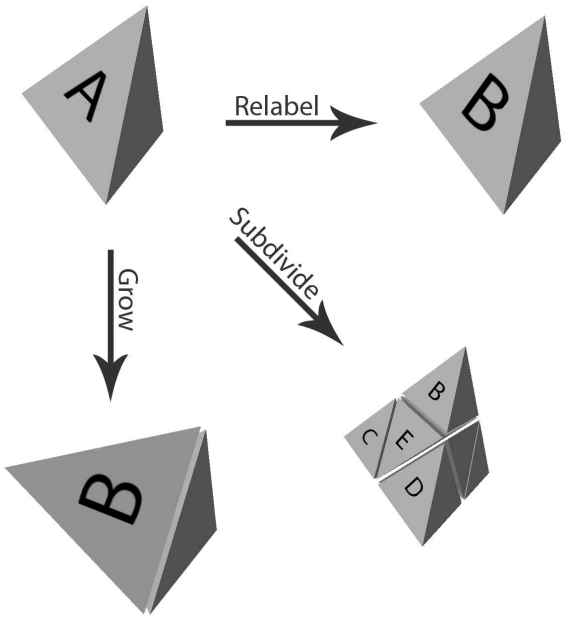


Figure 3: An illustration of the three rules which can be applied to the face of a tetrahedron. Clockwise from top left: the original tetrahedron with face labeled "A", **relabel** replaces "A" with "B", **subdivide** replaces the face with four smaller faces (this requires subdividing the entire tetrahedron), and **grow** adds a new tetrahedron with face labels "B","C","D"

$A \rightarrow divide[BCDE]$ subdivides a face 'A' into four smaller faces, 'B','C','D', and 'E'. The underlying tetrahedron must also be subdivided into eight component tetrahedra in to provide attachment points for the new faces and vertices.

Armed with these rules, we can now grow tetrahedral meshes of arbitrary size by iteratively applying them to an initial "seed" tetrahedron.

Each exposed face of the soft body kept in a queue, and is associated with three vertices (in counterclockwise order so that we can calculate surface normals) and exactly one tetrahedron. (A face can be shared by two tetrahedra, but then it wouldn't be exposed). For every generation of growth, the open faces are iteratively removed from the queue and the appropriate rule is applied. For $relabel$, a new face with the new label is enqueued. For $grow$ and $divide$, new vertices and tetrahedra are computed and added, and then the resulting three ($grow$) or four ($subdivided$) new faces are enqueued.

This entire cycle is repeated a fixed number of times to create progressively larger and more complex soft bodies. Figure 4 shows the iterative application of rewrite rules to

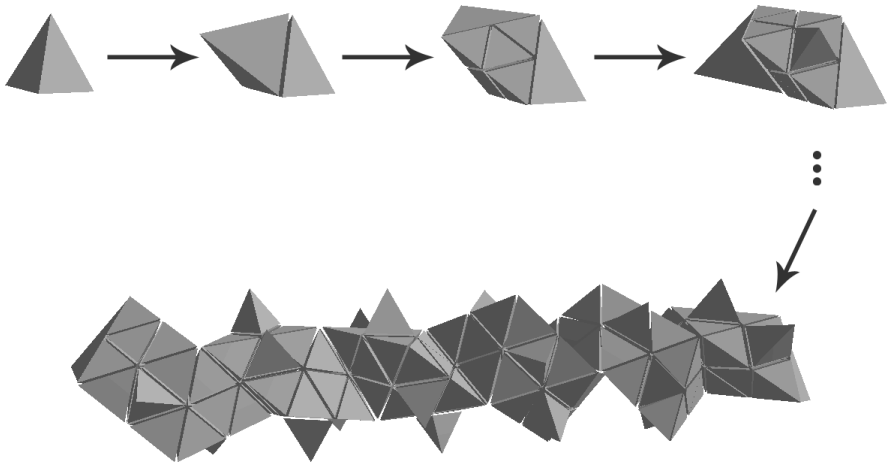


Figure 4: The growth of a larger tetrahedral mesh by iteratively applying a face-encoding grammar to an initial "seed" tetrahedron.

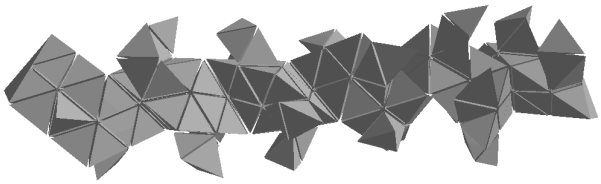


Figure 5: A small change in the grammar underlying the production of a tetrahedral mesh can produce profound and co-ordinatd change in the final result. The above figure was produced with a single mutation to the grammar which produced the mesh in Figure 4.

a single starting tetrahedron. Like in other grammatical encodings, a small change in a single production rule can have profound and co-ordinated effects upon the final product.

Technical Challenges

Although the rules may seem simple, there are several technicalities which may the implementation of a face encoding grammar difficult. First, as previously mentioned, when subdividing faces we also subdivide the associated tetrahedron. This is necessary because the new, smaller faces need new vertices and their own tetrahedra to attach to. While in principle it may be possible to subdivide less than the entire tetrahedron, during a divide, it requires more complicated bookkeeping, and the symmetry of our solution is appealing.

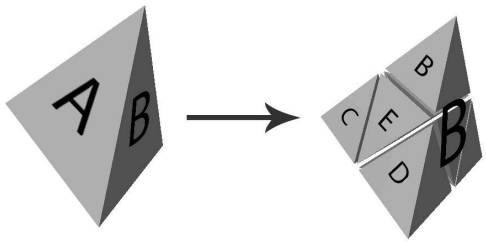


Figure 6: Subdividing a face "A" on the left hand tetrahedra actually requires splitting the entire tetrahedron. The remaining faces, such as "B" remain defined in terms of their original three vertices, and are left alone. Book-keeping must be maintained to ensure that any subsequent call to divide the original face "B" is aware that the underlying tetrahedron has already been split.

However, subdividing the full tetrahedron when a single face is divided raises the question of how to treat the remaining faces. In principle, we want to act as if the remaining faces still exist and are still associated with the original tetrahedron, even though the tetrahedron they belonged has been subdivided into smaller tetrahedra, as illustrated by Figure 6. This works fine as long as the other faces want to grow or relabel – they can proceed as usual, because to do either of those things doesn't rely on the underlying tetrahedron. A special case arises if a second original face wants to subdivide, in order to ensure the work isn't duplicated.

A similar scenario occurs when we want to subdivide two adjacent tetrahedra. Before subdivision they are connected only at their corners, but after subdivision they should be connected in the middle of each of the edges they share as well, where the smaller, subdivided tetrahedra are now adjacent to each other. We solve this problem and other similar special cases largely by ignoring them during growth, and then removing duplicate and redundant vertices during a post processing stage.

Examples

As an example of the complexity of features achievable with this grammatical encoding, consider the rule set shown in Table 1, and the soft body which results by iterating the grammar over a single "seed" tetrahedron 10 times, as shown in Figure 7.

A	$\rightarrow$	grow	$\{DBF\}$
B	$\rightarrow$	grow	$\{ADF\}$
C	$\rightarrow$	grow	$\{EDF\}$
D	$\rightarrow$	relabel	(D)
E	$\rightarrow$	grow	$\{DCF\}$
F	$\rightarrow$	divide	$[DDDG]$
G	$\rightarrow$	grow	$\{DDG\}$

Table 1: The rule set used to grow Figure 7.

At each iteration, faces labeled with an A, B, C, E, or G are grown, and the three new faces created are labeled as shown. For example, the faces of the tetrahedron grown from any A face will be labeled as D, B, and F faces. Face D, meanwhile, is relabeled as itself. This serves effectively as a no-op, and is a dead-end for growth. Face F is subdivided,into three dead-end D faces and one G face. In the final soft body, faces A, B, C, and E work together to grow the "legs" of the soft body, while the F and G faces work together to grow the smaller "tentacles" that protrude at every angle.

Figure 8 illustrates how further iterating the grammar in Table 1 20 times produces a structure which can be considered an elaboration of the smaller 10-step mesh of Figure 7

Discussion: Applications to Soft Robotics

Soft bodies, both natural and virtual, bring with them fascinating new questions about the relationship between morphology and control. Soft and deformable objects can possess near-infinite degrees of freedom, and elasticity in the system means that local perturbations can propagate to distal regions with interesting consequences. One might be inclined to think that this would create intractable control challenges, and yet the animal kingdom is full of soft and deformable animals. The *Manduca sexta* caterpillar, for instance, which might seem a relatively simple organism, is in fact rife with non-linearities and complex dynamics imposed by the interaction of hydrostatics, an elastic body wall, and nonlinear muscular behavior. New insights from biomechanics and neuro-ethology (Trimmer, 2007) suggest that rather than being hobbled by these complex dynamics, soft creatures in fact are able to exploit them as an advantage, via a form*morphological computation*(Valero-Cuevas et al., 2007; Pfeifer and Bongard, 2006).

This is particularly relevant to the budding field of soft robotics. Imagine a machine that can squeeze through holes, climb up walls, and flow around obstacles. Though it may sound like science fiction, thanks to modern advances in materials such as polymers (Huang et al., 2007), and nanocomposites (Capadona et al., 2008) such a "soft robot" is becoming an increasing possibility.

The largest outstanding problem in soft robotics is that while we possess the means to build them, no principled method exists to design or control them. There are no textbooks on soft robot design and control. And, while intuition suggests that the best way to control soft structures is, like caterpillars, to exploit their complex body dynamics via *morphological computation*, the dynamics are too complex to hand-code a solution.

The most promising approach is probably body-brain co-evolution (Pollack et al., 1999). The grammatical encoding we have presented here is a vital tool for the the co-evolution soft robotic design and control.

Conclusion

The face-encoding grammar presented in this paper provides us with a principled way of generating large and complex *structured* soft objects. The ability to generate complex and life-like soft structures (via this face encoding grammar) and to efficiently simulate them (via hardware-accelerated physics simulators) broadens the horizons of artificial life research, and provides entire new sources of bio-inspiration. Instead of mimicking (relatively) rigid vertebrates such dogs and horses, we can now begin to create artificial creatures which resemble octopii, squid, slugs and caterpillars.

References

Banzhaf, W. and Harding, S. (2009). Accelerating evolutionary computation with graphics processing units. In *GECCO '09: Proceedings of the 11th annual conference companion on Genetic and evolutionary computation conference*, pages 3237–3286, New York, NY, USA. ACM.

Bongard, J. and Pfeifer, R. (2003). *Morpho-functional Machines: The New Species (Designing Embodied Intelligence)*, chapter Evolving complete agents using artificial ontogeny, pages 237–258. Springer-Verlag, Berlin.

Capadona, J., Shanmuganathan, K., Tyler, D. J., and Rowan, S. (2008). Stimuli-responsive polymer nanocomposites inspired by the sea cucumber dermis. *Science*, 319(7).

Hemberg, M. and O'Reilly, U.-M. (2004). Extending grammatical evolution to evolve digital surfaces with genr8. In *EuroGP*.

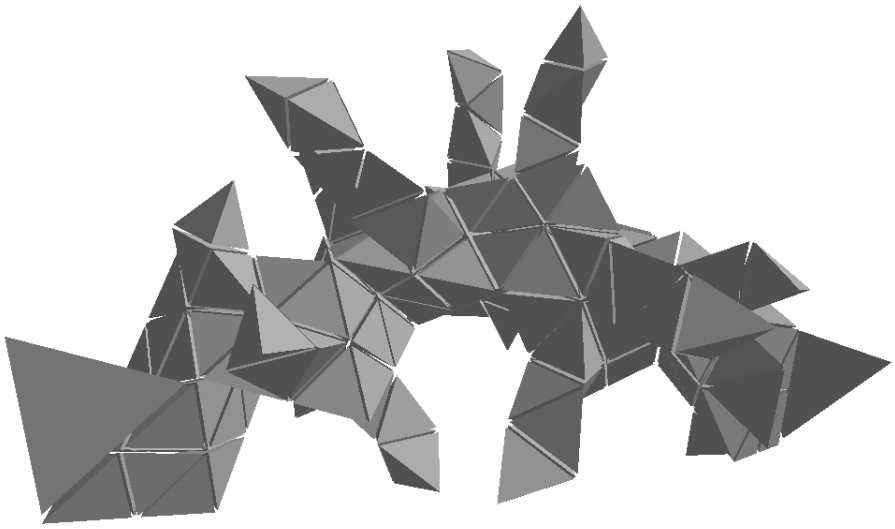


Figure 7: A larger grammatically-produced tetrahedral structure which shows several desirable features, most notably modular structure and varied tetrahedral resolution. This particular mesh was created by iterating the grammar in Table 1 10 times.

Hornby, G. S., Lipson, H., and Pollack, J. (2003). Generative encodings for the automated design of modular physical robots. *IEEE Transactions on Robotics and Automation*, 19(4):703–719.

Hornby, G. S. and Pollack, J. B. (2001). The advantages of generative grammatical encodings for physical design. In *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*, pages 600–607, COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea. IEEE Press.

Huang, J., Foo, C. W. P., and Kaplan, D. (2007). Biosynthesis and applications of silk-like and collagen-like proteins. *Polymer Reviews*.

Jakobsen, T. (2001). Advanced character physics. In *Game Developer's Conference 2001*.

Lohn, J. D., Hornby, G. S., and Linden, D. S. (2005). An Evolved Antenna for Deployment on NASA's Space Technology 5 Mission. In O'Reilly, U.-M., Riolo, R. L., Yu, T., and Worzel, B., editors, *Genetic Programming Theory and Practice II*. Kluwer.

Luke, S. and Spector, L. (1996). Evolving graphs and networks with edge encoding: Preliminary report. In *Late Breaking Papers at the Genetic Programming 1996 Conference*, pages 117–124.

Pfeifer, R. and Bongard, J. C. (2006). *How the Body Shapes the Way We Think: A New View of Intelligence (Bradford Books)*. The MIT Press.

Pollack, J., Lipson, H., Funes, P., Ficici, S., and Hornby, G. (1999). Coevolutionary robotics. In *EH '99: Proceedings of the 1st NASA/DOD workshop on Evolvable Hardware*, page 208, Washington, DC, USA. IEEE Computer Society.

Pollack, J. B., Lipson, H., Hornby, G., and Funes, P. (2001). Three generations of automatically designed robots. *Artificial Life*, 7(3):215–223.

Prusinkiewicz, P. and Lindenmayer, A. (1990). *The Algorithmic Beauty of Plants*. Springer–Verlag, New York, USA.

Rieffel, J., Valero-Cuevas, F., and Lipson, H. (2009). Automated discovery and optimization of large irregular tensegrity structures. *Computers & Structures*, 87(5-6):368 – 379.

Rieffel, J. A., Valero-Cuevas, F. J., and Lipson, H. (2010). Morphological communication: exploiting coupled dynamics in a complex mechanical structure to achieve locomotion. *Journal of The Royal Society Interface*, 7(45):613–621.

Sims, K. (1994). Evolving 3d morphology and behavior by competition. In Brooks, R. and Maes, P., editors, *Artificial Life IV Proceedings*, pages 28–39. MIT Press.

Stanley, K. O. (2008). Generative and developmental systems. In *GECCO '08: Proceedings of the 2008 GECCO conference companion on Genetic and evolutionary computation*, pages 2849–2864, New York, NY, USA. ACM.

Trimmer, B. (2007). New challengees in biorobotics: incorporating soft tissue into control systems. In *IEEE International Conference on Robotics and Automation*.

Valero-Cuevas, F., Yi, J., Brown, D., McNamara, R., Paul, C., and Lipson, H. (2007). The tendon network of the fingers performs anatomical computation at a macroscopic scale. *IEEE Trans Biomed Eng.*, 54:1161–6.

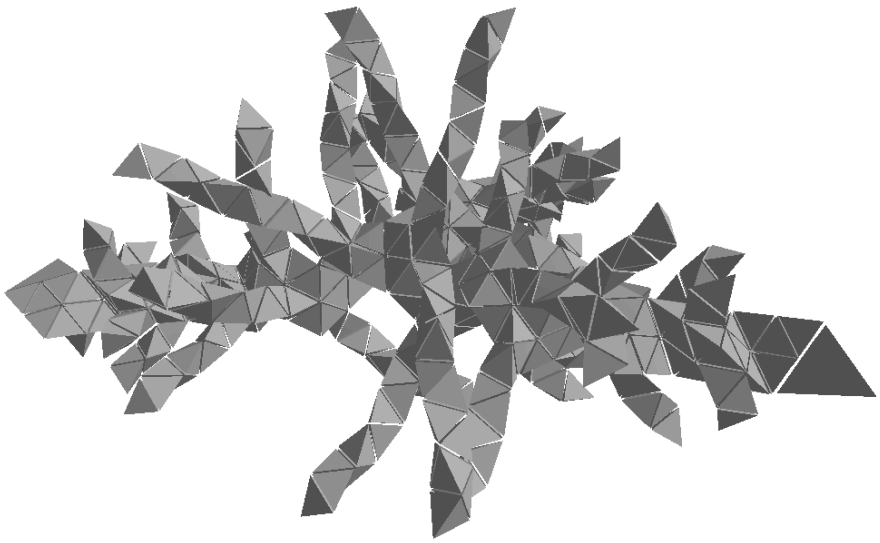


Figure 8: Continuing the growth of the grammar in Table 1 for another 10 cycles produces a mesh which is more elaborate that the earlier one in Figure 7, but which maintains much of the coarse structure.

Atlas of Patterns from One-rule Firing Cellular Automata

J.K. Shin

School of ME, Yeungnam University, South Korea
jkshin@yu.ac.kr

Abstract

Growing cellular automata (CA) can generate diverse patterns such as flowers and snow crystals, especially when one-rule firing scheme is employed. In the present paper, hexagonal CAs on 2D plane is used to investigate the patterns generated under the one-rule firing scheme. Rules define the state of an empty cell in the next time-step as a function of the present states of its six neighborhood cells. Among the empty cells currently subject to the update condition, only one-rule is fired in a given time step, which causes differential growth of the patterns and, as a result, the emergence of many interesting patterns. An efficient method to identify the rule-set, or equivalently the patterns, is presented, which is just a list of the *Fired* color *codes*(F-codes) to make the specific pattern. Numerical simulation showed that such patterns can have F-codes of various lengths, ranging from one to a few hundreds. When we imagine the F-code as a genetic code for the pattern generated, the F-code system suggests an ecological system composed of a complete atlas of species. It will be interesting to investigate the complexity of the species, considering the length of the F-codes as a measure of complexity. For example, consider the complexity of patterns generated in random situation. It is found that the number of possible F-codes for a given length increases with the length. On the other hand, patterns with longer F-codes are the less likely to be obtained in random simulations. Why is the complex patterns rare than the simpler one when the former can have the larger number of variations? The present paper tries to answer this question theoretically.

Introduction

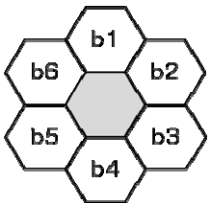
Cellular automata (CA) have been widely used in the study of physical, biological and social systems[Wolfram 1984]. Recently the author presented a new firing scheme on 2D cellular automata[Shin 2010]. In this new firing scheme, only one rule is fired at a given iteration. A system composed of 2D hexagonal cells is used to study growing patterns from a single seed cell. A one-rule firing scheme is found to generate myriad of patterns not reported in the CA literatures. In addition to the simple geometric patterns, the natural patterns such as snow flakes and flower-like ones emerged depending on the rule sets used. An efficient method of identifying the patterns, called an F-code, is suggested. Being composed of the rule values fired for generating the specific pattern, the F-code is decodable. Patterns were identified to have F-codes of length a few to a few hundreds. The length of the F-code suggested a natural measure of the complexity of patterns. Because the F-code looks like a genetic code and each of the F-codes corresponds to a pattern in two dimensional space, the F-code system suggests a complete set of an artificial ecology composed of almost an infinite number of species of varying complexities. During numerical study, it were found that the complex patterns with longer F-codes are the less likely to be found under random simulations, while the number of possible patterns increases with the length. This seems to be a contradiction. Why is the complex patterns rare than the simpler one when the former can have the larger number of variations? The present paper tries to answer this question theoretically.

One-rule firing cellular automata

CA system in the present study is composed of hexagonal array of cells on 2-D plane. An occupied cell has values or color codes from 1 to m and is called an *element*. For convenience, an *empty cell* is defined to have color code of 0. Thus a cell can be in any of the M=m+1 possible cell states. Only six-color problem with m=6 and M=7 is considered in the present study, unless stated otherwise. Once defined, an element does not change its value nor return to empty cell. A cell has a set of neighborhood cells composed of six nearest cells. An empty cell is called a *surface cell* if it has at least one element in its neighborhood. A *rule* determines the value(or color code) of a surface cell at the next time step as a function of the states of the neighboring cells at the present time step.

The number of neighborhood states possible is M^6 and the number of rule sets, without any symmetry condition, will be M^{M^6} . Throughout the present study, symmetric rule sets are considered. Thus two neighborhood conditions that are equivalent under cyclic rotation are equivalent. In typical CAs with synchronous updating, the rules are applied to *all* of the candidate cells at the given time step. In the present paper, only one rule is fired in a time step. Among the many different ways to choose a single rule to be fired in each step, 'the last nonzero rule firing' scheme is discussed in the present study. This will be explained below. Numbering of elements and surface cells are important for a standard implementation of the firing scheme. Elements are numbered in the order of their birth. The numbering of the neighborhood cells on an element is always starts from the top and counted clock-wise. The surface cells are numbered element-wise first and then neighborhood-wise(See Fig. 1).

To explain key concepts of the present paper, let us start with a single seed element at time step 0 (See Fig. 2). Without loss of generality, the color of the first element, e1, is set to be 1. Now the first element has six neighbors. Each of the six neighbor cells has the same neighborhood state of 000001. Assume, for example, the rule value is rule(000001)=3. Applying the rule to the six surface cells will end the iteration 1. Even though the updating is synchronous, we should consider the order of the updating, as it determines the element number. Thus rules should be applied from the surface cell number 1, or from the lower numbered surface cells in general. After iteration 1 we reach at Fig. 2. Elements are denoted by shaded cells whose numbers are marked at the centers of the cells. They are numbered in the order of their birth. Small numbers in each of the element cells represent the colors of the elements. The color(=3) of e2 to e7 was determined from the application of the rule number 000001. After the first iteration, the system has total of 12 surface cells , s1 to s12, as shown in the Fig. 2. The surface cells are numbered element-wise first and then neighborhood-wise. The numbering of the surface cell is scratched after each of the iterations and restarts in every iteration. On the contrary, for the elements, the numbers once defined are maintained throughout the later time steps.



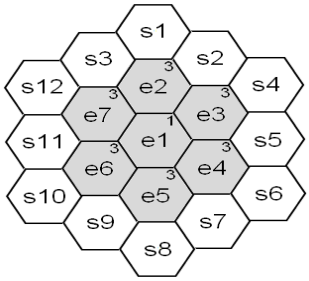


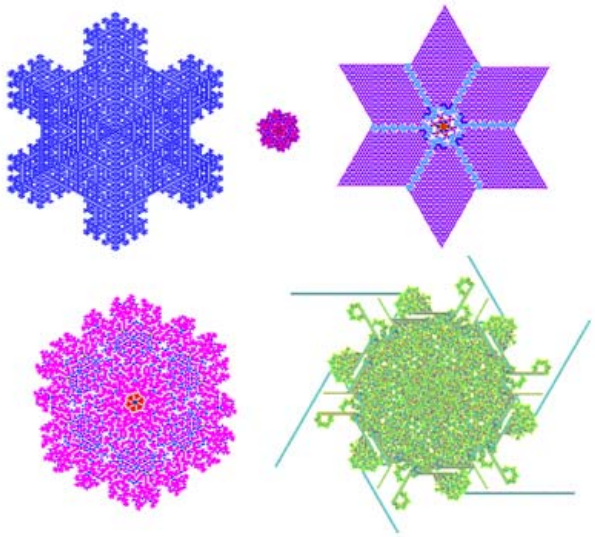
Figure 1. Six neighbors of an element.(Left)
Figure 2. Numbering of elements and surface cells, rule(000001)=3 is applied.

After time step 1, we have twelve surface cells, s1 to s 12, as shown in Fig.2. Depending on the neighborhood conditions, the twelve surface cells can be sorted into two groups. The neighborhood condition of surface cells s0, s1, s4, s6, s8, and s12 is equivalent to 000003. For the remaining surface cells, it is 000033. Among the two rules, we fire only the rules for the last surface cell which is s12 in this case. Assuming rule(000003)=4, the six surface cells s0,s1, etc will be occupied by an element of color code 4. On the contrary, surface cells s2, s3, s5, s7, s9 and s11 will still remain as empty cells even after the iteration 2. Before starting the third iteration, we should renumber the surface cells from the scratch, while the numbering of the elements should be inherited from the previous iteration. The efforts we have to pay for keeping track of the numberings of elements and surface cells got rewarded by the efficient coding scheme of the patterns. The patterns generated can be completely defined form the sequence of the fired codes. For example, two codes were fired up to two iterations. The rule values or fire codes were 3 and 4 respectively. Thus a code f=34 is enough to define the pattern. This code is called an F-code, for convenience. There's no need to remember the rule number such as 000003 or 000033, etc. The rule number is embedded in the pattern itself. If a rule is already fired in earlier time step(s) and appears again to be fired in later time step(s), it does not enter into the F-code again.

The F-code is very efficient to define patterns generated. There remains one thing to be treated. What if the code(rule value) for the last surface cell is 0. For example, assume rule(000003)=0 in the above example. When the rule for the last surface cell is 0, then we chose next to the last surface cell to be fired. In the above example, assume rule(000033)=6. Then we fire this rule on the surface cells s2, s3, etc. In this case the f code looks like f=306. Observe that the 0 is inserted to remember that the rule for the 0 value has been skipped.

Atlas of patterns

In Fig.3, a few example patterns are shown with corresponding F-codes. As iteration proceeds, the patterns grow and the length of the F-codes can increase as new rules appear to be applied. For this reason, the patterns and the F-codes should be described with the number of iterations at the same time. The simplest pattern of code F=1 is the well known Packard's snowflake as shown in Fig. 3(a) [Levy 1992]. To generate this pattern only one rule is necessary. At least in principle, the pattern can grow infinitely if we continue the iteration. But in some cases the patterns stop growing at some iteration as shown in Fig. 3(b). This happens when all the rules for the surface cells have value of 0 at the same time. Figure 3(c) shows a geometric pattern which has a relatively short F-code. The F-codes for the first two patterns are finished in the sense that they do not grow because the patterns is dead at some iteration or same rules apply infinitely. A finished F-code is represented by an uppercase letter F, while that for the unfinished is by lowercase letter f. Thus the code f=24036066053444252006l5 for Fig. 3(d) means that the f-code is not finished up to the iteration 150. If the iteration is continued, the f-code increases. Because every F-code must have finite length, the unfinished F-codes happen because we stopped the iteration at a certain number. Due to the computation time, we cannot continue the iteration long enough to identify an F-code to its full length in many cases. Furthermore, except for some cases, we cannot prove that the F-code is finished or unfinished at the present iteration. But in some case, we can prove that the F-code is finished at the present time step. For example, the F-code shown in Fig. 3(e) is a finished one. In actual, the complex pattern in Fig.3 (e) shows a most frequent mechanism by which a pattern stops growing in its F-code. This pattern has a finished F-code of length $|F| = 304$. If iteration continues from the point shown in this figure, only the outermost lines grow out. Because this growing process is not interrupt by anything, it will repeat endlessly while keeping the length of F-code at the present value. In general, the finished F-codes happen when the rules apply periodically, as in the case of Fig. 3(e). If we cannot prove that an F-code is finished or not at the present iteration, we treat that it is unfinished.



(a) Packard's snowflake. F=1 (iter=60)(upper left).
(b) Pattern dying at iteration 80(upper middle).
F=320110132021010133031…2101123103023(42 digts)
(c) Geometric pattern. f=606302400540232063452002(24 digits, iter=390).
(d) Flower like pattern. f=240360660534442520061 5(22 digits, iter=150). (lower left)
(e) Complex pattern. F=3603606…3625626(304 digits, iter=3900)(lower right)

Figure 3 Sample patterns.

Table 1 Number of patterns with given length of F-codes up to iteration 30.

$\|F\|$	No. of patterns (a)	Early death	Total enumerated (=$7^{\|F\|}$)(b)	Prob.=a/b
1	1	0	7	0.142857
2	10	0	49	0.204082
3	70	5	343	0.204082
4	305	25	2,401	0.12703
5	875	45	16,807	0.052062
6	4,115	560	117,649	0.034977
7	22,360	240	823,543	0.027151
8	121,350	1,825	5,764,801	0.02105
9	579,745	10,920	40,353,607	0.014367
10	1,461,880	153,720	282,475,249	0.005175
Sum	2,190,711			0.832832

In search of life-like properties in CA, self-replicating shapes are widely studied in the literature[Neumann 1966, Langton 1984, Sayama 1999, Wuensche 2004, Pan and Regia 2010]. Self-replication is also frequently found in the present study. Figure 4 shows a pattern growing through self-replicating loops. This kind of repeating pattern frequently happen for complex shapes characterized by its long F-codes, say $|f|>30$, based on an iteration number of 60. These numbers are not meant to be precise. To identify these numbers more rigorously, we need more study. The self-replication patterns from the present model suggest a hypothesis, that complex patterns are composed of hierarchy of simple repeating substructures. Classifying the patterns depending on the periodicity of the applied rules, on the existence of hierarchical structures as well as on the length of their F-codes will be also a topic of a future study.

It is an interesting question to ask how many patterns can there be at a given length of F-codes. This question can be answered, at least for relatively short lengths of the F-codes, by numerical search. The result is shown in Table 1. To obtain this table, all the possible F-codes of up to length 10 are generated and decoded exhaustively to see if the patterns exist for each of the given F-codes. As a result, we have 1,461,880 of patterns of $|F|$=10 up to iteration 30. As can be seen from the last row of Table 1, about 83% among all the possible combination of f-codes of length 10 did not need additional rules, or codes. But for the remaining 17%, additional rules (and codes) are required to finish the iteration 30. For example, we find f-Codes of f=243562678000(12 digits) up to iteration 30. Then the first 10 digits of this code f=2435626780 is counted among the 17% discussed above. Table 1 shows a general tendency that the number of patterns increases with the increase in $|F|$. An important thing to be noted is that this table is compiled from the data obtained through a simulation of up to iteration 30. The specific numbers may change under different setting of the iteration number.

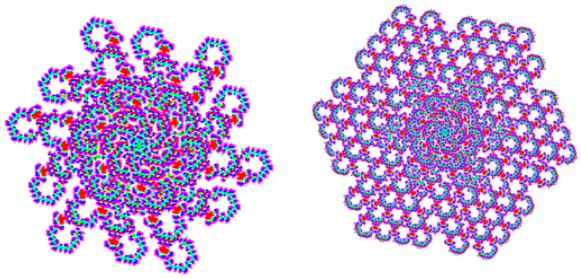


Fig. 4 A complex pattern composed of self replicating loops(shown in different scales).
Left : f=350101400…51465(88 digits, iter=481)
Right : f=350101400…514652000(92 digits, iter=780)

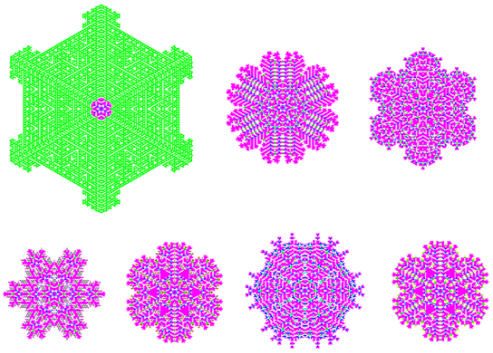


Fig. 5 Seven patterns having F-codes starting with F=2024155. Depending on the 8-th F-code, the patterns vary diversely. This looks like a speciation through mutation of the genetic codes. All patterns are up to iteration 90. F-codes and lengths are from top left to right: X5($|F|$=8), X0(13), X1(9), X2 (12), X3(19), X4(25),X6(19) with X=2024155.

The F-codes can be imagined as representing genetic codes for the patterns implied. For example, Fig. 5 shows patterns whose first 7 codes are the same in their F-codes, represented by an X(=2014155). The change, or mutation, in 8-the gene(code) can lead to a speciation-like change in their patterns. For the last pattern, the length of the F-code grows to 19 up to iteration 90. All the following codes not shown in Fig. 5 are set to 0. Thus the full F-codes for the case last pattern is f=X600000000000. The length of an F-code suggests a natural measure of complexity of the pattern, as it means the number of different rules applied to generate the pattern. The result in Table 1 shows that the number of possible patterns increases with the increase in the complexity of the patterns. But the increase in the number does not mean that we can find complex patterns easily than the simpler patterns. Figure 6 shows the frequency of patterns as a function of $|F|$. This graph is obtained through 100,000 random patterns obtained using random rule-sets up to iteration 100. It is clear that the longer the F-codes are, the lesser probable they can be met in random simulations.

Why is the complex patterns rare than the simpler ones when the former can have the larger number of variations? This may be best explained through the schematic diagram shown in Fig. 7. The abundance of F-codes are illustrated for three color case up to $|F|$=5. Each box in each of the columns corresponds to a gene(code) in the F-code. The three boxes in the first column represents color code 0, 1 and 2 respectively. Of course, the size of the box has no meaning. If an F-code is finished at a given length, the corresponding part of the column is left as a blank. The number of boxes just ending in a specific column represents the number of F-codes by that length. For example, there 1, 2 and 4 boxes ending in the columns 2, 3 and 4, respectively. The number of boxes is like the one shown in column (a) in Table 1. The increasing number of the smaller boxes with increasing $|F|$ implies the increase in the number of patterns with longer F-codes. But it should be remembered that those seemingly finished rows may not be finished forever. They maybe continued if we increase the number of iterations, as explained above. Thus the diagram shown in Fig. 7 should also be understood in terms of an iteration number. Assume the total height of the figure shown in Fig. 1 is 1.0. Then the sum of the heights of the smaller boxes in each column denotes the probability that the F-codes are of length greater or equal to the column number, when simulated with random set of rules. When $|F|$ goes to infinity, the set of boxes shown in the last column of Fig. 7 reminds us the Canto set [http://en.wikipedia.org/wiki/Cantor_set]. It is known that the Canto set is of measure zero. There are infinitely many smaller boxes (F-codes) but the probability to obtain them in random simulation is zero, as implied by the measure 0. Because the length of F-codes cannot go to infinity and the fraction of the eliminated parts in each of the columns are not constant, the present case does not exactly match the definition of the Cantor set. But it will be a convenient concept to explain the present issue here. The decrease in the probability is calculated in the last column of the Table 1.

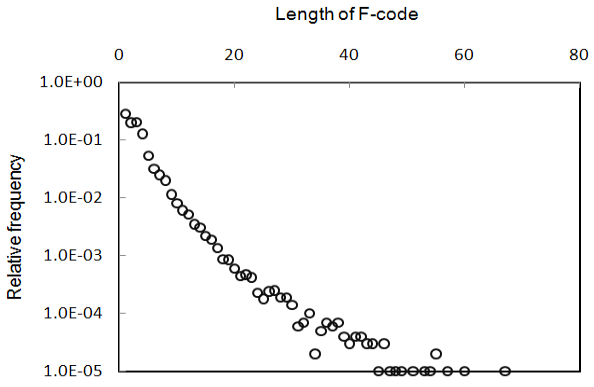


Fig. 6 Relative frequency of patterns obtained in a random generation of 100K cases(iteration=100).

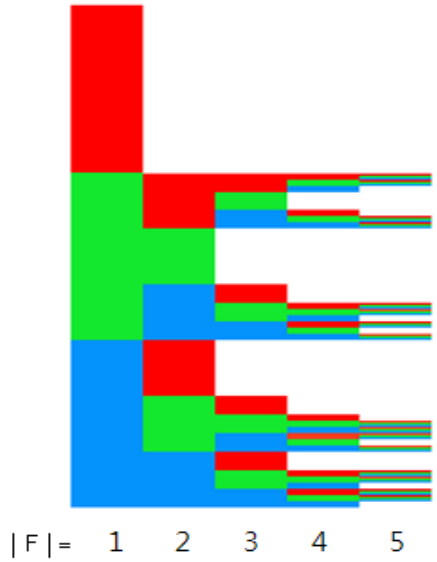


Fig. 7 Canto-like set diagram can explain the relation between the complexity(Length of F-code) and relative frequency of the patterns found in random simulations.

The complexity-frequency issue treated in the present study could be applied, for example, to explain the species abundance in real ecological space. It is known that the body-size distribution of mammals show a right-skewed distribution such that the larger animal is rare compared to the smaller ones. In a theoretical model, for example, the shape of these curve is explained only in terms of the body-size itself [Clauset and Erwin 2008]. No attention is paid on the nature of the possible space of the genetic codes, or in terms of the present study, space of the F-codes itself. To illustrate this, let us look at the mutations shown in Fig. 5 again. Consider a species represented by the last pattern shown in Fig. 5. If there happen a mutation in the eighth gene such that the F-code changes from X6 to X5, the mutation causes a serious reduction in its complexity, which probably mean that the corresponding mutation cannot survive. But at the present, this is just an imagination. Applying the present model to investigate the concept of complex systems will be a topic of our future study.

Conclusions

Patterns emerging from a one-rule firing scheme are investigated. Through an exhaustive numerical simulation, it was shown that a more complex system has the larger number of variations. But in random simulation, it was found that the complex system is the less likely to be found. This situation is explained in terms of a Cantor-like set proving schematically why the more complex system is rare when there are a larger number of variations for the complex patterns.

References

Clauset, A. and Erwin, D., (2008), The evolution and distribution of species body size, Science 321, 399

Langton,C.G., (1984), Problem solving during artificial selection of self-replicating loops Physica D: Nonlinear Phenomena 10 (1-2), pp. 135-144

Levy, S., (1992), Artificial Life, Vintage books.

Neumann, J., (1966), The theory of Self-Reproducing Automata, Burks, A. W. (Ed.), Univ. of Illinois Press, Urbana and London.

Pan, Z., Reggia, J.A. (2010), Computational discovery of instructionless self-replicating structures in cellular automata, *Artificial Life* 16 (1), pp. 39-63.

Sayama, H. (1999), A new structurally dissolvable self-reproducing loop evolving in a simple Cellular Automata space, Artificial Life, 5 (4), pp. 343-365.

Shin, J., (2010), Flower patterns from one-rule firing cellular automata, submitted.

Wolfram, S., (1984), Universality and complexity in cellular automata, Physica D, 10:1-35.

Wuensche, A., (2004), Self replication by glider collisions: the beehive rule, Artificial Life IX, 286-291.

Evolutionary Dynamics

Potential and Promise of Open-Ended Evolution in Artificial Life

Nicolas Chaumont, Christoph Adami

Keck Graduate Institute, Claremont, CA 91711

Abstract

Since Bedau et al. identified the simulation of open-ended evolution in digital life media as one of the key problems in the field of Artificial Life (Bedeau et al., 2000, Artificial Life 6, p.363), no attempt has convincingly solved the problem until this day. Creating open-ended evolution ultimately boils down to creating niches: A new evolutionary feature can only be retained if there is an ecological niche in which it becomes an innovation. An environment with a limited potential for hosting niches is inherently restricted as far as evolutionary innovations and open-ended evolution are concerned. Moreover, static niches, even in a very large number, are not enough to enable open-ended evolution, they need to appear persistently.

Here, we present an in-silico system in which ecological niches are not explicitly defined, but arise as the consequence of the combination of the environmental layout and the adaptation of its resident population. The population consists of three-dimensional, autonomously foraging, blocky creatures (Sims, 1994, Artificial Life 1, p.353)(Chaumont et al., 2007, Artificial Life 13, p.139) with sensory-motor capabilities that are controlled with a neural network, coexist in the world, and compete for its resources. In this implementation they reproduce asexually, and the genome that codes for its morphology and behavior (via the neural network that controls its motions) undergoes mutations during reproduction. The world in which the creatures live is a three- dimensional, physically simulated environment where energy resources are continuously replenished, decay, and eventually absorbed by foragers. Creatures die if their energy is depleted, and are born from a parent that has accumulated enough energy to reproduce. There is no explicit fitness function in this system; however since poor foragers quickly die out, we witness a strong selective pressure to pass on genes for increasingly sophisticated foraging behavior to the offspring. Niches are not explicitly defined either. Since there is a wealth of possible foraging behaviors, the actual number of niches is impossible to determine. Moreover, as the population changes in number and in foraging strategies, the opportunities for any individual organism change as well, creating or removing niches dynamically as the population evolves in time.

In the initial construction of the world, we included several types of food sources placed at varying heights on pedestals, in addition to food sources distributed at ground level (See Figure 1). We believe that specialized morphological traits or behaviors that are necessary to exploit a particular resource can, if coupled with sexual recombination, allow disruptive selection to split the initial population into two or more morphologically distinct groups that will become increasingly isolated post-zygotically (Via, 2001, Trends Ecol. Evol., 16, p.381). Thus, in such an Artificial Life system new species can in principle emerge by speciating in sympatry, parapatry, or allopatry.

We believe that in such a system, open-ended evolution as understood by the Artificial Life community (Bedeau et al., 2000, Artificial Life 6, p.363) can ultimately be observed. A number of as yet un-implemented features are possible that will aid in open-ended evolution, such as the definition of chemical pathways that dictate a creature's affinity to metabolize specific food sources, and the possibility of emergence of trophic levels, by specifying that the blocks from which the creatures are created have nutritional value, and can either be scavenged, or hunted.

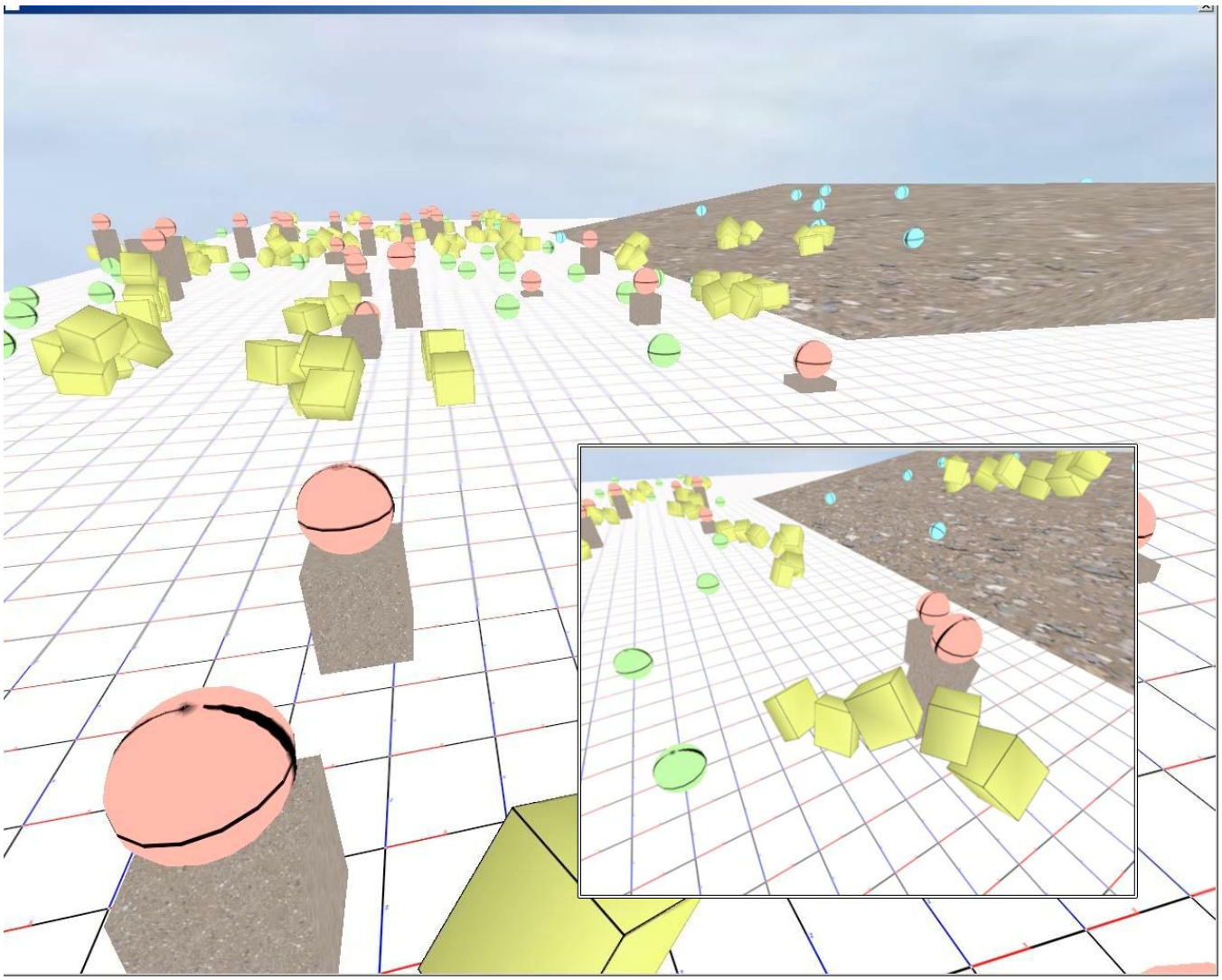


Figure 1: A snapshot of a world with three types of resources (green, red, and blue spheres) that require different morphologies or behaviors access. 3D organisms are yellow. In the inset, a virtual creature is toppling a pedestal to reach a red resource sphere.The blue resources are on inclines and require a form of locomotion that can counteract the low friction of the surface. Standard organisms cannot climb this incline.

Genetic and Environment-Induced Innovation: Complementary Pathways to Adaptive Change that are Facilitated by Degeneracy in Multi-Agent Systems

James M. Whitacre

CERCIA, School of Computer Science, University of Birmingham, UK
j.m.whitacre@cs.bham.ac.uk

Extended Abstract

Understanding how heritable and selectively relevant phenotypes are generated is fundamental to understanding evolution in biotic and artificial systems. With few exceptions (e.g. viral evolution), the generation of phenotypic novelty is predominantly discussed from two perspectives. The first perspective is organized around the concept of fitness landscape neutrality and emphasizes how the robustness of fitness towards mutations can facilitate the discovery of heritable adaptive traits within a static fitness landscape (Wagner 2008).

A somewhat distinct perspective is organized around the concept of cryptic genetic variation (CGV) and mostly emphasizes the importance of particular population properties within a dynamic environment (Gibson and Dworkin 2004). CGV is defined as standing genetic variation that does not contribute to the normal range of phenotypes observed in a population, but that is available to modify a phenotype after environmental change (or the introduction of novel alleles). In short, CGV permits genetic diversity in populations when selection is stable yet exposes heritable phenotypic variation that can be selected upon when populations are presented with novel conditions. Both pathways to adaptation (genetic and environment-induced phenotypic variation) are likely to have contributed to the evolution of complex traits (Palmer 2004) and theories of evolution that cannot account for both pathways are either fragile to or reliant upon environmental dynamics.

Here we use requirements from these pathways to evaluate the merits of a new hypothesis on the mechanics of evolution. In particular, Gerald Edelman has proposed that degeneracy – the existence of structurally distinct components with context dependent functional similarities – is a fundamental source of heritable phenotypic change at most/all biological scales and thus is an enabling factor of evolution (Edelman and Gally 2001) (Whitacre 2010). While it is well-documented (and intuitive) that degeneracy contributes to trait stability for conditions where degenerate components are functionally compensatory (Whitacre and Bender 2010), Edelman argues that the differential responses outside those conditions provide access to unique functional effects, some of which can be selectively relevant given the right environment.

We recently reported evidence that degeneracy supports the first pathway by creating particular types of neutrality in static fitness landscapes that can increase mutational access to heritable phenotypes (Whitacre and Bender 2010), and fundamentally alter a system's propensity to adapt (Whitacre et al. in press).

Using models from (Whitacre et al. in press), here we present findings that degeneracy within evolving multi-agent systems may create characteristic features of CGV at the population level; thereby allowing the model to also exploit an environment-induced pathway to adaptation. In particular, we show that for static environments, degeneracy facilitates high genetic diversity in populations that is phenotypically cryptic, i.e. individuals remain similar in fitness (Figure 1). When the environment changes, trait differences across the population are revealed and some individuals display a phenotypically plastic response that is highly adaptive for the new environment. These CGV features are not observed in populations when degeneracy is absent from our model. We discuss the theoretical significance of a single mechanistic basis (degeneracy) for complementary pathways to adaptation.

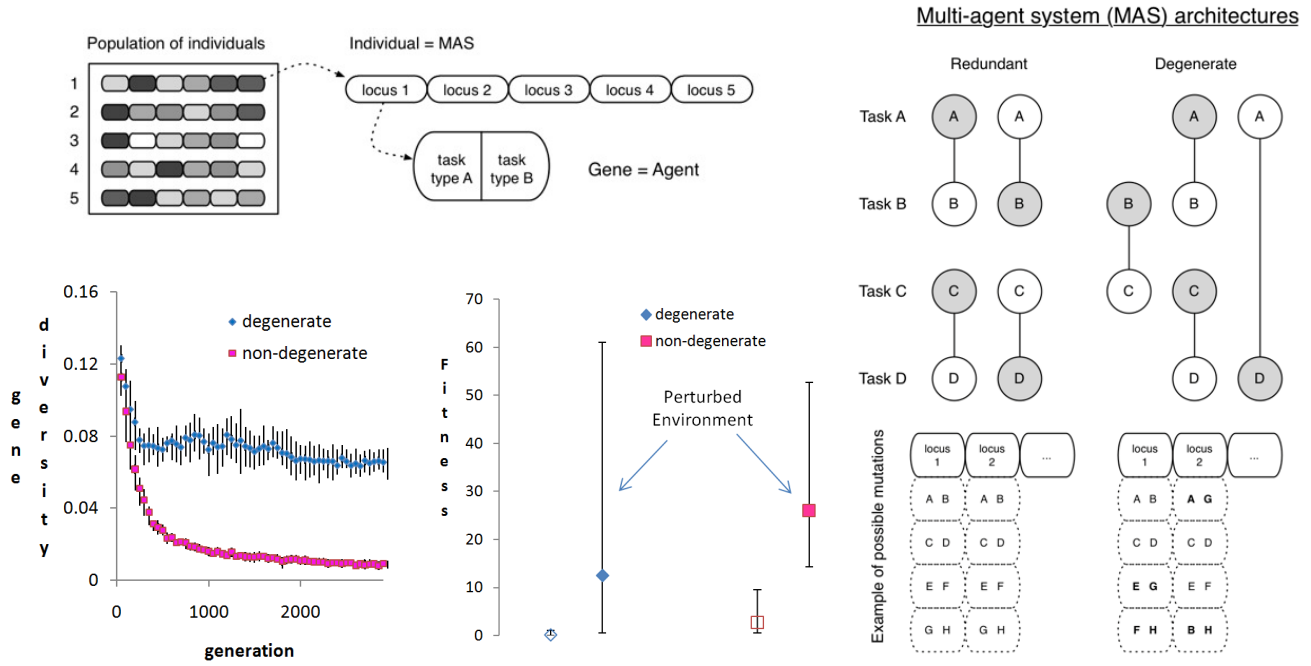


Figure 1: Top-Left Panel) Multi-Agent System (MAS) encoded within a genetic algorithm. Agents perform tasks to improve MAS fitness in its environment, see (Whitacre et al., in press). **Top-Right Panel**) Illustration of genetic architectures for degenerate and non-degenerate MAS. Each agent is depicted by a pair of connected nodes, with the two nodes representing two types of (genetically determined) tasks that the agent can perform. **Bottom-Right Panel**) The number of task type combinations (alleles) possible in a degenerate MAS is larger than non-degenerate MAS so it is necessary to artificially restrict experiments to similar genotype space sizes as illustrated here; for more details see mutation operator description in (Whitacre et al., in press). **Bottom-Left Panel**) Genetic diversity (Hamming distance in genotype space between population members) plotted over 3000 generations of evolution within a static environment. **Bottom-Middle Panel**) Fitness of population members at generation 3000 is recorded and then reevaluated within a moderately perturbed environment. In these results, we observe high genetic diversity in the degenerate population that is cryptic (negligible fitness differences) within the stable environment, but that is released/exposed when the same population is presented with a new environment. Some of the observed plastic phenotypic responses are found to be highly adaptive in the new environment. CGV was largely absent in the evolution of non-degenerate MAS, even when environments are modified to increase mutational robustness (not shown). Optimal fitness = 0 for original and perturbed environments.

Acknowledgements: This research was partially supported by DSTO and CERCIA.

References

Edelman, G. M. and J. A. Gally (2001). "Degeneracy and complexity in biological systems." Proceedings of the National Academy of Sciences, USA **98**(24): 13763-13768.

Gibson, G. and I. Dworkin (2004). "Uncovering cryptic genetic variation." Nature Reviews Genetics **5**(9): 681-690.

Palmer, A. (2004). "Symmetry breaking and the evolution of development." Science **306**(5697): 828.

Wagner, A. (2008). "Robustness and evolvability: a paradox resolved." Proceedings of the Royal Society of London, Series B: Biological Sciences **275**: 91-100.

Whitacre, J. M. (2010). "Degeneracy: a link between evolvability, robustness and complexity in biological systems." Theoretical Biology and Medical Modelling **7**(6).

Whitacre, J. M. and A. Bender (2010). "Degeneracy: a design principle for achieving robustness and evolvability." Journal of Theoretical Biology **263**(1): 143-153.

Whitacre, J. M. and A. Bender (2010). "Networked buffering: a basic mechanism for distributed robustness in complex adaptive systems." Theoretical Biology and Medical Modelling **7**(20).

Whitacre, J. M., P. Rohlfshagen, X. Yao and A. Bender (in press). The role of degenerate robustness in the evolvability of multi-agent systems in dynamic environments. 11th International Conference on Parallel Problem Solving from Nature (PPSN 2010), Krakow, Poland.

On the Interplay of Kinetics, Thermodynamics, and Information in Simple Replicating Systems

Bernat Corominas-Murtra[1], Harold Fellermann[1,2], Ricard Solé[1,3,4], and Steen Rasmussen[2,3]

[1]ICREA-Complex Systems Lab, Universitat Pompeu Fabra, Dr. Aiguader 80, 08003 Barcelona, Spain
[2]Dep. of Physics and Chemistry, University of Southern Denmark, Campusvej 55, 5230 Odense, Denmark
[3]Santa Fe Institute, 1399 Hyde Park Road, Santa Fe NM 87506 USA
[4]Institut de Biologia Evolutiva, CSIC-UPF, Passeig Maritim de la Barceloneta, 37-49, 08003 Barcelona, Spain
bernat.corominas@upf.edu, harold@ifk.sdu.dk

Extended Abstract

Life uses energy to acquire and process information. The process of gaining information through evolutionary search cannot be uncoupled from its physico-chemical embodiment and the energetic needs and entropic constraints of the latter (Morowitz (1979); Smith (2008)). Therefore, a serious study of biological as well as prebiotic information processing requires: *(i)* an explicit accounting of the thermodynamics underlying replication, mutation, and selection of self-replicating systems, and *(ii)* an explicit treatment of the influence of information on the metabolism and kinetics of the replicating system, *(iii)* an explicit description of the thermodynamic instability that drives replication, and *(iv)* a concept of information that explicitly takes into account the evolutionary path through a fitness landscape.

Because this approach clearly exceeds the current description of contemporary living organisms, we develop our framework for a minimal coupled container-information-metabolism system (protocell) that is presumably able to self-replicate and evolve (Rasmussen (2003)). Thanks to the simplicity of this system, it is possible to gain a detailed understanding of the atomistic processes that underlie information replication, metabolic regulation, aggregate replication, as well as mutation and selection.

To study *(i)*, we take into account the detailed thermodynamic needs for replication of the entire protocell and possible mutation of its information component. The simplicity of the protocell allows us to define reasonable estimates for a quantitative fitness function, i.e. kinetic rate influence of the information component on the metabolic rate, which accounts for point *(ii)*. By further estimating the thermodynamic container stability depending on composition (point *(iii)*) we derive a Master equations governing protocell population dynamics in information as well as container fitness spaces.

To deal with *(iv)*, we propose a concept of information that overcomes the explicit treatment of genetic sequences but focuses instead on the complexity of the evolutionary path. This is achieved by identifying a genetic lineage, i.e., a sequence of cell duplications and possible mutations, as a decision making process (where the outcome of each decision is evaluated depending on whether the offspring has a higher or lower fitness). This enables us to express the evolutionary path as a chain of decisions, i.e. evolutionary improvements, stagnations or aggrevations. Under suitable units, the sequence of decisions can be identified as a symbolic string, whose information content is its associated Kolmogorov Complexity – a conceptual, more powerful precursor of statistical information (Li and Vitányi (1993)).

Equipped with this framework, we are able to analyze the interplay of thermodynamics, kinetics, and information in a quantitative manner. In particular, we can quantitatively derive the *maximum power principle* (MPP) (Lotka (1922); Cai (2004)) that postulates a connection between evolutionary acquired information and the underlying kinetics of life, and we derive a quantitative analogue of the *Landauer principle* (Landauer (1961)) for evolving replicators (LPER), that postulates a relation between thermodynamics and acquirable information in a physical system. We explore the outcome of these relations for several limiting cases, as well as for the particular protocell design under consideration.

References

Cai, T. T., Olsen, T. W., and Campbell, D. E. (2004). Maximum (em)power: a foundational principle linking man and nature *Ecol. Model.* 178:115-119

Landauer, R. (1961). Irreversibility and heat generation in the computing process. *IBMJ. Res. Dev.* 3:183-191.

Li, M. and Vitányi, P. M. B. (1993). *An Introduction to Kolmogorov Complexity and its Applications*, Springer-Verlag, New York, 1993.

Lotka, A. (1922). Contribution to the energetics of evolution *Proc. Nat. Acad. Soc. USA* 8(**6**):147-151

Morowitz, H. J. (1979). *Energy Flow in Biology*. OxBow Press, Woodbridge, 1979.

Rasmussen, S., Chen, L., Nilsson, M., and Abe S. (2003). Bridging nonliving and living matter. *Artif. Life* 9:269-316.

Smith, D. E. (2008). Thermodynamics of natural selection I: Energy flow and the limits on organization. *J. Theor. Biol.* 212(**2**):185-197.

Evolving Learning Ability in Cyclically Dynamic Environments: The Structuring Force of Environmental Heterogeneity

Solvi Arnold[1], Reiji Suzuki[2] and Takaya Arita[2]

[1]Graduate School of Natural Sciences, Utrecht University, The Netherlands
[2]Graduate School of Information Science, Nagoya University, Japan
solvi.arnold@phil.uu.nl

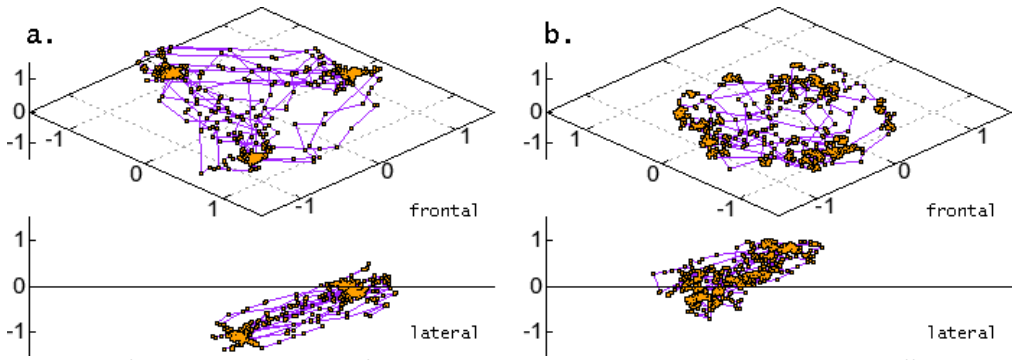
In nature, adaptation occurs at multiple levels (learning, multiple levels of evolution). Adaptation processes at different levels are known to interact in various ways. Especially the mechanism by which learning guides evolution (the Baldwin effect) has become a common theme in Artificial Life (see e.g. Suzuki and Arita, 2004, 2008). This research focuses on the opposite direction: how evolution facilitates learning by devising innate structures that guide learning processes.

In the computational model presented here, weight and plasticity structure of simple artificial neural networks are evolved in an environment with a cyclic dynamic, switching through 3 phases (or "seasons"), each requiring a distinct behaviour. To allow evolution to shape the networks' weight dynamic, the genotype contains a separate plasticity (learning-rate) gene for every individual connection. It is shown that in response to the environmental dynamic, evolution devised a modular network structure, containing one rigid behaviour module for each phase, and a flexible module governing the switching between behaviours. The evolution process shows a pronounced Baldwin effect, indicating that the evolution of the innate structures guiding learning is itself guided by the presence of learning ability. The evolved networks show a highly structured plasticity differentiation. Comparison with networks using only a single global plasticity gene reveals that this differentiation facilitates learning by allowing the nets to learn without deteriorating their modular structure.

Both a functional and a mathematical interpretation of the evolved network structure are given. Mathematically, plasticity differentiation induced a large reduction in dimensionality of the networks' active weight-space, and a high degree of consistency in weight-configuration between subsequent environmental cycles. Functionally, we find that through internalization of environmental structure, the networks gain an ability to improve their responses to unseen stimuli, in a way that similarity-based generalization alone cannot account for. The alignment of internal (network) structure with external (environmental) structure enables the nets to process a given piece of learning data as evidence for being in a particular environmental phase, and to adjust the whole of their behaviour accordingly. This feature might be understood as a primitive analogue of "latent learning" (see e.g. Gould and Gould, 1994) or the "poverty of the stimulus" phenomenon (Chomsky, 1980).

To further investigate the role of internalization, we compare performance of networks with varying numbers of hidden nodes. Reducing this number below the minimum necessary for successful internalization causes a marked drop in performance, while increasing the number beyond this minimum has virtually no effect. Next, as internalization should show as improved robustness against noise, we compare performance of networks with and without plasticity differentiation in a noisy environment. We find that the difference in performance is indeed increased in the noisy environment.

These findings are considered in the context of evolution of cognition, and linked to the idea that cognition is to be understood as adaptation to structured environmental heterogeneity (Spencer, 1855; Godfrey-Smith, 1994, 2002). Finally, extension to larger networks and more complex tasks is discussed.



Comparison of connection weight dynamics of lop layer connections in networks with (a) and without (b) plasticity differentiation, over the course of 8 environmental cycles of 3 phases each. The clusters in (a) each correspond to one of the environmental phases, while in (b), subsequent occurences of the same phase fail to produce identical weight configurations.

References

Chomsky, N. (1980). *Rules and representations*. Oxford: Basil Blackwell.

Godfrey-Smith, P. (1996). *Complexity and the Function of Mind in Nature*. Cambridge University Press.

Godfrey-Smith, P. (2002). Environmental Complexity and the Evolution of Cognition. In Sternberg, R. J. and Kaufman, J. C. (Ed.), *The Evolution of Intelligence*, pp. 233-249. Mahwah: Lawrence Erlbaum.

Gould, J. L. and Grant Gould, C. (1994). *The Animal Mind*. Scientific American Library.

Spencer, H. (1855). *The Principles of Psychology*. Longman.

Suzuki, R. and Arita, T. (2004). Interactions between learning and evolution: Outstanding strategy generated by the Baldwin effect. *Biosystems*,77(1-3):57–71.

Suzuki, R. and Arita, T. (2008). How learning can guide evolution of communication. In S. Bullock, J. Noble, R. Watson, and M. A. Bedau (eds.) *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, pp. 608-615. MIT Press, Cambridge, MA.

Evolution of Cooperation and Developmental Constraints: a GA-driven Approach

Moritz Buck and Chrystopher L. Nehaniv

Adaptive Systems Research Laboratory
University of Hertfordshire
Hatfield, Hertfordshire
AL10 9AB
United Kingdom
m.buck@herts.ac.uk

Abstract

In this paper we present a model of evolution of cooperation driven by a Genetic Algorithm (GA) and a two-level fitness function representing cooperative and individualistic behaviours. The GA drives the evolution of artificial Genetic Regulatory Networks (GRNs) that controls colonies of artificial cells in a grid. This set-up is used to study the effect of computational complexity on the evolution of cooperation. Computational complexity being linked to the concept of developmental constraints in evolution. We show that there is a trade of between the computational complexity of a behaviour and the increase in fitness it bestows. Cooperation (being a more complicated behaviour than the individualistic one) will only (stably) evolve if the fitness reward of it is above a certain threshold. We also argue the importance of Artificial Life models (as opposed to mathematical ones) for the study of dynamical aspects of evolution.

The study of evolution is a fascinating yet very complex field. The main concepts of evolution are very simple but the very nature and time scales of it makes it very difficult to study *in vivo*. From biology one can only "easily" study the genotypical and phenotypical snapshots of the organisms alive nowadays (and the few ones we have palaeontological data from). And depending on which perspective one looks at a problematic in Evolution one can get very different analyses. As can be seen in the big debates between Richard Dawkins and Stephen Jay Gould (Sterelny, 2001), and particularly the debate about the importance of developmental constraints (Gould and Lewontin, 1979). The idea of developmental constraints is that every organism carries a certain evolutionary baggage, and this baggage influences how the species can evolve. For example a mountain lion might be fitter with an extra pair of legs, but the evolution of an extra pair of legs is very improbable due to the developmental history of the lion. The debate about developmental constraints is not so much about the existence of them in evolution, but about their power to shape it (Beatty, 1997). This kind of debate is very difficult to solve due to the issue of lacking quantifiable data.

Developmental constraints are linked to a known aspect optimisation: the fitness landscape. In an optimisation problem a fitness landscape describes the fitness of each solution. If one uses algorithms like Genetic Algorithms (GAs) to solve such a problem, each solution is encoded in a genome, and the phenotypical expression of that genome has a certain fitness. A problem can have multiple local optima, and the difficulty of going from one optima to another can be likened the difficulty of overcoming certain "developmental" constraints.

In this paper we present a methodology to study certain constraints linked with the evolution of cooperation. As a first approximation the nature of the evolution of cooperation is a classical problem of optimization. It can be represented as a fitness landscapes with two main fitness peaks: one of individualistic cell behaviour, and one of cooperative cell behaviour. The main question being: "how to get from one peak to another?". This depends a lot on the shape of the fitness landscape, and in the case of a GA the shape of the genotype-phenotype mapping. The dependency on the fitness landscape is quite trivial, but the importance of genotype-phenotype mapping might need some explanation.

What is meant by genotype-phenotype mapping? In our model the genotype is a string of booleans, and the phenotype is a network. We use this to illustrate the notion of phenotype-genotype mapping. If one mutates booleans in the genotype, it can have an impact on the network, but not every mutation will have the same impact, and also the way the genotype maps the phenotype influences the impact of mutations. The effects can be of various amplitudes, changing the dynamics of the network gradually or directly. Also their can be imbalances in the effect of mutation: the effects of mutations can be similar for every boolean of the genotype, or very different for certain positions.

The ease with which one can go from one fitness peak to the next one depends directly on the shape of the fitness landscape and the genotype-phenotype mapping. In this experiment, we implemented two levels of fitness, one requiring a higher level of organization requiring inter-cellular cooperation (the formation of a checker-board pattern), and an individualistic behaviour. The peak for *individual behaviour* is very flat and lower (or equal, the height of this peak is a parameter) than the *cooperative behaviour* peak which is

narrower and higher. This is the case because the individualistic part of the fitness is very simple for the cells to compute, the cells just need to stay in a stable state independent from their neighbours, for the cooperative behaviour all the cells need to have some communication with their neighbours so to synchronise their states, which is much more complicated for the cells to evolve. In this situation if the genotype-phenotype mapping is too "soft" (effect of mutations are small) evolution might never leave individuality, whereas if it is to "rugged" (mutations have dramatic effects), the risk is that evolution finds the peak for cooperation but loses it again before stabilizing correctly. So we hope that our genotype-phenotype mapping have some elements of both types: the capacity of moving around across the fitness landscape with mutations that have a big effect on the phenotype, yet not every mutation should have these big effects so that the GA can explore the area around interesting phenotypes without risking to lose the peak.

To do this study, we use a GA to evolve Genetic Regulatory Networks (GRNs). To measure the fitness of a genome the GRN controls artificial cells in a six by six grid, all the cells having the same GRN. We then use a two-level fitness function to measure the quality of that genome, each of the two levels representing one of the peaks in the fitness landscape. Each of the peaks represent one behaviour that can evolve (individualistic and cooperative), each behaviour being qualitatively of different computation complexities (individualistic: very simple and no need to communicate, cooperative: more complex and necessity to communicate). We vary the height of the peaks and the population size of the GA. With this set-up we can find out under which circumstances evolution will go towards the higher peak of cooperation and when not, hence see how the difference in complexity between the two solutions limit the evolution of new behaviours.

Models

Artificial Cell

The main part of our model is an artificial cell. It is reasonably simple and composed of two main elements: a genome and a genetic regulatory network, the genome encoding for the network. The GRN being based on Boolean Networks.

Genetic Regulatory Network The GRNs used for this experiment operate as Boolean control networks. The same model has been used in (Buck and Nehaniv, 2006a, 2007), and is similar to Kauffman's random Boolean networks (Kauffman, 1993), but our networks interact continually with their ambient environment (cf. (Quick et al., 2003; West-Eberhard, 2003)), and the GRN-controlled cells interact with each other in a manner similar to that in Bull and Alonso-Sanz (2008). The structure of a single genome is shown in Figure 1. Inside a cell there are n different proteins, the level of each protein is modelled by a Boolean value reflecting its presence (*true*) or absence (*false*). The network structure is derived from the genome as described in section . The cell's genome consists of a string of genes, with each gene composed of a regulatory part and a part specifying its protein product as in nature (Watson et al., 2003; Davidson, 2001b).We use a two-level genetic regulatory structure (see Schilstra and Nehaniv (2008) for other models genetic control logic). The regulatory part represents the inbound connections of the gene in the network whereas the product part represents the outbound. The inbound part (regulatory part) is structured in so-called *cis-sites*, which themselves each consist of a number of binding sites. A binding site returns a Boolean value depending on the presence in the cell of the protein it is supposed to bind. The values returned by all the binding sites of a cis-site are joined by an *AND* operator. The obtained value is then negated if the cis-site is an *inhibitory* one. Then all the values returned by the cis-sites of a gene are joined by an *OR* operator. This value is then finally negated if the gene is *default on*, if the final value of this operation is *true* then the protein encoded by the gene will be produced, i.e. the value indicating the presence of this protein in the cell will be set to *true*. If more than one gene can produce the same protein, to set the value for that protein to true for the cell, any one of them suffices. The system has a one time step 'memory'; at every simulation time step it takes the protein state vector of the cell in the previous step and creates a new protein state vector for the next time step using the genetic regulatory network.

Formally, for each gene of a cell's genome, we have for each protein-binding site i, potentially binding some protein p_ℓ, the present binding value b_i,

$$b_i = \begin{cases} true & \text{if binding protein } p_\ell \text{ is present} \\ false & \text{if binding protein } p_\ell \text{ is not present.} \end{cases}$$

The expression value c_j of a cis-site j,

$$c_j = \begin{cases} \bigwedge_{\text{all i}} b_i & \text{if j is activatory} \\ \neg \bigwedge_{\text{all i}} b_i & \text{if j is inhibitory} \end{cases}$$

where the logical AND-operation is taken over all binding sites b_i of the given cis-site c_j. The final protein production p_k of the gene k is

$$p_k = \begin{cases} \bigvee_{\text{all j}} c_j & \text{if k is default off} \\ \neg \bigvee_{\text{all j}} c_j & \text{if k is default on} \end{cases}$$

where the logical OR-operation is taken over all cis-sites c_j of gene k. The new value of p_k *for the cell* will be *true* if and only if at least one gene produces p_k. It can be shown that this system is complete in the sense of combinatorial logic: given a Boolean vector of size n (the vector of the

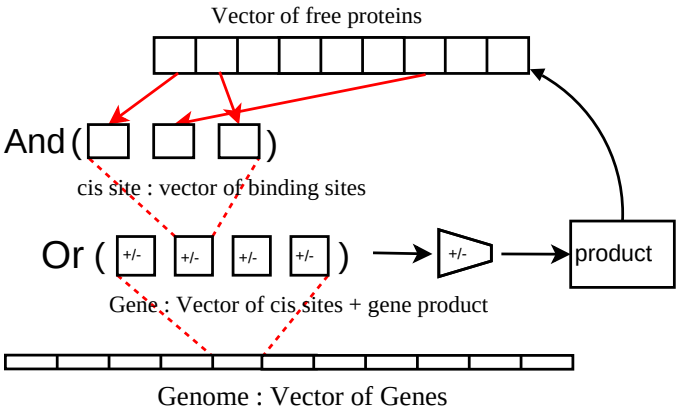


Figure 1: Schematic of the Boolean genetic regulatory network model

n proteins of the cell) there always exists at least one network computing every one of the $(2^n)^{(2^n)}$ possible Boolean functions. (This can be easily seen by writing the logical function to determine the presence or absence of each protein in conjunctive normal form as function of the activation levels of all proteins in the cell, and translating this form into a genome with n default-on genes.).

This model has also some other interesting characteristics. It is quite robust to mutation, at least in principle, for example if you duplicate one gene the function represented by the network is not altered, which is not the case for most continuous GRN models (Buck and Nehaniv, 2006b; Knabe et al., 2006).

Simulation

The simulations take place in a 2D toroidal grid, with a von-Neumann neighbourhood. Each position of the grid is occupied by an agent (cell) controlled by a GRN. All of the cells in the grid have the same controlling GRN. We use in this experiment GRNs with 16 different proteins, therefore each cell can be in one of 2^{16} different states, but not all of these proteins have an actual effect on the environment most of them are internal states used to control the cells.

This architecture gives the cells the potential to communicate. The communication is controlled by five proteins. Four proteins control with which neighbouring cells will be communicated and one is the protein to be "sent" to those neighbours. If the protein to be sent is present the neighbouring cells with which the cell communicates "will be given" the protein (e.g. set to *true*).

The cell can be in three possible "visual" states, two of them being "cooperative" and one "individualistic" state. One protein controls the "individualistic" state, if it present in the cell this cell is in that state, if it is not it is in one of the "cooperative" states. Those states are controlled by another protein, if it is present the state will be "red" else "green". Those different states are independent of the communication, an "individualistic" cell can still communicate and receive communication, the "visual" states are used during the computation of the fitness function.

The regulation networks, the communication and the "visual" states are updated in a random synchronistic way. The cells are updated in a random order but each cell only one time during each time step. This is the only non-deterministic component of the simulation. Each simulation has a finite fixed number of time steps.

The Genetic Algorithm

The Genetic Algorithm (GA) used in this experiment is quite standard. We only use bit flips as th only source of variation and tournament selection as selection routine. No elitism has been used.

Encoding The encoding we chose for the networks is a highly simplified version of the encoding of GRNs in real biology (Hawkins, 1996; Davidson, 2001a). We wanted to keep a certain number of characteristics of the double-stranded DNA helix which encodes the regulatory networks of all living organisms on earth. Our genome as in biology is composed by a very small alphabet: in nature the four nucleotides: adenine, thymine, guanine and cytosine; in our genome only two bases, 0 and 1. Our genome is sectioned as in biology by different tags which are recognised by the cellular machinery: certain combinations of bases have a certain specific meaning for the genome. There are some main differences between the encoding we use and the natural one. First our encoding is deterministic. For example, the fact that biological genomes are situated in a three dimensions, which can bring a high amount of modulation into the expression patterns. Another point to notice that our genome is of the single stranded sort.

The genome is sectioned in genes. A gene is tagged by a so-called *gene tag* a pattern composed by four ones (`'1111'`). This tag is followed by one bit to set the type

of gene ('1' for *default on*, '0' for *default off* gene) and a certain amount of bits to define the produced protein (in our experiment we used a 64 protein system so six bits are necessary to encode the binary representation for each protein). Preceding a *gene tag* is the regulatory region of that gene, that region is separated into cis-sites each one of those starting with a *cis-site start pattern* consisting of a double zero ('00') followed by a bit for the type (*inhibitory* or *activatory*) and a certain number of binding sites (each of six bits to characterise the protein to bind at the site). Using a certain set of predetermined rules (minimum space between cis-site start tags, minimum space between two gene tags, ordering and precedence rules, ...) we can give to each bit of the genome a certain unequivocal function (even if this is merely to identify the bit as uninterpretable other than as "junk") so as to build the GRN represented by that genome. This structure allows a genetic regulatory network to be unambiguously constructed from the genome.

The encoding is illustrated in Figure 2, which shows the encoding of a single gene. A genome consists of a string of such genes. The number and lengths of genes may vary between genomes in the evolving population. In the present model a gene encodes at most one protein product.

Fitness Developing an environment with a natural (implicit) fitness is not easy and usually needs many parameters. Therefore we chose to work with an explicit fitness function. This fitness here has the particularity to be actually two fitness functions representing two levels of selection, one trying to reach a high level goal needing cooperation and one representing a low level single cell goal, both goals being exclusive, so both goals are in competition.

The lower level fitness is simply to stay as long as possible in the "individualistic" state. We check for each cell in the grid which cell has stayed longest in that state and normalise that time to 1. If $t_{\text{ind}}(i)$ is the time cell i has spend in the "individualistic" state the "individualistic" fitness F_{ind} of a GRN in a certain simulation is

$$F_{\text{ind}} = \frac{\max\limits_{\text{all cells i}} t_{\text{ind}}(i)}{t_{\text{sim}}},$$

where t_{sim} is the length of a simulation.

The higher level goal is to create a checker-board with the "red" and "green" cells. At each time step of a simulation, for each cell of the grid in a "cooperative" state we check the neighbourhood, for each of the neighbouring cell which is in a different state but not individualistic that cell gets a score of 0.25 (remark : 0.25 is 1 divided by the number of neighbours 4). So at each time step each cell can get a score between 0 and 1. Those scores are then summed for each time step over all cells and normalized to 1. If $n_i(j,t)$ is equal to 0.25 if the j^{th} neighbour of cell i is in the same state than cell i but not the individual one at time t, else 0, $f_{\text{group}}(i)$ the fitness of cell i is

$$f_{\text{group}}(i) = \begin{cases} \frac{1}{t_{\text{sim}}} \sum\limits_{t=1}^{t_{\text{sim}}} \sum\limits_{j=1}^{\text{neighbours}} n_i(j,t) & \text{if i cooperative} \\ 0 & \text{if i individualistic} \end{cases},$$

hence the higher level fitness of the GRN after a simulation F_{group} is the average of f_{group} over the colony

$$F_{\text{check}} = \frac{1}{n_{\text{cells}}} \sum_{\text{all cells i}} f_{\text{check}}(i),$$

where n_{cells} is the total number of cells in the grid.

The final fitness of a GRN is the maximum between the higher level and the lower level fitness weighted by $\alpha \in [0,1]$, a parameter weighting the advantage/disadvantage of being individualistic. So the fitness F of a GRN lies in the interval $[0,1]$ and is

$$F = \max(F_{\text{check}}, \alpha \cdot F_{\text{ind}}).$$

Experimental Investigation

We have for this experiment run a 10 GAs (mutation rate: 0.002, cross-over rate: 0.5, starting genome size: 1000, size of tournament: 25, size of the grid: 6×6, length of simulation: 30). The values of α studied were between 0 and 1 included in steps of 0.1, and the population sizes 125, 250, 500, and 1000. If α is set to zero, there is no contribution to fitness from the individualistic fitness, the evolution is only driven by the high level fitness. We have done the same experiment for three different length of GA, 200 and 1000 generations, and an experiment with 200000 fitness evaluations (which is equivalent to 1600 generations for population size 125, 800 for population size 250, 400 for population size 500, and 200 generations for a population size of 1000).

The smaller α, the higher the incentive for the cellular colonies to evolve cooperation because the reward of cooperation is so much greater than the simple non-cooperation.

Also we have used the OR-unconstrained communication protocol (a problem independent communication protocol, where a cell communicates only with their direct neighbour) described in (Buck and Nehaniv, 2008) with six communication proteins.

In this experiment we are not directly interested in the actual fitness achieved, rather we are interested in the local optimum in which an evolutionary run stabilizes. There are, as mentioned earlier, two local optima, one for individual behaviour (shallow peak), one for cooperative behaviour (steep peak), the steep peak being always higher or equal to the shallow one. The shallow peak's height is characterized by the parameter α, so any GA run that has stabilized on a fitness value above α has certainly achieved some degree of multicellular cooperation. So for each set of 10 GA-runs we have computed the proportion of runs that have achieved

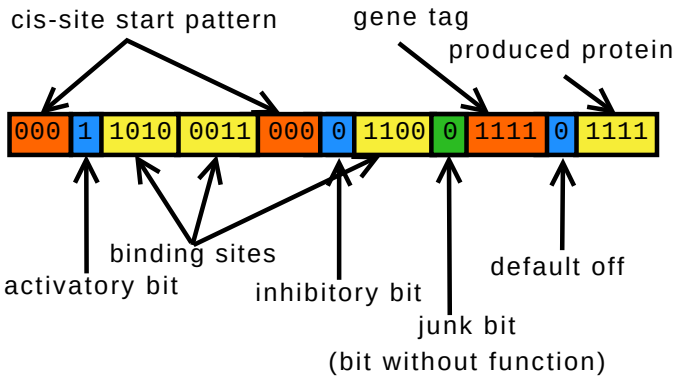


Figure 2: Example of the gene structure. This gene encodes a product protein 1111 and has a two cis-site regulatory region. The first cis-site is activatory and comprised of two binding sites, while the second is inhibitory and has a single binding site. The gene is off by default. Genomes are concatenations of such genes. The logical function computed by this example is $p_{15}^{t+1} = (p_6^t \cap p_3^t) \cup \neg(p_{12}^t)$, where p_i^t is the Boolean value attributed to the protein i at time step t.

this, we call this the proportion of multicellularity, and this is the value plotted on the graphs of Figure 3 to 8. This proportion of multicellularity is an approximation of the probability that an evolutionary run with a set population size will stabilize on multicellular behaviour in a set number of generation.

Results

Figures 3 to 8 are the results of this experimental set-up.

The first remark is that for most of the plots one can notice a non-linear transition. Only for the plots with a population size of 125 it is not obvious (which is probably due to the fixed-sized tournament selection). This signifies that there is a tipping point at which the behaviour of the evolutionary algorithm changes. Before that point evolution has a very high probability of reaching a multicellularity and then, for a very small increase of α this probability drop very close to null. The dependence of the tipping point on the population size is slightly unclear, in figures 3, 4, and 5, one can see that the tipping points for population sizes 250 and 500 are very close, yet for population sizes 125 and 1000 they are respectively lower and higher. One has to be slightly careful, with the analysis of figures 3 and 4, because as the number of generations is fixed and the population size in not the same for every line, the number of fitness evaluations for each line of the plots are different. Naturally a GA with a smaller population size will take more time (generation-wise) to explore the fitness landscape. For this purpose we have included the results of figure 5, where all the GAs could take the same amount of sample points in the fitness landscape (the same number of fitness evaluations), but one can see that the resulting plot is qualitatively similar to the two previous ones.

In figures 6 and 7, we have presented some of the same results but with a fixed population size, and varying number of generations. We can see that qualitatively the lines are the same, hence the number of generations does not matter for the transition, or at least for the explored parameter space. This means that the minimum number of generations we have picked (200) is enough for the GA to get to a stable point.

This result allows us to compute figure 8, which is a combination of the previous graphs. We recomputed every point of the graph using the data from figures 3 to 5, without considering the number of generations (basically, supposing that all the GA-runs had been stopped at the same number of generation, or at stabilization). This allows figure 8 to have a better definition on the vertical axis.

We can still, in figure 8, notice the transition, the two curves for population sizes 250 and 500 that are very close, the line for a population of 1000, that drops a bit later, and the one for a population of 125 that starts to drop already for small values of α.

Conclusion

First, there is a non-linear shift of the evolutionary behaviours of the GAs. Both evolutionary attractors (individuality and cooperation) have clearly defined domains of attraction depending on α, which parametrizes the contribution of organismal vs. cellular levels fitness. We are supposing that a colony's fitness can always be higher if cooperating, then if not, this transition shows, that even though the higher fitness would always push towards cooperation, due to the combination of a complex fitness landscape and genotype-phenotype mapping, this high fitness is not always achieved. Even more the behaviour on which the evolutionary runs stabilize seem to be in an almost deterministic way depending on a set of parameters. One could consider α an environmental parameter defining the difficulty of cooperation in that environment (or the fitness gain of being a cooperative colony). In that case one could say that evolu-

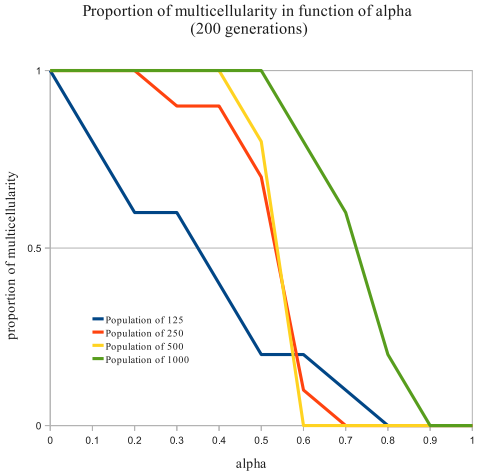


Figure 3: Proportion of evolutionary runs that have stabilized on the multicellular state after 200 generations, for different values of α.

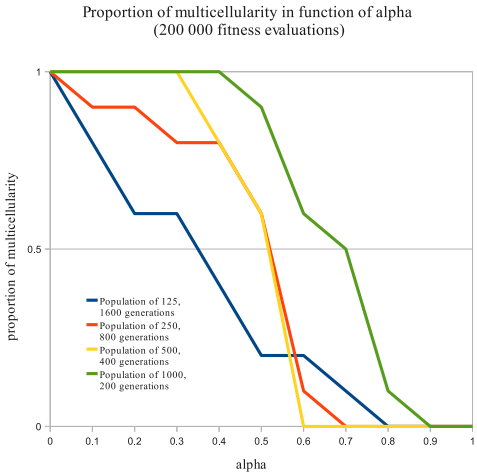


Figure 5: Proportion of evolutionary runs that have stabilized on the multicellular state after 200000 fitness evaluations, for different values of α.

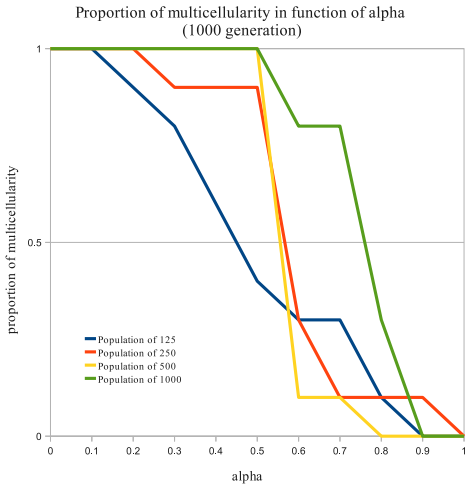


Figure 4: Proportion of evolutionary runs that have stabilized on the multicellular state after 1000 generations, for different values of α.

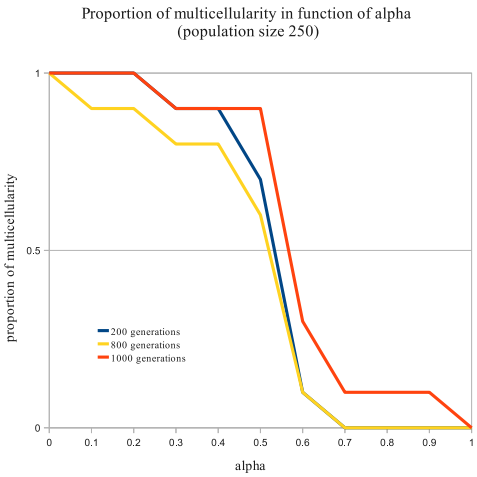


Figure 6: Proportion of evolutionary runs that have stabilized on the multicellular state for varying number of generations, for different values of α, for a population size of 250.

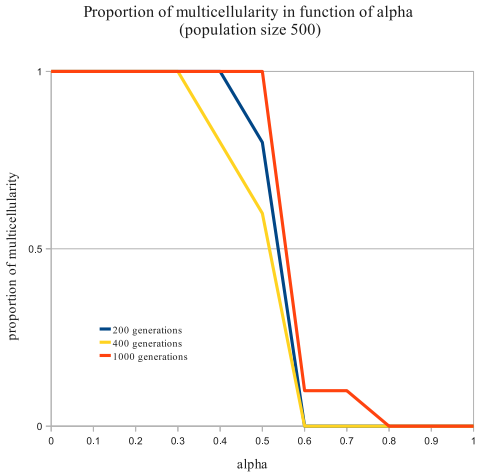


Figure 7: Proportion of evolutionary runs that have stabilized on the multicellular state for varying number of generations, for different values of α, for a population size of 500.

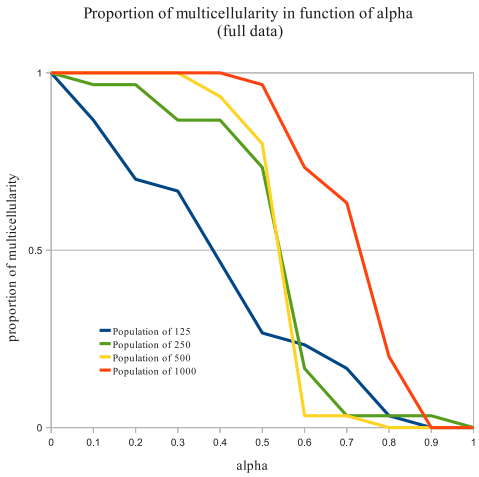


Figure 8: Proportion of evolutionary runs that have stabilized on the multicellular state for varying number of generations, for different values of α (full data).

tion is not only quantified by absolute fitness, but also by the computational complexity of the way of achieving this fitness.

So in a certain sense we can "see" the effect of developmental constraints on the evolution of cooperation. Cooperation can only evolve when the benefits of cooperation "compensate" for its complexity. To compare this result with the example we presented in the introduction: the mountain lion could eventually evolve a third pair of legs if the fitness reward of this extra pair of legs would "outweigh" its cost in complexity.

This kind of results are not easily discovered through classical models of evolution. In most mathematical or game theoretical approaches the system will always stabilize at the stable point of highest pay-off, which in the case of this model design would have been the multicellular peak. Of course one could design a model to take into account a parameter representing computational complexity and complexity of the genotype-phenotype mapping, but as for the purposes of identification of new hypotheses traditional models of population genetics and game theory would not have been able to show this kind of behaviour. Also, in mathematical or game theoretical approaches, the cooperative or individualistic behaviours are fixed by the genotype, in the model we presented in this article, they are partially determined by the genotype but through a complex genotype-phenotype mapping, hence the cells can switch their behaviour during their lifetime. This is very important to study, and cannot be done with more classical models. Of course this is mostly a toy model, neither of both behaviours are very complex, and this kind of effect could be even more dramatic in more complex environments, yet it can be a first step to a different way of the study of diverse aspects of evolution.

Acknowledgments

Partial support for this work by the OPAALS EU project FP6-034824 is gratefully acknowledged.

References

Beatty, J. (1997). Why do biologists argue like they do? *Philosophy of Science*, 64.

Buck, M. and Nehaniv, C. L. (2006a). Discrete developmental genetic regulatory networks for the evolution of cooperation. In Kumar, S., Hornby, G. S., and Bongard, J., editors, *Developmental Systems: Papers from the AAAI Fall Symposium (October 13-15, 2006, Arlington, Virginia)*, pages 9–15. Association for the Advancement of Artificial Intelligence.

Buck, M. and Nehaniv, C. L. (2006b). Effect of multi-level fitnesses on the development of multicellular artificial organisms. In *Proceedings of the* 7^{th} *German Workshop in*

Artificial Life (GWAL-7), Berlin. Akademische Verlagsgesellschaft Aka.

Buck, M. and Nehaniv, C. L. (2007). Colouring graphs using a grn/cell-based system. In *Proceedings of IPCAT 2007*.

Buck, M. and Nehaniv, C. L. (2008). Communication and complexity in a GRN-based multicellular system for graph colouring. *Biosystems*, 94(1-2):28–33.

Bull, L. and Alonso-Sanz, R. (2008). On coupling random boolean networks. In et al., A., editor, *Automata 2008: Theory and Applications of Cellular Automata*, pages 292–301. Luniver Press.

Davidson, E. H. (2001a). *Genomic Regulatory Networks: Development and Evolution*. Academic Press.

Davidson, E. H. (2001b). *Genomic Regulatory Systems: Development and Evolution*. Academic Press.

Gould, S. J. and Lewontin, R. C. (1979). The spandrels of san marco and the panglossian paradigm: a critique of the adaptationist programme. *Proc R Soc Lond B Biol Sci*, 205(1161):581–598.

Hawkins, J. D. (1996). *Gene Structure and Expression*. Cambridge University Press, 3rd edition.

Kauffman, S. A. (1993). *The origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press.

Knabe, J. F., Nehaniv, C. L., Schilstra, M. J., and Quick, T. (2006). Evolving biological clocks using genetic regulatory networks. In Rocha, L. M., Yaeger, L. S., Bedau, M. A., Floreano, D., Goldstone, R. L., and Vespignani, A., editors, *Artificial Life X: Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, pages 15–21. MIT Press.

Quick, T., Nehaniv, C. L., Dautenhahn, K., and Roberts, G. (2003). Evolving embodied genetic regulatory network-driven control systems. In *Advances in Artificial Life (Proc. European Conference on Artifical Life - ECAL'03)*, volume 2801 of *Lecture Notes in Artificial Intelligence*, pages 266–277. Springer Verlag.

Schilstra, M. J. and Nehaniv, C. L. (2008). Bio-logic: gene expression and the laws of combinatorial logic. *Artificial life*, 14(1):121–133.

Sterelny, K. (2001). *Dawkins vs. Gould : survival of the fittest / Kim Sterelny*. Icon, Cambridge, U.K. :.

Watson, J. D., Baker, T. A., Bell, S. P., Gann, A., Levine, M., and Losick, R. (2003). *Molecular Biology of the Gene*. Benjamin Cummings, fifth edition.

West-Eberhard, M. J. (2003). *Developmental Plasticity and Evolution*. Oxford University Press.

Darwinian Evolution of Cooperation via Punishment in the "Public Goods" Game

Arend Hintze[1], Christoph Adami[1]

[1]Keck Graduate Institute, 535 Watson Dr., Claremont CA 91711
adami@kgi.edu

Abstract

The evolution of cooperation has been a perennial problem for evolutionary biology because cooperation is undermined by selfish cheaters (or "free riders") that profit from cooperators but do not invest any resources themselves. In a purely "selfish" view of evolution, those cheaters should be favored. Evolutionary game theory has been able to show that under certain conditions, cooperation nonetheless evolves stably. One of these scenarios utilizes the power of punishment to suppress free riders, but only if players interact in a structured population where cooperators are likely to be surrounded by other cooperators. Here we show that cooperation via punishment can evolve even in well-mixed populations that play the "public goods" game, if the synergy effect of cooperation is high enough. As the synergy is increased, populations transition from defection to cooperation in a manner reminiscent of a phase transition. If punishment is turned off the critical synergy is significantly higher, illustrating that indeed punishment aids in establishing cooperation. We also show that the critical point depends on the mutation rate so that higher mutation rates actually promote cooperation, by ensuring that punishment never disappears.

Introduction

"Tragedy of the commons" is the name given to a social dilemma (Hardin, 1968) that occurs when a number of individuals maximize their self-intertest by exploiting a public good, and by doing so harm their (and other's) own long-term interest. This is but one dilemma (Frank, 2006) that can be described within the framework of Evolutionary Game theory (Smith, 1982; Axelrod, 1984; Dugatkin, 1997; Hofbauer and Sigmund, 1998; Nowak, 2006). While the tragedy of the commons is important in social science and politics (overfishing and the destruction of the environment in general come to mind), it also plays an important in role in biology: both the evolution of virulence (Frank, 1996) and the manipulation of a host by a group of parasites (Brown, 1999) can be viewed as a dilemma of the public goods type.

An environment where cooperators provide goods and share synergy is vulnerable to defectors. It has been shown that *punishment* is an effective way to counteract defectors (Fehr and Gachter, 2002; Fehr and Fischbacher, 2003; Hammerstein, 2003; Nakamaru and Iwasa, 2006; Camerer and Fehr, 2006; Gürerk et al., 2006; Sigmund et al., 2001; Henrich and Boyd, 2001; Boyd et al., 2003; Brandt et al., 2003; Helbing et al., 2010). Because punishment involves an additional cost to the co-operators that already invest into the public good (Yamagishi, 1986; Fehr, 2004; Colman, 2006), these cooperators (termed "moralists" by Helbing et al. 2010) are themselves vulnerable to the invasion of non-punishing cooperators called "secondary free-riders". As a consequence, we might expect that moralists ultimately become extinct, either because they were outcompeted by defectors, or by cooperating free-riders who benefit from the punishment without the associated cost. Alternatively, if moralists are ultimately successful in eliminating defectors, the punishment gene stops to be under selection and should drift, again resulting in the demise of moralists.

It has recently been shown that, instead, in simple spatial games, moralist can win direct competitions (Helbing et al., 2010) if the environmental conditions are favorable, namely if the cost to benefit ratio of punishment favors moralists over defectors. Spatial games, where the offspring of successful strategies are placed near the parent, and where as a consequence strategies are more prone to play against kin strategies, give rise to spatial reciprocity (Sigmund et al., 2001). This appears to be the advantage that moralists need to gain superiority. In the simulations of Helbing et al., evolution proceeded by the *imitation* of successful neighboring strategies rather than Darwinian evolution, but the dynamics are similar. However, because strategies in those simulations are deterministic (limiting genetic space to four genotypes), large grids had to be used in order to prevent premature extinctions.

Here, we show that spatial reciprocity is in fact not a necessary condition for the evolution of cooperation via punishment and the dominance of moralists, if stochastic strategies can evolve via Darwinian dynamics in a framework where decisions are encoded within genes that adapt to their environment. There are conditions where cooperation evolves even without punishment, but absent those, punishment can promote the evolution of cooperation, as long as punishment

is effective and cheap, in well-mixed populations. If cooperation becomes so dominant that defectors are brought to extinction, the punishment gene drifts to neutrality. Finally, we also observe that stable environments that are believed to be more predictable for players also increase the chance for cooperators to evolve and to be stable, as observed earlier within the iterated Prisoner's Dilemma (Iliopoulos et al., 2010).

Experimental Design

We evolve stochastic strategies playing the public goods game with punishment. Each individual in a group of k players ($k = 5$ in the present implementation) can decide to cooperate by making a contribution of 1 unit to the public good, while defecting individuals do not contribute. We encode this choice as a probability p_C, which can be thought of as the outcome of a network of genes that encode this decision. When mutating strategies, instead of mutating the individual genes that make up the decision pathway, we simply replace the parental probability p_C by a uniformly drawn random number in the offspring. We will call the locus encoding the probability p_C simply the "C gene".

The sum of all contributions from cooperating players is multiplied by r (the synergy factor) and divided among all players. In addition, each player has the option to punish players who do not contribute. This decision is encoded by an independent probability p_P, called the "P gene". Following Helbing et al. 2010, those players who defect suffer a fine β/k levied by the punishers in the group, whereas the punishers suffer a penalty of γ/k. At each update, every player engages in a game with all its assigned opponents. The number of cooperators N_C, defectors N_D, moralists N_M and immoralists (players who defect but also punish Helbing et al. (2010)) N_I is computed, and the payoff is assigned as follows: A cooperator receives

$$P_C = r\frac{(N_C + N_M + 1)}{k+1} - 1 \,, \tag{1}$$

while a defector takes away

$$P_D = r\frac{(N_C + N_M)}{k+1} - \beta\frac{(N_M + N_I)}{k} \,. \tag{2}$$

Moralists receive

$$P_M = P_C - \gamma\frac{(N_D + N_I)}{k} \,, \tag{3}$$

while immoralists earn

$$P_I = P_D - \gamma\frac{(N_D + N_I)}{k} \,. \tag{4}$$

The population consists 1,024 individuals who each have four assigned opponents. Since all opponents are also players, each individual plays five games per update. The choices of each individual are determined by their probabilities to cooperate p_C and to punish p_P. After each round, 2 percent of the population is replaced using a Moran-process (Moran, 1962) in a well-mixed fashion, that is, the identity of the players in the group is unrelated to their ancestry. Players that are not replaced are allowed to accumulate their score, which is used to calculate the probability that this player's strategy will be chosen to replicate and fill the spot of a player that was removed in the Moran process. Every individual's genes mutates with a probability μ when replicated. As mentioned earlier, the mutation of a gene replaces the probability with a uniformly distributed random number. After 500,000 updates, the line of descent (LOD) of the population is reconstructed, by picking a random organism of the final population and following its ancestry all the way back to the starting organism, which has $p_C = 0.5$ and $p_P = 0.5$. Because there is only one species in these populations, the LOD of the population coalesces to a single LOD (which is why it is sufficient to pick a random genotype for following the LOD).

As the strategies adapt to the environmental conditions (specified by the parameters that define the game, as well as the spatial properties, the mutation rate, and the replacement rate), the probabilities that appear on the LOD tell the story of that adaptation, mutation by mutation. While the LOD in each particular run can show probabilities varying wildly, averaging many such LODs can tell us about the selective pressures the populations face. In particular, averaging the probabilities on the LODs after they have settled down (from the transient beginning at the random strategy $(p_C, p_P) = (0.5, 0.5)$) can tell us the *fixed point* of evolutionary adaptation (Iliopoulos et al., 2010). We determine this fixed point by discarding the first 250,000 updates of every run (the transient), along with the last 50,000 (in order to remove the dependence of the LOD on the randomly chosen anchor genotype) and averaging the remaining 200,000 updates. Note that this fixed point is a computational fixed point only: we do not mean to imply that the population's genotypes all end up on this exact point. Rather, due to the nature of the game, the evolutionary trajectories approach this point and then fluctuate around or near it. Thus, the fixed point reflects the *mean* successful strategy given the conditions of the game.

Results

When mapping the possible parameters β (fine) and γ (cost) each in the range from 0.0 to 1.0 and at low synergy $r = 3.0$, we find that defection is the most prevalent strategy on the LOD (see Figures1 a and b), as was found previously (Brandt et al., 2003; Helbing et al., 2010). When β and γ vanish, punishment has no effect, nor is there a cost associated with that punishment. At this point, the P gene is not under selection and drifts. A drifting gene can be recognized by a mean of 0.5 and a variance of $1/12 \approx 0.083$ at

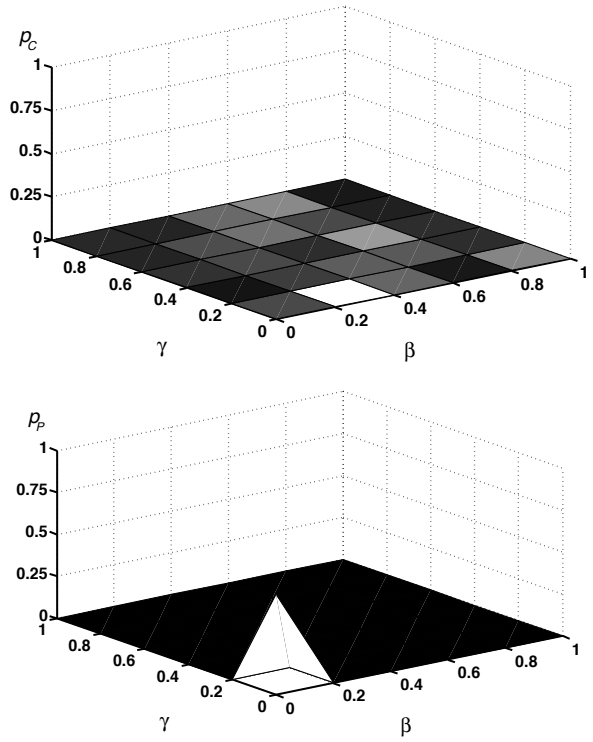


Figure 1: Mean probabilities for p_C (a) and p_P (b) measured on the LOD, for β and γ ranging from 0.0 to 1.0 in 0.2 increments, at $r = 3$.

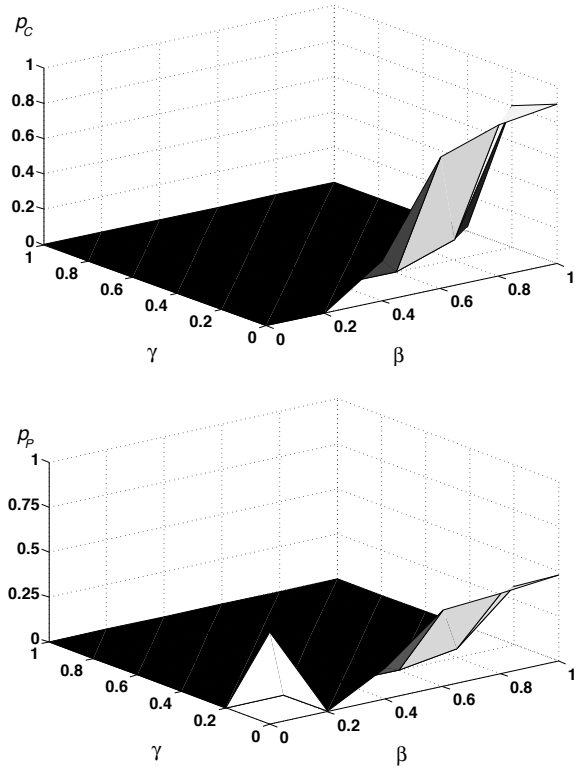


Figure 2: Mean probabilities for p_C (a) and p_P (b) measured on the LOD, for β and γ ranging from 0.0 to 1.0, in increments of 0.2, at r=4.

the fixed point, as expected for the average and variance of a uniform random number on the interval (0,1). Thus, for this value of synergy (and lower), we find that the strategy fixed point is defection without punishment, except for the values γ=β=0, where punishment is random.

As the degree of synergy increases to $r = 4$, cooperation starts to appear even in this well-mixed population (while it appears as early as $r = 2$ for sufficiently high β and low γ in the spatial version of the game Brandt et al., 2003; Helbing et al., 2010). We find players cooperating ($p_C \approx 0.8$) at high β and low γ (see Figure 2a), which indicates that under conditions where punishment is not very costly or even free, punishment pays off. In addition we notice that the probability to punish increases under the same conditions that allows cooperation (high β and low γ, that is high impact, low cost of punishment), indicating that punishment is indeed used to enforce cooperation (Fig. 2b). The mean punishment probability grows to 0.5, but at the same time the variance shows that this gene is not under drift (data not shown). Still, the distribution of probabilities on the LOD is fairly broad, indicating that periods of strong punishment give way to periods where agents are much more forgiving. Thus, it appears that punishment under these conditions is effective even if it is engaged in only intermittently.

Increasing the synergy level even higher towards r=4.5 shows the emergence of dominance of cooperation (p_C >0.5) for most of the range of punishment cost and effectiveness, see Figure **??**a. At the same time the punishment probability reaches 0.5 for a larger range of parameters (Fig. 3b), but the mean payoff probability on the LOD never exceeds 0.5, implying that full persistent punishment is not stable. Increasing synergy to $r = 5$ reveals a population that engages in cooperation for almost all parameter settings (see Figure 4), even at conditions where punishment is costly without much impact ($\beta < 0.5$, $\gamma > 0.5$) but the variance suggests that at high punishment effect and low cost, this gene may be drifting (as it is only selected for if defectors are prominent). This outcome is expected because at $r = 5$, the cooperators' payoff is equal to or higher than the defectors, and exactly equal in the absence of punishment. Thus, defectors should disappear and punishment become random.

Critical Behavior

Previously, a phase transition between cooperative and defective behaviour in the public goods game was observed for the spatial version Szabo and Hauert (2002); Brandt et al. (2003) of the game (but not the well-mixed version). In

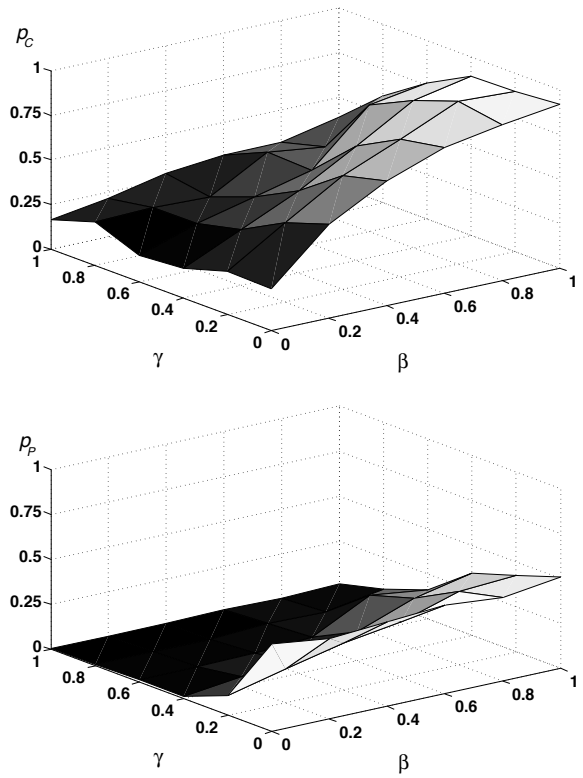


Figure 3: Mean probabilities for p_C (a) and p_P (b) measured on the LOD, for β and γ ranging from 0.0 to 1.0 in 0.2 increments, at r=4.5

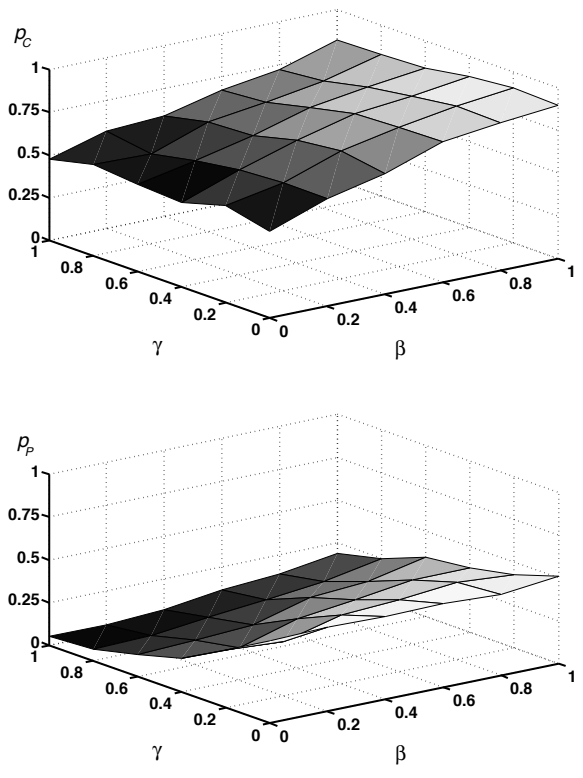


Figure 4: Mean probabilities for p_C (a) and p_P (b) measured on the LOD, for β and γ ranging from 0.0 to 1.0 in 0.2 increments, at r=5

Fig. 5 we show the mean probability at the evolutionary fixed point of both the C gene (black lines) and the P gene (grey lines) as a function of the synergy level r, for different mutation rates (dotted lines: $\mu = 0.001$, dashed lines: $\mu = 0.01$ and solid lines: $\mu = 0.02$, which is the mutation rate we used in Figs 1-4). We note the sudden emergence of cooperation at a critical synergy level, but that this level depends on the mutation rate. For the highest mutation rate (black solid line in Fig. 5) cooperation emerges the earliest. As the mutation rate is lowered, the critical point moves to the right and the fixed point probability is higher. The emergence of punishment (grey lines in Fig. 5) follows the same trend, and again we notice that the mean never exceeds 0.5.

It is instructive to study how punishment affects the critical point. To do this, we ran a control of the experiment where punishment did not exist. In that case, we observe a critical r that is significantly higher that what we observe with punishment (see Fig. 6, showing again how punishment aids in the establishment of cooperation. Note also that the levels of cooperation achieved are significantly higher when punishment exists.

We can calculate approximately the point at which cooperation is favored in a mean-field approach that does not take mutation and evolution into account, by writing Eqs. (1-2) in terms of the density of cooperators ρ_C in the population. Both naked cooperators and punishing cooperators (moralists) contribute to this density, i.e., $\rho_C = (N_C + N_M)/N$, where N is the total number of players in the population. We can also introduce the mean density of punishers $\rho_P = (N_M + N_I)/N$. Because the mean density of cooperators and punishers is the *same* for both cooperators and defectors in a well-mixed scenario (but not for spatial play!), we can then write

$$P_C = r\frac{k\rho_C + 1}{k+1} - 1 \tag{5}$$

and

$$P_D = r\frac{k\rho_C}{k+1} - \beta\rho_P \ , \tag{6}$$

and we expect cooperation to be favored if

$$P_C - P_D = \frac{r}{k+1} - 1 + \beta\rho_P > 0 \tag{7}$$

or

$$r > (k+1)(1 - \beta\rho_P) \ . \tag{8}$$

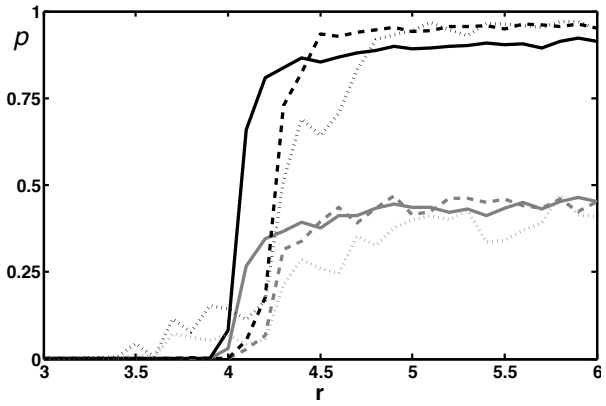


Figure 5: Mean probability of cooperation p_C (black lines) and punishment p_P (grey lines) at the evolutionary fixed point of the trajectory, as a function of the synergy r for three different mutation rates: dotted: $\mu = 0.001$, dashed: $\mu = 0.01$, and solid: $\mu = 0.02$. [Note: Statistics for the lowest mutation rate will be improved for camera-ready version]

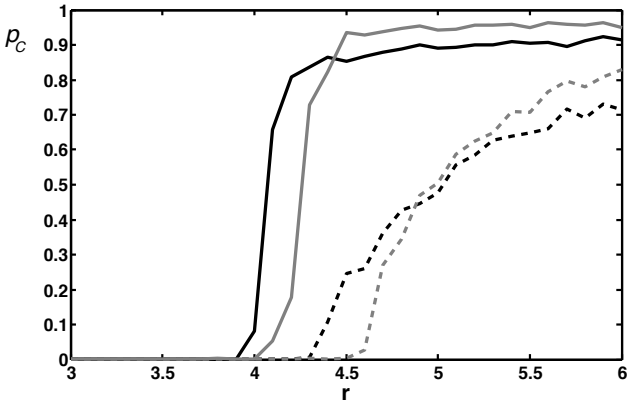


Figure 6: Mean probabilities for p_C measured on the LOD, for cost of punishment $\beta = 0.8$ and effectiveness of punishment $\gamma = 0.2$, as a function of synergy r. Solid line is the standard protocol, while dashed line represents experiments with punishment turned off ($p_P = 0$).

This equation implies that the emergence of cooperation depends crucially on the density of punishers. In fact, the mean-field theory predicts that cooperation in the absence of punishment emerges only at $r = 5$, while we see it emerge quite a bit earlier than that (see Fig. 6, dashed lines). Note, however, that the critical point moves towards the predicted value $r = 5$ as the mutation rate is lowered, which would not be surprising as the theory holds strictly only for vanishing mutation rate. Because we expect that the density of punishers increases as the mutation rate increases (because mutations can introduce defectors at an elevated rate, necessitating a more pronounced punishment response), we can also expect the critical mutation rate to drop commensurately, but it is clear from the previous comment that there are mutation rate effects in the dynamics of the population that are independent of punishment.

Because of the critical importance of punishers in determining the synergy level at which cooperation emerges, the public goods game with a genetic basis implies a curious dynamics close to the critical point. Below the critical point, defection is a stable strategy, and punishment is absent. Only when cooperation emerges as a possibility, punishment becomes more and more important, leading to a lowering of the critical synergy for cooperation. Thus, cooperation emerges rapidly and decisively once a critical level has been achieved. Once cooperation is dominant and defectors all but driven to extinction, punishment becomes irrelevant and the gene begins to drift. As this happens, the fraction of punishers drops, raising the critical synergy. Thus, a drifting punishment gene can lead to the sudden re-emergence of defectors as stable states. Once those have taken over, the reverse dynamics begins to unfold. In other words, we should observe periods of cooperation and defection follow each other closely as the synergy is near the critical point. An investigation of the population dynamics at the critical point will be the subject of a subsequent investigation.

Discussion

We studied Darwinian evolution of stochastic strategies in the public goods game for a well-mixed populations, using genes that encode the probabilities for cooperation and punishment. It is known that punishment can drive the evolution of cooperation above a critical synergy level as long as there is a spatial structure in the environment (Brandt et al., 2003; Helbing et al., 2010). It was also previously believed that in well-mixed populations cooperation can only become successful if additional factors like reputation (Sigmund et al., 2001) are influencing the evolution. Here we show that cooperation readily emerges in a well-mixed environment above a critical level of synergy. This critical level is influenced by a number of factors, such as the rate of punishment and the mutation rate.

If the conditions for punishment are good (that is, the cost for punishment is low and the effect is high) we find cooperative strategies that also have elevated probabilities to punish, that is, they are moralists. But if punishment is cheap and effective, we also see that defectors practically vanish, which in turn obviates the need for punishment, so much so that the punishment gene begins to drift. This effect, however, is also mutation rate dependent, because higher mutation rates will automatically create a higher influx of defectors even if they cannot be maintained by selection.

We conclude that in well-mixed populations cooperation

can emerge if the synergy outweighs the defectors' reward. If the mutation rate is low enough, the loss of defectors makes punishment obsolete, that is, the selective pressure to punish disappears. Naturally, once this has occurred defectors can again gain a foothold, and the balance of power between cooperators and defectors could shift. Such a shift, however, reinstates the selective pressure to punish, leading to a re-emergence of moralists that can drive defectors out once more. Thus, for synergy factors near the critical point, we can expect oscillations between cooperators and defectors, and no strategy is ever stable (Hintze et al., 2010).

Acknowledgements

We thank the member of the Evolutionary Dynamics group at KGI for discussions. This work was supported by the National Science Foundation's Frontiers in Integrative Biological Research grant FIBR-0527023.

References

Axelrod, R. (1984). *The Evolution of Cooperation*. Basic Books, New York, NY.

Boyd, R., Gintis, H., Bowles, S., and Richerson, P. J. (2003). The evolution of altruistic punishment. *Proc Natl Acad Sci U S A*, 100(6):3531–5.

Brandt, H., Hauert, C., and Sigmund, K. (2003). Punishment and reputation in spatial public goods games. *Proc Biol Sci*, 270(1519):1099–104.

Brown, S. (1999). Cooperation and conflict in host-manipulating parasites. *Proceedings of the Royal Society of London Series B-Biological Sciences*, 266(1431):1899–1904.

Camerer, C. F. and Fehr, E. (2006). When does "economic man" dominate social behavior? *Science*, 311(5757):47–52.

Colman, A. M. (2006). The puzzle of cooperation. *Nature*, 440(744-745).

Dugatkin, L. A. (1997). *Cooperation Among Animals: An Evolutionary Perspective*. Princeton University Press, Princeton, NJ.

Fehr, E. (2004). Human behaviour: Don't lose your reputation. *Nature*, 432(7016):449–450.

Fehr, E. and Fischbacher, U. (2003). The nature of human altruism. *Nature*, 425(6960):785–791.

Fehr, E. and Gachter, S. (2002). Altruistic punishment in humans. *Nature*, 415(6868):137–140.

Frank, S. (1996). Models of parasite virulence. *Quarterly Review of Biology*, 71(1):37–78.

Frank, S. A. (2006). *Foundations of Social Evolution*. Princeton University Press.

Gürerk, O., Irlenbusch, B., and Rockenbach, B. (2006). The competitive advantage of sanctioning institutions. *Science*, 312(5770):108–11.

Hammerstein, P., editor (2003). *Genetic and Cultural Evolution of Cooperation*, Cambridge, MA. MIT Press.

Hardin, G. (1968). The tragedy of the commons. *Science*, 162:1243–1248.

Helbing, D., Szolnoki, A., Perc, M., and Szabo, G. (2010). Evolutionary establishment of moral and double moral standards through spatial interactions. arxiv.org:1003.3165v1, to appear in PLoS Comp. Biol.

Henrich, J. and Boyd, R. (2001). Why people punish defectors. weak conformist transmission can stabilize costly enforcement of norms in cooperative dilemmas. *J Theor Biol*, 208(1):79–89.

Hintze, A., Iliopoulos, D., and Adami, C. (2010). Stablility of strategies in Darwinian evolution. Manuscript in preparation.

Hofbauer, J. and Sigmund, K. (1998). *Evolutionary Games and Population Dynamics*. Cambridge University Press, Cambridge, UK.

Iliopoulos, D., Hintze, A., and Adami, C. (2010). Evolution of cooperation by natural selection. arxiv.org.

Moran, P. A. P. (1962). *The Statistical Processes of Evolutionary Theory*. Clarendon Press, Oxford.

Nakamaru, M. and Iwasa, Y. (2006). The coevolution of altruism and punishment: role of the selfish punisher. *J Theor Biol*, 240(3):475–88.

Nowak, M. (2006). *Evolutionary Dynamics*. Harvard University Press, Cambridge, MA.

Sigmund, K., Hauert, C., and Nowak, M. A. (2001). Reward and punishment. *Proc Natl Acad Sci U S A*, 98(19):10757–62.

Smith, J. M. (1982). *Evolution and the Theory of Games*. Cambridge University Press, Cambridge, UK.

Szabo, G. and Hauert, C. (2002). Phase transitions and volunteering in spatial public goods games. *Physical Review Letters*, 89(11).

Yamagishi, T. (1986). The provision of a sanctioning system as a public good. *Journal of Personality and Social Psychology*, 51:110–116.

Dynamic Resolution in the Co-Evolution of Morphology and Control

Joshua E. Auerbach[1] and Josh C. Bongard[1]

[1]Morphology, Evolution and Cognition Laboratory
Department of Computer Science
University of Vermont
Burlington, VT 05405
joshua.auerbach@uvm.edu

Abstract

Evolutionary robotics is a promising approach to overcoming the limitations and biases of human designers in producing control strategies for autonomous robots. However, most work in evolutionary robotics remains solely concerned with optimizing control strategies for existing morphologies. By contrast, natural evolution, the only process that has produced intelligent agents to date, may modify both the control (brain) and morphology (body) of organisms. Therefore, co-evolving morphology along with control may provide a better path towards realizing intelligent robots. This paper presents a novel method for co-evolving morphology and control using CPPN-NEAT. This method is capable of dynamically adjusting the resolution at which components of the robot are created: a large number of small sized components may be present in some body locations while a smaller number of larger sized components is present in other locations. Advantages of this capability are demonstrated on a simple task, and implications for using this methodology to create more complex robots are discussed.

Introduction

There are many reasons why it would be useful to have autonomous robots operating in our homes and offices. These range from freeing people from repetitive tasks to the ability to perform actions that humans are incapable of. However, with the exception of a few robots designed to accomplish simple tasks, the vast majority of autonomous robots currently in use operate only in factories and other highly structured environments. In order to make the migration out of the factories and into our everyday lives robots will need to be adaptive and exhibit intelligent behavior.

There has been much work in recent years in the area of embodied artificial intelligence (Brooks, 1999; Anderson, 2003; Pfeifer and Bongard, 2006; Beer, 2008) which has led to the conclusion that such intelligent behavior must arise out of the coupled dynamics between an agent's body, brain and environment. This means that the complexity of an agent's controller and morphology must increase commensurately with the task or tasks that it is required to perform. However, when designing complex autonomous robots it is often not clear how responsibility for different behaviors should be distributed across an agent's controller and morphology. A good example of this is that if a robot is solely tasked with moving over flat terrain while following a light source then wheels and a direct sensory motor mapping are an appropriate solution (Braitenberg, 1986), but if the robot must be able to navigate over varied terrains while performing more complicated tasks a more complex control strategy and/or morphology are required. This issue of scaling up morphological and control complexity has been a major obstacle in developing autonomous robots capable of operating in most real world situations.

Background

The only truly intelligent agents to have yet existed, as far as we are aware, are biological organisms. Therefore the only known pathway to creating intelligent agents is evolution by natural selection. Guided by this observation, the field of evolutionary robotics (Harvey et al., 1997; Nolfi and Floreano, 2000) attempts to realize intelligent agents by means of artificial evolution. Generally how this methodology works is that control policies for human designed or bio-mimicked robots are optimized to perform a desired task via evolutionary algorithms. This has allowed for the creation of robust, non-liner control strategies for autonomous agents that are not bound by the limits of human intuition. However, natural evolution does not operate on one part of an organism (brain) to the exclusion of others (body). In fact under evolution by natural selection any and all parts of an organism may be, and at some point in the past necessarily were, modified. This allows for the realization of organisms whose brains and bodies are co-optimized for specific ecological niches.

Luckily, artificial evolution is not necessarily limited to acting solely on a robot's brain or control strategy. Evolutionary frameworks in which the morphology and control of simulated machines are co-optimized in virtual environments are possible and indeed have been created, starting with Sims (1994) and followed by various other studies (Dellaert and Beer, 1994; Lund and Lee, 1997; Adamatzky et al., 2000; Mautner and Belew, 2000; Lipson and Pol-

lack, 2000; Hornby and Pollack, 2001a,b; Stanley and Miikkulainen, 2003; Eggenberger, 1997; Bongard and Pfeifer, 2001; Bongard, 2002; Bongard and Pfeifer, 2003). With this approach body plans and control policies uniquely suited for a machine's task environment may be found. This offers a substantial improvement over relying on body plans created by human designers who have inherent biases or copying animal body plans more suited to a given ecological niche.

The current work continues in this tradition while presenting several important advantages over previous approaches. First, the genomes of evolved agents are represented by compositional pattern producing networks (CPPNs) (Stanley, 2007), a form of indirect encoding that have been shown able to capture geometric symmetries appropriate to the system being evolved, are capable of reproducing outputs at multiple resolutions (Stanley et al., 2009), and have shown promise in producing neural network control policies for legged robots (Clune et al., 2009a,b). Second, through novel extensions of the CPPN outputs evolution can differentially optimize the resolution of the simulated robots such that a larger number of smaller sized components may be present in some body locations while a smaller number of larger sized components is present in other locations. To see why this is desirable consider evolving a creature capable of locomoting and grasping different objects. In this case evolution may choose to increase the resolution of the hands or grippers in order to achieve more fine grained control of the object to be grasped while at the same time using a lower resolution model of the trunk which will result in fewer components keeping the morphology from becoming unnecessarily complex and therefore providing faster simulations without sacrificing performance.

This paper extends the work presented in (Auerbach and Bongard, 2010) to allow for evolution of control as well as dynamic resolution as just discussed. The paper is organized as follows: the next section describers the CPPN encodings used, describes how they are evolved and presents how these encoding are used to grow actuated robots. Following that a description of two experiments is presented which compare this dynamic resolution method with a similar method lacking this ability. Some observations of how evolution makes use of the dynamic resolution capability are discussed, and finally some conclusions and directions for future work are presented.

Methods

This section presents a brief description of CPPNs and the evolutionary algorithm used to evolve them. This is followed by a description of the methods used for generating actuated robots from evolved genotypes. After this a description is presented of the fitness function used for evaluating these robots.

CPPNs

Compositional Pattern Producing Networks (CPPNs) are a form of artificial neural network (ANN). Unlike most ANNs where each internal node uses a form of sigmoid function, each internal node of a CPPN can have an activation function drawn from a diverse set of functions. This function set includes functions that are repetitive such as sine or cosine as well as symmetric functions such as gaussian. By composing these functions CPPNs can produce motifs seen in the majority of natural systems such as symmetry, repetition, and repetition with variation. It is important to note that these motifs come out of this encoding for free without the need for a human expert to explicitly enforce or select for them.

CPPN-NEAT

In this work the CPPNs are evolved via CPPN-NEAT (Stanley, 2007). CPPN-NEAT uses the NeuroEvolution of Augmenting Topologies (NEAT) method of neuroevolution (Stanley and Miikkulainen, 2001) to evolve increasingly complex CPPNs. An extension of CPPN-NEAT —HyperNEAT— has been used (Stanley et al., 2009; Clune et al., 2009a,b) to evolve traditional ANNs, where each node of the ANN is embedded in a geometric space and whose coordinates are fed to an evolved CPPN to determine the presence and weights of connections. In effect these connections are "painted" on to the network from the output patterns produced by the CPPN. As shown by Stanley et al. (2009) this has the crucial benefit that a CPPN evolved to produce the connectivity patterns of small ANNs can be requeried at a higher resolution to produce the connectivity patterns of larger ANNs without needing to re-evolve these large ANNs. Similarly as shown in (Auerbach and Bongard, 2010) it is possible to change the resolution at which CPPNs grow physical structures.

Growing Actuated Robots from CPPNs

In this work actuated robot morphologies and control strategies are grown from evolved CPPNs. Each robot is composed of many spherical cells which connect to each other either rigidly or via single degree of freedom rotational joints. For an example of robots produced in this way see Figure 1.

The growth procedure begins with a single cell, henceforth referred to as the root, with a predefined radius r_{init} located at a designated origin. A cloud composed of n points is cast around this cell with the n points being evenly distributed on the surface of the root sphere (all n points are at distance r from the center of the root). In the current work, n is restricted to 2, such that the points are directly opposite each other along the y-axis. In the coordinate system used here z is the vertical axis, and so the y-axis represents a horizontal axis that passes through the center of each cell. It is convenient to think of this as a cloud of points though, as is

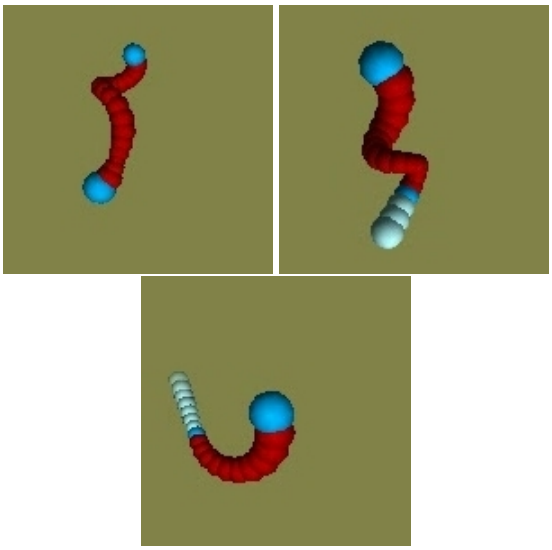


Figure 1: A few samples of robots evolved for directed locomotion.

the case in (Auerbach and Bongard, 2010), because in future work this restriction will once again be lifted allowing for a greater number of morphologies.

Once this cloud is cast, every point in the cloud is used to query a CPPN. The CPPN is queried by providing as input the Cartesian coordinates (x, y, z) of the point in question, the radius r_{parent} of the sphere to which it will attach ($r_{\text{parent}} = r_{\text{init}}$ when considering points around the root), and a constant bias input. These values are propagated through the CPPN to produce multiple output values. The first of these outputs is m. This output value can be thought of as a concentration of matter at that point, such that when m is over a certain matter threshold, T_{matter}, a cell will be placed at that point. The more that m exceeds the matter threshold the denser the cell placed at that point will be. This creates a continuum from no cell existing at that location up to having a very dense cell at that location with all intermediate levels of density in between being possible. The second of these outputs is a radius scaling factor r_{scale} which will determine the size of the cell to be added at that location.

Once the m and r_{scale} values have been determined for all n points in the cloud the points are sorted in descending order of the matter output m. The sorted points are then looped through and the algorithm considers adding a cell centered at each point in turn. Specifically a cell, centered at point p is added to the structure if (a) the output value of point p is above the threshold T_{matter} and (b) no other cell, besides the one to which this new cell will be attached (its parent) has previously been added to the structure with center located at distance $< r$ away from p.

1. **GrowRobot**(CPPN)
2. Initialize priority queue q, with priority based on cell density
3. Create cell c at origin with full density and radius r_{init}, add to morphology M and flag its coordinates 'discovered'
4. Enqueue c in q
5. WHILE $\sim q$.isEmpty
6. $c \leftarrow q$.front
7. Cast point cloud C centered at c
8. Initialize vector V of neighboring cells
9. FOR EACH point p in C
10. Query CPPN at p to get output values m and r_{scale}
11. Add p with values m and r_{scale} to vector V
12. Sort V by descending value of m
13. FOR EACH point p with value m in sorted vector V
14. IF coordinates of p not yet 'discovered'
15. Flag p 'discovered'
16. IF CanAdd(p,m,c,r)
17. Add cell centered at p with density $\propto m$ and radius $r = r_{\text{parent}} * r_{\text{scale}}$ to morphology M
18. Re-query CPPN at $\frac{c+p}{2}$ to get output values j, θ and Δ.
19. IF $j > T_{\text{joint}}$
20. Determine joint normal $\vec{n}$ from θ
21. Connect cell with 1-DOF rotational joint with normal $\vec{n}$, range $\propto j$ actuated by CPG with phase offset $\propto \Delta$
22. ELSE
23. Connect cell rigidly
24. Enqueue (p, v) in q

25. **CanAdd**(p, m, c, r)
26. IF $m > T_{\text{matter}}$ AND $\forall$ cells $d \in M, d \neq c$ dist(p,d) $\geq r$ AND p is within bounding cube
27. Return true
28. ELSE
29. Return false

Figure 2: **Grow Robot pseudo code.** The growth procedure starts with a root cell at the origin (line 3). Then, as long as there are cells in the queue to consider it takes the cell at the front of the queue, casts a point cloud around it and considers adding a cell at each point in turn (lines 5-17). A cell is added at a given point if all of the following hold: it does not conflict with a previously added cell, the CPPN outputs a value above the threshold T_{matter} when queried at that point, and the point is within the bounding cube (lines 25-29). If a cell is to be added the CPPN is queried once again to determine connectivity and control parameters (lines 18-23).

The radius r of a cell is determined from the radius of its parent r_{parent} and the output value r_{scale}. Specifically

$$r = \begin{cases} r_{\text{parent}} * r_{\text{scale}} & r_{\text{min}} \leq r_{\text{parent}} * r_{\text{scale}} \leq r_{\text{max}} \\ r_{\text{min}} & r_{\text{parent}} * r_{\text{scale}} < r_{\text{min}} \\ r_{\text{max}} & r_{\text{parent}} * r_{\text{scale}} > r_{\text{max}} \end{cases}$$

That is, the cell to be added will have radius equal to that of its parent scaled by a factor determined by the CPPN output capped by a minimum and maximum possible radius.

If a cell has been selected for addition to the robot the CPPN will be queried once more to determine connectivity and control parameters. In particular the CPPN will be fed the coordinates where a joint may be added: a cell centered at point p connecting to a parent cell centered at point p_{parent} may be connected by a single degree of freedom (DOF) rotational joint located halfway between p and p_{parent} ($\frac{p+p_{\text{parent}}}{2}$). These coordinates are input to the CPPN along with r_{parent} to retrieve additional outputs: a joint "concentration" j, an angle θ and a phase offset Δ.

If the output j exceeds a joint threshold T_{joint} the cell will attach to its parent with a 1-DOF rotational joint. The more j exceeds this threshold the greater the range of motion of the connecting joint will be. Similar to the matter case this creates a continuum from connecting rigidly when $j \leq T_{\text{joint}}$ to connecting via a joint with a very narrow range to connecting via a joint with a large range of motion.

If indeed a given cell will connect to its parent via a joint there are two more important properties of this connection to be determined. First, the direction of motion of this joint is defined by a normal vector $\vec{n}$. This vector will be normal to the axis $\vec{a}$ defined by the center of the cell and the center of its parent. To choose one vector out of the infinitely many such vectors the cross product of $\vec{a}$ and a default vector $\vec{d}$ is taken. This results in a single vector normal to $\vec{a}$ which is then rotated around $\vec{a}$ by angle θ. In this way all possible vectors normal to $\vec{a}$ may be used in constructing the joint and it is left up to the CPPN to output a single angle to choose a specific normal vector.

The second property to be determined in the case where a cell connects via a joint is what control signal drives the motor actuating this joint. In this work all motors are controlled by time dependent harmonic oscillators. A central sinusoidal oscillation is used, but each individual motor is allowed to be out of phase with this central control signal. The phase offset of each motor is determined by the final CPPN output Δ when queried at the joint's location. In this way the CPPN also determines the control policy of the robot being grown in addition to its morphology.

Once a cell is added to the structure and its connectivity and control have been determined it gets placed into a priority queue whose priority is based on its matter concentration m. When all points from the current cloud have been considered the algorithm takes the cell at the top of the priority queue and casts a point cloud around it, and this process continues until there are no valid possible points at which to place cells. Points are valid if they are within a bounding cube with side lengths l. This bounding cube constraint was imposed so that in the future it will be possible to physically fabricate the entire evolved robots within the confines of a 3D-printer. Figure 2 gives pseudo code for this growth procedure.

There are several reasons why it is desirable to have a growth procedure such as this. Merely querying CPPNs over a sampling of three-dimensional space may lead to disconnected objects. Even if all but one of these objects are thrown out much computational resources will have been wasted querying these regions of space. Additionally, imposing a grid over space to determine which points to query imposes a specific resolution on the morphology and thus removes much of the benefit of the dynamic resolution (radius) method used in this work because the spacing of the cells will have been predetermined by the grid.

Selecting for robots with desirable properties

This paper aims to demonstrate that CPPN-NEAT coupled with the growth procedure just presented is capable of evolving actuated robot morphologies and control policies for a given task. In particular the property selected for in this work is maximum directed displacement of the robot in a fixed amount of time.

To select for this property, an evolved virtual robot is placed in a physical simulator[1] for that set amount of time. The fitness of this robot (and hence its encoding CPPN) that CPPN-NEAT attempts to maximize is simply the y-coordinate of the robot's center of mass after the simulation completes subject to a few conditions. The first of these conditions is to prevent robots from exploiting simulation faults. There are a number of ways these faults could be avoided such as reducing the step size used in running the simulation, but this would lead to increased simulation runtimes. The technique used here is to throw out any solution where the robot's linear or angular acceleration exceed predefined thresholds by giving 0 fitness. The second condition is to prevent solutions where the robot moves by rolling on a subset of its cells. These solutions tend to be common but are less interesting than other solutions that may be found, therefore any robot that has a subset of its cells remain in contact with the ground for over 95% of the time is discarded and given a fitness of 0 once again.

Results

This section presents experiments comparing how the dynamic resolution method presented above performs is comparison to a similar method restricted to using cells with a

[1] Simulations are conducted in the Open Dynamics Engine (`http://www.ode.org`), a widely used open source, physically realistic, simulation environment

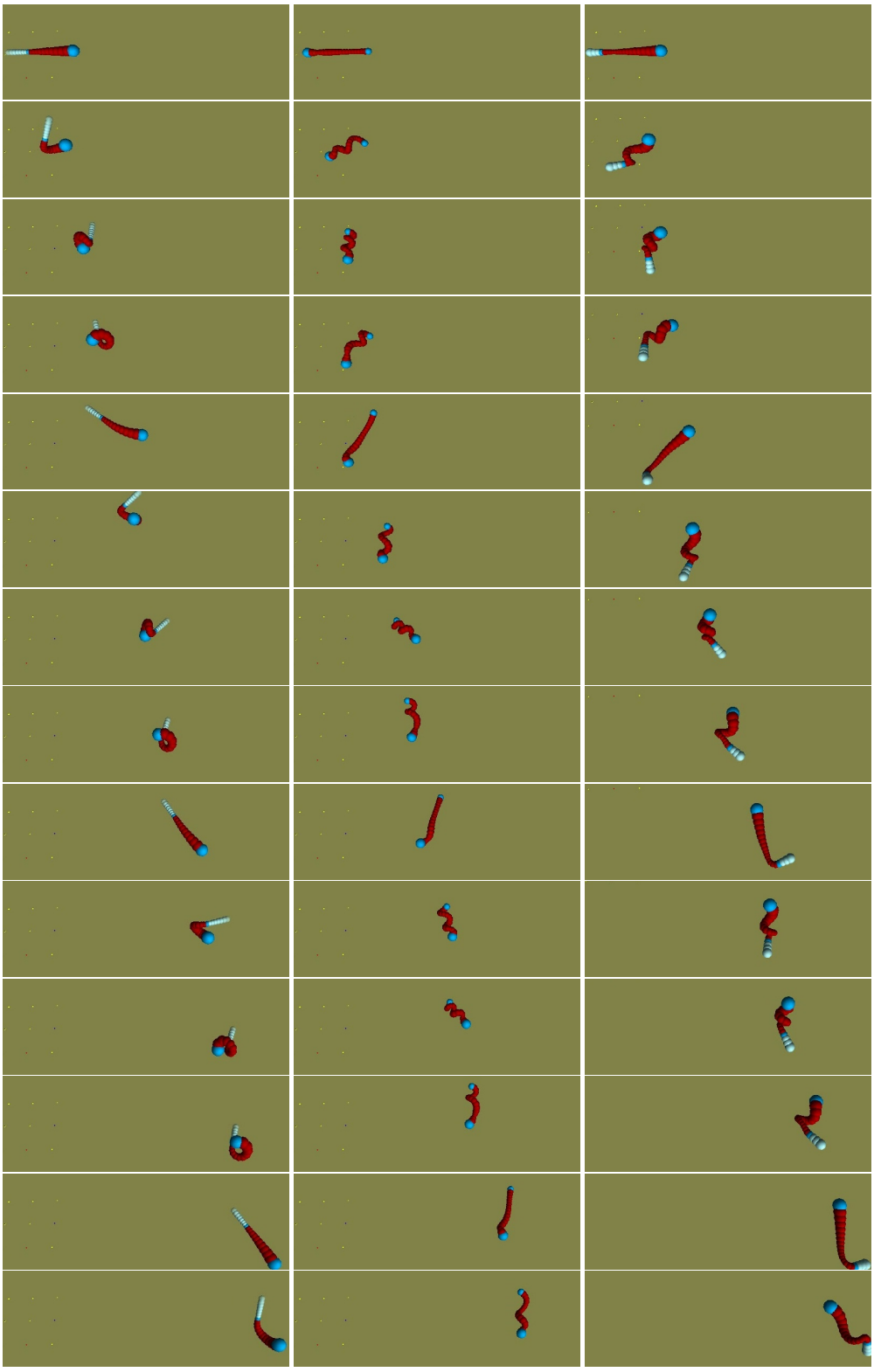


Figure 3: Each column shows the behavior of a different dynamic resolution robot evolved for directed locomotion (with time going from top to bottom). Three different robots are shown. Red cells are attached to two joints while the darker blue cells attach to a single joint. The lighter blue cells all connect rigidly. Enlarged pictures of each of these robots are shown in Fig. 1.

fixed radius. It should be noted that using a fixed radius in this case would be equivalent to omitting the growth procedure and merely querying the evolved CPPN over a gridded region of space and then taking those cells which connect to the cell at the origin as the resulting morphology, however as mentioned above this procedure would require more computational resources than using the growth procedure to accomplish the same result.

Specifically, two experiments are conducted each consisting of a set of 30 evolutionary trials. All experiments attempt to evolve simulated robots with CPPN-NEAT capable of directed locomotion using the fitness criteria presented above. Moreover, all experiments are configured to use a population size of 150, and run for 500 generations with each fitness evaluation given 2500 time steps. Additionally in all experiments the values T_{matter} and T_{joint} are both fixed at 0.7, and each cell of the structure is restricted to having its center initially located in interval $(0, [-2, 2], 0)$ (coordinates all in meters). Before being placed in the simulator the morphologies are translated vertically such that the largest component is resting on the ground. The CPPN internal nodes are allowed to use the signed cosine, gaussian, and sigmoid activation functions. All other parameters of the evolutionary algorithm are kept at the default values provided with the C++ implementation of HyperNEAT[2].

The trials in the first experiment grow structures using the dynamic resolution method introduced in this paper. In this case r_{init} was set to 0.1 meters, r_{min} set to 0.01 meters, and r_{max} set to 0.5 meters. Additionally the output value r_{scale} is normalized to the range $[0.5, 1.5]$; that is, a newly added cell can have radius at the most 50% larger and at the least 50% smaller than its parent. Figure 3 demonstrates the behavior of a few of the more successful robots to evolve in evolutionary trials in this experiment.

The second experiment is exactly the same as the first one, but it is restricted to growing robots composed of cells with a fixed radius. CPPN-NEAT is still used to evolve CPPNs which are used to grow the morphologies and control strategies under the procedure outlined above, but the r_{scale} output is not included in the CPPNs. In lieu of determining cell size from this output this experiment builds robots from cells all having radius $r_{\text{fixed}} = 0.1$ meters.

Discussion

One advantage of using the dynamic resolution method over keeping resolution fixed is that it allows evolution to explore a greater variety of possible solutions. The first evidence of this is observational. Looking at the behavior of the three robots shown in Figure 3 a variety of dynamics can be observed. The left most robot resembles a whip in that it has one thicker end and tapers off to a thinner end. Additionally we see that the thin end is rigid. This can be inferred from the light blue coloring of the cells at that end which represent cells that are not connected to any joint (while red cells connect to two joints and dark blue cells to a single joint). Scanning down the panels one can see that this rigid end is utilized as a paddle to propel the robot forward while curling over at the other end.

The middle robot on the other hand has no rigid connections. This robot moves by coiling and uncoiling to move itself in the desired direction. The right most robot has yet a different morphology and movement pattern than the other two. While it has one rigid end like the left most robot this end is composed of fewer spheres and actually includes cells that are larger than those in the middle of its body, flaring back out like a baseball bat. This configuration is actually the most successful one discovered and its movement pattern is different from the other two robots.

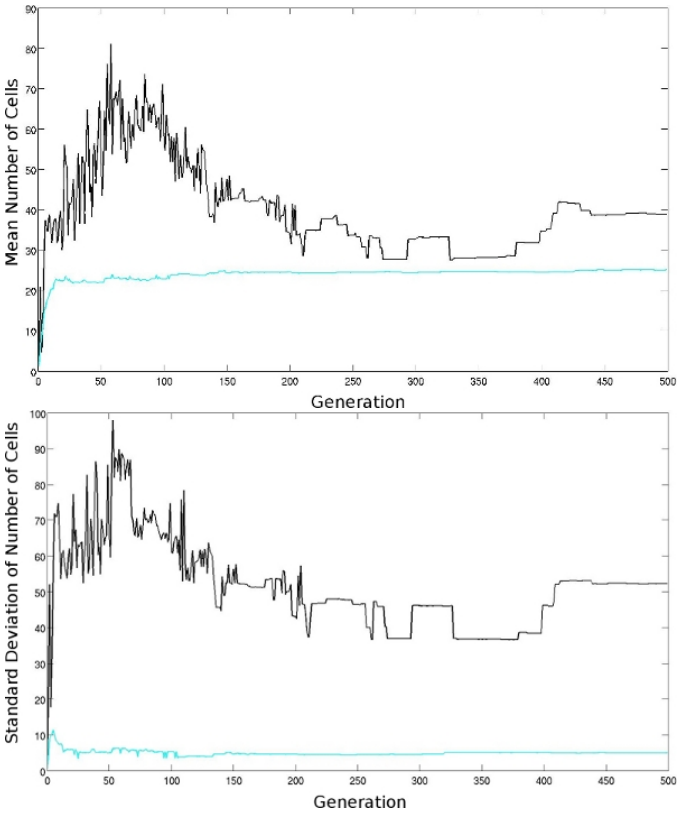


Figure 4: **Top**: Mean number of cells of best individual in each generation across the 30 evolutionary trials for the dynamic resolution set (black) and the fixed resolution set (light blue). **Bottom**: Standard deviation from the mean number of cells by generation.

Additional evidence of the dynamic resolution runs exploring a greater variety of morphologies is shown in Figure 4. The top part of this figure shows the mean number of cells used by the best individual from each generation across the

[2] Available at `http://eplex.cs.ucf.edu/hyperNEATpage/HyperNEAT.html`

30 evolutionary trials from both the dynamic resolution set and the fixed resolution set. The bottom portion of this figure shows the standard deviation from the means shown in the top. One can see here that the trials in the dynamic resolution set tend to explore morphologies with a large number of small cells early on, followed by exploring a fewer number of cells on average later on in the trials. However, while the fixed resolution robots tend to converge to a narrow range of cell numbers as exemplified by the constant mean and small standard deviation, the dynamic resolution robots continue to explore a wide array of different number of cells and cell sizes which can be inferred by observing that their standard deviation never comes back down.

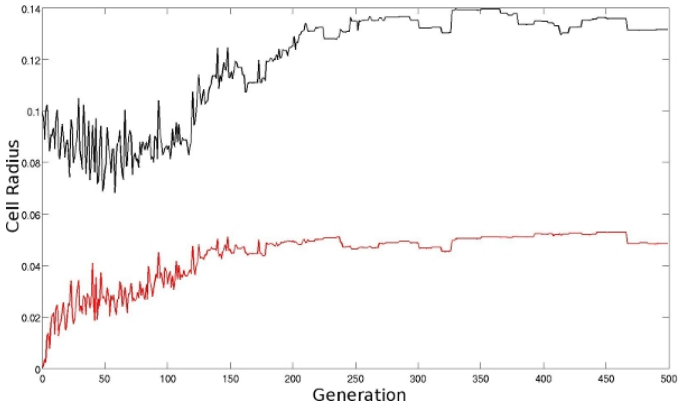


Figure 5: Mean (black) and standard deviation from the mean (red) of cell radii *within* each best of generation individual from the dynamic resolution set averaged across the 30 evolutionary trials.

This evidence is corroborated by Figure 5 which plots the mean and standard deviation of cell radii *within* each best of generation individual averaged across the 30 evolutionary trials. Here it is shown in a different way how the dynamic runs tend to explore smaller cell sizes early on in the evolutionary trials followed by larger cell sizes later. While this is the case on average, by looking at the standard deviations we see that as evolution progresses morphologies with a wide variety of cell sizes come into being (the standard deviation trends upwards). This means that the dynamic resolution runs are exploring the space of solutions with variable cell sizes which is not possible in the fixed resolution case.

Conclusion

This paper has demonstrated how one can implement a growth mechanism that can generate robots composed of variable sized components. This ability was then shown to be actually utilized by demonstrating how evolutionary trials that incorporate this dynamic resolution mechanism explore a greater variety of possible solutions than evolutionary trials that are restricted to constructing robots out of fixed sized components.

While it is not directly evident what performance advantage using dynamic resolution offers on a task as simple as the one utilized in this work, intuitively one can see the benefit of such a mechanism when generating more complex robots for more complex tasks. Specifically in any task that requires object manipulation it will be useful to adapt the component sizes of the parts of the morphology that will be in contact with external objects while not creating overly complex morphologies as would be the case if such a high resolution were employed for the entire robot. Additionally, it may not be possible to know the ideal component size *a priori*, and so using a dynamic resolution method such as this can help steer evolution towards constructing robot morphologies with the proper component sizes.

Much work remains to be done in exploring the possibilities of this methodology. The logical next step will be to relax some of the restrictions imposed in this work such as allowing robots to grow in arbitrary trajectories as opposed to along only a single axis. The authors additionally plan to tackle more complex tasks including object manipulation to test whether using dynamic resolution will result in the additional predicted advantages discussed here. This will require the use of more complex control strategies such as neural networks, and the inclusion of a mechanism for endowing the robots with sensors in order to close the control loop. The methods used here for generating joint and motor parameters via additional CPPN outputs seem promising and the authors plan to further leverage this technique for determining sensor and neuron positions and parameters.

References

Adamatzky, A., Komosinski, M., and Ulatowski, S. (2000). Software review: Framsticks. *Kybernetes: The International Journal of Systems & Cybernetics*, 29(9/10):1344–1351.

Anderson, M. (2003). Embodied Cognition: A field guide. *Artificial Intelligence*, 149(1):91–130.

Auerbach, J. E. and Bongard, J. C. (2010). Evolving CPPNs to Grow Three-Dimensional Physical Structures. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*. To Appear.

Beer, R. D. (2008). The dynamics of brain-body-environment systems: A status report. In Calvo, P. and Gomila, A., editors, *Handbook of Cognitive Science: An Embodied Approach*, pages 99–120. Elsevier.

Bongard, J. and Pfeifer, R. (2001). Repeated structure and dissociation of genotypic and phenotypic complexity in Artificial Ontogeny. *Proceedings of The Genetic and Evolutionary Computation Conference (GECCO 2001)*, pages 829–836.

Bongard, J. and Pfeifer, R. (2003). Evolving complete agents using artificial ontogeny. *Morpho-functional Machines: The New Species (Designing Embodied Intelligence)*, pages 237–258.

Bongard, J. C. (2002). Evolving modular genetic regulatory networks. In *Proceedings of The IEEE 2002 Congress on Evolutionary Computation (CEC2002)*, pages 1872–1877.

Braitenberg, V. (1986). *Vehicles: Experiments in Synthetic Psychology*. MIT Press.

Brooks, R. (1999). *Cambrian intelligence*. MIT Press Cambridge, Mass.

Clune, J., Beckmann, B., Ofria, C., and Pennock, R. (2009a). Evolving Coordinated Quadruped Gaits with the HyperNEAT Generative Encoding. In *Proceedings of the IEEE Congress on Evolutionary Computing*, pages 2764–2771.

Clune, J., Pennock, R. T., and Ofria, C. (2009b). The sensitivity of hyperneat to different geometric representations of a problem. In *Proceedings of the Genetic and Evolutionary Computation Conference*.

Dellaert, F. and Beer, R. (1994). Toward an evolvable model of development for autonomous agent synthesis. *Artificial Life IV, Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*.

Eggenberger, P. (1997). Evolving morphologies of simulated 3D organisms based on differential gene expression. *Procs. of the Fourth European Conf. on Artificial Life*, pages 205–213.

Harvey, I., Husbands, P., Cliff, D., Thompson, A., and Jakobi, N. (1997). Evolutionary robotics: the sussex approach. *Robotics and Autonomous Systems*, 20:205–224.

Hornby, G. and Pollack, J. (2001a). Body-brain co-evolution using l-systems as a generative encoding. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 868–875.

Hornby, G. and Pollack, J. (2001b). Evolving L-systems to generate virtual creatures. *Computers & Graphics*, 25(6):1041–1048.

Lipson, H. and Pollack, J. B. (2000). Automatic design and manufacture of artificial lifeforms. *Nature*, 406:974–978.

Lund, H. H. and Lee, J. W. P. (1997). Evolving robot morphology. *IEEE International Conference on Evolutionary Computation*, pages 197–202.

Mautner, C. and Belew, R. (2000). Evolving robot morphology and control. *Artificial Life and Robotics*, 4(3):130–136.

Nolfi, S. and Floreano, D. (2000). *Evolutionary Robotics: The Biology,Intelligence,and Technology*. MIT Press, Cambridge, MA, USA.

Pfeifer, R. and Bongard, J. (2006). *How the Body Shapes the Way We Think: A New View of Intelligence*. MIT Press.

Sims, K. (1994). Evolving 3D morphology and behaviour by competition. *Artificial Life IV*, pages 28–39.

Stanley, K., D'Ambrosio, D., and Gauci, J. (2009). A Hypercube-Based encoding for evolving Large-Scale neural networks. *Artificial Life*, 15(2):185–212.

Stanley, K. and Miikkulainen, R. (2003). A taxonomy for artificial embryogeny. *Artificial Life*, 9(2):93–130.

Stanley, K. O. (2007). Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines*, 8(2):131–162.

Stanley, K. O. and Miikkulainen, R. (2001). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10:2002.

Virus-host coevolution, killing the winner, and the Red Queen

Hywel T. P. Williams

Computational Biology Laboratory, School of Computing Sciences, University of East Anglia, UK
h.williams@uea.ac.uk

Extended Abstract

Viruses are the most abundant replicating entities on Earth, with an estimated 10^{30} virus particles in Earth's oceans alone (Suttle, 2005). Viruses play an important role in the marine carbon cycle, by viral mortality effects on the food web and by the `viral shunt' of material from higher to lower trophic levels (Fuhrman, 1999). Understanding the ecological and evolutionary interactions between viruses and their hosts is thus an important challenge if we are to understand the marine ecosystem and the global carbon cycle. Viruses are obligate parasites that replicate by taking control of infected cells and forcing them to create new virus particles, which are released during cell lysis. Each lysis (cell burst) event may release ~10^1-10^2 new virus particles. Growth rate asymmetries and time-lags during viral infection mean that virus-host population dynamics are hard to model as a standard predator-prey interaction. Furthermore, population-based and analytical approaches to modelling host-virus coevolution are problematic due to massive viral diversity and rapid evolution. Here I describe a novel individual-based simulation model of host-virus coevolution in a spatial aquatic environment. Individual host cells grow at a density-dependent rate up to a parameterised carrying capacity. Virus particles may adsorb to and infect host cells with which they come into contact. After a latent period during which virus particles are replicated inside an infected cell, lysis of the infected cell releases a large number of new virus particles into the environment. This asymmetric and time-lagged interaction results in boom-bust cycles of virus and host abundance, in which uninfected host populations grow until they are infected and destroyed, with associated exponential growth and collapse of viral abundance. To explore virus-host coevolution, the model focuses on the process of adsorption, in which virus tail-fibres bind to nutrient uptake receptors on the cell surface, allowing viral DNA to be injected into the cell. The `fit' between receptors and tail-fibres is thus an important locus for coevolution. The model represents this interaction in abstract form using evolvable bit-strings that represent nutrient uptake receptor configuration of host cells and tail-fibre orientation of viruses; infection occurs when these bit-strings match. This creates a coevolutionary pursuit in which hosts evolve novel strings to avoid infection, while viruses evolve strings that match their host. The need for host nutrient uptake receptors to fulfil their primary function of nutrient acquisition limits the ability of hosts to evade viral attack and creates an evolutionary trade-off between growth rate maximisation and defence. Results from the model support and quantify a theoretical prediction known as the `kill-the-winner' hypothesis (Thingstad et al, 1997), in which hosts that become abundant due to uptake efficiency become targets of viral attack. This negative density-dependent selection leads to increased host diversity. The coevolutionary dynamics of the model are characteristic of the well known `Red Queen' effect (Van Valen, 1973), whereby both viruses and hosts show continual evolutionary adaptation while maintaining broad constancy in relative fitness. Interestingly, the Red Queen effect is most pronounced in abundant host populations, while scarce host populations can achieve progressive fitness increase by improving uptake efficiency until they reach a critical abundance at which viral mortality becomes significant.

References

Fuhrman, J.A. (1999) Marine viruses and their biogeochemical and ecological effects. *Nature* 399: 541-548.

Suttle, C.A. (2005) Viruses in the sea. *Nature* 437:356-361.

Thingstad, T.F. (2000) Elements of a theory for the mechanisms controlling abundance, diversity, and biogeochemical role of lytic bacterial viruses in aquatic systems. *Limnology and Oceanography* 45(6): 1320-1328.

Van Valen, L. (1973) A new evolutionary law. *Evolutionary Theory* 1: 1—30.

Dynamics of Adaptive Radiation

Sergey Gavrilets[1,2,3]

[1]Department of Ecology and Evolutionary Biology
[2]Department of Mathematics
[3]National Institute for Mathematical and Biological Synthesis
University of Tennessee, Knoxville TN 37996 USA
gavrila@utk.edu

Abstract

Biologists have long been fascinated by the exceptionally high diversity displayed by some evolutionary groups (e.g., Darwin's finches, Anolis lizards, cichlid fishes of the African Great Lakes). Adaptive radiation in such clades is not only spectacular, but is also an extremely complex process influenced by a variety of ecological, genetic, and developmental factors and strongly dependent on historical contingencies. Using large-scale spatially and genetically explicit individual-based simulations, we identify a number of general patterns concerning the temporal, spatial, and genetic/morphological properties of adaptive radiation. Some of these are strongly supported by empirical work, whereas for others, empirical support is more tentative. In almost all cases, more data are needed. Future progress in our understanding of adaptive radiation will be most successful if theoretical and empirical approaches are integrated, as has happened in other areas of evolutionary biology.

References

Gavrilets, S. and A. Vose. (2005). The dynamic patterns of adaptive radiation. *Proceedings of the National Academy of Sciences USA* 102: 18040-18045.

Gavrilets, S. and A. Vose. (2009). Dynamic patterns of adaptive radiation: evolution of mating preferences. In Butlin, R.K., J. Bridle, and D. Schluter (eds.) *Speciation and Patterns of Diversity*. Cambridge University Press, pp. 102-126.

Gavrilets, S. and J. B. Losos. (2009). Adaptive radiation: contrasting theory with data. *Science* 323: 732-737.

Social Structure and the Maintenance of Biodiversity

Brian D. Connelly, Luis Zaman, Charles Ofria, and Philip K. McKinley

Department of Computer Science and Engineering
Michigan State University, East Lansing, MI 48824 USA
bdc@msu.edu

Abstract

Traditional ecological models assume well-mixed populations, where all members are equally likely to interact with one another. These models have been used successfully to explain competitive interactions; however, positive interactions such as intraspecific cooperation and interspecific facilitation cannot readily be captured. Previous work has highlighted the importance of spatial structure in explaining these behaviors as well as its role in maintaining biodiversity. These spatial structures have frequently been modeled using lattices, where all organisms have an equal number of interactions. Although these models capture the spatiality of interactions, natural populations are unlikely to follow such rigid patterns. There has been little work investigating the dynamics of populations with levels of social interactions that occur between these two extremes.

In this work, we investigate the dynamics of a 3-strategy non-transitive system in populations with different social structures. We first describe how extending the neighborhood of interactions in traditional lattice models diminishes a population's ability to maintain diversity. Populations are then moved to graphs where interactions are limited to cells within a defined distance of each other in Cartesian space. This method allows for a more fine-grained examination of the effects that increasing interactions have on maintaining diversity. Finally, we examine small world topologies and find that the introduction of random edges into the graph quickly disrupts the maintenance of diversity.

Introduction

The maintenance of biodiversity has long bemused ecologists. Under most models, the number of species that can coexist within a given ecosystem is significantly less than that observed in nature. Traditional differential-equation-based models, which assume well-mixed populations, often lead to the single species with the fastest growth outcompeting all others, as demonstrated in Kerr (2007). Further, these models have difficulty capturing cooperative interactions among organisms, as these behaviors have associated fitness costs, which slow growth rates and hinder a species' ability to compete.

Ecological models that incorporate spatial structure and local interactions, such as that developed by Durrett and Levin (1994), have been shown to more accurately describe the interactions of organisms. In these models, spatial structure is imposed by limiting the interactions of an organism to its surrounding neighbors instead of all organisms in the system. This can enable rare mutations to persist, especially if a number of these mutants exist together in close proximity. Further, if costly but beneficial behaviors are localized, the benefits of these interactions on its recipients may outweigh their costs, allowing them to spread in the population.

Allelopathic bacteria are a natural system that is frequently used to study the effects of spatial structure and cooperation, and localized interactions have been shown to contribute significantly to the coexistence of multiple strains (Kerr et al. (2002); Iwasa et al. (1998); Czárán et al. (2002)). In these systems, bacteria produce toxins called bacteriocins, which cause surrounding cells that do not express resistance to lyse. In the process, the toxin producer is killed. However, this act makes the newly-freed space and resources available to neighboring cells (ideally, the kin of the producer). Toxin production is genetically linked to resistance, so *producer* strains are also resistant to the toxin they produce. It is possible, however, to evolve resistance independent of production. Because such *resistant* strains do not pay the cost associated with production, they are able to grow faster than producer strains, while still maintaining their immunity. These strains, however, still grow more slowly than a *susceptible* strain that neither produces toxin nor is resistant. Therefore, in the absence of toxin, a resistant strain will be outcompeted by a susceptible strain. This combination of three strategies is considered a *non-transitive system*, where each strain dominates another strain, but is dominated by a third. These dynamics are captured in the classic rock-paper-scissors (RPS) game, where rock crushes scissors, scissors cuts paper, and paper covers rock.

Traditionally, spatial models of such systems have used lattices containing a fixed number of vertices, or *cells*, distributed uniformly in space. A cell is typically connected to its eight nearest cells (Moore neighborhood) by an undirected edge. To prevent boundary effects, periodic boundaries are often used, which form a toroidal grid by creat-

ing edges between cells on the periphery of the graph. This results in regular graphs in which each cell has the same number of neighbors, and the distance between any cell and its farthest neighbor is the same for all cells. This regularity indicates that any cell in the grid interacts with as many other cells as any other cell. Further, this distance property indicates that no matter where a dominant strategy begins, it must interact with the same minimum number of cells in order to spread throughout the population.

In this paper, we examine the role social structure plays in the maintenance of biodiversity by studying the above non-transitive system on graphs with differing vertex degrees, and hence different patterns of social interactions. We use the terms spatial- and social structure interchangably, as an organism's potential social interactions are limited to its neighbors. Our intent is to observe the dynamics of populations in the space between the regular graphs used in lattice models and well-mixed populations to determine at what point diversity breaks down. To accomplish this, we describe three models. First, we adopt the use of lattices, and the number of interactions is increased by expanding the radius of interactions surrounding each cell. This model gives us a high-level overview of the social structures in which diversity can be maintained. To achieve a more fine-grained control over a cell's interactions, we develop a method for creating graphs from a set of points in Cartesian space. Finally, we examine diversity on small world graphs, where interactions are primarily localized with the exception of some potential long-range interactions.

The spread of a two-strategy system on graphs with different properties was previously studied by Ohtsuki et al. (2006), who formulated a simple rule for the maintenance of diversity. Our work differs in that we are using a three-strategy system, and the benefits of a particular strategy are not fixed, but rather depend on the composition of each cell's neighborhood. More similar to our work, Károlyi et al. (2005) studied increases in social interactions through imperfect mixing of the spatial structure on a lattice. The primary difference is that their work used some measure of mixing, while the work presented here maintains fixed neighborhoods while differing the number of potential interactions. Finally, Buckley and Bullock (2007) used an information theoretic approach to investigate how space contributes to the complexity of a system. Although the focus of their work was different, complexity can play a large role on a population's ability to maintain diversity.

Methods

To study the effects of social structure on biodiversity, we developed a model based on graphs. This model consisted of *cells*, which were connected to each other by undirected edges, making both cells neighbors of each other. Interactions in this system were limited to a cell and each of its neighbors. In all experiments, populations consisted of 90 000 cells. Each cell exhibited one of four possible strategies:

1. *Susceptible* cells produced no toxin, nor were they resistant to toxin production by neighboring cells. Because susceptible cells did not pay any cost to maintain such behaviors, their growth was faster than other strategies.

2. *Producer* cells produced toxin which could kill neighboring susceptible cells. Additionally, since resistance is a trait that is genetically linked with production, producer cells were also resistant to toxin produced by neighboring producer cells.

3. *Resistant* cells can be viewed as producers that cheat. They reaped the benefits provided by adjacent producer cells without themselves paying the costs of toxin production. As such, they exhibited faster growth than producer cells, but slower growth than susceptible cells due to the added cost of resistance.

4. *Empty* cells had no effect on their neighbors. When chosen, an empty cell adopted the strategy of a randomly-selected neighbor.

We refer to these different cell types as "strategies", however they can easily be viewed as species, strains, or subspecies. At the beginning of each experiment, cells were randomly assigned one of these strategies.

Importantly, the growth of each strain was controlled by its rate of mortality. All strategies shared an intrinsic death rate, and the costs associated with resistance and toxin production manifested themselves as increases in death rate. This means that at any given time, a producer cell was more likely to die than a resistant cell, and a resistant cell was more likely to die than a susceptible cell. When a cell died, it became empty. For a cell to change from one strategy to another, it had to first die and then later adopt a neighboring strategy.

Populations were run for 10 000 *epochs*. During each epoch, 90 000 cells were chosen at random, and their states were updated asynchronously according to the rules described below. Following Kerr (2007), the probabilities of a resistant or producer cell dying during one of these updates were 0.312 and 0.333, respectively. Because the fate of a susceptible cell was tied to the presence of neighboring producer cells, its chance of death was modeled according to Equation 1, where Δ_S^0 is the intrinsic death rate for susceptible cells (0.250 in this work), τ is the toxicity of producers (0.65), and f_p is the fraction of producers in the cell's neighborhood.

$$\Delta_S = \Delta_S^0 + \tau f_p \tag{1}$$

Studies examining the maintenance of cooperative behaviors often compare the fitness cost of a strategy with the benefits it provides (e.g., Axelrod and Hamilton (1981); Ohtsuki

et al. (2006)). In most game theoretic models, these costs and benefits are explicitly defined in payoff matrices. In our model, the costs can be viewed as the increase in mortality seen by resistant and producer cells. In this sense, the cost of each strategy is fixed and continually incurred. However, due to the spatial nature of this and most other biological systems, the benefits depend on the current distribution of strategies in a cell's neighborhood. For example, toxin production may be highly beneficial when surrounded by susceptible cells, but have no benefit when all neighbors are producers. Likewise, resistance is beneficial in the presence of producer cells, but not in the presence of susceptible or resistant cells.

Lattice Models with Increasing Interactions

To examine the effects of increasing social interactions in populations, we began by adopting the lattice model as used in previous work (e.g., Iwasa et al. (1998); Czárán et al. (2002); Kerr (2007)). In these models, 90 000 cells were arranged in a 300x300 grid, with each cell interacting with its 8 surrounding neighbors. Periodic boundary conditions were used in order to prevent edge effects, producing 8-regular graphs.

As a simple method for expanding a cell's interactions, we first used lattices with increasing radii of interactions. That is, with radius 1, a cell was connected to its 8 surrounding neighbors. With radius 2, a cell's neighbors were the 24 cells within a 2-hop radius. This process continued with increasing radii until diversity was no longer maintained in the populations.

Cartesian Topology

Lattice models are well suited for studying spatial effects, but the geometric growth of neighborhood size is too fast and not necessarily representative of natural systems. In order to investigate the effects of increased neighborhood size on a finer scale, we moved from using lattice models to randomly-generated graphs that still accounted for the spatial relationships among cells.

To build these graphs, we uniformly placed 90 000 points in a unit Cartesian plane. Each point in this plane represented a cell in the world, and its neighbors consisted of the other points that fell within a circle of specified radius. Since a unit plane was used, the area of the circle was proportional to the expected number of points that it encompassed. That is, the area of a particular circle divided by the area of the plane represented the proportion of points which should, on average, fall within the circle. This construction was similar to that reported by Barnett et al. (2007), who examined how embedding space on random graphs affected various graph properties.

$$\frac{a}{1} = \frac{K}{|V| - 1} \tag{2}$$

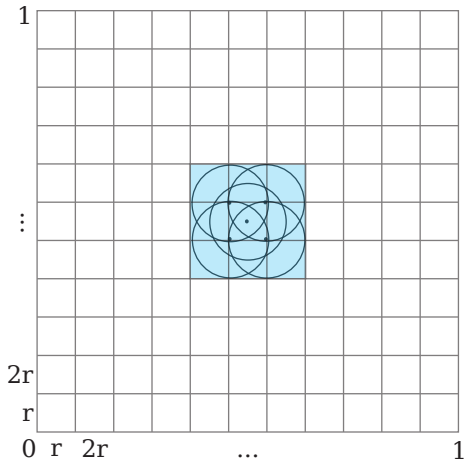


Figure 1: Unit Cartesian plane split into bins. Circles show the area where neighbors may fall, and the shaded region is the Moore neighborhood of the central bin.

In Equation 2, a is the area of a circle, 1 in the left-hand denominator represents the area of a unit plane, K is the expected average number of points within the circle (expected neighborhood size plus one for the cell the circle is centered on), and $|V|$ is the number of cells in the world, where $|V| - 1$ is the number of *potential* neighbors for a particular cell. Since a is the area of a circle with radius r, we can solve for the particular radius that will, on average, encompass K cells, as shown in Equation 3.

$$r = \sqrt{\frac{K}{\pi(|V| - 1)}} \tag{3}$$

This treatment also used periodic boundaries, which are achieved by allowing this circle to wrap around the edges of the plane. To reduce the running time for distance calculations, we partitioned the plane using a grid of two-dimensional *bins*, where each bin contained points that fell within a square area with side length r. Since the bins were $r * r$ sized, any point that may have fallen in a circle of radius r around a single point could not be outside of the immediate eight bin neighbors. Figure 1 shows the bin structure overlaying the Cartesian plane and several of the extreme circles with radius r, illustrating the fact that all neighboring points must fall within the Moore neighborhood of the bin. This method dramatically reduced the number of points considered as potential neighbors. Additionally, since edges were undirected and the neighbor relation was reciprocal, once the neighbors of a point had been found, that point no longer needed to be considered. This property allowed us to proceed bin-by-bin, eliminating all points contained within the bin from further consideration after exhausting it.

Figure 2 shows the average distribution of neighborhood

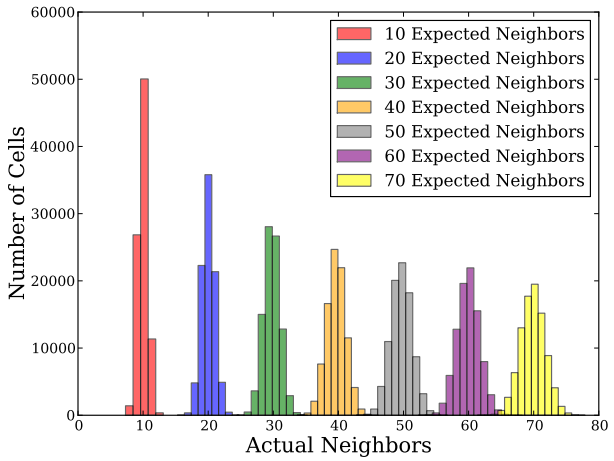


Figure 2: Histogram of the average neighborhood sizes from 20 replicates for different radii yielding expected neighborhoods from 10 to 70 cells in increments of 10

sizes when varying the expected number of neighbors from 10 to 70. The mean number of neighbors for each treatment was equal to the expected neighborhood size calculated. This method provides fine-grained control over neighborhood size while maintaining spatial interactions similar to those of lattices. Random graphs created in this way are arguably more representative of biological systems than lattice models, since the number of interactions for each organism in a population is not likely to be regular, even with explicit spatial structuring. This model allows for a distribution of neighborhood sizes around a specified expected value, as opposed to a fixed uniform neighborhood. We used this *Cartesian* method to generate random worlds with expected neighborhood sizes from 10 to 70 neighbors.

Biodiversity in Small World Networks

Finally, we examine the stability of these strategies in small world networks, which consist primarily of localized interactions with some long-range interactions, as defined by Watts and Strogatz (1998). These interactions often result in graphs where the number of interactions separating any two cells is surprisingly small. This property is familiar to those who have played the "Six Degrees of Kevin Bacon" game, where players are able to connect any person to actor Kevin Bacon through at most six social interactions, as described in Collins and Chow (1998). Although these networks likely do not capture the highly-localized interactions of microbial populations, they have been observed to capture several natural phenomena and may offer some insight into the maintenance of biodiversity in the presence of gene flow through these long-range interactions.

To construct these graphs, 90 000 cells were arranged on a ring, and each cell was connected to its nearest 8 neighbors. For each cell, additional interactions were created by probabilistically adding an edge to a randomly-chosen cell. At probability 0, these graphs were regular and had a diameter equal to the number of cells divided by the neighborhood size. At probability 1, the resulting graphs become random, mimicking interactions in well-mixed populations. For this work, we examine the effect that long-range interactions have on maintaining the biodiversity of this system.

Graph Metrics

In order to compare the structure of the different graphs used in this work, their clustering coefficients and diameters were calculated using the NetworkX package from Hagberg et al. (2008). The local clustering coefficient of a particular cell, defined by Watts and Strogatz (1998), measures how well connected that cell is in its particular network, and is defined in Equation 4, where i is the vertex (cell) in question, k_i is the number of neighbors of i, N_i is the set of i's neighbors, and E is the set of edges.

$$C_i = \frac{2|\{e_{jk}\}|}{k_i(k_i-1)} : v_j, v_k \in N_i, e_{jk} \in E \qquad (4)$$

A clustering coefficient of 0 indicates that none of a cell's neighbors are connected to each other, while a clustering coefficient of 1 indicates that all of a neighbor's cells are connected to one another. The graph's clustering coefficient is defined as the average of the clustering coefficients of its cells. This property is important in this system, as an area with a higher clustering coefficient allows for indirect interactions such as "the enemy of my enemy is my friend". The diameter of a graph is defined as the longest shortest path between any two cells. The diameter therefore provides an indication of how long it would take for a dominant strategy to spread to all cells in the graph.

For each of the treatments described above, 20 replicate populations were studied. Each replicate started with a different random seed, which led to differences in the structure of the graphs used in the Cartesian and small world treatments, the initial distributions of strategies, the stochastic processes of cell death, and the selection of random replacements for empty cells. These differences allowed populations to follow different trajectories.

Results

In all treatments, we found that diversity quickly declined with increasing neighborhood size. Increasing the radius of interactions in Moore graphs allowed us to observe this, however at a coarse granularity. The generated Cartesian graphs provided more insight into the maintenance of diversity, most importantly in intermediate ranges. Finally, small world graphs highlighted the significant effect that long-range interactions can have in these systems. Next, we discuss each of these results in detail.

Expanded Radius of Interaction on Lattices

As the radius of interaction was increased in lattices, diversity quickly diminished. As Figure 3 shows, at radius 3, several populations were unable to maintain all three strategies, while at radius 4, none did.

Due to the nature of this system, the loss of one strategy will break the non-transitivity of the system, which quickly leads to the loss of a second strategy. As an example in rock paper scissors, if no paper remained, rock would outcompete scissors, as rock no longer faced competition. Alternatively, if scissors were lost, paper would dominate rock.

As is common in this type of system, in cases where all three strategies were able to coexist, the strategies remained in patches, as is shown in Figure 4.

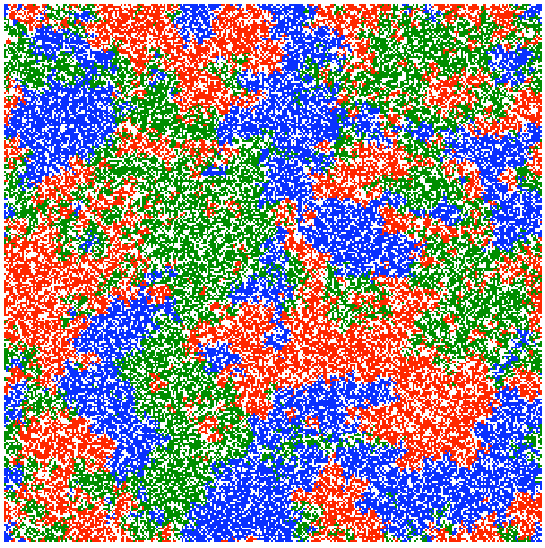


Figure 4: Spatial patterns observed in typical populations. When diversity is present, strategies exist in clusters. Sensitive cells are colored blue, resistant are green, and producer cells are red.

Although these experiments allowed us to investigate the role that the number of interactions has on diversity, the geometric increases in neighborhood size prohibited studying these features in detail. Table 1 highlights the effects that increasing the radius of interactions in a Moore neighborhood has on the structure of the resulting graphs. The sharp decrease in diameter allows a faster-growing strategy to spread quickly, outcompeting competitors regardless of their capabilities. This corresponds with Figure 3(d), where the sensitive strategy quickly eliminates the other strategies.

Increasing Interactions in Cartesian Space

The Moore topology provided only one treatment in which some runs maintained all three strategies while others collapsed to a single strategy, and the spread between conditions did not allow us to more closely examine the rate at

Table 1: Properties of Lattice Graphs Studied

Neighbors	Diameter	Clustering Coefficient
8 (r=1)	150	0.429
24 (r=2)	75	0.522
48 (r=3)	50	0.543
80 (r=4)	38	0.551

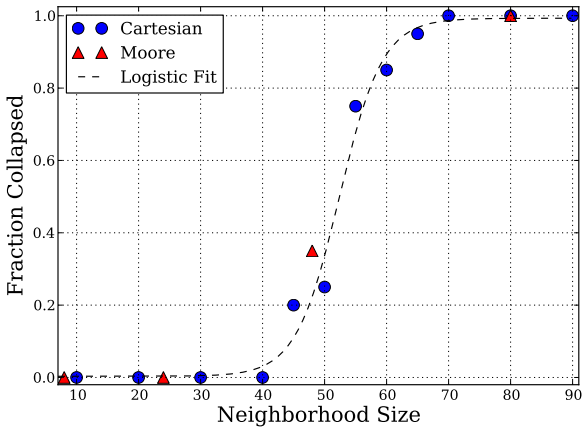
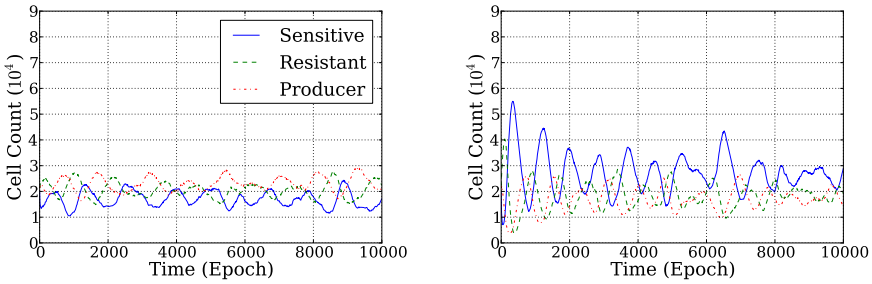


Figure 5: Fraction of runs (out of 20 replicates) that collapsed to a single strategy across different expected neighborhood sizes - F value 247.62 ($p \ll 0.001$), adjusted R^2 0.985

which biodiversity was lost. With just these data points, any number of possible curves could be drawn with equally good fit. The Cartesian topology allowed us to more closely investigate the effect of neighborhood size on the proportion of populations that lost biodiversity. The properties of the resulting graphs are listed in Table 2. It should be noted that several of the graphs generated with expected neighbor size of 10 were disconnected, as one might expect in a natural population with limited interactions. Figure 5 plots these proportions for a range of neighborhood sizes, where we focused on the range that produced intermediate loss of biodiversity. The logistic curve of best fit is highly significant, with an F statistic of 247.62 ($p \ll 0.001$), and an adjusted R^2 of 0.985.

The cell count plots for varying radii of this topology look similar to those in Figure 3, thus they are not included. Instead, we provide simplex phase planes for runs with different radii. A simplex phase plane depicts the proportion of strategies that were in the population at a given time and the trajectory the population took over all. The three corners of the triangle represent the three strategies, producer (P), sensitive (S), resistant (R), and the relative distance from each corner depict the proportion of the population the strategies



(a) $radius = 1$ (b) $radius = 2$

(c) $radius = 3$ (d) $radius = 4$

Figure 3: Strategy counts over time for different neighborhood sizes from sample runs. All three strategies remain in all replicates when neighborhood radius is 1 (a) or 2 (b). At radius 3 (c), diversity was maintained in 13/20 replicates, while diversity did not persist at radius 4 (d).

Table 2: Properties of Cartesian Graphs Studied

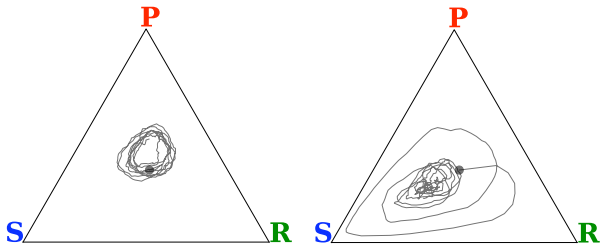
Expected Neighbors	Diameter	Clustering Coefficient
10*	45.5	0.585
20	83.25	0.587
30	57.25	0.588
40	51.5	0.589
50	59.0	0.588
60	53.0	0.586
70	49.0	0.587
80	45.0	0.587
90	38.0	0.591

comprise. Thus, a point in the center of the simplex would have equal frequency of each strategy, and a point at the P corner of the triangle would represent a population completely composed of producers.

Figure 6 depicts four simplex phase planes for different neighborhood sizes roughly corresponding to those from the Moore topology. The oscillatory dynamics observed in Figure 3 are also present in this topology, and are distinguishable by the circular path within the phase plane in Figure 6(a). Similarly, the large swings in cell counts with increased neighborhood sizes form the larger circular paths depicted in Figure 6(b) and 6(c).

Several runs that maintained biodiversity despite having larger neighborhood sizes (such as in Figure 6(c)) exhibited drastic transient dynamics, where the population of one or more strategies came dangerously close to being eliminated. It is these initial transient dynamics that stochastically led to population collapse as the mean neighborhood size increases. That is, in those runs that survive the transient dynamics, the population ends up in a *safer* region of phase space, one that is less susceptible to stochastic extinction. Of course, as the neighborhood size continues to increase, so does the magnitude of oscillations, and eventually all populations will collapse to a single strategy as the others are driven to extinction, as is shown in 6(d).

These transient dynamics are due to initial conditions where each cell strategy (including empty cells) is uniformly distributed throughout the world. As depicted in Figure 4, clusters of strategies emerge, and it is during the transition between the initial and self-organized states that populations often collapse. Essentially, we are starting the population in a random state with respect to clusters of strategies. While this approach biases the population towards larger cycles, it means our estimates for the collapse of biodiversity are conservative.



(a) 10 neighbors (b) 30 neighbors

(c) 50 neighbors (d) 80 neighbors

Figure 6: Simplex phase planes for Cartesian topology runs with increasing number of neighbors. The initial distribution of strategies is indicated with a dot.

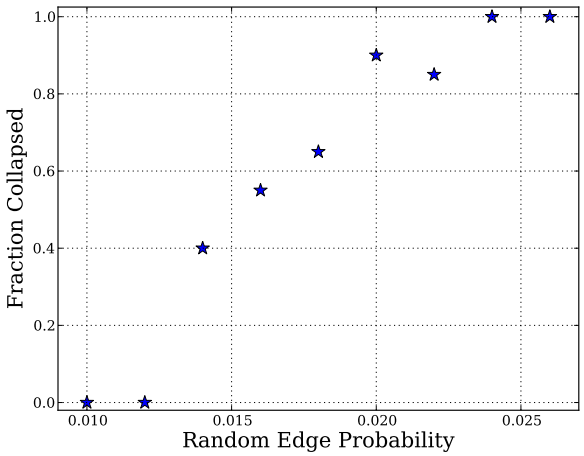


Figure 7: Fraction of runs (out of 20 replicates) that collapsed to a single strategy in small world networks with increasing probabilities additional random interactions

Interactions in Small World Graphs

Finally, we evaluated the effect of long-range interactions on diversity. As shown in Figure 7, even a small probability of such interactions had a dramatic effect on the system. We found that diversity quickly waned when the probability of adding these interactions was between 1% and 2%, which resulted in an additional 900 and 1800 pairs of interactions, respectively, on average. These additional interactions decreased the diameter of the resulting graphs to an average of 54.5 when the probability was 1% and 32.3 when the probability was 2%. The clustering coefficients for these configurations were uniformly 0.631 and 0.620, respectively. The difference in dynamics between systems at 1% and 2% edge creation possibility is shown in Figure 8.

Considering the small diameters typical of small world graphs, it is perhaps not surprising that diversity is quickly lost when long-range interactions are added. In the absence of these long-range interactions, the diameter of these graphs is 11 250. Adding additional edges with probabilities between 1% and 2% quickly shrank the diameters in these environments, which made the formation of clusters of strategies difficult. Nonetheless, these experiments provide a dramatic insight into how small increases in interactions can hinder diversity.

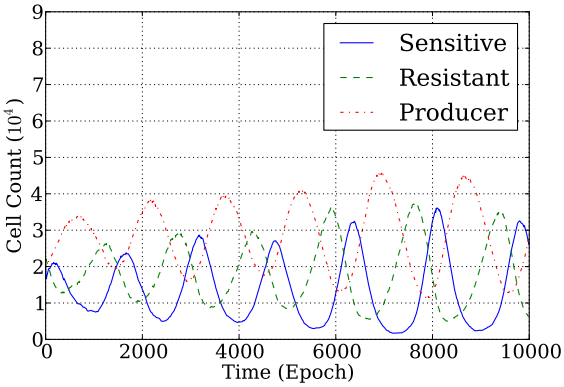
Conclusions

Understanding how the interactions among organisms affects biodiversity is critical to building a more complete picture of the forces that shape ecosystems. As such, this knowledge can inform conservation efforts and help to understand the ramifications of living in an increasingly-connected world.

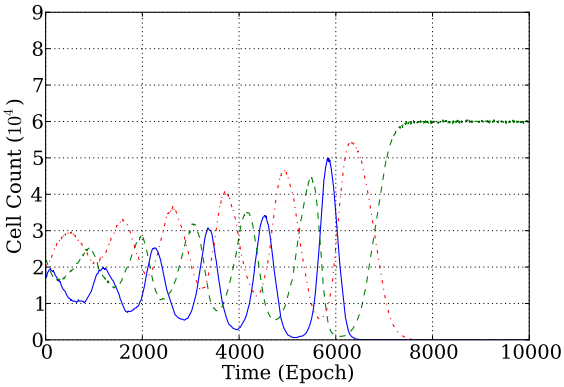
This work has demonstrated the strong effect social structure has on the maintenance of biodiversity in a model non-transitive system. Specifically, we have seen in three different models that as the number of interactions among cells increases, the magnitude of oscillations between the different strategies increases and quickly leads to the loss of diversity. Further, we have observed in small world networks that when a small number of long-range interactions are added, diversity is quickly lost, perhaps necessitating the use of kin discrimination or other mechanisms to promote the maintenance of diversity and cooperative behaviors in higher-order species.

Extending this model to include independent subpopulations and migration between them would allow the effects of gene flow to be examined, which could significantly change the dynamics of these populations. For example, this flow could enable the persistence of so-called “fugitive” species, which are not able to outcompete other species, but are able to persist through quick reproduction and constant migration. Although we claim that the long-range links in the small world networks studied in this work could represent gene flow between clusters of cells, this feature does not necessarily capture the effects of having multiple independent subpopulations.

It is worth noting that this work examined the maintenance of biodiversity from a purely ecological perspective. Allowing cells to mutate and change their strategies through the evolutionary process can have significant effects on a population’s diversity. Previous work has examined the effects on populations when mutations allow a cell



(a) $p = 0.01$



(b) $p = 0.02$

Figure 8: Strategy densities over time in small world networks. (a) At 1% probability of creating a random edge, biodiversity is maintained. (b) At 2%, diversity is lost.

to change its investment in a particular strategy (Prado and Kerr (2008),Czárán and Hoekstra (2009)) or to change its strategy completely (Mobilia (2010)). These works examined biodiversity in regular and well-mixed populations, respectively. Variations to social structure, as presented in this paper, could present different dynamics in evolutionary studies, and therefore lends itself to investigation in the presence of evolution.

Acknowledgments

This work benefited tremendously from the help of Ben Kerr, Christopher Klausmeier, Abdol-Hossein Esfahanian, Chris Strelioff, Ben Beckmann, Matt McGill, Anu Pakinati, the MSU HPCC staff, and three anonymous reviewers. Support was provided in part by National Science Foundation grants CNS-0915885, CCF-0820220, CNS-0751155, CCF-0643952; U.S. Army Grant W911NF-08-1-0495; DARPA Fundamental Laws of Biology; and by a Quality Fund Grant from Michigan State University. Luis Zaman was supported by an AT&T Labs Fellowship.

References

Axelrod, R. and Hamilton, W. (1981). The evolution of cooperation. *Science*, 211(4489):1390–1396.

Barnett, L., Di Paolo, E., and Bullock, S. (2007). Spatially embedded random networks. *Physical Review E*, 76(5):56115.

Buckley, C. and Bullock, S. (2007). Spatial embedding and complexity: The small-world is not enough. *Advances in Artificial Life*, pages 986–995.

Collins, J. and Chow, C. (1998). Its a small world. *Nature*, 393(6684):409–410.

Czárán, T., Hoekstra, R., and Pagie, L. (2002). Chemical warfare between microbes promotes biodiversity. *Proceedings of the National Academy of Sciences*, 99(2):786–790.

Czárán, T. L. and Hoekstra, R. F. (2009). Microbial communication, cooperation and cheating: Quorum sensing drives the evolution of cooperation in bacteria. *PLoS ONE*, 4(8):e6655.

Durrett, R. and Levin, S. (1994). The importance of being discrete (and spatial). *Theoretical Population Biology*, 46(3):363–394.

Hagberg, A. A., Schult, D. A., and Swart, P. J. (2008). Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference*, pages 11–15.

Iwasa, Y., Nakamaru, M., and Levin, S. (1998). Allelopathy of bacteria in a lattice population: competition between colicin-sensitive and colicin-producing strains. *Evolutionary Ecology*, 12(7):785–802.

Károlyi, G., Neufeld, Z., and Scheuring, I. (2005). Rock-scissors-paper game in a chaotic flow: The effect of dispersion on the cyclic competition of microorganisms. *Journal of theoretical biology*, 236(1):12–20.

Kerr, B. (2007). *Bacteriocins: ecology and evolution*. Springer.

Kerr, B., Riley, M., Feldman, M., and Bohannan, B. (2002). Local dispersal promotes biodiversity in a real-life game of rock-paper-scissors. *Nature*, 418(6894):171–174.

Mobilia, M. (2010). Oscillatory dynamics in rock-paper-scissors games with mutations. *Journal of Theoretical Biology*, 264(1):1–10.

Ohtsuki, H., Hauert, C., Lieberman, E., and Nowak, M. (2006). A simple rule for the evolution of cooperation on graphs and social networks. *Nature*, 441(7092):502–505.

Prado, F. and Kerr, B. (2008). The evolution of restraint in bacterial biofilms under nontransitive competition. *Evolution; international journal of organic evolution*, 62(3):538–548.

Watts, D. and Strogatz, S. (1998). Collective dynamics of "small-world" networks. *Nature*, 393(6684):440–442.

The Impact of Clonal Mixing on the Evolution of Social Behaviour in Aphids

John Bryden, Vincent A. A. Jansen

School of Biological Sciences,
Royal Holloway University of London,
Egham, TW20 0EX
United Kingdom
john.bryden@rhul.ac.uk

Extended Abstract

To clarify how selection operates in social aphids, and to disentangle direct and indirect fitness components, we present a model (Bryden and Jansen, 2010) of the life cycle of a typical colony-dwelling aphid (characterised in Figure 1). The model incorporates ecological factors and includes a trade-off between investing in social behaviour and investing in reproduction.

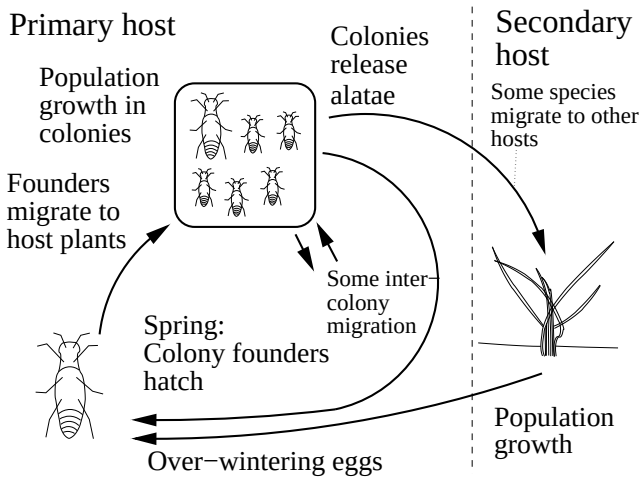


Figure 1: The typical aphid life cycle showing the movement of aphids between different habitats. The eggs hatch on the primary host producing *fundatrices* (colony founder aphids) which find suitable colony sites where they reproduce parthenogenetically for several generations. After a period of several weeks, the colonies open and *alatae* (winged aphids) are released. In most species, population growth continues on a second host before the aphids eventually lay eggs at the over-wintering site.

Altruistic or cooperative behaviour can be worthwhile for an 'acting' individual if the 'recipient' is more likely than an average member of the population to have the same trait. Conditions which are beneficial to such biased interractions can occur when there is population structure - i.e., when an individual only interracts with a subset of the population. These subsets can be observed in social aphid populations in the form of colonies which grow on plant leaves. These colonies produce soldier aphids that are prepared to die for the good of their colonies.

Reports of substantial clonal mixing measured in social aphid colonies (Abbot, 2009) seem, however, to rule out population structure as an explanation of this enigmatic insect's social behaviour. The mean proportion of immigrants per colony can be as high as 25% for some species.

Our model of the aphid life cycle approaches this problem by deriving a variant of Hamilton's (1964) rule. We are then able to demonstrate a simple relationship between the colony carrying capacity and immigration rates into colonies. The results indicate that the levels of clonal mixing reported are not inconsistent with social behaviour.

We discuss our model in terms of the evolutionary origins of social behaviour in aphids, social insects and arficial organisms in general. We also appraise our modelling approach, of deriving Hamilton's rule, in light of the results of the study.

References

Bryden, J. and Jansen, V. A. A. (2010). The impact of clonal mixing on the evolution of social behaviour in aphids. *Proceedings of the Royal Society B: Biological Sciences*, 277:1651-1657.

Abbot, P. (2009). On the evolution of dispersal and altruism in aphids. *Evolution*, 63:2687-2696.

Hamilton, W. D. (1964). The genetical evolution of social behaviour. i & ii. *Journal of Theoretical Biology*, 7:152.

Effects of Temporal Locality of Ecological Processes on Coevolution of Learning and Niche Construction

Reiji Suzuki and Takaya Arita

Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan
{reiji, arita}@nagoya-u.jp

Abstract

Roles of ecological processes in evolution are attracting much attention in evolutionary studies. Learning and niche construction are regarded as ecological processes that can affect the course of evolution directly or indirectly. However, the effects of mutual interactions between them on evolution are still poorly understood. Our purpose is to provide insight into the coevolutionary dynamics of learning and niche construction. For this purpose, we constructed a simple individual-based model in which individuals can perform both a niche construction of their shared environmental factor and an acquisition of the adaptive phenotype through their lifetime learning. In particular, we focus on the effects of the temporal locality of ecological processes, which is the degree of simultaneous occurrence of ecological processes performed by individuals. We report that a cyclic coevolution of genes for learning and niche construction can occur when the temporal locality of ecological processes is low.

Introduction

In the standard view of the modern evolutionary synthesis, organisms are basically regarded as passively evolving entities based on selection and mutations. However, there are two ways, based on ecological activities, for modifying the selection pressure as conceptualized in Fig. 1. One is for individuals to change their own phenotype called learning, and the other is to change their environmental condition, called niche construction (Odling-Smee et al., 2003). Recently, the roles of these ecological processes in evolution are attracting much attention in evolutionary studies called Evo-devo (West-Eberhard, 2003) or Eco-devo (Gilbert and Epel, 2009).

A wide variety of species have abilities to modify their own traits to make themselves more adaptive in their existing environments. It has been controversial how this ecological process, called individual learning, or ontogenetic adaptation based on phenotypic plasticity, can affect evolution indirectly. Since Hinton and Nowlan's pioneering work (Hinton and Nowlan, 1987), ALife researchers have focused on the Baldwin effect (Baldwin, 1896; Weber, 2003), which is typically interpreted as a two-step evolution of the genetic acquisition of a learned trait without the Lamarckian mechanism (Turney et al., 1996). An important finding is that the balances between the benefit and cost of learning can modify the shape of the fitness landscape, and can either accelerate or decelerate adaptive evolution (Paenke et al., 2009). A recent study has also discussed effects of the ruggedness of the fitness landscape (Suzuki and Arita, 2007). This study showed that if the shape of the fitness landscape is rugged, the learning can bring about a complex three-step evolution through the Baldwin effect.

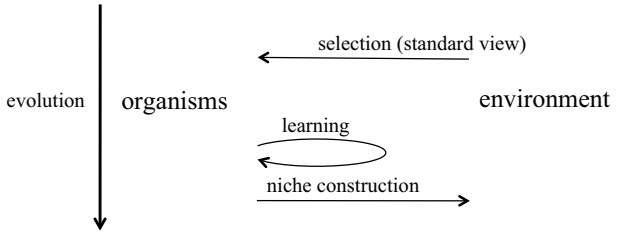


Figure 1: Two processes affecting the selection.

Niche construction is another ecological process, performed by organisms that modify their own niches or the niches of others, altering selection pressures through their ecological activities by changing their external environments (Odling-Smee et al., 2003). Such niche-constructing processes are observed in various taxonomic groups such as bacteria (decomposition of vegetative and animal matter), plants (production of oxygen), non-human animals (nest building) and humans (cultural process).

Recently, conditions for niche-constructing traits to evolve have been analyzed using theoretical or constructive approaches, in some cases leading to stable polymorphism (Laland et al., 1996), co-evolutionary dynamics of multiple species induced by their niche constructions (Suzuki and Arita, 2005), and so on. Self-regulation mechanisms of the environment caused by niche-constructing behaviors of individuals has also been investigated using several versions of the Daisyworld model (Harvey, 2004; Dyke, 2008).

So far, the effects of individual learning and niche con-

struction on evolution have typically been analyzed separately. We can interpret them as different processes in that the former is a change in the phenotype of the learning individual itself and the latter is the change in the surrounding environment of the niche-constructing individual. However, it is clear that both processes can interact indirectly with each other through changes in the relationship between the environmental conditions and individual phenotypes, suggesting that both processes can co-evolve in complex ways. That is, a niche construction can change an environmental factor, which can in turn modify the selection pressures on individuals that share the modified environment. Such an environmental change can further affect their learning process. Both gene-culture coevolution and language evolution appear to exemplify such situations, in that their mutual interactions were implicitly incorporated. In addition, it was recently pointed out that evolutionary developmental biology and niche-construction theory have much in common, in that both place emphasis on the role of ontogenetic processes in evolution, despite independent intellectual origins (Laland et al., 2008). However, as far as we know, there are still few approaches that have focused on interactions between learning and niche construction explicitly, in spite of their importance as ecological activities that can affect evolution.

Locality of ecological processes is an important factor for evolution of ecological traits in general, because it can affect the difference in the fitness between the performing individuals and the other individuals. One can distinguish two different kinds of locality: spatial and temporal locality of ecological processes. For example, it has been reported that the strong spatial locality of the effects of niche construction can contribute to the evolution of niche-constructing traits (Suzuki and Arita, 2006; Silver and Di Paolo, 2006), because it leads to difference in the fitness between the niche-constructing individuals and other, non-niche-constructing, individuals in distant locations. Temporal locality of ecological processes has received much less attention.

Our purpose is to consider whether and how learning and niche construction can interact with each other (Suzuki and Arita, 2009). For this purpose, we construct a simple individual-based evolutionary model in which the individuals can perform both a niche construction of their shared environmental factors and acquire an adaptive phenotype through their lifetime learning. Especially, we focus on the temporal locality of ecological processes, which is defined as the degree of simultaneous occurrence of ecological processes performed by individuals. There could be two extreme situations. One is a case in which individuals perform their ecological activities one by one, and the other is a case in which all individuals perform their ecological processes at the same time. The former corresponds to the situation in which the temporal locality is lowest, and the latter corresponds to when temporal locality is highest. It is not clear what aspects of these situations will contribute to the evolution of learning and niche construction. Through computational experiments with these two types of ecological processes, we show that temporal locality can strongly affect the evolutionary dynamics of learning and niche construction. Especially, we show that a cyclic coevolution of genes for niche construction and learning may occur in experiments with serial processes of ecological activities.

Model

Environment and genetic description of individuals

In our model, an environmental state shared by all N individuals is represented as a single real value e ($\in [0, 1]$). Each agent has a real-valued phenotype p ($\in [0, 1]$) whose initial value is determined by its genotype g_p ($\in [0, 1]$). The fitness contribution of p depends on e, and is determined by the following triangular shaped function $f(p, e)$:

$$f(p,e) = \begin{cases} 1 \quad |p \quad e|/L & \text{if } |p \quad e| \quad L, \\ 0 & otherwise. \end{cases} \tag{1}$$

Fig. 1 shows an example situation of the model. This function has a peak value 1 at e. Its value decreases linearly from the peak, and reaches 0 when the distance between p and e becomes L. Thus, the closer each agent's p is to e, the more fit it is.

Learning and niche construction

Each agent also has real-valued genes for learning g_l ($\in [0, 1]$) and niche construction g_n ($\in [\ \ 1, 1]$).

A learning process of each individual moves its phenotypic value p closer to e by (at most) g_l so as to increase its fitness contribution. Note that we assume that g_l can take a positive value because learning is a process that can increase the current fitness in general. The actual phenotypic value of an agent after its learning process p' is calculated from the equations as follows:

$$p' = \begin{cases} e & \text{if } |e \quad p| < g_l, \\ p \quad sgn(p \quad e) \times g_l & otherwise. \end{cases} \tag{2}$$

$$sgn(x) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{if } x = 0, \\ 1 & \text{if } x < 0. \end{cases} \tag{3}$$

This means that if the distance between the phenotype p of the focal individual and the environmental value e is smaller than its g_l, it can make its own p the same value as e completely. Otherwise, it can move its own p closer to e by g_l.

In addition, each individual can perform either positive or negative niche construction, which means that a niche construction can increase or decrease the fitness of the performing individual. This is because that niche construction is not always beneficial for performing individuals (i.e., there may

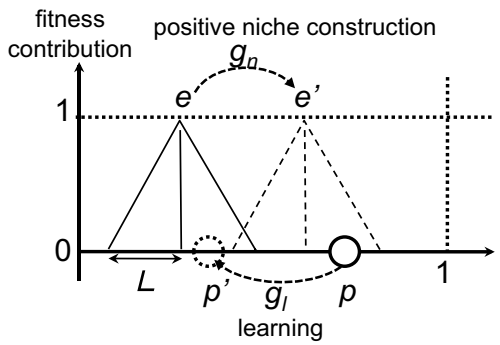


Figure 2: A learning and a niche construction in the proposed model.

be environmental pollution). If g_n of an individual is positive (or 0), its niche construction is positive and the actual environmental value e' after its niche-constructing process is calculated from the equation as follows:

$$e' = \begin{cases} p & \text{if } |e \quad p| < g_n, \\ e \quad sgn(e \quad p) \times g_n & otherwise. \end{cases} \tag{4}$$

On the other hand, if its g_n is negative, its niche construction is negative, and e' is calculated as follows:

$$e_{temp} = e \quad sgn(e \quad p) \times g_n, \tag{5}$$

$$e' = \begin{cases} 0 & \text{if } e_{temp} < 0, \\ e_{temp} \quad 0 \quad e_{temp} \quad 1, & \\ 1 & \text{if } e_{temp} > 1. \end{cases} \tag{6}$$

When g_n is positive, a niche construction moves e closer to its p (at most) by g_n. That is, a positive niche-constructing process is basically similar to a learning process except that it shifts the environmental value e rather than its own phenotype p. On the other hand, if g_n is negative, it makes e more distant from its p by $|g_n|$ within the range of the domain of $e \in [0, 1]$. If g_n is negative and p is exactly the same as e, we randomly add g_n or -g_n to e.

Ecological processes and evolution

In each generation, there are T sets of ecological processes, in each of which there are N steps. In each set, the individuals randomly decide which kind of ecological process to perform. We assume the two extreme types of temporal locality of ecological processes as follows:

Serial processes (low temporal locality) The individuals perform ecological processes serially in each set as shown in Fig. 3. In each set, an individual who has not done its ecological process yet in the current set is randomly selected and performs an ecological process. After the phenotypic value of the learning individual or the environmental value is modified, the fitness contribution of all individuals' phenotype are evaluated independently. This situation corresponds to the low temporal locality of ecological processes.

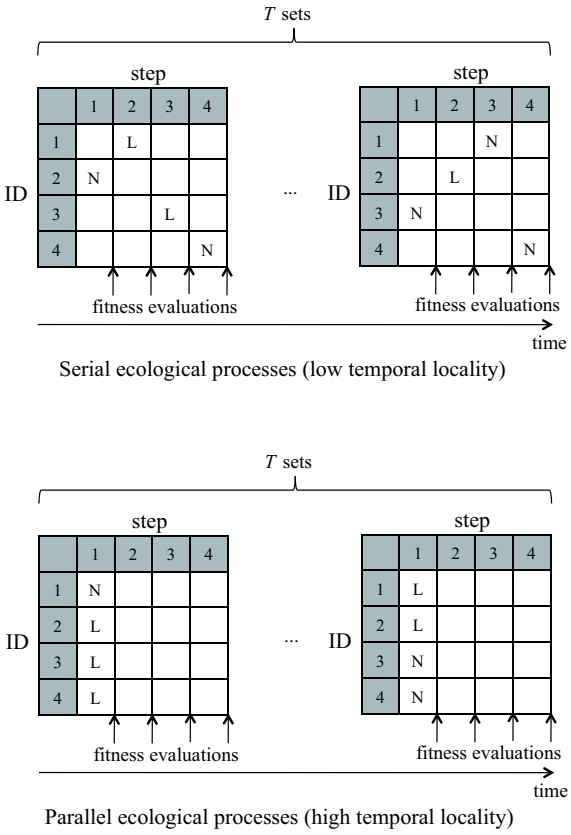


Figure 3: Serial and parallel processes of ecological activities. "L" or "N" represents an occurrence of learning or niche construction performed by an individual with the corresponding ID.

Parallel processes (high temporal locality) All individuals perform ecological processes at the same time at the initial step in each set as shown in Fig. 3. Before they actually modify the phenotypic and environmental values, they determine the amount of change in them using the current environmental value. Then, they update their phenotypic values, and the average amount of change in the environmental value determined by niche-constructing individuals is added to the current value. This situation corresponds to the high temporal locality of ecological processes.

The final fitness of each individual is defined as the average fitness contribution evaluated in all $T \times N$ steps. The evolutionary process is based on a "roulette wheel selection" according to fitness. For each gene, a mutation occurs with a

small probability p_m, which randomly determines its genotypic value.

The model incorporates a mechanism called ecological inheritance. This means that an environmental state can be passed on to the next generation. In this model, the value of e at the last step in the previous generation is used as the initial value in each generation.

Results

Serial processes of ecological activities

We examined evolution based on serial processes of ecological activities. We conducted evolutionary experiments for 2000 generations using the following parameters: N=250, T=300, L=0.1, p_m=0.05. In the initial population, the values of genotypes g_p, g_l and g_n were randomly decided within their domains, and the environmental state e was set to the intermediate value 0.5.

So as to clarify a possible dynamics of interactions between learning and niche-constructing processes, we focused on the evolutionary trajectory of g_l and g_n shown in Fig. 4. The horizontal axis is the average g_n and the vertical axis is the average g_l among all individuals at each generation. Although there were large fluctuations, we could see a cyclic evolutionary behavior of both indices, in which four typical states from (i) to (iv) (in Fig. 4) were traversed in a clockwise fashion. This means that the evolutionary trend of learning behaviors was strongly affected by existing niche-constructing behaviors and vice versa. Essentially, this evolutionary scenario was observed when N and T were relatively large and L was sufficiently small.

More detailed analyses, described later, clarified that the transitions between these states shown in Fig. 4 could be summarized as follows: $(i) \rightarrow (ii)$ the nearly neutral evolution of niche-constructing behavior, which brought about large fluctuations of the environmental state, $(ii) \rightarrow (iii)$ the adaptive evolution of learning behavior in dynamically changing environment, $(iii) \rightarrow (iv)$ the adaptive evolution of positively niche-constructing behavior, which made the environment stable, and $(iv) \rightarrow (i)$ the adaptive evolution of non-learnable individuals due to the implicit cost of learning (a kind of over-learning) in the stable environment. This cyclic behavior implies that the change in the stability of the environmental state arising from positive and negative niche constructions dynamically altered the balances between benefit and cost of learning behaviors. So as to clarify the universal mechanism of interactions between learning and niche construction inherent in this behavior, we investigated in more detail the dynamics of the observed evolutionary process by focusing on the effects of the environmental changes on evolution, and on the benefit and cost of learning.

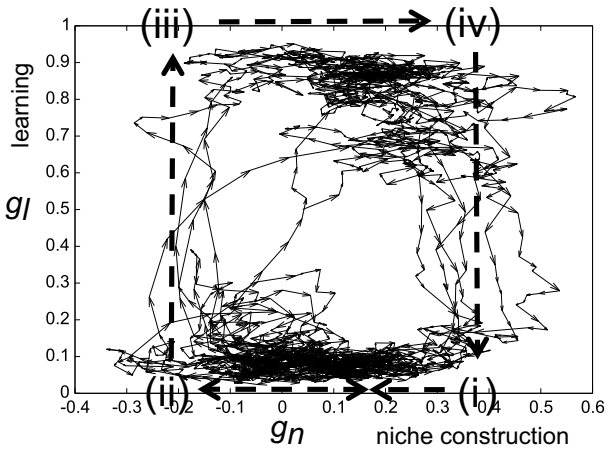


Figure 4: An example evolution of the average g_l and g_n through 2000 generations in the case of serial processes of ecological activities.

The detailed analyses of coevolution of learning and niche construction

Fig. 5 shows the evolution of the average and standard deviation of g_n, g_l, g_p and e through the initial 1000 generations in the same experiment as that shown in Fig. 4. Each value of g_n, g_l and g_p is derived from the values of all individuals in each generation, which means that their standard deviation represents their genetic variation in the population. Each value of e is derived from the values in all steps in each generation, which means that its standard deviation represents its temporal variation through steps in the generation.

Let us start from a situation around the state (i) near the 500th generation in Fig. 4 in which positively niche-constructing but non-learnable individuals dominated the population. As shown in Fig. 5, the standard deviation of g_p was relatively small (less than 0.2), which means that most individuals had basically the same, intermediate phenotypic value g_p. In this situation, there was nearly neutral selection pressure on the niche-constructing gene g_n because it could increase or decrease the fitness contribution of all individuals' phenotypes equally. Thus, the average g_n reached 0.0 and fluctuated around it because of the relatively small population size.

When the average g_n became negative as in the state (ii) at around the 600th generation, the environmental state e began to fluctuate by often taking either extreme value 0.0 or 1.0 and its standard deviation increased to higher values (around 0.4). Note that collective behaviors with positive and negative niche construction tend to make the environment state stable and unstable, respectively. In this case, the learnable individuals became adaptive because they can catch up with such environmental changes through their learning processes. Thus, the individuals with larger g_l and negative g_n

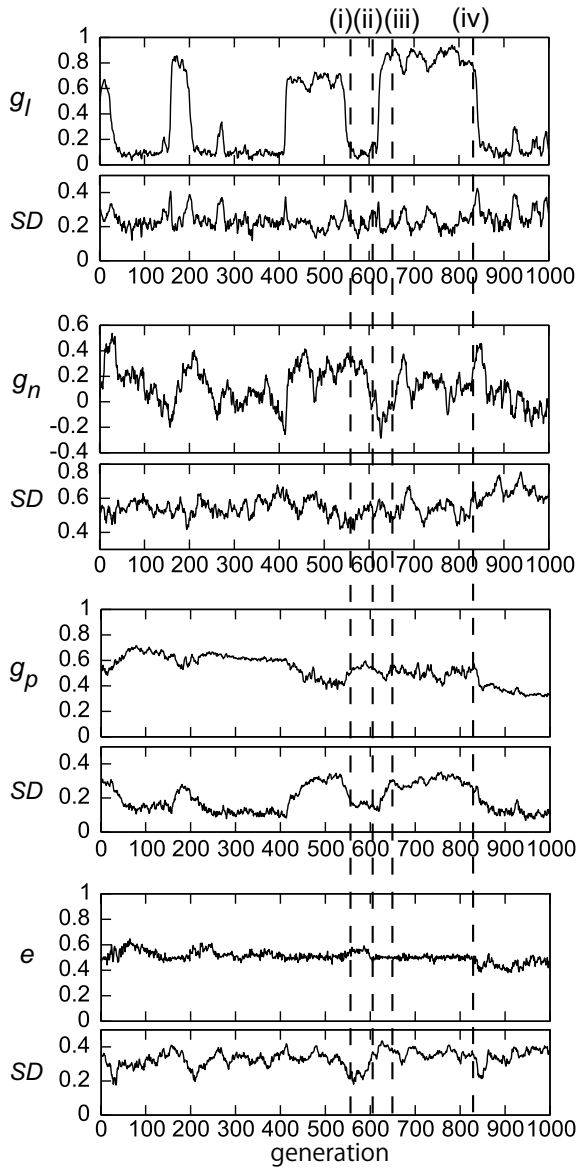


Figure 5: The evolution of the average and standard deviation of g_l, g_n, g_p and e through the initial 1000 generations in the case of serial processes of ecological activities.

rapidly occupied the population by keeping and even decreasing the stability of the environment. As a result, the average g_l increased quickly, and the population reached the state (iii) at around the 650th generation.

In the state (iii), individuals were changing their own phenotypic values dynamically so as to keep them closer to the fluctuating environmental values, which brought about a large variation among their phenotypic values. In such a situation, the positively niche-constructing individuals occupied the population because they can keep the environmental values close to their own phenotypes dynamically changed by learning. Thus, the population reached the state (iv) at around the 840th generation. During this period, the standard deviation of g_p remained high because learning reduced the selection pressure on the initial phenotypic values. This effect of learning on genetic evolution is sometimes called a hiding effect (Mayley, 1997).

Finally, when the number of such individuals increased enough, the standard deviation of the environmental value began to decrease and the environmental value come to fluctuate around the intermediate value (around 0.5) as a result of a "tug-of-war" between positively niche-constructing individuals. It should be noticed that the environmental value still takes the extreme values 0.0 or 1.0 even in this situation. If individuals with the larger g_l modify their own phenotype to either extreme value, that individual's fitness tends to become quite small in the remaining steps because the environmental value stays around the intermediate value or sometimes takes the other extreme value. Such a negative effect, caused by a kind of over-learning, could be interpreted as an implicit cost of learning, in that the learning behavior made the individual's fitness smaller than the one's with less ability to learn, even under the assumption of no explicit cost of learning, such as an energetic cost for performing the learning behavior itself. On the other hand, the individuals with the smaller g_l and the intermediate g_p can obtain relatively high fitness consistently by keeping its phenotypic value around the intermediate value. Thus, these positively niche-constructing individuals without learning could occupy the population quickly by keeping or even increasing the environmental stability. As a result, the population got back to the state (i).

Parallel processes of ecological activities

We also conducted the experiments under the condition of parallel processes of ecological activities. The experimental setting was the same as the one in the previous section except for updating process. Fig. 6 shows the evolutionary trajectory of g_l and g_n in an example trial, and Fig. 7 shows the evolution of the average and standard deviation of g_n, g_l, g_p and e through initial 1000 generations.

Fig. 6 clearly shows that the evolutionary dynamics of the population was quite different from the one with serial processes. There was no clear correlation between the genes for learning and niche-constructing traits. More specifically, Fig. 7 shows that g_n largely fluctuated between -0.2 and 0.2 through generations, which means that the evolution of niche-constructing trait was neutral in this case. This is expected to be due to the fact that niche-constructing behavior by an individual was cancelled, on average, by niche-constructing behaviors of others performed in parallel. Because this neutral evolution made the environment unstable, the learning behavior was always beneficial, and thus g_l stayed around 0.6, as shown in Fig, 7.

As a whole, under the condition of parallel processes of

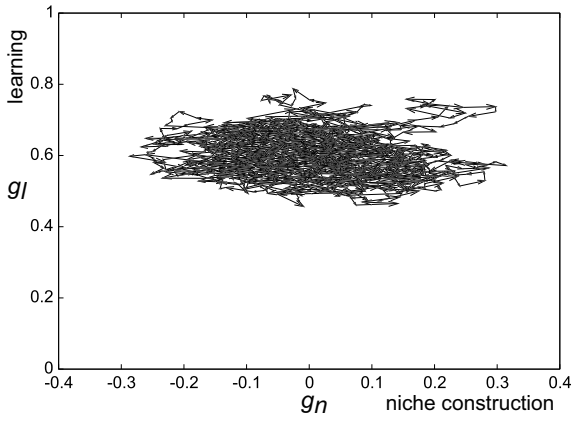


Figure 6: An example evolution of the average g_l and g_n through 2000 generations in the case of parallel processes of ecological activities.

ecological activities, there is basically no selection pressure on the niche-constructing trait, but its neutral evolution can cause selection pressure on the learning trait.

Conclusion

We studied the general nature of coevolution of learning and niche construction by using a simple evolutionary model of learning and niche-constructing genes. By comparing the cases with different temporal locality of ecological processes, we found that the adaptive benefit of learning and niche construction can change, and this strongly affects their coevolutionary dynamics. In the case of the low temporal locality of ecological processes, the positive effect of niche-construction directly affected the adaptivity of the niche-constructing individuals, which brought about a cyclic co-evolution of genes for learning and niche construction. The detailed analyses showed that the changes in the stability of the environmental state arising from positive and negative niche constructions is a key factor that dynamically determines the benefit and cost of learning behaviors. On the other hand, in the case of the high temporal locality, the neutral evolution of niche-constructing traits led to adaptive evolution of the learning trait.

One of the controversial topics that relates to this discussion is the interaction between evolution and learning in the context of language evolution, in that the fitness of each individual is determined by its linguistic niche composed of the other individuals' linguistic abilities based on learning. Yamauchi showed that the accumulated linguistic information through an ecological inheritance masks selection pressure on the innate linguistic traits acquired through the Baldwin effect (Yamauchi, 2007). Suzuki and Arita also showed that the Baldwin effect can occur repeatedly on dynamically changing fitness landscapes (linguistic niches) which arise from communicative interactions among individuals, and

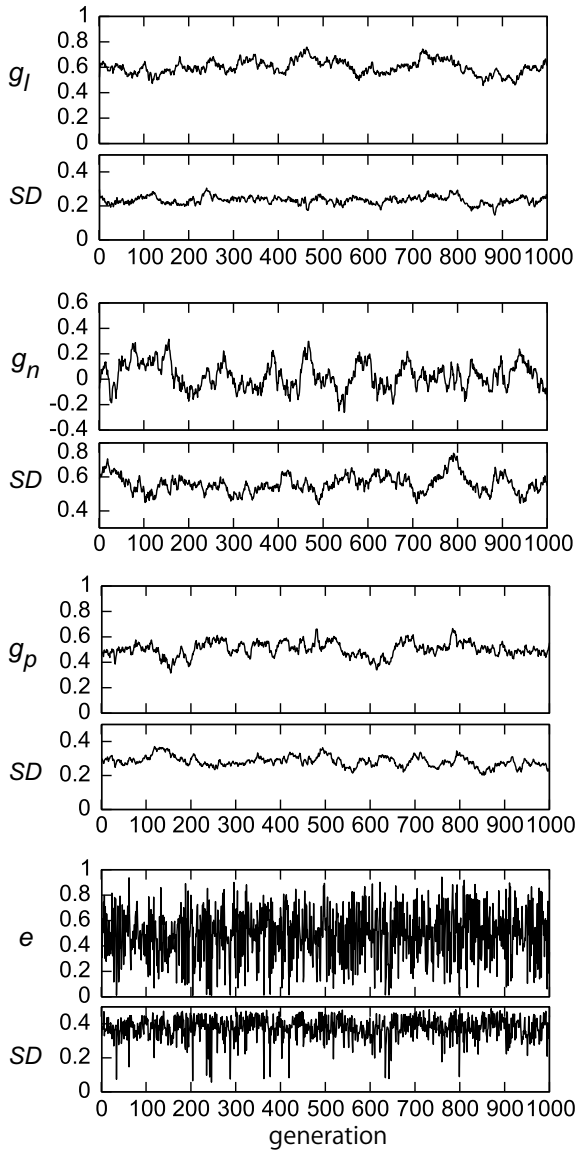


Figure 7: The evolution of the average and standard deviation of g_l, g_n, g_p and e through the initial 1000 generations in the case of parallel processes of ecological activities.

facilitates genetic evolution as a whole (Suzuki and Arita, 2008).

If we regard the horizontal axis in Fig. 2 as a space of possible language and each agent has a specific language determined by its p, the value of the environmental state e can be regarded as the most adaptive language due to the accumulation of its linguistic resources, which can contribute to its fitness increase, for example. In this case, a learning behavior corresponds to the process in which each agent changes its own language to a more adaptive one in its current linguistic environment, and a positive or negative niche construction corresponds to the production of linguistic re-

sources which can make its own language more or less adaptive. Our results with the low temporal locality of ecological activities imply that the intrinsic dynamics of coevolution of the abilities of learning language and constructing linguistic niche can bring about the dynamic and diverse aspects of language evolution even without any effects from external environments.

References

Baldwin, J. M. (1896). A new factor in evolution. *American Naturalist*, 30: 441–451.

Dyke, J. (2008). Entropy production in an energy balance Daisyworld model. In Bullock, S., Noble, J., Watson, R., and Bedau, M. A. editors, *Proceedings of Artificial Life XI*, pages 189–196. MIT Press.

Gilbert, S. F. and Epel, D. (2009). *Ecological Developmental Biology: Integrating Epigenetics, Medicine, and Evolution*. Sinauer Associates.

Harvey, I. (2004). Homeostasis and rein control: From daisyworld to active perception. In Pollack, J., Bedau, M. A., Husbands, P., Ikegami, T. and Watson, R. A. editors, *Proceedings of Artificial Life IX*, pages 309–314. MIT Press.

Hinton, G. E. and Nowlan, S. J. (1987). How learning can guide evolution. *Complex Systems*, 1: 495–502.

Laland, K. N., Odling-Smee, F. J. and Feldman, M. W. (1996). Evolutionary consequences of niche construction: A theoretical investigation using two-locus theory. *Journal of Evolutionary Biology*, 9: 293–316.

Laland, K. N., Odling-Smee, J. N. and Gilbert, S. F. (2008). EvoDevo and niche construction: Building bridges. *Journal of Experimental Zoology Part B: Molecular and Developmental Evolution*, 310B(7): 549–566.

Mayley, G. (1997). Guiding or hiding: Explorations into the effects of learning on the rate of evolution. In Husbands, P. and Harvey, I. editors, *Proceedings of the Fourth European Conference on Artificial Life*, pages 135–144. MIT Press.

Odling-Smee, F. J., Laland, K. N. and Feldman, M. W. (2003). *Niche Construction -The Neglected Process in Evolution-*. Princeton University Press.

Paenke, I., Kawecki, T. J. and Sendhoff, B. (2009). The influence of learning on evolution: A mathematical framework. *Artificial Life*, 15(2): 228–245.

Silver, M. and Di Paolo, E. (2006). Spatial effects favour the evolution of niche construction. *Theoretical Population Biology*, 70: 387–400.

Suzuki, R. and Arita, T. (2007). The dynamic changes in roles of learning through the Baldwin effect. *Artificial Life*, 13(1): 31–43.

Suzuki, R. and Arita, T. (2005). How niche construction can guide coevolution. In Capcarrere, M. S., Freitas, A. A., Bentley, P. J., Johnson, C. G. and Timmis, J. editors, *Proceedings of the Eighth European Conference on Artificial Life*, pages 373–382. Springer-Verlag.

Suzuki, R. and Arita, T. (2006). How spatial locality affects the evolution of niche construction. In Rocha, L. M., Yaeger, L. S., Bedau, M. A., Floreano, D., Goldstone, R. L. and Vespignani, A. editors *Proceedings of Artificial Life X*, pages 452–458. MIT Press.

Suzuki, R. and Arita, T. (2008). How learning can guide evolution of communication. In Bullock, S., Noble, J., Watson, R., and Bedau, M. A. editors, *Proceedings of Artificial Life XI*, pages 608–615. MIT Press.

Suzuki, R. and Arita, T. (2009). Coevolution of learning and niche construction. In Matsushita, M., Aruka, Y., Namatame, A. and Sato, H. editors, *Proceedings of the 9th Asia-Pacific Complex Systems Conference*, pages 95–99.

Turney, P., Whitley, D. and Anderson, R. W. (1996). Evolution, learning, and instinct: 100 years of the Baldwin effect. *Evolutionary Computation*, 4(3): 4–8.

Weber, B. H. and Depew, D. J. editors (2003). *Evolution and learning -The Baldwin effect reconsidered -*. MIT Press.

West-Eberhard, M. J. (2003). *Developmental Plasticity and Evolution*. Oxford University Press.

Yamauchi, H. (2007). How does niche construction reverse the Baldwin effect?. In e Costa, F. A., Rocha, L. M., Costa, E., Harvey, I. and Colutinho, A. editors, *Proceedings of the Ninth European Conference on Artificial Life*, pages 315–324. Springer-Verlag.

Aipotu: a Simulated Microworld Based on a Realistic Model of Gene Expression and Protein Folding.

Brian White[1]

[1]Biology Department;
University of Massachusetts, Boston
100 Morrissey Blvd
Boston, MA 02125 USA
brian.white@umb.edu

Extended Abstract

Aipotu ("utopia" reversed; pronounced "ay poh too") is an in silico microworld based on a highly realistic model of gene expression and protein folding.

Aipotian organisms are sexually-reproducing diploid organisms with DNA genomes. Their genes are expressed by transcribing from a promoter sequence until a transcription terminator is reached. The resulting pre-mRNA is then spliced based on intron start and end sequences. The mature mRNA is then translated using the standard genetic code. Proteins produced are folded on a 2-dimensional hexagonal lattice using realistic non-covalent interactions (hydrogen bonds, ionic bonds, and the hydrophobic interaction). The shapes and compositions of these proteins then determine their effect on the phenotype of the organism. In the current prototype version, the phenotype color is determined in a manner analogous to Green Fluorescent Protein: most proteins are colorless (white); a protein with a particular shape can be colored; the particular color depends on the amino acids present.

When the genomes of a population of these organisms are subjected to random mutation and selection based on color, the organisms show a variety of interesting evolutionary behaviors. These include: heterogeneity between runs with the same starting conditions; evolution of one color from another; loss of color in the absence of selection; convergent evolution of proteins with the same color; and evolution of colored from colorless starting proteins.

I have used Aipotu to teach evolution to undergraduate Biology students; I am currently evaluating its impact on students' understanding of evolution. Because it is based on a familiar and biologically reasonable underlying mapping of genotype to phenotype, it is likely to be more effective than other alife simulations used for teaching.

Because the underlying model involves realistic gene and protein sequences, Aipotu also has potential as a research tool. For example, it would be possible to explore and test the assumptions of molecular phylogeny by comparing the actual ancestry of Aipotian organisms with molecular phylogenetic reconstructions under different mutation regimes. Furthermore, because all of the key features of the underlying model of gene expression and mutation are variable, it will be possible to explore the evolutionary effects of changing these parameters. For example, currently, the mutations are only point mutations; the mutational spectrum could be expanded to include insertions, deletions, and gene duplications. It would be possible to add other structure to phenotype mappings besides color. For example, proteins with certain shapes could act as regulators of other genes, encode other phenotypes, or contribute to multi-protein pathways; entire organisms with hundreds of genes are possible. Finally, it would be possible to observe the effects of changing the genetic code or even the rules of protein folding. The underlying molecular genetic engine is fully functional; extensions are only limited by the imagination of the investigator.

Aipotu is open source and freely-available from http://intro.bio.umb.edu/aipotu/

Importance of the rearrangement rates on the organization of genome transcription

David P. Parsons[1,3], Carole Knibbe[2,3] and Guillaume Beslon[1,3]

[1]Université de Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205, F-69621, France
[2]Université de Lyon, CNRS, Université Lyon 1, LIRIS, UMR5205, F-69622, France
[3]IXXI, Institut Rhône-Alpin des Systèmes Complexes, Lyon, F-69007, France
guillaume.beslon@liris.cnrs.fr

Abstract

The organization of genomes shows striking differences among the different life forms. These differences come along with important variations in the way genomes are transcribed, operon structures being frequent in short genomes and the exception in large ones, while ncRNAs are frequent in large genomes but rare in short ones. Here, we use the digital genetics model "aevol" to explore the influence of the mutation rates on these structures, showing that their diversity can be accurately reproduced when varying the rearrangement rate. This result points us to the mutational burden hypothesis as one of the main explanation. In this view, a specific level of mutational robustness indirectly leads to genome and transcriptome streamlining.

Introduction

Genome organization is well known to be very different throughout the different domains of life. On one extreme, viral genomes can be as short as 400 base-pairs long (Gago et al., 2009) and are usually very dense, with nearly no non-coding sequences and a lot of overlapping genes, although some exceptions were reported (Raoult et al., 2004). Eukaryotic multicellular organisms on the other extreme, have very long genomes (billions of base-pairs), a huge proportion of which is composed of non-coding sequences. These differences come along with variations in the way the genome is transcribed: On the one hand, short genomes, that are almost entirely transcribed, are commonly transcribed into long RNAs that can contain several genes. In extreme cases, the whole genome can be transcribed in only a couple of RNAs (Zheng and Baker, 2006). On the other hand, long genomes usually give rise to short RNAs (after splicing), very few of which contain more than one single gene and most containing no genes at all. These non-coding RNAs have received a great deal of attention in the last few years (Ponjavic et al., 2007; Will et al., 2007), in particular microRNAs that are thought to play a major role in the regulation of gene expression (Mattick and Makunin, 2006; Kapranov et al., 2007).

What mechanisms are responsible for these variations in the organisation of transcripts and their relative importance remain open questions. Most efforts in these matters have been focused in understanding the evolution of operon structures. Operons are very interesting RNA structures where several coding sequences (often functionally-related) are packed together on a single RNA. Operons have been the subject of a great number of studies resulting in a set of theories that try to explain their assembly and maintenance. The following summarizes the most defended of these theories:

- The coregulation model is the original theory that came along with the discovery of the operon structure (Jacob et al., 1960). It claims that packing several functionally related genes together on the same RNA is beneficial because they share their regulation sites, which means that mutations on the promoter will preserve the relative expression levels of the gene products. According to this, genes within an operon should be likely to be functionally related.
- The selfish operon theory postulates that clustering genes for weakly selected functions together is beneficial for the genes themselves as it allows them to be horizontally transferred as a whole (fully functional unit), hence conferring a better advantage to the receiver than they would have provided individually (Lawrence, 1999). In the light of this theory, horizontal transfer is a necessary condition for the emergence of operons, which should contain preferentially genes that are functionally related.
- Finally, the mutational burden theory propounds that it is the mutational hazard that constrains the total amount of DNA: The larger the amount of excess DNA (intergenic DNA, 3' and 5' UTRs, ...), the higher the probability of a mutation (or rearrangement) to occur within it, potentially inactivating coding sequences or else disturbing the dynamics of existing genes. Following this idea, a population subject to high mutation rates will face a pressure for making genomes denser (Lynch, 2006; Knibbe et al., 2007). In some cases, this densification may reach a point where transcribed regions can actually merge or where a transcribed region can contain several translated sequences thus composing an operon. In extreme situations, genes can even share a part of their sequence and

overlap. This further reduces the size of the mutational target of the phenotype. This second order selective pressure for "streamlining" makes no assumption regarding gene function or horizontal transfer, operons should then be able to arise in the absence of transfer, putting together genes "working together" as well as functionally unrelated genes. In this view, the presence of operons must depend on the mutation rates, the selection strength and the population size.

Each of these theories have received evidence both for and against it. For instance, Pál and Hurst (2004) argue that the gene composition of operons in *E. Coli* is incompatible with the selfish operon theory but Hershberg et al. (2005) and Rensing (2002) suggest that it can explain at least some operon structures. As a matter of fact, it is very difficult to validate any of these models either *in vivo* or *in vitro* as the underlying processes are complex and act on a very long time scale. Comparative genomics approaches are a way to circumvent this difficulty. However, they are based upon the static snapshots of the contemporary sequences and have to *infer* their evolutionary past.

Artificial life and *in silico* simulations have shown to be very useful in such cases, providing us with insights into complex mechanisms and shedding light onto second-order pressures that would have been difficult to identify otherwise (Wilke et al., 2001; Adami, 2006; Misevic et al., 2006; Knibbe et al., 2007; Beslon et al., 2009). They provide a dynamic view of the evolutionary process in a reasonable time and with a near-to-absolute control over parameters. In this paper, we propose to investigate the organization of transcripts using a modelling-simulation approach.

Aevol: A digital genetics model

To study the evolution of genome structure, we have developed an integrated model, Aevol, that simulates the evolution of a population of N artificial organisms. Although a description of the model has already been published (see Knibbe et al. (2008) and its supp. mat.), we provide here an overview of the most important principles that are necessary to have a good understanding of the results presented here.

Overview

In Aevol, each artificial organism owns a genome whose structure is inspired by prokaryotic genomes. It is organized as a circular double-strand binary string containing a variable number of genes separated by non-coding sequences (figure 1). At the beginning of the run, all organisms are initialized with a same random sequence (of 5,000 base-pairs here) containing at least one gene. Genes are identified and decoded thanks to predefined signalling sequences and to an explicit transcription-translation process. Then, an abstract "folding" process gives rise to artificial "proteins" that are able to realize or deflect a particular range of abstract "biological functions". The interaction of all these proteins yields the set of functions the organism is able to perform, which will in turn be compared to an environmental target to determine how well-adapted this individual is.

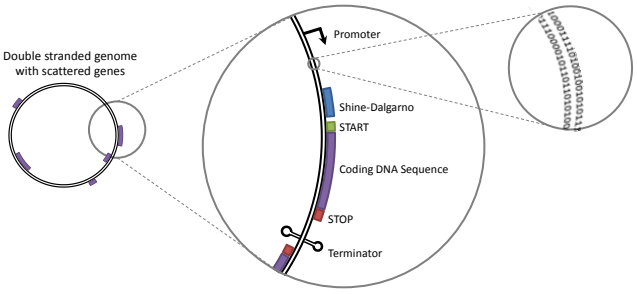


Figure 1: In Aevol, each individual owns a circular double-stranded binary genome upon which coding sequences are identified thanks to predefined signalling sequences: Promoters and terminators mark the boundaries of transcribed sequences and, inside these transcribed regions, coding sequences can exist between a START signal and an in-frame STOP codon (see figure 2 for the genetic code).

The best adapted individuals have higher chances of reproduction: At each generation, N new individuals are created by reproducing preferentially the best individuals of the parental generation which is then completely replaced. During the replication process, the chromosome can undergo different kinds of modifications: local mutations (single base substitutions, small insertions and small deletions), but also large chromosomal rearrangements (duplications, deletions, translocations and inversions).

From genotype to phenotype

The way a genotype is mapped to a phenotype in Aevol has been inspired by the prokaryotic transcription and translation processes. We defined a set of signalling sequences that enable us to identify the sequences that will be transcribed into RNAs and those that will be translated into proteins. Besides, a simple "folding" process was defined that allows us to interpret a protein's primary sequence as a set of "biological functions".

Transcription In prokaryotes, transcription initiates at particular sites, called promoters, where the RNA-polymerases recognize a consensus sequence to which they can bind and begin the RNA synthesis process. In Aevol, we defined a long consensus sequence, a promoter being a sequence whose Hamming distance d with this consensus is less than or equal to d_{max}. In the experiments presented here, the consensus was the 22-base-pairs (bp) sequence 0101011001110010010110 and up to d_{max} = 4 mismatches were allowed. This consensus sequence is long enough to

ensure that random, non-coding sequences have a low probability to become coding by a single mutation event. It is not a palindrome, meaning that a given promoter can initiate transcription on only one strand.

When a promoter is found, the transcription goes on until a terminator is reached. Terminators must be more frequent than promoters to limit the overlapping of transcribed sequences. Thus, if we had used a consensus sequence as for promoters, this sequence would have had to be very short. This would have forbidden this short motif to be present in any coding sequence, hence heavily constraining the evolutionary process. We therefore defined terminators as sequences that would be able to form a stem-loop structure, as the ρ-independent bacterial terminators do. In these experiments, the stem size was set to 4 and the loop size to 3, terminators thus had the following structure: $abcd***\bar{d}\bar{c}\bar{b}\bar{a}$, where $a, b, c, d = 0$ or 1.

The probability of a random 22-bp long sequence to be a promoter (*i.e.* of being at most 4 mismatches away from the consensus) is of roughly $1/460$, which means that the average distance between two promoters that can be expected in a random double-stranded sequence is of 230 bases. Terminators should be much more frequent: An 11-bp long sequence has a probability of $1/16$ to be a terminator.

The expression level e of an RNA is determined according to its promoter sequence. The closer the promoter is from the consensus, the higher the expression level: $e = 1 - \frac{d}{d_{max}+1}$. This modulation of the expression level models in a simplified way the basal interaction of the RNA polymerase with the promoter, without additional regulation. It provides duplicated genes with a way to reduce temporarily their phenotypic contribution while diverging toward other functions. It also induces a link of co-regulation between the coding sequences of a same transcribed region, which is a necessary property to test the coregulation hypothesis.

Translation Transcribed sequences (RNAs) do not necessarily result in a protein. The translation process of an RNA takes place when a Shine-Dalgarno-like sequence is found, followed, a few base-pairs away, by a START codon (see genetic code on figure 2). We thus defined the translation initiation signal as the motif $011011***000$. Whenever this signal is found, the following sequence is read three bases (one codon) at a time until the termination signal (the STOP codon 001) is found on the same reading frame. Each codon lying between the initiation and termination signals is translated into an abstract "Amino-Acid" using an artificial genetic code, therefore giving rise to the protein's primary sequence (figure 2).

As in real organisms, and because we read our genetic sequences three bases at a time, genes can be found on six different reading frames (three on each strand), giving the possibility for the organisms to evolve out-of-phase overlapping genes, which are commonly found in bacterial operons

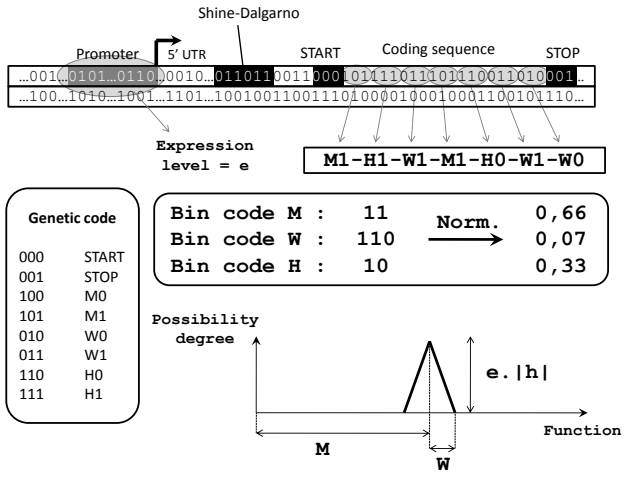


Figure 2: Overview of the transcription-translation-folding process in Aevol. Transcribed sequences are those that start with a promoter (consensus sequence) and end with a terminator sequence (hair-pin), not shown on the figure. Coding sequences (genes) are searched within the transcribed sequences; They begin with a Shine-Dalgarno-START sequence and end with a STOP codon. An artificial genetic code (right) is used to convert a gene into the primary sequence of the corresponding protein and a "folding process" enables us to compute the metabolic activity of this protein (functional abilities).

(Johnson and Chisholm, 2004; Palleja et al., 2008).

Protein "folding" and phenotype computation To model the activity of proteins and the resulting phenotype, we defined a simple "artificial chemistry" (Dittrich et al., 2001) that describes the organism's metabolism in a mathematical language. In our simplified artificial world, we assume that there is an abstract, one-dimensional space $\Omega = [0, 1]$ of possible metabolic processes (that is, in this model, a metabolic process is just a real number). In this "metabolic space", each protein is involved in a subset of processes (either realising it or preventing other proteins from realising it) which is described using the fuzzy set formalism: A given protein can be involved in a metabolic process with a possibility degree lying between 0 and 1. A protein is thus fully characterized by a mathematical function that associates a possibility degree to each metabolic process. For simplicity, we use piecewise-linear functions with a symmetric, triangular shape (figure 2). In this way, only three numbers are needed to characterize the metabolic activity of a protein: The position m ($m \in \Omega$) of the triangle on the axis, its half-width w and its height h (positive when realizing a function, negative when inhibiting it). This means that the protein contributes to the range $[m-w, m+w]$ of metabolic processes, with a preference for the processes closest to m

(for which the highest efficiency, h, is reached). Thus, various types of proteins can co-exist, from highly efficient and highly specialized ones (small w, high h) to polyvalent but poorly efficient ones (large w, low h).

In this framework, each protein's primary sequence is decomposed into three interlaced binary subsequences that will in turn be interpreted as the values for the m, w and h parameters. For instance, the codon 010 (resp. 011) is translated into the single amino acid $W0$ (resp. $W1$), which means that it contributes to the value of w by adding a bit 0 (resp. 1) to its binary code. Small mutations in the coding sequence (substitutions, indels, possibly causing frame shifts) will change these parameters, resulting in a modification of the protein's metabolic activity.

Once all the proteins encoded on the genotype of the organism have been identified and characterized, their activities are combined into a fuzzy set representing the individual's phenotype P, using Lucasiewicz' fuzzy operators. This phenotype indicates to what extent the individual can realize each metabolic process in our abstract metabolic space.

Environment, adaptation and selection

In Aevol, the environment is represented by a phenotypic target: The fuzzy set E defined on Ω that represents the optimal degree of possibility for each "biological function". To evaluate an individual, we compare its phenotype P to the optimal phenotype E. The "metabolic error" g is computed as the geometric area between these two sets (figure 3). The lower the metabolic error, the better the individual. This measure penalizes both the under-realization and the over-realization of each function.

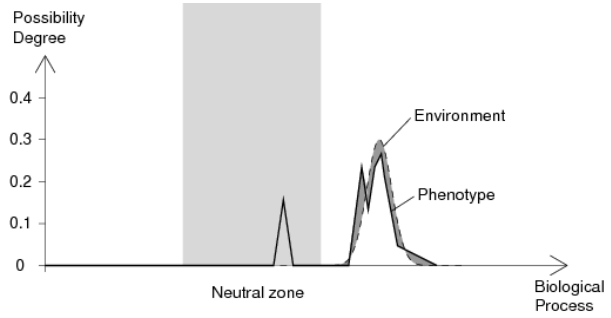


Figure 3: Measure of an individual adaptation. Dashed curve: Environmental target E. Solid curve: Phenotypic distribution P (resulting metabolic profile obtained after combining all the proteins). Dark grey filled area: Metabolic error g. The part of the phenotype that is located inside the neutral zone (light grey) is not considered as being part of the gap. This allows for the evolution of non-essential genes.

In the current version of Aevol, the population size is constant (here $N = 1,000$ individuals) and the population is entirely renewed at each generation. A probability of reproduction is assigned to each individual according to its metabolic error and a multinomial drawing determines the actual number of offsprings each individual will have. In the experiments presented here, we used an exponential ranking selection (Blickle and Thiele, 1996). The individuals are sorted by decreasing metabolic error so that the worst individual has rank $r = 1$ and the best $r = N$. The probability of reproduction of an individual is then given by $\frac{s-1}{s^N-1}s^{N-r}$, with $s = 0,998$ being the intensity of selection in all the experiments presented here.

Genetic operators

During their replication, genomes can undergo seven different kinds of modifications, three of which are local mutations (single nucleotide substitutions and insertions or deletions of 1 to 6 bp) and the four others, chromosomal rearrangements (duplications, deletions, translocations and inversions). The breakpoints for these rearrangements are randomly chosen on the chromosome.

Mutations and rearrangements affect the genome but do not necessarily have a phenotypic effect. For instance, a mutation that takes place in an untranscribed region will be completely neutral unless it creates a new promoter, which is reasonably rare given the size of the consensus sequence.

The rates at which each type of genetic modification i occurs (μ_i) are parameters of the model. They are defined as the per-base, per-replication probability of each type of modification to take place. Although horizontal transfer is possible in Aevol, we disabled it entirely in these experiments to avoid the assembly of operons due to the selfish operon effect.

Aevol is hence a digital genetics model in which the structure of the genome is free to evolve. It integrates major genetic features and mechanisms, introducing a transcription-translation level between the genetic and the phenotypic levels and allowing both local mutations and large chromosomal rearrangements. These particularities make Aevol a model that is particularly suited for the study of genome organization.

Results

The typical use of digital genetics models is very close to experimental evolution procedures (Elena and Lenski, 2003): Populations of organisms are initialized and left to evolve in controlled conditions. By observing the products and the dynamics of the evolutionary process in different conditions and by comparing them, we can unravel the direct or indirect pressures that constrain the structure of the organisms.

We let 147 populations of 1,000 individuals evolve during 20,000 generations in near identical conditions where the only changing parameters were the mutation rate and the

rearrangement rate (one common rate μ_m for the three different types of local mutations and one, μ_r, for the four types of rearrangements) for which values ranged from 1.10^{-6} to 1.10^{-4} per base-pair (7 rates tested). Each combination of mutation and rearrangement rates was tested with 3 independent seeds.

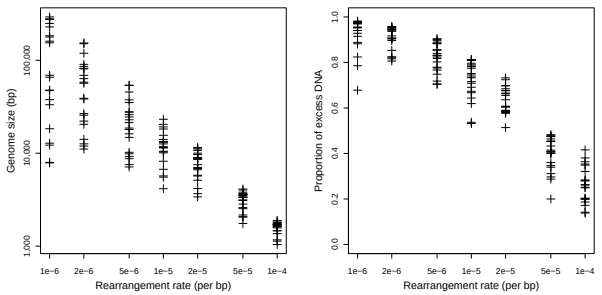
These populations evolved in identical environments composed of a single Gaussian curve placed on the right hand side of the metabolic axis (figure 3). The central zone of the axis was neutralized, meaning that the organisms receive no penalty for evolving proteins in that zone (even though they are of no use). This will enable us to test whether non-essential genes can be packed together with other genes in an operon structure.

This experiment was designed as a null-experiment for the selfish operon theory: The populations evolved in a strictly clonal framework where no horizontal transfer was allowed. According to the selfish operon theory, operons should not be observed in such conditions. Operons that would arise nevertheless could be explained by either the co-regulation or the mutational burden hypotheses. The variations of mutation and rearrangement rates will enable us to test the mutational burden hypothesis, and the co-regulation theory can be tested by analysing the functional relatedness of genes organized in operons.

Evolution of the structure of the genome

During the evolutionary process, the organisms progressively acquire new genes and modify them in such a way that the whole gene repertoire fulfils the task the organisms are selected for. All the simulations proceed qualitatively in a similar way, evolving quickly in the first stage of evolution (rapid gene acquisition mostly by duplication-divergence) then slowing down the process of gene acquisition while optimizing the sequence of existing genes and promoters. However, looking at the evolution of the size of the genome and the number of genes, we can see a clear trend for individuals evolving under lower rearrangement rates to have larger genomes containing both more genes and a greater proportion of non-coding sequences (figure 4). The rate of rearrangements is the major factor explaining the variability of genome compactness, the rate of small mutations has a much lower effect. Interestingly, the genome size stabilizes even though there is no direct cost for neither the replication of the genome nor for its expression.

As we have already shown, these effects are the consequence of the long-term selection of a specific level of mutational robustness (Knibbe et al., 2007). Indeed, we have estimated the fidelity of the replication for each of the 147 final best individuals, by a mutagenesis-like experiment: We let each of them reproduce 10,000 times and counted the number of offspring that had retained the ancestral fitness, in order to estimate the fraction of neutral offspring, F_ν. Figure 5 shows that in all cases, the genome had evolved in such a way that F_ν was greater than $1/2.31$. Thus, on the 2.31 offsprings expected for the best individual during the runs (given the selection intensity), at least 1 of them would retain the ancestral fitness, while the other ones would explore other phenotypes. This reflects the indirect selection of an appropriate trade-off between exploitation and exploration: under a high mutation rate per base-pair, the only way to reach a good trade-off is to keep the genome small. This phenomenon, known as an "error threshold" effect (Eigen, 1971), sets an upper bound to the total coding length, but also, here, on the non-coding length. Indeed, when rearrangements are taken into account, non-coding sequences are actually mutagenic for the genes they surround, because they provide breakpoints for large duplications or deletions (Knibbe et al., 2007).



(a) Genome size (b) Proportion of excess DNA

(c) Number of genes (d) Metabolic error

Figure 4: Genome size, proportion of excess DNA, number of genes and metabolic error for the best individual of each simulation after 20,000 generations. The fittest individuals are those with the lowest metabolic errors. Excess DNA includes here the intergenic DNA (between two coding RNAs) and the untranslated regions of the RNAs.

Evolution of the structure of transcripts

Looking more specifically at transcription-related features, our attention was drawn by the clear trend for higher rearrangement rates to favour long RNAs (figure 6(a)). The dynamics that leads to this lengthening of transcripts is very interesting: Indeed, as figure 7 shows, only the terminators seem to be gotten rid of during the whole evolutionary time,

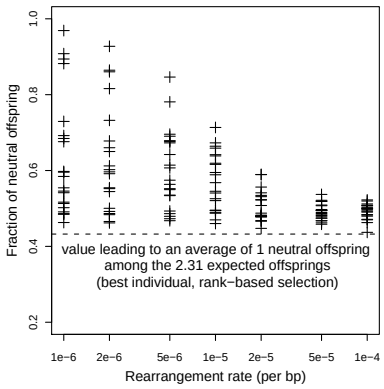
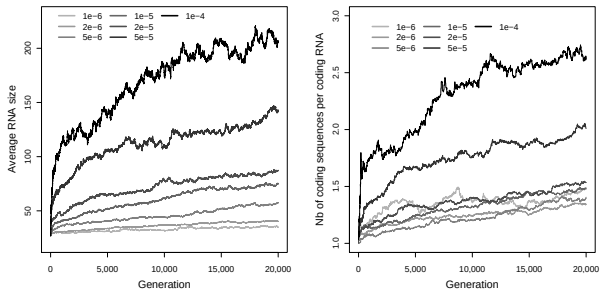


Figure 5: Fraction of neutral offspring estimated for the final best individual, after 20,000 generations of evolution.

the promoter density remaining stable after the first stage of evolution.

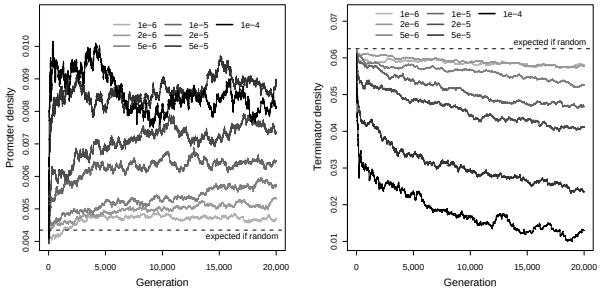


(a) RNA size (b) Number of genes per RNA.

Figure 6: Evolution of the average size of RNAs (regardless of whether they are coding or non-coding) and the average number of genes per coding RNA (RNAs containing at least one CDS). For clarity purpose, the data displayed here has been averaged over the different small mutation rates and seeds. Each line is hence the average value of the 21 simulations that were run under the same rearrangement rate.

Selection against terminators under high rearrangement rates leads to a lengthening of RNAs. But why are long RNAs selected for? What are the benefits of postponing transcription termination? The answer apparently resides in the packing of coding sequences: On average, RNAs belonging to organisms that evolved under high rearrangement rates own way more genes than those under low rates (figure 6(b)).

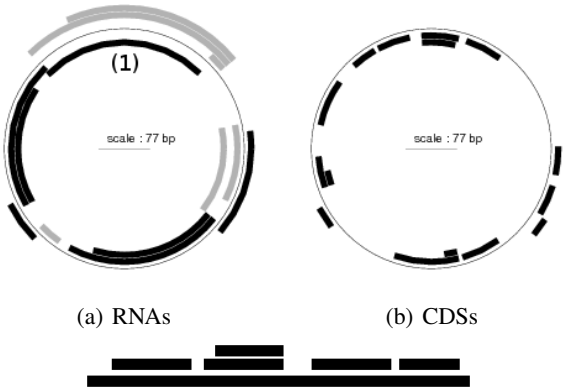
Figures 8 and 9 show the translation and transcription organization of the best individuals (after 20,000 generations) of 2 typical simulations with respectively high and low mutation and rearrangement rates. Under low rearrangement rates, almost every single CDS is transcribed by a different RNA. On the contrary, the individual that evolved under high



(a) Density of promoters (b) Density of terminators

Figure 7: Evolution of the average density of promoters (a) and terminators (b) for the different rearrangement rates. See figure 6 for details about data aggregated.

rearrangement rates has but one RNA containing only one gene, all the other transcripts carrying at least two. These figures also show a great difference regarding non-coding RNAs. At high mutation rates, a huge proportion of RNAs are ncRNAs whereas they become rare at high rearrangement rates, this reproduces what is observed in real organisms, eukaryotes having way more ncRNAs than prokaryotes have. Putting the focus on this aspect of our data, we found a clear scaling law between the rearrangement rate and the proportion of ncRNAs (data not shown). This scaling is a direct consequence of the proportion of non-coding sequences on the genome.



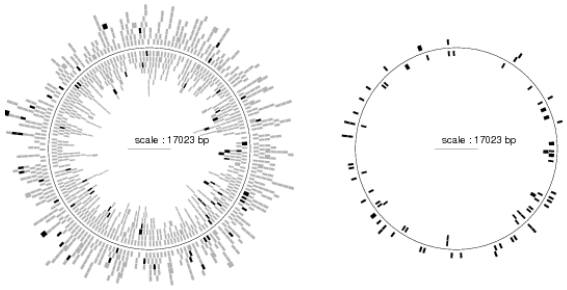
(a) RNAs (b) CDSs

(c) Zoom on operon (1) with its 5 genes

Figure 8: Genome of the best individual of generation 20,000 of a typical simulation with mutation and rearrangement rates of 1.10^{-4} per base-pair. In subfigure (a), coding RNAs are represented in black and ncRNAs in grey.

Discussion

In the experiments presented here, the organization of the genomes after 20,000 generations of evolution reproduces the whole range of genome organizations observed in real



(a) RNAs (b) CDSs

Figure 9: Genome of the best individual of generation 20,000 of a typical simulation with mutation and rearrangement rates of 1.10^{-6} per base-pair. In subfigure (a), coding RNAs are represented in black and ncRNAs in grey.

organisms. In our simulations, we observed a clear tendency for organisms having evolved under low rearrangement rates to have a eukaryote-like genome and for those under high mutation rates to resemble prokaryotic genomes.

Although a very small proportion of eukaryotic genomes is translated into proteins, a substantial fraction of these genomes is transcribed into non-coding RNAs. Not all of these ncRNAs have a known function and a great deal of effort is put into identifying these putative functions. In our model, ncRNAs have absolutely no function, yet they are very common when rearrangement rates are low. Interestingly, they are found at a proportion close to that which would be expected in a random sequence. Hence, it seems that ncRNAs are naturally present in intergenic regions making them available for acquiring new functions. It is tempting to suggest that these RNAs constitute a good substrate for the appearance of novel genes but this question will require a precise analysis of the dynamics of gene acquisition.

Another interesting feature we have observed is the emergence, under specific conditions (*i.e.* under high rearrangement rates), of operon structures.

Since operons appeared in a total absence of horizontal transfer, the selfish operon theory can easily be discarded as an explanation of the emergence of these operons. Indeed, horizontal transfer is a central and necessary feature of the selfish operon theory.

One of the remaining candidates to account for the emergence of the observed operons is the co-regulation model, under which hypothesis genomes should be more modular than expected at random. To compute the functional modularity of a genome, we conducted a pairwise comparison of the proportion of functionally related genes within operons and on the whole genome. Two genes were considered functionally related when they shared a subset of metabolic functions, *i.e.* when their corresponding phenotypic triangles overlapped. Given that the individuals evolved in a stable environment, no regulation is needed whatsoever. Modularity was shown to promote evolvability in the presence of inter-individual recombination (Pepper, 2000). However, here, reproduction was strictly clonal, which makes it difficult to imagine how the modularity of a genome could improve a lineage's evolutionary fate.

Yet, the results show a moderate tendency to pack functionally related genes together on the same operon: The proportion of pairs of functionally-related genes within operons was 1.26-fold higher (median value) than the same proportion on the whole genome. Although the effect is small, the ratio is significantly different from 1 (non parametric sign test, p-value = 7.10^{-4}).

These results do not allow us to conclude either in favor of or against the co-regulation theory and further experiments and analyses will be necessary to tackle this question.

According to the results presented in figure 6(b), there seems to be a threshold in the rearrangement rate above which operons become the rule rather than the exception. This is relevant when considered in the light of the mutational burden theory: As we have previously stated, the selection for a correct level of mutational robustness that was unravelled by Knibbe et al. (2007) leads to a strong pressure on the genome size. The higher the rearrangement rate, the smaller the genome must be to be transmitted faithfully to the offspring. Besides, the selection of the individuals that best fulfil the metabolic task (*i.e.* approximate the target) gives rise to a pressure for having many genes. Taken together, these two pressures result in the emergence of a composed pressure on the density of genes.

At medium rearrangement rates, the optimal gene density can be achieved by simply reducing the proportion of non-coding sequences, the coding sequences themselves remaining mostly unaffected. However, when the rates are really high, the amount of excess DNA (inter-RNA sequences, 3' and 5' UTRs, ncRNAs) shrinks to nearly nothing. At high rates, a further compaction can be done by several means such as making genes overlap (either on the same strand or on both strands) or getting rid of some of the transcription signals (promoters and terminators), hence merging consecutive RNAs into one single RNA (thus creating an operon).

We therefore expected to observe both overlapping genes and a lengthening of transcript length under high rearrangement rates. We indeed observed both of these phenomena (figures 8 and 6(a)) but were surprised by the dynamics leading to RNA lengthening: When the density of promoters appears to be stable over time, suggesting that they are not selected against, the density of terminators is constantly decreasing. Terminators fragment the genome, forbidding the sequences directly downstream from them (on both strands) to be translated, until a promoter is found. There is hence unmistakably a loss of gene density for each terminator on the genome. The solution that evolution found to efficiently pack genes together is then to limit this loss by decreasing the number of terminators on the genome, leading to a

lengthening of the average size of RNAs which in turn facilitates the emergence of operons.

Conclusion

In this paper, we have presented results that clearly reproduce features of genome organization that are observed in real organisms, in particular the structuration of genes in operons. The emergence of these operons specifically under high rearrangement rates points us to the mutational burden hypothesis, where a second-order pressure for a specific level of mutational robustness leads to genome streamlining. We now plan to conduct further experiments to investigate the role of horizontal transfer and how it interacts with this second-order pressure. We also plan to determine to what extent the co-regulation model can participate in the creation and maintenance of operon structures. Finally, we would like to analyse the role of non-coding RNAs in gene acquisition and to test whether they are innovation hot spots.

Acknowledgements

The authors would like to thank the BSMC group, who provides us with the computing resources and Fabien Chaudier for his invaluable help.

References

Adami, C. (2006). Digital genetics: unravelling the genetic basis of evolution. *Nat. Rev. Genet.*, 7(2):109–118.

Beslon, G., Sanchez-Dehesa, Y., Parsons, D., Peña, J. M., and Knibbe, C. (2009). Scaling laws in digital organisms. In *Proc. Information Processing in Cells and Tissues IPCAT'09*, pages 111–114.

Blickle, T. and Thiele, L. (1996). A comparison of selection schemes used in evolutionary algorithms. *Evol. Comput.*, 4(4):361–394.

Dittrich, P., Ziegler, J., and Banzhaf, W. (2001). Artificial chemistries-a review. *Artif Life*, 7(3):225–275.

Eigen, M. (1971). Self-organization of matter and the evolution of biological macromolecules. *Naturwissenschaften*, 58:456–523.

Elena, S. F. and Lenski, R. E. (2003). Evolution experiments with microorganisms: the dynamics and genetic bases of adaptation. *Nat. Rev. Genet.*, 4(6):457–469.

Gago, S., Elena, S. F., Flores, R., and Sanjuan, R. (2009). Extremely high mutation rate of a hammerhead viroid. *Science*, 323(5919):1308.

Hershberg, R., Yegerlotem, E., and Margalit, H. (2005). Chromosomal organization is shaped by the transcription regulatory network. *Trends Genet.*, 21(3):138–142.

Jacob, F., Perrin, D., Sánchez, C., and Monod, J. (1960). L'opéron : groupe de gènes à expression coordonnée par un opérateur. *C. R. Acad. Sci. Paris 250*, pages 1727 – 1729.

Johnson, Z. I. and Chisholm, S. W. (2004). Properties of overlapping genes are conserved across microbial genomes. *Genome Res.*, 14(11):2268–2272.

Kapranov, P., Willingham, A. T., and Gingeras, T. R. (2007). Genome-wide transcription and the implications for genomic organization. *Nat. Rev. Genet.*, 8(6):413–423.

Knibbe, C., Coulon, A., Mazet, O., Fayard, J.-M., and Beslon, G. (2007). A long-term evolutionary pressure on the amount of noncoding DNA. *Mol. Biol. Evol.*, 24(10):2344–2353.

Knibbe, C., Fayard, J.-M., and Beslon, G. (2008). The topology of the protein network influences the dynamics of gene order: from systems biology to a systemic understanding of evolution. *Artif. Life*, 14(1):149–156.

Lawrence, J. (1999). Selfish operons: the evolutionary impact of gene clustering in prokaryotes and eukaryotes. *Curr. Opin. Genet. Dev.*, 9(6):642–648.

Lynch, M. (2006). Streamlining and simplification of microbial genome architecture. *Annu. Rev. Microbiol.*, 60(1):327–349.

Mattick, J. S. and Makunin, I. V. (2006). Non-coding RNA. *Hum. Mol. Genet.*, 15 Spec No 1(suppl_1):R17–29.

Misevic, D., Ofria, C., and Lenski, R. E. (2006). Sexual reproduction reshapes the genetic architecture of digital organisms. *Proc. R. Soc. B.*, 273(1585):457–464.

Pál, C. and Hurst, L. D. (2004). Evidence against the selfish operon theory. *Trends Genet.*, 20(6):232–234.

Palleja, A., Harrington, E., and Bork, P. (2008). Large gene overlaps in prokaryotic genomes: result of functional constraints or mispredictions? *BMC Genomics*, 9(1):335+.

Pepper, J. W. (2000). The evolution of modularity in genome architecture. In Maley, C. C. and Boudreau, E., editors, *Proceedings of the Evolvability Workshop at Alife VII*, Portland, USA.

Ponjavic, J., Ponting, C. P., and Lunter, G. (2007). Functionality or transcriptional noise? Evidence for selection within long noncoding RNAs. *Genome Res.*, 17(5):556–565.

Raoult, D., Audic, S., Robert, C., Abergel, C., Renesto, P., Ogata, H., La Scola, B., Suzan, M., and Claverie, J.-M. (2004). The 1.2-megabase genome sequence of mimivirus. *Science*, 306(5700):1344–1350.

Rensing, C. (2002). The role of selective pressure and selfish dna in horizontal gene transfer and soil microbial community adaptation. *Soil Biol. Biochem.*, 34(3):285–296.

Wilke, C. O., Wang, J. L., Ofria, C., Lenski, R. E., and Adami, C. (2001). Evolution of digital organisms at high mutation rates leads to survival of the flattest. *Nature*, 412(6844):331–333.

Will, S., Reiche, K., Hofacker, I. L., Stadler, P. F., and Backofen, R. (2007). Inferring noncoding RNA families and classes by means of genome-scale structure-based clustering. *PLoS Comput. Biol.*, 3(4):e65+.

Zheng, Z.-M. M. and Baker, C. C. (2006). Papillomavirus genome structure, expression, and post-transcriptional regulation. *Frontiers in bioscience*, 11:2286–2302.

Evolution of Recombination on an HIV-1 Derived Fitness Landscape

João Martins[1], Roger Kouyos[1], Trevor Hinkley[1], Colombe Chappey[2],
Mojgan Haddad[3], Neil Parkin[3], Jeanette Whitcomb[3],
Christos Petropoulos[3], and Sebastian Bonhoeffer[1]

[1]ETHZ, Institute of Integrative Biology, CHN H 72, Universitätstr. 16, 8092 Zürich, Switzerland
[2]Genentech, 1 DNA Way, South San Francisco, CA 94080-4990, USA
[3]Monogram Biosciences, Inc., 345 Oyster Point Blvd., South San Francisco, CA 94080-1913, USA
joao.martins@env.ethz.ch, sebastian.bonhoeffer@env.ethz.ch

Abstract

The maintenance of recombination is among the most important unsolved problems of evolutionary biology. The Hill-Robertson effect, which states that the interaction between genetic drift and selection generates unfavorable linkage disequilibria (hence favoring recombination), offers one of the most promising hypotheses to solve this problem. In particular, it has been argued that this hypothesis works independently of epistatic interactions. However, this result has been derived on the basis of smooth fitness landscapes, which may be unrealistic (Otto and Feldman (1997)). We estimated the fitness effects of 1'857 single mutations and of 257'536 pairs of mutations found in a 60'000 HIV-1 B pol-genotypes assayed for in vitro replication capacity (Hinkley et al. (2010)) to develop a reasonably realistic model of a fitness landscape on which we run a genetic algorithm to mimic the evolution of HIV populations. By adding a recombination rate modifier to the genome, we address the question of whether genetic drift outweighs epistasis as a factor for the evolutionary maintenance of recombination in the case of the fitness landscape of our model. Despite the fairly rugged nature of the fitness landscape, which could be characterized by the presence of a large number of local optima, we find that recombination is robustly favored in finite populations. This result suggests that the Hill-Robertson effect provides a powerful explanation for the evolutionary maintenance of recombination even if fitness landscapes are rugged.

References

Otto, S. and Feldman, M., (1997). Deleterious mutations, variable epistatic interactions, and the evolution of recombination. *Theor Popul Biol* 51: 134–147.

Hinkley, T., Petropoulos, C., and Bonhoeffer, S. (2010). The Evolutionary Systems Biology of HIV Drug Resistance. *In press*.

Modelling the Role of Aneuploidy in Tumour Evolution

Arturo Araujo [1], Peter Bentley [2] and Buzz Baum [3]

[1]UCL CoMPLEX
[2]UCL Computer Science
[3]UCL MRC Laboratory for Molecular Cell Biology

a.araujo@cs.ucl.ac.uk

Abstract

The role of aneuploidy (the cellular state of having an abnormal number of chromosomes) in cancer is not well understood. A recent theory suggests that aneuploidy may be an initial step towards the generation of variation in cancer. This theory however is very difficult to test in biological experiments. To address this theory and explore the role that aneuploidy has on the development of cancer, a computational model of cancer evolution has been developed. Results show that, depending on the arrangement of tumour suppressors, proto-oncogenes and regulators of chromosome segregation in the genome, aneuploidy induces distinct pathways for the generation of novel genotypes leading to emergent cancer-like behaviour.

1 Introduction

Cancer is a disease through which a group of cells proliferate beyond the normal limits of division, destroying adjacent tissue and sometimes spreading to other locations in the body. Tumours evolve in the body behaving almost like infecting pathogens with the cells undergoing a sequence of genetic mutations until they are able to proliferate almost without limit. Cancer affects people of all ages and ethnicities, with risk increasing with age. Cancer is one of the leading causes of death worldwide, with cancer deaths projected to continue rising (Parkin et al. 2005). To tackle this disease, efforts are being made to generate knowledge about the causes of cancer and the management of the disease. Cancer research, a field ranging from molecular bioscience to clinical trials, seeks to increase our understanding of the fundamental principles of cancer. Through this kind of research, we have been able to identify many of the key factors that influence cancer and the development of treatments and prevention strategies. Because of the complexity of cancer development, which involves the evolution of somatic clones with increasingly aggressive behaviour that eventually undergo metastasis, computational modelling has become a very valuable tool for refuting or supporting theories that explain the underlying individual cell behaviour in tumours (Nagl et al. 2007).

In the field of Artificial Life, efforts are being made to simulate and understand the properties of cancer systems. These contributions are an important part in the development of a more general theory of cancer (Abbott et al. 2006). They have inspired new ways of thinking and revolutionized the way we explore, describe and explain complex biological phenomena. One such phenomenon, aneuploidy, has recently gained much interest in the cancer community.

In the absence of sexual recombination, the path to cellular evolution is through mutation, the generation of chromosome aberrations and aneuploidy– the cellular state of having an abnormal number of chromosomes. Evolutionary pressure selects for genetic changes that enable cells to avoid death and over proliferate. This can be achieved by the overexpression of growth signals, adaptation to hypoxia and evasion of reproductive limits amongst others (Gibbs 2003). Unfortunately it is extremely difficult to devise biological experiments to isolate the effects of aneuploidy in cancer (Weaver and Cleveland 2007). Because of the extreme difficulties encountered when trying to devise this kind of biological experiment, in this work we propose a computational model to address some of the fundamental questions of tumour formation and help further guide experiment and theory.

The aim of this work is investigate the role of aneuploidy and its effect on the dynamics in cancer. By making abstractions of current biological knowledge, data and theories that describe the behaviour of cancer, a computational model that addresses this theory is presented. The model explores the role

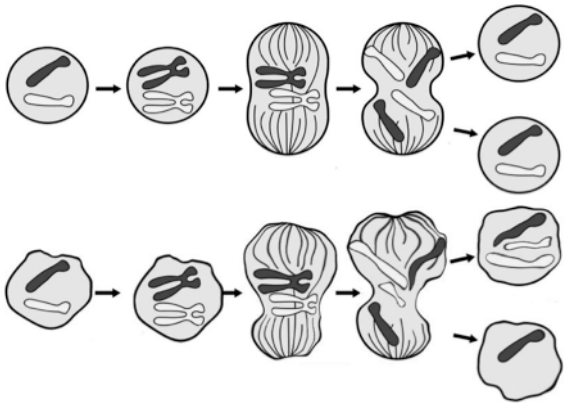


Figure 1- Schematic of normal cell division (top) and the missegregation of chromosomes during mitosis (bottom).

that aneuploidy plays as a main driver for the origin and the subsequent stages of cancer. It is an individual-based evolutionary model, similar to models used in ALife work for other population-based simulation studies (Gras et al. 2009).

In the next section, the essential theories of the origins of cancer are summarised. Section 3 presents the details of the computational model. Section 4 continues with a discussion of the different simulations carried out. Conclusions and future work are provided in the final section.

2 Background

What we currently consider to be cancer includes, in reality, a very broad spectrum of diseases known as malignant neoplasms. Biological systems are complex, and cancer in particular may be best described as an emergent behaviour of a complex system (Nagl et al. 2007). Because it is very difficult to understand a complex system by examining only its components, the exact mechanisms by which cancer can arise are a matter of heated debate (Basanta and Deutsch 2008).

There are two predominant theories regarding the origins of cancer. The first theory suggests that DNA damage over decades leads to many thousands of random mutations in the cell's genome that confers on the cell new proliferative capabilities (Chin et al. 2006). Chemical carcinogens such as ionizing radiation (x-rays, etc) may cause chromosomal breaks and translocations that contribute to cancer development. This kind of damage is largely stochastic and raises the question of how can such a comprehensive genome reprogramming be carried out so consistently for the development of a cancer genotype by means of random mutations.

The second theory suggests that damage to a few "cancer genes", such as those depicted in Table 1, would activate pathways that would lead to tumourigenesis by means of accumulative changes (Hanahan and Weinberg 2000). This theory suggests that the accumulation of very particular alterations (also known as "gate-keeper" mutations) in proto-oncogenes (genes that contribute to cancer because of their increased expression) and tumour suppressor genes (genes that contribute to cancer when its function is reduced) could be a main driver for many cancers (Gatenby et al. 2007). This theory does not directly address the underlying evolutionary and selective forces that play an important role in cancer development, nor the interaction with a particular microenvironment in which phenotype selection takes place.

A third theory, proposes that an abnormal number of chromosomes, or aneuploidy (described in Figure 1), in a cell may be a first step towards generating malignant genotypes (Gibbs 2003). This theory (as first proposed by Boveri in 1914) has recently gained support due to many recent articles that describe the presence of aneuploidy and chromosomal instability in many types of cancers (Rajagopalan and Lengauer 2004). More significantly, mutations leading to chromosome instability lead to a genetic predisposition to cancer (Hanks et al. 2004). The high number of different cellular states that are considered as aneuploid and the different behaviours and interactions that these cells may exhibit make it difficult to trace an evolutionary pathway through this complex system. Because of a lack of a clear pathway, the contribution of aneuploidy as a cause or a consequence of malignant transformation, remains unknown (Holland and Cleveland 2009).

3 The Model

In order to investigate the theory of aneuploidy as a driver for the development of malignant cancer, a model was created. The computational model consists of individual agents that are abstractions of individual cells, incorporating a set of biologically-inspired features dealing with cell division and more specifically chromosome segregation.

The model abstracts biological behaviour at the genetic level, and studies the behaviour at a tissue level that emerges through the interaction of the individual cells under diverse conditions. In the model, abstractions of genes known to play a relevant role in tissue homeostasis are considered. This kind of model could not only provide us with an insight as to the origins and the evolution of cancer, but also with a new tool for developing new cancer therapies.

Gene	Role in Cancer	Biological Function
BUB1	**Aneuploidy**	Chromosome segregation
MYC	**Proto-oncogene**	Promotes growth
PTEN	**Tumour suppressor**	Inhibits growth
RAS	**Proto-oncogene**	Promotes growth, cell cycle progression
RB1	**Tumour suppressor**	Inhibits cell cycle progression
P53	**Tumour suppressor**	Promotes cell death
NF2	**Tumour suppressor**	Regulates contact inhibition

Table 1- Known human cancer genes considered. The function of the genes as given is a broad summary and approximation of their true behaviour, which is still the subject of research.

3.1 Biological Abstractions

In order to develop a computational model to study the biological phenomenon of aneuploidy, it was decided to investigate the behaviour of a few known cancer genes (Futreal et al. 2004), as seen in Table 1. Although alterations in these cancer genes may account for specific cellular misbehaviours, the genetic evolutionary pathway that cells follow when they become cancerous remains unknown. To address this question, behaviour was abstracted from genes that regulate cell death, proliferation, and fidelity during chromosome segregation.

The core of the model is an abstraction of individual cells and their genomes. Each simulated genome is composed of 3 types of genes in diploid chromosomes (pairs of chromosomes, the chromosomes of each pair having identical genes) as the normal state within cells, as seen in Figure 2. The collection of individual cells comprises a simulated tissue, whose population size is determined for each

experiment through an *allocated space* parameter, whose dynamics are determined by the gene expression of the individual cells across time. Although the effects of differences in chromosome number on gene expression patterns in biological systems are only beginning to be assessed (Huettel et al. 2008), the model assumes the up and down regulation of behaviour to be proportional to the number of copies of genes available. Each of the three genes code for corresponding actions at a cellular level, inspired by biological systems. The genes present and their functions, described below, are:

- *Tumour Suppressors- Apoptosis Regulatory Genes (A)*
- *Proto-oncogenes- Cell Division Regulatory Genes (D)*
- *Aneuploidy- Chromosome Segregation Regulatory Genes (S)*

Apoptosis regulatory genes are an abstraction of tumour suppressor genes that regulate cell death by mechanisms such as contact inhibition. Contact inhibition is the natural process by which, when two or more cells come into contact with each other, there is an arrest of the cell growth and division, which is used by the system to maintain homeostasis. (Zeng and Hong 2008). The abstracted genes are used to compare a measurement of the overall number of cells and, if this number exceeds the *carrying capacity* of the tissue (predefined by the initial conditions of the simulation), it stops proliferation and raises the probability of cell death. Malignant cells usually have lost this important homeostatic property (Carmona-Fontaine et al. 2008). Although based on global cell counts, this model is not spatially explicit, but rather of the "well-stirred" kind, akin to the more abstract theoretical models used to describe artificial chemistries (Dittrich et al. 2001).

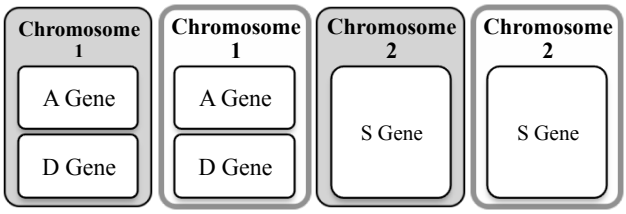


Figure 2- Abstracted Genes in Diploid Chromosomes for Gene Configuration A.

To balance cellular death and maintain homeostasis, *cell division regulatory genes* provide an abstraction of proto-oncogenes that promote growth and progression through the cell cycle. *Apoptosis regulatory genes* and *cell division regulatory genes* together maintain a constant population of cells close to the carrying capacity of the simulated tissue (homeostasis).

The inclusion of the concept of aneuploidy generates variation amongst the cell population (no other form of mutation is modelled in the system). Inspired by genes that limit chromosome missegregation events, *chromosome segregation regulatory genes,* when up regulated, help maintain homeostatic conditions for a prolonged period of time. The role that the up or down regulation of these kinds of genes has in cancer progression is currently unknown (Rajagopalan and Lengauer 2004).

The model contains a population of individual cells, where each cell is initialized with 2 copies of each gene, within diploid chromosomes, as shown in Figure 2. When dividing, the genome of each cell is duplicated and one set of genes then segregated into a daughter cell. It is during this stage that chromosome missegregation events can occur. The behaviour generated by the gene expression is dependent on the number of copies of a given gene within the genome of each individual cell. The algorithm is described in the following section.

3.2 The Algorithm

Inspired by the processes in biological cellular behaviour through which homeostasis is maintained in organisms, the algorithm is as follows:

1. An initial population of 100 cells is created, each with diploid chromosomes, each chromosome with 1 copy of each type of gene (Figure 2). The normal carrying capacity of the tissue is fixed at 200 cells.
2. For each time step, the total number of cells is measured and is not updated until the next time step.
3. For each cell during each time step, if the cell has less than 2 chromosomes in its entire genome, the cell dies.
4. If the cell has not died and if the measurement of the number of cells is greater than the predefined tissue's *carrying capacity*, then the probability of cell death is calculated. The probability of death is dependent on the number of available copies of the *apoptosis regulatory genes*, N_A, within each cell's genome. The probability of apoptosis, P_A, is determined by:
$$P_A = N_A / r_A$$
Where r_A is a parameter for the rate of apoptosis. The cell is then killed with a probability of P_A.
5. If the cell has not died, it has a chance to divide. The probability of division depends on the number of available copies of the *division regulatory genes*, N_D, and a parameter that determines the rate of division, r_D. The probability that a cell divides, P_D, is:
$$P_D = N_D / r_D$$
6. If dividing, the probability of chromosome missegregation is calculated. The probability of chromosome missegregation, P_S, in the model is:
$$P_S = r_S / (N_S + 1)$$
Where N_S is the number copies of the *chromosome segregation regulatory genes* within the cell's genome, and r_S is a parameter for the rate of chromosome missegregation.

If there is no chromosome missegregation, the genome is duplicated and copied with fidelity, thus generating two identical daughter cells. Otherwise, one chromosome chosen at random is misseggregated during cell division. As the mother cell divides into two daughter cells, this results in two daughter cells with a different number of chromosomes, as seen in Figure 1.

4 Experiments

To investigate the properties and the dynamics of the system, and specifically the role that chromosome segregation regulatory genes have, three genome configurations were considered. The parameter settings were determined through a series of preliminary experiments, in order to ensure that the behaviour of the system was both biologically plausible and computationally feasible. Simulations were carried out with the following initial parameters:

- Initial population: 100 cells
- Carrying capacity of the tissue: 200 cells
- Number of time steps: 100
- $r_A = 10$, $r_D = 10$, $r_S = 0.03$

For the analysis of the simulations, the emergent genotypes were assessed. By quantifying the number of chromosomes that a cell has at a given time, a genotype state G_T is defined as:

$$G_T=(N_A, N_D, N_S)$$

Where N_A, N_D *and* N_S are the number of copies of *Apoptosis Regulatory Genes, Cell Division Regulatory Genes* and *Chromosome Segregation Regulatory Genes* respectively. The initial genotype consists of two functional copies of each chromosome: genotype state (2, 2, 2).

Three different gene configurations (Figure 2, 5 and 7), 20 simulations were investigated for each experiment. As will be shown, although the systems tended to converge on similar results, the evolutionary trajectories were usually different. For this reason a representative simulation for each configuration is given in the results sections rather than an average. Future work will investigate an appropriate statistical analysis of the distribution of evolutionary pathways across simulations.

4.1 Gene Configuration A

4.1.1.Objective and Setup

To investigate the role of the *chromosome segregation regulatory genes*, the following configuration was used:

- Chromosome 1: *apoptosis regulatory genes (A)* and *cell-division regulatory genes* (D)
- Chromosome 2: *chromosome segregation regulatory genes (S)*

This gene configuration, as seen in Figure 2, isolates the effects of the loss or gain of Chromosome 2 to those caused by the loss or gain of the *chromosome segregation regulatory genes*.

4.1.2.Results

Homeostatic behaviour can be observed in Figure 3. In normal conditions this kind of homeostatic behaviour provides the tissue with robustness if there were a sudden loss of cells (wound-healing capabilities), maintaining the total number of cells close to that of the carrying capacity of the tissue (200 cells). For 20 simulations of Configuration A, the average

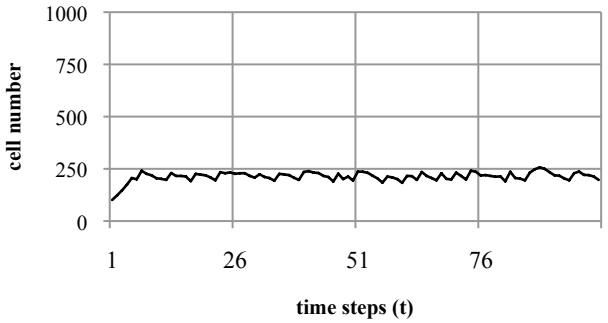


Figure 3- Total number of cells in a 100-time step simulation with Gene Configuration A.

total number of cells at the last time step (t=100) was 210 cells, with a standard deviation of 17.

4.1.3 Analysis

As expected, a comparison of the plot of the total number of cells across the simulations of Configuration A reveals the high variability of the simulation outcomes, as seen in Figure 4. Thus, it is difficult to distil meaningful information with traditional statistical methods. Despite the stochastic nature of the final cell number across experiments, an invariant qualitative behaviour can be observed for each configuration. Although the actual evolutionary pathway exhibits a high degree of variation, a representative simulation captures qualitatively the kind of evolutionary pathway that most of the simulations follow.

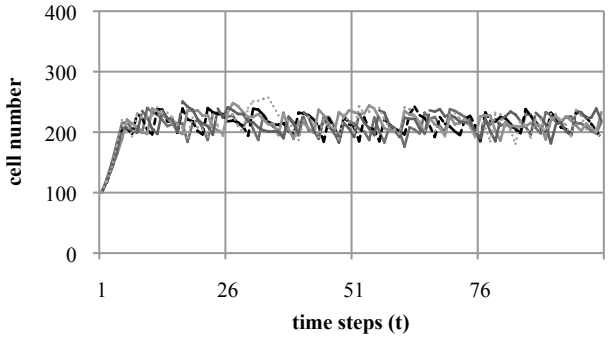


Figure 4- Distribution of the total amount of cells of 5 100-time step simulations with Gene Configuration A. Variability across experiments can be observed.

The initial genotype, genotype state (2, 2, 2), contains 2 functional copies of each gene. For there to be cancer-like behaviour, oncogenes need to have their function reduced and tumour suppressor genes in turn must have an increase in their expression. Because the abstracted genes that model the role of oncogenes and tumour suppressor genes are found in the same chromosome, they become self-regulated. As the system evolves however, novel genotypes emerge but, because of the self-regulation of the cancer genes, the overall behaviour generated by the new genotypes is not dissimilar to that of the

original cell population, as depicted in Figure 9a. This leads to a micro diversity of homeostatic genotypes. However, it is of interest that the more successful genotypes naturally acquire more resistance against chromosome missegregation. In this representative simulation, genotype state (2, 2, 3) accounts for more than 30% of the population at the last time step (t=100), as seen in a quantification of the distribution of genotypes (Table 2).

Genotype	t=0 (%)	t=25 (%)	t=50 (%)	t=75 (%)	t=100 (%)
(2, 2, 2)	**100**	**93.56**	**79.90**	**70.76**	**58.88**
(2, 2, 3)	0	1.72	**8.76**	**20.34**	**31.47**
(3, 3, 2)	0	0	4.12	4.66	0.51
(1, 1, 2)	0	0.43	3.09	2.97	5.58
(2, 2, 1)	0	3.00	2.06	0	0.51
(1, 1, 1)	0	0.43	1.55	0.42	1.02
(2, 2, >3)	0	0	0.52	0.85	1.02
(1, 1, 3)	0	0.86	0	0	0
(>3, >3, 2)	0	0	0	0	1.02

Table 2- Distribution of genotypes at 4 time intervals (0, 25, 50, 75 and 100) for a representative simulation of Gene Configuration A.

4.2 Gene Configuration B

4.2.1.Objective and Setup
To better understand the role of the distribution of the genes in the chromosomes, the initial configuration was modified to:

- Chromosome 1: *apoptosis regulatory genes* (A)
- Chromosome 2: *cell-division regulatory genes (D) and chromosome segregation regulatory genes* (S)

This gene distribution is depicted in Figure 5.

4.2.2.Results
During the 100-time step experiment, a stable homeostatic behaviour can be observed for a period of time. After that homeostatic period however, an uncontrolled proliferative behaviour follows. The total number of cells increases exponentially, reaching the values of the order of thousands in a very short period of time, as shown in Figure 6. This kind of behaviour is obtained across simulations. For 20 simulations of Configuration A, the average total number of cells across simulations at the last time step (t=100) was 59,388 cells, with an expected high standard deviation of 87,215. The

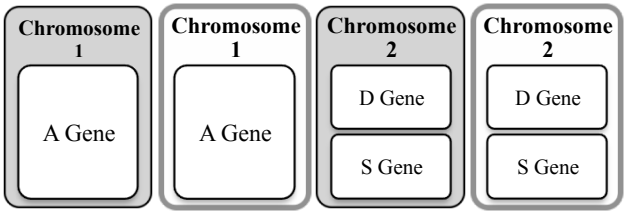


Figure 5- Distribution of Genes in Gene Configuration B

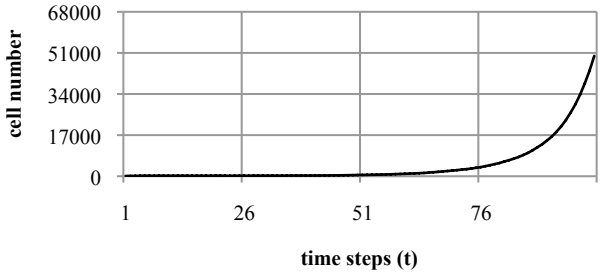


Figure 6- Total number of cells in a 100-time step simulation with Gene Configuration B.

representative simulation shown, ignoring the limits set by carrying capacity of the tissue, had a final number of 49,765 cells.

4.2.3 Analysis
An analysis of the emergent genotypes reveals that a newly evolved genotype takes over the population: Genotype state (1, 2, 2). From this novel genotype, two different kinds of genotypes are further evolved: an apoptosis-resistant genotype (0, 2, 2) and an over-proliferative genotype (1, 3, 3), which can be appreciated on Figure 9b.

The loss of function of the tumour suppressor-inspired *Apoptosis regulatory genes* through chromosome missegregation leads to the generation of a niche of these mutants. However, because of the low levels of chromosome missegregation, this population remains relatively homeostatic until the emergence of two cancer-like genotypes, as described by Table 3.

Genotype	t=0 (%)	t=25 (%)	t=50 (%)	t=75 (%)	t=100 (%)
(2, 2, 2)	**100**	**75.85**	9.74	0.72	0.14
(1, 2, 2)	0	19.81	**88.24**	**88.50**	**44.42**
(0, 2, 2)	0	0	0.41	4.50	**24.14**
(1, 3, 3)	0	0	0	4.90	**21.23**
(2, 3, 3)	0	2.42	1.42	0.72	0.15
(0, 3, 3)	0	0	0	0.17	9.04
(3, 2, 2)	0	0.97	0	0	0
(1, 1, 1)	0	0	0.20	0.49	0.36
(2, 1, 1)	0	0.97	0	0	0
(1, >3, >3)	0	0	0	0	0.36
(0, >3, >3)	0	0	0	0	0.14
(0, 1, 1)	0	0	0	0	0.02

Table 3- Genotype distribution (percentage) for a representative simulation of Gene Configuration B.

The evolution of the system with low levels of aneuploidy resulted in the generation of few very successful mutants that quickly dominated the entire population as seen in Table 3, suggesting a counterintuitive pathway for cancer-like behaviour with low aneuploidy. This kind of mutations are seen in leukemias, lymphomas and some mesenchymal tumours, where there are simple, disease-specific abnormalities (Johansson et al. 1996).

4.3 Gene Configuration C

4.3.1.Objective and Setup

To further study the role of the distribution of the genes in the chromosomes in a third configuration (Figure 7):

- Chromosome 1: *cell-division regulatory genes (D)*
- Chromosome 2: and *apoptosis regulatory genes (A) and chromosome segregation regulatory genes (S)*

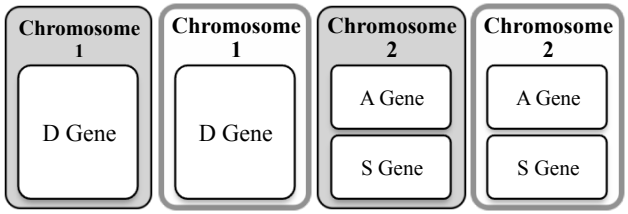


Figure 7- Distribution of Genes in Gene Configuration C

4.3.2.Results

Although this new genetic configuration yields a similar over-proliferative behaviour to that obtained through the simulations with Gene Configuration B, as seen in Figure 8, there are significant differences. The emergence of novel genotypes is less gradual, as can be appreciated in Figure 9c. In the representative simulation presented for this configuration, the total number of cells obtained at the last time step was 61,836 cells. The average final number of cells of the simulations carried out was 74,201, with a standard deviation of 114,736.

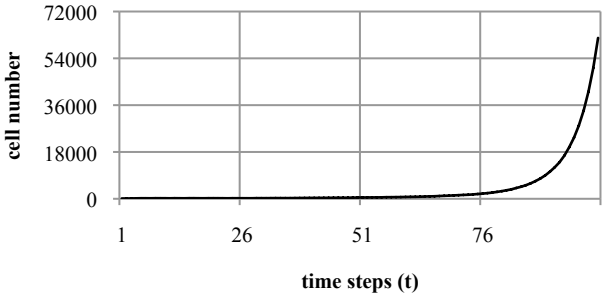


Figure 8- Total number of cells in a 100-time step Simulation with Gene Configuration C.

4.3.3 Analysis

An analysis of the genotype evolution sheds some light onto the emergence of the proliferative, cancer-like genotypes, as depicted in Figure 9c. Although the behaviour is similar to that of Gene configuration B, the evolution of a genotype that produces the cancer-like behaviour is significantly different. The analysis of the emergent genotypes reveals that the first mutation leads to an increase in the function of genes that model proto-oncogenes, increasing proliferation. However, contact inhibition induced cell death (the tumour suppressor genes) heavily restrict the mutant genotype from dominating the entire population. By acquiring mutations that reduce the contact inhibition forces, chromosomal instability is also induced. This instability leads to an explosion of genotypic diversity, as seen in Table 4, making it easier for cells to acquire mutations that lead to cancer-like behaviour.

This pathway may help shed some light on the reports of increasing levels of chromosome instability during premalignant neoplastic progression (Lai et al. 2007) and the development of tumours characterized by multiple and nonspecific aberrations, similar to most epithelial tumour types (Johansson et al. 1996)

Genotype	t=0 (%)	t=25 (%)	t=50 (%)	t=75 (%)	t=100 (%)
(2, 2, 2)	**100**	**92.38**	**40.88**	3.43	0.11
(2, 3, 2)	0	6.19	**40.25**	**31.17**	2.22
(1, 2, 1)	0	1.43	16.35	**31.93**	6.82
(2, >3, 2)	0	0	2.31	**17.71**	**24.70**
(1, 3, 1)	0	0	0.21	13.90	**21.71**
(1, >3, 1)	0	0	0	1.09	**24.80**
(0, 3, 0)	0	0	0	0.22	7.44
(0, >3, 0)	0	0	0	0	**11.22**
(0, 2, 0)	0	0	0	0.11	0.65
(1, 1, 1)	0	0	0	0.33	0.05
(3, >3, 3)	0	0	0	0	0.27
(3, 3, 3)	0	0	0	0.11	0.02
(0, 1, 0)	0	0	0	0	0.00

Table 4- Genotype distribution at different time intervals for Gene Configuration C.

5 Conclusions and Future Work

In this a work a computation model was created in order to investigate the role of chromosome missegregation in tumour evolution. By integrating the concept of chromosome missegregation in an otherwise homeostatic model, new genotypes were evolved. From the resulting novel genotypes, those that had acquired mutations that enabled them to express higher levels of cell division and lower levels of cell death quickly spread through the population. This gave rise to even more malignant genotypes exhibiting emergent cancer–like behaviour.

Although the model makes a number of assumptions including the assumption that the number of copies of a gene has a direct effect on the up or down regulation of that gene, the interactions and results can be interpreted in terms of actual biological behaviour (i.e, the up or down regulation of an oncogene or a tumour suppressor gene). The model suggests that through chromosome missegregation, the arrangement of genes on chromosomes has a profound effect on genetic diversity, giving rise to different kinds of cancer-like behaviours, which resemble key differences observed in real cancers (Cahill et al. 1999).

The role that *chromosome segregation regulatory genes* play in this model is largely determined by its position with respect to the other genes in the chromosomes. The model suggests that high levels of chromosome missegregation lead to a

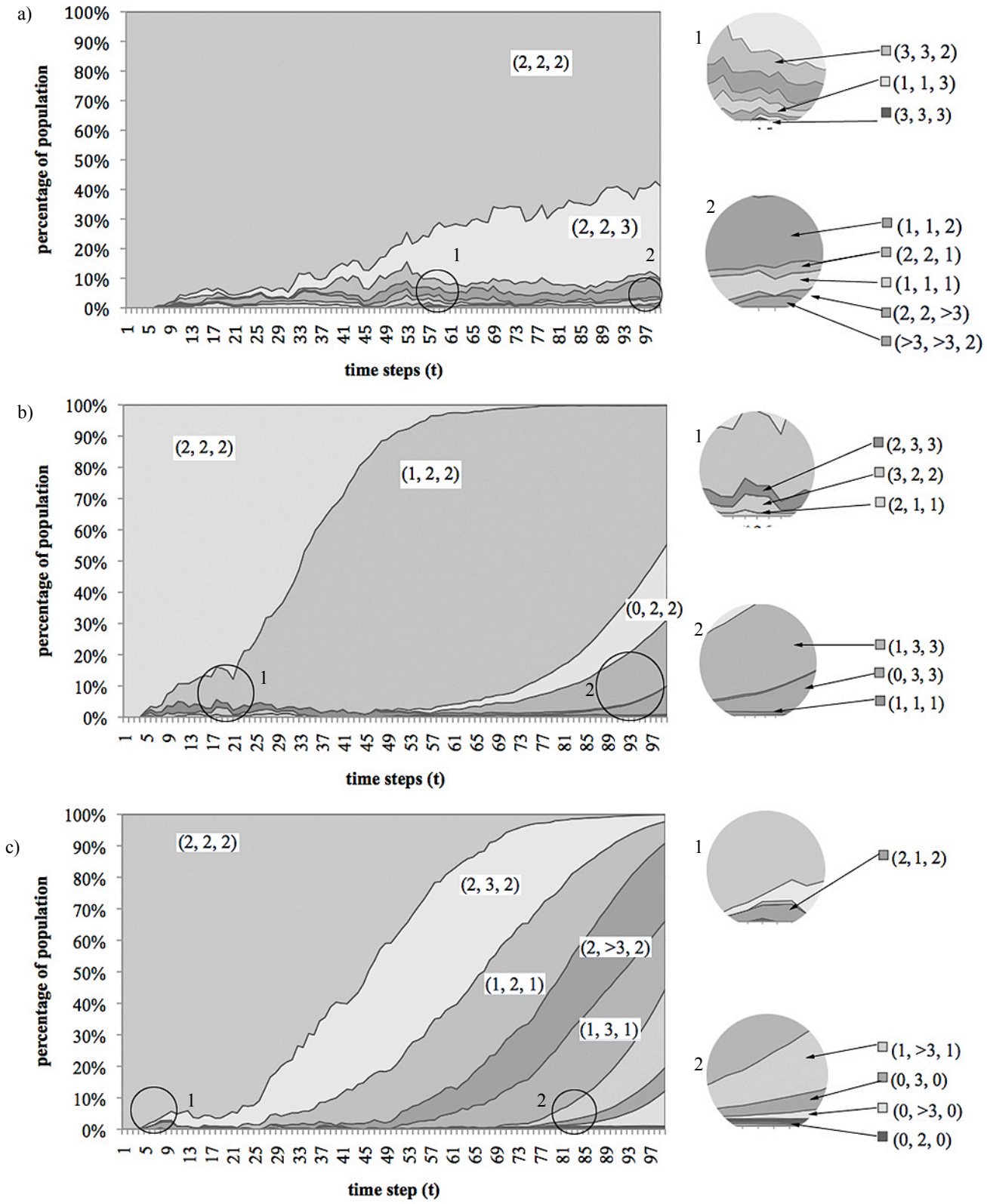


Figure 9. Genotype state population analysis for a) Gene Configuration A b) Gene Configuration B c) Gene Configuration C. The genotypes populations are stacked for each time step according to the percentage of the total population that they account for.

genetic diversity that help cells overcome the low probability of oncogenic mutations, as shown in the analysis of Gene Configuration C. Surprisingly, low levels of chromosome missegregation may also give rise to a different kind of cancer-like behaviour, as shown in the simulations of Gene Configurations B. By maintaining a relatively uniform population, specific mutations are conserved and spread throughout the population until a cancer-like genotype is reached. To determine the precise role of that *chromosome segregation regulatory genes* have in cancer systems, the development of appropriate tools for statistical analysis and further experiments are needed.

It is of interest to consider the real locations of known cancer genes to incorporate in an extension of the model. This could yield more realistic behaviour and may better inform theory and experiment. Mutations in oncogenes or tumour suppressor genes are not the only key players in real cancer systems though. Because microenvironment selection may also cooperate with aneuploidy to promote tumour progression (Anderson et al. 2006), it is also of interest to incorporate a more realistic version of the environment into the model.

Through computational models such as the one presented in this article, we anticipate that we may gain a deeper understanding of the effects of aneuploidy on cancer initiation. Identifying the key events in cancer progression may help us devise new cancer treatments that account aneuploidy and its dynamics.

Acknowledgements

Thanks to CONACYT and UCL CoMPLEX for providing the funding for this research, and to both the Baum Lab (UCL LMCB) and the Digital Biology Group (UCL CS) for the valuable input and support. BB was funded by Cancer Research UK.

References

Abbott, R. G., S. Forrest, et al. (2006). "Simulating the hallmarks of cancer." Artif Life 12(4): 617-34.

Anderson, A., A. Weaver, et al. (2006). "Tumor morphology and phenotypic evolution driven by selective pressure from the microenvironment." Cell 127(5): 905-915.

Basanta, D. and A. Deutsch (2008). "A game theoretical perspective on the somatic evolution of cancer." Selected topics on cancer modelling: genesis, evolution, inmune competition, therapy. Birkhauser, Boston.

Cahill, D., K. Kinzler, et al. (1999). "Genetic instability and darwinian selection in tumours." Trends Biochem Sci 24(12): M57-M60.

Carmona-Fontaine, C., H. Matthews, et al. (2008). "Contact inhibition of locomotion in vivo controls neural crest directional migration." Nature. 456:957-961

Chin, K., S. DeVries, et al. (2006). "Genomic and transcriptional aberrations linked to breast cancer pathophysiologies." Cancer Cell 10(6): 529-541.

Dittrich, P., J. Ziegler, et al. (2001). "Artificial Chemistries—A Review." Artificial Life 7(3): 225-275.

Futreal, P., L. Coin, et al. (2004). "A census of human cancer genes." Nat Rev Cancer 4(3): 177-183.

Gatenby, R. A., K. Smallbone, et al. (2007). "Cellular adaptations to hypoxia and acidosis during somatic evolution of breast cancer." Br J Cancer 97(5): 646-53.

Gibbs, W. (2003). "Untangling the roots of cancer." Scientific American vol. 289 (1) pp. 56-65.

Gras, R., D. Devaurs, et al. (2009). "An Individual-Based Evolving Predator-Prey Ecosystem Simulation Using a Fuzzy Cognitive Map as the Behavior Model." Artif Life 15(4): 423-463.

Hanahan, D. and R. Weinberg (2000). "The hallmarks of cancer." Cell 100(1): 57-70.

Hanks, S., K. Coleman, et al. (2004). "Constitutional aneuploidy and cancer predisposition caused by biallelic mutations in BUB1B." Nature Genetics 36(11): 1159-1161.

Holland, A. J. and D. W. Cleveland (2009). "Boveri revisited: chromosomal instability, aneuploidy and tumorigenesis." Nature Rev Mol Cell Biol 10(7): 478-487.

Huettel, B., D. P. Kreil, et al. (2008). "Effects of aneuploidy on genome structure, expression, and interphase organization in Arabidopsis thaliana." PLoS Genetics 4(10): e1000226.

Johansson, B., F. Mertens, et al. (1996). "Primary vs, secondary neoplasia-associated chromosomal abnormalities - Balanced rearrangements vs, genomic imbalances?" Genes Chromosome & Cancer 16(3): 155-163.

Lai, L. A., T. G. Paulson, et al. (2007). "Increasing genomic instability during premalignant neoplastic progression revealed through high resolution array-CGH." Genes Chromosomes & Cancer 46(6): 532-542.

Nagl, S., M. Williams, et al. (2007). "Objective Bayesian nets for systems modelling and prognosis in breast cancer." Innovations in Bayesian networks: theory and applications. Springer.

Parkin, D. M., F. Bray, et al. (2005). "Global cancer statistics, 2002." CA: A Cancer Journal for Clinicians 55(2): 74-108.

Rajagopalan, H. and C. Lengauer (2004). "Aneuploidy and cancer." Nature 432(7015): 338-41.

Weaver, B. and D. Cleveland (2006). "Does aneuploidy cause cancer?" Current opinion in cell biology 18(6): 658-667.

Zeng, Q. and W. Hong (2008). "The emerging role of the hippo pathway in cell contact inhibition, organ size control, and cancer development in mammals." Cancer Cell 13(3): 188-92.

Sticky Feet: Evolution in a Multi-Creature Physical Simulation

Greg Turk
School of Interactive Computing
Georgia Institute of Technology, Atlanta, GA 30332
turk@cc.gatech.edu

Abstract

We demonstrate artificial evolution in a system that combines physical simulation with competition between creatures. The simulated creatures are constructed using point masses that are connected by oscillating springs. The creatures pull themselves across their 2D environment by varying the amount of friction at different point masses, giving them sticky feet. Creatures combat one another, and the victor of such an encounter earns the right to reproduce, possibly with mutation. Rather than testing one individual against another in pairs, as many as 100 creatures move and interact with each other in the same 2D environment. Over time, the initial creatures are replaced by new creatures that are more agile and better at combating others. The evolved creatures from such simulations exhibit a wide array of body plans, locomotion styles, and interaction behaviors.

Introduction

The animal kingdom displays an astonishing variety of creature body plans, methods of locomotion, and styles of interactions between individuals. The engine that produces this seemingly endless array of forms and behaviors is Darwinian evolution. One of the goals of Artificial Life is to demonstrate that a similar degree of richness can be produced by unguided evolution in a computer-simulated world. Success in creating rich simulated worlds can inform our understanding of real-world evolution and may also be a valuable teaching tool, allowing students to witness a process that is slow in nature.

Our research is inspired by prior work in Artificial Life, and in particular, by simulated creature evolution through the use of physical simulation. A particular goal of our work is to create a single environment in which many creatures interact with one another, reproduce and evolve. We wish to simulate as many creatures at one time as possible in order to have a sufficiently large population in the environment. This led us to seek the most simple virtual bodies that would still exhibit a variety of shapes and behaviors. We selected point masses that are connected by springs as our representation of a creature's body. Each virtual spring may change its rest length in a cyclic manner. By changing the friction on either end of such an oscillating spring, a creature uses its sticky feet to pull itself through the environment. The creatures live in a 2D world that has no gravity and no ground plane, so the creatures may crawl in any direction. Although this is a simple virtual physics model, the evolved creatures based on this model show a considerable variety in their body shapes and motions.

In our experiments, just one small moving creature is introduced into the virtual environment. This ancestral creature is initially surrounded by stationary creatures that cannot defend themselves, and these act as food for moving creatures. The lone moving creature "eats" the stationary ones, and it replicates after doing so. After a while, many of these small moving creatures are crawling through the environment. An occasional mutation occurs during replication, and the environment is soon filled with a variety of creature types. Some of these new creatures are more successful at combat and reproduction, and eventually the ancestral creature is supplanted by its more agile descendants. Different simulation runs have exhibited a wide variety of successful creature body plans and modes of locomotion.

The remainder of the paper is divided as follows. After discussing related work, we then describe the creature bodies and the physics simulator in detail. Next, we describe the mechanism by which creatures interact and reproduce, followed by a description of the allowed creature mutations. We then present the results of our simulation runs, followed by a discussion of future work.

Related Work

There are two main lines of research that are closely related to our own, and we review the research in each area in turn. The first area of research that is related to ours is simulated physics for creature locomotion. In 1993, two research groups demonstrated the evolution of creature locomotion that is based on simple virtual physics. Van de Panne and Fiume constructed creatures from rigid segments in 2D that use linear and angular actuators in order to move. They use simulated annealing to search for control networks that lead to efficient locomotion, such as walking and jumping, for a given creature body (Van de Panne and Fiume, 1993). Ngo and Marks simulate 2D creatures that are composed of rigid linear elements and creature-controlled angular joints, and they use a genetic algorithm to evolve more efficient locomotion. Their approach produces a variety of

walking, crawling and jumping creatures (Ngo and Marks, 1993). These initial approaches were used to develop controllers for a fixed creature body plan. Sims extended this work by evolving the creature bodies as well as their controllers using a genetic algorithm (Sims, 1994b). His virtual creatures are entirely 3D, and they are composed of blocks that are connected with joints that are controlled by the creature. This approach produces compelling examples of creature motion for creatures that walk, jump and swim. Komosinski and Rotaru-Varga use a physical simulator to investigate the effectiveness of different genotype encodings to explore the space of locomotion strategies (Komosinski and Rotaru-Varga, 2001). Lipson and Pollack use physical simulation to evolve crawling creatures made of rods that they then manufacture using rapid prototyping (Lipson and Pollack, 2000). Taylor and Massey give an excellent review of much of the research that has been done using physical simulation (Taylor and Massey, 2001).

The second area related to our research is the study of virtual creature interactions. In most of this research, the creature's bodies are simple and fixed, and the creature motions are the result of simple steering. Many of the Artificial Life models for creatures that sense and move have been inspired by the essays of Braitenberg on vehicles whose behaviors are governed by simple neural circuitry (Braitenberg, 1984). Yeager's PolyWorld simulator consists of creatures with a simple body, but with complex neural circuitry to control behavior (Yaeger, 1994). A large number of PolyWorld creatures compete for food, mate, and reproduce in a single environment that allows any creature to interact with each other. Reynolds uses the game of tag to co-evolve creatures that are good at pursuit and evasion (Reynolds, 1994). He evolves more skilled creatures using a genetic algorithm, and evaluates a fitness function by playing creatures against one another in pairs. Miller and Cliff argue that co-evolving pursuit and evasion strategies is an important topic for robotics and other applications (Miller and Cliff, 1994). Ventrella demonstrates evolution of swimming creatures in a simulated environment in which many creatures interact (Ventrella, 1996). His swimmers compete for food and then mate in order to reproduce. Miconi simulates a micro-world of block creatures that inflict damage on one another and that reproduce based on their health status (Miconi, 2008). The systems of Ventrella and Miconi are similar to our own in combining physical simulation with a multi-creature environment that fosters between-creature competition.

Creature Locomotion

A main goal of our research is to create a simulated environment in which many creatures can interact with one another. Because we wanted to simulate many creatures at once, we use an artificial physics that is computationally inexpensive, yet still capable of creating a wide range of motions. Our creatures are made of a collection of point-masses that are connected by segments that are each linear springs. Each of these segments can be directed to change its rest length in a periodic manner, which causes the segment to oscillate in length. These creatures live on a 2D plane that has no gravity, so that there is no preferred orientation. This means that a given creature can approach another creature from any direction.

The equations that govern the motion of these creatures are those of a damped spring. For a segment that connects points with positions P_i, P_j and velocities V_i, V_j, the spring force F_i^{spring} acting on mass i is given by:

$$\begin{aligned} L &= P_i - P_j \\ \dot{L} &= V_i - V_j \\ F_i^{spring} &= -(k_s(L - L_{rest}) + k_d \frac{(\dot{L} \cdot L)}{|L|}) \frac{L}{|L|} \end{aligned}$$

In the above equations, after (Baraff et al., 1999), the spring strength k_s and spring damping coefficient k_d are set to be the same across all springs of all creatures. L_{rest} is the rest length of a particular segment, and it can vary periodically. For a segment with an original length L_{seg}, oscillation amplitude a, frequency f, and phase p, the change to its rest length is given by:

$$L_{rest} = L_{seg}(1 + a \sin(ft + 2\pi p))$$

The most simple creature that can move is composed of two point masses that are connected by a single segment. If this segment oscillates in the absence of other external forces, the creature's center of mass will remain unchanged. In order for such a creature to move, this creature must have a way to gain traction. Miller solved the traction problem by used directional friction in order to simulate the motion of worms and snakes (Miller, 1988). Our simulated creatures gain traction in a similar manner, by periodically changing the coefficient of friction at the two endpoints of the spring in synchrony with the oscillation of the segment itself. That is, each point alternates being sticky or slippery. Figure 1 shows a one-segment creature where the leading point is on the top and the trailing point is on the bottom. When the segment is elongating, the lead point is frictionless (shown using a smaller radius) and the trailing point is given a high friction coefficient (shown with a larger radius). This pushes the creature in the direction of the leading point. When the segment is shortening, the leading point is sticky and the trailing point is allowed to slide, and this pulls the creature towards the leading point. This single-segment creature moves along much like an inchworm.

To modify the stickiness of a given point-mass i, a per-point friction force is calculated that is proportional to the point's velocity and a global friction coefficient k_f:

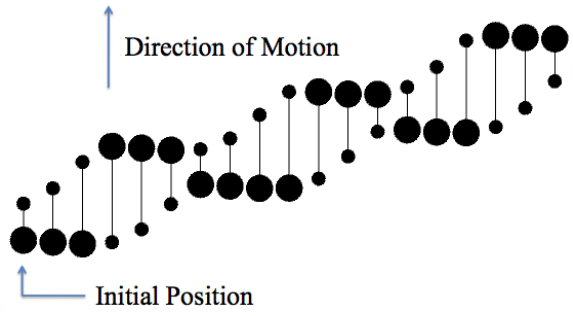


Figure 1: A simple one-segment creature that moves upwards. The creature's initial state is shown at the left, and subsequent positions in time are shown to the right of this. Points with high friction have a large radius, and the smaller points have lower friction.

$$F_i^{friction} = -k_f f_i V_i$$

This per-point friction is modulated based on the phase of any spring segment that is attached to point i. We define a coefficient f_k for a given spring that is based on a per-spring friction magnitude m_k and the phase of the oscillation:

$$f_k = \begin{cases} -m_k & \text{if } \cos(ft + 2\pi p) \leq 0 \\ m_k & \text{if } \cos(ft + 2\pi p) > 0 \end{cases}$$

If a given spring k connects particles i and j, the friction coefficient f_k is *added* to particle i's friction accumulation f_i, and f_k is *subtracted* from particle j's friction accumulation f_j. Thus a spring will cause one of its particles to become more sticky and will cause the particle on its other end to be more inclined to slide. Once the friction accumulation f_i for a given particle has been modified by all of the attached springs, the value of f_i is then clamped to the range $[0, 1]$. Freely sliding particles have a value for f_i at or near zero, and points with larger values of f_i are "sticky", and resist motion.

Figure 2 shows a more complicated creature body, consisting of three point-masses and three segments that form a triangle. Assume that two of the segments oscillate out of phase with each other, so these sides of the creature lengthen and shorten alternately. Further assume that the third segment does not oscillate. If the friction magnitudes m of the two changing segments are the same, then the two sides take turns pushing the creature forward. Such a creature moves forward with a locomotion gate that looks like a waddle.

Our use of oscillating springs was partly inspired by the SodaPlay mass-spring simulator (Burton, 2007). Constructions in SodaPlay consist of point masses that are connected by springs. Any spring may be set to vary its length in a periodic manner, and constructions from such springs and point masses move around in a 2D environment. Unlike our model, SodaPlay constructions live in an environment with gravity and a floor.

Competition and Sensing

Pursuit and evasion contests are among the most common types of creature interactions in nature. Predators chase their prey, and the prey try to evade capture. Creatures of the same species chase each other when they are competing for food or mates. Because of the real-world importance of these behaviors, several researchers have made convincing arguments in favor of studying pursuit and evasion in artificial simulations (Reynolds, 1994; Miller and Cliff, 1994). Our own work takes inspiration from this prior work, and the artificial evolution in our simulator is driven by the success that the creatures have in pursuing one another.

Because our simulated creatures are composed of multiple point masses and segments, we must define what it means for one creature to capture another. Each creature has one of its point masses designated as its *mouth*, and a different point-mass as its *heart*. One creature successfully captures another when the mouth of the first creature comes within a specified radius of the heart of the second creature. There are several consequences of this model of competition. First, it means that any creature may be the aggressor or the chased. Second, it is very unlikely that a pair of creatures simultaneously capture each other. Finally, it allows the morphology of a creature to be tailored to the nature of the mouth and the heart. For instance, a successful creature is likely to have its mouth placed forward relative to its direction of motion.

All of our virtual creatures live together in one large 2D world, and a typical population consists of 100 creatures. Creatures encounter each other as they crawl forward in this world. When one creature successfully captures another, the victor of the encounter is rewarded by being copied (reproduction), and the loser of the encounter is removed from the simulation. In this manner, the creatures that are more suc-

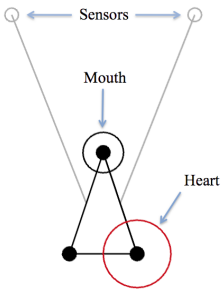


Figure 2: A simple triangular creature with three point-masses and three segments. Two of the segments have sensors attached to them that extend in front of the creature. The mouth of this creature faces its direction of motion, and its heart is in a trailing position.

cessful at pursuit become more numerous, and the creatures that often lose such competitions are eventually eliminated from the population. In most cases, the winner of the competition is duplicated exactly, but on occasion there may be one or more mutations that occur during reproduction. In this way, new creature body plans and behaviors can emerge.

This mechanism of closely tying competition with reproduction is similar to a steady-state genetic algorithm, since exactly one member of the population is replaced at a given time. In our simulations, the fitness evaluation is the outcome of a single encounter. This is a departure from much of the prior work on artificial creature evolution, where the fitness function for a creature is usually determined by mini-tournaments between pairs of creatures (Reynolds, 1994; Miller and Cliff, 1994; Sims, 1994a). We believe that having all the creatures in one large environment is closer to modeling the real world than the alternative of mini-tournaments. In addition, placing all of the creatures in a single environment allows for a richer set of encounters. A creature has to select its own prey, and may change to another creature target mid-way through an attack. It is possible for one creature to be both the pursuer of a second creature, and to simultaneously be chased by a third creature.

In order for a creature to recognize the presence of another creature, each creature can modulate the motion of its segments based on the output of proximity sensors. More specifically, each segment can have one sensor that is tied to that particular segment. Thus a creature that is composed of three segments may have up to three sensors, one for each of its segments. Each sensor recognizes the presence of either a heart or a mouth of another creature. A sensor is defined by several attributes: its position relative to the segment, its sensing radius, what body part it senses (heart or mouth), its modulation strength, and the type of modulation that it uses to affect its segment.

A sensor has an all-or-nothing response, depending on whether another creature's heart or mouth is inside the sensor's radius. Each sensor has a modulation strength m that can be positive or negative. If a sensor is triggered, it changes the property of the oscillating segment that it is tied to in one of three ways. When triggered, the sensor modulation strength m may be added to the amplitude of the segment's length a, it may be added to the friction force f_k of the segment, or it may be added to the friction magnitude m_k of the segment. Thus a sensor may cause a spring to oscillate more or less, it may cause points to become more or less sticky, or it may alter which of the endpoints of a spring are sticky at a given time (possibly slowing or reversing the direction of motion).

Figure 2 shows a three-segment creature that has two sensors that are positioned forward of the creature's direction of motion. Assume that each of these sense the proximity of another creature's heart, and that upon doing so, this causes the magnitude of the spring oscillations to decrease. If the presence of another creature's heart sets off the right-hand sensor, this will cause the creature to be pushed forward more weakly on its right side. This makes the creature turn towards the creature that triggered its sensor. In this way, a simple creature can sense and pursue other creatures. This method of steering based on sensors and motors is in the spirit of Braitenberg's vehicles (Braitenberg, 1984).

Both proximity sensing and the determination of creature capture require testing whether one point is within a given radius of another point. In a naive implementation, testing whether each creature's mouth is near any other creature's heart requires $O(n^2)$ operations for n creatures. We speed up this test by first noting that each creature's heart has a fixed radius r. To rapidly determine mouth/heart proximity, we first create a grid of square cells with side lengths r that is superimposed on the 2D environment in which the creatures live. Each cell in this grid maintains a list of the hearts that fall within the cell at the current time-step. To test whether a creature mouth is near to any hearts, only nine cells need to be checked, namely the cell that the mouth is currently in and the eight neighboring cells. Testing whether a sensor is close to a mouth or a heart is similar, but in this case the cell size is given by the maximum radius of all sensors.

Creature Reproduction

When one creature captures another, it is rewarded by being replicated, possibly with mutation. In our simulations we used a mutation rate of 0.1, so that one out of ten creature replications occurs with mutation. This is a much higher mutation rate than is typically used in a genetic algorithm. Note that in a genetic algorithm, most of the variation is generated by crossing-over, and we do not have such a mechanism in our simulator. We also have a fairly high probability of multiple mutations during reproduction. If a creature is to be mutated, there is a probability of 0.4 that it will have a second mutation, 0.4^2 that it will have three mutations, 0.4^3 for four mutations, and so on. Mutations can be grouped into three categories: per-segment physical parameters, sensor parameters, and creature body shape.

Per-segment mutations alter parameters that are specific to a segment that is chosen at random. The possible mutated parameters are segment length, amplitude of oscillation, frequency of oscillation, phase of oscillation, and the magnitude of change that the segment uses to alter the friction of its endpoints.

A sensor mutation alters one of the parameters that guides the action of a creature. For most of these, a segment is chosen at random and the parameters of the segment's sensor is altered. Potential changes include the angle of the sensor relative to the segment's orientation, the distance of the sensor from the segment, the radius of the sensor, and the sensor type (senses mouth or heart). There are three other

mutations that change what the sensor modifies (segment length amplitude a, friction force f_k, or friction magnitude m_k). When one of these three mutations occurs, the sensor is switched to modifying a particular segment parameter, and a new sensor modulation strength m is chosen. A final type of behavioral change that can occur is the verbatim copying of all the sensor parameters from one segment to another. This mutation was designed in recognition of the fact that many advances in biological evolution occur due to duplication and divergence.

The final class of mutations are changes to the creature's body plan. One such mutation modifies the position of either the heart or the mouth at random. Another body mutation deletes a segment at random. This mutation is only deemed valid if deleting the segment would not separate the creature into disjoint components. Another mutation adds a segment that is attached to the other segments only at one end, producing a dangling segment. Note that such dangling segments can still contribute to a creature's motion. One mutation fuses two such dangling segments, and another connects two dangling ends with a new segment. Finally, one mutation attaches two new segments to an already existing segment in a manner that forms a new triangle.

When a creature is replicated, regardless of whether or not it is a mutation, the new creature is placed in the 2D environment at a random position and orientation. The placement algorithm makes sure that the creature's segments do not overlap with any already existing creatures. This is done by repeated attempts to place the new creature at random locations in a non-overlapping manner. The placement algorithm can in rare cases terminate unsuccessfully after a fixed number of placement attempts, and the maximum number of placement trials is set to 40 in our simulations. Such placement failures are an indication that the creatures are developing substantially larger bodies, and in such cases the population size gradually decreases (through placement failures) to accommodate this change in creature body size.

Simulation Results

We ran three classes of simulations, namely lone ancestor runs, between-generation contests of evolved creatures, and a tournament across creatures from many different simulation runs. We report on each of these kinds of simulations in turn. (Video of these results can be found at `http://www.cc.gatech.edu/~turk/stickyfeet/`)

All of the lone ancestor simulations were conducted using the same initial conditions, with the only difference between runs being differences in the random number seeds. In each of these runs, the simulation begins with 100 creatures. All but one of these initial creatures are motionless one-segment creatures with hearts but without mouths. By design, these static creatures cannot win an encounter with another creature. In effect, these 99 motionless creatures act as a potential food source for other creatures. The one moving creature had a one-segment body, and it moves by changes to its segment length and by synchronized changes in friction to its two point-masses. This forward motion is illustrated in Figure 1. The forward point-mass of this creature is its mouth, and the back point is its heart. Sensors are not shown in this and later figures to avoid visual clutter.

At the start of the simulation, the lone moving creature

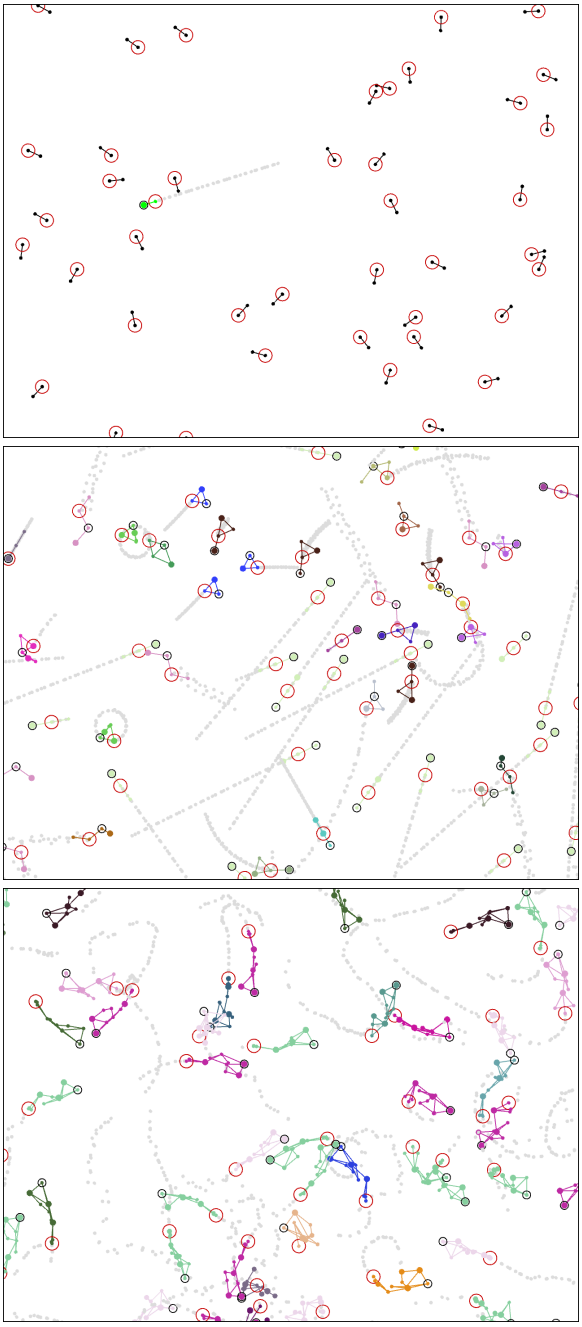


Figure 3: The initial state of the simulator, with a single green moving creature (top), and later snapshots of such a simulation run (middle, bottom).

crawls through an environment that is filled with stationary creatures, as in Figure 3 (top). This moving creature will be the ancestor of all the subsequent creatures in the simulation. At some point, this ancestral creature strikes the heart of a static creature. The static creature dies, and the ancestral creature is replicated. Further encounters with static creatures occur, and more single-segment creatures are born. All of these early creatures only travel in a straight line. The 2D environment is a rectangle with toroidal boundary conditions, so that a creature that wanders off one side of the world re-appears on the opposite side.

At some point, one of the early creatures is replicated with mutation, and then there is variation in the creature population. As the proportion of moving creatures increases, encounters between pairs of moving creatures start to occur. Encounters between identical one-segment creatures are won based on their relative positions and orientations (that is, essentially at random). Encounters between creatures with different body plans are more interesting, since there is the possibility that one of the creatures is more likely to win based on its body plan and behavior.

The creatures that evolve are different each time a lone ancestor simulation is run, due to using different random number seeds. (A typical simulation to 2,000,000 time steps requires roughly 4 hours of simulation time on a single 2.8 GHz processor.) Some general trends in creature success become apparent by observing the creatures in such runs. First, it is an advantage for a creature to move fast. Faster motion implies more frequent encounters with other creatures, and thus more opportunities to reproduce by winning such encounters. A pair of commonly successful features is to have the mouth near the front of the creature and have the heart near the back with respect to the direction of motion. Having a forward-positioned mouth means that this creature will be more likely to strike another creature's heart first, before that other creature has an opportunity to do so. A similar reasoning holds for the advantage of having a rear-positioned heart. Related to this is that many successful creatures cause their mouth to wave back and forth rapidly. This is an advantage because such a moving mouth is more likely to strike another creature's heart. Conversely, the motion of the heart in a successful creature is typically quite damped in comparison, and the heart is often dragged by a segment that has little or no oscillatory motion.

There is remarkable variation in body plans for fast moving creatures. In order to get a sense of the variation between runs, we performed 100 such lone ancestor simulations. Figure 4 shows the bodies of the most successful creatures from these 100 lone ancestor runs. Some creatures have elongated worm-like bodies, and they coordinate their segment oscillations to make rapid forward progress. Other creatures are composed of one or more triangles, and often such creatures seem to pulse in a manner that helps their forward progress while at the same time causing their mouth to swing back and forth. Some creatures do not move in a straight line, but instead rotate in a circle, usually quite rapidly. Some creatures have triangles that form a compact body, but also have one or more "legs" that help to push them forward. Some of these legged creatures move with a limping gate, while others move in a smoother manner. One effective mode of locomotion is to have two trailing segments whose oscillation periods are offset from each other, so that while one segment is shortening and pushing the body forward, the other segment is elongating in the recovery phase of its duty cycle. The trails of creatures in Figure 3 (middle) illustrates some of the variations in motions of different creatures.

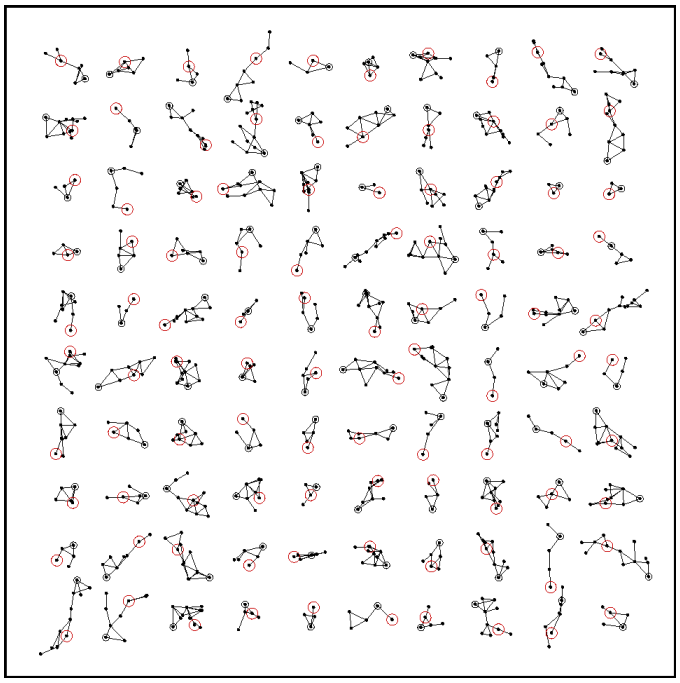


Figure 4: The most successful creatures from 100 different lone ancestor simulations. Each represents the most numerous type of creature at time step 2,000,000 for a particular simulation.

There is a limit to how fast a creature can move, given that there is a limit on segment lengths and on the frequency and magnitude of segment oscillations. There is, however, another avenue for creature evolution, and that is the ability to sense and react to other creatures. In many lone ancestor simulations, eventually some creatures arise that will turn their bodies towards the heart of another creature. This is the beginning of hunting behavior. Early in the development of this trait, a creature typically can only sense and turn to one side (e.g. just towards the right). Even more successful creatures are ones that can sense and turn towards creatures on either side. There is considerable room for fine-tuning this hunting behavior, including adjusting the placement and radius of the sensors, and modifying the magnitude of the turning response when a sensor is triggered. Figure 3 (bottom) shows creatures that have evolved hunting behaviors.

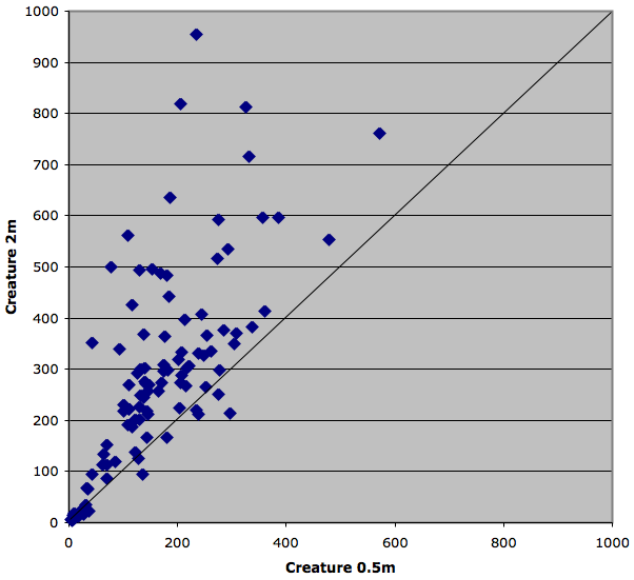


Figure 5: Results from 100 between-generation creature contests. The horizontal axis is the number of captures by the earlier generation creature (time step 0.5 million), and the vertical axis is the capture count for the later generation creature from the same simulation run (time step 2 million).

Contests Between Generations

Although it appears to a human observer that later generations of creatures are more successful than earlier creatures, we used between-creature contests to determine whether this is in fact the case. Specifically, each of these contests is between two creatures that both evolved in the same simulation run. In a given contest, one of the creatures is the most numerous from time step 500,000 and the other is the most abundant creature from the same simulation run at the later time of 2,000,000. The goal is to see which of the two creatures can score the most captures in a fixed number of time steps. We ran 100 of these contests, one for each lone ancestor simulation run.

At the start of each between-generation contest, there are 50 copies of each creature. The rules of reproduction are modified so that this 50-to-50 ratio is always maintained throughout the contest. Instead of reproducing the victor of a creature encounter, the loser is removed from its current location and placed at a random location elsewhere in the environment. This transportation of the loser is performed regardless of whether a creature captures a different kind of creature, or whether it captures a replica of itself. There is also no mutation during the contest. A count is kept of the number of captures for each of the two types of creatures, and this count ignores same-type captures.

Figure 5 reports on the results of the 100 between-generation contests. Each point represents one contest. The horizontal axis is the number of creature captures by the 0.5 million time step creature (the earlier creature) and the vertical axis is the number of captures by the 2 million time step creature. Points below the diagonal line indicate more captures by the earlier creature, and points above the diagonal indicate that the later creature had more captures. There were 11 contests that were won by the earlier creature, 87 won by the later creature, and 2 ties. Note that later generations often made substantially more captures in many of the contests. We take this as verification that our rules for capture and reproduction are indeed effective at evolving creatures that are better suited for survival in a multi-creature environment.

Tournament Across Simulations

Although all of the creatures that evolved from the lone ancestor runs appeared to have adaptations for survival, there was a considerable variation in their modes of locomotion and their behavior. We wanted to find how these creatures from different simulations compared to each other when placed in the same environment. In order to explore this, we created a two-tier creature tournament. The first tier consisted of 10 contests, with 10 creatures in each contest. The 10 winners from these contests advanced to the second tier, where these creatures competed in a final contest that had a single victor. Figure 6 shows a frame from such a second tier tournament.

Each of the contests in the tournament began with 10 different types of creatures, and 10 copies of each of these creature types. The contest rules in this tournament differ from the between-generation contests. In particular, the winner of each encounter is copied, causing some creature types to become less or more numerous over time. No mutations occur during these contests. A contest ends when one type of creature is the lone survivor.

The bright green creature in Figure 6 is the tournament winner. As judged by these tournament, this is the most effective predator from the 100 lone ancestor simulation runs. This creature has 10 mass points and 13 segments. The body of this creature exhibits several innovations that evolved in order to make it a success. These innovations include a mouth that swings from side-to-side, a heart that is positioned on a “tail” that is dragged behind, the overall coordination between the oscillating segments that propels it forward, and sensors on both sides that steer it towards prey. The creature had been molded into this form by its numerous encounters with other creatures. In its own simulation environment, this creature is more deadly than all of its rivals. Nevertheless, the most simple real-world bacteria cell is still vastly more complex than this artificial creature. Despite this wide gulf in complexity, we believe that our results give an indication that multi-creature physical simulations can bring Artificial Life closer to simulating open ended complexity.

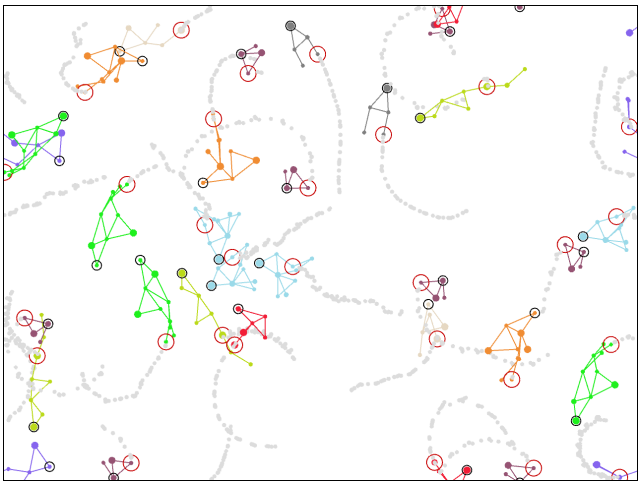


Figure 6: Tournament between the best 10 creatures. For better visibility, only three of each creature has been placed into the environment. The bright green creature is the tournament winner.

Conclusion and Future Work

We make two claims of novelty in our approach to simulated evolution:

- We simulate locomotion by dynamically changing the friction at either end of an oscillating spring.
- Our simulator combines pursuit/evasion behavior with the ability to evolve new physical configurations for locomotion.

A third important attribute of our simulator, shared by other researchers (Ventrella, 1996; Miconi, 2008), is that our simulated life-forms evolve in a large multi-creature environment that is driven by a simple physics engine. Taken together, these attributes create a rich synthetic environment for the evolution of artificial creatures.

There are several logical avenues for future research. First, there are other physical attributes that the virtual creatures could use to broaden their styles of locomotion even further. Oscillating torsional springs is one such possible addition. Another direction would be to add a more realistic energy model to the simulator. Still another fruitful avenue would be to replace the asexual reproduction model with sexual reproduction. Finally, it would be interesting to add a developmental process to our creatures, since some researchers have found that more successful body plans can result (Komosinski and Rotaru-Varga, 2001).

Acknowledgements

We thank the anonymous reviewers for their excellent suggestions for improving this paper. We are grateful to Mary McFarlane for video narration. This work was funded in part by NSF grant CCF-0811485.

References

Baraff, D., Kass, M., and Witkin, A. (1999). Physically Based Modeling, Course Notes. *Course 36, SIGGRAPH 1999*.

Braitenberg, V. (1984). *Vehicles: Experiments in synthetic psychology*. MIT press Cambridge, MA.

Burton, E. (2007). Sodaplay constructor. `http://www.sodaplay.com/`.

Komosinski, M. and Rotaru-Varga, A. (2001). Comparison of different genotype encodings for simulated three-dimensional agents. *Artificial Life*, 7(4):395–418.

Lipson, H. and Pollack, J. (2000). Automatic design and manufacture of robotic lifeforms. *Nature*, 406(6799):974–978.

Miconi, T. (2008). Evosphere: evolutionary dynamics in a population of fighting virtual creatures. In *IEEE Congress on Evolutionary Computation 2008*, pages 3066–3073.

Miller, G. (1988). The motion dynamics of snakes and worms. In *SIGGRAPH '88*, pages 169–173.

Miller, G. and Cliff, D. (1994). Protean behavior in dynamic games: Arguments for the co-evolution of pursuit-evasion tactics. In *From animals to animats 3*.

Ngo, J. and Marks, J. (1993). Spacetime constraints revisited. In *SIGGRAPH '93*, pages 343–350.

Reynolds, C. (1994). Competition, coevolution and the game of tag. In *Proc. Artificial Life 4*, pages 59–69.

Sims, K. (1994a). Evolving 3D morphology and behavior by competition. In *Proc. Artificial Life 4*, pages 28–39.

Sims, K. (1994b). Evolving virtual creatures. In *SIGGRAPH '94*, pages 15–22.

Taylor, T. and Massey, C. (2001). Recent developments in the evolution of morphologies and controllers for physically simulated creatures. *Artificial Life*, 7(1):77–87.

Van de Panne, M. and Fiume, E. (1993). Sensor-actuator networks. In *SIGGRAPH '93*, pages 335–342.

Ventrella, J. (1996). Sexual swimmers: Emergent morphology and locomotion without a fitness function. *From animals to animats*, 4:484–493.

Yaeger, L. (1994). Poly world: Life in a new context. In *Proc. Artificial Life III*, pages 263–263.

Interactive Evolution of Camouflage

Craig Reynolds

Sony Computer Entertainment, US R&D
craig_reynolds@playstation.sony.com

Abstract

This paper presents an abstract computation model of the evolution of camouflage in nature. The 2d model uses evolved textures for *prey*, a background texture representing the *environment* and a visual *predator*. In these experiments, the predator's role is played by a human observer. They are shown a *cohort* of ten evolved textures overlaid on the background texture. They click on the five most conspicuous prey to remove ("eat") them. These lower fitness textures are removed from the population and replaced with newly bred textures. Biological morphogenesis is represented in this model by *procedural texture synthesis*. Nested expressions of generators and operators form a texture description language. Natural evolution is represented by *genetic programming,* a variant of the *genetic algorithm*. GP searches the space of texture description programs for those which appear least conspicuous to the predator.

Introduction

That animals often resemble their environment has been observed since ancient times. This sometimes incredible visual similarity highlights the adaptation of life to its environment. Since the earliest publication on evolution, camouflage has been cited as a key illustration of natural selection's effect:

> When we see leaf-eating insects green, and bark-feeders mottled-gray; alpine ptarmigan white in winter, the red-grouse the colour of heather, and the black-grouse that of peaty earth, we must believe that these tints are of service to these birds and insects in preserving them from danger.
> – Charles Darwin, 1859
> *On the Origin of Species by Means of Natural Selection*

Natural camouflage appears to result from coevolution between predator and prey. Many predators use vision to locate their prey, so prey have a survival advantage if they are harder to see. Predators with superior vision are better able to find prey, giving them a survival advantage. Over time this leads to well camouflaged prey and to predators with excellent eyesight and a talent for "breaking" camouflage.

The hypothesis for these experiments was that selection pressure from a visual predator will gradually eliminate the most conspicuous (least well camouflaged) prey from the evolving population. Prey would then converge on more effective camouflage. The results presented here lend support to this idea and point the way to more powerful human-computer hybrid systems as well as future simulation studies of the *coevolution* of prey camouflage and predator vision.

As defined in (Stevens and Merilaita, 2009) the term *camouflage* includes all strategies of *concealment*. To distinguish from *hiding*, this is taken to mean reducing the chance of recognizing an animal which is otherwise in plain sight. (Thayer, 1909) describes a bird "in plain sight but invisible." The more specific term *crypsis* refers to preventing initial detection, including the sort of *cryptic coloration* commonly implied by the term camouflage. For comparison, crypsis helps prey avoid detection while *mimicry* protects by leading predators to misclassify prey after detection.

A common misconception about camouflage is that ideally it should match the background. This is generally untrue except for homogenous environments like white snow or green leaves. Consider a color-matched and borderless photographic print of an environment, say the surface of a rock. If the print is placed on the rock it will not be perfectly cryptic. Discontinuities at the edge of the print stimulate low level edge detectors in the visual system, causing a strong perception of a rectangle. Moving the print to another location on the rock will also reveal subtle variations in color and texture which add additional contrast at the edge of the print.

Much recent work on camouflage (see next section) has focused on the importance of *disruptive camouflage*. While

Figure 1: camouflaged "prey" overlaid on the background image for which they were evolved
(a) tree bark, (b) twisty wire, (c) flowers, (d) serpentine, (e) Yosemite granite

Figure 2: these camouflaged prey are only partially or occasionally effective, features in this *peppers* background were too large to "solve"

these patterns often echo colors and textures from the environment, their effectiveness comes from their ability to visually disrupt the visual silhouette of an animal. This can prevent a predator from recognizing that an object is an animal, or even prevent the detection of an "object" in the first place, see (Schaefer and Stobbe, 2006). Paradoxically, camouflage that does not match the background can be more effective through the use of strong visual features (false edges) that intersect the object's real edges (Stevens and Cuthill, 2006). Most of the effective camouflage patterns evolved in these experiments appear to have disruptive qualities.

The work described here lies between computer science and evolutionary biology. This multidisciplinary middle ground is variously called *theoretical biology*, *mathematical biology* or *artificial life*. Research in this middle area has the potential to benefit all related fields. From a computer graphics perspective, this could be seen as a special case of *goal-oriented texture synthesis* where new textures can be created from a description of desired image properties. To biologists, a computation model of camouflage evolution could allow new types of theoretical experiments to be conducted in simulation which are not subject to constraints imposed by working in the field, or with live animals, and in general is not limited to examples found in Earth's biosphere.

Related Work

Over the last century several seminal works have surveyed the broad topic of camouflage in nature. These include (Beddard, 1895), (Thayer, 1909) and (Cott, 1940). The latter two continue to be widely cited today. Over the last 20-30 years there has been a significant renaissance in the study of camouflage. Before that, work in this area tended to be more descriptive than experimental. It is challenging to design well-controlled studies of the effectiveness of camouflage in either the field or the laboratory. Still with careful design and patient experimentation, studies providing new insights have appeared regularly in the biological literature. For an excellent recent survey, see (Stevens and Merilaita, 2009).

Of particular relevance to the work presented here are various experiments offering *artificial prey* to real predators. Many valuable results have been obtained with a similar experimental design involving "cardboard moths" (Cuthill, et al. 2005) and avian insectivores: wild birds that naturally prey on moths. During the day these nocturnal moths rest on tree trunks protected by their cryptic wing coloration. Artificial moths are constructed with cardboard wings decorated with a color printer, a worm is attached to serve as an edible "body," and the "moth" is attached to a tree trunk. A missing worm is taken to indicate that a wild bird detected and attacked the moth. This technique has shown the key important of disruptive coloration (Schaefer and Stobbe, 2006), measured the disadvantage of symmetrical camouflage (Cuthill, et al. 2006), and several related topics.

Other experiments have used live captive birds (Bond and Kamil, 2002) and humans (Sherratt, et al. 2007) as predators of "virtual artificial prey" on a display screen. In both cases this predation was used to drive a evolutionary computation like in the work described here. In (Merilaita, 2003) artificial predators learn to detect artificial prey whose camouflage evolves to avoid detection. However the textures used are quite small, 4 to 8 symbolic pixels. A recent simulation-based study looked at a unique three-player camouflage game based on evolution of flower color (Abbott, 2010).

The original idea of using an interactive task as the fitness function for an evolutionary computation goes back to the *Blind Watchmaker* software that accompanied (Dawkins, 1986). That application displayed a grid of *biomorphs*, small tree-structured line drawings with a genetic description. The user picked a favorite which was mutated several times to produce a new generation. Dawkins introduced the idea of intentionally evolving toward a goal, a biomorph he called the "holy grail." Karl Sims combined a similar approach with genetic programming and a rich set of image processing functions to create an interactive system for aesthetic evolution of texture patterns (Sims, 1991). In (Funes, et al. 1998) and other papers, Jordan Pollack's DEMO group describe their TRON project where game-playing agents were evolved in competition with each other and then in competition with human players over the web. A survey of related techniques used to create game content is presented in (Togelius, et al. 2010). A deep survey of the whole field of *interactive evolutionary computation* is found in (Takagi, 2001).

This work conceptually overlaps the large body of work in *example-based texture synthesis*, also known as *texture extension*, which creates arbitrarily large texture patterns to match a small exemplar texture (Wei, et al. 2009). Using this technique to generate camouflage image puzzles is described in (Chu, et al. 2010). In contrast, the synthesis of camouflage texture described in this paper does not "see" or otherwise access the input texture. Instead the background can only be *inferred* from the indirect evidence of predation, as it is in evolution of natural camouflage.

Texture Synthesis

In nature, patterns of surface coloration on plants and animal result from complex genetic and developmental processes collectively called *morphogenesis* (see for example, (Eizirik, et al. 2010)). In this simulation model, pattern formation is

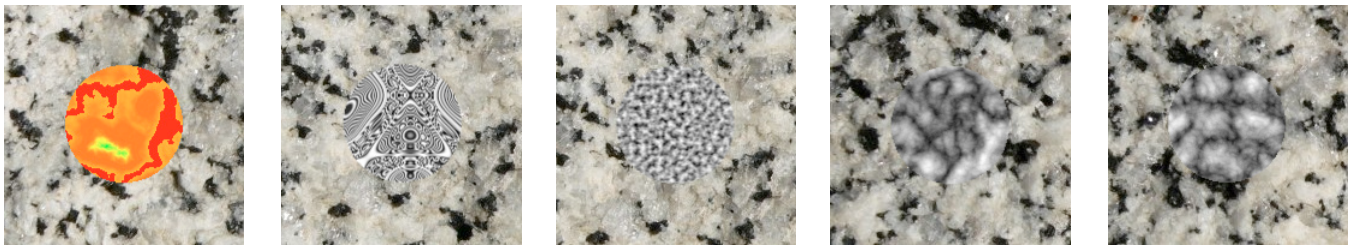


Figure 3: progression of camouflage patterns during a run with the granite environment

represented by procedural texture synthesis (Ebert, et al. 1994). More specifically, this work uses *programmatic* texture synthesis. Textures are defined by nested expressions of generators and operators, forming a programming language for textures. Generators produce results of type Texture from simple types (numbers, 2D vectors and RGB colors). Operators are similar but have one or more Texture parameters. These nested expressions look like composition of functions (see Figures 12 and 13) although in this implementation they are specifically constructors for C++ classes representing the various types of procedural textures. Once the tree of procedural texture objects is constructed, its root provides an interface for rendering pixels.

This texture synthesis library (Reynolds, 2009) brings together several preexisting techniques. Its generators include uniform colors and simple patterns like spots and color gradations. There are a collection of gratings (e.g. a sine wave grating) and an assortment of noise patterns such as *noise* and *turbulence* (Perlin, 1985) plus variations on these. The library's collection of texture operators include simple geometrical transformation (such as scale, rotate and translate), simple image processing operations (add, subtract, multiply, adjustment of intensity, hue, saturation), convolution-based operations (blur, edge detect, edge enhance), operators to produce multiple copies of a texture (row, array, ring), and a collection of image warping operators (stretch, wrap, twist, ...). Several operators use a 1D "slice" of a texture, such as colorizing one texture by mapping its brightness into colors along the y=0 axis of another texture. Only convolution-based texture operators have fixed pixel resolution, all others use floating point coordinates. The complete texture synthesis API used in these experiments is listed in Appendix 1. Missing from the library are reaction-diffusion and other compute heavy textures, awaiting a GPGPU implementation.

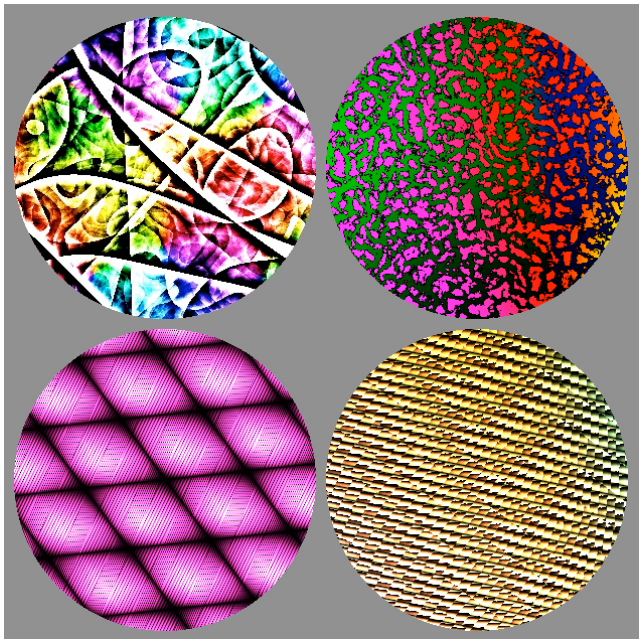


Figure 4: "random" textures automatically evolved with genetic programming using a non-interactive *ad hoc* fitness function

Evolutionary Computation

The texture synthesis library described in the previous section was designed for use with *genetic programming* (Koza, 1992). Like the closely related *genetic algorithm* (Holland, 1975), GP is a stochastic technique for population-based (parallel) search and optimization in high dimensional spaces. These *evolutionary computation* (EC) techniques are inspired by evolution in the natural world and share some of its attributes. While GP is used in this work as a model of natural evolution it is important to keep in mind the vast differences between the two. For example, natural evolution works with very large populations and very long time scales. Much of the engineering in evolutionary computation has to do with getting useful results without requiring billions of individuals or waiting millions of years.

A genetic programming system maintains a population of *individuals*, each of which represent a program expressed in a given grammar. In this work, each individual is a program that defines a procedural texture. These programs can be thought of as nested expressions of composed functions, or as a tree of functional nodes. The GP population is initialized to randomly generated programs. GP uses a given fitness function (objective function) to evaluate each individual. Fitness is used to select which individuals will reproduce to create new offspring programs to replace lower fitness individuals in the population. New individuals are created by genetic operators such as *crossover* and *mutation*. GP crossover involves replacing a sub-node of one program with a sub-node of another program. It is essentially "random syntax-aware cut-and-paste" between programs. Over time, programs containing beneficial code fragments become more numerous in the population. Crossover tweaks these programs, juxtaposing code fragments in new ways. Some changes improve fitness and some reduce fitness, but the population is biased to collect the good and discard the bad.

For these experiments, genetic programming was implemented with the excellent open source library Open BEAGLE (Gagné and Parizeau, 2006), (Open BEAGLE, 2002). This flexible framework provides support for many common types

of evolutionary computation while also allowing customization of all aspects of the process. For example Open BEAGLE supports the variation on GP used here that allows mixtures of data types known as *strongly typed genetic programming* or STGP (Montana, 1995). In addition Open BEAGLE's structure allows changing its population replacement strategy operator and fitness evaluation operator to implement the novel *cohort fitness* used for interactive evaluation of relative camouflage effectiveness.

In these experiments GP populations consist of 100 or 120 individual texture programs. These are run, on average, for the equivalent of 100 generations using *steady state* replacement. So roughly 10000 individuals are bred and have their fitness tested in 1000 cohorts of 10 individuals each. The population is divided into 4 or 5 *demes* (islands, isolated breeding populations, with occasional migration) of 20 or 30 individuals each. In addition to GP crossover between programs, the floating point constants in each program were subjected to incremental ("jiggle") mutation. Figure 4 shows early tests (before the interactive camouflage experiments) of evolved textures using using an *ad hoc* fitness function. This fitness function merely measures simple image properties such as a somewhat uniform brightness histogram and some color variation. These textures were created automatically with no human in the loop, then interesting results were hand selected for Figure 4.

Interactive Evaluation of Camouflage

The role of predator in these experiments is played by a human observer who visually compares the quality of evolved camouflage patterns. This happens in a simple graphical user interface. The user sees a blank window and clicks the mouse or trackpad to begin a "round" of the camouflage game. The window is redrawn to show a background texture on which is overlaid a *cohort* of circular prey objects, each with an evolved camouflage texture, see Figure 5. In these experiments a cohort contains ten individuals. Prey are placed on the background in random non-overlapping positions. They were allowed to extend partially outside the window, perhaps a poor choice.

The user's task is to inspect the scene, locate prey objects, and select the one that appears most conspicuous—that contrasts most strongly with the background. This selection is indicated with a mouse click on the prey object, signaling the act of abstract predation. In response the GUI records the selection, removes the selected prey from the cohort and redisplays, erasing the prey. Now the scene consists of the background with n-1 prey objects and the user selects the next most conspicuous. This process is repeated five times, leaving five survivors from the original cohort of ten. (Cohort size and the number "eaten" can be varied, 10 and 5 seemed to work well in these experiments.) The window returns to its blank state and awaits the next round.

In typical GA/GP application, fitness conveys fine gradations of quality. In this model, fitness is binary: life or death. Individuals selected by the predator are removed from the population. This is similar to the *selective breeding* of (Unemi, 2003). Survivors, spared by the predator, retain their high fitness and pass into the next generation (called *elitism* in evolutionary computation). For each "round" of the camouflage game, the predator looks at a cohort of ten textured prey. These are drawn randomly from a deme population which is half newly bred and half survivors from earlier generations. From this cohort of 10 new and old prey, the predator "eats" those with the least effective camouflage in the cohort. This culls out both ineffective new prey and old obsolete prey. Improvement during one run is shown in Figure 3.

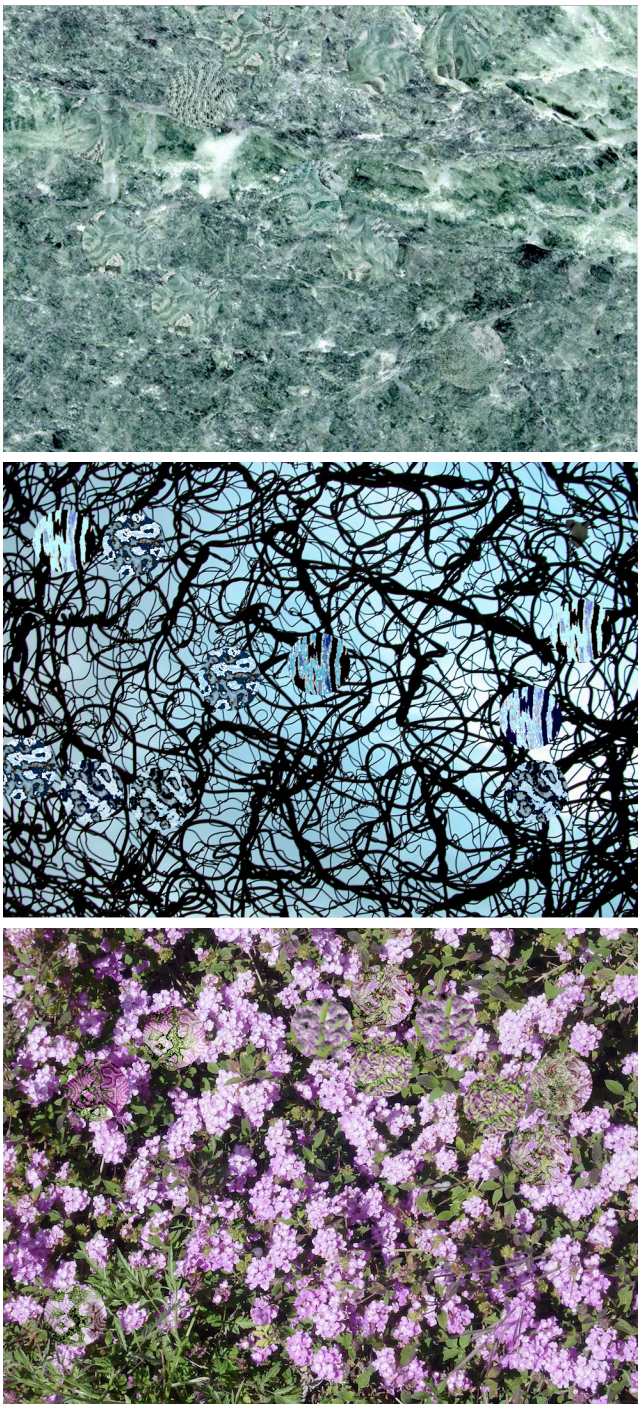


Figure 5: screen shots showing interactive sessions with "serpentine" (top) and "twisty wires" (middle) and "flowers" (bottom) environments. In all three, a new cohort of ten evolved textures is shown overlaid on the background.

The original plan was that a static image of prey over background would be presented to the user who would then click on the prey in order of conspicuousness. However it seemed the user might lose track of which prey had already been selected. Some sort of mark could be drawn to indicate which had been selected. But the presence of those already selected (more conspicuous) prey, if not the marks themselves, might interfere with finding the *n*th most conspicuous prey. Erasing prey as they are selected removes this potential distraction. It gives the user a less demanding cognitive task: scan the image and identify the most conspicuous remaining prey. This kind of *salience* detection seems to happen at a low level in the vision system and requires little or no abstract reasoning (Itti, et al. 1998).

Still this task can be ambiguous for the human observer. Given a green background and a collection of red, purple and checkerboard prey textures—as might happen in the early stages of an evolution run—it can be hard to decide which of the conspicuous prey is the worst match to the background.

Results

While not all evolutionary runs found convincing results, some produced effective camouflage. In fact some evolved camouflage was so effective that they were missed in the user's initial scan for prey. They were overlooked until a count revealed a "missing prey" and a second, more careful, visual search was made. That a jaded experimenter was actually fooled by evolved camouflage is a significant success. This happened with the "bark" background in Figure 1a. Similarly it was very hard to pick out some of the prey in the run with the "serpentine" background shown in Figure 5.

In these experiments, the evolving prey population usually moved toward matching the typical color or texture of the background. Matching on multiple characteristics was apparently harder. Sometimes a run would find the exact color but never really get the pattern right (see Figures 10(right) and 11) and vice versa (Figure 10(middle)). A few times both came together to produce a compelling result. Combinations of multiple colors seemed a much harder target for adaptation. This was especially true when features in the background were larger than the prey (as for example with "berries" (Figure 6) and "peppers" (Figures 2 and 7)). Prey size implies an upper bound on the size of features (lower bound on spatial frequencies) that can be matched. In the extreme, an environment made up of large areas of uniform appearance allows no effective camouflage for small prey.

Evolutionary computation commonly produces a mix of successful and unsuccessful runs. Some variability is inevitable using a stochastic optimization technique. When too many bad runs are seen, a typical fix is to run the evolutionary computation with a larger population. For a standard EC application this is just a matter of investing more processors or time. With an interactive fitness function there is a trade-off between bigger populations and the limits of human endurance. In these experiments, a typical run has 1000 cohorts, so requires about 5000 mouse clicks. If the user can keep up a blistering pace of one evaluation and click per second, a run costs about 1.5 hours of mind numbing work. My rate is significantly slower, plus I cannot work steadily at it for more than

Figure 6: accidental "blue" berries

Figure 7: pattern on prey similar to stem on red pepper above it.




Figure 8: progression of camouflage patterns during a run with the pebbles environment
(nice color-matched texture, followed by better frequency matching, then something like feature matching)

Figure 9 lentils
(near feature size limit)

Figure 10: early results on "leaves" (left) and "cracked wheat" (middle) both based on the *Wrapulence* texture which features edges at many scales and so helps create disruptive camouflage. The right hand texture appears to be based on the cloud-like *Brownian* texture which is not appropriate for the "berries" background but managed to match three colors of the environment: red, white and blue.

15-30 minutes at a sitting. See Future Work about addressing this problem with distributed human computation.

These experiments are based on the hypothesis that camouflage can be evolved, given only that an observer can identify the most conspicuous prey in a group. While effective camouflage patterns have been found, this idea is not clearly proved. The methodology used here presents a risk of *experimenter bias*. The same person advances the hypothesis and serves as the subject in an experiment to test it. With knowledge of how the interactive task is mapped into fitness, it is possible to "game" the task, using it for aesthetic selection as in (Sims, 1991). For example, the user might be reluctant to "eat" a prey with a particularly interesting camouflage pattern, even if it were more conspicuous than others in the cohort.

It would be inappropriate to call it an instance of "mimicry" but some interesting shapes evolved in a run using the mixed berries background (see Figure 6). While the colors are wrong and the shapes and textures are off, some of the prey looked a bit like blueberries with a frosted white surface and a suggestion of the "crown" (remains of the flower) at the end of a blueberry. Similarly in a "peppers" run a prey was found that looked a lot like the top of a red bell pepper with its green stem (see Figure 7). These chance similarities do not say much about mimicry in nature, except that one can see how easily it can arise and then be amplified and refined by even a small survival advantage.

See http://www.red3d.com/cwr/iec/ for additional results.

Future Work

These initial experiments were intended as the first steps in a more comprehensive study of camouflage evolution. Beyond refining this technique, two new research directions are planned.

Refinements on the current approach include improvements to the texture synthesis library and modified user interaction. Cohorts now contain a fixed number of camouflaged prey. It may be helpful to vary this number to remove a clue that well camouflaged prey have been overlooked. (Kashtan, et al. 2007) suggests that periodically changing evolutionary goals provides better results. For camouflage evolution, this might equate to periodically cycling between several related background images, perhaps several photographs of a similar environment.

The first new research direction is to use *distributed human computation* over the Internet to allow using larger genetic populations. This should provide stronger results and allow tackling more challenging kinds of background images. One approach is simply to pay people to perform the interactive fitness test. Utilities like Amazon Mechanical Turk (Amazon, 2005) provide infrastructure to *crowdsource* small tasks like these requiring human judgement. Another approach is to entice people to participate voluntarily by casting the task as a game—a "game with a purpose" like the Google Image Labeler (Google, 2006) and other examples at gwap.com. Several techniques have been identified to change a mundane task into a game, such as scores, time limits, leader-boards and live competition against other human players, see (von Ahn and Dabbish, 2008).

The second new research direction is to investigate *synthetic predators* to allow evolving camouflage without a human in the loop. Using techniques from machine vision and machine learning, the goal would be to train an agent to "break" camouflage. It would need to analyze an image, identify unusual *salient* regions (Itti, 1998), and classify them as being either part of the background or potential camouflaged prey. Such an agent could then be coupled with the texture synthesis and evolutionary computation components of the current work to form a closed co-optimization loop (see (Wilson, 2009) for a similar proposal). Camouflaged prey would demonstrate fitness by avoiding detection while predator vision agents would demonstrate fitness by detecting camouflage prey. Such a system would provide a useful computation model of the coevolution of camouflage.

Acknowledgments

This research was made possible by the generous support of my employer, Sony Computer Entertainment and particularly my manager: Dominic Mallinson, Vice President, US R&D. I

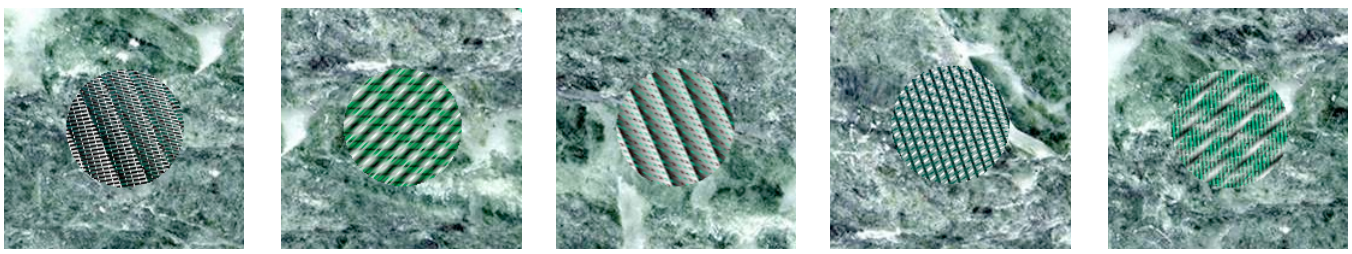
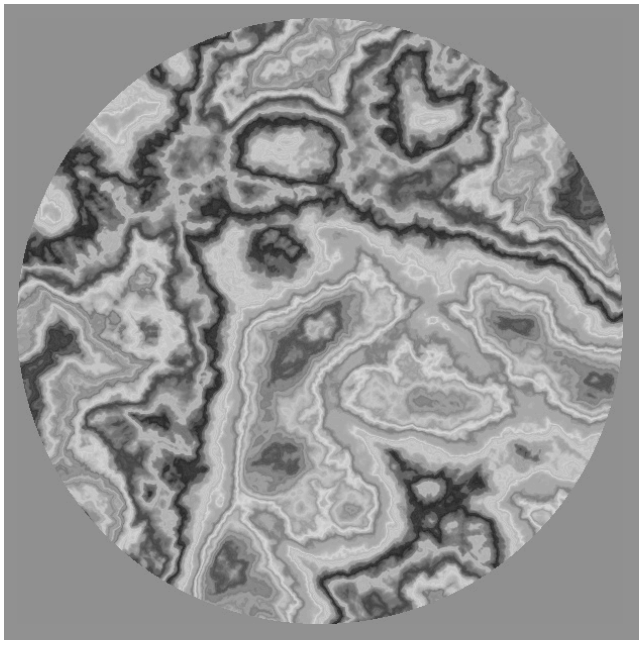


Figure 11: an unsuccessful early run using the "serpentine" background and a rank-based fitness scheme that was later abandoned



```
Colorize (Ring (5.80532,
                Vec2 (-2.12073, 0.411024),
                Stretch (0.0449509,
                         -1.06448,
                         Vec2 (-1.37922, 0.946741),
                         Furbulence (1.21806,
                                     Vec2 (1.62529, 2.9815)))),
          Furbulence (1.21806,
                      Vec2 (-2.94693, -1.86416)))
```

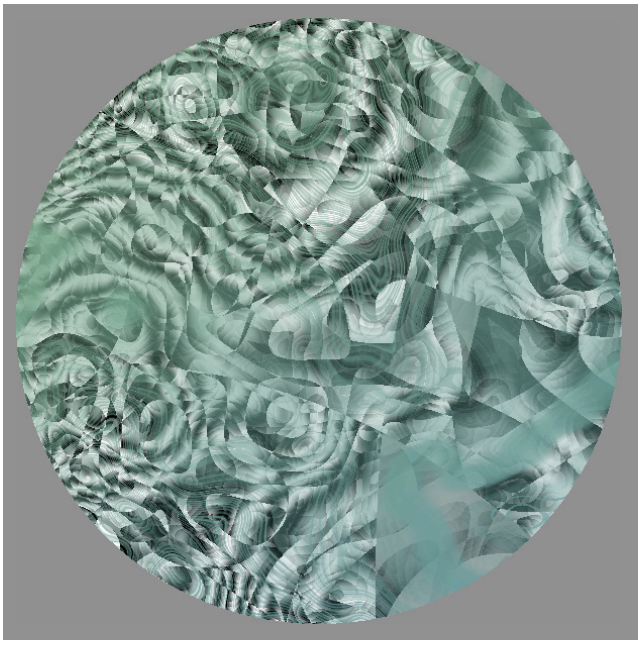
Figure 12: disruptive oak bark camouflage of Fig. 1(a), re-rendered at 600x600 resolution, with its evolved source code

owe special thanks to my remote friend and collaborator Bjoern Knafla for writing the Cocoa application that served as GUI for these experiments. Christian Gagné helped extensively with the interface to Open BEAGLE. Thanks to readers of early versions of this paper and my research proposal: Daniel Weinreb, Iztok Lebar Bajec and again Bjoern Knafla. Others have contributed helpful suggestions and discussions: James O'Brien, Lance Williams, Ken Perlin, Michael Wahrman, Andy Kopra and Karen Liu. Thanks to many coworkers at SCE US R&D, my patient and supportive wife Lisa, my son Eric, and my daughter Dana who frequently contributed feedback and suggestions about this research.

Appendix 1: Texture Synthesis Details

One input to *Strongly Typed Genetic Programming* (Montana, 1995) is a description of a set of functions and the types associated with their inputs and outputs. The texture synthesis library used in this work included types for procedural textures, 2d Cartesian vectors, RGB colors and numbers. There are five numeric types, all floating point, with unique ranges (and so whether negative or zero values are included). Random constants (GP calls them *ephemeral constants*) are generated according to these types.

The texture synthesis library contained 52 texture producing elements. Some of the names are self-descriptive, for others, and for description of parameter types for each, see (Reynolds, 2009). **Texture generators:** UniformColor, SoftEdgeSpot, Gradation, SineGrating, TriangleWaveGrating, SoftEdgedSquareWaveGrating, RadialGrad, Noise, ColorNoise, Brownian, Turbulence, Furbulence, Wrapulence and NoiseDiffClip. **Texture operators:** Scale, Translate, Rotate, Mirror, Add, Subtract, Multiply, Max, Min, SoftMatte, ExpAbsDiff, Row, Array, Invert, Tint, Stretch, StretchSpot, Wrap, Ring, Twist, VortexSpot, Blur, EdgeDetect, EdgeEnhance, SliceGrating, SliceToRadial, SliceShear, Colorize, Gamma, AdjustSaturation, AdjustHue, BrightnessToHue, BrightnessWrap, BrightnessSlice4, HueIfAny, SoftThreshold, SpotsInCircle and ColoredSpotsInCircle.



```
Invert (SoftMatte (HueIfAny (Colorize (Twist (-1.76008, Vec2
(-2.90822, -1.26208), Multiply (Brownian (0.880861, Vec2
(2.80615, 1.14405)), Wrap (6.21909, 5.55726, Vec2 (1.88101,
-1.10475), Add (VortexSpot (-2.95874, 4.37424, Vec2 (-2.24113,
-0.804409), Row (Vec2 (-1.20827, -0.80333), Wrapulence (5.81646,
Vec2 (1.46969, 0.464754)))), Multiply (TriangleWaveGrating
(15.0552, 0.251605, 4.92253), Wrap (6.21909, 5.25948, Vec2
(-2.90822, -1.26208), Add (ColoredSpotsInCircle (146.485,
0.573184, 0.103147, Stretch (1.92016, 0.932767, Vec2 (0.994563,
1.8778), SineGrating (17.4233, 0.477075)), Translate (Vec2
(1.3634, -3.05406), Colorize (SineGrating (87.1581, 1.2438),
SoftEdgedSquareWaveGrating (138.03, 0.0101831, 0.894823,
1.03307))), SliceToRadial (Vec2 (-1.20827, -0.80333), ColorNoise
(1.09284, Vec2 (1.24907, -3.11514)))), Brownian (4.15562, Vec2
(-1.20827, -0.80333))))))))), Brownian (0.880861, Vec2 (2.80615,
1.14405)))), SliceToRadial (Vec2 (-1.20827, -0.80333), ColorNoise
(1.09284, Vec2 (1.24907, -3.11514))), Colorize (Twist (-1.90423,
Vec2 (0.977825, -0.533419), Twist (-1.90423, Vec2 (0.977825,
-0.533419), RadialGrad (195.316, Vec2 (1.24907, -3.11514)))),
Wrapulence (5.81646, Vec2 (0.0918581, -0.543768)))))
```

Figure 13: camouflaged prey evolved on “serpentine” background with its evolved source code

References

Abbott, K. (2010). Background evolution in camouflage systems: A predator–prey/pollinator-flower game. *Journal of Theoretical Biology.* 262(4):662-678.

Amazon Mechanical Turk. Launched 2005, accessed 2010: http://www.mturk.com/

Beddard, F. E. (1895). *Animal Coloration. An Account of the Principal Facts and Theories Relating to the Colours and Marking of Animals.* Swan Sonnenschein & Co., London.

Bond, A. B. and Kamil, A. C. (2002). Visual predators select for crypticity and polymorphism in virtual prey. *Nature.* 415(6872): 609-613.

Chu, H-K, Hsu, W-H, Mitra, N. J., Cohen-Or, D., Wong, T-T, Lee, T-Y. (2010). Camouflage Images. To appear in *ACM Transactions on Graphics* 29(3).

Cott, H. B. (1940). *Adaptive Coloration in Animals.* Methuen and Co., London.

Cuthill, I. C., Stevens, M., Sheppard, J., Maddocks, T., Parraga, and C. A., Troscianko, T. S. (2005). Disruptive coloration and background pattern matching. *Nature.* 434(7029):72-74.

Cuthill, I. C., Hiby, E., and Lloyd, E. (2006). The predation costs of symmetrical cryptic coloration. *Proc. Biol. Sci.* 273(1591):1267-1271.

Darwin, C. (1859). *On the Origin of Species by Means of Natural Selection*. John Murray, London.

Dawkins, R. (1986). *The Blind Watchmaker.* W. W. Norton & Company, Inc., New York, NY.

Ebert, D. S., Musgrave, F. K, Peachey, D., Perlin, K. and Worley, S. 1994. *Texturing and Modeling: A Procedural Approach.* AP Professional. ISBN 0-12-228760-6.

Eizirik, E., David, V., Buckley-Beason, V., Roelke, M., Schaffer, A., Hannah, S., Narfstrom, K., O'Brien, S., Menotti-Raymond, M. (2010). Defining and Mapping Mammalian Coat Pattern Genes: Multiple Genomic Regions Implicated in Domestic Cat Stripes and Spots. *Genetics* 184(1):267-275.

Funes, P., Sklar, E., Juillé, H. and Pollack, J. (1998). Animal-animat coevolution: using the animal population as fitness function. In *From animals to animats 5*, 525-533. MIT Press, Cambridge, MA.

Gagné, C. and Parizeau, M. (2006). Genericity in Evolutionary Computation Software Tools: Principles and Case-Study. *International Journal on Artificial Intelligence Tools*, 15(2):173-194.

Google Image Labeler. Launched 2006, accessed 2010: http://images.google.com/imagelabeler/

Holland, J. H. (1975). *Adaptation in natural and artificial systems.* University of Michigan Press, Ann Arbor.

Itti, L., Koch, C., and Niebur, E. (1998). A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* 20(11):1254-1259.

Kashtan, N., Noor, E., and Alon, U. (2007). Varying environments can speed up evolution. *Proceedings of the National Academy of Sciences.* 104(34):13711-13716.

Koza, J.R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press, Cambridge, MA. ISBN 0-262-11170-5

Merilaita, S. (2003). Visual background complexity facilitates the evolution of camouflage. *Evolution.* 57(6):1248-1254.

Montana, D. J. (1995). Strongly typed genetic programming. *Evolutionary Computation* 3(2):199-230.

Open BEAGLE. (2002). Open BEAGLE evolutionary computation library, version 3.0.3. Website for code and documentation, launched 2002, accessed 2010: http://beagle.gel.ulaval.ca/

Perlin, K. (1985). An image synthesizer. *SIGGRAPH Comput. Graph.* 19(3):287-296.DOI=http://doi.acm.org/ 10.1145/325165.325247

Reynolds, C. (2009) Texture Synthesis Diary (blog/lab notebook), accessed 2010: http://www.red3d.com/cwr/texsyn/diary.html

Schaefer, H. M. and Stobbe, N. (2006). Disruptive coloration provides camouflage independent of background matching. *Proc. Biol. Sci.* 273(1600):2427-2432.

Sims, K. (1991). Artificial evolution for computer graphics. In *Proceedings of the 18th Annual Conference on Computer Graphics and interactive Techniques SIGGRAPH '91*. ACM, New York, 319-328. DOI= http://doi.acm.org/10.1145/122718.122752

Sherratt, T. N., Pollitt, D., and Wilkinson, D. M. (2007). The evolution of crypsis in replicating populations of web-based prey. *Oikos*. 116(3):449-460.

Stevens, M. and Cuthill, I. C. (2006). Disruptive coloration, crypsis and edge detection in early visual processing. *Proc. R. Soc. B.* 273(1598):2141-2147.

Stevens, M. and Merilaita, S. (2009). Animal camouflage: current issues and new perspectives. *Phil. Trans. R. Soc. B.* 364(1516): 423-427.

Takagi, H. (2001). Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation. *Proceedings of the IEEE*. 89(9) 1275-1296.

Togelius, J., Yannakakis, G., Stanley, K., and Browne, C. (2010). Search-based Procedural Content Generation. *Proceedings of 2nd European event on Bio-inspired Algorithms in Games (EvoGAMES 2010)*.

Thayer, G. H. (1909). *Concealing-coloration in the animal kingdom: an exposition of the laws of disguise through color and pattern: being a summary of Abbott H. Thayer's discoveries*. Macmillan, New York, NY.

Unemi, T. (2003). Simulated breeding – a framework of breeding artifacts on the computer. *Kybernetes*. 32(1/2) 203-220.

von Ahn, L. and Dabbish, L. (2008). Designing games with a purpose. *Commun. ACM* 51(8):58-67. DOI= http://doi.acm.org/10.1145/1378704.1378719

Wei, L-Y, Lefebvre, S., Kwatra, V., and Turk, G. (2009) State of the Art in Example-based Texture Synthesis, in *Eurographics '09 State of the Art Reports (STARs)*.

Wilson, S.W. (2009). Coevolution of Pattern Generators and Recognizers. Illinois Genetic Algorithms Laboratory TR 2009006.

CC Background Image Sources

Bark by Six Revisions:
http://www.flickr.com/photos/31288116@N02/3752674533/

Cracked wheat by Sanjay Acharya:
http://commons.wikimedia.org/wiki/File:Sa-cracked-wheat.jpg

Flowers (*lantana montevidensis* in our backyard) by Craig Reynolds
http://www.red3d.com/cwr/iec/

Granite Yosemite by David Monniaux:
http://commons.wikimedia.org/wiki/File:Granite_Yosemite_P1160483.jpg

Leaves by Scott M. Liddell (www.scottliddell.net) with permission:
http://www.morguefile.com/archive/display/90656

Lentils by Daniel Kulinski (Daniel*1977)
http://www.flickr.com/photos/didmyself/2126646787/

Mixed berries by Angelo Juan Ramos:
http://commons.wikimedia.org/wiki/File:Summer_Fruits.jpg

Pebbles by Sean Hattersley:
http://commons.wikimedia.org/wiki/File:Pebbleswithquarzite.jpg

Peppers by Elavats:
http://www.flickr.com/photos/elavats/549041490/

Serpentine by Kevin Walsh:
http://commons.wikimedia.org/wiki/File:Serpentine-texture.jpg

Twisty wires by Clara Natoli (used with permission)
http://www.morguefile.com/archive/display/10850

Evolving Homeostatic Tissue Using Genetic Algorithms

Philip Gerlee[1], David Basanta[2], and Alexander R.A. Anderson[2]

[1]Center for Models of Life, Niels Bohr Institute, Blegdamsvej 17, 2100 København N, Denmark
[2]H. Lee Moffitt Cancer Center and Research Institute Integrated Mathematical Oncology
12902 Magnolia Drive, Tampa, FL 33612, USA
Corresponding author: gerlee@nbi.dk

Extended Abstract

Homeostasis is a critical property of living beings that involves the ability to self-regulate in response to changes in the environment in order to maintain a certain dynamic balance affecting form and/or function. The importance of homeostasis is pronounced in multi-cellular organisms where function and structure needs to be regulated at ever increasing levels of organisation (Cunliffe, 1997).

In this talk we will address the evolution of homeostasis in a computational framework and investigate structural homeostasis in the simplest of cases, a tissue formed by a mono-layer of cells. To this end, we made use of a 3-d hybrid cellular automaton, an individual-based model in which the behaviour of each cell depends on its local environment (Gerlee and Anderson, 2009). This was implemented by using a response network, which for each cell takes extra-cellular cues as input, and whose output determines the phenotype or behaviour of the cell (cell division, movement, death).

Instead of dictating a given mapping from environment to phenotype, we made use of an evolutionary algorithm (EA) to evolve cell behaviour which gives rise to a homeostatic tissue (Streichert et al., 2003; Stanley and Miikkulainen, 2003; Andersen et al., 2009). The fitness of a genotype (response network) was evaluated by running the cellular automaton seeded with a single cell for given number of time steps. Cell types which can fill the domain with a mono-layer of cells are given the highest fitness, while those which either over-grow or fail to fill the domain are punished. We made use of two different fitness functions, one which uses a constant fitness evaluation where each cell type is tested for 200 time steps (constant), and another which increases the evaluation time for each successive generation (incremental). An example of run with a constant fitness evaluation scheme is shown in fig. 1.

Analysis of the solutions provided by the EA shows that the two evaluation methods gives rise to different types of solutions to the problem of homeostasis. The constant method leads to almost optimal solutions, which rely on a very high rate of cell turn-over, and this is achieved by fine-tuned balance between cell birth and death. The solutions from the incremental scheme on the other hand behave in a more conservative manner, only dividing when necessary, and generally have a lower fitness.

In order to test the robustness of the solution we subjected them to environmental stress, by wounding the tissue, and to genetic stress, by introducing mutations. The cell types with high turn-over were more robust with respect to wounding, healing faster and more accurately. The sensitivity to genetic perturbations depends on what type of mutations we consider. Copy mutations, which only occur when the cells divide, affect the tissues with a high turn-over, while cosmic ray mutations, which occur at a constant rate, are more detrimental to the conservative cell types.

The two evolved cell types analysed present contrasting mechanisms by which tissue homeostasis can be maintained. This compares well to different tissue types found in multi-cellular organisms. For example the epithelial cells lining the colon in humans are shed at a considerable rate (Podolsky and Babyatsky, 2003), while in other tissue types, which are not as exposed, the conservative type of homeostatic mechanism is normally found (Hooper, 1956).

These results will hopefully shed light on how multi-cellular organisms have evolved and what might occur when homeostasis fails, as for example in the case of cancer (Preston-Martin et al., 1990).

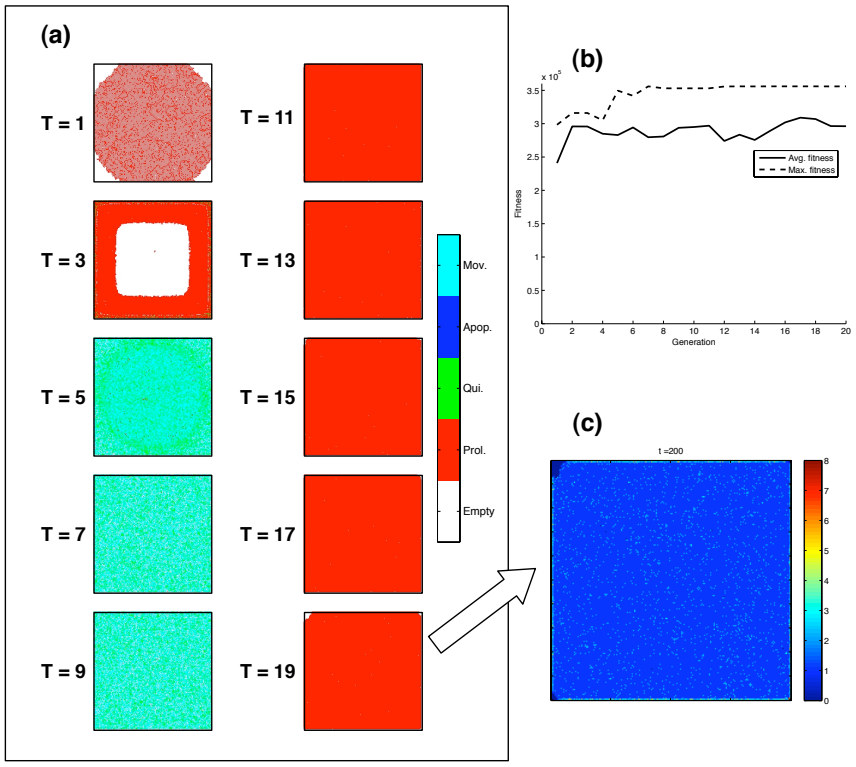


Figure 1: Time evolution of the EA. (a) shows the most fit genotypes at different generations in the run, where the process converges on a genotype which predominately proliferates. The time evolution of the average and maximum fitness is shown in (b), which, because of a weighted multi-objective fitness function, does not necessarily increase over time (Bentley and Wakefield, 1998). The cell density of the final genotype (T = 19) is shown in (c), and reveals that the solution arrived upon by the EA forms a mono-layer, and thus satisfies our criteria for a homeostatic genotype.

References

Andersen, T., Newman, R., and Otter, T. (2009). Shape homeostasis in virtual embryos. *Artificial Life*, 15(2):1–23.

Bentley, P. and Wakefield, J. (1998). *Soft Computing in Engineering Design and Manufacturing*, chapter Finding acceptable solutions in the pareto-optimal range using multiobjective genetic algorithms, pages 231–240. Springer Verlag.

Cunliffe, J. (1997). Morphostasis: an evolving perspective. *Med. Hypotheses*, 49(6):449–459.

Gerlee, P. and Anderson, A. R. A. (2009). Modelling evolutionary cell behaviour using neural networks: application to tumour growth. *Biosystems*, 95:166–174.

Hooper, C. E. S. (1956). Cell turnover in epithelial populations. *J Histochem Cytochem*, 4(6):531–540.

Podolsky, D. and Babyatsky, M. (2003). *Textbook of Gastroenterology*, chapter Growth and development of the gastrointestinal tract, pages 546–577. Lippincott, Philadelphia, PA.

Preston-Martin, S., Pike, M. C., Ross, R. K., Jones, P. A., and Henderson, B. E. (1990). Increased cell division as a cause of human cancer. *Cancer Res*, 50(23):7415–7421.

Stanley, K. and Miikkulainen, R. (2003). A taxonomy for artificial embryogeny. *Artificial Life*, 9(2):93–130.

Streichert, F., Spieth, C., Ulmer, H., and Zell, A. (2003). Evolving the ability of limited growth and self-repair for artificial embryos. In *Lect Notes Artif Int*, volume 2801, pages 289–298.

Identifying the Location of a Target Object in the Weakly Electric Fish through Spatiotemporal Filtering Process

Miyoung Sim and DaeEun Kim

Biological Cybernetics Lab
School of Electrical and Electronic Engineering,
Younsei University, Schinchon, Seoul, 120-749, Corea (South Korea)
{simmi, daeeun}@yonsei.ac.kr

Abstract

The weakly electric fish use their electric organ discharge (EOD) and electroreceptors to identify their prey, explore in their surrounding environment, and communicate with their members in the same species. They are specialized in active electrolocation. They can detect the distortion of the self-generated electric field, which is caused by a target object. There are two types of electric signals, wave-type and pulse-type, that the weakly electric fish can generate. In this paper, we suggest that periodic EOD signals are helpful to extract localization features from noisy electrosensory signals. The cross-correlation between an efference copy signal and the sensory afferent signals in the waveform can produce accurate relative slope in noisy environment. This process has two-phase filtering. The noise-filtering with cross-correlation with respect to the temporal axis and additional filtering with respect to rostrocaudal spatial axis can effectively remove noise, and thus this process provides accurate information of the distance of a target object.

Introduction

Weakly electric fishes localize a target object by their electrolocation system. They are known as only creatures that use active electrolocation with their self-generated electric field (Lissmann, 1958). Electric organ (EO) consists of a modified nerve and muscle cells, and is generally located in caudal area (Kramer, 1999). The EO composed of electrocytes produces an EOD. EODs have waveform characteristics. There are two types of waveforms, pulse-type and wave-type. A lot of Gymnotiformes and all of Mormyriformes (except *Gymnarchus niloticus*) generate a pulse-type of EOD. The pulse-type waveform has short pulses with large intervals between pulses. It is believed that electric fish use a waveform of EOD to recognize another electric fish (Bastian, 1994). In this paper, we focus on the electrolocation of weakly electric fish and an advantage of periodic characteristics of EOD waveform in noisy environment.

There are two types of electroreceptors, tuberous and ampullary electroreceptors (Nelson et al., 2000; von der Emde and Fetz, 2007). These electroreceptors respond to electric stimuli. Usually, ampullary electroreceptors are found in elasmobranch, such as sharks and rays, and they lack in active electric organ. Elasmobranch do not generate the electric field, but just detect the bio-electric signals generated by another creatures. All living animals produce bio-electric signals generated by activation of their muscle and nerve cells. Weakly electric fish have another type of electroreceptors. They detect the change of their own electric signal by tuberous electroreceptors through active sensing (Nelson et al., 2000). About $14,000$ tuberous electroreceptors are distributed all over the body surface of *Apteronotus albifrons*, a species of weakly electric fish. Sensor readings of electroreceptors can provide information to localize their prey, navigate in space, and communicate with conspecifics.

The localization of a target object is very important to capture a prey, avoid their predators, or navigate in the environment. Weakly electric fish produces the electric field and senses the distortion of electric field with many electroreceptors on the whole skin surface. These sensor readings are considered as 'a stimulus image' observed at the set of electroreceptors and it is called 'electric image' (Caputi and Budelli, 2006; von der Emde, 2006). The intensity value of sensor readings are inversely proportional to the distance between a target object and the sensor location on the surface.

When a target object is located near the weakly electric fish, sensor readings of electroreceptors draws a bell-shaped curve. The rostrocaudal (from head to tail) position of a target object can be easily measured with maximal amplitude of an electric image(Rasnow, 1996; Chen et al., 2005). When the target object becomes far away from the electric fish, the maximal value of sensor readings decreases. The maximum amplitude of the electric image is also affected by the size and conductivity of the target object. To measure the lateral distance of a target object from the midline axis of weakly electric fish, the relative slope and full-width at half-maximum (FWHM) have been suggested as a distance measure (Schwarz and von der Emde, 2001; Chen et al., 2005).

If we have a clean electric image without noise, it is not difficult to get the lateral distance by the relative slope or FWHM. The relative slope is the ratio of the maximal slope to the maximal amplitude of sensor readings in the rostrocaudal axis (Schwarz and von der Emde, 2001). The FWHM

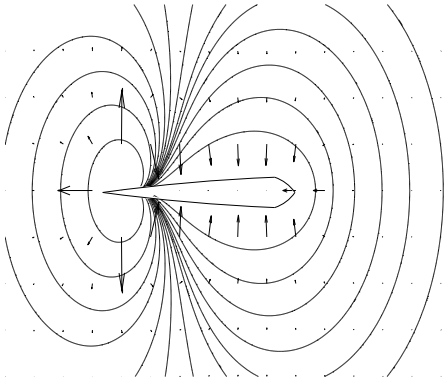


Figure 1: Electric field generated by the EO of weakly electric fish (solid contour lines indicates equipotential lines)

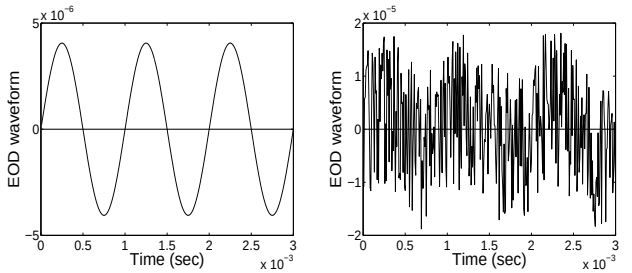
is the width of the bell-shaped curve at half of the maximal amplitude (Chen et al., 2005). The change of electric signal is affected by the size and lateral distance of a target object. The width becomes larger when the size of a target object increases. Thus the ratio between maximum amplitude and width, or the ratio between maximum amplitude and slope can be a cue for the lateral distance without considering another properties of the target object, for example, size and conductivity. However, when electric potentials at the electroreceptors include noisy signals, the preprocessing step is needed to extract noise-free signals. We suggest a method using a waveform of EOD to extract the denoised electric image and measure the lateral distance of a target object.

In the previous researches, it has been pointed out that electric properties of a target object can be measured by the distortion of EOD waveform (von der Emde, 1998). Yet, how to handle noisy signals for the relative slope information has not been studied so far. In this paper, we observe a waveform of EOD to measure the lateral distance, and then the filtering process with respect to time axis as well as spatial axis is applied to obtain noise-free signals. Ultimately we can estimate the distance of a target object very accurately. Here, we use the cross-correlation between an efference copy signal and the sensory afferent signals to obtain the filtered output in the temporal axis and then apply a low pass filter to the output of electroreceptors along the rostrocaudal axis.

Localization of a target object

Fig. 1 shows electric field generated by the EO of weakly electric fish. We use an electric field model of *A. albifrons* which belongs to Gymnotiformes species established by Rasnow (1996) and Chen et al. (2005). The electric field is radically spread to every direction of the body of weakly electric fish.

Gymnotiform fishes generate continuous periodic waveform which has symmetric maximum and minimum point



(a) (b)

Figure 2: EOD waveform (a) original self-generated waveform (b) noisy waveform)

with respect to the zero point (Fugere and Krahe, 2010). Fig. 2 shows the simulated EOD waveform that has frequency $1kHz$. It is known that *A. albifrons* generates such EOD waveforms which have about $1kHz$ frequency (Nelson and MacIver, 1999).

Electric field modeling

The EO is modeled as a collection of electric poles (Rasnow, 1996; Chen et al., 2005). Then the electric potential can be calculated as a total sum of potential from each electric pole. When there are n electric poles, $n-1$ positive poles and one negative electric pole, arranged along the midline of the weakly electric fish, the electric potential, $V(\vec{x})$, derived as

$$V(\vec{x}) = \sum_{i=1}^{n-1} \frac{q/(n-1)}{|\vec{x} - \vec{x}_p^i|} - \frac{q}{|\vec{x} - \vec{x}_p^n|} \tag{1}$$

where $\vec{x}$ is the position of measured position, $\vec{x}_p^i$ the position of i-th electric pole, $\vec{x}_p^n$ last n-th negative pole. The value of q means the normalized potential magnitude which ranges from $8mV$ to $20mV$ (Chen et al., 2005). The total sum of potential magnitude of the whole electric poles including the negative pole should be zero. Thus, the magnitude of a positive pole is q/m and a negative pole $-q$. The electric field $E(\vec{x})$ at the position of $\vec{x}$ is derived as the gradient of the electric potential as

$$E(\vec{x}) = \sum_{i=1}^{n-1} \frac{q/(n-1)}{|\vec{x} - \vec{x}_p^i|^3}(\vec{x} - \vec{x}_p^i) - \frac{q}{|\vec{x} - \vec{x}_p^n|^3}(\vec{x} - \vec{x}_p^n) \tag{2}$$

To consider the component of the incident electric field vertical to the surface of a weakly electric fish, the transdermal potential difference, $V_{td}(\vec{x})$, is calculated as

$$V_{td}(\vec{x}_s) = E(\vec{x}_s) \cdot \hat{n}(\vec{x}_s) \frac{\rho_{skin}}{\rho_{water}} \tag{3}$$

where $\hat{n}(\vec{x}_s)$ is the normal vector at the electroreceptor on the skin, and ρ_{skin} and ρ_{water} resistivity of skin surface and water, respectively.

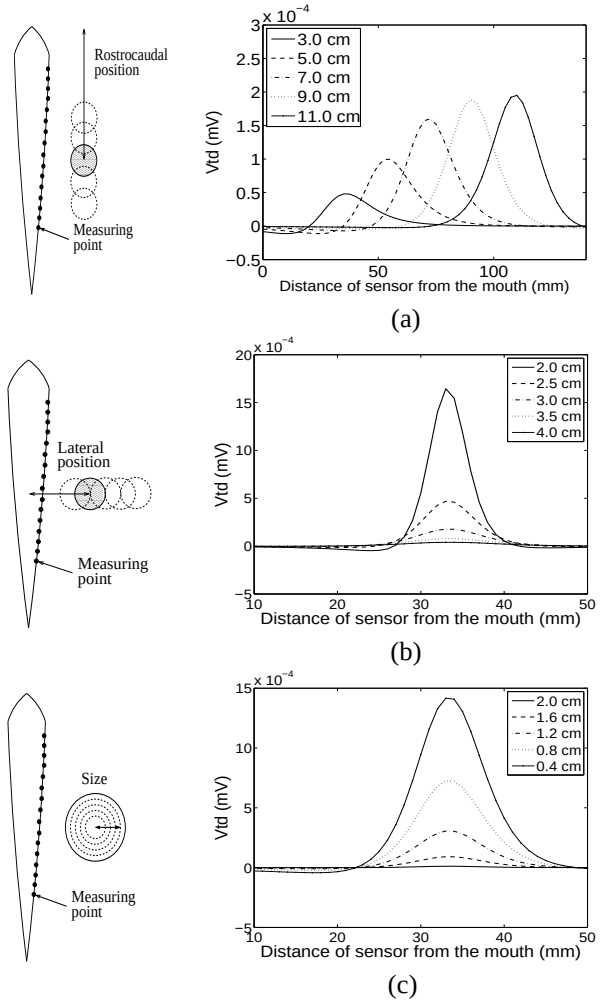


Figure 3: Electric image distorted by a neighboring target object along the rostrocaudal line on the surface of weakly electric fish with varying (a) the rostrocaudal position (b) the lateral distance (c) the size of a target object (modified from (Sim and Kim, 2010))

Rasnow (1996) and Chen et al. (2005) show the effect of a simple spherical object as a targt object. The distortion of electric field caused by a neighbor target object, $\Delta V(\vec{x})$, is calculated as

$$\Delta V(\vec{x}) = \chi \frac{a^3 E(\vec{x}_{obj}) \cdot (\vec{x} - \vec{x}_{obj})}{|\vec{x} - \vec{x}_{obj}|^3} \quad (4)$$

where a is the radius and $\vec{x}_{obj}$ the center of a spherical target object. The transdermal potential difference of an object perturbation $\Delta V_{td}(\vec{x}_s)$ is given by

$$\Delta V_{td}(\vec{x}_s) = -\nabla(\Delta V(\vec{x})) \cdot \hat{n}(\vec{x}) \frac{\rho_{skin}}{\rho_{water}} \quad (5)$$

Electric image

The change of transdermal potential value (equation (5)) due to a target object along the rostrocaudal axis draws a bell shaped curve (see Fig. 3) when the position and size of the object change. It forms one-dimensional electric image. Fig. 3 (a) shows the variation of electric images when the rostrocaudal position of the target object changes. The maximal amplitude of the electric image is found at the rostrocaudal position of the target object. The level of intensity depends on the interaction with positive and negative poles. If the object is closer to the tail, the stronger intensity can be observed for the same lateral distance. In Fig. 3 (b) and (c), the rostrocaudal position of the target object is fixed, and thus the location of the maximum amplitude has no shift, but only changes of maximal amplitudes are observed at a fixed rostrocaudal position. The intensity is affected by not only the lateral distance but also the size of the target object. Therefore, the intensity is not a direct cue for the distance.

In a three-dimensional space, we can consider rostrocaudal, lateral, and dorsoventral axis (from dorsal to ventral side) with respect to the fish body. The rostrocaudal and dorsoventral position of a target object can be determined directly from the location of the maximum intensity. The maximal amplitude can be observed at the point close to the target object. In contrast, the lateral distance can be estimated by the ratio between the maximal value, slope, and width of the electric image.

We use the relative slope to measure the lateral distance of a target object. To extract proper features from noisy signals, we need to consider filtering process. Here, we suggest spatiotemporal filtering process over noisy electric signals.

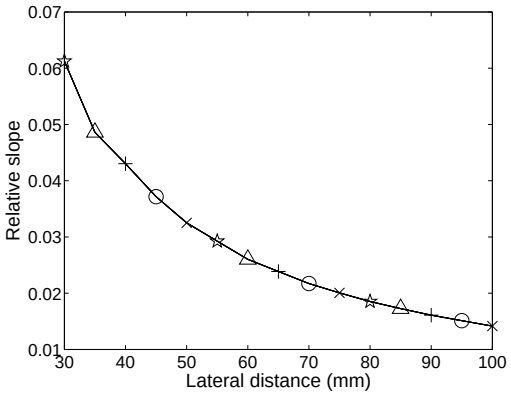


Figure 4: Relative slope when the lateral distance of the target object changes with varying object sizes (each marker represents a radius of 0.4, 0.8, 1.2, 1.6, $2.0cm$) (modified from (Sim and Kim, 2010))

Relative slope

The relative slope is the ratio of the maximal slope to the maximal amplitude of the object perturbation curve (electric image) and it is not affected by size and conductivity of the target object. Fig. 4 shows the change of relative slope when the target object moves away along the lateral axis with varying object sizes. The relative slope is not affected by the conductivity, either.

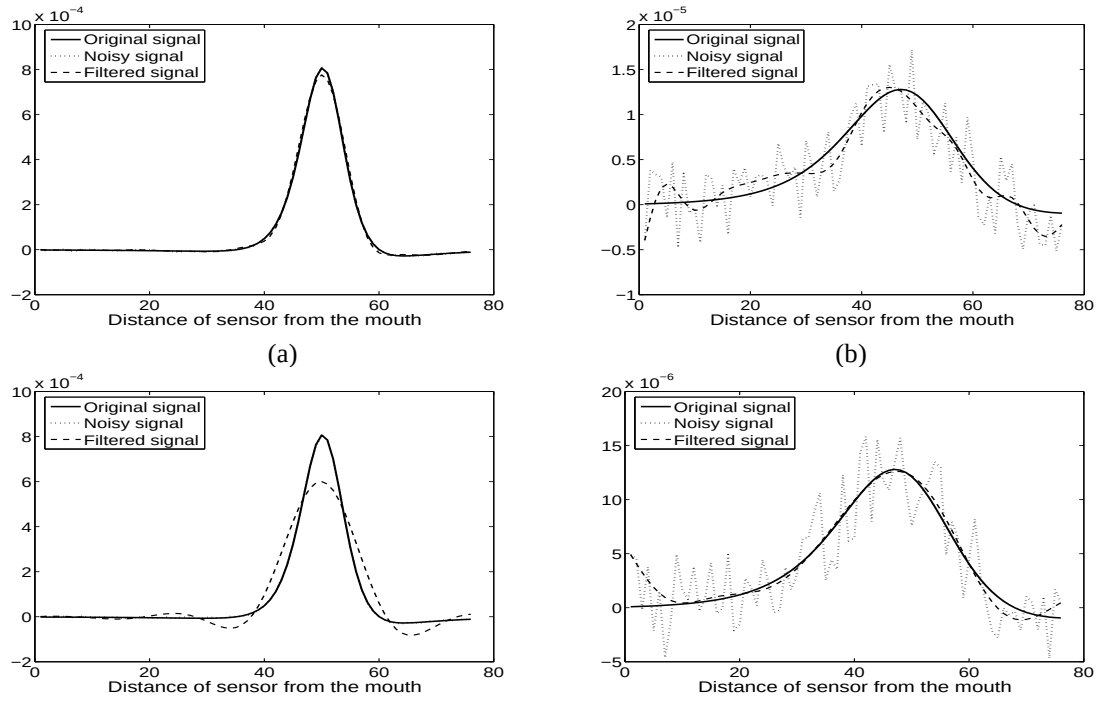


Figure 5: Electric image when noise is distributed uniformly from -5×10^{-6} to 5×10^{-6}; (a) and (c) lateral distance of a target object is $2cm$; (b) and (d) $4.8cm$ (solid : electric image without noise, dotted : distorted electric image, dashed : filtered image with cut-off frequency (a) and (b) 20% (c) and (d) 10% of the spatial sampling rate)

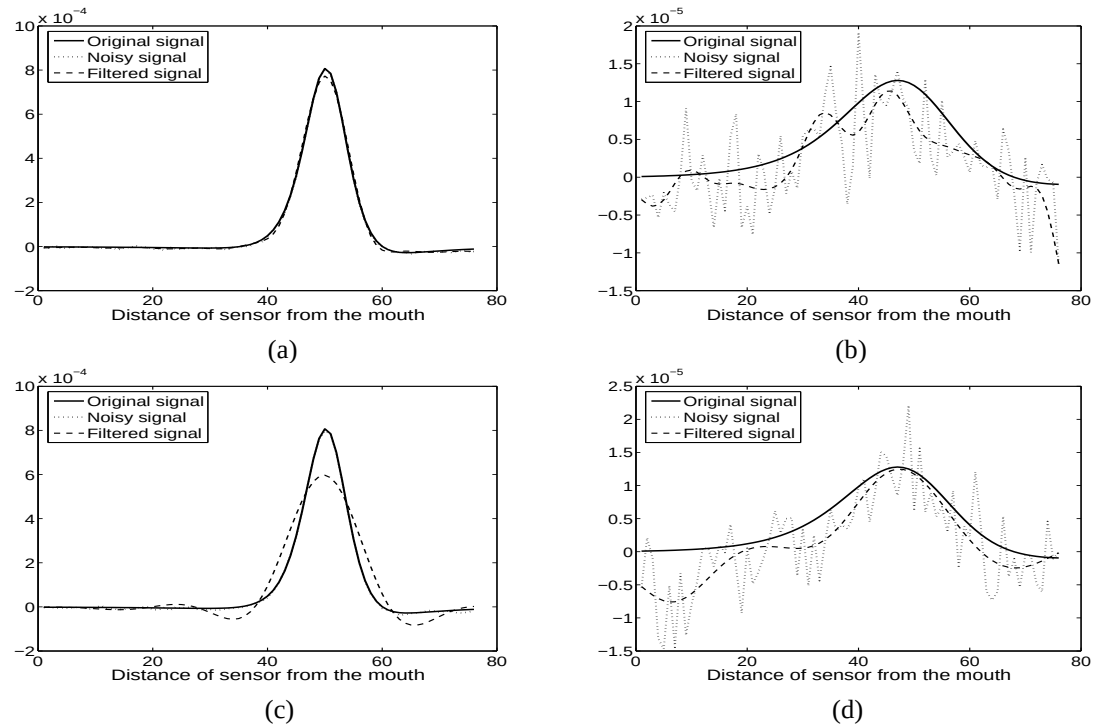


Figure 6: Denoised electric image using low pass filter when there exist Gaussian noise with variance 5×10^{-6}; (a) and (c) lateral distance of a target object is $2cm$; (b) and (d) $4.8cm$ (solid : electric image without noise, dotted : distorted electric image, dashed : filtered image with cut-off frequency (a) and (b) 20% (c) and (d) 10% of the spatial sampling rate)

We use relative slope to measure the lateral distance. However, in the natural environment, noisy signals are inevitably observed in electric images. Pure electric signals of object perturbation are mixed up with noise. It is difficult to estimate the relative slope accurately with the two noisy parameters, amplitude and slope in the electric image. Thus, we suggest a possible noise-filtering analysis over the spatiotemporal sensor readings. To smooth these distorted electric signals, we take two phase of filtering process, cross-correlation with self-generated EOD waveform and low pass filter over a collection of sensor readings along the rostrocaudal axis.

Method1 : Low pass filtering

We use a fifth order butterworth filter as a low pass filter. Generally, the noise has high frequency characteristics. Fig. 5 shows the result of that filter application. The cut-off frequency determines the frequency range of filtered electric signal. The sensor readings of electroreceptors are spatially distributed along the rostrocaudal axis. The filter is applied to the spatial distribution of the electric signals which is the result of object perturbation.

Fig. 5 shows the noisy electric image and the filtered image when the lateral distance of a target object is $2.0cm$ in Fig. 5 (a) and (c), and $4.8cm$ in Fig. 5 (b) and (d). Here, we assume random noise. The range of uniform random noise is 10×10^{-6} and it is about 8% noise level of the maximal amplitude observed when the lateral distance of the target object is $3cm$. The cut-off frequency is set to 20% and 10% of the spatial sampling rate, respectively. When a target object moves away from the weakly electric fish, the intensity decreases radically. With the filtering process, the original electric signal can be hardly restored. In Fig. 5 (b) and (d), the low pass filtering is applied with different cut-off frequencies. The smaller cut-off frequency is more effective to smooth the noisy electric signal, but the filtered signal is a little deviated from the original signal purely depending on the lateral distance.

Fig. 6 shows the noisy and denoised electric images when the noise is modeled as Gaussian noise with variance 5×10^{-6} and zero mean. In Fig. 6, the noise level is about 8% when the lateral distance of the target object is $3cm$. The distortion of electric image is similar to that with uniform random noise. In this case, the cut-off frequency 20% of the spatial sampling rate is appropriate to obtain the desired filter output.

Method2 : Cross-correlation

The self-generated EOD waveform at the tail produces the sensory afferent signals at each electroreceptor. If there is any object near the fish body, the distorted afferent signals can be measured. Reafference cancellation process can be expected in the sensory-motor loop. Here we consider another aspect of motor signal feedback.

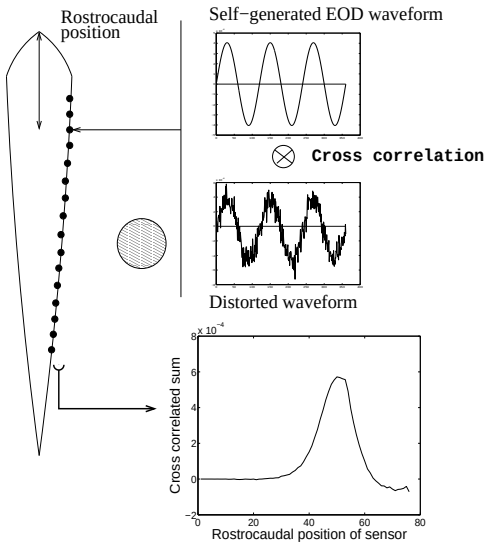


Figure 7: Process of denoising electric image using cross-correlation

The cross-correlation between an efferency copy signal and the sensory afferent signals in the waveform can lead to an interesting feature of noise removal. The cross-correlation equation is given below :

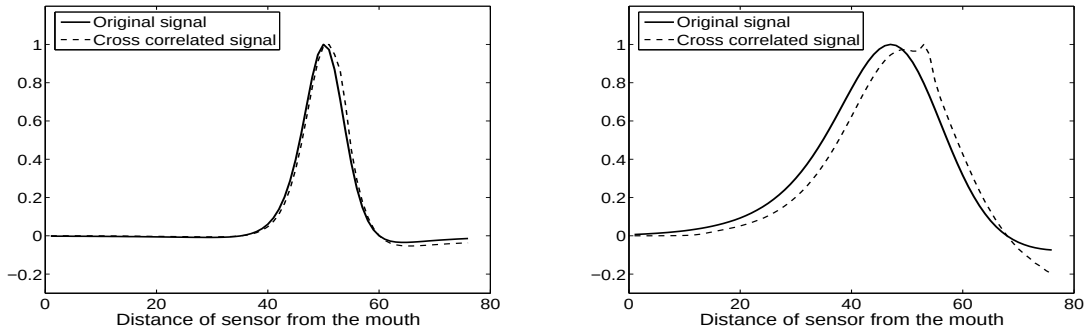
$$a * b = \max_k \{\sum_i a[i]b[k+i]\} \quad (6)$$

where $a[i]$ is the i-th efferency copy signals and $b[i]$ is sensory afferent signal. Normally the cross-correlation has been applied for template matching or for sound localization in the auditory system. We suggest this correlation method can estimate the level of sensory afferents depending on the efference command signals. The electroreceptors can reflect the perturbed signal by neighboring objects. The senosr readings disturbed by other factors should be taken as noise. Thus, the cross-correlation with a sinusoidal waveform of efference copy signals can obtain the noise cancellation. In simulation experiments, noise is modeled as uniform random noise or Gaussian noise to reflect the real electroreception.

Each electroreceptor can process the cross-correlation over the two waveform signals, the common self-generated EOD waveform and the distorted electric signal affected by a target object and noise. Fig. 7 shows the diagram and the result along the rostrocaudal position. Fig. 8 shows the result of the denoised electric signal by cross-correlation.

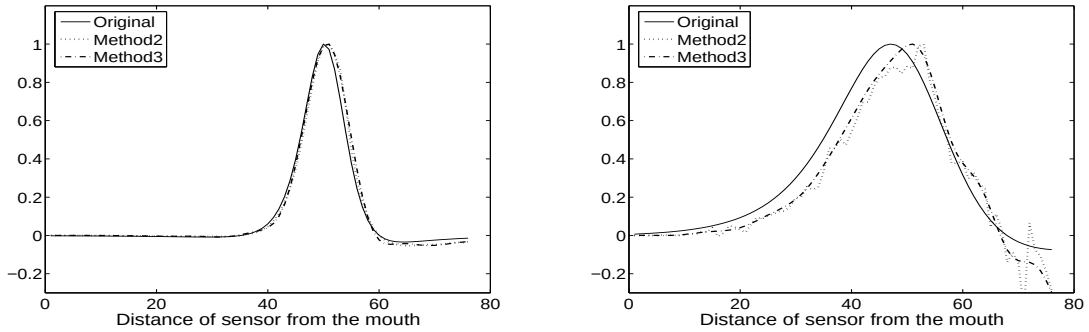
Method3 : Filtering after cross-correlation

After applying the cross-correlation, we obtain noise cancellation for each electroreceptor along the temporal axis. However, the electric image is still noisy along the rostrocaudal line. For accurate localization of a target object, we need to calculate the relative slope, that is, the two param-



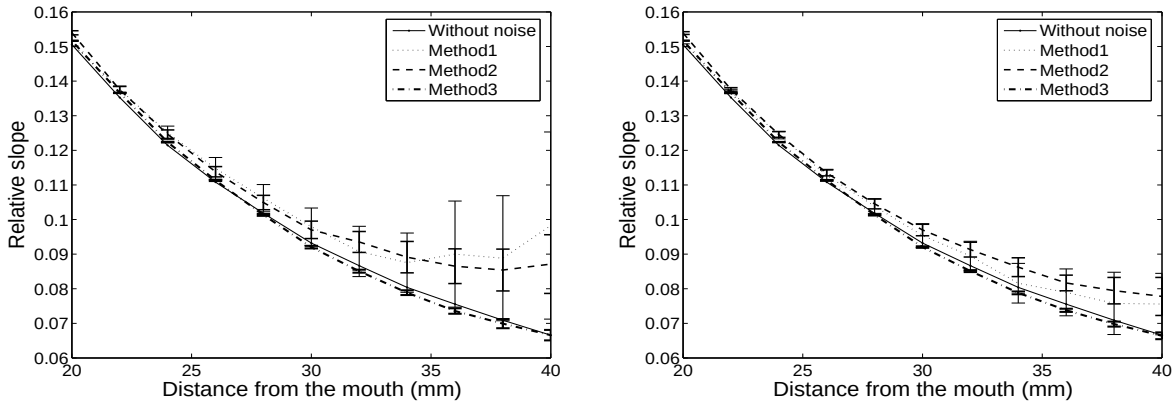
(a) (b)

Figure 8: Normalized denoised electric image using cross-correlated sum when there exist noise uniform noise from -5×10^{-6} to 5×10^{-6}; (a) lateral distance of a target object is $2cm$; (b) $4.8cm$ (solid : electric image without noise, dotted : distorted electric image, dashed : filtered image with cut-off frequency (a) and (b) 20% of the spatial sampling rate)



(a) (b)

Figure 9: Normalized denoised electric image using cross-correlation and a filtering when there exist uniform random noise with distribution range 30×10^{-6} and (a) lateral distance of a target object is $2cm$ (b) $4.8cm$ (solid : relative slope without noise, dotted : using low pass filter, dashed : cross-correlation, dashed dot : filtering after cross-correlation)



(a) (b)

Figure 10: Relative slope (a) uniform noise with range from -5×10^{-6} to 5×10^{-6} (b) Gaussian noise with variance 5×10^{-6} (solid : relative slope without noise, dotted : using low pass filter, dashed : cross-correlation, dashed dot : filtering after cross-correlation)

eters, maximal amplitude and maximal slope. The maximal amplitude can be estimated with the temporal cross-correlation result. However, the maximal slope is involved with the sensor readings along the rostrocaudal spatial axis. We apply a low pass filter over the electric image obtained from the cross-correlation method.

Fig. 9 shows a noise-free original electric image, and the denoised image by cross-correlation over temporal waveforms (method2) and by low pass filtering over the cross-correlation result along the rostrocaudal axis (method3).

	Amount	(1)	(2)	(3)	(4)	(5)	(6)
Method1	RMS	0.0177	0.0054	0.0014	0.0530	0.0047	0.0020
	STD	0.0091	0.0037	0.0009	0.0212	0.0032	0.0015
Method2	RMS	0.0130	0.0065	0.0027	0.0308	0.0045	0.0032
	STD	0.0038	0.0020	0.0004	0.0099	0.0014	0.0008
Method3	RMS	0.0015	0.0014	0.0014	0.0016	0.0014	0.0014
	STD	0.0007	0.0003	0.0001	0.0011	0.0002	0.0001

Table 1: Performance comparison of two method as a mean of error that is difference between relative slopes acquired from clean electric image and denoised image and a mean of standard deviation when the target object moves from $2.0cm$ to $5.0cm$ with interval $0.2cm$ and trial number is 100 (distribution range of uniform noise (1) 10×10^{-6} (2) 5×10^{-6} (3) 1×10^{-6} and variation of Gaussian noise (4) 5×10^{-6} (5) 1×10^{-6} (6) 5×10^{-7} (RMS: root mean square of difference, STD: standard deviation)

When the target object is at a far distance, the cross-correlation outputs over a set of electrosensors still show a rugged pattern of electric image along the spatial axis. The combination of the cross-correlation and low pass filter produces smooth electric image close to the original electric image. It indicates the two-phase filtering process can restore the original electric image from very noisy signals.

The method takes two steps in spatiotemporal dimensions. The electric image is first denoised in the temporal axis and then noise is removed along the spatial axis again. The two-phase filtering process in the spatiotemporal provides desirable slope information along the rostrocaudal axis, and we can extract most accurate relative slope.

Distance measure in noisy environment

From electric images, we can extract the relative slope and Fig. 10 shows the result. The relative slope is dependent on the lateral distance of a target object. The simulation with random noises is repeated fifty times and the performance has been measured. Fig. 10 (a) shows relative slope when the noise is distributed uniformly from -5×10^{-6} to 5×10^{-6} and Fig. 10 (b) shows the result with Gaussian noise whose variance is 5×10^{-6}. When the noise level decreases, we can acquire more similar curves to the relative slope curve in noise-free environment.

When we use low pass filtering after cross-correlation, we can acquire most similar relative slope to the relative slope obtained from noise-free electric signals. Table. 1 shows the performance comparison of three methods to remove noise when uniform and Gaussian noise are tested. The root mean squared error between noise-free relative slope and the filtered relative slope has been measured. We can easily see that the spatiotemporal filtering process greatly improves the performance.

Fig. 11 shows the relative slope changes for each filtering method. When the noise level increases from 1% to 20% of the maximal amplitude, only cross-correlation along the temporal axis, or only low pass filtering along the spatial axis is not much effective to obtain the desired relative slope.

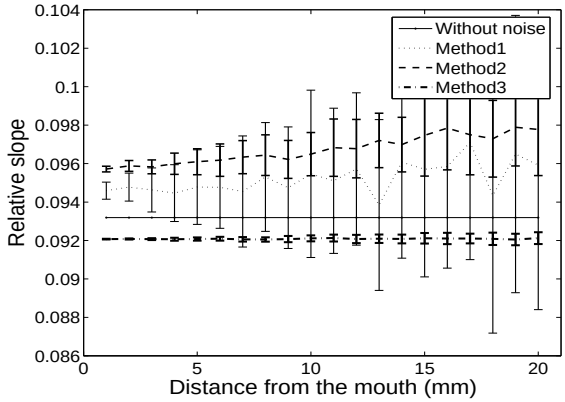


Figure 11: Relative slope when the noise level changes with a fixed target object (solid : relative slope without noise, dotted : using low pass filter, dashed : cross-correlation, dashed dot : filtering after cross-correlation)

It would be difficult to extract the accurate information of the object distance. We note that the cross-correlation can find the appropriate electric signals even for 40% of noise level signals. Weakly electric fish generate periodic EOD signals and we suggest that the self-generated electric signals help obtain the accurate information of distance of a target object in noisy environment.

Conclusion

Noisy signals are inevitable in the underwater environment. The electric signals generated by other underwater animals may be mixed up with the signals that the electric fish produces. In that environment, it is important to extract pure information of its own electric signal in the sensor readings.

An easy and simple method to remove noise in electric image is the filtering method. In this paper, it is shown that an electric image can be restored by low pass filter along the rostrocaudal axis when the noise level is small enough to remove. However, when the maximum amplitude of an electric signal decreases, the electric signal is distorted severely. The distance range in which the weakly electric fish can de-

tect an object is very narrow, and it is known that weakly electric fish use the electrolocation based on distance (Nelson and MacIver, 2006; Babineau et al., 2007). Direct measurement of relative slope over raw electric signals can produce wrong estimation of the distance of a target object.

We use cross-correlation as an alternative method to obtain denoised electric image. Cross-correlation is generally used to measure the similarity of two signals. The cross-correlated sum becomes maximal when the frequency and phase of the two waveforms exactly matches. It is known that individual weakly electric fish discriminate electric signals that are characterized by species, sex, and another member of conspecifics (Kramer, 1994). If frequencies of EOD waveforms are different, then the cross-correlated sum has small value. Consequently, the cross-correlation has advantage to separate their own electric signals from another electric signals.

As shown in Fig. 10, we notice that the desired relative slope can be obtained when we take two steps for elimination of noise, cross-correlation and low pass filtering in spatiotemporal dimensions. The root mean square of difference and variance become much smaller even when a target object is far away from the weakly electric fish. The periodic efference copy signal used in the cross correlation is critical to remove a high level of noise. We suggest that the periodic waveform of EOD signals help localization of a target object such as prey or predator.

The electroreception of weakly electric fish can be applied to a robotic system to localize a target object in the underwater. The electric field can spread to every direction and it can be used to detect not only the location of a target object but also shape and size (Schwarz and von der Emde, 2001). These characteristics of the electroreception can be useful in the dark underwater environment. For the future work, we will test the electrolocation system with a robotic fish and show the possibility of application of electrosensors in the submarine system.

Acknowledgements

This work was supported by Mid-career Researcher Program through NRF grant funded by the MEST (No. 2010-0000460).

References

Babineau, D., Lewis, J., and Longtin, A. (2007). Spatial acuity and prey detection in weakly electric fish. *PLoS Comput Biol*, 3(3):e38.

Bastian, J. (1994). Electrosensory organisms. *Physics Today*, pages 30–37.

Caputi, A. and Budelli, R. (2006). Peripheral electrosensory imaging by weakly electric fish. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 192(6):587–600.

Chen, L., House, J., Krahe, R., and Nelson, M. (2005). Modeling signal and background components of electrosensory scenes. *Journal of Comparative Physiology A: Sensory, Neural, and Behavioral Physiology*, 191(4):331–345.

Fugere, V. and Krahe, R. (2010). Electric signals and species recognition in the wave-type gymnotiform fish Apteronotus leptorhynchus. *Journal of Experimental Biology*, 213(2):225.

Kramer, B. (1994). Communication behavior and sensory mechanisms in weakly electric fishes. *Advances in the Study of Behavior*, 23:233–270.

Kramer, B. (1999). Waveform discrimination, phase sensitivity and jamming avoidance in a wave-type electric fish. *Journal of Experimental Biology*, 202(10):1387.

Lissmann, H. (1958). On the function and evolution of electric organs in fish. *Journal of Experimental Biology*, 35(1):156.

Nelson, M. and MacIver, M. (1999). Prey capture in the weakly electric fish Apteronotus albifrons: sensory acquisition strategies and electrosensory consequences. *Journal of Experimental Biology*, 202(10):1195.

Nelson, M. and MacIver, M. (2006). Sensory acquisition in active sensing systems. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 192(6):573–586.

Nelson, M., MacIver, M., and Coombs, S. (2000). Modeling electrosensory and mechanosensory images during the predatory behavior of weakly electric fish. *Brain, Behavior and Evolution*, 59(4):199–210.

Rasnow, B. (1996). The effects of simple objects on the electric field of Apteronotus. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 178(3):397–411.

Schwarz, S. and von der Emde, G. (2001). Distance discrimination during active electrolocation in the weakly electric fish Gnathonemus petersii. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 186(12):1185–1197.

Sim, M. and Kim, D. (2010). Understanding the Electrosensory System of Weakly electric Fish with a Spatial and Temporal Information (in preparation).

von der Emde, G. (1998). Capacitance detection in the wave-type electric fish Eigenmannia during active electrolocation. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 182(2):217–224.

von der Emde, G. (2006). Non-visual environmental imaging and object detection through active electrolocation in weakly electric fish. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 192(6):601–612.

von der Emde, G. and Fetz, S. (2007). Distance, shape and more: recognition of object features during active electrolocation in a weakly electric fish. *Journal of Experimental Biology*, 210(17):3082.

Ecology

Enhancing Creativity with Niche Construction

Jon McCormack

Centre for Electronic Media Art, Monash University, Clayton 3800, Australia
Jon.McCormack@monash.edu

Abstract

We address the question of how processes from evolutionary biological ecosystems can be abstracted and beneficially applied in creative domains. Evolution is a process capable of generating appropriate (fit) novelty in biological systems, so it is interesting to ask if it can do so in other, non-biological systems. Past approaches have focused on optimisation via fitness evaluation (either machine representable or human evaluated), but this is ill-suited to creative systems, as creativity is not necessarily an optimisation process. Our approach is to consider the creative system as a virtual evolutionary ecosystem, specifically adopting the process of niche construction. We show how the abstracted niche construction process can be applied to an agent-based line drawing system, enhancing the diversity and heterogeneity of drawings produced over a version without niche construction.

Introduction

Two well known systems exhibiting creativity are the human brain and evolution. While advances in neurological understanding of creative processes and aesthetics are ongoing (Perlovsky, 2010; Griffiths, 2008; Ramachandran and Hirstein, 1999), both the cognitive and social processes that lead to creative outcomes remain difficult to quantify, and hence, to simulate. Evolutionary processes, on the other hand, are far better understood and continue to be successfully studied using a variety of simulation methods.

In this paper we explore the adaptation of evolutionary ecological processes to problems in creative design. As a process, evolution is eminently capable of novel design, having innovated things such as prokaryotes, eukaryotes, higher multicellularity and language, through a non-teleological process of replication and selection (Maynard Smith and Szathmáry, 1995; Nowak, 2006). While much exists on what constitutes human creativity – e.g. Boden (2004); Sternberg (1999)) – for the purposes of this paper we consider creativity more generally as the *appropriate novelty* exhibited by a system. 'Appropriate' in that the artefacts produced are fit or useful in some domain, and 'novel' in that the system is capable of repeatedly producing artefacts that it has not produced before[1].

Darwinian processes of selection and replication with difference only provide a simplified picture of natural evolution. Many have argued that explaining the growth of complexity that typifies the creativity of evolution requires a broader consideration of the systems of the natural world (Maynard Smith and Szathmáry, 1995; Laland et al., 1999; Gould, 2002). In recent years, that has meant, for example, increasing our understanding of (i) the effects of evolution on the processes of ontogenetic development (Carroll, 2005) (ii) the interdependent relationships between species and their environment: *ecosystems*. This second approach is the one adopted in the work described here.

Evolution and Aesthetic Creativity

The field of Evolutionary Computing (EC) has adopted the metaphor of genetic evolution to successfully solve problems in search, optimisation and learning. Where EC has been less successful, however, is in tackling problems of creativity, in particular artistic creativity, as it is difficult to conceptualise creative artefacts in terms of a single (or multi-objective) optimisation or general machine-representable fitness evaluations.

A popular EC approach to using evolution in creative contexts is the *Interactive Genetic Algorithm* (IGA), in which the fitness evaluation of a standard genetic algorithm is performed by a human, who may use any (subjective) criteria to assign fitness to individuals in a population (Takagi, 2001).

In the context of the application presented in this paper (line drawing) the system of Baker and Seltzer (1994) used variable length genomes representing an ordered set of strokes to define a line drawing. Each stroke included parameters in the genome to affect the way drawing is interpreted, including space enclosing, relation to the next stroke (e.g. separate or joined) and symmetry operations. Drawings were evolved using an IGA. The system could be seeded with random genotypes or genotypes created by interpret-

[1] For a more formal specification of this relatively informal definition, see McCormack (2010).

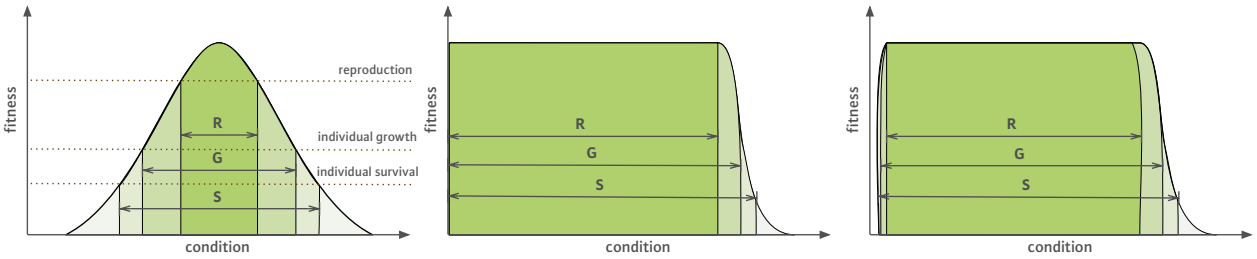


Figure 1: Example organism viability curves for reproduction, growth and survival, from (Begon et al., 2006).

ing the strokes of a human artist. The *Drawbots* system of Bird et al. (2008) attempted to create a line-drawing robot using evolutionary robotics. They defined "implicit" fitness measures that did not restrict the type of marks the robot drawer should make, including an "ecological model" involving interaction between environment resource acquisition and expenditure through drawing. However, the results demonstrated only minimal creativity, and the authors concluded that fitness functions that embodied "artistic knowledge about 'aesthetically pleasing' line patterns" would be necessary if the robot were to make drawings worthy of exhibition.

Formalised "aesthetically pleasing" fitness measures of any generality have been difficult to find, despite a number of attempts (see e.g. Birkhoff (1933); Staudek (2002); Ramachandran (2003); Svangåard and Nordin (2004); Machado et al. (2008)), hence the use of the IGA. While the IGA has achieved some success in a variety of domains, in general it suffers from a host of problems, particularly for creative applications (McCormack, 2005). The most commonly cited of these is "user fatigue", where human users quickly tire of the repetitive act of phenotype evaluation (Takagi, 2001), limiting the range of evolutionary exploration possible. In general, IGAs are more valuable to non-experts, who may lack the sophisticated understanding of how to design and manipulate a medium for creative purposes.

More importantly, for most creative domains the idea of evolving towards a single optimum is counterintuitive, as an artist or designer normally produces many new artefacts over their professional lifetime. New designs often 'evolve' from previous ones, offspring of both the originating artist and her peers (Basalla, 1998). Indeed, as Basalla (1998) and others have pointed out using the example of technological evolution, the Western emphasis on individual creativity (reinforced socially through patents and other awards) obscures the important roles played in the evolutionary ecosystem of interactions between environment and prior work of many individuals.

Thus, an alternate approach to the narrow individual optimisations of standard EC methods, is to consider the interaction of components in an evolutionary ecosystem, as such a system can potentially exploit facets of evolution other than single optimisations. In the research presented in this paper, we examine the biological process of *niche construction*, whereby organisms modify their heritable environment. The concept of *niching* has been successfully used in EC previously, particularly in problems requiring multiple solutions (Mahfoud, 1995). However, niching in EC is primarily about maintaining stable sub-populations to improve the efficiency and efficacy of search – in general these methods do incorporate the biological concept of niche construction in their methodology, as is the case with the methods described in this paper. Before explaining the concept in more detail, we give a brief overview of the concept of a niche and niche construction.

Niches

In broad terms, biological environments have two main properties that determine the distribution and abundance of organisms: *conditions* and *resources*. Conditions are physiochemical features of the environment (e.g. temperature, pH, wind speed). An organism's presence may change the conditions of its local environment (e.g. one species of plant may modify local light levels so that other species can be more successful). Conditions may vary in cyclic patterns or be subject to the uncertainty of prevailing environmental events. Conditions can also serve as stimuli for other organisms. Resources, on the other hand, are consumed by organisms in the course of their growth and reproduction. One organism may become or produce a resource for another through grazing, predation, parasitism or symbiosis, for example.

For any particular condition or resource, an organism may have a preferred value or set of values that favour its survival, growth and reproduction. Begon et al. (2006) define three characteristic curves, which show different "viability zones" for survival, growth and reproduction (Fig. 1).

The complete set of conditions and resources affecting an organism represent its *niche*, which can be conceptualised as a hypervolume in n-dimensional space. As an example, for two conditions c_1 and c_2, two different types of species

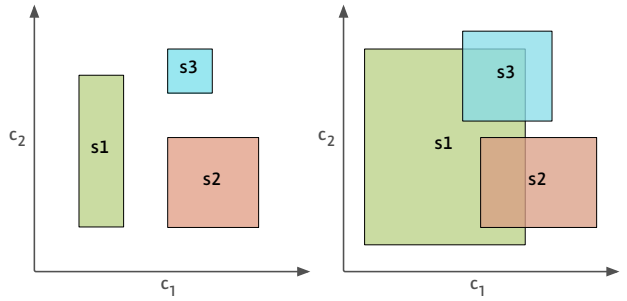


Figure 2: Example exclusive and overlapping niche areas for a two-dimensional set of conditions.

relationships are shown in Fig. 2. The shaded area represents the viability zone for the species. A species will only survive if conditions are maintained within this shaded area. A relatively large distance in any single dimension denotes a generalist *in that dimension* (s_1 is relatively generalist in c_2), specialists have small distances (s_3 is more specialised in both c_1 and c_2). This size is referred to as *niche width*, and may vary for each dimension. If the mean viability zones overlap in a particular dimension, multiple species can coexist within the range of overlap.

Competition and other species interactions are important in determining habitat distribution. Niche differentiation can permit coexistence of species within a biotope. Higher number of species can coexist by utilising resources in different ways. It is reasonably well understood in Biology how these mechanisms give rise to species diversity and specialisation.

The challenge addressed in this paper is to devise useful ways of employing these mechanisms in non-biological contexts. An important problem is in devising *appropriate* mappings between conditions and resources, and establish trade-offs for an individual's survival based on tolerances to specific conditions in order to enhance the quality and diversity of output in a creative generative system.

Niche Construction

Niche construction is the process whereby organisms change their own and each other's niches. They do this by modifying or influencing their local environment. Proponents of niche construction argue for its importance in understanding the feedback dynamics of evolutionary process in nature (Odling-Smee et al., 2003). By modifying their niche, either reinforcing or degrading it, organisms provide a heritable environment for their offspring. Hence niche construction can create forms of feedback that modify the dynamics of the evolutionary process, because ecological and genetic inheritance co-influence the evolutionary process. Computational models of niche construction show that it can influence the inertia and momentum of evolution and introduce or eliminate polymorphisms in different environments (Day et al., 2003). Other models have demonstrated that a simple niche constructing ecosystem can support homeostasis and bi-stability similar to that of Lovelock's popular *Daisyworld* model (Dyke et al., 2007).

Whereas standard evolutionary algorithms tend to converge to a single (sub)-optimum, niche construction can promote diversity and heterogeneity in an otherwise fixed and homogeneous evolutionary system. In creative systems where the design of an explicit fitness function may be difficult or impossible, niche construction provides an alternate mechanism to explore a generative system's diversity over more traditional methods, such as the IGA. An "ecosystemic" approach to creative systems recognises that multiple designs may be equally valid and interesting, the emphasis shifting from single optimised solutions to the exploration of appropriate novelty offered through the feedback dynamics of an evolutionary ecosystem (McCormack, 2007).

Processes such as niche construction may serve as a type of "design pattern" (Gamma, 1995) that facilitates the building of creative evolutionary systems. To illustrate the utility of niche construction, we will describe a series of experiments where niche construction influences the structure and variation of the creative artefacts produced in an agent-based line drawing system.

Case 1: Line Drawing Agents

We will consider a simple creative system that autonomously draws lines with ink on a page. This system is inspired by Mauro Annunziato's *The Nagual Experiment* (Annunziato, 2002), which consisted of simple line drawing agents controlled by stochastic processes. In Annunziato's original system he changed the global characteristics of the drawings produced through manual adjustment of line-drawing probability parameters, such as fecundity, mortality and curvature. The resulting drawings have been acknowledged as artistically interesting and demonstrate the richness of creative output possible from a relatively simple generative specification.

Our system consists of a population of haploid line-drawing agents who inhabit a two-dimensional drawing surface or *canvas*. The canvas is initially blank (white). Agents roam over the surface, leaving a trail of black ink that marks out the path they travel. If a drawing agent intersects with an existing line, drawn either by itself or another agent, it dies. An agent may undergo reproduction during its lifetime, with offspring placed adjacent to the parent. The canvas is seeded with a small initial population of *founder agents*, initialised with uniformly distributed random genomes, that proceed to move, draw and reproduce. There is no limit to the number of offspring an agent may have, but in general the lifespan of agents decreases as the simulation progresses since the density of lines becomes greater, making it increasingly difficult to avoid intersection with existing lines. Eventually the entire population dies out (predominantly due to the intersec-

tion rule), and the image is finished. This finished drawing represents the "fossil record" of all the generations of lines that were able to live over the lifetime of the simulation.

In this first experiment, agents have no sensory information about their environment, for example they cannot detect proximity to an existing line or other agent. Thus, the characteristics of the line an agent draws are determined by genetics, with the genome serving as the control parameters of a stochastic process. An agent's genome is specified by the following alleles, each represented as a normalised floating point value:

curvature (σ), controls the rate of curvature of the line ($\frac{d\theta}{dt}$, where θ is the heading direction). Curvature varies from a straight line (0) to a maximum curvature rate (1);

irrationality (r), controls the rate and degree of change in the rate of curvature according to a stochastic algorithm (detailed below, see also Fig. 3);

fecundity (f), the probability of the agent reproducing at any time step. New agents are spawned as branches from the parent;

mortality (m), the probability of the agent dying at any time step;

offset (ϕ), the offset angle of child filaments from the parent;

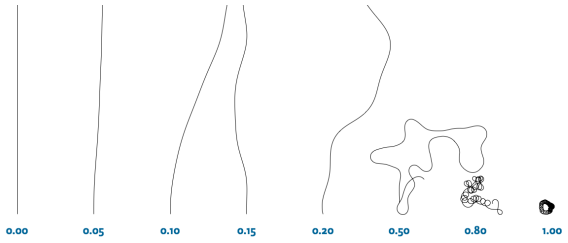


Figure 3: Individual line drawing agents with different measures of irrationality. Note that the 'die if intersect' rule has been turned off for these examples.

In addition each agent maintains *state* information which includes the current position on the canvas, heading direction, speed and current rate of curvature. Changes to the rate of curvature are determined by the **curvature** and **irrationality** alleles, with the overall rate of change given by

$$\frac{d\theta}{dt} = \sigma + \texttt{fracSum}(\boldsymbol{p}, k \cdot r)^{0.89r^2}, \qquad (1)$$

where $\boldsymbol{p}$ is the agent's current position, k a constant known as the *octave factor*, and `fractSum` a function that sums octaves of Perlin (2002) 2D noise. This function was chosen as it gives band limited, continuous stochastic variation with second order continuity, and is statistically invariant under affine transformation. Increasing r (irrationality) increases the octaves of noise, changing the rate of change in direction in increasingly finer detail. Fig. 3 shows the effects of varying the irrationality allele, r, over its normalised range.

This system was run a number of times varying the random number seed and location of founder agents on the blank canvas. At each time step the **fecundity** and **mortality** alleles determine probabilistically if an agent will die or reproduce. In the case of reproduction, child agents are placed next to the parent line, with their heading determined by the offset allele (ϕ). A child agent's genome may undergo mutation (modification of an allele by adding a Normally distributed random number with mean 0). Additionally, children have a short gestation period before they begin to draw, allowing the parent to continue drawing past the point where reproduction took place, avoiding intersection with their offspring.

The images that emerge from this process demonstrate a wide variety of output possible from this system (two sample images are shown in Fig. 4). While there is no explicit fitness function or evaluation, implicit agent fitness is determined by a combination of genetics and environment. Importantly, the environment is constantly changing. As drawing progresses, it becomes increasingly difficult to reproduce and live, since the probability of intersecting with an existing line typically becomes higher as more lines crowd the canvas.

While the images produced by this system are interesting, in general they lack a changing dynamic or visual counterpoint, that is, they are largely homogeneous in structure, or have progressive changes that take place as genes mutate through drift. Much of the overall structure is determined by the founder lines, who can carve up large areas of blank canvas for themselves and their offspring, preventing other lines from entering. Genetically similar offspring continue to reproduce inside these boundaries until the space is filled.

Case 2: Line Drawing with Niche Construction

In a second experiment we tested the hypothesis that by introducing an ecosystemic process of niche construction into the system, the overall diversity and heterogeneity of images produced by the system could be significantly enhanced. To do this, each agent was given an additional allele in its genome: a local density preference δ_i (a normalised floating point value). This defines the agent's preference for the density of lines already drawn on the canvas in the immediate area of its current position, i.e. its *niche* (Fig. 5). In a preferred niche, an agent is more likely to give birth to offspring and has a better chance of survival. As children inherit their parent's genes they are more likely to survive as they have a similar density preference. So in a sense, parents may construct a niche and pass on a heritable environment well-suited to their offspring.

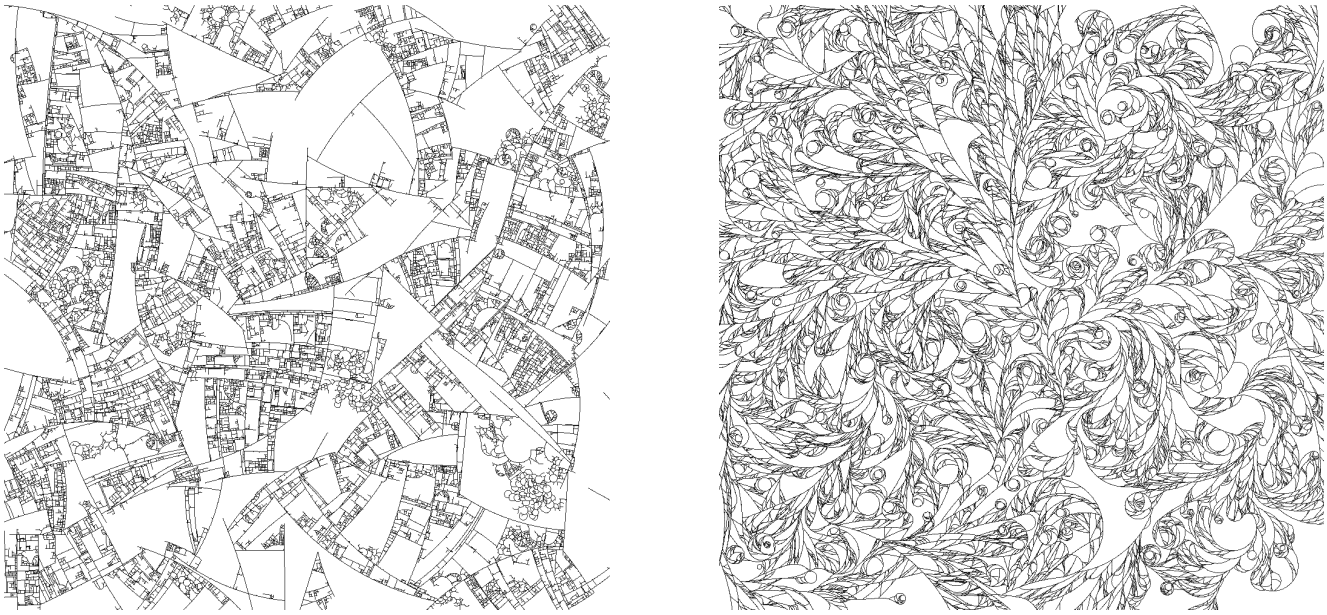


Figure 4: Two sample outputs from the line drawing system (without niche construction).

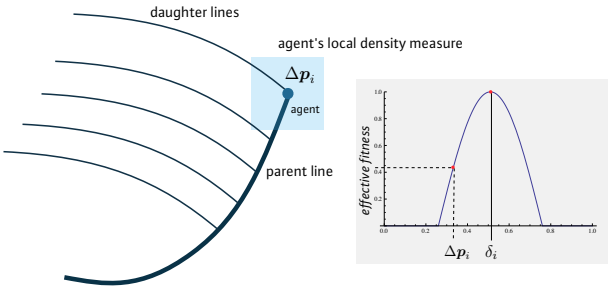


Figure 5: The niche construction mechanism for drawing agents, who try to construct a niche of local density that satisfies their genetic preference.

For each agent, i, δ_i defines it's preferred niche. Local density, defined as the ratio of inked to blank canvas per unit area, is measured over a small area surrounding the agent at each time step. Proximity to the preferred niche determines the probability of reproduction, given by

$$Pr(rep) = f_i \cdot \cos^{\omega}(\texttt{clip}(2\pi(\Delta_{\boldsymbol{p_i}} - \delta_i)), -\frac{\pi}{2}, \frac{\pi}{2}), \quad (2)$$

where $\Delta_{\boldsymbol{p_i}}$ is the local density around the point $\boldsymbol{p_i}$, the agent's position, ω a global parameter that varies the effective niche width, f_i is the agent's fecundity and `clip` is a function that limits the first argument to the range specified by the next two. Being in a non-preferred niche similarly increases the probability of death.

Founder agents begin with a low density preference, uniformly distributed over $[0, 0.2]$. Beginning the drawing on a blank canvas means that only those agents who prefer a low density niche will survive. As the drawing progresses however, more ink is added to the canvas and agents who prefer higher densities will prosper. As with the previous experiment, at birth the agent's genome is subject to the possibility of mutation (proportional to the inverse of the genome length), allowing offspring to adapt their density preference and drawing style as the drawing progresses. Eventually the population becomes extinct, since higher density favouring agents don't have much room to move, and the drawing finishes. Some example drawings are shown in Fig. 6. Notice the greater stylistic variation and heterogeneity over the images shown in Fig. 4.

Analysis and Discussion

Visually, the examples appear to show that by adding niche construction, the line drawing system is capable of producing images with greater heterogeneity, variation in density, counterpoint and overall visual interest (Fig. 7). We might even be tempted to say it is more creative.

To support this intuition, a number of images produced using the niche constructing and non-niche constructing versions were analysed statistically. A total of 40 images were sampled: 20 niche constructed and 20 non-niche constructed. For each image, the mean density ($\bar{\Delta}$) and variance of density over the entire image was computed. Then for each set (non-niche constructed, niche constructed) the variance of mean density and the mean density variance was calculated. Table 1 summarises this analysis. p-values were calculated using a Welch t-test. As shown in the table, niche

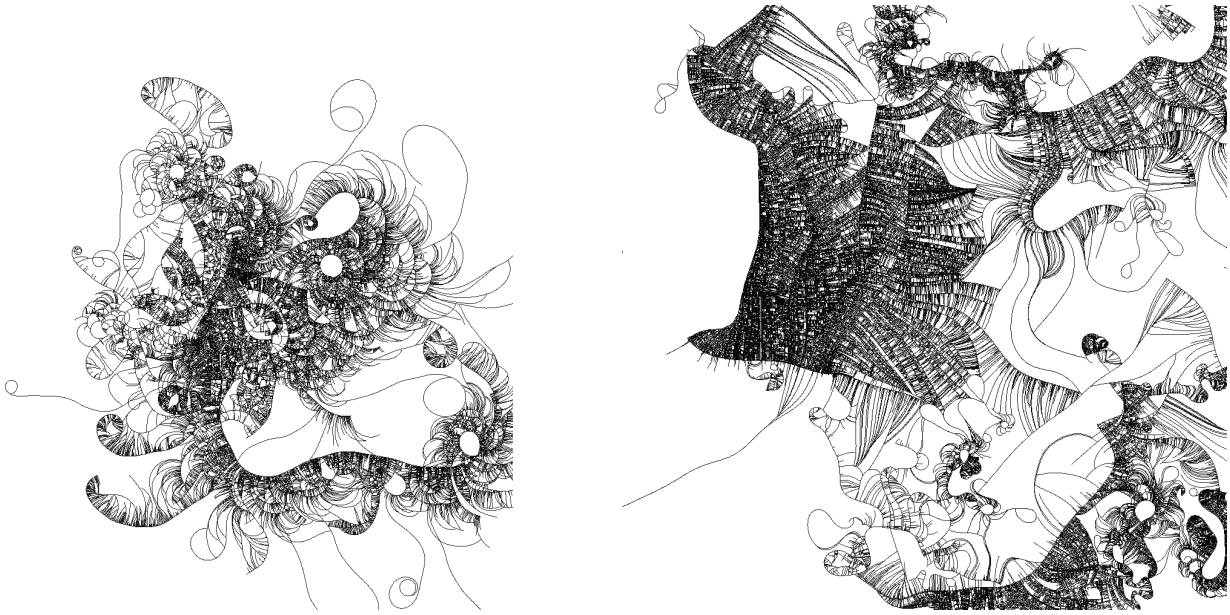


Figure 6: Two sample outputs from the line drawing system with niche construction.

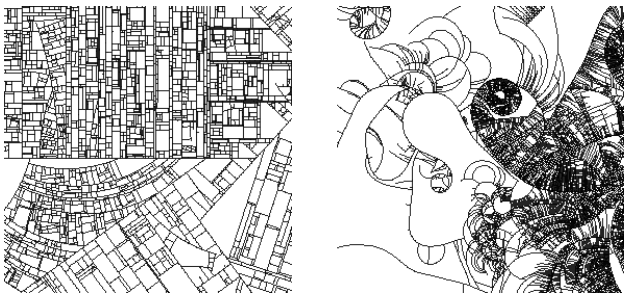


Figure 7: Detail from two drawings, showing density variation (left) without niche construction, and (right) with niche construction.

constructed images exhibit a far greater variation in overall density (by a factor of 3.83). Significantly, the density variation *over each image* is, on average, 4.31 times greater for the niche constructed over non-niche constructed drawings.

	Non NC	NC	*p*-value
Number of Images	20	20	–
Variance of $\bar{\Delta}$	0.00298	0.0114	0.0634
Mean Variance	0.0140	0.0604	1.57×10^{-10}

Table 1: Density variation between non-niche constructed and niche constructed drawings.

Analysis of the mean agent density preference, $\bar{\delta} = \frac{1}{n}\sum_{i=1}^{n}\delta_i$, at each epoch shows an overall adaption to the mean image density ($\bar{\Delta}$) over the lifetime of the drawing, indicating that agents evolve to fit niches (Fig. 8). On average, agents favour slightly denser niches than currently exist (the line in the figure is always positive), we infer this is because an agent's density measure is always centred around the agent's current location, and this will necessarily include parts of the images with lines drawn (even if only the agent's own trail). The value of $\bar{\Delta}$ tends to increase over the life of the drawing. This is not surprising, as there is no mechanism for an agent to *reduce* the density of its niche[2]. The best any parent can do is carve out the largest possible border around empty space, so that its offspring can grow without fear of intersecting with other parents or their offspring.

Conclusions and Future Work

We have demonstrated how the ecological "design pattern" of niche construction can be used to enhance the creative output of a generative line-drawing system. Elsewhere, (McCormack and Bown, 2009), we have also applied a similar process in the sound domain, leading to on-going change in an agent-based sound generation system. While it may be premature to suggest the generality of this method, our on-going experiments demonstrate that with the appropriate design, niche construction can introduce heterogeneity and useful variation into creative generative systems.

The line-drawing agents described in this paper have only one way to sense their environment: through their density preference. A more sophisticated system might give agents greater sensory capabilities so that they can better optimise

[2]An observed (short-lived) strategy is to draw a closed circular area and not place any offspring in it, but this only generates a low-density niche after death!

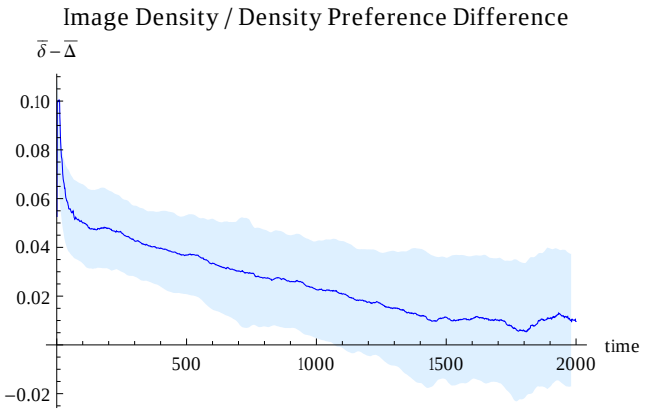


Figure 8: Difference between mean image density and mean agent density preference averaged over 40 runs. The standard deviation is shown in light blue.

their niche construction to their environment. For example being able to sense proximity to another line would allow more graphically complex strategies to evolve.

Additionally, the agents are limited in their productive utilisation of evolution, as any adaptation must take place over the life of a single drawing. Typically, 10^3 – 10^5 offspring may be produced in a single image, but less than 10 – 30 generations from the initial parent. Essentially, all lines are of the same species. An improved strategy would be to allow different species of line-drawing agents to be pre-evolved on test canvases, permitting better optimisation for different density niches and inter-species interactions. These pre-evolved species could then share a common drawing canvas in order to produce a more complex finished drawing, better adapted to their specific niche requirements. We are currently exploring this idea. One can imagine that the next generation of artist's drawing systems could incorporate such pre-evolved drawing agents as "intelligent brushes"; the artist selecting from a palette of pre-evolved styles and applying them to the canvas at various stages. Agents with different niche density preferences try to draw in order to construct their preferred niche, but their interactions with each other could result in the emergence of competitive or cooperative strategies.

In summary, we believe that niche construction is a useful technique that can be successfully exploited in generative creative systems to enhance the dynamics and heterogeneity of output produced. The ecosystemic approach favoured in this paper is in contrast to previous IGA or fitness-based GA systems aimed at search or optimisation to singular outcomes or subjective criteria. The complex dynamics of ecosystem processes are a source of rich and varied inspiration that has much to offer as we develop autonomous creative systems.

Acknowledgements

This work was supported by Australian Research Council Discovery Projects grant DP0877320.

References

Annunziato, M. (2002). The nagual experiment. http://www.plancton.com/papers/nagual.pdf (Accessed 11 November 2008)

Baker, E. and Seltzer, M. I. (1994). Evolving line drawings. In *Graphics Interface '94*, pages 91–100, Banff, Canada.

Basalla, G. (1998). *The Evolution of Technology*. Cambridge University Press, Cambridge, MA.

Begon, M., Townsend, C., and Harper, J. (2006). *Ecology: from individuals to ecosystems*. Wiley-Blackwell.

Bird, J., Husbands, P., Perris, M., Bigge, B., and Brown, P. (2008). Implicit fitness functions for evolving a drawing robot. In Giacobini, M., Brabazon, A., Cagnoni, S., Caro, G. D., Drechsler, R., Ekárt, A., Esparcia-Alcázar, A., Farooq, M., Fink, A., McCormack, J., O'Neill, M., Romero, J., Rothlauf, F., Squillero, G., Uyar, S., and Yang, S., editors, *EvoWorkshops*, volume 4974 of *Lecture Notes in Computer Science*, pages 473–478. Springer.

Birkhoff, G. D. (1933). *Aesthetic Measure*. Harvard University Press, Cambridge, MA.

Boden, M. A. (2004). *The creative mind: myths & mechanisms*. (2nd edition), Routledge, Oxon.

Carroll, S. (2005). *Endless forms most beautiful: The new science of evo devo and the making of the animal kingdom*. WW Norton & Company.

Day, R. L., Laland, K. N., and Odling-Smee, J. (2003). Rethinking adaptation: the niche-construction perspective. *Perspectives in Biology and Medicine*, 46(1):80–95.

Dyke, J. G., McDonald-Gibson, J., Di Paolo, E., and Harvey, I. R. (2007). Increasing complexity increases stability in a self regulating ecosystem. In *Proceedings of IXth European Conference on Artificial Life, ECAL 2007*, LNCS 4648, pages 133–142.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design patterns: elements of reusable object-oriented software*. Addison-Wesley professional computing series. Addison-Wesley, Reading, Mass.

Gould, S. J. (2002). *The Structure of Evolutionary Theory*. Harvard University Press, Cambridge, MA.

Griffiths, T. D. (2008). Capturing creativity. *Brain*, 131(1):6–7.

Laland, K. N., Odling-Smee, J., and Feldman, M. W. (1999). Niche construction, biological evolution and cultural change. *Behavioral and brain sciences*, 23(01):131–146.

Machado, P., Romero, J., and Manaris, B. (2008). Experiments in computational aesthetics. In Romero, J. and Machado, P., editors, *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*, Natural Computing Series, pages 381–415. Springer.

Mahfoud, S. W. (1995). A comparison of parallel and sequential niching methods. In *Proceedings of the 6th International Conference on Genetic Algorithms*, page 143, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Maynard Smith, J. and Szathmáry, E. (1995). *The major transitions in evolution*. W.H. Freeman Spektrum, Oxford; New York.

McCormack, J. (2005). Open problems in evolutionary music and art. In Rothlauf, F., Branke, J., Cagnoni, S., Corne, D. W., Drechsler, R., Jin, Y., Machado, P., Marchiori, E., Romero, J., Smith, G. D., and Squillero, G., editors, *EvoWorkshops*, volume 3449 of *Lecture Notes in Computer Science*, pages 428–436. Springer.

McCormack, J. (2007). Creative ecosystems. In Cardoso, A. and Wiggins, G., editors, *Proceedings of the 4th International Joint Workshop on Computational Creativity*, pages 129–136.

McCormack, J. (2010). Creative evolution. In *preparation*.

McCormack, J. and Bown, O. (2009). Life's what you make: Niche construction and evolutionary art. In Giacobini, M., Brabazon, A., Cagnoni, S., Caro, G. A. D., Ekárt, A., Esparcia-Alcázar, A., Farooq, M., Fink, A., Machado, P., McCormack, J., O'Neill, M., Neri, F., Preuss, M., Rothlauf, F., Tarantino, E., and Yang, S., editors, *EvoWorkshops*, volume 5484 of *Lecture Notes in Computer Science*, pages 528–537. Springer.

Nowak, M. A. (2006). *Evolutionary Dynamics: exploring the equations of life*. The Bekknap Press of Harvard University Press, Cambridge, Massachusetts, and London, England.

Odling-Smee, J., Laland, K. N., and W., F. M. (2003). *Niche Construction: The Neglected Process in Evolution*, Monographs in Population Biology. Princeton University Press.

Perlin, K. (2002). Improving noise. *ACM Transactions on Graphics (TOG)*, 21(3):681–682.

Perlovsky, L. I. (2010). Intersections of mathematical, cognitive, and aesthetic theories of mind. *Psychology of Aesthetics, Creativity and the Arts*, 4(1):11–17.

Ramachandran, V. S. (2003). *The Emerging Mind*. Reith lectures; 2003. BBC in association with Profile Books, London.

Ramachandran, V. S. and Hirstein, W. (1999). The science of art: A neurological theory of aesthetic experience. *Journal of Consciousness Studies*, 6:15–51.

Staudek, T. (2002). *Exact Aesthetics. Object and Scene to Message*. PhD thesis, Faculty of Informatics, Masaryk University of Brno.

Sternberg, R. J. (1999). *Handbook of creativity*. Cambridge University Press, Cambridge, U.K.; New York.

Svangåard, N. and Nordin, P. (2004). Automated aesthetic selection of evolutionary art by distance based classification of genomes and phenomes using the universal similarity metric. In Raidl, G. R., Cagnoni, S., Branke, J., Corne, D., Drechsler, R., Jin, Y., Johnson, C. G., Machado, P., Marchiori, E., Rothlauf, F., Smith, G. D., and Squillero, G., editors, *EvoWorkshops 2004*, volume 3005 of *Lecture Notes in Computer Science*, pages 447–456. Springer.

Takagi, H. (2001). Interactive evolutionary computation: Fusion of the capabilities of ec optimization and human evaluation. *Proceedings of the IEEE*, 89:1275–1296.

Maintenance of Species Diversity by Predation in the Tierra System

Jie Shao and Thomas S. Ray

Department of Zoology, University of Oklahoma, Norman, Oklahoma 73019, USA
jshao@ou.edu

Abstract

One of the ecological theories has proposed that high species diversity can be maintained by predation, and several experimental studies showed that a few predator individuals with positive frequency-dependent behavior were able to maintain the coexistence of two prey types. However, in a natural environment, when a single predator species regulates the diversity of prey species, it is likely to be a full population of predators, not just a few individual predators. The role of a predator population in maintaining species diversity has not been carefully investigated in laboratory experiments but has been seriously questioned by computer simulations. In this paper, we introduce predation into the Tierra system and the dynamic relationship between prey and predator populations is examined. The robust appearance of the "Lotka-Volterra-like" cycle in Tierra suggests that the digital creatures may follow the same fundamental principles as their organic counterparts. Moreover, when each predator in a large predator population searches for prey in its neighboring area and performs positive frequency-dependent predation based on local prey abundance, a global pattern of coexistence of prey species emerges. This suggests that positive frequency-dependent predation may be a reasonable mechanism to maintain species diversity in nature.

Introduction

Species diversity is one of the most ubiquitous and spectacular phenomena in nature, but how it may arise, persist and shape the evolutionary process is poorly understood. One of the ecological theories has proposed that high species diversity can be maintained by predation. A few dominant species grow rapidly and crowd out many of the other species, but this reduction of species diversity due to competitive exclusion can be avoided by the presence of predators. Predators limit the populations of dominant species and thus more resources become available to support the survival of other prey species. Several experimental studies demonstrated that the presence of predator species prevented the diversity of prey species from declining (Paine, 1974; Morin, 1981). At the same time, the coexistence of multiple prey species provides more feeding options for predators. To avoid competing for the same resource, predator species may specialize to adapt to different prey types (Stanley, 1973). Therefore, predation may facilitate the increase of diversity in both prey and predator species.

Further experimental studies on predation mechanisms revealed that a predator may switch among different prey types in response to their abundance and positive frequency-dependent predation was executed by predators. This means that predators disproportionately consumed the more abundant prey type, maintaining the coexistence of two prey types (Allen, 1988; Murdoch, 1969; Murdoch et al., 1975). Although only a few predator individuals were used to conduct the experiments, based on the assumption that a population would have an equivalent behavior as a few individuals, it was concluded that a population of such predators in a natural environment would also be able to maintain the diversity of prey species. But this conclusion was seriously questioned by computer simulations of an individual-based model which showed that over a variety of parameter settings, the duration of the coexistence of two prey phenotypes dramatically decreased as the number of predator individuals increased (Merilaita, 2006).

In this study, we conduct simulations in the well-known Tierra system to explore the predation mechanism for maintaining species diversity in an ecological scenario. In Tierra, self-replicating computer programs continuously evolve in a resource-limiting environment (Ray, 1991). This system of Darwinian evolution inside a computer, besides being applied to many evolutionary challenges (Wilke and Adami, 2002), can also be used to study intriguing ecological problems when we set all the mutation rates to zero. With fast generation times (on the order of seconds) and precise measurements, the ecological processes in Tierra can be accurately repeated and thoroughly examined under various parameter settings. Therefore, the Tierra system provides an alternative but powerful experimental method to explore the general principles in ecology.

In order to investigate the maintenance of species diversity by positive frequency-dependent predation, we first design a digital predator which is able to capture multiple prey and acquire energy (CPU time) from them. Then we evaluate our design by comparing the dynamic relationship between the prey and predator populations in Tierra with that in nature. The simulation results show that a cyclic oscillation, similar to the "Lotka-Volterra" cycle (a fundamental pattern displayed by natural prey and predator populations), robustly appears in Tierra. Next, we apply a set of simple rules to specify the behavior of digital predators as they encounter different prey types and verify that the predation in Tierra is essentially the same as positive frequency-dependent behavior exhibited by real predators in laboratory experiments (Merilaita, 2006). Then we allow each digital predator to search for prey in its neighboring area and perform predation based on local prey abundance. We then explore the conditions under which the

presence of a predator population supports the coexistence of two different types of prey. This mechanism of positive frequency-dependent predation for the persistence of species diversity is further examined as we increase the number of prey species from two to three.

Methods

The predator is 100 instructions long and shares the same basic structures of self-examination, reproduction loop and copy procedure as the ancestral creature in the original Tierra implementation (Ray, 1991). However, the predator has an additional predation loop inserted before reproduction. This loop is used to search for multiple prey in the predator's local area. If the predation template in a prey is complementary to the one in the predator and that prey has not been eaten by other predators yet, the predator eats that prey, that is, a certain percent of the prey's CPU time is delivered to the predator and the prey's CPU time is reduced to a small amount. In Tierra, each digital creature is a self-replicating computer program whose execution requires CPU time. Therefore, the survival and reproduction of a digital creature depend on the amount of CPU time that the creature possesses, similar to the energy requirement for the survival and reproduction of an organic creature in nature. After the predator acquires energy (CPU time) from its prey, it finds a space for its daughter and enters the copy procedure for replication. Following the release of its mature daughter, the predator enters the predation loop again to accumulate more energy for future reproduction. This loop of predation and

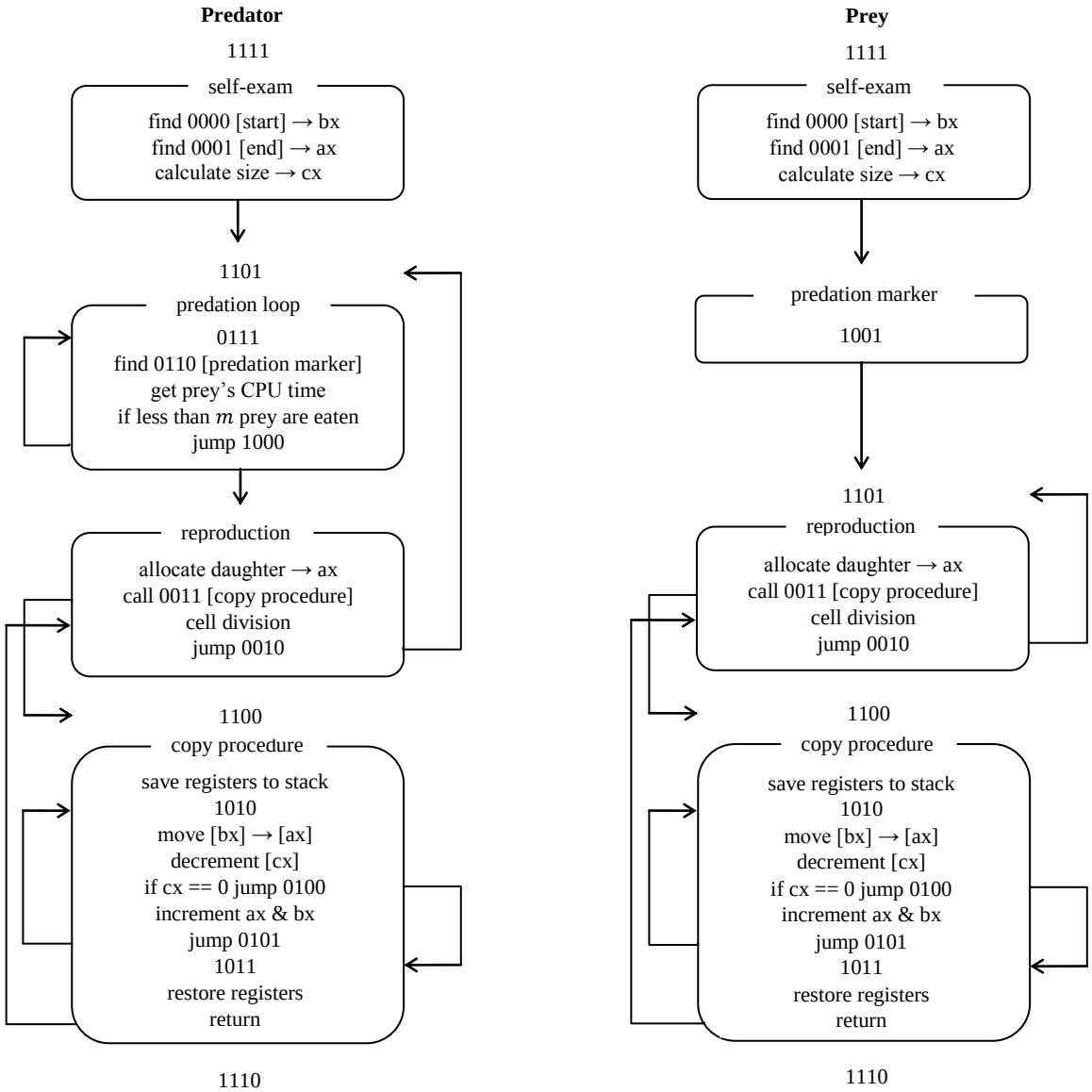


FIGURE 1 Algorithmic flow chart for the predator and prey in the Tierra system. The predation template in the predator (0110) is complementary to the one in the prey (1001), which allows the predator to catch the prey and acquire CPU time from it.

then reproduction repeats until death (Figure 1). We also design two types of prey which are the same as the ancestral creature in the original Tierra system except for the predation template before the reproduction loop (Figure 1). The two prey types differ only in their genome lengths and the predator can detect both of them by matching the predation marker. A type-A prey with the length of 86 instructions reproduces faster than a type-B prey with the length of 96 instructions. The Tierra system assigns a standard amount of CPU time to each prey, but a predator receives only a very small amount of CPU time from the system which supports the predator to execute its first predation loop to try to capture a prey. If the predator fails to capture a prey, it does not have CPU time to execute more instructions. Therefore, predators have to catch prey to obtain energy for survival and reproduction.

The dynamics of the interactions of the predators and prey are examined in ecological simulations, in which Tierra is run without mutation. We seed the soup with 300 predator individuals evenly distributed among 3000 individuals of type-A prey. Each predator is allowed to search for prey in its local area, about 10 creatures long on either side of the predator. In each predation loop, a predator can eat at most m (m = 6) prey and it receives 15% of CPU time from each prey. The amount of CPU time of a captured prey is reduced to 15% of its original value. In a simulation run, we use the number of instructions that have been executed to measure the passage of time. The runs in this experiment last until 1000 million instructions have been executed. Then we use exactly the same parameter settings, except replacing type-A prey with 3000 individuals of type-B prey, to explore the relationship between the predator and type-B prey populations. To confirm that the dynamic pattern between the predator and its prey population results from the predation, rather than random fluctuations in the Tierra system, we design a type-A* prey which shares the same genome length as a type-A prey. Because each prey receives, on the average, the same amount of CPU time from the system, the two prey types with the same length theoretically have the same reproduction rate and thus their population sizes should be maintained at a constant level. Therefore, the variations of the population sizes of type-A and type-A* prey reflect the randomness in the system. We seed the soup with 300 individuals of type-A* prey evenly distributed among 3000 individuals of type-A prey and run the simulation until 1000 million instructions have been executed. Then we compare the population dynamics between type-A and type-A* prey with those between type-A prey and predators.

To investigate positive frequency-dependent behavior of a predator population, we apply the following rules to each predator as it encounters two types of prey in its neighborhood.

(1) Initially, each predator is assigned an equal probability to eat type-A and type-B prey when encountered, that is $P_A = P_B = 0.5$

(2) If the predator eats a type-A prey, its probability to eat type-A prey is increased by ΔP and to eat type-B prey is decreased by ΔP, that is,

$$P_A = P_A + \Delta P \qquad P_B = P_B - \Delta P$$

(3) If, instead, the predator eats a type-B prey, its probability to eat type-A prey is decreased by ΔP and to eat type-B prey is increased by ΔP, that is,

$$P_A = P_A - \Delta P \qquad P_B = P_B + \Delta P$$

(4) All eating probabilities are bounded by P_{min} and P_{max}, that is,

$$0 \le P_{min} \le P_A, P_B \le P_{max} \le 1$$

The simulation results reported in this paper are obtained when $\Delta P = 0.1, P_{min} = 0$ and $P_{max} = 1$, if not otherwise mentioned.

In a laboratory experiment, positive frequency-dependent behavior of a predator is revealed by computing the percentage of one type of prey in the predator's diet as the percentage of that prey type in environment increases from 0 to 100%. In our simulations, the behavior of a predator population in which each predator obeys the above predation rules is examined through the following setup: we run nine separate simulations and in each simulation, we seed the soup with 3000 prey individuals and 300 predator individuals. In each predation loop, a predator can eat at most m (m = 4) prey and acquire 35% of CPU time from each prey and the CPU time of a captured prey is reduced to 40% of its original value. The only difference among the nine simulations is the proportion of two prey types, that is, the percentage of type-A prey in the 3000 prey individuals increases from 10% to 90% in 10% increments. Ideally, we should calculate the percentage of type-A prey in the predators' diet while the ratio of type-A in environment remains constant. However, in our simulations, as the predators start to consume different prey types, the proportion of two prey types changes. We allow the predators to explore the prey populations sufficiently but not to appreciably modify the ratio between type-A and type-B populations. Typically, when the percentage of type-A prey differs from its initial value by 5%, we calculate the percentage of type-A prey in the predators' diet. For example, one of the simulations starts with 600 individuals of type-A prey evenly distributed among 2400 individuals of type-B prey, that is, the percentage of type-A in the 3000 prey individuals is 20%. When type-A prey increase to 25%, we calculate the percentage of type-A prey in the predators' diet (the number of type-A prey that have been eaten is divided by the total number of prey that have been eaten by the predator population).

The maintenance of prey diversity by predators is explored by comparing the results of two simulations. In the control run, we seed the soup with a type-A population of 1500 individuals and a type-B population of 1500 individuals and observe the dynamics of those two prey populations in the absence of predators. The simulation run stops when one of the prey types goes extinct. In the experimental run, we introduce a predator population of 300 individuals into the two initial prey populations used in the control run. Each predator searches for prey in its neighboring area and executes positive frequency-dependent predation based on the type of prey actually captured. In each predation loop, a predator is allowed to eat at most m (m = 4) prey and acquires 35% of CPU time from each prey. The CPU time of a captured prey is reduced to 40% of its original value. The simulation run lasts until 1800 million instructions have been executed and we record the population sizes of the predator and two prey species during the run.

To explore the robustness of positive frequency-dependent predation in maintaining the coexistence of type-A and type-B populations, we systematically vary the two parameters which affect the predation behavior of a predator, the adjustment rate

ΔP and the adjustment range $P_{min} - P_{max}$, and the initial proportion of two prey types, respectively. The default setting of those three parameters is that $\Delta P = 0.1$, $P_{min} - P_{max} = 0 - 1$ and the percentage of type-A prey in the 3000 prey individuals is 50% (1500 individuals of each prey type) and when one parameter is varied, the other two remain unchanged. We set ΔP = 0, 0.005, 0.01, 0.015, 0.02, 0.025, 0.05, 0.1 and 0.2, respectively, to examine the effect of ΔP on the maintenance of prey diversity. Then we set ΔP back to 0.1 and gradually shrink the adjustment range, $P_{min} - P_{max} = 0 - 1, 0.1 - 0.9, 0.2 - 0.8, 0.3 - 0.7, 0.4 - 0.6, 0.5 - 0.5$. Finally, after set $P_{min} - P_{max}$ back to $0 - 1$, we increase the percentage of type-A prey in the 3000 prey individuals from 10% to 90% in 10% increments. For each parameter setting, we record the duration (the number of instructions that have been executed) that the two prey types coexist.

To further examine the role of positive frequency-dependent predation in maintaining species diversity, we add one more species, type-C prey with a length of 90 instructions. Except that the initial prey populations in the control and experimental runs are type-A, type-B and type-C populations of 1000 individuals each, we use the same procedure and parameter settings as those used in the above case of two prey species. We compare the dynamics of prey populations in the absence of predators with those in the presence of predators.

Results

Lotka-Volterra-like Cycle between Digital Prey and Predator Populations

In a natural environment, in order to survive and reproduce, predators have to catch prey and acquire energy from them. This energy transfer from prey to predators leads to the famous "Lotka-Volterra" cycle: an abundant prey population provides more food for predators and thus supports a larger predator population. But as the number of predators increases,

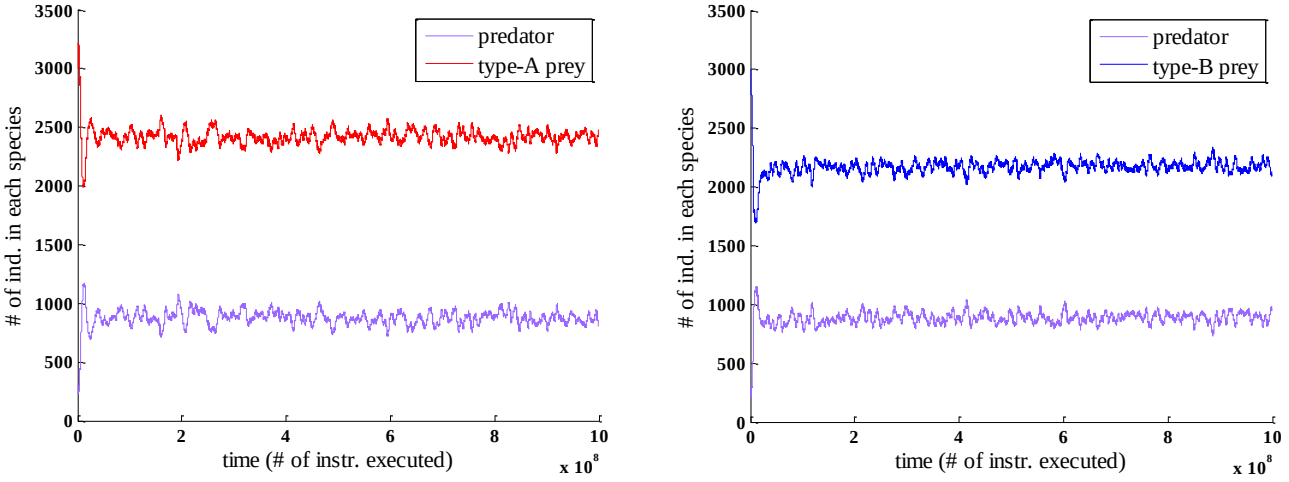


FIGURE 2 Coexistence of a predator population and a prey population in the Tierra system (a) The predator and type-A prey populations stably coexist. (b) The predator and type-B prey populations stably coexist.

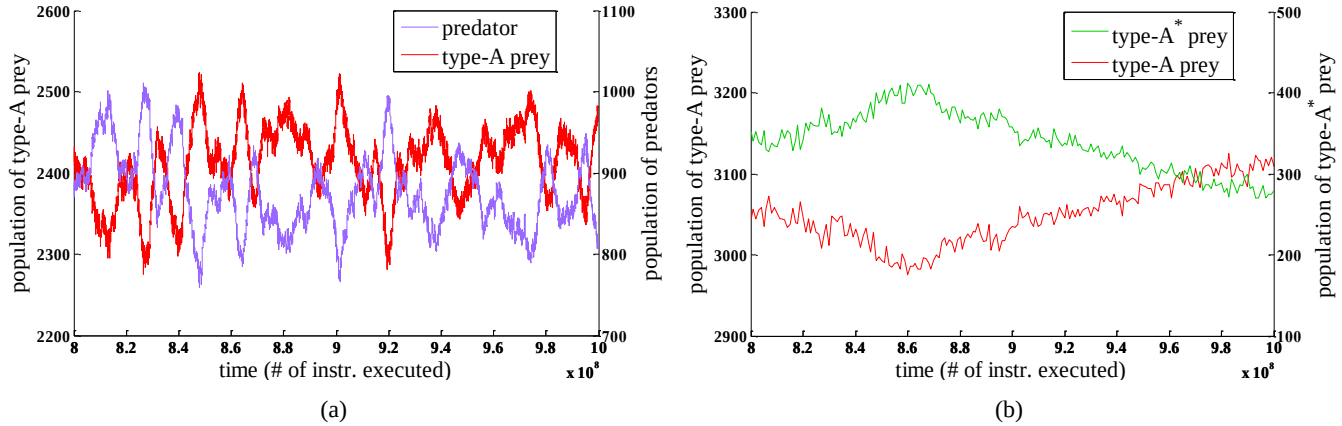


FIGURE 3 (a) "Lotka-Volterra-like" cycle between the predator population and type-A prey population at the steady state from 800 to 1000 million instructions executed in the Tierra system. (b) Population sizes of two prey species with the same genome length slowly drift from 800 to 1000 million instructions executed in the Tierra system.

the growing predation pressure depresses the prey population. When less prey are available, the predator population decreases which reduces the predation pressure and leads to the rebound of the prey population. In Tierra, each digital prey receives a certain amount of CPU time from the system but a digital predator, similar to its counterpart in nature, acquires energy only through predation. When a digital predator searches for multiple prey in its neighboring area and obtains a small amount of CPU time from each prey, the "Lotka-Volterra-like" cycle between the prey and predator populations forms. As shown in Figure 2(a), after the transient initial stage, the type-A prey population rapidly reaches a constant level of about 2400 individuals and stably coexists with the predator population of about 900 individuals. As we examine the population dynamics at the steady state between 800 and 1000 million instructions executed, as shown in Figure 3(a), we find that following the increase of type-A prey population, the predator population increases, which ceases the expansion of the prey population and causes it to decline. Likewise, the decrease of the prey population causes the predator population to decrease, which leads to the rebound of the prey population. In contrast, the population dynamics caused by the randomness in the Tierra system exhibit a completely different pattern. As shown in Figure 3(b), between 800 and 1000 million instructions executed, the population sizes of type-A and type-A* prey species slowly drift without visible cycling. Therefore, the coupled cyclic oscillation between the prey and predator populations in Figure 3(a) is not the result of random fluctuations in the system, but rather results from the energy dependence of the predators on their prey, the very critical component which supports the "Lotka-Volterra" cycle in nature. Similarly, in Figure 2(b), the type-B prey population of about 2200 individuals steadily coexists with the predator population through the establishment of the "Lotka-Volterra-like" cycle. Moreover, as we vary the number of prey that a predator can eat in each predation loop in the range of 3 to 6($m = 3, 4, 5, 6$) and adjust the amount of CPU time transferred from a prey to its predator in the range of 15% to 35%, the "Lotka-Volterra-like" cycle robustly appears in Tierra. This suggests that our design of digital prey and predators may capture some essential properties of predation which allow the creatures in Tierra to follow the same fundamental relationship between prey and predator populations observed in nature.

Positive Frequency-dependent Behavior of Predators at a Population Level

Positive frequency-dependent predation means that the predation risk of a prey individual correlates positively with the frequency of that prey type in environment. That is, a predator is more likely to eat the common prey type than the rare one. In Tierra, each predator has a higher probability of eating a previously encountered prey type, as specified by the rules in the "Methods" section. As shown in Figure 4, when the percentage of type-A prey in the environment is less than 50%, the predator population disproportionately eats less type-A prey and when type-A becomes the abundant prey type (>50%), the predator population disproportionately consumes more type-A prey. The switch of the preferable prey type occurs exactly when the type-A prey change from a rare type to a common one (50%). Therefore, although each digital predator exhibits prey preferences based on the prey types actually encountered, which may not agree with the relative frequency of prey types at a global scale, the predator population executes almost perfect positive frequency-dependent predation on the prey populations.

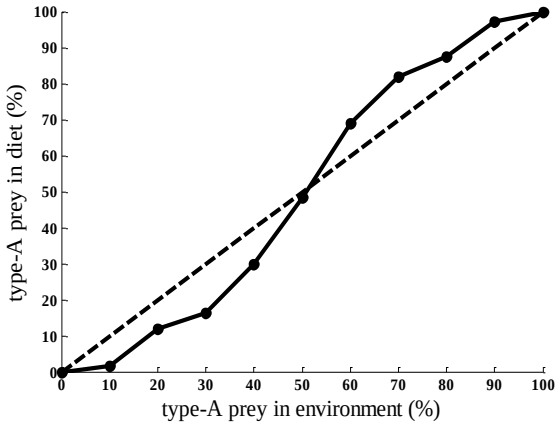
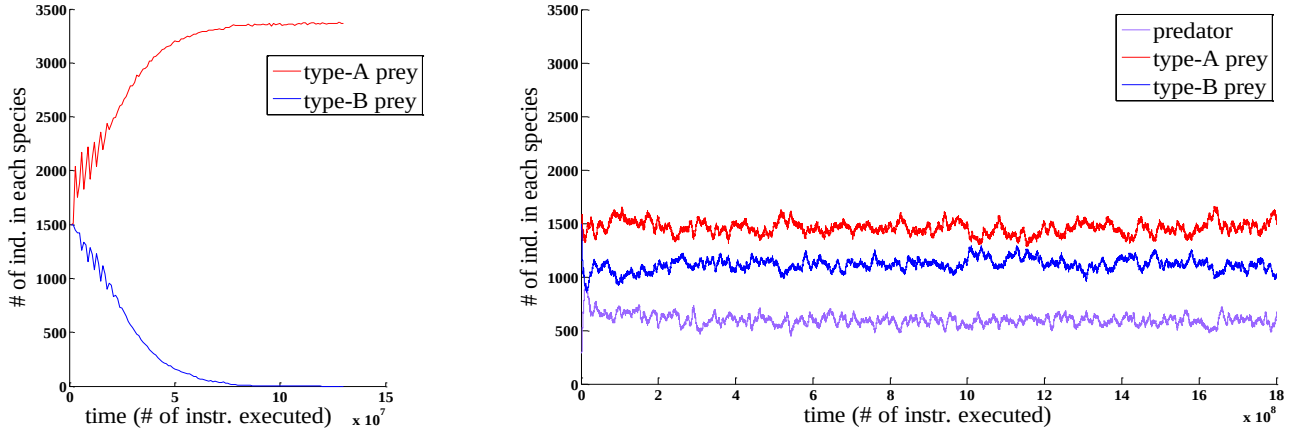


FIGURE 4 A predator population in the Tierra system exhibits positive frequency-dependent behavior. The dashed line indicates the hypothetical situation in which the relative frequency of a prey type in the environment does not affect the predators' eating preference.

Maintenance of Two Prey Species by Positive Frequency-dependent Predation

Many field experiments showed that in the absence of predators, two prey species which shared the same limiting resource could not coexist indefinitely. The more competitive prey species would gradually occupy more and more resources and drive the less competitive prey species to go extinct (Gause, 1934; MacArthur, 1958). This competitive exclusion is also observed in Tierra when type-A prey compete with type-B prey in the environment with limiting CPU time and space. Because a type-A prey (86 instructions long) is shorter than a type-B prey (96 instructions long), when both prey types receive, on the average, the same amount of CPU time from the system, type-A prey reproduce more offspring than type-B prey do. Therefore, although the two types of prey start with the same population size of 1500 individuals, the more rapid replicating type-A prey gradually crowd out type-B prey and drive them to go extinct after 120 million instructions have been executed, as shown in Figure 5(a).

However, after a predator population of 300 individuals which exhibits positive frequency-dependent behavior is introduced into the two prey populations of 1500 individuals of each type, the dynamics of the prey populations change dramatically. As shown in Figure 5(b), after the transient initial stage, the predator population reaches a steady level of about 600 individuals and the two prey populations stably coexist with approximately 1500 individuals of type-A and 1100 individuals of type-B. The stable population size of each prey type indicates that the diversity of prey species may persist forever under positive frequency-dependent predation.

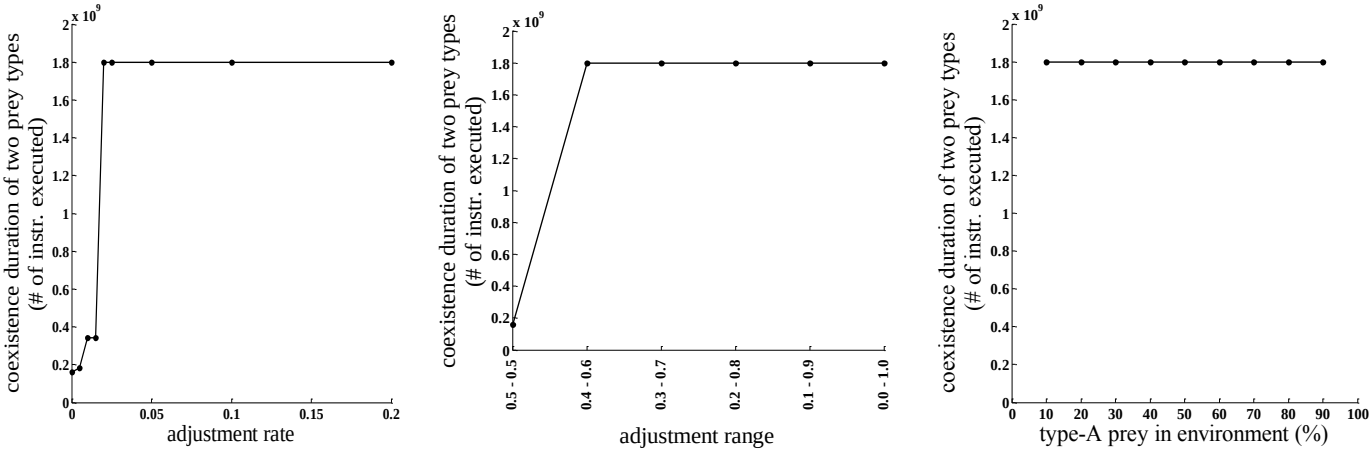


(a) (b)

FIGURE 5 Coexistence of two prey species is maintained by a predator population with positive frequency-dependent behavior (a) Competitive exclusion between two types of prey; type-B prey go extinct. (b) Type-A and type-B prey stably coexist under the predation from a predator population.

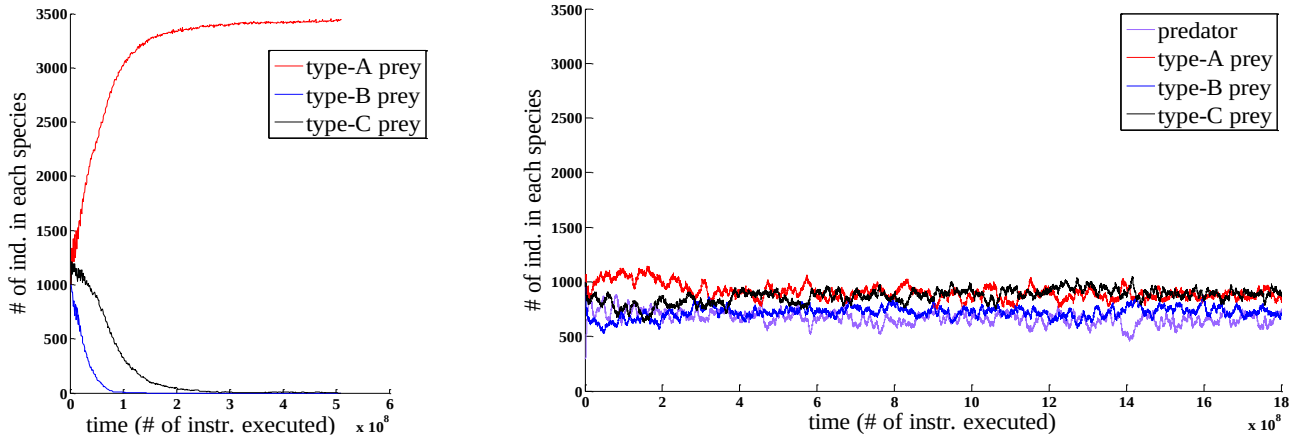
Robustness of Frequency-dependent Predation on Maintaining the Coexistence of Two Prey Species

The adjustment rate ΔP directly affects the strength of positive frequency-dependent predation. When $\Delta P = 0$, a predator always has the same probability, $P_A = P_B = 0.5$, to eat type-A and type-B prey regardless of the abundance of those two prey types in its local area. As ΔP increases, a predator can more effectively adjust its probability of eating different types of prey based on the prey it actually captures. As shown in Figure 6(a), when $\Delta P \geq 0.02$, the predator population has sufficient frequency-dependent behavior to maintain the coexistence of the two prey populations over the entire simulation run of 1800 million instructions executed. The adjustment range $P_{min} - P_{max}$ specifies the lower and upper boundaries of the eating probability, which indirectly limits a predator's ability to prefer the more abundant prey type. For example, when $P_{min} - P_{max} = 0.5 - 0.5$, a predator's probabilities to consume different prey types are fixed at $P_A = P_B = 0.5$, that is, a predator fails to adjust its eating probabilities based on local prey abundance even if $\Delta P = 0.1$. However, this limitation is gradually relaxed as the adjustment range extends towards $P_{min} - P_{max} = 0 - 1$. As shown in Figure 6(b), except for $P_{min} - P_{max} = 0.5 - 0.5$, which eliminates the effect of positive frequency-dependent predation, the two prey populations coexist under all other adjustment ranges over the simulation run of 1800 million instructions executed. By disproportionately consuming more abundant prey type, positive frequency-dependent predation can maintain the coexistence of two prey types even when the initial sizes of the two prey populations vary dramatically. As shown in Figure 6(c), when the percentage of type-A prey in



(a) (b) (c)

FIGURE 6 Robustness of positive frequency-dependent predation in maintaining the coexistence of two prey types (a) When the adjustment range is $0 - 1$ and the percentage of type-A prey in the environment is 50%, type-A and type-B prey populations stably coexist as $\Delta P \geq 0.02$. (b) When $\Delta P = 0.1$ and the percentage of type-A prey in the environment is 50%, type-A and type-B prey populations stably coexist under all the adjustment ranges except for $0.5 - 0.5$. (c) When $\Delta P = 0.1$ and the adjustment range is $0 - 1$, type-A and type-B prey populations stably coexist at nine different initial ratios of the two prey populations.



(a) (b)

FIGURE 7 Coexistence of three prey species is maintained by a predator population with positive frequency-dependent behavior (a) Competitive exclusion among three types of prey; type-B prey and then type-C prey go extinct. (b) Type-A, type-B and type-C prey stably coexist under the predation from a predator population.

the 3000 prey individuals increases from 10% (300 individuals of type-A prey and 2700 individuals of type-B prey) to 90% (2700 individuals of type-A prey and 300 individuals of type-B prey), the two prey types coexist under each of the nine initial ratios of the two prey populations over the simulation run of 1800 million instructions executed. Those simulation results suggest that positive frequency-dependent predation may robustly support the coexistence of two prey species.

Maintenance of Three Prey Species by Positive Frequency-dependent Predation

We increase the number of prey species by adding one more species, type-C prey which is 90 instructions long. In the absence of predators, three prey types compete with one another and the creatures with a shorter genome length reproduce faster than those with a longer genome length as each creature receives approximately the same amount of CPU time from the system. When the simulation run starts with 1000 individuals of each prey type, due to competitive exclusion, type-B prey go extinct after 144 million instructions have been executed and then type-C prey are crowded out by type-A prey after 504 million instructions have been executed, as shown in Figure 7(a). However, after a predator population of 300 individuals is introduced into the three prey populations of 1000 individuals of each type, as shown in Figure 7(b), all three prey types stably coexist. This result further supports the idea that positive frequency-dependent predation is able to maintain the diversity of prey species.

Discussion

In the original Tierra implementation, a form of predation emerged through evolution of hyper-parasites which were able to reproduce themselves and steal additional CPU energy from parasites to enhance their reproduction rate (Ray, 1991). Because the survival of hyper-parasites did not rely on the existence of parasites, the predation relationship between hyper-parasites and parasites may not be consistent with that between organic predator and prey populations. In nature, when a prey is caught by a predator, only a small amount of energy is transferred to the predator. A predator has to catch multiple prey in order to acquire sufficient energy. Similar to its counterpart in nature, a predator in Tierra catches multiple prey in its local area and obtains a small amount of energy from each prey. The simulation results show that the "Lotka-Volterra-like" cycle robustly appears in Tierra over a wide range of parameter settings which suggests that the digital predators and prey may be suitable for exploring predator-prey population dynamics.

Positive frequency-dependent predation is one of the proposed mechanisms for maintaining species diversity in nature (Gendron, 1987). It has been supported by several laboratory experiments in which one or a few predators that constantly consumed the more common prey type were able to maintain the coexistence of two prey phenotypes (Allen, 1988). But in a natural environment, it is likely to be a full predator population, rather than a few predator individuals, to regulate prey populations. In the paper (Merilaita, 2006), the author used an individual-based model to explore the dynamics of positive frequency-dependent predation at a population level with one predator species and two prey species. The simulation results showed that although one or two predator individuals could maintain the diversity of prey species, which was consistent with the laboratory experiment results, five or ten predator individuals failed to do so. Because the duration that two prey species coexisted decreased dramatically as the number of predator individuals increased, it was concluded that positive frequency-dependent predation may not be a sufficient mechanism to maintain species diversity in nature. However, the setup of the simulations in the paper (Merilaita, 2006) may not agree with the natural behavior of a predator population. In the laboratory experiment with one or two predator individuals, each predator was able to explore the entire populations of two prey types and switched to the common type based on the

global abundance of different types. The author in the paper (Merilaita, 2006) also allowed each predator to obtain prey from the entire prey populations regardless of the number of predator individuals. It was found that a single predator individual maintained prey species diversity longer than ten predator individuals. This result was rationalized as follows: "when there were ten predators, the behavior of each individual predator was formed by only one tenth of the information about prey type frequencies in relation to the total number of consumed prey, compared to the one-predator case." (Merilaita, 2006) Because each predator in the ten-predator case lacked global information on prey type frequencies, those ten predators could not maintain prey diversity as efficiently and accurately as a single predator individual. But in a natural environment, a predator individual can neither access the entire prey populations nor acquire complete information about them. Rather, each predator searches for prey only in its local area and switches to the common type based on the local prey abundance which may not be consistent with the frequency of the prey types at the global scale. This feature of local predation is elegantly executed in the Tierra system where a predator searches for prey in the range of 10 creatures on either side. Our simulation results show that when each predator in Tierra, similar to its organic counterpart, implements positive frequency-dependent predation based on the prey type actually encountered and does not have any information about the entire prey populations, a population of 600 predator individuals maintains the coexistence of two prey types. This emergent global pattern of species coexistence from the local interactions between prey and predators is robust to the variations of the parameters that affect either the predation behavior of predators or the initial proportion of the two prey types in the environment. Furthermore, as we increase the number of prey types from two to three, the predator population also successfully maintains the coexistence of three prey species. Therefore, our results strongly suggest that positive frequency-dependent predation may be a reasonable mechanism to maintain species diversity in nature.

The simulation results we report here are obtained under an ecological scenario in which all mutations are blocked. Our future research will explore the hypothesis that positive frequency-dependent predation may facilitate the increase and maintenance of species diversity in an evolutionary scenario. It is a more complex but more intriguing situation: when various types of random mutations are introduced into the Tierra system, the genomes of digital creatures will be modified and thus new types of prey and predator species will continuously emerge. Therefore, unlike the ecological scenario in which the prey types are known and the number of prey types is fixed, in the evolutionary scenario the prey types that can be detected by predators change over time. In the original Tierra system, when one or a few successful species emerged through mutation, they usually gained reproductive advantages either by effectively exploiting other creatures or by shortening their own lengths and rapidly crowded out other existing species. Thus, the soup was repetitively dominated by very few species. However, with the introduction of positive frequency-dependent predation, the dominant prey species may be depressed by predators. This may provide resources to support the populations of other prey species and thus more prey species may have the opportunities to evolve. With this increase in the number of coexisting prey species, more food sources may be available to predators which may promote the differentiation of predator species, with each specializing on a certain type of prey. Moreover, in order to produce more offspring, new prey species may evolve novel escape strategies to avoid being eaten and new predator species may develop innovative predation tactics to acquire more energy from prey. Therefore the co-evolution between prey and predator species may be observed in the Tierra system. Additionally, the introduction of predation may elongate an evolutionary process in Tierra. One of the causes of the cessation of evolution in the original Tierra system was that ecological interactions only emerged when selection favored smaller genomes (when all creatures received equal amounts of CPU time). Selection favoring smaller genomes eventually led to stasis when genomes reduced their sizes as much as possible, and no significant genetic variants were possible. Predation is a mechanism of allowing ecological interactions in the absence of selection for smaller genomes, and thus may allow evolution to continue longer.

References

Allen, J. A. (1988). Frequency-dependent selection by predators. *Philosophical Transactions of the Royal Society B*, 319: 485-503.

Gause, G. F. (1934). *The struggle for existence*, Williams & Wilkins, Baltimore, MD.

Gendron, R. P. (1987). Models and mechanisms of frequency-dependent predation. *The American Naturalist*, 130: 603-623.

MacArthur, R. H. (1958). Population ecology of some warblers of northeastern coniferous forests. *Ecology*, 39: 599-619.

Merilaita, S. (2006). Frequency-dependent predation and maintenance of prey polymorphism. *Journal of Evolutionary Biology*, 19: 2022-2030.

Morin, P. J. (1981). Predatory salamanders reverse the outcome of competition among three species of anuran tadpoles. *Science*, 212: 1284-1286.

Murdoch, W. W. (1969). Switching in general predators: experiments on predator specificity and stability of prey populations. *Ecological Monographs*, 39: 335-354.

Murdoch, W. W., Avery, S., and Smyth, M. E. B. (1975). Switching in predatory fish. *Ecology*, 56: 1094-1105.

Paine, R. T. (1974). Intertidal community structure: experimental studies on the relationship between a dominant competitor and its principal predator. *Oecologia*, 15: 93-120.

Ray, T. S. (1991). An approach to the synthesis of life. In Langton, C. *et al*., editors, *Artificial Life II*, pages 371-408. Addison-Wesley publishing company, Redwood City, CA.

Stanley, S. M. (1973). An ecological theory for the sudden origin of multicellular life in the late Precambrian. *Proceedings of the National Academy of Sciences*, 70: 1486-1489.

Wilke, C. O. and Adami, C. (2002). The biology of digital organisms. *Trends in Ecology & Evolution*, 17: 528-532.

Patch-level Selection in Darwinian Daisyworld

Matthew Bardeen

Universidad de Talca, Talca, Chile
mbardeen@utalca.cl

Abstract

Scientists have used Richard Dawkins' ideas of the extended phenotype to postulate levels of selection higher than an individual in evolution. Dawkins rejects this extension and insists that there must be a reproductive bottleneck for the extended phenotype, and thus, higher levels of selection to exist. In this research, a model is presented that shows levels of selection higher than the individual, without the reproductive bottleneck insisted upon by Dawkins. A 2-dimensional cellular automata Daisyworld model is extended with a gene that controls the rate of albedo mutation. A large number of runs of the model are performed with a variety of different parameters, and the statistics for the runs are analyzed. The results show that contrary to expectations, the mutation rate does not stay low but instead rises to high levels. The reasons for this are analyzed and it is shown that patch level selection pressures are acting upon the individuals. It is concluded that selection pressures higher than the individual can exist, mimicking the extended phenotype, without the need for a reproductive bottleneck.

Introduction

The existence of multiple levels of selection in evolution has been under much debate (Okasha, 2007; Sober and Wilson, 1999). Traditionally many biologists believed that selection could operate on a group of individuals of one species. The justification for this belief was the apparent willingness of one individual to put itself in danger for the good of the group. However, in 1964 Hamilton published two articles showing this behavior could be explained by a process called *kin selection*, where individuals aid relatives based on the probability of having shared genetic code (Hamilton, 1964a,b). Based on this work and others (Trivers, 1971), Dawkins (1976) postulated the existence of the selfish gene, describing a view where the gene is the *unit* of selection. Genes, he argued, are inherently selfish – favoring behaviors that serve to help them reproduce. A gene that inspired its carrier to commit suicide before reproduction, for example, would not survive very long in the gene pool. Genes together in the body of an individual are forced to work together by virtue of having to pass through the same reproduction event, and are the *vehicle* of selection. Dawkins' "vehicle of selection" is analogous to the "level of selection" used by other authors (Okasha, 2007), a phrasing I will use in this paper.

Dawkins (1982) later recognized that some genes have influences outside their bodies, a concept which he called the *extended phenotype*. Here genes in one individual can be tied to genes existing in other bodies by way of environmental modifications – the classic example is the beaver dam, where the genes for building & maintaining dams enhance the survival of the immediate organism and others within its colony. The genes still remain as the unit of selection, but, in Dawkins' terms, the group becomes the vehicle of selection. Recently there has been debate on how far these effects extend beyond the organism and under what conditions (Biernaskie and Tyerman, 2005; Dawkins, 2004; Laland, 2004; Jablonka, 2004; Turner, 2004; Whitham et al., 2003, 2005). In particular, Dawkins (2004) insisted that there must be a single reproductive event (a bottleneck) for all the genes involved in the extended phenotype to force the genes to work together.

Swenson et al. (2000) showed that it is possible for real ecosystems to respond to artificial selection. They theorized that such selection could happen in the natural world, suggesting that small scale "microecosystems" could be selected upon given the differential survival of such systems. Also, they noted that discrete boundaries are not necessary for an ecosystem to be a level of selection. The key is "localized interactions, such that one patch fares better than another on the basis of its properties, even when the boundaries between patches are fuzzy" (Swenson et al., 2000). Penn and Harvey (2004) showed a similar response to artificial selection in non-evolving artificial ecosystems.

In this paper I use a cellular automata Daisyworld model to show the existence of patch-level selection and I demonstrate that it arises from the transfer of heat across the planet. I introduce a heritable albedo mutation rate to the daisy genotype and show that although its variation cannot be seen on the individual level, it is subject to selection pressure. This is because variations in the albedo mutation rate can be seen by looking at groups of individuals in a larger popula-

tion and because these groups compete among themselves for space.

To give a brief view of how this paper is organized, in the following section I describe the basic ideas of the Daisyworld model. I describe the model in more mathematical terms in the Model Description section, and then describe the experiments and show the results in graphical and numerical forms in the Results section. The Discussion section deals with the explanations and implications of the results and is followed by the concluding remarks of the paper.

Methodology

Watson and Lovelock (1983) presented the Daisyworld model to address some of the more prevalent doubts about the Gaia theory. This model has also proved useful in studying evolution under the assumption that organisms affect their own environment (Dyke et al., 2007). Recently the idea of niche construction, where organisms exert influence on environment has gained prominence in the discussion of evolutionary theory (Odling-Smee et al., 2003). While this influence on environment has been acknowledged previously (Dawkins, 1976), the implications for evolutionary theory have not been obvious (Bardeen, 2009).

Daisyworld was a toy-world, intended as a proof of concept of the Gaia hypothesis, rather than a model of a real physical system. The idea behind it was simple – localized interactions can affect global dynamics and generate homeostatic behavior. The model consisted of a "planet", heated by the sun and populated by black and white daisies. The black daisies have a lower albedo (reflectiveness) than the white daisies, and they absorb a greater amount of solar radiation and raise the local temperature. The growth rate of the daisies is linked to the local temperature, which is directly influenced by albedo. This difference in growth rate causes the area covered by black and white daisies to vary, causing the overall temperature of the planet to vary in turn. This creates a homeostatic response to external forces, such as increasing incoming solar radiation (insolation) and keeping the temperature of the planet relatively constant. This process is mainly due to the niche construction aspects of the individual daisies on their local environment.

I use a variant of the 2D cellular automata Daisyworld model first described by von Bloh and Schellnhuber (1999). The growth patterns of the daisies are given by a cellular automata model. There is heat transfer between neighboring cells, so a daisy can affect its local neighborhood. This model is useful in that all the effects seen are, by definition, local. Any global effects that are seen must be emergent properties of local interactions.

Another attraction of this model is the ability to "tune" the diffusion rate, which permits experimentation with how quickly and strongly effects of local daisies are transmitted to their neighbors, and by extension, the global environment. This will allow me to quantify the probability that group-level selection will arise in the system based on the diffusion rate of heat across the planet.

To this model I add a gene that affects the mutation rate of the daisy albedo. This gene will not affect the fitness of individual daisies immediately, but will allow the effects of a selection pressure at levels higher than an individual daisy. There is biological evidence of different mutation rates between species, and even evidence of differential mutation rates on the same genome (Wolfe et al., 1989), so this extension is not pure fantasy. When asked once about mutation rates in natural systems, the eminent biologist John Maynard-Smith replied that he expected them to be set as low as possible (Bedau and Seymour, 1994; Maynard Smith, 1989). The reasoning is that, according to Travis and Travis (2004): "..in constant environments, most mutations are deleterious, hence mutation occurs at a low rate, constrained only by the costs of error avoidance and error repair".

The expected role of evolution by natural selection is that of optimization and adaptation, and this should be no different in the context of the Daisyworld (Ackland et al., 2003; Ackland, 2004; Bardeen, 2009; Stöcker, 1995).

Model Description

The base model for this article is an extended version of the Daisyworld model described in von Bloh et al. (1997).

The temperature field $T(x, y, t)$ is represented by the energy balance equation:

$$C\frac{\partial T(x,y,t)}{\partial t} = D_T(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2})T(x,y,t) \\ -\sigma_B T(x,y,t)^4 + S(1 - A(x,y,t)), \quad (1)$$

where D_T is the heat diffusion constant and $A(x, y, t)$ is the space/time distribution of albedo. The diffusion uses the von Neumann neighborhood (the four adjacent neighbors to the cell). S is the current solar radiation, and at $S = 917$ the albedo which produces the optimal temperature for daisy growth is around 0.53.

Growth patterns for the daisies are generated using a cellular automata (CA) model. If a cell is empty, then there is a chance that a daisy in a neighboring cell (Moore neighborhood) will produce offspring in the empty cell. This chance is based upon the temperature of that cell and given by

$$\beta(T) = \frac{4}{(T_{max} - T_{min})^2}(T - T_{min})(T_{max} - T) \quad (2)$$

where T_{max} and T_{min} are the maximum and minimum temperatures at which the daisies can grow and T is the current temperature of the cell. T_{opt} is equivalent to $\frac{1}{2}(T_{min} + T_{max})$. In this paper $T_{max} = 313$ Kelvin and $T_{min} = 278$ Kelvin, meaning $T_{opt} = 295.5$ Kelvin.

The chance a daisy will die is given by:

$$\gamma(T) = 1 - \frac{4\rho}{(T_{max} - T_{min})^2}(T - T_{min})(T_{max} - T) \quad (3)$$

where $\rho \in [0, 1]$ and serves set the base mortality rate. If ρ is large, then the base mortality rate will be low.

Each daisy genotype consists of two floating-point values. The first is the color of the daisies albedo, which is in the range of $0 : 1$ inclusive and has a 50% chance of being changed at birth by adding a random value to the parent albedo, drawn from a Gaussian distribution with a standard deviation of r. If the parent albedo plus mutation falls outside the range of the albedo, the mutation is redrawn from the Gaussian distribution.

The second value, r, is essentially the mutation rate of the albedo, which is also in the range of $0 : 1$ inclusive and is also mutated at birth by adding a random value drawn from a Gaussian distribution with a standard deviation of 0.001 (the mutation rate of the albedo mutation rate). If the parent mutation rate plus the delta falls outside the range of the albedo, r is redrawn from the same Gaussian distribution.

Results

The principal set of experiments in this chapter are designed to test the long-term stable solution of the Daisyworld with a heritable albedo mutation rate. To this end, a 200^2 cell world is populated randomly (the chance of daisy in a given cell is 10%) with daisies having uniform albedos of 0.53 (near optimal for the starting insolation) and initial albedo mutation rates of 0.01. This world is allowed to evolve for one million timesteps with a constant incoming solar radiation ($S = 917$), at which point the simulation is stopped. This process forms one evolutionary run. Each run is repeated 50 times for each variation in parameter values; Tested are different diffusion constants (D_T = 50, 100, 500, 1000, 1500, 2000, 2400). These experiments will show the adoption of a high albedo mutation rate by most of the daisies under high diffusion regimes.

Figure 1(a) shows the average planetary temperature over two separate runs. In one, the planetary temperature oscillates closely around the optimum for life on the planet. In the other, the average planetary temperature climbs past the optimum. The evolutionary trajectory of the mutation rate shows the reason for this difference (Figure 1(b)). In the first run the mutation rate stays low, as is expected, while in the second the mutation rate climbs past 0.2. As the mutation rate climbs, the average albedo of the planet drops (seen in Figure 1(c)). This has the effect of increasing the average mortality rate and decreasing the average birth rate of the daisies.

A closer look (Figure 2) reveals that the mutation rate is not uniform over the entire planet, but rather is limited to patches of daisies.

Figure 3 shows that the average final mutation rate is influenced both by the diffusion rate of heat between the cells and the base mortality rate. Higher diffusion results in a

Figure 2: *Snapshots of world state from one run, Insolation $L = 1.0$, Diffusion $D_T = 1500$, Mortality rate is 10% and the mortality model is the variable mortality model. Black is lower mutation, white is higher mutation, blue is dead. World is mostly dominated by low mutation daisies with well defined patches of high mutation daisies*

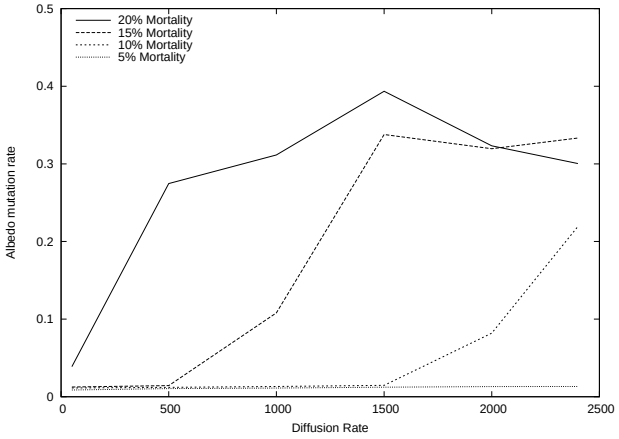
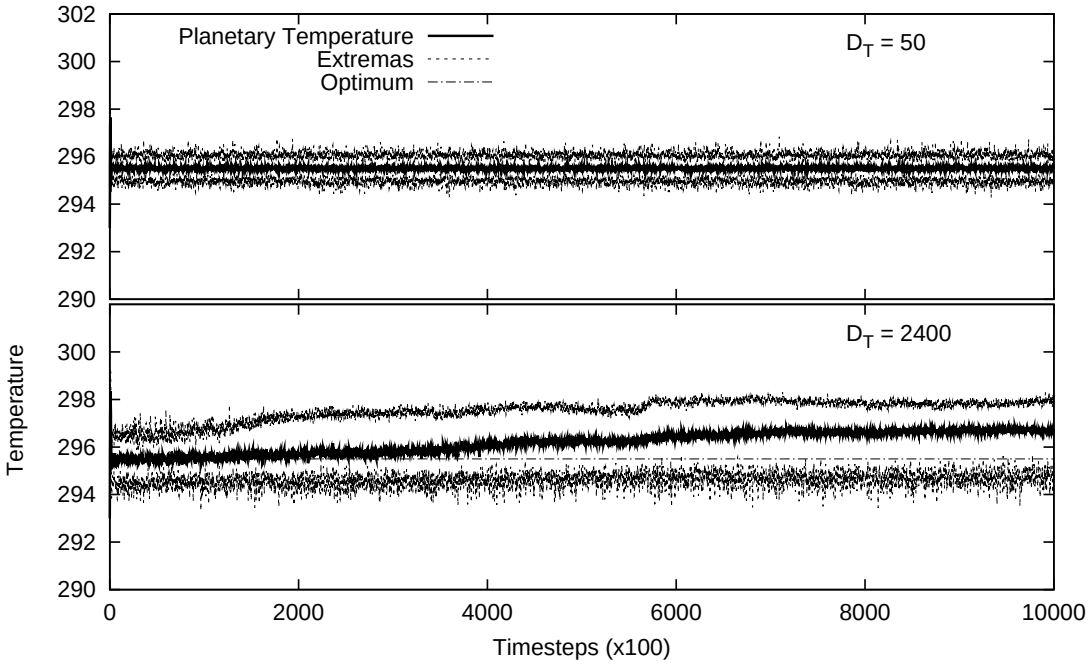
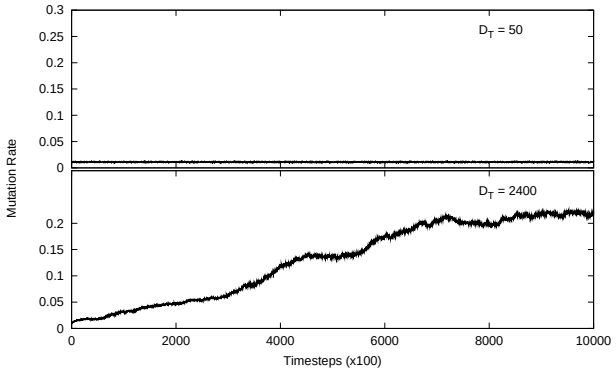


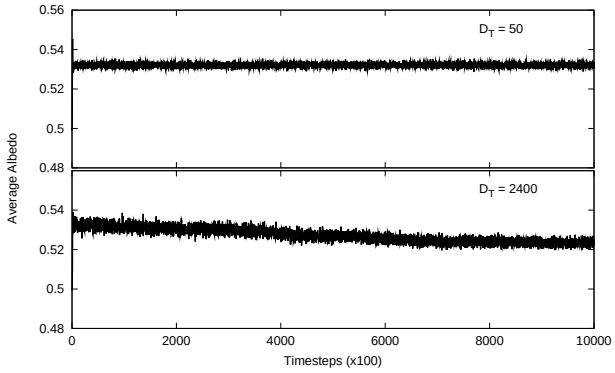
Figure 3: Average final mutation rate of 50 runs, each consisting of 2 million timesteps.



(a) *Steady state temperatures with a variable albedo mutation rate.*



(b) *Average albedo mutation rate.*



(c) *Average albedo.*

Figure 1: *Evolution of planetary average temperature, average albedo mutation rate, and average albedo for 2 runs over 1 million timesteps. Mortality rate of 10%, Grid size is* 200^2.

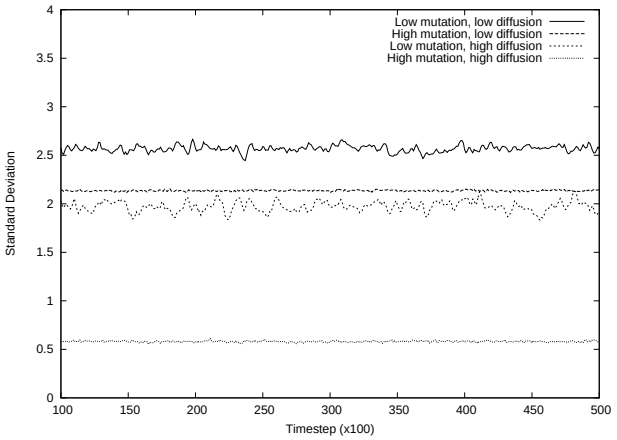


Figure 4: Standard deviation of temperature across the cells of the planet. Under high diffusion, the high mutation rate daisies present a much more uniform environment than the low mutation rate daisies. Under low diffusion, the temperature is more uniform for the high mutation rate daisies than the low mutation rate daisies, but not to the same extent. (High Diffusion $D_T = 2400$, Low diffusion $D_T = 50$, Base mortality rate = 15%, High mutation = 0.4, Low mutation = 0.01)

Table 1: Average number of children over 25 runs, Low mutation = 0.01, High mutation = 0.4

	High Diffusion $D_T = 2400$	Low Diffusion $D_T = 50$
Low Mutation	0.424	0.420
High Mutation	0.417	0.410

higher final mutation rate, as does a higher base mortality rate.

Inspecting the average number of children for runs with fixed low (0.01) and high (0.4) albedo mutation rates shows that there is little difference between the average number of children for both (Table 1). The major difference between the two strategies is that the temperature across the planet has less variance under the high mutation daisies than under the low mutation daisies (Figure 4). Under low diffusion rates the difference between the variance in temperature caused by the two strategies is much less than under high diffusion rates.

Discussion

The results of the experiments leave some questions:

- What is the cause of the increase in temperature?
- What are the implications of a high mutation rate?
- Why would a high mutation rate be a selective advantage under certain circumstances and not under others?
- Is the selective advantage caused by an individual level selection pressure or a higher level selection pressure?

In this section, I will answer these questions in turn, then discuss the wider implications of the answers.

What is the cause of the increase in temperature?

The plots of temperature and the mutation rate (Figures 1(a) and 1(b)) show that the increase in temperature is linked with that of mutation rate, however it does not reveal the cause. Inspecting the snapshot of the planet state shows that, unsurprisingly, the albedos are very diverse when the mutation rate is high.

In low diffusion environments, the heat is mainly retained within a single daisy cell and there is little transfer to other cells. In high diffusion environments the heat flows freely across the cells, and a group of daisies with random albedos will appear, at a higher level, to have the temperature of a single gray daisy with an albedo of around 0.5. Thus, as more daisies adopt the high mutation rate strategy, the average albedo of the planet becomes closer to 0.5 and the temperature rises away from the optimum.

What are the effects of a high mutation rate?

A high mutation rate causes a number of changes to the system. Comparing two planets, one with a fixed high mutation rate and one with a fixed low mutation rate, shows that the high mutation rate planet has a lower average growth rate, a higher average death rate, and lower number average number of children per daisy. Another notable difference is that the standard deviation of the cell temperature across the planet is lower on the high mutation planet.

The high mutation rate also affects the heredity of the daisy albedo. Lewontin (1978) gives the necessary conditions of natural selection as: individuals within a species differ, this variation is heritable, different variants leave differing amounts of offspring, and variations that favor an individual's reproductive success will be preserved. In the systems with a high albedo mutation rate, the selection of individual daisies seems to fail on the second of these principles – the variation in albedo does not seem heritable from parent to offspring.

Furthermore, it is unclear how the albedo mutation rate is being selected upon. Identifying the variation between the albedos of two individuals is easy. Identifying the variation between the albedo mutation rate of two individuals is much harder. The only way to measure this variation would be to look at the range of variability in the albedos of the respective offspring. However, with only an average of 0.4 offspring per parent, the quality of this measure for natural selection is limited. Thus the variation must be seen either above the level of the individual or over a large period of time.

Why would a high mutation rate be a selective advantage?

The previous subsection highlights a crucial question – why would a high albedo mutation rate be a selective advantage for the daisies bearing it, but only under certain circumstances? From the fact that the high mutation rate daisies are only seen primarily under high diffusion rate worlds, we can discard the idea that there is an unintended systemic effect from, for example, bias in the mutation operator (Bullock, 2001). If there was such a systemic effect, it would be seen in all parameter ranges, and not only under certain conditions.

This leaves non-systemic causes to blame. Figure 2 shows the existence of patches of daisies with similar mutation rates. With low mutation rates, patches of daisies with similar albedos are likewise seen. This is due to the cellular automata rules, since a new daisy can only be born next to another living daisy, they tend to clump into patches of similar daisies.

This phenomenon is a hindrance when the albedos are near identical – patches that have albedos higher or lower than the optimal are inherently unstable. They become too hot or too cold and die off, replaced by daisies bearing albedos which are more suitable to the changed environment. When they die, not only is their albedo gene lost, but their albedo mutation rate gene is lost too.

Having patches of daisies with highly variable albedos means that the patch temperature stays relatively constant, though not optimal. Less environmental change in the daisy patch signifies less change is needed by the genome. In this case, the gene for albedo mutation rate "uses" the albedo gene as a buffer between it and the environment. Its reproductive environment becomes more stable and, as a result, the high albedo mutation rate gene lasts longer – unchanged – within the gene pool.

The experiments with fixed mutation rates support this conclusion – the variability of the temperature on the planet populated with a fixed high mutation rate is much less than that of a planet with fixed low mutation rate daisies (Figure 4). It can be assumed that this same phenomena is seen on a smaller scale within the patches.

Is the selective advantage an individual level or a higher level selection pressure?

Now the question becomes on what level is the selective advantage operating – is it an individual level pressure or some pressure operating on a higher level? Williams (1966) gives the following guide: "Do these processes show an effective design for maximizing the number of descendants of the individual, or do they show an effective design for maximizing the number, rate of growth, or numerical stability of the population or larger system?".

The unit of selection here is surely an individual daisy. Daisies do not reproduce at the same time, nor do they share genetic information with one another. However, the level at which the selection pressure is operating is not clear.

If it was an individual level pressure, we would expect to see the maximization of birth rate, the minimization of the death rate, or a higher number of children born per individual. For this to happen, they should be at their optimal albedo, since that will maximize their chances of producing offspring and minimize their chances of dying. Likewise, the mutation rate should be very low. As seen, this is indeed the case under low diffusion environments.

However under high diffusion rates, we see the average mutation rate start to rise, for the reasons discussed prior.

Patches

However, the previous explanation leaves a conundrum: If having highly variable albedos is a such a wise strategy, then why do patches of high and low mutation daisies appear on the planet at the same time, as seen in Figure 2? Why don't all the daisies convert to high mutation rates?

The answer is that patches of daisies compete among themselves – those that are more successful at maintaining the high mutation rate gene have slower growth rates, but higher gene stability. Thus the incidence of daisies with high albedo mutation rates tends to increase within the population. But low mutation rate daisies with near optimal albedos occasionally find purchase with their higher reproductive rates and lower death rates, creating patches of their own.

Thus there is competition between the two strategies on the basis of their effect on the environment. Individual daisies are linked to others by means of their geographical vicinity. When the diffusion is high, those links are stronger than when it is low. In low diffusion environments, daisies with high albedo mutation rates are not competitive with those that have low mutation rates.

Frank (1996) pointed out that in parasitism, we often see selection between kin groups at high levels, but competition between individuals in lower levels. He says that "In the population of parasites within the host, a mutant parasite with a faster growth rate will usually increase in frequency." However if this growth rate causes host death before transmission to neighboring hosts, the effective long-term fitness of the mutant is non-existent. Thus there is a balance between exploitation of resources (individual level selection) and cooperation (kin group selection).

This is essentially what is happening in the Daisyworld model described here – competition between kin groups leads to cooperation within the group in some cases. This immediately calls to mind the example most used for the extended phenotype – beaver dams. Beaver dams are typically shared between kin groups. Beaver families that build better dams in more advantageous places are more successful than those that do not. The shared phenotype in this example is the dam. But can the beaver kin group be thought of as an

organism or vehicle of selection?

Central to the idea of the organism in the extended phenotype was the insistence that there be a reproductive bottleneck (Dawkins, 1982, 2004) to force cooperation. This insistence can be seen again in Frank (1996). In the beaver example, new dams are often created by a single breeding pair – a reproductive bottleneck.

However, here the daisies reproduce in a random fashion - there is no bottleneck. So what forces the cooperation between the daisies and causes high mutation rates? It can be nothing more than the shared environment of the daisies. The high diffusion rate links the fate of one daisy to the fate of its neighbors, forcing cooperation. This is why high mutation rates are only seen in high diffusion rate environments.

Furthermore. this is why the extended phenotype is not limited by the reproductive bottleneck described by Dawkins. If there is a tight enough coupling between different organisms, such that the the fate of one is linked to the fate of another, they will evolve as a group, rather than individuals. Further work is necessary to quantify how strong the linkage needs to be and under what conditions this linkage can come about.

Conclusion

The results of this study can be generalized relatively easily – the "mutation rate" here really refers to the rate of phenotypic change in the daisies in comparison to the change in environment. The diffusion rate is analogous to the impact an individual has on its neighbors and competitors. With high diffusion rates, the influence of individual daisies on their local temperature is minimal. The variation part of evolution as seen from the planetary perspective is no longer one daisy, but clumps of daisies, since that is where most of the phenotypic variation lies. Conversely when the diffusion rate is low, the focus of evolution is on individual daisies, since the selection method (birth rate/death rate) is very much dependent on the individual daisy phenotype.

This idea has important consequences in evolution. One can imagine how natural selection works on all levels simultaneously. Micro-organisms (like soil fungi) would be subject to individual level selection at their own level since their effects are more immediate and diffuse slowly in comparison to their reproductive speed. From higher levels (i.e., from a forest level) they could be selected upon as groups since their effects appear to diffuse rapidly in relation to the reproductive speed of other organisms at the higher level.

In this work I have demonstrated the existence of patch-level selection upon individuals in a model world. Necessary conditions for this development were: a spatial structure, the modification of local environment by individuals, the transmission of local effects to neighboring organisms, and a gene that controls the rate of change in the phenotypic property that modifies the local environment. These necessary conditions can be found, without great difficulty, in nature.

Furthermore, it shows that the reproductive bottleneck in Dawkins ideas of the extended phenotype is more strict than it needs to be. All that is really needed is the existence of some force linking the fate of the genes in one organism to the fate of genes in another. And if this is the case, then the arguments presented by Laland (2004), Turner (2004), and Whitham et al. (2003) for the extension of the extended phenotype do indeed hold merit.

Acknowledgements

I would like to thank Narciso Cerpa and the three anonymous reviewers for their comments and suggestions – they have resulted in a vast improvement in the arguments of the paper.

References

Ackland, G. (2004). Maximization principles and Daisyworld. *Journal of Theoretical Biology*, 227:121–128.

Ackland, G., Clark, M., and Lenton, T. (2003). Catastrophic desert formation in Daisyworld. *Journal of Theoretical Biology*, 223:39–44.

Bardeen, M. (2009). *Lessons from Daisyworld: Survival of the Stable*. Dphil, University of Sussex.

Bedau, M. A. and Seymour, R. (1994). Adaptation of Mutation Rates in a Simple Model of Evolution. In Stonier, R. J. and Yu, X. H., editors, *Complex Systems: Mechanism of Adaptation*, pages 37–44. IOS Press, Amsterdam.

Biernaskie, J. and Tyerman, J. (2005). The overextended phenotype. *Écoscience*, 12:3–4.

Bullock, S. (2001). *Smooth Operator? Understanding and Visualising Mutation Bias*, volume 2159 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg.

Dawkins, R. (1976). *The Selfish Gene*. Oxford University Press.

Dawkins, R. (1982). *The Extended Phenotype*. Oxford University Press.

Dawkins, R. (2004). Extended Phenotype - But Not Too Extended. A Reply to Laland, Turner, Jablonka. *Biology and Philosophy*, 19:377–396.

Dyke, J., McDonald-Gibson, J., Di Paolo, E., and Harvey, I. (2007). Increasing complexity can increase stability in a self-regulating ecosystem. *Lecture Notes in Computer Science*, 4648:133.

Frank, S. A. (1996). Models of Parasite Virulence. *The Quarterly Review of Biology*, 71(1):37.

Hamilton, W. D. (1964a). The genetical evolution of social behaviour. I. *Journal of Theoretical Biology*, 7(1):1–16.

Hamilton, W. D. (1964b). The genetical evolution of social behaviour. II. *Journal of Theoretical Biology*, 7(1):17–52.

Jablonka, E. (2004). From Replicators to Heritably Varying Phenotypic Traits: The Extended Phenotype Revisited. *Biology & Philosophy*, 19(3):353–375.

Laland, K. N. (2004). Extending the Extended Phenotype. *Biology & Philosophy*, 19(3):313–325.

Lewontin, R. C. (1978). Adaptation. *Scientific American*, 239:212–228.

Maynard Smith, J. (1989). The Limitations of Evolutionary Theory. In Maynard Smith, J., editor, *Did Darwin Get It Right?*, pages 180–191. Chapman and Hall, New York.

Odling-Smee, J., Laland, K., and Feldman, M. (2003). *Niche Construction: The Neglected Process in Evolution*. Princeton University Press.

Okasha, S. (2007). *Evolution and the Levels of Selection*. Oxford University Press, New York, NY, USA.

Penn, A. and Harvey, I. (2004). The Role of Non-Genetic Change in the Heritability, Variation, and Response to Selection of Artificially Selected Ecosystems. In Pollack, J., Bedau, M., Husbands, P., Ikegami, T., and Watson, R. A., editors, *Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems, Alife IX*. MIT Press.

Sober, E. and Wilson, D. S. (1999). *Unto others: The evolution and psychology of unselfish behavior*. Harvard University Press, Harvard, MA, first pape edition.

Stöcker, S. (1995). Regarding Mutations in Daisyworld Models. *Journal of Theoretical Biology*, 175:495–501.

Swenson, W., Wilson, D. S., and Elias, R. (2000). Artificial ecosystem selection. *Proceedings of the National Academy of Sciences of the United States of America*, 97(16):9110–4.

Travis, E. R. and Travis, J. M. J. (2004). Mutators in space: the dynamics of high-mutability clones in a two-patch model. *Genetics*, 167(1):513–22.

Trivers, R. (1971). The evolution of reciprocal altruism. *The Quarterly Review of Biology*, 46(1):35.

Turner, J. S. (2004). Extended Phenotypes and Extended Organisms. *Biology and Philosophy*, (Dawkins 1982):327–352.

von Bloh, W., Block, A., and Schellnhuber, H. (1997). Self stabilization of the biosphere under global change: A tutorial geophysiological approach. *Tellus*, 49B:249–262.

von Bloh, W. and Schellnhuber, H. (1999). Tutorial Modelling of geosphere-biosphere interactions: the effect of percolation-type habitat fragmentation. *Physica A*, 266:186–196.

Watson, A. and Lovelock, J. (1983). Biological homeostasis of the global environment: The parable of Daisyworld. *Tellus*, 35B:286–289.

Whitham, T., Lonsdorf, E., Schweitzer, J., Bailey, J., Fischer, D., Shuster, S., Lindroth, R., Hart, S., Allan, G., Gehring, C., Keim, P., Potts, B., Marks, J., Rehill, B., DiFazio, S., LeRoy, C., Wimp, G., and Woolbright, S. (2005). "All effects of gene on the world": Extended phenotypes, feedbacks, and multi-level selection. *Écoscience*, 12:5–7.

Whitham, T., Young, W., Martinsen G. Gehring, C., Schweitzer, J., Shuster, S., Wimp, G., Fischer, D., Bailey, J., Lindroth, R., Woolbright, S., and Kuske, C. (2003). Community and Ecosystem Genetics: A Consequence of the Extended Phenotype. *Ecology*, 84:559–573.

Williams, G. C. (1966). *Adaptation and Natural Selection*. Princeton: Princeton University Press.

Wolfe, K., Sharp, P., and Li, W. (1989). Mutation rates differ among regions of the mammalian genome. *Nature*, 337:283–285.

The Daisystat: A Model to Explore Multidimensional Homeostasis

James Dyke

Max Planck Institute for Biogeochemistry, 07745 Jena, Germany
jdyke@bgc-jena.mgp.de

Abstract

The Homeostat was a physical device that demonstrated Ashby's notion of 'ultrastability'. The components interact in such a way as to maintain sets of essential variables to within critical ranges in the face of an externally imposed regime of perturbations. The Daisystat model is presented that bears a number of similarities to Ashby's Homeostat but which can also be considered as a higher dimensional version of the Watson & Lovelock Daisyworld model that sought to explain how homeostasis operating at the planetary scale may arise in the absence of foresight or planning. The Daisystat model features a population of diverse individuals that affect and are affected by the environment in different ways. The Daisystat model extends Daisyworld in that homeostasis is observed with systems comprised of four environmental variables and beyond. It is shown that the behaviour of the population is analogous to the 'uniselector' in the Homeostat in that rapid changes in the population allows the system to 'search' for stable states. This allows the system to find and recover homeostatic states in the face of externally applied perturbations. It is proposed that the Daisystat may afford insights into the evolution of increasingly complex systems such as the Earth system.

Introduction

This paper introduces a new model that demonstrates homeostasis in the face of external perturbations: the Daisystat. The Daisystat is a hybrid of 'Daisyworld' and 'Homeostat' as it shares salient features with both models. The Daisyworld model (Lovelock (1983); Watson and Lovelock (1983)) was initially intended as a cybernetic proof of concept for planetary homeostasis as formulated in Gaia Theory which proposed that the Earth system (where 'Earth system' is defined as the Earth's atmosphere, oceans, cryosphere, lithosphere and biota) was a homeostatic entity that maintained conditions to within the range that allowed widespread life (Lovelock, 1979). The Homeostat was a physical device that exhibited ultrastability - the ability to respond to a particular regime of perturbations in such ways as to maintain certain essential variables to within essential ranges (Ashby, 1960). While the spatial and temporal scales of Daisyworld and the Homeostat are very different (Daisyworld considers self-regulation at a planetary scale over aeons whereas the Homeostat was built from four decommissioned Royal Air Force bomb aiming devices and operated at millisecond speed) both systems exhibit very similar behaviour that can be observed in the Daisystat.

In the following sections, the Homeostat and Daisyworld models will be described. The Daisystat is then presented and two sets of results shown. The first set shows how a single-environmental-variable-Daisystat responds to a progressive driving perturbation, the second set shows how a four environmental variable Daisystat responds to instantaneous shocking perturbations. The establishment and maintenance of homeostasis in both cases is given in terms of 'rein control'. It will be shown that the behaviour of the population is analogous to the behaviour of the electromechanical Homeostat in that the volume of possible connections between elements of the system is 'searched' until new feedback values are found that produce homeostatic states. Such a process is the result of natural selection operating on a population of diverse individuals. No notions of higher level selection, altruism or kin selection are required to explain the homeostatic behaviour of the system. The 'law of requisite variety' (Ashby, 1956) is seen operating in the Daisystat in that there are lower bounds for the amount of genetic and phenotypic diversity in the population in order for homeostasis to be established and maintained. It is proposed that the Daisystat can be used as a tool to explore the evolution and emergence of real world complex systems such as the Earth system.

The Homeostat

The Homeostat was an electromechanical device designed and constructed by W. R. Ashby. The Homeostat consisted of four units. Each unit produced an output that was fed into the inputs of the other units and back to itself via a recurrent connection. Fig. 1 shows a schematic of the Homeostat units and their connections. The inputs into the ith unit, I_i, are the sum of the outputs of the other units multiplied by a set of input weights:

$$I_i = \sum_{j=1}^{j=4} O_j \omega_{j,i} \quad (1)$$

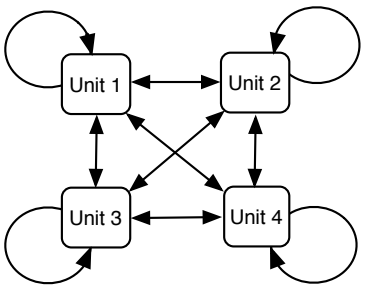


Figure 1: Schematic of the Homeostat and the connections between units. Double arrow headed lines represent the two connections that link two units. Each unit has an output connection to the other three units and one recurrent connection to itself.

where $\omega_{j,i}$ is the weight for the connection from the jth unit to the ith unit. A weight can either increase or decrease a connection input. Each unit has a target value, T. The unit's output, O, is the difference between the input and target value: $O = T - I$. This represents the first level of homeostatic control in the Homeostat. The second level of control is derived from the establishment of essential ranges for the output of the units. If the output of a unit moves outside of the essential range, then a uniselector component randomly generates connection weights for that unit until the unit output moves back within the essential range. For example, if the essential range is [-0.5,0.5] and $O = 0.6$ then the uniselector would generate new weights for all connections into that unit until the output moves back within the essential range. The Homeostat demonstrated ultrastability that was a consequence of Ashby's law of requisite variety. In order for the Homeostat to maintain stable states in the face of perturbations, it must be able to reconfigure itself in at least as many ways as these perturbations demand. Consequently, the volume of possible connection weight values must encompass all possible values that would be required to produce stable states.

Homeostat simulations start by having the uniselectors for each unit create random weights. This produces initially chaotic behaviour whereby one unit drives another unit out of its essential range which responds with new uniselector values which may drive another unit of of its essential range and so on. Given sufficient iterations of the uniselector process, a set of weights will be generated that proves to be stable in that the outputs of all units remain within their essential ranges. An example Homeostat simulation is shown in Fig. 2. The Homeostat finds a stable state and is then perturbed when Time = 200 by decreasing the output of one unit by 1. This leads to all units moving out of their essential range and a period of uniselector activity that creates new random weights which produces a new attractor which the system relaxes towards.

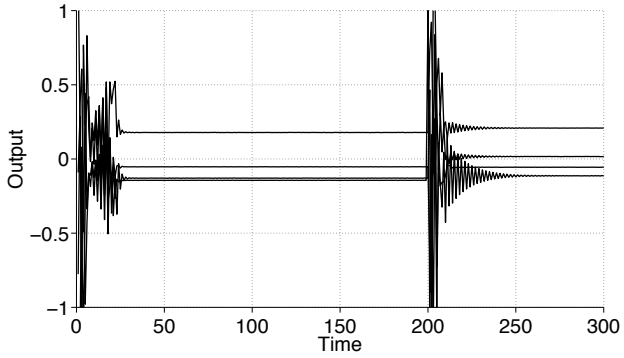


Figure 2: Output of the four Homeostat units. The third unit (second from bottom) is perturbed at Time = 200 by decreasing its output by 1. This drives the unit outside of its essential range of [-0.5,0.5] and actuates the uniselector that creates a new set of random weights. This produces large changes in all other units and actuation of their uniselectors until a new stable state is achieved.

Daisyworld

While Daisyworld is a simple model of a planetary system, it is more complicated than the Homeostat with a number of different feedback mechanisms that feature non-linear functions. However, at its heart it is similar in that two units in the form of two species or type of plants (commonly referred to as 'daisies') exert unidirectional effects on a regulated variable in the form of planetary temperature. These effects stem from the different albedo of the daisies. Albedo is a measure of the reflectivity of an object. Black daisies have lower albedo than white daisies. Changing the relative proportion of black and white daisies will affect the planetary albedo and so the global temperature. The black and white daisies share the same parabolic growth response to temperature. Both grow at maximum rates when their local temperatures are 22.5° Celsius with growth progressively decreasing, until it is zero when the temperature is 5° or 40° Celsius.

Daisyworld simulations consist of seeding a grey planet that has an intermediate albedo of 0.5 with black and white daisy seeds. This planet orbits a star much like the sun which over geological time scales increases in luminosity or brightness. On a lifeless planet, as the star increases in luminosity, the temperature increases approximately linearly (the actual temperature response being a quartic function of luminosity). The situation is markedly different when black and white daisies are present in that the temperature rapidly moves towards the maximum growth rate temperature and then stays within the range that the daisies are able to grow over as luminosity increases. This demonstrates how planetary regulation may emerge as a consequence of biological activity that is not the result of intentional design and

in ways compatible with natural selection. Fig. 3 shows planetary temperature being regulated when both daisies are present and Fig. 4 show how this regulation is the result of the change in the proportional coverage of the black and white daisies.

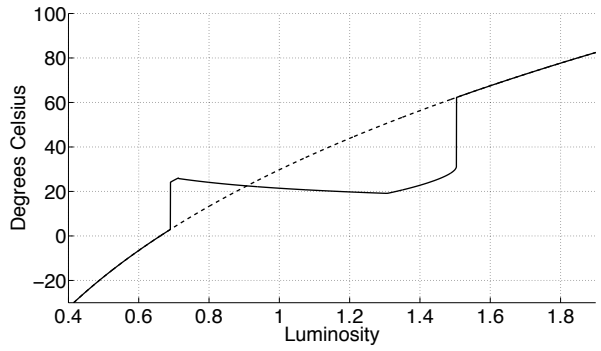


Figure 3: Temperature as a function of luminosity on Daisyworld. The dashed line represent temperature on a planet with no daisies. This increases approximately linearly with increasing luminosity. The solid black line shows planetary temperature with black and white daisies present. This increases suddenly, after which it is maintained within the growing range of the daisies for a range of luminosity values. There is a sudden increase in planetary temperature that corresponds to the collapse of the daisy populations.

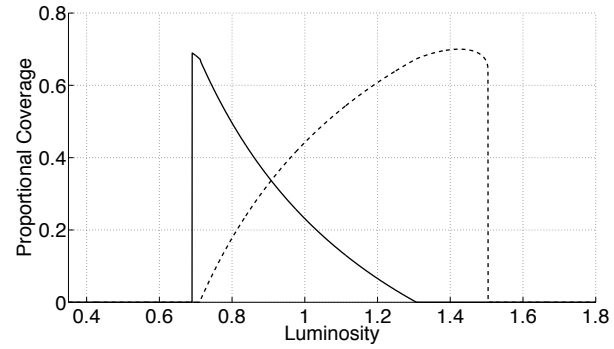


Figure 4: Coverage of black (plotted with solid line) and white daisies (plotted with dashed line) as a function of luminosity on Daisyworld. There is a sudden increase then progressive decline in black daisies that is mirrored by the coverage of the white daisies.

The Daisystat Model

While the original Daisyworld demonstrated that planetary homeostasis was at least conceivable, it was subject to a number of quite limiting assumptions. Some of these have been addressed in the literature. See Wood et al. (2008) for a review. The Daisystat is intended to address one of these more important limitations that was succinctly identified by J. Kirchner:

> "Daisyworld is a one-feedback model; there is only one environmental variable and it is regulated by extremely strong feedback with the simplest possible biosphere. Such a simple model necessarily exhibits simple behaviour. By contrast, on the real Earth many different environmental variables are coupled simultaneously, through many different feedback relationships, with a highly complex biosphere composed of organisms with diverse (and often incompatible) environmental requirements. Such a complex system can exhibit many kinds of behaviour that a simple Daisyworld model cannot." Kirchner (2003)

Daisystat features a number of environmental variables that are regulated so that they remain within essential ranges as a consequence of the effects of a diverse population of individuals that respond to selection pressure in ways that means they only ever 'seek' to increase their own abundance with no selection for their effects on the environmental variables. Daisystat can be understood as a development of an individual-based Daisyworld model first proposed in McDonald-Gibson (2006) and then analysed and extended in: Dyke et al. (2007); McDonald-Gibson et al. (2008); Dyke (2009). There are three important differences between the Daisystat and these previous models. Firstly, as already stated, Daisystat features multiple environmental variables. Secondly, mutation is not currently modelled in the Daisystat so there is no change in the total amount of genetic information in the population over time. Finally there is no single carrying capacity for the population. Previous Daisyworld studies typically assumed that all individuals within a population will be limited to a shared carrying capacity amount. Consequently the rate of change of all individuals is a function of the frequency of all other individuals. In Daisystat this assumption is relaxed in that all individuals have separate carrying capacities. The interaction between two individuals is then mediated only via their dependence on shared environmental variables. A population of K individuals are affected by and in turn affect their environment. In all results shown, unless otherwise specified, $K = 100$. The individuals may represent individual organisms, populations, species or guilds etc. All individuals experience the same environmental conditions in that the environment is homogenous so that there are no local conditions or micro-climates. The effect that any individual has on the environment lead to changes in the environment that all individuals experience in the same way. It is assumed that an individual's effect on this homogenous environment diffuses instantaneously. The term 'environmental resource' is used to denote those aspects or elements of the environment that affect individuals and in turn are affected by individu-

als. It is important to note that such environmental resources do not produce monotonically increasing fitness in individuals. It is possible to 'have too much of a good thing' so an increasing environmental resource can lead to a decrease in the fitness of an individual. This will be expanded on below. The change over time of the ith environmental resource, R_i, is given by:

$$\frac{dR_i}{dt} = \alpha I_i + \beta O_i \tag{2}$$

where I_i is the external perturbing input that is being applied to the ith resource and O_i is the population's effect on the resource which is the sum of the individual's effects:

$$O_i = \sum_{j=1}^{j=K} = E_{i,j} \tag{3}$$

The effect, $E_{i,j}$, that the jth individual has on the ith resource varies over the range [-1,1] and is given with:

$$E_{i,j} = A_j \epsilon_{i,j} \tag{4}$$

where $\epsilon_{i,j}$ is the phenotypic effect which is multiplied by the abundance, A, of the jth individual, where abundance could be interpreted as numbers of individuals, total biomass, frequency in the population, proportional coverage etc. α and β are parameters that determine the relative strengths of the perturbing input and population output. For all the results shown $\alpha = \beta = 1$. There is no momentum in environmental resources, consequently their rate of change will be zero when $\alpha I = -\beta O$. The abundance of the jth individual changes over time with:

$$\frac{dA_j}{dt} = A_j(k_j - A_j)F_j - A_j\gamma \tag{5}$$

where k_j is the carrying capacity of the jth individual. This equation is essentially identical to that used in Watson and Lovelock (1983) and gives logistic growth towards the carrying capacity, k. In all results shown all k values are set to unity. Therefore, the range of possible abundance values is $[0, 1-\gamma]$, where γ is a fixed death rate and for all results shown is fixed at 0.1. F_j is the 'fitness' function for the jth individual and is the sum of the fitness function responses for each environmental resource:

$$F_j = \sum_{i=1}^{i=R_{max}} F_{i,j} \tag{6}$$

where R_{max} is the number of environmental resources and $F_{i,j}$ is a normal distribution response that determines the jth individual's response to the ith environmental resource:

$$F_{i,j} = e^{(-(R_{i,j}-T_{i,j})^2)/2\sigma^2} \tag{7}$$

where $T_{i.j}$ is the 'target' ith resource value for the jth individual in that this is the resource values that gives the maximum fitness of unity. This is analogous to the growth response to temperature in Daisyworld. As the resource increases/decreases from this target value, fitness decreases at a rate determined by the variance, σ^2. For all results shown, σ^2 is set to unity.

Simulations consist of initialising a population of individuals with random ϵ and T values. The method used is to represent each individual as a two loci genome where each locus has a floating point number over the range [0,1]. These values are mapped to the ranges of [0,100] and [-1,1] for the phenotypic traits of T and ϵ respectively. Resource values are initialised at some value over the range [0,100]. The change over time in resources and abundances of individuals are then numerically integrated.

Results

Two sets of results are presented. The first set demonstrates Daisystat's ability to perform Daisyworld-type regulation; a system consisting of a single environmental resource is stabilised at a series of particular values in the presence of a perturbing driving input that would in the absence of the effects of the individuals increase the resource. The second set demonstrates Daisystat's ability to perform Homeostat-type regulation or higher dimensional Daisyworld-type regulation; a system consisting of four environmental resources is subjected to a shock which the population responds to with a period of rapid change until a new stable state is achieved.

Daisyworld-type regulation

Fig. 5 and Fig. 6 show changes in resource and abundances over time for a system that consists of a single resource when $dI/dt = 3/\tau$, where $\tau = 2000$ is the number of units of time simulated. These results show the resource being maintained at a number of values during a simulation. Decreasing the rate of change of the perturbing input will typically lead to homeostatic states in which the resource is held at one value for the duration of the simulation. The perturbing input progressively seeks to drive the resource higher and higher. Fig. 7 shows that the population responds to this driving so as to produce a counteracting force so that there is no change in the resource: $I = -O$. This regulation proves to be robust to a wide range of parameter values. K can be decreased to approximately 20 and its only upper limit is computational resources for numerically integrating the equations (maximum K value simulated is 10,000). The width of the fitness functions which is determined by σ^2 can be decreased or increased by a magnitude with no significant effects. The rate of change of the perturbing input, dI/dt cannot be set arbitrarily high. In the original Daisyworld study it was assumed that the rate of change of the luminosity of the star was sufficiently slow and the change in the population was sufficiently fast so as to keep the luminosity value fixed while the population was integrated to steady state. The Daisystat can significantly relax this assumption, however there must be sufficient time for the population to respond to perturbations by changing the abundances of individuals.

It is important to note that what value the resource remains fixed at is not prescribed in the model. Moreover there appears to be no initial reason why the resource should remain fixed at any level. Natural selection can be seen operating on the population via the different target values that each individual has. Individuals with target values nearer to the current resource level would increase in abundance and their effects on the resource would increase. Such effects range over [-1,1] and are an incidental 'by-product' of the individual in that there is no selection pressure for these effects. As there is selection pressure for an individual's response to the environment but no selection pressure for an individual's effect on the environment, it may appear strange that the population responds to changes in perturbations that affect the environment by changing the effects they have on the environment while keeping their responses fixed. The explanation for this behaviour can be given in terms of 'rein control'.

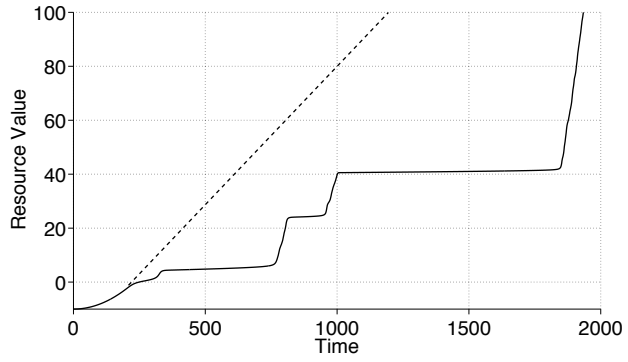


Figure 5: Daisystat with a single environmental resource. The resource is plotted with a solid line. The approximated resource value in the absence of any individuals is plotted with dashed line. This increases as the perturbing input is increased over time whereas the simulation with individuals present shows that the resource initially increases with increasing perturbations but then remains approximately fixed when it enters the range of values that produce non-zero fitness. There are three periods of relatively rapid change in the resource with homeostasis being recovered after the first two periods.

Rein control

The term rein control was coined by M. Clynes in Clynes (1969) within a discussion of unidirectional communication and control in biological organisms. Saunders et al. (1998) and Saunders et al. (2000) developed the notion into a mathematical description of regulatory systems that are comprised of separate 'reins' that can only pull a controlled variable in one direction. The notion of rein control has been previously applied to the analysis of Daisyworld-type models: Harvey (2004), Dyke and Harvey (2006), Dyke et al. (2007), McDonald-Gibson et al. (2008), Wood et al. (2008), Dyke (2009). The Daisystat extends the rein control notion in that homeostatic states feature diverse populations that are not necessarily dominated by two individuals/types/species. Fig. 8 shows the establishment of a rein control stable state. Two sub-populations can be seen in that a group of individuals that have T values lower than the current R value will collectively have an increasing effect on R, while a group of individuals that have T values higher then the current R value will collectively have a decreasing effect on R. The

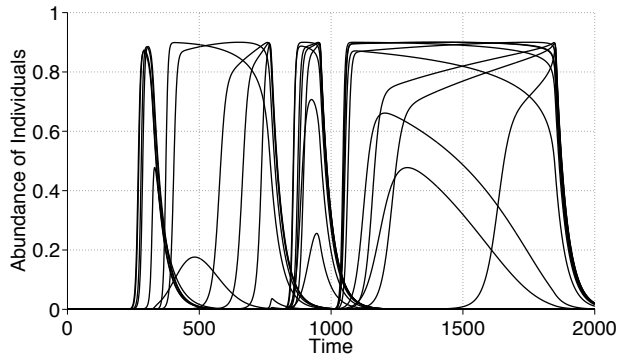


Figure 6: Abundance of individuals changing over time. The change in abundance is analogous to the change in the coverage of black and white daisies in Daisyworld. As the perturbing input seeks to drive the resource higher, the population responds by altering the proportion of increasing and decreasing effect individuals.

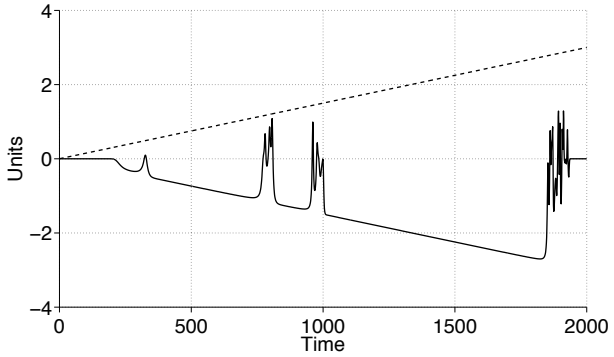


Figure 7: Population output changing over time. The effect that the population has on the resource is plotted with a solid line. The driving perturbing force is plotted with a dashed line. The increasing perturbing input produces an equal magnitude, but opposite sign response from the population. At Time $\approx$ 800 and 900 there are rapid changes in the population output before it is recovered so that $I = -O$ again.

sum of the individual's effect will equal that of the perturbing input, I. As I changes, the abundance of individuals and the net effect of the two sub-populations changes so that $I = -O$ and so R remains fixed.

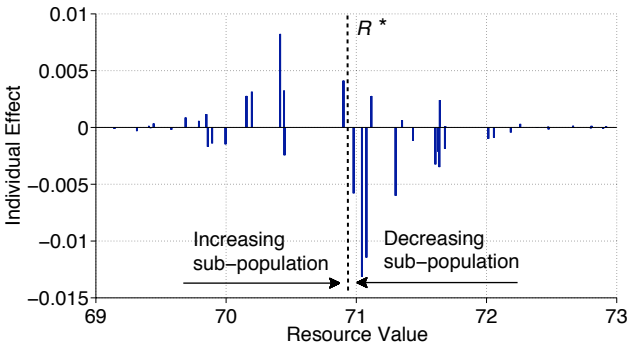


Figure 8: The origins of a rein control stable state when $K = 1000$. The effects that the individuals have on the resource are shown where these effects are the product of the individual's phenotypic effect on the resource, ϵ, and the abundance of that individual, A. Individuals are ranked in order of their T values. Individuals at the left hand side of the horizontal axis have maximum fitness when $R = 69$ while individuals at the right hand side have maximum fitness when R= 73. The resource, R, is being fixed around the value of 70.9 which is denoted by the dashed lined labelled R^*. To the left of the dashed line, the sum of the sub-population effects is positive. To the right of the dashed line, the sum of the sub-population effects is negative. As the perturbing input, I, alters, the population responds so that the relative strengths of the two populations adjust such that $I = -O$ and hence R is maintained near R^*.

Homeostat-type regulation

The Daisystat exhibits Homeostat-type behaviour in response to sudden perturbations. A Daisystat that was comprised of 4 environmental resources was allowed to relax to a stable state in the absence of any perturbations ($I = 0$). This was then subjected to a 'shock' in that one resource value was instantaneously increased by 5 units. This lead to a rapid change in the values of all other resource values as the abundance and so population output on the resources varied rapidly as shown in Fig. 9 and Fig. 10. The change in the abundances continued until a new stable state was found.

Discussion

Daisystat displays the ability to resist external driving perturbations much the same way as the original Daisyworld model. An important difference from the original Daisyworld model is that the effects the individuals have on their

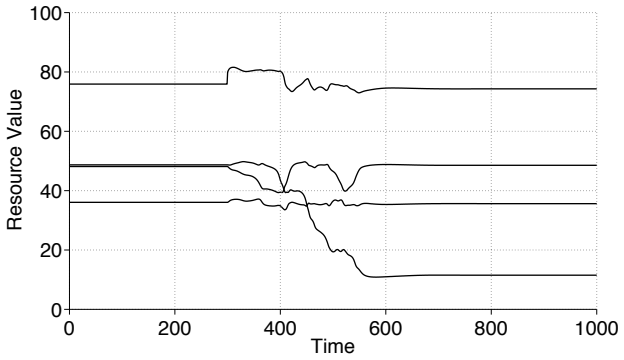


Figure 9: Resource values are shown for a 4 resource Daisystat. The system is perturbed at Time = 300 by increasing R_1 (the top line) by 5 units. This leads to a period of continual change in all environmental resources until Time $\approx$ 600 when a new set of stable resource values are established.

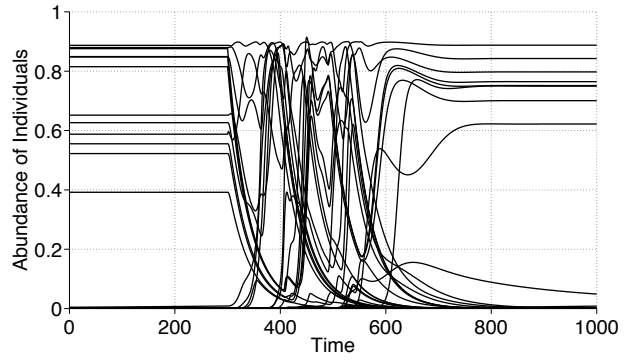


Figure 10: Abundances of individuals are shown for a 4 resource Daisystat. The perturbation of R_1 at Time = 300 produces a period of rapid change in the abundance of individuals as the population 'searches' for a new stable state.

environment and how they are affected by their environment are not prescribed. Consequently, homeostasis may be established anywhere over the range [0,100]. The explanation of homeostasis was given in terms of the rein control effects of a population. This also produced uniselector-type behaviour in that if a resource is driven outside of the range of the individuals that are currently regulating it, a sequence of events leads to all resource values being similarly driven and large changes in the population. Such changes continue until a new set of population responses and effects emerges that produce stability. The change in the abundances of individuals in the population can be described in terms of selection pressure, however there is no meaningful selection pressure for a population's effect on its resources. The homeostatic behaviour of the Daisystat is not a result of higher level selection.

Increasing the number of environmental resources demonstrated that the rein control system will operate in higher dimensions, an observation first made in Saunders et al. (2000). Regulation operating at planetary scales would be a very high dimensional system with a wide range of time and spatial scales. Daisystat can be considered as a first step in exploring higher dimensional regulation that emerges via population dynamics. In the Homeostat, as the number of units increases and so the size of the matrix of weights increases, the probability of randomly generating weight values that will produce a stable system decreases. Such observations resonate with the long-lasting debate surrounding Gaian regulation, that as there is only a single Earth, planetary homeostasis could not have evolved. While population dynamics may provide a possible account for a biological uniselector that can establish and recover stable states, it cannot explain how high dimensional systems could emerge. If we simplify the Daisystat into a network topology of feedback from and to environmental resources, then making the network more complex by increasing resources leads to the probability of it being stable reducing much in the same way as formulated in May (1972). However, the Earth system did not suddenly come into being 4.5 billion years as it is today. The hypothesis is that an effectively intractable problem in the form of determining a set of feedback values that will lead to stability for a high dimensional system can be made tractable by 'growing' such a system from initially low dimensions. In more concrete terms, this could involve incrementally adding new environmental resources to currently stable Daisystat systems. This may be seen as the emergence of new 'guilds' of organisms that both exploit and affect aspects of the environment that was either previously separated from the biota or did not even exist. Such an account has been proposed for the increase in complexity for the Earth system (Lenton et al., 2004)

Limitations and future work

The Daisystat is a very simple model intended as an 'opaque thought experiment' (Di Paolo et al., 2000) much in the same spirit as the original 'parable' of Daisyworld. Assumptions concerning population dynamics were very basic. It is important to note they resulted in no individual completely dying and being removed from the population. The number of individuals remained constant. Consequently biodiversity remained constant (if biodiversity is calculated as simply the number of existent species). However the abundances may be so small (approximately 10^{-5}) that their effects on the resource values can be safely ignored. Moreover, many Daisyworld studies including the original Watson & Lovelock model assumed a constant supply of either daisy 'seeds' or floor for the coverage of daisies. However, allowing species to go extinct in Daisystat could lead to a significant decrease in homeostatic behaviour due to the absence of the 'required' rein control species for a particular state of the system. Changing the total number of species via extinction in the absence of mutation and so creation of new species can be seen as reducing the Daisystat's amount of Ashbian variety. The connection between Ashby's law of requisite variety and biodiversity can be expressed as the greater the variety of the system (species in Daisystat) the greater the system's ability to reduce variety in the environment via regulation. There is significant scope to explore the relationship between biodiversity and stability in the Daisystat and how it changes as the dimensions of the environment changes.

A major assumption of the model is that all possible genomes are specified at the start of a simulation. There is no mutation of the alleles that determines an individual's effect on the environment and how it is affected by the environment. Introducing mutation would allow a range of evolutionary mechanisms to be explored and is a planned item for future work. The current approach of randomly initialising a population of individuals is consistent with the notion that 'everything is everywhere, but the environment selects' (see O'Malley (2007) for a historical review) which would support the assumption that it may be sufficient to generate sufficiently diverse simulated populations and then allow environmental conditions to select those individuals that will survive and perish.

No significant assessment of altering the rates at which individuals respond to and affect resources has been undertaken. This corresponds to $\alpha = \beta = 1$ in equation 2. These values can be seen as analogous to the 'viscosity' term in models of the Homeostat that modulates the rate of change of a unit's effect on the other units. There is much scope to explore the parameter space of different rates of change in Daisystat.

All the results presented featured Daisystats that were completely connected; all individuals were affected by and in turn affected all resources. Initial experiments that relaxed this assumption lead to more complex behaviour. For example when the connections were made more sparse, stable states that featured oscillations and limit cycles were observed. Exploring the effects of changing the density of connections in Daisystat represents a fertile area of future research.

Conclusion

A homeostatic model, the Daisystat, has been presented. This shares certain features and behaviour of the Daisyworld and Homeostat models. The Daisystat proved to be robust to two types of perturbation: instantaneous changes in one of the environmental resource values (analogous to one element in the Homeostat being subject to a sudden jolt); progressive driving of environmental resources (analogous to increasing luminosity in Daisyworld). This has demonstrated that Daisyworld-type homeostasis can be observed under minimal assumptions and with numerous environmen-

tal resources being subject to regulation (the original Daisyworld featured a single environmental resource in the form of planetary temperature). This has also demonstrated that a population of diverse individuals can perform the same function as a Homeostat uniselector by generating rapid changes in the feedback operating between the resources until new stable states are found. A plan of future research was outlined that would investigate the ability to incrementally increase the complexity of homeostatic systems and so provide a conceptual framework in order to understand how real world complex systems such as the Earth system have evolved from simpler states.

Acknowledgements

The author thanks the Helmholtz-Gemeinschaft as this research has been supported by the Helmholtz Association through the research alliance "Planetary Evolution and Life". The author would like to acknowledge the contributions of Richard Watson to the formulation of a number of ideas related to the Daisystat and the comments of three anonymous reviewers that greatly improved the paper.

References

Ashby, W. R. (1956). *Introduction to Cybernetics*. Chapman and Hall, London.

Ashby, W. R. (1960). *Design for a brain*. Chapman and Hall, London, 2nd edition.

Clynes, M. (1969). Cybernetic implications of rein control in perceptual and conceptual organization. *Annals of New York Academy of Science*, 156:629–670.

Di Paolo, E., Noble, J., and Bullock, S. (2000). Simulation models as opaque thought experiments. In Bedau, M. A., McCaskill, J. S., Packard, N., and Rasmussen, S., editors, *Artificial Life VII, Proceedings of the Seventh International Conference on the Simulation and Synthesis of Living Systems*, pages 497–506. MIT Press, Cambridge MA.

Dyke, J. G. (2009). *The Daisyworld control system*. PhD thesis, University of Sussex, UK.

Dyke, J. G. and Harvey, I. R. (2006). Pushing up the daisies. In Rocha, L. M., Yager, L. S., Bedau, M. A. Floreano, D., Goldstone, R. L., and Vespignani, A., editors, *Artificial Life X, Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, pages 426–431. MIT Press, Cambridge MA.

Dyke, J. G., McDonald-Gibson, J., Di Paolo, E., and Harvey, I. R. (2007). Increasing complexity can increase stability in a self-regulating ecosystem. In Almeida e Costa, F., Rocha, L. M., Costa, E., Harvey, I. R., and Coutinho, A., editors, *Proceedings of IXth European Conference on Artificial Life, ECAL 2007*, pages 133–142. Springer, Berlin.

Harvey, I. R. (2004). Homeostasis and rein control: From daisyworld to active perception. In Pollack, J., Bedau, M., Husbands, P., Ikegami, T., and Watson, R. A., editors, *Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems, ALIFE'9*, pages 309–314. MIT Press, Cambridge MA.

Kirchner, J. W. (2003). The gaia hypothesis: conjectures and refutations. *Climatic Change*, 58:21–45.

Lenton, T. M., Caldeira, K. G., and Szathmary, E. (2004). What does history teach us about the major transitions and the role of disturbances in the evolution of life and of the earth system? In *Earth System Analysis for Sustainability*. Dahlem Workshop Report 91. H.-J.

Lovelock, J. E. (1979). *Gaia: a new look at life on Earth*. Oxford University Press, Oxford.

Lovelock, J. E. (1983). Daisy world - a cybernetic proof of the gaia hypothesis. *The Co-evolution Quarterly*, Summer:66–72.

May, R. M. (1972). Will a large complex system be stable? *Nature*, 238:413–414.

McDonald-Gibson, J. (2006). Investigating gaia: A new mechanism for environmental regulation. Master's thesis, University of Sussex.

McDonald-Gibson, J., Dyke, J. G., Di Paolo, E., and Harvey, I. R. (2008). Environmental regulation can arise under minimal assumptions. *Journal of Theoretical Biology*, 251(4):653–666.

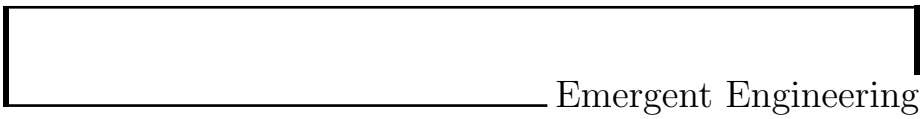
O'Malley, M. A. (2007). The nineteenth century roots of 'everything is everywhere'. *Nature Reviews of Microbiology*, 5(8):647–651.

Saunders, P., Koeslag, J. H., and Wessels, J. A. (1998). Integral rein control in physiology. *Journal of Theoretical Biology*, 194:163–173.

Saunders, P., Koeslag, J. H., and Wessels, J. A. (2000). Integral rein control in physiology ii: A general model. *Journal of Theoretical Biology*, 206:211–220.

Watson, A. J. and Lovelock, J. E. (1983). Biological homeostasis of the global environment: the parable of daisyworld. *Tellus Series B-Chemical and Physical Meteorology*, 35B:284–289.

Wood, A. J., Ackland, G. J., Dyke, J. G., Williams, H. T. P., and Lenton, T. M. (2008). Daisyworld: a review. *Reviews of Geophysics*, 46:RG1001.


Emergent Engineering

Evolution-Inspired Approaches for Engineering Emergent Robustness in an Uncertain Dynamic World

James M. Whitacre

CERCIA, School of Computer Science, University of Birmingham, UK
j.m.whitacre@cs.bham.ac.uk

Extended Abstract

Engineering involves the design and assemblage of elements that work in specific ways to achieve a predictable purpose and function. In systems design, engineering takes a conceptual "top-down" approach to problem solving that aims to decompose a complicated problem into separable and more manageable sub-problems. While this strategy has been successful in designing systems that deftly operate under predetermined conditions, these same systems are often notoriously fragile when conditions change unexpectedly.

In contrast, biological systems operate in a highly flexible manner with no pre-assignment between components and system traits. Instead of relying on the prediction of future environments, biological systems (e.g. immune systems, cell regulation) quickly learn/explore appropriate responses to novel conditions and inherit new routines to remain competitive under persistent environmental change.

Taking examples throughout biology, it has been proposed that degeneracy - the existence of multi-functioning components with context-dependent functional similarity - is a primary determinant of biological flexibility and a key differentiating factor in the robustness and evolvability of designed and evolved systems (Edelman and Gally 2001) (Whitacre 2010) (Whitacre and Bender 2010) (Whitacre and Bender 2010). Degeneracy is routinely eliminated in engineering design and its role in the robustness of biological traits is well-documented, however the influence that degeneracy might have on the flexibility of engineered and artificial systems has only begun to be investigated (Whitacre et al. in press).

Here we present evidence (Figure 1) that degeneracy enhances the robustness and evolvability (i.e. the rate and magnitude of heritable adaptive change) of multi-agent systems (MAS) that are taken from (Whitacre et al. in press) and modified to more closely reflect systems engineering problems subject to heterogeneous and unpredictable environments. First, we find degeneracy can increase MAS robustness toward a set of environments experienced during the MAS lifecycle. When robustness is important to fitness, we also find degeneracy can be selectively (not only passively/neutrally) acquired. However, and unbeknownst to myopic selection, this acquisition of degenerate robustness ultimately promotes faster rates of MAS design adaptation when the environment changes dramatically (at generation 3000, Figure 1), i.e. evolvability has been indirectly enhanced through the selection of degenerate forms of robustness. In contrast, robustness and evolvability are lower in MAS comprised of multi-functioning agents that are never degenerate, i.e. agents do not exhibit partially overlapping functionality but instead are either identical or completely dissimilar to other agents. In a forthcoming article, we further show that many of these findings can be reversed if environments are simplified and decomposable, i.e. environments show little variability during the MAS lifecycle and those environmental variations that are experienced are separable/modular.

In presenting these findings, we discuss how degeneracy might lead to new prescriptive guidelines for complex systems engineering: a nascent field that applies Darwinian and systems theory principles with the aim of improving flexibility and adaptation for systems that operate within volatile environments. We propose that versatile and functionally similar agents/sub-systems/software/vehicles/machinery/plans may sometimes dramatically improve a system's robustness to unexpected environments in ways that cannot be accounted for by economic portfolio theory.

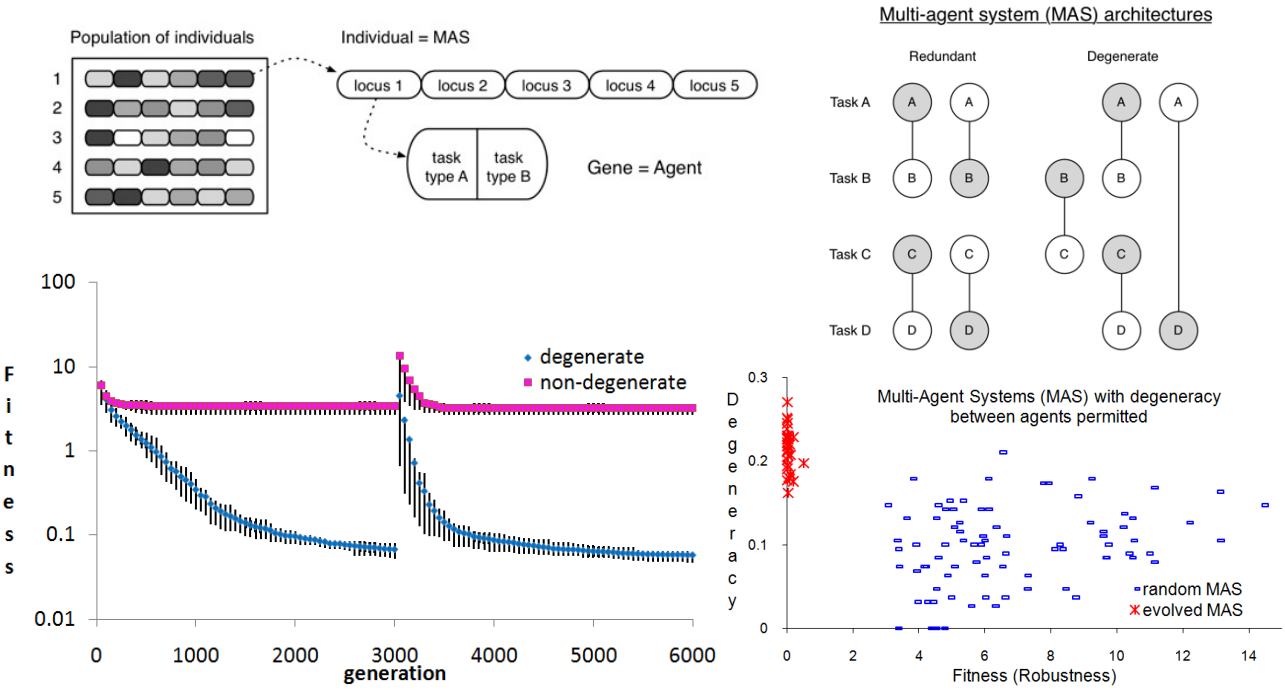


Figure 1: **Top-Left Panel**) Multi-Agent System (MAS) encoded within a genetic algorithm; for details, see (Whitacre et al. in press). Agents perform tasks to improve MAS fitness in its environment. **Top-Right Panel**) Illustration of genetic architectures for degenerate and non-degenerate MAS. Each agent is depicted by a pair of connected nodes, with the two nodes representing two types of (genetically determined) tasks an agent can perform. Models are adapted from (Whitacre et al. in press) to reflect a systems engineering context that is to be fully described in a forthcoming article. Differences in modeling conditions, compared with (Whitacre et al. in press), include: larger MAS (120 agents), each agent takes on more tasks during its interaction with the environment (20 tasks), agent behaviors are simulated using an unordered asynchronous updating scheme, environments are defined by more types of tasks (20 types, 48000 tasks in total), and new constraints in function combinations within each agent (to be described in forthcoming paper). **Bottom-Left Panel**) Evolution of MAS Fitness under one set of environments and then (at gen. 3000) evolution continues under a new set of environments. Optimal fitness = 0 for both original and new environments. Within the new environments, degenerate MAS appear to evolve more quickly while non-degenerate MAS evolve somewhat more gradually. **Bottom-Right Panel**) Degeneracy and fitness calculations for MAS in which degeneracy is permitted. Results show MAS evolved under random selection and MAS evolved to be robust within the environment. Here we see selection has increased degeneracy levels in the MAS (reported results are taken immediately after the first 3000 generations of evolution).

Acknowledgements: This research was partially supported by DSTO and CERCIA.

References

Edelman, G. M. and J. A. Gally (2001). "Degeneracy and complexity in biological systems." Proceedings of the National Academy of Sciences, USA **98**(24): 13763-13768.

Whitacre, J. M. (2010). "Degeneracy: a link between evolvability, robustness and complexity in biological systems." Theoretical Biology and Medical Modelling **7**(6).

Whitacre, J. M. and A. Bender (2010). "Degeneracy: a design principle for achieving robustness and evolvability." Journal of Theoretical Biology **263**(1): 143-153.

Whitacre, J. M. and A. Bender (2010). "Networked buffering: a basic mechanism for distributed robustness in complex adaptive systems." Theoretical Biology and Medical Modelling **7**(20).

Whitacre, J. M., P. Rohlfshagen, X. Yao and A. Bender (in press). The role of degenerate robustness in the evolvability of multi-agent systems in dynamic environments. 11th International Conference on Parallel Problem Solving from Nature (PPSN 2010), Krakow, Poland.

Re-Engineering Evolution: A Study In Self-Organising Synchronisation

Vito Trianni and Stefano Nolfi

Institute of Cognitive Sciences and Technologies,
National Research Council, Rome, Italy
vito.trianni@istc.cnr.it

Abstract

Evolutionary Robotics (ER) is a powerful approach for the automatic synthesis of robot controllers, as it requires little a priori knowledge about the problem to be solved in order to obtain good solutions. This is particularly true for collective and swarm robotics, in which the desired behaviour of the group is an indirect result of the control and communication rules followed by each individual. However, the experimenter must make several arbitrary choices in setting up the evolutionary process, in order to define the correct selective pressures that can lead to the desired results. In some cases, only a deep understanding of the obtained results can point to the critical aspects that constrain the system, which can be later modified in order to re-engineer the evolutionary process towards better solutions. In this paper, we present a case study about self-organising synchronisation in a group of robots, in which some arbitrarily chosen properties of the communication system hinder the scalability of the behaviour to large groups. We show that by modifying the communication system, artificial evolution can synthesise behaviours that properly scale with the group size.

Introduction

The synthesis of controllers for autonomous robots is a complex problem that has been faced with a large number of different techniques (Siciliano and Khatib, 2008). Among the various possibilities, Evolutionary Robotics (ER) represents a viable approach for the automatic synthesis of robot controllers requiring little a priori knowledge about the solution of a given problem (see Nolfi and Floreano, 2000). In fact, the evolutionary process proceeds in the bottom-up direction, directly evaluating controllers for their suitability to the requirements defined by the designer. When dealing with collective or swarm robotics systems, the usage of automatic techniques like ER is even more compelling, in particular when the group behaviour should be the result of a self-organising process arising from numerous interactions among robots. In such conditions, in fact, there is an indirect relationship between the desired group behaviour and the individual control rules. By evaluating the robotic system as a whole (i.e., by testing the global behaviour that results from the individual rules encoded into the individual genotype), ER provides an automatic process for identifying the mechanisms that produce and support the collective behaviour, and for implementing those mechanisms into the individual controller rules that regulate the robot/environment interactions (Trianni et al., 2008).

However, the advantages offered by Artificial Evolution are not costless, as pointed out by Matarić and Cliff (1996). In particular, it is necessary to identify the conditions that assure the *evolvability* of the system, i.e., the possibility to progressively synthesise better solutions starting from scratch. To do so, the experimenter has to make several choices in setting up the evolutionary process. Some of these choices are arbitrary if performed without any *a priori* knowledge of the system features, and may have a strong impact on the solutions found. This is often the case for the communication abilities provided to a collective robotics system. In fact, communication regulates the interactions among robots, and should be rich enough to support the emergence of the desired group behaviour. On the other hand, ER privileges simple sub-symbolic communication forms, as it contextually develops the behavioural and communication strategies, which co-evolve as a single whole. The selection of the best communication protocol should therefore face this tradeoff, and often only the experimenter intuition makes the difference between a valuable or an unfortunate choice.

Negative results should however be exploited to acquire information on the system dynamics and re-engineer evolution accordingly. In fact, by understanding the properties of unsuccessful systems it may be possible to recognise which are the critical aspects that constrain the system in sub-optimal solutions. In this paper, we present a case study of such an approach. We have studied self-organising synchronisation, in order to understand which are the minimal behavioural and communication strategies that would allow a group of robots to synchronise their periodic behaviour (Trianni and Nolfi, 2009). In particular, we are interested in the scalability property of the evolved behaviours to large groups. By analysing the evolved behaviours, we discovered that the arbitrary choice made in the communication protocol was hindering the evolved behaviour to suitably scale

to large groups. This finding allowed us to re-engineer the characteristics of the robots by identifying a new communication protocol, and to run further evolutionary experiments that resulted in properly scalable behaviours.

Evolution of Self-Organised Synchronisation

Self-organised synchronisation is a common phenomenon observed in many natural and artificial systems: simple coupling rules at the level of the individual components of the system result in an overall coherent behaviour (Strogatz, 2003). Probably, the most common synchronisation phenomenon is related to the flashing behaviour of some firefly species in South-East Asia, which aggregate at dusk and engage in massively synchronous displays (Buck, 1988). Models of this behaviour describe fireflies as a population of pulse-coupled oscillators with equal or very similar frequencies. These oscillators can influence each other by emitting a pulse that shifts or resets their oscillation phase. The numerous interactions among the individual oscillator-fireflies are sufficient to explain the synchronisation of the whole population (for more detail, see Buck (1988); Mirollo and Strogatz (1990); Strogatz and Stewart (1993)). This model has been often exploited to engineer systems capable of synchronous behaviour, also in collective and swarm robotics (Wischmann et al., 2006; Christensen et al., 2009). In this study, we have investigated which are the minimal behavioural and communicative conditions that can lead to synchronisation in a group of robots, in which each individual presents a periodic behaviour. For this purpose, we chose to provide robots with simple reactive controllers and basic communication abilities. The period and the phase of the individual behaviour are defined by the sensory-motor coordination of the robot, that is, by the dynamical interactions with the environment that result from the robot embodiment. We show that such dynamical interactions can be exploited for self-organised synchronisation, allowing to keep a minimal complexity of both the behavioural and the communication level (for more details, see Trianni and Nolfi, 2009).

Experimental setup

The evolutionary experiments are performed in simulation, using a simple kinematic model of the *s-bot* robot (see Fig. 1 and refer to Mondada et al., 2004, for details), and the results are afterwards validated on the physical platform. The experimental scenario for the evolution of self-organising synchronisation requires that each robot in the group displays a simple periodic behaviour, which should be entrained with the periodic behaviour of the other robots present in the arena. The individual periodic behaviour consists in oscillations along the y direction of a rectangular arena (see Figure 2). Oscillations are possible through the exploitation of a symmetric gradient in shades of grey painted on the ground. The gradient presents a white stripe for $|y| < 0.2\ m$, and black stripe for $|y| > 1\ m$.

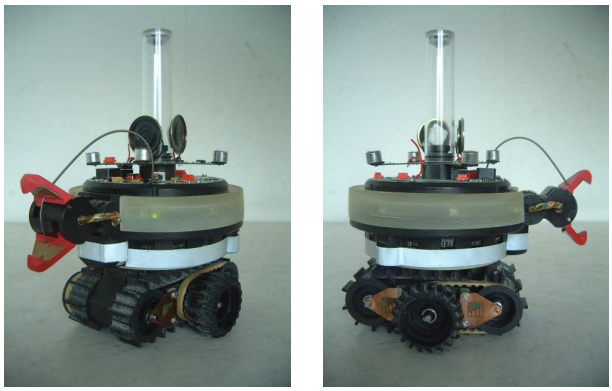


Figure 1: The *s-bot*, the robot used in the experiments.

For the purpose of engineering the evolutionary system, both the characteristics of the arena and the capabilities of the robots give several constraints to the experimental setup. According to these constraints, we select among the various possibilities the minimal set of sensors and actuators that are required to accomplish the task, that is, individual periodic oscillations over the grey gradient and synchronisation of the oscillation phase. Certainly, the controller needs access to the wheels' motors, and we set $\omega_M \approx 4.5\ s^{-1}$ as the maximum angular speed of the wheels. The grey gradient of the arena can be perceived by the robots through four infrared sensors placed under their chassis (ground sensors), which are appropriately scaled to encode the grey-level in the range $[0, 1]$, where 0 corresponds to black and 1 to white. The perception of the gradient through these sensors provides the robot with enough information to perform oscillations along the y axis. Additionally, robots need to use the infrared proximity sensors placed around their cylindrical body, in order to avoid collisions with walls or with other robots. These choices, which are mainly constrained by the arena setup and by the features of the physical robot, are sufficient for

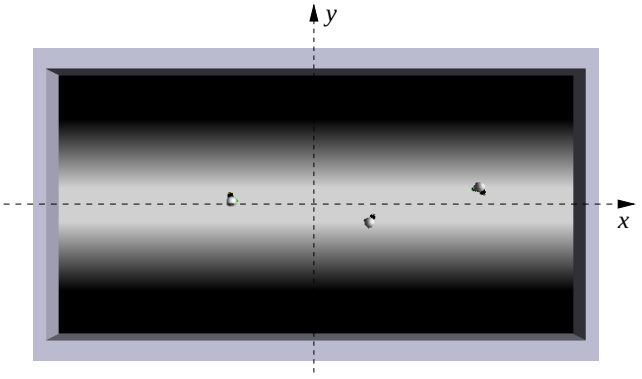


Figure 2: Snapshot of a simulation showing three robots in the experimental arena. The dashed lines indicate the reference frame used in the experiments.

the individual behaviour.

For what concerns the group behaviour, instead, we need to provide the robots with suitable interaction modalities that can lead to synchronisation of their movements. The choice of the communication system is the aspect we focus on in this paper. In fact, the *s-bot* platform features various communication devices, and we need to select among them the one that fits our experimental scenario. Robots are provided with speakers and microphones for sound communication. Moreover, robots can exploit coloured LEDs positioned around their turret to display a colour pattern that can be perceived through the omni-directional camera. Finally, robots have wireless communication abilities. Therefore, there is a large freedom in choosing the communication system. In order to maintain a minimal configuration, we decided to provide the robots with a *global* and *binary* communication system:

$$s(t) = \max_r S_r(t), \tag{1}$$

where $S_r(t) \in \{0, 1\}$ is the binary signal emitted by robot r at time t, and $s(t) \in \{0, 1\}$ is the binary signal perceived by all robots. In other words, each robot r can produce a signal $S_r(t)$. Signals produced by different robots cannot be distinguished, and result in a single signal $s(t)$ perceived by every robot in the arena, including the signalling one. Signals are perceived in a binary way: either there is someone signalling in the arena, or there is no one. This communication protocol is probably the poorest one in terms of the amount of information that can be conveyed. However, this is sufficient for our purposes, as we will see in the following. Note that this communication protocol can be easily implemented with sound signals: a robot can emit a single frequency tone with an intensity high enough to be perceived everywhere in the arena. Note that, differently from the other sensors and actuators, the choice of the communication system is not constrained by the robotic hardware or by other aspects of the experimental setup, but is only dictated by the communication protocol we have chosen.

Evolutionary Setup

Evolution was carried out using homogeneous groups of three robots, each controlled by a fully connected, feed forward neural network—a perceptron network. The neural controller takes as input the information coming from ground sensors, proximity sensors and perceived signals, and it controls the two wheels of the robot's differential drive system and the emission of binary signals. Connection weights and bias terms are genetically encoded parameters. The evolutionary algorithm is based on a population of 100 genotypes, which are randomly generated. This population of genotypes encodes the connection weights of 100 neural controllers. Each connection weight is represented with a 8-bit binary code mapped onto a real number ranging in $[-10, +10]$. Subsequent generations are produced by a combination of selection with elitism and mutation. Recombination is not used. At each generation, the four best individuals—i.e., the *elite*—are retained in the subsequent generation. The remainder of the population is generated by mutation of the 20 best individuals. Each genotype reproduces at most 5 times by applying mutation with 3% probability of flipping a bit. The evolutionary process runs for 500 generations.

The evolved genotype is mapped into a control structure that is cloned and downloaded onto all the robots taking part in the experiment, therefore obtaining a homogeneous group of robots. During evolution, we use groups composed of three robots only in order to obtain fast simulations. The performance of a genotype is evaluated by a 2-components function: $F = 0.5 \cdot F_{\mathcal{M}} + 0.5 \cdot F_{\mathcal{S}} \in [0, 1]$. The movement component $F_{\mathcal{M}}$ simply rewards robots that move along the y direction within the arena at maximum speed. This component rewards the movements of the robot from the observer perspective, without explicitly indicating how to perform a periodic behaviour: the oscillatory behaviour derives from the fact that the arena is surrounded by walls, so that oscillations during the whole trial are necessary to maximise $F_{\mathcal{M}}$. The second fitness component $F_{\mathcal{S}}$ rewards synchrony among the robots as the cross-correlation coefficient between the distance of the robots from the x axis. This component is therefore maximised by robots performing synchronous oscillations (either in-phase or anti-phase), and it is null when robots are maximally desynchronised. In addition to the fitness computation described above, two indirect selective pressures are present. First of all, a trial is stopped when a robot moves over the black-painted area, and we assign to the trial a performance $F = 0$. In this way, robots are rewarded to exploit the information coming from the ground sensors to perform the individual oscillatory movements. Secondly, a trial is stopped when a robot collides with the walls or with another robot, and also in this case we set $F = 0$. In this way, robots are evolved to efficiently avoid collisions. For more details on the fitness computation, refer to Trianni and Nolfi (2009).

Design and Evolution

Before presenting the obtained results, it is useful to discuss which are the features that are fixed by the experimenter, and those that are adaptively set by the evolutionary process. We have defined an experimental scenario that is intrinsically cooperative, because robots are homogeneous and are explicitly rewarded to display a desired group behaviour. We have also fixed the sensory-motor configuration and the controller architecture. In particular, we have fixed the interaction modality between different robots, which mainly happens through the binary and global communication signal. Notwithstanding this, the motor and communicative behaviour is not at all pre-determined, but it is the result of the evolutionary process. The individual behaviour and the syn-

chronisation mechanisms are completely determined by the parameters of the neural controller (i.e., connection weights and biases). Individual behaviour and communication signals co-evolve and mutually influence: the individual behaviour determines how the robot moves and experience the environment, which influences the signals emitted. In turns, perceived signals change the way in which the robot reacts to the environment. During evolution, the group behaviour is shaped in order to maximise the user-defined utility metric, within the constraints imposed by pre-determined features. In the following, we will see how the communication protocol we have chosen influences the obtained results.

Behavioural and scalability analyses

We performed 20 evolutionary replications, each starting with a different population of randomly generated genotypes. Each replication produced a successful synchronisation behaviour, in which robots display oscillatory movements along the y direction and synchronise with each other, according to the requirements of the devised fitness function. In general, it is possible to distinguish two phases in the evolved behaviours: an initial transitory phase during which robots achieve synchronisation, and a subsequent synchronised phase. The transitory phase may be characterised by physical interferences between robots due to collision avoidance, if robots are initialised close to each other. The collision avoidance behaviour performed in this condition eventually leads to a separation of the robots in the environment, so that further interferences to the individual oscillations are limited and synchronisation can be achieved. The synchronous phase is characterised by a stable synchronous oscillations of all robots, and small deviations from synchrony are immediately compensated.

The individual ability to perform oscillatory movements is based on the perception of the gradient painted on the arena floor, which gives information about the direction parallel to the y axis and about the point where to perform a U-turn and move back towards the x axis, therefore avoiding to end up into the black painted area. Each evolved controller produces a signalling behaviour that varies while the robots oscillate. The main role of the evolved signalling behaviour is to provide a coupling between the oscillating robots, in order to achieve synchronisation. In response to a perceived signal, robots react by moving in the environment, changing the trajectory of their oscillations. This results in a modulation of the oscillation amplitude and frequency, which allows the robots to reduce the phase difference among each other, and eventually synchronise. In a previous work (Trianni and Nolfi, 2009), we developed a mathematical model and exploited dynamical systems theory to thoroughly analyse the synchronisation behaviour. We invite the reader to refer to that work for further details on the synchronisation mechanisms, which are out of the scope of the present paper.

Once analysed the synchronisation behaviours evolved using three robots only, we tested their ability to scale up with the group size. To do so, we compared the performance of the evolved behaviour varying the group size. To avoid overcrowding, we performed the scalability analysis in larger arenas, ensuring a constant density of robots across the different settings. By ensuring a constant initial density we limit the negative effects of overcrowding and we are able to compare the performance of robotic systems with varying group size. In order to keep a constant robot density equal to the one used in the evolutionary experiments, we lengthened the arena in the x direction, trying to keep an initial density of 0.25 robots per square meter. Despite the increased arena length, we still keep the same communication protocol, that is, communication continues to be binary and global, with all robots affecting each other. This choice allows us to evaluate the scalability of a behaviour as it was evolved, without modifying the features of the communication channel. We evaluated all best evolved controllers 100 times using six different group sizes (3, 6, 12, 24, 48 and 96 robots). The obtained results are presented in the top part of Figure 3. It is possible to notice that most of the best evolved controllers have a good performance for groups composed of 6 robots. Performance degrades for larger group sizes and only few controllers produce scalable behaviours up to groups formed by 96 robots. The main problem that reduces the scalability of the evolved controllers is given by the physical interactions among robots. Despite the constant initial density we introduced in order to limit the disruptive effect of collision avoidance, physical interactions nevertheless occur with a higher probability per time step, as the group size increases. Every collision avoidance action provokes a temporary desynchronisation of at least two robots, which have to adjust their movements in order to re-gain synchronous oscillations with other robots. In such cases, the whole group is influenced by the attempt of few robots to re-gain synchronisation, due to the global and binary communication.

To summarise, the above analysis showed that physical interactions and collision avoidance have a disruptive effect on the synchronisation ability of the robots, and this effect is more and more visible as the group size increases. However, the synchronisation mechanism evolved may scale with the group size if we ignore physical interactions. To test this hypothesis, we performed an identical scalability analysis, but in this case we ignore the physical interactions among the robots, as if each robot was placed in a different arena and perceived the other robots only through communication signals. The obtained results are plotted in the bottom part of Figure 3. Differently from what was observed above, in this case many controllers present good scalability, with only a slight decrease in performance due to the longer time required by larger groups to perfectly synchronise (namely, controllers evolved in replication number 2, 8, 10, 12, 14, 18 and 19). This result confirms the analysis about the neg-

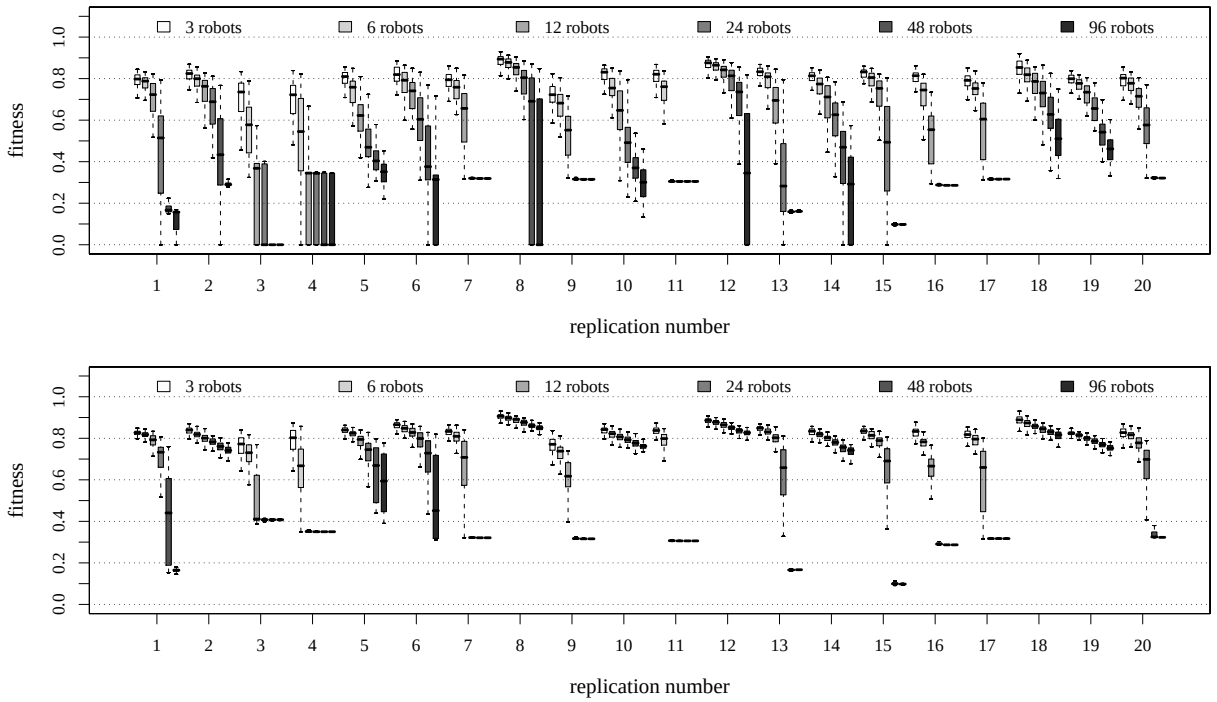


Figure 3: Scalability analysis. The boxplot shows, for each evolved controller, the performance obtained in tests with 3, 6, 12, 24, 48, and 96 robots. Each box represents the inter-quartile range of the data, while the black horizontal line inside the box marks the median value. The whiskers extend to the most extreme data points within 1.5 times the inter-quartile range from the box. Outliers are not shown. Top: scalability of the evolved controllers under normal conditions. Bottom: scalability of the synchronisation mechanism.

ative impact of physical interferences and collisions among robots. In fact, removing the necessity to avoid collisions leads to scalable self-organising behaviours.

Nevertheless, many other controllers present a strange behaviour (namely, controllers evolved in replication number 3, 4, 7, 9, 11, 13, 15, 16, 17, 20). It is possible to notice that the performance presents a high variability up to a certain group size. The variable performance indicates that in some cases the robots are able to synchronise, and in other cases not. With larger group sizes, the performance stabilises to a low, constant value, independent from the initial conditions and the number of robots used. This value, which is characteristic of each non-scaling controller, represents the performance of the robotic system trapped into the basin of an *incoherent attractor*. In other words, the robotic system always converges into a dynamical condition in which no robot can synchronise with any other. By observing the actual behaviour produced by these controllers, we realised that the incoherent condition is caused by a communicative interference problem: the signals emitted by different robots overlap in time and are perceived as a constant signal (signals are global and are perceived in a binary way, preventing a robot from recognising different signal sources). If the perceived signal does not vary in time, it does not bring enough information to be exploited for synchronisation, and the system remains desynchronised. This result is confirmed by the dynamical system analysis that we performed, which revealed how the individual signalling behaviour is responsible for producing such communicative interference, allowing also to predict which controllers present scalability just looking at the individual behaviour (see Trianni and Nolfi, 2009, for more details).

Re-engineering for scalability

The analysis of the unsuccessful controllers revealed that scalability cannot be always obtained, due to the physical and communicative interferences among robots. In particular, the communication protocol we selected has a strong impact on the scalability of the system. In fact, communication is global and binary, that is, the signal emitted by a robot is perceived by any other robot everywhere in the arena. Moreover, from the robot point of view, there is no difference between a single robot and a thousand signalling at the same time. Therefore, a single robot can influence the whole group. This has no negative effect as long as robots are synchronous, but can have severe consequences when a robot modifies its behaviour due to collision avoidance following some physical interaction with other robots. Further-

more, the binary communication channel generates the communicative interference we described above, which prevent the group from synchronising in certain conditions.

The main problems are therefore related to the absence of *locality*—i.e., signals are perceived everywhere in the arena—and of *additivity*—i.e., signals overlap without adding, preventing to recognise how many robots are contemporaneously signalling. The lack of locality and additivity is the main cause of failure for the scalability of the evolved synchronisation mechanisms.[1] We therefore decided to re-engineer our evolutionary experiments changing the communication protocol, which was arbitrarily chosen in the first place. Given that we are interested in studying global synchronisation, we decided to re-engineer our experiments focusing only on the additivity of the communication system. This allows us to make only minor changes to the experimental setup and directly compare the effects of the re-engineering approach.

Modified Experimental Setup

We evolved self-organising synchronisation behaviours exploiting exactly the same setup as above, but changing the way robots signal and perceive emitted signals. Specifically, we change the binary communication system with a continuous one:

$$\tilde{s}(t) = \frac{1}{N}\sum_{r=1}^{N}\tilde{S}_r(t), \qquad (2)$$

Now, robots always emit a signal $\tilde{S}_r(t) \in [0,1]$, encoding a number in a continuous range. The emitted signals are perceived as the average $\tilde{s}(t)$ among all the perceived signals. By doing so, the influence of an individual robot on the global perceived signal—which is equal for all robots in the arena—depends on the signalling behaviour of the whole group: the bigger the group, the smaller the influence of the single individual. This communication protocol can be easily implemented on the *s-bots*. For instance, signals could be sent as messages over the wireless network containing a real number in [0,1]. On the basis of the analysis performed so far, we expect that self-organising synchronisation behaviour can be evolved with such a communication system, and that they are more scalable.

Analysis of the Obtained Results

Also in this case, we performed 20 evolutionary runs for groups of three robots. All evolutionary runs were successful, and produced synchronisation behaviours that are qualitatively similar to those obtained with the binary communication system: robots perform oscillations over the painted gradient and react to the perceived signal by modifying the individual behaviour, in order to synchronise with other robots. The scalability analysis was performed with the same modalities as described above, and the obtained results are presented in Figure 4.

In the upper plot, scalability is tested including physical interactions. Also in this case, we notice that collisions prevent the scalability of some controllers, in which a good avoidance behaviour was not evolved. Recall that when a collision is detected, the group scores a null performance. However, it is possible to notice that the usage of an additive communication system leads to better performance even with large groups. Most controllers present good scalability for every tested group size, and only collisions substantially reduce the performance. Here, differently from what was observed before, physical interactions and collision avoidance do not have a severe impact on the performance of the whole group. In fact, the signals of few non-synchronous robots are averaged with those emitted by the rest of the group. As a consequence, the influence on the group of a robot attempting to synchronise decreases with increasing group size. This leads to a quick convergence to synchrony and to an improved group performance.

To better understand the effects of the re-engineering approach, we also performed a scalability analysis for the evolved synchronisation mechanisms, again removing the physical interactions among robots. The results plotted in the lower part of Figure 4 show that all evolved synchronisation mechanisms perfectly scale, and they do not suffer from the communicative interference observed with binary signals. In fact, the perceived signal brings information about the average signalling behaviour of all robots. As a consequence, synchronisation is always achieved, no matter the group size. Notice also that all controllers present a linear decrease in performance in correspondence to an exponential growth of the group size. This observation suggests that the self-organising synchronisation mechanism is very efficient, and is only slightly affected by the group size.

Discussion and Conclusions

In this paper, we have presented a case study about the evolution of self-organising synchronisation in a robotic system. In setting up the experiments, some characteristics of the system were chosen arbitrarily, given that no *a priori* knowledge was available about the possible solutions to the given problem. The results obtained with the initial approach proved that self-organising synchronisation can be actually achieved with a minimal complexity at the level of the control and communication strategy. However, the analysis of the scalability results also pointed to some characteristics of the system that hindered the group from scoring a good performance. We identified the problem in the communication system being global and binary, and to the effects of physical and communicative interferences. To solve this problem, we re-engineered the arbitrarily-chosen communication protocol exploiting the knowledge acquired by analysing the evolved behaviours. The newly devised continuous signals

[1] However, as we have seen, this problem affects only some of the analysed controllers.

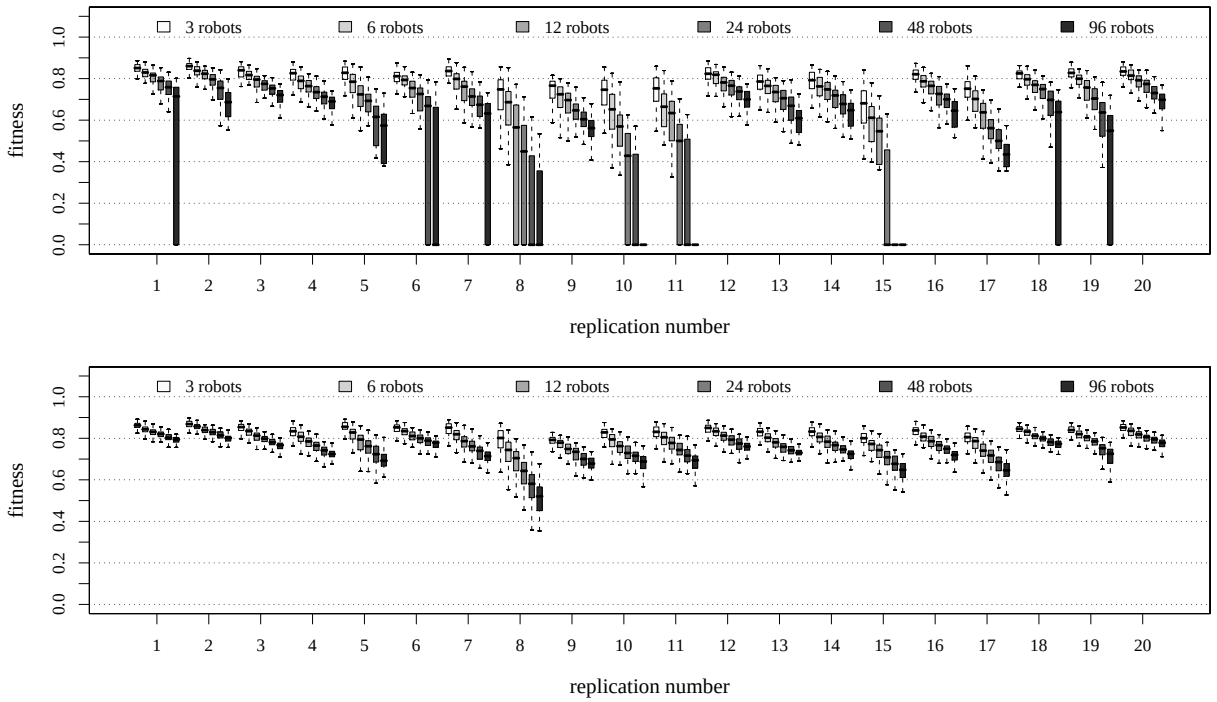


Figure 4: Scalability analysis for the continuous communication system. Top: scalability of the evolved controllers under normal conditions. Bottom: scalability of the synchronisation mechanism.

resulted in better synchronisation behaviours, and in an optimally scaling communication system.

The methodology described here may be generalised. Evolutionary Robotics is actually very useful for the automatic synthesis of controllers for robotic systems. However, it does not exclude arbitrary choices. The advantage given by ER is that, despite such arbitrary choices, it can find good solutions to a given problem. However, much as in conventional engineering methods, multiple design loops may be needed to find optimal results. This paper demonstrates that it is possible to engineer some features of a system undergoing artificial evolution on the basis of the outcomes of the evolutionary process itself. Contrary to trial and error methods without any guidance, we showed that an attentive analysis of negative results conveys knowledge on how to modify the system for evolving better solutions. Note that this is not in contradiction with respect to the need of little *a priori* knowledge in the design of the evolutionary experiment, as mentioned in the introduction. The knowledge we put into the system should not be related to the design of the solution, which is left to the evolutionary process, but rather to the preconditions required for obtaining good solutions.

We believe that it is necessary to formalise an engineering approach to Evolutionary Robotics, which can guide the design of evolutionary experiments. This is particularly true for collective and swarm robotics, in which the desired behaviour of the group is an indirect result of the control and communication rules followed by each individual. Let's consider here the case in which the robotic hardware available is fixed, and the problem to be solved is well defined, as in any engineering application. In these conditions, it is possible to identify four major issues in the design of the evolutionary system: (i) the definition of the robot sensory-motor configuration (ii) the definition of the genotype-to-phenotype mapping, (iii) the definition of the fitness function, and (iv) the definition of the ecological selective pressures. In this paper, we have just dealt with the robot configuration, and in particular with the communication protocol. In the following, we briefly discuss the other issues.

With respect to the genotype-to-phenotype mapping, the design choices concern mainly the type of controller to be used, and the way in which the genotype is translated into such controller. A widely used approach in the literature consists in encoding into the genotype a fixed number of parameters of the robot controller (typically realized through an artificial neural network), while keeping constant the controller structure. Other approaches are possible, such as evolving the controller architecture (Stanley and Miikkulainen, 2002), or evolving controller programs instead of neural networks (Koza, 1992). In collective robotics, another characteristics that has to be determined concerns the genetic relatedness between the individuals forming the group, that is, whether they are *genetically homogeneous* (i.e., they are clones) or *heterogeneous* (i.e., they differ from

each other). The advantage of homogeneous groups are given by a very compact encoding for the parameters of the controllers of the whole group, independently of its size. This advantage comes at the cost of a higher difficulty in obtaining roles that are well defined and differentiated. If this is a requirement, then heterogeneous groups might be more indicated. On the other hand, heterogeneous groups lead to a larger search space, require to estimate each individual contribution to the group performance, or need to identify in advance the role played by different individuals.

For what concerns the fitness function, it is difficult to suggest general principles for properly engineering it, because it strongly depends on the particular experimental conditions. Floreano and Urzelai (2000) propose the usage of a three-dimensional *fitness space*, in which the different dimensions refer to important features of a fitness function. In a collective robotics setup, the definition of a fitness function is more complex, due to the indirect relationship between individual actions and group organisation. A viable approach is given by functions that reward the final outcome of the collective behaviour, rather than the way in which the goal is achieved. This can be done, whenever possible, by measuring group variables that are available to the observer.

Finally, a typical problem of ER is the correct estimation of the performance of a genotype. The fitness function should evaluate the quality of the robot behaviour with respect to some variability of the environment. Typically, the behaviour must be robust with respect to varying initial position and orientation of the robot, and with respect to other parameters that contribute to define the *ecological niche* in which the behaviour is evolved. In order to obtain a reasonable fitness estimate, it is necessary to sample the space of the possible ecological conditions in an appropriate way. In a collective robotics setup, the problem is worsened by the presence of multiple robots, which increase the variability of the ecological niche. It is important to notice that indirect selective pressures may be created through the definition of the ecological niche and through the sampling employed to estimate the fitness. Given that the group is evaluated for presenting a robust behaviour within the parameter space of the ecological niche, the choice of the sampling may influence the evolutionary path. For these reasons, a careful design is required.

In our view, these are the main methodological choices that need to be performed when setting up an evolutionary experiment. In future work, we plan to carefully analyse these issues with both a theoretical and experimental work, in order to better formalise an engineering approach to Evolutionary Robotics

Acknowledgements

This work was supported by the Swarmanoid project, funded by the Future and Emerging Technologies programme (IST-FET) of the European Commission under grant IST-022888. The information provided is the sole responsibility of the authors and does not reflect the European Commission's opinion. The European Commission is not responsible for any use that might be made of data appearing in this publication.

References

Buck, J. (1988). Synchronous rhythmic flashing of fireflies. II. *The Quarterly Review of Biology*, 63(3):256–289.

Christensen, A. L., O'Grady, R., and Dorigo, M. (2009). From fireflies to fault tolerant swarms of robots. *IEEE Transactions on Evolutionary Computation*, 13(4):754–766.

Floreano, D. and Urzelai, J. (2000). Evolutionary robots with on-line self-organization and behavioral fitness. *Neural Networks*, 13(4–5):431–443.

Koza, J. (1992). *Genetic Programming*. MIT Press, Cambridge, MA.

Matarić, M. J. and Cliff, D. (1996). Challenges in evolving controllers for physical robots. *Robotics and Autonomous Systems*, 19(1):67–83.

Mirollo, R. E. and Strogatz, S. H. (1990). Synchronization of pulse-coupled biological oscillators. *SIAM Journal on Applied Mathematics*, 50(6):1645–1662.

Mondada, F., Pettinaro, G. C., Guignard, A., Kwee, I. V., Floreano, D., Deneubourg, J.-L., Nolfi, S., Gambardella, L. M., and Dorigo, M. (2004). SWARM-BOT: A new distributed robotic concept. *Autonomous Robots*, 17(2–3):193–221.

Nolfi, S. and Floreano, D. (2000). *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press/Bradford Books, Cambridge, MA.

Siciliano, B. and Khatib, O., editors (2008). *Springer Handbook of Robotics*. Springer Verlag, Berlin, Germany.

Stanley, K. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127.

Strogatz, S. H. (2003). *Sync: The Emerging Science of Spontaneous Order*. Hyperion Press, New York, NY.

Strogatz, S. H. and Stewart, I. (1993). Coupled oscillators and biological synchronization. *Scientific American*, 269(6):102–109.

Trianni, V. and Nolfi, S. (2009). Self-organising sync in a robotic swarm. A dynamical system view. *IEEE Transactions on Evolutionary Computation*, 13(4):722–741.

Trianni, V., Nolfi, S., and Dorigo, M. (2008). Evolution, self-organisation and swarm robotics. In Blum, C. and Merkle, D., editors, *Swarm Intelligence. Introduction and Applications*, Natural Computing Series, pages 163–192. Springer Verlag, Berlin, Germany.

Wischmann, S., Huelse, M., Knabe, J. F., and Pasemann, F. (2006). Synchronization of internal neural rhythms in multi-robotic systems. *Adaptive Behavior*, 14(2):117–127.

How to Facilitate Variability

Taivo Lints

Research Laboratory for Proactive Technologies
Tallinn University of Technology, Ehitajate tee 5, 19086 Tallinn, Estonia
taivo@taivo.net

Abstract

The concepts of *life* and *intelligence* almost *require* the system to be adaptive. And adaptivity, in turn, is usually strongly dependent on the continual generation of variations in the system. The paper discusses various ways of producing the required variations, and how to support these production processes.

Introduction

The property of being alive seems to almost *require* (if not yet with scientific rigor, then at least intuitively) the existence of adaptational processes in the system – it is difficult to imagine a lifeform whose internal processes and behavior would not depend in any reasonable (fitness-linked) way on the situation the organism is in. Adaptivity, in turn, has strong, though less strict, ties with the generation of *variations* in / by the system.

The evolution theory inspired approaches to adaptation consider it to be a process where variations of existing individuals are being generated and where selection operates on those variants, probabilistically eliminating the less fit ones. The variation-selection loop is not a strict requirement for adaptation in general (because adaptive behavior can also be displayed by a system that is able to accurately enough estimate the required states and actions and generate them in "one shot"), but nevertheless a notable portion of adaptational processes can be described as having such a character.

In cybernetics, too, the importance of variety for a system's ability to cope is emphasized, though in a slightly different sense: "The larger the variety of actions available to a control system, the larger the variety of perturbations it is able to compensate." (Ashby's (1956) idea of requisite variety, as summarized by Heylighen and Joslyn, 2001). Here, the variants are not exactly competing with each other for survival, but rather form an operational repertoire the system can draw from as required by the circumstances.

The widespread usage of the concept of diversity in debates about sustainability and problem solving furthermore suggests that the existence of variations in a system may increase its adaptivity as well as robustness.

And, finally, the need for some kinds of variations in a system that is considered adaptive derives directly from the essence of adaptation itself, which can be defined as "changing something (itself, others, the environment) so that it would be more suitable or fit for some purpose than it would have otherwise been" (Lints, 2010) – the term 'change' is pretty much synonymous with 'variation in time', i.e., something is transformed from one state to another and there are different variants of it at different time points (which, in turn, may, or may not, depending on the system, be facilitated by the existence of multiple simultaneously present variations of system elements (components, processes, relations, etc.)).

All in all, then, it is of great import for adaptation research, and, consequently, for ALife research, to study the ways how variability can be stimulated. At least three issues can be identified. Firstly, the very generation itself – what are the ways to produce variations. Secondly, how to support that generation, i.e., how to make it easier for the generative processes to operate well in a system. And thirdly, how to trigger the production of new relevant variations when the mechanisms are already in place but latent or unguided. This paper explores the first two of these issues. It should be noted that the paper grew out of the author's untested pondering on the topic of adaptivity and does *not* attempt to survey the variability related research done so far (and, accordingly, the given references are *not* representative of the main research efforts of that direction; but, on the other hand, it is exactly because of that why the paper might potentially provide some perspectives, connections and summarizations interestingly divergent from the usual).

Ways of Generating Variations

There exist several perspectives from which to dissect the ways of producing variations. One might be called a "creativity perspective", which lists the possibilities in accordance with how (or if) the novelty is produced (surely, the terms *creativity* and *novelty* are somewhat difficult to define, but for our current purposes they serve mostly as referential labels and thus the lack of rigorous definitions is not particularly problematic). The baseline would be having no nov-

elty at all, from the system's own perspective and (to keep the current discussion within reasonable limits) with regard to the set of variations, not the set of pairings of variations with the situations. This would be the case when, for example, all the possible variations already exist in some kind of an internal repository and the system merely draws them from this store.

Combinatorial novelty can be produced through, as the name suggests, producing novel combinations of existing elements, be they physical system parts or various signals, processes, arrangements, etc. In genetics, a typical example would be the crossover operation that basically takes some DNA strands from two individuals and swaps some of their sections with each other. But combinatorial novelty is not limited to preserving the sizes or numbers of inputs, of course, and may in principle use any kind of element pool to produce any other kind of element pool constructable from the (parts of the) initial material. If the arrangement of the elements is important for the system, then a mere rearrangement (permutation) can also be considered to produce a novel variant from existing parts. Another noteworthy possibility is the so-called *bootstrapping* where the products of one generational cycle are used as elementary building blocks in the next cycle (it is worth emphasizing, though, that bootstrapping is a powerful method not limited to combinatorial approach and can be used with most of the other techniques as well).

To produce new alterations in a possibly noncombinatorial way (though it can also be used with the combinatorial method), the first approach would be incremental tuning or modification of system's parameters and parts, i.e., moving around relatively smoothly in the space of modifiables. Whether this translates to the system moving around smoothly in its state space as well depends on the mappings from modifiables to system states and dynamics, as well as on the general complexity and nonlinearity of the system. In developmental systems the extent of the effect a modification has is usually also strongly dependent on how early in the development the modification was made – early changes often have strong effects (which helps to explain why, especially in biology, early development often remains relatively conservative in comparison to later development: the large impacts of early alterations render, in most cases, the system unfit (Bennett, 1997) and thus are selected against).

Moving up on the hypothetical creativity ladder we find the revolutionary, "truly creative" change, the existence of rigorous meaning and essence of which is somewhat questionable, but intuitively it implies the occurence of particularly noteworthy advances, strong originality and innovation, and large unexpected (but clever, at least in hindsight) changes in modifiables, as opposed to the more mundane step-by-step tuning. In practice, though, the line between incremental and revolutionary is blurry, and even more so with the occasional distinction between truly creative and "just" combinatorial, as it is actually common for the breakthrough ideas to stem from intensive work with extensive presence of both incremental and combinatorial methods. Also, in nonlinear systems the slight tuning of some system parameter can lead to substantial changes in other variables.

A classification somewhat orthogonal to the previously described one can be reached at when differentiating between the system being self-contained with regard to novelty creation versus it drawing some variants, or elements of them, from external sources. The most obvious situation would be using an external knowledge repository, the form of which can range from databases through helpful systems / agents up to the vast accumulated knowledge of the whole human, or other, culture. Another possibility is the incorporation of (or merging with) external components that supplement system's own capabilities. This might be done temporarily on the basis of need, or also permanently. In some cases even the temporary inclusion of a component (say, an employee) can permanently upgrade the system's abilities (say, in the form of idea exchange / extraction). Probably the most complex, but accordingly with the highest potential payoff, way of acquiring variations from external world is a (mutual, creative, constructive, temporally extended) interchange process between the system and various external agents.

Yet another perspective on producing variations can be constructed by focusing on the spectrum of possible uses of randomness and determinism in the system – whether the search for new variations (or the act of retrieving existing ones from some repository) is random or determined, guided by previous experience or not, and what characteristics the sources of randomness have.

A fully random search with a flat probability distribution samples the search space, by definition, uniformly and without any guidance from previous experience. A possibility to be noted, though, is that if the search space is not the same as the space of directly testable outcomes (e.g., genotypes are being varied but the selection is based on final organisms that develop under the guidance of those genotypes), the probability distribution may well become skewed somewhere in the mappings from modifiables to testables (the mappings can be very complex, involve generative rules, randomness, context-dependence, emergent behavior, self-organization, etc.). For the system this could be either a problem or an opportunity.

As the probability distributions become less and less flat, either through the changes in the aforementioned mappings or directly at the source of randomness, there will be more and more predictability (at least in principle) in the system, finally in the limit reaching full determinism. The shaping of distributions might be accidental, but a considerably more interesting case is when it is used as a way to store previous experience or externally acquired knowledge – those regions of search space that have become known to be more likely

to contain good solutions are searched more thoroughly and preferentially earlier than other regions. One has to be careful, though, to take into account the possibility of the circumstances changing or of the existence of special cases, for both of which the solutions may lie in areas previously experienced as solution-poor.

In some cases the search can also be exhaustive, generating all the possible variations of the modifiable(s). While exhaustive search is typically prohibitively costly, and purely random search too unintelligent, the option to use them should not be totally forgotten or immediately discarded, as occasionally they may really turn out to be the most viable ways to find good solutions (e.g., Wolfram, 2002, page 393).

As a fair share of interesting systems could be classified as nonlinear and complex, there is one more potentially important source of variations: deterministic chaos. It can amplify minor fluctuations and deviations, both deliberate and accidental, deterministic and random, into major changes in system dynamics totally, and in practice quite unpredictably, altering the system's behavior in the long run.

For the probabilistic and deviation-amplifying methods to work properly, it is necessary to have a source of randomness. This can be located either inside or outside the system, and be truly random or pseudorandom. If the usage of the source is deliberate, the values of the random variable might be explicitly acquired from the source, but in most cases the randomness kind of "leaks in" as noise in imperfect sensors, signal channels, processing elements, actuators, etc., or in the form of perturbations of the "normal" system behavior, composition or organization.

One more informative way of classifying the variation-producing methods rests on the sequential-parallel scale, distinguishing between systems that create new variations one by one in a row (and, in extreme cases, only allow the existence of one variant at a time) and systems that either spawn multiple simultaneously active variety generators or just generate a number of alternatives more or less instantaneously (at least from the practical viewpoint).

While it is educative to be aware of all the described techniques, it should be kept in mind that they are not mutually exclusive – it can often be advantageous to combine various approaches instead of relying on a single mechanism. The partial orthogonality of the "perspectives" is relatively obvious, but even within a single perspective there are possibilities for diversity, e.g., having both random and deterministic, or both parallel and sequential variation generators present in the same system. The different mechanisms can be applied to altering different modifiables, be cooperating on the same ones, act as backups for each other, and so on.

Supporting the Generation of Variations

For the various aforementioned methods to have a possibility to work well, the system they operate in should provide some specific support in the form of having certain features and resources. Some of the most important ways of help are described in the following subsections.

Making the Modifiables Easy to Change

The job of a variation generator could be roughly described as producing altered versions of the system, usually based on the system's previous state(s) or on some template or seed. An alteration is basically a change of some modifiable features of the system, executed either in the very same system (component) or by fabricating a new altered copy instead. It is quite straightforward to deduce, then, that making the modifiables easy to change can make the job of the generator much easier.

The specifics of how the effortlessness can be achieved depend, obviously, on the particular system, but in general the following keywords might give the first hints on the directions to pursue: tunability, reconfigurability, rearrangeability, reroutability, flexibility, plasticity, elasticity, adjustability. The main connective idea here, almost by definition, is to reduce the resistance to change. This includes reducing the cost of adjustment actions, increasing responsiveness (the speed at which the changes can be made), relaxing constraints (except maybe the ones that directly support variation generation by keeping the corresponding mechanisms functional, e.g., in genetic systems "the extremely high internal correlations underlying the transcription and translation mechanisms allow for a large ensemble of variants" (Conrad, 1983, page 338)), removing various barriers, and also increasing the number of options for each modifiable feature (both by expanding the range and by upping the density of allowed positions in that range) as well as the number of modifiables themselves. In addition to reducing the cost of adjustment actions, the (meta-level) costs of *maintaining* the flexibility are also important to be paid attention to and reduced as much as possible or feasible.

As of increasing the number of options, an interesting concept is neutral variation on a flat plateau of fitness landscape, meaning that something can be varied a lot without affecting the measure of system's current successfulness much. In general this is *not* what we would like to have when enlarging the set of options, because by definition the added options on the same plateau give the same fitness result as those already existing there. However, there still exist potential ways to use it. One is to notice that although different spots on the same plateau do have the same elevation, their neighboring areas might not, thus the new options might provide better access to new interesting places on the fitness landscape while being easy to reach themselves due to neutrality (because of being similar to other variants there is likely to be less resistance against moving into them) (e.g., Lenski et al., 2006). Another possibility is to look at some kind of an "opposition to alterations" landscape instead (the construction of which is trickier, though, as the resistance to moving into a given point depends not only on the static

paramater values of that point but also on dynamics, and is typically not the same for different origins of alteration), find plateaus there and define neutrality on such a basis. Then the areas of interest would be plains of low resistance but of useful variability in fitness-relevant dimensions.

A related concept, originating from physics, is referred to as the system having glassy properties and means, among other things, that there are "multiple low-energy minima in the energy landscape of the system" (Menashe et al., 2000). This, in turn, means that there is no uniquely predetermined state to which the system would always try to fall, but instead a variety of equi-energetic states to "choose" from. And that possibility of choice would increase the system's potential capacity to adapt, and would move it closer to (or further in) the domain of biology (Stec, 2004).

Yet another related idea for fostering variety is to keep the system sufficiently far from equilibrium so that it has plenty of stationary states to choose from (Heylighen, 2001).

Whereas neutral variation and ideas related to it definitely deserve further research about how to apply them for supporting variety generation, they are probably not the key concepts and were given a somewhat disproportionate attention here mainly due to their intellectual appeal. A considerably better studied and in all likelihood more important notion is that of modularity – something consisting of changeable pieces is typically a lot easier to modify than a monolithic structure. Although modularity promptly associates with some physical system or software being composed of distinct components, the idea has a lot wider applicability. To give a few examples, it is possible (and sometimes possibly enlightening) to talk about modularity in time, modularity of search space, state space, action space, or some more exotic space, modularity of representations, behaviors, signals, protocols, functionality, resources, and much more.

Linking the concepts of tunability and modularity, we can arrive at the idea of having tunable and exchangeable components. In general this is a thought too obvious maybe to even mention, but in some areas it does not necessarily come to mind that easily, yet is exceedingly useful nevertheless. An example would be for a system to have switchable sets of tunable behaviors where tuning improves the currently active set and changeovers are triggered by context changes, as opposed to having only a single tunable set that can slowly become another (distant) one as is common in simpler artificial learning systems (Moorman and Ram, 1992).

An additional option for supporting variation generation is to make the modifiables polyadjustable, that is, to have the same feature be adjustable by a variety of different mechanisms (Knoll and Järvenpää, 1994). Depending on the specific circumstances this can provide the system with the possibility to choose the most efficient change mechanism for given situation, to have backup if some of the mechanisms fail, to more effortlessly generate interesting and complicated variations by playing around with several interacting mechanisms, and so forth. But, assuredly, polyadjustability may also make it more difficult to tune something if the various mechanisms interact in a particularly intricate way. Polygenic control is an example of natural use of polyadjustability, where some characteristic of a biological organism is controlled by more than one gene.

Looking at the problem of reducing resistance to change from the viewpoint of psychology adds yet another perspective to the discussion, one that is concerned with systems being deliberate agents, or collections of them. In this view, the topic is more commonly referred to as *openness to new*, where "new" includes both the easier case of novel input that agrees well with agent's current worldview and the more challenging situation of input that does not.

The main problem with regard to variation generation (and to adaptivity in general) is that people and social groups have a tendency, after initial developmental period, to become quite fixed in their ways of thinking and doing. We have cognitive predispositions to confirmation bias, fallacy of centrality, hubris, normalization, typification, and bottom-up salience of cues, as well as to lock-in and fixation (Weick, 2005). Similarly, in social groups and institutions various behaviors and beliefs more or less spontaneously emerge and form the "culture of the organization", which will then create a great deal of inertia to change (Grisogono, 2005). To allow for novel variations to be introduced into such systems it is thus necessary to offset those cognitive predispositions (Weick, 2005), to induce openness to conflicting inputs (Harvey et al. 1961, page 333, as referred to by Hunt, 1966), to break the addiction to listen and accept only perspectives similar to one's own (Holley, 2005), etc. Whereas the common approach is to just inform people about how it would be better to act and then expect or require them to follow the guidelines, it would be considerably more effective to take the time and really help people (or whoever / whatever the deliberate agents are in the system of interest) break old behavioral habits in combination with establishing new ones. Also, enough psychological safety should be provided in order to combat the urge for closure and certainty. This means it should be assured that "it is much more important to be prepared to be wrong in order to learn, than to always be right (and therefore either or both risk-averse or in denial) and conversely, being prepared to 'decriminalise' others being wrong" (Grisogono and Ryan, 2007), as well as made sure that the group or organization is safe for interpersonal risk taking (speaking up, offering suggestions, critiques, expertise, advise) (Stagl et al., 2006). The habits of constantly challenging one's own thinking and being prepared to look for both confirming and contradictory evidence (Grisogono and Ryan, 2007), making explicit (even vocalizing, for particularly critical processes and decisions) the situation reviews, alternative diagnoses and plans (Weick, 2007), and being tolerant of uncertainty and responsibility (Ku, 1995, page 316) should be encouraged.

Also aimed mainly at deliberate agents is the suggestion to avoid various plans becoming too prescriptive (Holmqvist and Pessi, 2006). By using an extended understanding of what a plan is (can include blueprints, generative codes, various evolvable constraints, guidelines learned from experience, and much more) it can well be applied to most other adaptive systems, too. Having plans is, assuredly, very often beneficial, and the very process of planning itself can help a lot with understanding and solving the problem at hand. But if the plans are followed through rigidly, the adaptivity of the system in general and the generation of (unplanned) variations in particular may suffer a lot. Multiple ways of achieving plan flexibility exist. One is to just keep revising the plan dynamically, taking into account the new situational information (Burke et al., 2006). Another is to make the plans themselves somewhat loose, for example to have strategies suggesting boundaries on behavioral parameters rather than precise values (Ram and Santamaría, 1997). And finally there is a possibility to plain discard parts of the plan, or the whole of it, as deemed necessary. In group situations the latter option can be made easier by avoiding strongly binding contracts and building an ability to replace some of the planning with on-time communication (Andersen, 2003).

Ending the current list of the ways of making the modifiables easy to change, but certainly not closing the set of all possibilities, is the option of adding some form of redundancy to the system. Having multiple copies of the same components not only can increase the reliability of the whole system, but also facilitates transformability and mutability (Conrad, 1983, page 337): in addition to the straightforward potential benefit of having more elements to target with altering actions, the workings of the system do not depend critically on single components anymore and thus the unsuccessful variants of the elements do not immediately render the whole system inoperative (except in some particularly unfortunate cases of highly disruptive variants), which encourages more aggressive varying. A possibly even safer approach would be to decouple the exploration architecturally and functionally from the rest of the system. The better variants could then either directly and forcefully substitute the ones currently in effect in the main part of the system or, as suggested by Grisogono and Ryan (2007), "to work provisionally alongside established ways of doing things, without relying on them, but using the parallel system enough to identify and fix flaws with it until confidence in it grows sufficiently that users start transferring to it in preference to the previous system". Finally, taking this direction of adding redundancy and separating it from the main operational part to its logical conclusion, we reach virtual variation generation that is executed in models and simulations and thus potentially allows for particularly rapid alteration production and testing. But, surely, the use of models has various possible drawbacks as well, e.g., a less than ideal match with reality might lead to erroneous results and decisions.

Making the System Tolerant to Errors

In real life, variation generation almost inevitably produces a significant number of unfit alterations along with the acceptable ones. If those mistakes have a strong negative effect on the system, either real or imaginary (e.g., psychological problems), then the whole variation generating process may be considered undesirable and its activity reduced to minimum, with potentially dire consequences to system's adaptivity. Thus, making the system tolerant to errors is an important factor in supporting the generation of novel variations. For deliberate agents with psychological problems that might involve making them aware of the near unavoidability, or even desirability, of mistakes on the path of success, but in general it is mostly about increasing robustness, redundancy, reversibility and / or repairs, and actually also adaptivity (regardless of the slight touch of circularity that it seems to bring into our discussion) which would allow for incorporating some of the errors in a way that transforms them from mistakes into neutral or even useful features.

Robustness, as understood here, is the capacity to withstand various perturbations without needing an active, adaptive, response. It can come about in multiple ways, mostly by having the important functionality being just plain insensitive to disturbances (as in neutral variation discussed earlier), by making the critical parameters very difficult to change, or by having enough redundancy in the system so that single failures cannot eliminate important functionality. Redundancy can provide even more safety if it is implemented not by simply having multiple copies of the very same element, but by having different components with partially overlapping functionalities, because this protects better against systemic errors that affect all instances of some element type (e.g., Edelman and Gally, 2001).

The more active side of error tolerance – reversing, repairing, or adapting to mistakes – either tries to restore the pre-mistake state of the component or reorganize the system to now use what was previously considered a problem as a useful feature instead. Reversibility can be fostered, for example, by representing the targets of modification so that each modification would be a simple flip of some bit (or a switch between few alternatives), the undoing of which is relatively straightforward (except only when the rest of the system has already changed too much due to the unfit alteration and will not restore itself appropriately after reverse modification). Or, in some cases, the so-called system restore points can be occasionally created by saving the system state in a recoverable way, up to producing full back-ups every once in a while (especially before potentially dangerous modifications). Usually this would require the implementation of several special reversibility-related mechanisms, but sometimes there may also exist possibilities to achieve similar effects with less effort. An example would be to have the new variant just functionally override the previous one without actually removing it from the system immediately,

so that for recovery it would be enough to withdraw the new element and thus allow the previous one to function again. *Genic occlusion* is a natural instance of such a method: a gene is suppressed through addition of a further "upstream" gene to the epistatic set (a set of interacting genes), with no actual change to the original locus itself (Brock, 2000, page 245). As of repairing and readjusting, some options (in a military context) are listed by Unewisse and Grisogono (2007): shifting of essential tasks from damaged to undamaged elements, exploiting redundancy within system; redistribution of tasks within system, exploiting multiroled or multifunctioned elements; repair of damage, which requires the capacity to detect damage, assess and repair it, exploiting capacity for frontline repair and rapid mobilisation of logistic chains; redistributing tasks so that essential ones are done vice non-essential; compensate for the damage by changing the resources available to the system.

One has to be careful, though, with using the error tolerance increasing methods for supporting variability, because more often than not the system will also be less sensitive to the variations themselves, somewhat counteracting the expected positive effect. Occasionally the very opposite action would be beneficial instead, as illustrated by yet another example from genetics where one way to increase mutation rate (in conditions calling for higher adaptivity) is through inhibition of DNA repair processes (Hersh et al., 2004; Denamur and Matic, 2006). The latter option is particularly suitable for harsh situations where the survival of the system (usually a population) is put into considerable danger and the normal adaptational mechanisms are unlikely to be of enough help – then the high occurence of (totally) unfit variants is outweighed by the increased probability of also finding some new viable forms because the alternative would likely be an irreversible extinction of the whole system.

Choosing Suitable Representations

A large share of nontrivial systems make use of various internal representations in order to process information and store knowledge. In principle there can be a near infinite number of different representations that refer to the same "real" entities, and furthermore a near infinite number of mappings both from the referenceable set to representations and back. While equal in some ultimate respect, those alternative representations and mappings may present different practical opportunities and constraints for the system, including to the variation generation mechanisms. If the modifications executed in an adapting system target the very representations themselves, then the influence of the choice of representations on the variation generation is often obvious. But even if they do not, the representations may be important intermediaries in the chains from introduced modifications to systemic results and thus can still have a significant impact on how easy it is to produce relevant variations.

When representations are looked at as yet another kind of modifiables, then the general ideas discussed in current paper apply to them just as well as to other modifiables and are thus not repeated here. One problem worth a separate mentioning is about whether to use distributed and possibly implicit representations or not. Having "an ecology of cooperating and competing models, each partially representing some aspects" (Ryan, 2006) may help variation generation both by providing a large set of different combinable elements and possibly by making variations *emerge* even in the course of "normal" system behavior without any explicit generators in place. On the other hand, implicit, distributed and inscrutable internal representations make it difficult to use bootstrap learning processes (Provost, 2007, page 5), so the variations may remain to be generated on a very low level where it rarely leads to very complex solutions due to the vastness of search space down there. Thus some balance suitable for a given system should be searched for.

Regarding the mappings between entities and their representations, there are several issues to be paid attention to. If variation mechanisms are applied to representations (e.g., the genotype), but fitness is mainly dependent on the "real" features deriving from those representations (e.g., the phenotype), then one of the main concerns is the question of whether the representations and mappings allow the mechanisms to properly explore the phenotype space.

The first problem is coverage – which and how big parts of the phenotype are in principle derivable from the genotype. If no representations exist that lead to high-fitness phenotypes, then the variation generator cannot possibly reach them. If, on the contrary, most of the representations lead to only good solutions, then the generator is without much effort very good at producing fit variants, but only as long as the fitness landscape does not change radically with regard to what is covered. Thus in the longer perspective it would make sense to either have full coverage or, possibly even better, to have adaptive representation (or mapping) structure that keeps the coverage on high fitness areas.

Secondly, in addition to the static correspondence between genotype space and phenotype space there is also correspondence of dynamics – how does a movement in one space get reflected in the other. If the mapping is relatively straightforward (e.g., small movements of the modifiable in a certain "direction" generally produce small movements of the testable also in some certain "direction"), then variation generating mechanisms will have the possibility to guide the search in a systematic way. On the other hand, if the mapping is complicated and small changes in genotype space cause significant and difficult to predict jumps in phenotype, then the production of high diversity and large amount of novelty is made easy. Which of these is preferred depends on the particular system and / or situation. Similarly, there is a trade-off involved in the amplification factor: small movements in one space corresponding to small movements in the other makes fine-tuning easy, but small movements cor-

responding to large ones helps with the rate of exploration in the phenotype space, especially if representations are for some reason difficult to change in large steps.

The third point to be considered is somewhat related to both previous ones: would it be a good idea to have the representations together with mappings form nontrivial generative rules that produce the phenotype in a developmental, step-by-step fashion (as opposed to providing a fully detailed blueprint from which the structures are directly "copied" into reality)? If yes, then should they be deterministic or probabilistic, and context-sensitive or not (or to what extent)? The usage of generative rules can surely make the correspondence between modifiables and testables more complex and thus difficult to guide, but accordingly it can facilitate the production of novel interesting variants that would have been burdensome to explicitly encode in all detail. Then again, if the generative rules make good use of contextual information during execution, and possibly utilize self-organization, they can in principle provide valuable support in channeling the variants into high-fitness regions of solution space, with the almost inseparable flip side of reducing solution diversity. In less fortunate cases the channeling might also occur into low-fitness regions.

And the fourth interesting issue with representations is their abstractness. For example, psychology has observed that the ability to generalize (i.e., to abstract) and transfer knowledge and skills supports (or reflects) system's ability to adapt (Ployhart and Bliese, 2006), and that "greater abstractness is associated with lower stereotypy and greater flexibility in the face of complex and changing problem situations, toward greater creativity, exploration behavior, tolerance of stress, etc." (Harvey and Schroder, 1963, page 134, as referred to by Hunt, 1966). As of variability, the abstractness could be viewed as increasing the scope, or applicability, of each variant and thus reducing the number of different internal alternatives required to cover the areas of interest in phenotype and interaction space. On the other hand, though, abstract representations may be more difficult to interpret, therefore being better suited for advanced systems that possess enough processing capacity and knowledge for transforming between abstract and specific.

Providing Various Internal and External Resources

The generation of variations can also be supported by providing the corresponding mechanisms with an adequate supply of all the necessary and helpful resources. Particularly noteworthy among them are reservoirs of elements that can be used for combinatorial purposes, of prefabricated variants, of ideas, and of accumulated knowledge and experience to be used either directly or more loosely in the form of inspiration. These can be set up as, for example, repositories that can store the components or knowledge either in an explicit and ready-to-use state or also in some more implicit fashion where the full content is not readily extractable but usable nevertheless. The resource pools can also exist as secondary functions of some other subsystems, as well as be totally external. The various ways of using external resources for variation generation include obtaining / copying knowledge and ideas only, acquiring by incorporation of or by merging with external objects, and executing a more interactive process where there exists at least two-way communication between the system and external entities. The lines between these can occasionally be somewhat fuzzy, but the first one is generally thought of as taking place through system's sensory channels, while the second is likely to involve some special intake mechanism and the third can be a combination of the first two with the addition of outward communication. An obvious precondition for using external resources is the very existence of these resources in combination with them being accessible to the system. The latter could be supported by giving the system the necessary interfacing mechanisms, by having some external transportation and communication infrastructure in place, and by other, more elaborate, supportive systems.

Conclusion

Generating variations efficiently and wisely can sometimes be the key for making a system adaptive enough with regard to the goal at hand. And adaptivity, in turn, is one of the key ingredients of life and intelligence. As described in this paper, there are a lot of aspects to be paid attention to in this seemingly simple process of variation generation, and thus both further research of these issues and inventive application of the found ideas can be considered an important part of the fields of ALife and AI, as well as of the studies of Complex Adaptive Systems in general.

Acknowledgements

Thanks to Leo Mõtus for providing me the possibility to do highly unconstrained research. This work was supported in part by Research Laboratory for Proactive Technologies and Department of Computer Control in Tallinn University of Technology, Estonian Doctoral School in ICT, Estonian Information Technology Foundation, and Estonian Ministry of Education and Research (grant SF0140113As08). Conference participation was financed by ESF DoRa 8 via Archimedes Foundation.

References

Andersen, E. (2003). Genesis of an anthill: Wireless technology and self-organizing systems. *Ubiquity*, 3(49).

Ashby, W. R. (1956). *Introduction to Cybernetics*. Chapman & Hall.

Bennett, A. F. (1997). Adaptation and the evolution of physiological characters. In Dantzler, W. H., editor, *Handbook of Comparative Physiology, Vol. 12*, pages 3–16. Oxford University Press.

Brock, J. P. (2000). *The Evolution of Adaptive Systems: The General Theory of Evolution*. Academic Press.

Burke, C. S., Hess, K. P., and Salas, E. (2006). Building the adaptive capacity to lead multi-cultural teams. In Burke, C. S., Pierce, L. G., and Salas, E., editors, *Understanding Adaptability, A Prerequisite for Effective Performance within Complex Environments*, pages 175–211. Elsevier.

Conrad, M. (1983). *Adaptability. The Significance of Variability from Molecule to Ecosystem.* Plenum Press.

Denamur, E. and Matic, I. (2006). Evolution of mutation rates in bacteria. *Molecular Microbiology*, 60(4):820–827.

Edelman, G. M. and Gally, J. A. (2001). Degeneracy and complexity in biological systems. *PNAS (Proceedings of the National Academy of Sciences of the United States of America)*, 98(24):13763–13768.

Grisogono, A.-M. (2005). Co-adaptation. In Bender, A., editor, *SPIE Symposium on Microelectronics, MEMS and Nanotechnology*, pages 23–37.

Grisogono, A.-M. and Ryan, A. (2007). Operationalising adaptive campaigning. In *International Command and Control Research and Technology Symposium, 12th ICCRTS*.

Harvey, O. J., Hunt, D. E., and Schroder, H. M. (1961). *Conceptual Systems and Personality Organization*. Wiley.

Harvey, O. J. and Schroder, H. M. (1963). Cognitive aspects of self and motivation. In Harvey, O. J., editor, *Motivation and Social Interaction: Cognitive Determinants*, pages 95–133. Ronald.

Hersh, M. N., Ponder, R. G., Hastings, P. J., and Rosenberg, S. M. (2004). Adaptive mutation and amplification in Escherichia coli: two pathways of genome adaptation under stress. *Research in Microbiology*, 155(5):352–359.

Heylighen, F. (2001). The science of self-organization and adaptivity. In *The Encyclopedia of Life Support Systems*. Eolss Publishers, Oxford.

Heylighen, F. and Joslyn, C. (2001). The law of requisite variety. `http://pespmc1.vub.ac.be/REQVAR.HTML`.

Holley, J. (2005). Transforming your regional economy through uncertainty and surprise: Learning from complexity science, network theory and the field. In McDaniel Jr., R. R. and Driebe, D. J., editors, *Uncertainty and Surprise in Complex Systems*, pages 165–176. Springer.

Holmqvist, M. and Pessi, K. (2006). Agility through scenario development and continuous implementation: a global aftermarket logistics case. *European Journal of Information Systems*, 15:146–158.

Hunt, D. E. (1966). A conceptual system change model and its application to education. In Harvey, O. J., editor, *Experience, Structure & Adaptability*, pages 277–302. Springer.

Knoll, K. and Järvenpää, S. L. (1994). Information technology alignment or "fit" in highly turbulent environments: the concept of flexibility. In *SIGCPR '94: Proceedings of the 1994 computer personnel research conference on Reinventing IS: managing information technology in changing organizations*. ACM.

Ku, A. (1995). Modelling uncertainty in electricity capacity planning. Doctoral dissertation. London Business School.

Lenski, R. E., Barrick, J. E., and Ofria, C. (2006). Balancing robustness and evolvability. *PLoS Biology*, 4(12):2190–2192.

Lints, T. (2010). The essentials of defining adaptation. In *Systems Conference, 2010 4th Annual IEEE*, pages 113–116.

Menashe, D., Biham, O., Laikhtman, B. D., and Efros, A. L. (2000). Glassy properties and fluctuations of interacting electrons in two-dimensional systems. *Europhysics Letters*, 52(1):94–100.

Moorman, K. and Ram, A. (1992). A case-based approach to reactive control for autonomous robots. In *Proceedings of the AAAI Fall Symposium on AI for Real-World Autonomous Mobile Robots*.

Ployhart, R. E. and Bliese, P. D. (2006). Individual adaptability (I-ADAPT) theory: Conceptualizing the antecedents, consequences, and measurement of individual differences in adaptability. In Burke, C. S., Pierce, L. G., and Salas, E., editors, *Understanding Adaptability, A Prerequisite for Effective Performance within Complex Environments*, pages 3–39. Elsevier.

Provost, J. (2007). Reinforcement learning in high-diameter, continuous environments. Doctoral dissertation. The University of Texas at Austin.

Ram, A. and Santamaría, J. C. (1997). Continuous case-based reasoning. *Artificial Intelligence*, 90:25–77.

Ryan, A. (2006). About the bears and the bees: Adaptive responses to asymmetric warfare. In Minai, A., Braha, D., and Bar-Yam, Y., editors, *Proceedings of the Sixth International Conference on Complex Systems*. New England Complex Systems Institute.

Stagl, K. C., Burke, C. S., Salas, E., and Pierce, L. (2006). Team adaptation: realizing team synergy. In Burke, C. S., Pierce, L. G., and Salas, E., editors, *Understanding Adaptability, A Prerequisite for Effective Performance within Complex Environments*, pages 117–141. Elsevier.

Stec, B. (2004). Living and nonliving matter. *Science*, 305:41. Letter to the editor.

Unewisse, M. and Grisogono, A.-M. (2007). Adaptivity led networked force capability. In *International Command and Control Research and Technology Symposium, 12th ICCRTS*.

Weick, K. E. (2005). Managing the unexpected: Complexity as distributed sensemaking. In McDaniel Jr., R. R. and Driebe, D. J., editors, *Uncertainty and Surprise in Complex Systems*, pages 51–65. Springer.

Weick, K. E. (2007). Drop your tools: On reconfiguring management education. *Journal of Management Education*, 31(1):5–16.

Wolfram, S. (2002). *A New Kind of Science*. Wolfram Media, Champaign, IL.

Systems Aikido: A Novel Approach to Managing Natural Systems

Alexandra Penn[1], Richard Watson[1], Alexander Kraaijeveld[2],and Jeremy Webb[2]

[1]Science and Engineering of Natural Systems, School of Electronics and Computer Science, University of Southampton
Southampton, SO17 1BJ, United Kingdom
[2]School of Biological Sciences, University of Southampton,
Southampton United Kingdom
asp@ecs.soton.ac.uk

Extended Abstract

The potential of new technologies which emulate or exploit the unique properties of living systems is widely lauded. Such technologies however, create new engineering challenges which must be addressed before they can become broadly utilised (see for example, Braha et al. (2006); Bedau et al. (2010); Penn (2008)). Additionally, many pressing challenges for society today are inherently concerned with gaining a better ability to understand and manage interacting living or life-like systems upon which we rely. Well-documented examples include climate change, agricultural sustainability, city dynamics, demographic change and chronic infections. Problems in all these areas demand a better ability to manage complex biological systems than is currently available.

Conventional approaches to working with biological systems are, for the most part, brute force, attempting to effect control in an input and effort intensive manner and are often insufficient when dealing with the inherent non-linearity and complexity of living systems. Biological systems, by their very nature, are dynamic, adaptive and resilient and require management tools that interact with dynamic processes rather than inert artifacts. Our novel engineering approach which aims to exploit rather than fight those properties, presents a more efficient and robust alternative. Its essence is what I will call systems aikido, the basic principle of aikido being to interact with the momentum of an attacker and redirect it with minimal energy expenditure, using the opponents energy rather than ones own. In more conventional terms, this translates to a philosophy of equilibrium engineering, manipulating systems own self-organisation and evolution so that the evolutionarily or dynamically stable state corresponds to a function which we require.

I will discuss how we might move from this philosophy to a practical methodology for management of living systems and technologies, covering a variety of approaches: Designing-in of tools for adaptive management given unexpected indirect effects and continuous adaptation of living components; identification of appropriate points of intervention in particular systems; and methods for steering adaptive systems by altering either the fitness landscape which they experience or the attractor structure of their dynamics. Filling fitness valleys to escape local optima; expansion of basins of attraction of difficult to access, but favourable attractors and manipulating the effective level of selection within the system.

Detailed illustration is provided by a practical application: Manipulating the level of selection within bacterial biofilms, such that stable community species and genetic composition corresponds to a community function which we require(Penn et al. (2008b,a)). Different levels of selection produce particular types of community composition. Higher-level selection promotes co-operation and synergy useful for efficient bioremediation and bioproduction, whereas encouraging lower-level selection might allow us to engineer a tragedy of the commons in problematic bacterial communities. I will present methodology and results from ongoing experimental work with Psuedomonas aeruginosa biofilms in which direct or indirect manipulation of parameters affecting group structure and dispersal mechanisms modify the effective level of and hence response to selection. And will describe approaches to increase the robustness of the engineered community Finally I will contrast this methodology with a spectrum of more or less brute-force interventions, from traditional biofilm engineering approaches to imposition of higher-level selection(Swenson et al. (2000b,a); Penn (2006)).

References

Bedau, M., McCaskill, J., Packard, N., and Rasmussen, S. (2010). Living technology: Exploiting life's principles in technology. *Artificial Life*, 16:89–97.

Braha, D., Minai, A., and Bar-Yam, Y. (2006). *Complex Engineered Systems: Science Meets Technology*. Springer.

Penn, A. (2006). *Artificial Ecosystem Selection: Simulation, Theory and Experiment*. PhD thesis, School Life Sciences, Sussex University, UK.

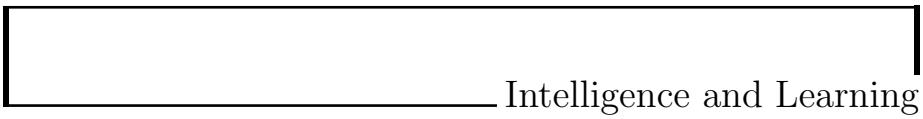
Penn, A. (2008). What can artificial life offer the development of methodologies in the field of socio-ecological sustainability? In Bullock, S., Noble, J., Watson, R., and Bedau, M., editors, *In proceedings ALife XI: Eleventh International Conference on the Simulation and Synthesis of Living Systems. Winchester, UK,*. MIT Press.

Penn, A., Powers, S., Conibear, T., Kraaijeveld, A., Watson, R., Bigg, Z., and Webb, J. (2008a). Co-operation and group structure in bacterial biofilms. In *In proceedings Society for General Microbiology, Autumn meeting, Trinity College, Dublin.*

Penn, A., Watson, R., Powers, S., Webb, J., Kraaijeveld, A., T., C., and Z., B. (2008b). Mechanisms for the initiation of multicellularity in bacterial biofilm. In Bullock, S., Noble, J., Watson, R., and Bedau, M., editors, *In proceedings ALife XI: Eleventh International Conference on the Simulation and Synthesis of Living Systems. Winchester, UK*. MIT Press.

Swenson, W., Arendt, J., and Wilson, D. (2000a). Artificial selection of microbial ecosystems for 3-chloroaniline biodegradation. *Environ. Mirobiol.*, 2:9365.

Swenson, W., Wilson, D., and Elias, R. (2000b). Artificial ecosystem selection. *PNAS*, 97:9110.


Intelligence and Learning

Language Change across Generations for Robots using Cognitive Maps

Ruth Schulz[1], Gordon Wyeth[2] and Janet Wiles[1]

[1] School of Information Technology and Electrical Engineering,
The University of Queensland
[2] Queensland University of Technology
ruth@itee.uq.edu.au

Abstract

Languages change over time, as new words are invented, old words are lost through disuse, and the meanings of existing words are altered. The processes behind language change include the culture of language acquisition and the mechanisms used for language learning. We examine the effects of language acquisition and learning, in particular the length of the learning period over generations of robots. The robots form spatial concepts related to places in an environment: toponyms (place names) and simple prepositions (distances and directions). The use of spatial concepts allows us to investigate different classes of words within a single domain that provides a clear method for evaluating word use between agents. The individual words used by the agents can change rapidly through the generations depending on the learning period of the language learners. When the learning period is sufficiently long that more words are retained than invented, the lexicon becomes more stable and successful. This research demonstrates that the rate of language change depends on learning periods and concept formation, and that the language transmission bottleneck reduces the retention of words that are part of large lexicons more than words that are part of small lexicons.

Introduction

Language change is a ubiquitous property of natural languages. One characteristic of language change is the production of neologisms, with new words created or existing words modified, combined, or separated (Brinton & Traugott, 2005). A shared language can be sustained within generations, while the words and concepts may change through generations. Although older generations are prone to deplore the language of younger generations, language change only becomes a problem when members of a population are no longer able to understand each other (Aitchison, 1991).

There are three timescales on which language change occurs: individual learning, cultural transmission, and biological evolution (Kirby, Dowman, & Griffiths, 2007). Language change is driven by both external sociolinguistic and internal psycholinguistic factors (Aitchison, 1991). Constraints that shape language include sensorimotor factors (the noisiness and variability of signals), cognitive limitations (learning, processing, and memory), thought (concepts and categorization), and pragmatic constraints (Chater & Christiansen, 2009). Language acquisition mechanisms influence the nature of language change (Niyogi, 2006), with the transmission of language from one generation to the next involving the mechanisms of language learning and production (Brighton, Smith, & Kirby, 2005).

Representation and culture influence the concepts that can be formed in a language and the ease with which agents learn these concepts. These factors are part of concept formation, language production, and language acquisition mechanisms. Together with learning mechanisms, representation affects how individual agents form concepts, which in turn affects the concepts that form in a population of agents. The cultural environment of the agents determines the words and concepts that agents are exposed to over their lifetimes.

A variety of representations and learning mechanisms have been used in studies investigating language evolution in computational agents. Recent studies have investigated the use of visual perceptions and spatial representations in forming a language for regions in geographical space and generative grounding using spatial representations (Schulz, Prasser, Stockwell, Wyeth, & Wiles, 2008). When agents ground concepts generatively, by combining existing concepts to form new concepts, there is increased flexibility and hence also ambiguity in the association between words and concepts.

In generational studies, agents start afresh with each new generation, learning the existing language and potentially expanding it. A reason that language is evolvable is that it is situated in a cultural environment that aids learning through generations, which can be implemented with iterated learning (Brighton, et al., 2005; Kirby & Hurford, 2002), in which agents learn language from the utterances of other agents. The strategies used by language speakers and hearers in determining what to talk about and how to talk about it are also a part of culture.

One feature of culture that has been studied previously is the bottleneck of language transmission (Brighton, et al., 2005; Kirby, 2002; Smith, 2007; Tonkes & Wiles, 2002). The bottleneck has been found to be important for the development of compositional and productive language. Previous spatial language studies have investigated how the rate at which agents enter and leave the population affected whether the agents were able to sustain a shared spatial language (Bodík & Takáč, 2003). These studies found that when the length of time agents spent in the population was sufficiently long (i.e. the bottleneck was sufficiently large), a shared spatial language was able to be sustained. These results

have also been found in language studies with arbitrary feature representations (Smith, 2007).

Studies investigating the language transmission bottleneck have either considered a single class of words or analyzed the success of the whole language, with individual words used as examples. However, different classes of words, such as nouns and prepositions, play different roles in meaningful communication, and all classes of words may not be equally likely to pass through a language bottleneck.

The challenge for this project is to determine how spatial languages can change through generations and to determine how the length of the learning period and lifetime of the agents affect language change. The main questions to answer include how to interpret spatial language change over generations and whether different types of spatial words have different rates of change. In particular we are interested in how learning by successive generations affects the turnover of individual words. The study described in this paper investigated the effect of the length of the learning period and the lifetime of the agents on the various spatial concepts that form and how the language changes throughout the generations.

A Spatial Language with Cognitive Maps

In language studies, the agent interactions influence the words and concepts that a language agent is exposed to and chooses to use throughout its lifetime. The specific games played determine which niches of concept space will be filled and the words chosen by the agents determine which words will survive through generations. In the study presented here, generations of simulated robots played language games to form concepts for toponyms (place names) and simple prepositions (directions and distances). The length of each generation was varied from four interactions per generation up to 1000 interactions per generation to investigate the affect of the length of the learning period and agent lifetimes on language change. The nature of the language change was investigated by comparing rates of word invention, retention, and persistence for the different concept types of toponyms, directions, and distances.

Location Language Games

The language games used in these studies are location language games (see Figure 1). To play a location language game, the agents require a representation of the world acquired through exploration carried out independently of other agents in the world. Shared attention for location language games is co-presence, that is, the agents are within hearing distance. While autonomously exploring the world, the agents intermittently send a "Hello" signal. If a "Hello" signal is heard, the hearing agent sends a "Hear" signal and the agents play a game. After shared attention is established, the speaker chooses a topic, which in a location language game relates to the current location of the agents or a location at a distance from the agents, depending on the game being played. After the topic is determined, the speaker uses its lexicon to determine which word should be used in the current situation and produces an utterance. Both agents then update their representations and lexicon. In the location language

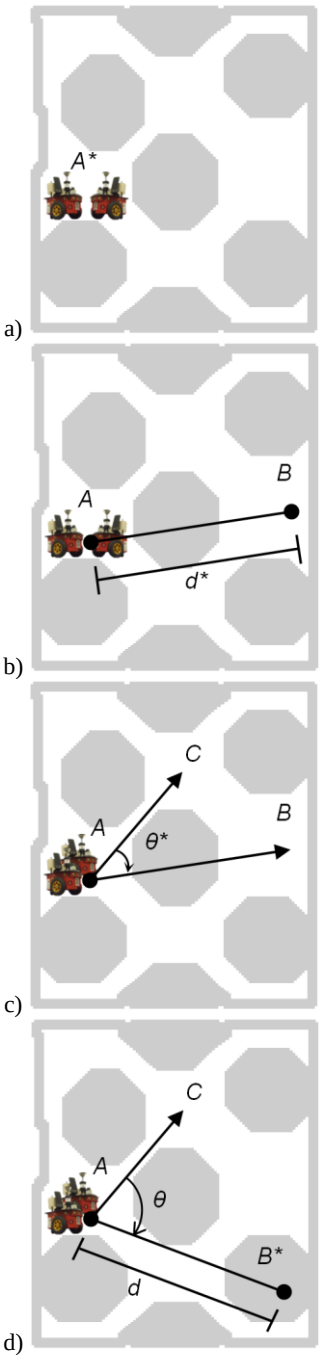


Figure 1. Referents used in the language games: a) The *where-are-we* game involves a single location: the current location, *A*, of both robots. b) The *how-far* game involves two locations (current, *A*, and target, *B*) and a distance, *d*. c) The *what-direction* game involves three locations (current, *A*, target, *B*, and orientation, *C*) and a direction, θ. d) The *where-is-there* game involves three locations (current, *A*, target, *B*, and orientation, *C*), a direction, θ, and a distance, *d*. The figures show the robots located in the open plan office of the simulation world, with gray lines representing walls and gray octagons representing desks. A star (*) indicates that the speaker may invent a new word and that both agents will update their lexicon for the marked word.

games played in this study the hearer receives the utterance and updates their representations, but does not explicitly evaluate the speaker's utterance and no feedback is given to either agent. Repeated encounters enable coherent languages to form even without explicit feedback (a phenomenon reported in a variety of studies including Smith, 2007; and Vogt, 2004).

In the study, the agents played *where-are-we*, *how-far*, *what-direction*, and *where-is-there* games. The premise of a *where-are-we* game is a location language game where the topic is the current location of the agents (see Figure 1a). The speaker produces a word for the current location and both agents update their lexicon based on the speaker's utterance.

The *how-far* game is based on naming two locations: Both agents are located at the first location (A) and they talk about the second location (B), specifying the distance between the two locations (see Figure 1b).

The *what-direction* game is based on naming three locations: As in the *how-far* game, both agents are located at the first location (A) and they talk about the second location (B). The agents are both facing the third location (C), and the direction between the two distant locations is specified (see Figure 1c).

The *where-is-there* game, adapted from previous spatial language games (Bodík & Takáč, 2003; Steels, 1995), extends the *how-far* and *what-direction* games and is based on naming three locations, as specified in the *what-direction* game (see Figure 1d). The agents describe the relationship between the locations with spatial words of distance and direction. The *where-is-there* game is interesting because it allows the grounding of toponyms relative to existing toponyms, and therefore allows agents to refer to places that they have never visited or can never visit.

Cognitive Map

To build a representation of the world, the simulated robots used RatSLAM, a method of Simultaneous Localization And Mapping (SLAM) that has been developed over the past decade to enable autonomous robots to explore and map their environments (Milford & Wyeth, 2007). RatSLAM is a computational model inspired by the rodent hippocampal complex. Through exploration of an environment, each robot constructs a unique representation of the world as a topological map of experiences, each with an estimate of global pose within an approximate *x-y* representation of the world. An active experience encodes the robot's best estimate of its position (for more information see Milford, Schulz, Prasser, Wyeth, & Wiles, 2007). The experience map provides a cognitive map representation of the world (O'Keefe & Nadel, 1978).

A simulation world was built to mirror the real world, with images from the real world used in constructing the views of the robot. The simulation world includes an open plan office in a university building. Exploration was performed by left and right wall following. The robots used a single forward facing camera. In real-world studies, language games between real robots were based on actual hearing distances (Schulz, Wyeth, & Wiles, submitted). The study in this paper was completed in the simulation world for computational tractability. The simulation world enables simulated robots to pass messages to other robots within a set distance of their current locations, allowing the hearing distance to be explicitly set. For the study reported here a hearing distance of 3m was used.

Toponymic Lexicon

The associations between experiences and words are stored in distributed lexicon tables, a method inspired by the distributed nature of inputs to neural networks combined with the lexicon table structure (Schulz, et al., 2008). Forming concepts with a distributed lexicon table differs from most other conceptualization methods in that it is directly linked to the language formation, allowing concepts and words to have boundaries that are not explicitly defined. In many language game studies, concepts are formed using discrimination trees (Bodík & Takáč, 2003; Smith, 2007; Steels, 1997), which allows the agents to form concepts with well defined boundaries. The discrete concepts, formed through a discrimination tree or similar categorization method, may then be associated with words through a lexicon table. With a distributed lexicon table, concept formation and association with words occurs concurrently by increasing associations between experiences and words. An association value is stored for each experience–word pair, which is a value of 0.0 or greater. Experiences are related to each other by their proximity, based on their global pose estimates. The association between an experience and a word is strengthened when they are used together.

The toponymic lexicon data structures include the toponym lexicon, the toponym lexicon table, and toponym associations. The toponym lexicon comprises the set of words used as toponyms where each word is a unique string of consonants and vowels. The toponym lexicon table comprises a set of toponym associations between experiences and words.

In both the *where-are-we* and *where-is-there* games, the toponym association value for the specified experience and the word used is incremented by 1.0. A word for a location is chosen by the speaker in both the *where-are-we* and *where-is-there* games. For a specified location the word with the highest confidence value is chosen. The confidence value, h_{ij}, at the experience, *i*, for the word, *j*, is the relative association of the word within a neighborhood of size *D* compared to the total association of the word, calculated as follows:

$$h_{ij} = \frac{\sum_{k=1}^{X} a_{kj}^{T} (D^{T} - dist_{ki}^{T}) / D^{T}}{\sum_{m=1}^{E} a_{mj}^{T}} \quad (1)$$

where *X* is the number of experiences within *D* of the experience, *i*; a^{T}_{ij} is the association between an experience, *i*, and the word, *j*; $dist^{T}_{ki}$ is the distance between experiences, *k* and *i* within the experience map of the robot; and *E* is the total number of experiences in the robot's experience map. For the study presented here a neighborhood size, *D*, of 3m was used. In each interaction, words are invented with probability, *p*, as follows:

$$p = \exp\left(\frac{-h_{ij}}{(1 - h_{ij})T} \right) \quad (2)$$

where h_{ij} is the confidence value of the experience-word combination; and T is a scaling parameter called the temperature, which effectively sets the invention rate for new words. Eq. 2 allows agents to use existing words when a word is associated with the current location with a high confidence, and to probabilistically invent words otherwise. Varying the temperature alters the rate of word invention, where a higher temperature increases the probability of inventing a new word. For the study presented here the temperature was decreased linearly from 0.3 to 0.1 over the course of each generation.

Relational Lexicon

In addition to locations, the simulated robots have words for directions and distances. The data structures include the distance and direction lexicons, elements, associations, and lexicon tables. The distance lexicon comprises the set of words used to refer to distances, and the distance lexicon table comprises a set of distance associations between distance elements and words. Each distance element is a distance measured in meters in global pose space.

Direction words used data structures similar to those for distance words. The direction lexicon comprises the set of words used to refer to directions (i.e. angular distances), and the direction lexicon table comprises a set of direction associations between direction elements and words. Each direction element is an angle measured in radians.

In each *how-far* game, the association values stored in the distance lexicon for the distance word used are updated. Experiences are grouped to the nearest distance element based on their distance from the current experience in global pose space. For the topic, j, of the interaction, a distance association value, a^D_{ij}, is calculated for each distance element, $i \in 1..K^D$, by summing the target toponym associations for each experience grouped to that distance element, and smoothing using a distance neighborhood, as follows:

$$a^D_{ij} = \sum_{m=1}^{Y} \frac{\left(\sum_{k=1}^{X} a^T_{kw}\right)\left(D^D - dist^D_{mi}\right)}{D^D} \quad (3)$$

where Y is the number of distance elements within a neighborhood of size D^D from the distance element, i; X is the number of experiences grouped to the distance element, i; a^T_{kw} is the toponym association between the experience, k, and the toponym, w; and $dist^D_{mi}$ is the distance between the two distance elements, m and i. For the studies reported here, 50 distance elements were used in the range 0 to 25m and a distance neighborhood of 1.5m was used.

In each *what-direction* game, the association values stored in the direction lexicon for the direction word used are updated. Experiences are grouped to the nearest direction element based on the direction from the agent's facing at the current experience. For the topic, j, of the interaction, a direction association value, a^Θ_{ij}, is calculated for each direction element, $i \in 1..K^\Theta$, by summing the target toponym associations for each experience grouped to that direction element, and smoothing using a direction neighborhood, as follows:

$$a^\Theta_{ij} = \sum_{m=1}^{Y} \frac{\left(\sum_{k=1}^{X} a^T_{kw}\right)\left(D^\Theta - dist^\Theta_{mi}\right)}{D^\Theta} \quad (4)$$

where Y is the number of direction elements within a neighborhood of size D^Θ from the direction element, i; X is the number of experiences grouped to the direction element, i; a^T_{kj} is the toponym association between the experience, k, and the toponym, w; and $dist^\Theta_{mi}$ is the angular distance between the two direction elements, m and i. For the studies reported here, 50 direction elements were used in the range 0 to 2π, and a direction neighborhood of $3\pi/25$ (21.6°) was used.

For distances and directions, the word with the closest match to the current distance or direction concept is used. The probability of inventing spatial words is calculated as for the toponyms using the match, $match_{ij}$, between the normalized vectors of the calculated, i, and stored, j, spatial associations, in place of the confidence value, calculated as follows:

$$match_{ij} = \sum_{k=1}^{K} \min\left(\frac{a_{ki}}{\sum_{m=1}^{K} a_{mi}}, \frac{a_{kj}}{\sum_{m=1}^{K} a_{mj}}\right) \quad (5)$$

where K is the number of spatial elements; a_{ki} is the association for the spatial element, k, and the topic, i, calculated using Equation 3 or 4; and a_{kj} is the association stored in the lexicon table for the spatial element, k, and spatial word, j.

Evolving Spatial Languages

In the study described in this paper, agent populations evolved languages over generations of agents. Generations consisted of a set number of interactions. In the initial population two agents played negotiation games. In subsequent generations, the older agent was replaced by a new agent. The new agent was the hearer (student) in all language games. When the new agent replaced the older agent in the following generation, all language games were played as the speaker (teacher). Note that the agents do not have fitness awarded and do not compete to be part of the next generation. There are always two agents per generation, with the older agent coming from the previous generation and the younger agent forming the next generation. In this view of language change, evolution refers to the change in the language rather than to the agents. Note that this use of evolution is consistent with its original Darwinian meaning as "descent with modification". Language change under this definition does not require direct competition of elements, rather it requires generations through which it is propagated, with features of the language affected by the generational transmission process.

The order in which concepts are formed by the agent can be constrained by the games played by the agent and the concepts chosen to be used in each game. In this study, the agents play *where-are-we* games initially to allow the separate formation of a set of toponyms then play *how-far* and *what-direction* games to form a set of relational terms and finally play *where-is-there* games. Agents play *where-are-we* games in all of the interactions of the generation, playing only *where-are-we* games for the first half of the interactions. In the third quarter of the interactions, agents may also play *how-far* and *what-*

direction games with equal probability, with the constraint that the agent must have at least two toponyms in order to play a *how-far* game and at least three toponyms to play a *what-direction* game. In the final quarter of interactions, the agents may also play *where-is-there* games, with the constraint that the agent must have at least one distance and one direction word.

The Language Bottleneck

The language transmission bottleneck refers to limited transmission of a language between generations. During its lifetime, a student may not be exposed to the entire lexicon of its teacher, or even when exposed to words, will learn its own grounded meaning and therefore will not perfectly learn the teacher's language. In this simulation study, the language bottleneck is due to limits on both the number of interactions per generation, and also the number of locations in the world where the agents interact. The student must therefore generalize from its experience of the teacher's language. How well the student can generalize depends on the number of interactions and the distribution of locations at which the interactions take place. The number of interactions per generation determines the proportion of the teacher's language that the student experiences during its lifetime. An initial investigation was performed with nine conditions based on 4, 8, 16, 32, 64, 128, 250, 500, and 1000 interactions per generation. The study comprised three runs of each condition with 20 generations per run.

The size of the language increased as the number of interactions per generation increased (see Table 1). The size of each lexicon differed, with larger toponym lexicons and smaller distance lexicons for more than 16 interactions per generation. For 4, 8, and 16 interactions per generation the direction lexicon was the smallest of the three lexicons. For each of the types of words (toponyms, distances, and directions), there was a crossover between more words invented per generation and more words retained per generation (see Figure 2). The crossover point indicates the number of interactions per generation where the language transmission bottleneck is sufficiently wide that more words are retained than invented. If a student learns a comprehensive language from its teacher, then proportionately fewer words will need to be invented in the next generation.

Language Change across Generations

For the conditions in which more words were preserved than invented, the language change can be investigated further. The three conditions considered further were a) 250, b) 500, and c) 1000 interactions per generation. The study comprised three runs of each condition to 20,000 interactions, consisting of a) 80, b) 40, and c) 20 generations.

In all three conditions, the simulated robots formed a shared set of toponyms, distances, and directions. The number of words in the lexicon of each agent for each type of word increased rapidly over the first few generations, and agents in all conditions continued to invent words for toponyms, distances, and directions throughout their lifetimes. The invention of words occurred at different rates in each

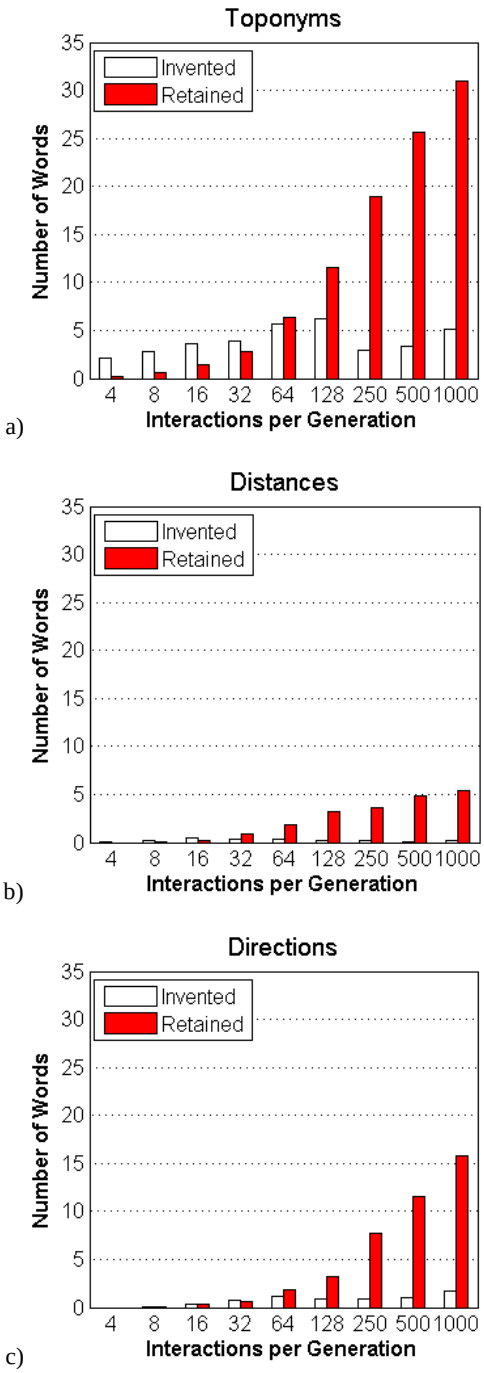


Figure 2. Words invented and retained per generation for each condition (4, 8, 16, 32, 64, 128, 250, 500, and 1000 interactions per generation) for a) Toponyms, b) Distances, and c) Directions. For each condition, the number of words invented and retained in each generation was averaged over the final ten generations of the three runs. Note the crossover between more words invented and more words retained occurs between 32-64 interactions per generation for toponyms (a) and directions (c), and 16-32 interactions per generation for distances (b).

condition and concept type (see Figure 3), with word loss closely matching word invention after the initial spurt of invention. The persistence of words in the lexicon through generations can be measured by considering when the words used in the final generation were initially invented. If a large proportion of the words were invented in earlier generations, then the words are persistent and the lexicon is stable. The persistence of words varied over the conditions and the concept types (see Figure 4).

Table 1. Average toponym, distance, and direction lexicon size over generations 11 to 20

Interactions per generation	Lexicon Size (mean (standard deviation))		
	Toponym	**Distance**	**Direction**
4	4.6 (1.1)	0.3 (0.5)	0.0 (0.0)
8	6.3 (1.5)	0.6 (0.8)	0.1 (0.4)
16	8.6 (2.2)	1.3 (0.8)	1.1 (0.7)
32	10.6 (1.7)	1.8 (0.6)	2.1 (1.1)
64	17.6 (2.8)	2.5 (0.8)	4.2 (1.2)
128	23.5 (4.4)	3.7 (0.5)	5.4 (1.5)
250	24.1 (3.1)	4.0 (0.7)	9.4 (1.3)
500	31.9 (3.2)	5.0 (0.5)	13.6 (1.7)
1000	40.4 (5.9)	5.8 (0.7)	19.3 (2.5)

Discussion

Learning with culture is different to inventing language from scratch. Agents begin their lives by learning words from older agents, and can later choose to use these words or invent new words. As agents start afresh in every generation, words that are no longer used do not remain in the lexicon. A change in language over time where one word or structure replaces another does not mean that the original is directly replaced by its replacement. Rather there may be an intermediate state in which either the old or the new word or structure may be chosen (Brinton & Traugott, 2005). In the studies presented here, an agent can learn a word for a location, but probabilistically also can invent a new word for the same location, while retaining representations for the old word.

The results show that a major effect of the length of the learning period was on the size of the resulting lexicon for the toponyms and the simple prepositions of distances and directions. The number of words used increased with the number of interactions per generation, as each agent had more interactions in which to learn the existing lexicon and invent new words. With shorter generations, the agents do not play a sufficient number of language games for a stable shared language to emerge.

The size of each lexicon is due to several factors, including the space of possible concepts, the neighborhood size used when choosing the appropriate word, the temperature used to set the probability of word invention and the opportunities to use words from that lexicon. The space of possible concepts is the size of the world for location and distance concepts and all directions for direction concepts. The neighborhood size for each word type is currently set to 3m for location concepts, 1.5m for distance concepts and $3\pi/25$ (21.6°) for direction concepts. The opportunities to use the words are in the number of games of each type played.

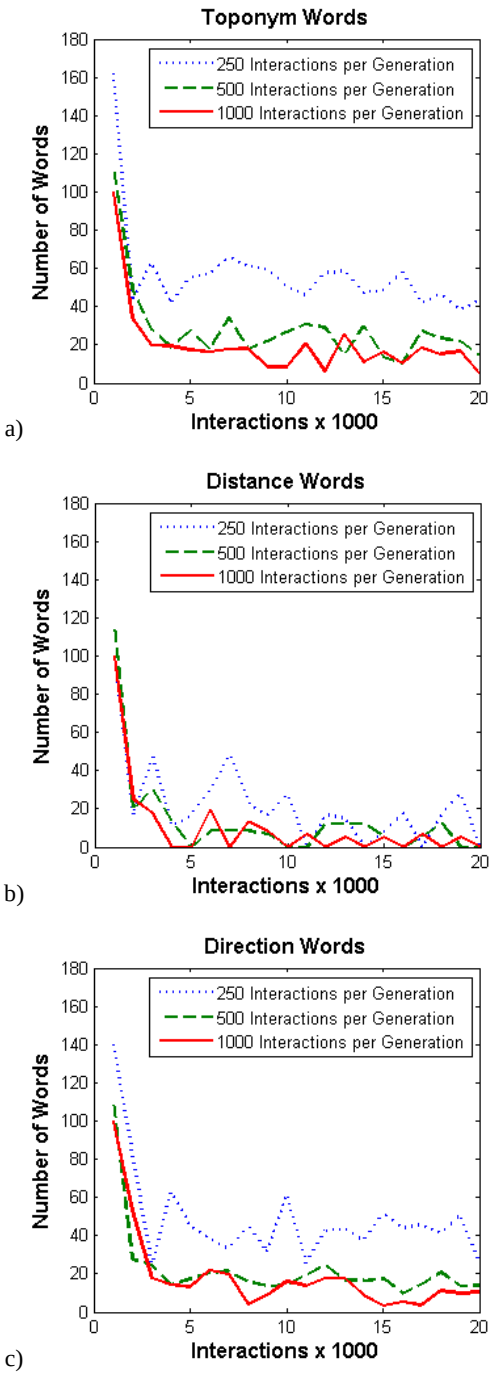


Figure 3. Words invented per 1000 interactions for the three conditions for a) toponyms, b) distances, and c) directions, averaged over all runs for each condition. In all conditions the word invention rate began high as the agent's lexicons developed over the first few generations. Distance words were more stable than direction words and toponyms, with fewer words invented and lost in each generation. The word invention rate for each type of word stabilized at a higher rate with a smaller number of interactions per generation.

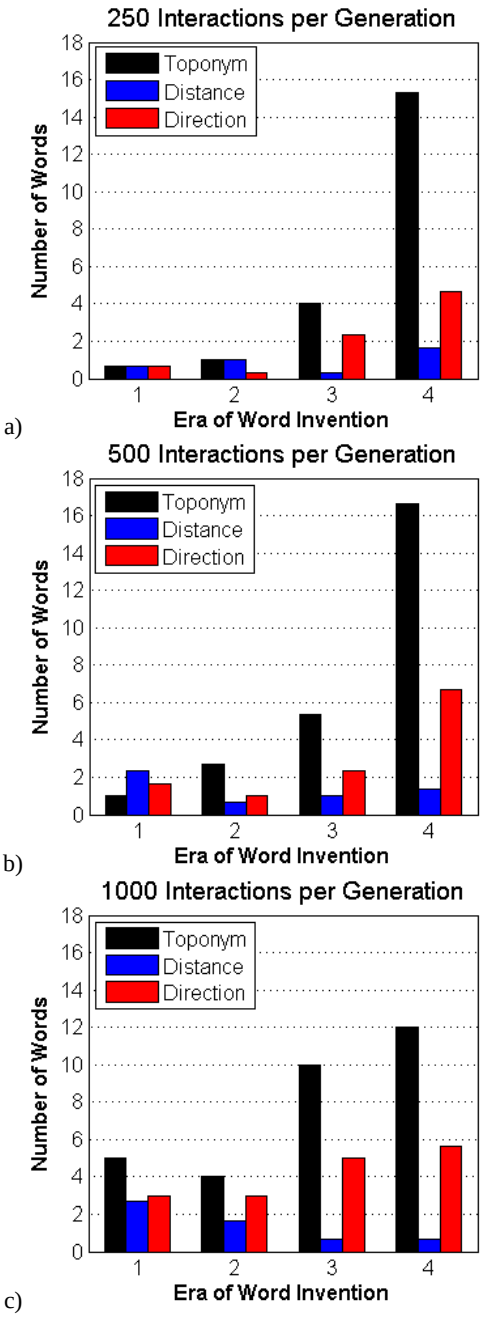


Figure 4. Word age in the final generation. The words used in the final generation are clustered into four eras based on the interaction in which each word was first used: 1. the early era (interactions 1 to 5,000), 2. the early-middle era (interactions 5,001 to 10,000), 3. the middle-late era (interactions 10,001 to 15,000), and 4. the late era (interactions 15,001 to 20,000). (a) For 250 interactions per generation few words were retained from earlier generations. (b) For 500 interactions per generation a higher proportion of distance words were retained from earlier generations. (c) For 1000 interactions per generation as well as retaining a higher proportion of distance words from earlier generations, the direction words in the final generation were invented more evenly across the generations, and a higher proportion of toponym words were invented in later generations.

With small numbers of interactions per generation, the small size of each lexicon is due to insufficient opportunities to play games that involve all possible locations. With larger numbers of interactions per generation, there is a trend towards a large toponym lexicon and a small distance lexicon. Smaller lexicons form when there is no noise in transmission and therefore no concepts that cover the same region in concept space. Larger lexicons form when the full concept space is covered. The main reason for the small size of the distance lexicon is likely to be that the size of the world has constrained the possible distances referred to by the agents. Increasing the size of the directly experienced world would result in the formation of a greater number of location and distance concepts. Direction concepts are restricted to one full rotation.

As shown by Smith (2007) and Bodík & Takáč (2003) a stable shared language can emerge in each longer generation, but the meaning of words may shift over generations, with new words entering the lexicon and old words forgotten. Bodík & Takáč (2003) found that more specific terms change faster than more general terms. If words enter and leave a language stochastically, the effect of the bottleneck would be the same for different classes of words. An alternative hypothesis is that unambiguous or frequently used words would pass through the bottleneck more easily than ambiguous or infrequent words. In the studies, we found differential rates of transmission for different classes of concepts, and saw the influence of the language transmission bottleneck on languages formed in conditions with both small and large numbers of interactions per generation.

The distance words were found to be more stable throughout the generations than the direction words. The stability of the words may be due in part to the smaller size of the distance lexicon. However, we conjecture that an equally important reason for more stable distance words is that compared to direction words, the creation of distance words is less noisy with only two toponyms used rather than three, and therefore their use is more reliable.

For the conditions explored in this study, in which word retention is higher than word invention, the bottleneck of language transmission is still evident in the trends for word age across the conditions and types of words. Proportionately more words were invented in later generations for all condition and concept types except for distance words in the conditions of 500 and 1000 interactions per generation. In these conditions, the early distance words pass through the bottleneck unchanged. In all other conditions and word types, the language transmission bottleneck reduces the retention of words through generations of agents.

As discussed in the introduction, a variety of factors have been identified as contributing to language change (for example, see Aitchison, 1991; Kirby, et al., 2007; Niyogi, 2006). Some factors contributing to language change have been demonstrated in the studies presented here. The size of the lexicon was affected by the social interactions and the period of individual language learning, and the rate of change for different concept types was affected by the concept formation for each word type. We have shown that learning periods and concept formation affect the rate at which words are retained, invented, and lost from the lexicon of the agent population. The key contribution of this research is a

demonstration of the impact of language acquisition (in the form of individual language learning, concept formation, and social interactions) on language change, in particular showing that the bottleneck of language transmission can still affect word retention between generations even when a stable shared language forms within each generation.

Acknowledgements

We thank members of the RatSLAM team and colleagues in the Thinking Systems project. This research was supported by Australian Research Council grants to GW and JW.

References

Aitchison, J. (1991). Language change: progress or decay? (2nd ed.). Cambridge: Cambridge University Press.

Bodík, P., & Takáč, M. (2003). Formation of a common spatial lexicon and its change in a community of moving agents. In B. Tessem, P. Ala-Siuru, P. Doherty & B. Mayoh (Eds.), Frontiers in Artificial Intelligence and Applications: Eighth Scandinavian Conference on Artificial Intelligence SCAI'03 (pp. 37-46). Amsterdam: IOS Press Inc.

Brighton, H., Smith, K., & Kirby, S. (2005). Language as an evolutionary system. Physics of Life Reviews, 2(3), 177-226.

Brinton, L. J., & Traugott, E. C. (2005). Lexicalization and language change. Cambridge, UK: Cambridge University Press.

Chater, N., & Christiansen, M. H. (2009). Language Acquisition Meets Language Evolution. Cognitive Science.

Kirby, S. (2002). Learning, bottlenecks and the evolution of recursive syntax. In E. J. Briscoe (Ed.), Linguistic Evolution through Language Acquisition: Formal and Computational Models: Cambridge University Press.

Kirby, S., Dowman, M., & Griffiths, T. L. (2007). Innateness and culture in the evolution of language. Proceedings of the National Academy of Sciences, 104(12), 5241-5245.

Kirby, S., & Hurford, J. R. (2002). The emergence of linguistic structure: an overview of the iterated learning model. In A. Cangelosi & D. Parisi (Eds.), Simulating the Evolution of Language (pp. 121-148). London: Springer Verlag.

Milford, M., Schulz, R., Prasser, D., Wyeth, G., & Wiles, J. (2007). Learning spatial concepts from RatSLAM representations. Robotics and Autonomous Systems - From Sensors to Human Spatial Concepts, 55(5), 403-410.

Milford, M., & Wyeth, G. (2007). Spatial mapping and map exploitation: A bio-inspired engineering perspective Spatial Information Theory (pp. 203-221). Berlin: Springer.

Niyogi, P. (2006). The computational nature of language learning and evolution. Cambridge, Massachusetts: MIT Press.

O'Keefe, J., & Nadel, L. (1978). The hippocampus as a cognitive map. New York: Oxford University Press Inc.

Schulz, R., Prasser, D., Stockwell, P., Wyeth, G., & Wiles, J. (2008). The formation, generative power, and evolution of toponyms: Grounding a spatial vocabulary in a cognitive map. In A. D. M. Smith, K. Smith & R. Ferrer i Cancho (Eds.), The Evolution of Language: Proceedings of the 7th International Conference (EVOLANG7) (pp. 267-274). Singapore: World Scientific Publishing.

Schulz, R., Wyeth, G., & Wiles, J. (submitted). Lingodroids: Grounding spatial language in a cognitive map.

Smith, A. D. M. (2007). Language change and the inference of meaning. In C. Lyon, C. Nehaniv & A. Cangelosi (Eds.), Emergence of Communication and Language (pp. 323-337). London: Springer.

Steels, L. (1995). A self-organizing spatial vocabulary. Artificial Life, 2(3), 319-332.

Steels, L. (1997). The origins of syntax in visually grounded robotic agents. In M. Pollack (Ed.), Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97) (Vol. 2, pp. 1632-1641). San Francisco, California: Morgan Kauffman Publishers.

Tonkes, B., & Wiles, J. (2002). Methodological issues in simulating the emergence of language. In A. Wray (Ed.), The Transition to Language. Oxford: Oxford University Press.

Vogt, P. (2004). Minimum cost and the emergence of the Zipf-Mandelbrot law. In J. Pollack, M. A. Bedau, P. Husbands, T. Ikegami & R. A. Watson (Eds.), Artificial life IX: Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems: MIT Press.

Aging in the Emergence of Linguistic Categories

A. Mukherjee[1], A. Baronchelli[2], V. Loreto[1,3], A. Puglisi[4,3] and F. Tria[1]

[1]Complex Networks Lagrange Laboratory, ISI Foundation, Torino, Italy
[2]Departament de Fisica i Enginyeria Nuclear, Univ. Politecnica de Catalunya, Barcelona, Spain
[3]Sapienza University of Rome, Physics Department, Rome, Italy
[4]CNR-INFM-SMC, Rome, Italy

Extended Abstract

Categories are fundamental to recognize, differentiate and understand the environment. They are meant to provide a coarse-grained description of the world we perceive. For instance, few "basic color terms", present in natural languages, coarse-grain the infinite number of different colors that humans can possibly perceive. An important question is whether categories are a manifestation of an underlying structure of nature or an emergent property of the complex interactions among individuals themselves as well as with the environment. The current work attempts to seek for an answer to this question by modeling a population of individuals who co-evolve their form-meaning repertoire by playing elementary language games.

The Category Game is a computational model designed to investigate how a population of individuals can develop a shared repertoire of linguistic categories, i.e. co-evolve their own system of symbols and meanings, by playing elementary language games (Puglisi et al., 2008). Consensus is reached through the emergence of a hierarchical category structure made of two distinct levels: a basic layer, responsible for fine discrimination of the environment, and a shared linguistic layer that groups together perceptions to guarantee communicative success. The only parameter of the model is the Just Noticeable Difference (JND) of the agents defined as the average detectable difference between two stimuli. Remarkably, the number of linguistic categories turns out to be finite and small, as observed in natural languages, even in the limit of an infinitesimally small JND.

The Category Game also allowed to focus on the question of the origins of *universal* categorization patterns across cultures. In this framework, it has recently been possible to reproduce the outcomes of the World Color Survey (WCS) (Baronchelli et al., 2010). Through the Category Game model, a certain number of non-interacting populations has been simulated, each one developing its own synthetic language. Universal categorization patterns have been discovered among populations whose individuals are endowed with the human JND function, describing the resolution power of the human eye to variations in the wavelength of the incident light (Long et al., 2006). It turns out that a simple perceptual constraint shared by all humans, namely the human Just Noticeable Difference (JND), is sufficient to trigger the emergence of universal patterns that unconstrained cultural interaction fails to produce.

A wide open question about the emergence of linguistic categories, and more generally of shared linguistic structures, concerns the role of the timescales. How to reconcile the apparent static character of most of the linguistic structures we learn with the evidences of a fluid character of modern communication systems? Here we report about preliminary studies that suggest how the structure of linguistic categories undergoes *aging* (Henkel et al., 2006): at relatively early stages changes are very frequent but they become progressively more rare as the system ages; a phenomenon whose intensity increases with the population size. From this point of view shared linguistic conventions would not emerge as attractors of a language dynamics, but rather as metastable states.

References

Baronchelli, A., Gong, T., Puglisi, A., and Loreto, V. (2010). Modelling the emergence of universality in color naming patterns. *PNAS*, 107:2403–2407.

Henkel, M., Pleimling, M., and Sanctuary, R. (2006). Statistical physics of ageing phenomena and the glass transition. *Journal of Physics: Conference Series*, 40(1).

Long, F., Yang, Z., and Purves, D. (2006). Spectral statistics in natural scenes predict hue, saturation, and brightness. *PNAS*, 103(15):6013–6018.

Puglisi, A., Baronchelli, A., and Loreto, V. (2008). Cultural route to the emergence of linguistic categories. *PNAS*, 105:7936.

The Evolution of Behavioural and Linguistic Skills to Execute and Generate Two-word Instructions in Agents Controlled by Dynamical Neural Networks

Elio Tuci, Tomassino Ferrauto, Gianluca Massera and Stefano Nolfi

Institute of Cognitive Sciences and Technologies, ISTC-CNR, Rome, Italy
{elio.tuci, tomassino.ferrauto, gianluca.massera, stefano.nolfi}@istc.cnr.it

Abstract

This paper illustrates an agent-based simulation model focused on the acquisition of linguistic skills. Populations of simulated agents controlled by dynamical neural networks are trained by artificial evolution to perfom two tasks: the behaviour-production task which consists in accessing and executing linguistic instructions; and the behaviour-recognition task which consists in linguistically recognising behaviours. During training the agent experiences only a subset of all linguistic instructions/behaviours. Trained agents successfully acquire an ability to perform both tasks. Moreover some of the successfull agents proved to be able to access and execute also linguistic instructions not experienced during training. However, none of the successfull agents manage to linguistically recognise behaviours corresponding to the execution of linguistic instructions not experienced during training. We conclude by speculating on potential factors that may have inhibited the agents from developing fully compositional semantics structures.

Introduction

The main objective of this study is to design neural mechanisms to allow autonomous agents to develop the linguistic skills necessary to perform both a behaviour-production task and a behaviour-recognition task. The behaviour-production task requires the agents to access linguistic instructions and to correctly execute them. The instructions are made of two parts: a part that defines the type of action, and a part that defines the object on which to perform the action. The behaviour-recognition task requires the agents to observe their own behaviours during the successful execution of each linguistic instruction and to generate the corresponding linguistic instruction (i.e., the object label and the action label).

Successful agents will be further post-evaluated to learn more about the semantics structures underpinning their linguistic skills. We will look at how the development of behavioural and linguistic skills required for the comprehension and the generation of the linguistic instructions changes the way in which the agents represent linguistic labels and attach meaning to them. For example, in the behaviour-production task, we are interested in whether, and eventually at which point in the learning phase, the agents perform the task by exploiting a flexible conceptual system in which object labels and action labels are parsed in a way that even never experienced object-action pair can be conceived as a recombination of previously experienced linguistic elements. In the behaviour-recognition task, we are also interested in whether, and eventually when, the capability of recognising the linguistic instructions associated with the perceived behaviours is underpinned by a compositional semantic system. Owing to this system, previously unexperienced behaviours are seen to be made of elementary behavioural units corresponding to already experienced elementary linguistic labels.

The broad objective of this study is to capture and to systematically investigate, through the use of simulated agent-based modelling, phenomena related to language learning observed in humans. Models of embodied (physical or simulated) agents focused on the study of phenomena related to language learning have become more significant with recent psychological and neuroscientific evidence of close links between the mechanisms of action and those of language (Glenberg and Kaschak, 2002; Gallese, 2008). This is because embodied and situated agent-based models represent a suitable methodological platform to test or to generate various hypothesis concerning the relationship between the development of motor and linguistic skills (Hutchins and Johnson, 2009). In recent years, various types of agent-based models have been employed to generate proof-of-concept demonstrations on how language-like symbolic systems can be acquired by artificial agents through interactions with a physical and/or social environment (e.g., Cangelosi and Parisi, 2002; Steels, 2002; Roy, 2002; Cangelosi and Riga, 2006).

Particularly inspiring for our work is a series of articles specifically focused on the acquisition of a compositional semantics (Sugita and Tani, 2005, 2008). That is, a compositional system grounded on the agent's sensory-motor skills (see Harnard, 1990, for the meaning of grounding in language learning). In (Sugita and Tani, 2005, 2008), the authors investigate this issue on tasks that require the shift from rote knowledge to systematised knowledge. This work has

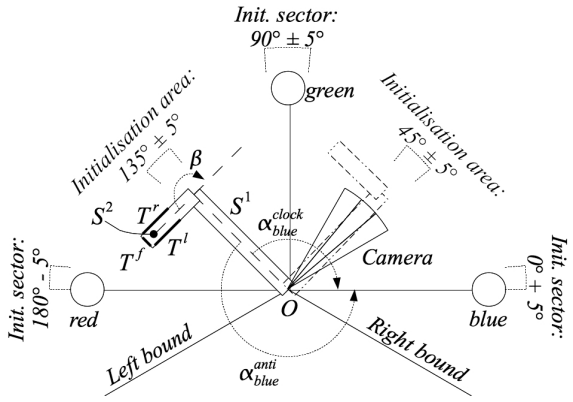


Figure 1: The agent structure and its world. The vision system of the agent is drawn only with respect to the arm initialised on the right initialisation area.

contributed evidence for a dynamical perspective on compositional semantic systems, an alternative perspective to the one in which neural correlates of language are viewed as atomic elements semantically associated to basic units of the linguistics systems (see also Van Gelder, 1990, on this issue).

This study complements previous research on the development of compositional semantics by looking at circumstances in which the development of linguistic skills concerns both the domain of language comprehension and language production. The analysis of the obtained results indicatates that the agents successfully develop a semantic space, grounded on their sensory motor capability and organised in a way that enable linguistic compositionality and generalisation in the case of behaviour generation but not in the case of behaviour recognition. That is, the recognition of behaviour through the production of linguistic instruction seems to be acquired by rote knowledge. We conclude by speculating on potential factors that may have inhibited the agents from developing fully compositional semantics structures.

The task and the agent

Each agent lives in a two-dimensional world and is composed of an arm with two segments referred to as S^1 (100 cm) and S^2 (50 cm), and two degrees of freedom (DOF). Each DOF comprises a rotational joint which acts as the fulcrum and an actuator. One actuator causes S^1 to rotate clockwise or anticlockwise around point O, with the movement restricted within the right ($-30°$) and the left ($210°$) bound. The other actuator causes S^2 to rotate within the range [$-90°$, $90°$] with respect to S^1. Friction and momentum are not considered (see Fig. 1). In the environment there are three rounded objects of different colours (i.e., a blue, a green, and a red object). The objects are placed at

Table 1: The linguistic instructions. In grey the non-regular instructions, that is, those not experienced during training.

	MOVE $Inst_o^M$					
	Object			Action		
	I_{15}	I_{16}	I_{17}	I_{18}	I_{19}	I_{20}
Blue	1	1	0	0	1	1
Green	1	0	1	0	1	1
Red	0	1	1	0	1	1

	TOUCH $Inst_o^T$					
	Object			Action		
	I_{15}	I_{16}	I_{17}	I_{18}	I_{19}	I_{20}
Blue	1	1	0	1	0	1
Green	1	0	1	1	0	1
Red	0	1	1	1	0	1

	INDICATE $Inst_o^I$					
	Object			Action		
	I_{15}	I_{16}	I_{17}	I_{18}	I_{19}	I_{20}
Blue	1	1	0	1	1	0
Green	1	0	1	1	1	0
Red	0	1	1	1	1	0

150 cm from point O with their centre placed anywhere on the chord delimiting their corresponding Init. sector (see Fig. 1). The objects do not move unless pushed by the arm. The agent is equipped with a linear camera with a receptive field of $30°$, divided in three sectors, each of which has three binary sensors (C_i^B for blue, C_i^G for green, and C_i^R for red, with $i \in [1, 2, 3]$ sectors). Each sensor returns 1 if the blue/green/red object falls with the corresponding sector. The camera and S^1 move together. The experimental set up is built in a way that at each time step there can be only one object in the camera view. If no coloured object is detected, the readings of the sensors are set to 0. The agent is also equipped with right and left bound binary sensors (B^r and B^l) which activate (i.e., their reading is set to 1) whenever S^1 reaches the right or the left bound, respectively. Finally, three binary touch sensors (i.e., T^r, T^f, T^l) are placed on the right, front, and left side of S^2. Collisions between the agent and an object are handled by a simple model in which whenever S^2 pushes the object the relative contact points remain fixed.

Agents are trained on both a behaviour-production task and on a behaviour-recognition task. The behaviour-production task consists, for the agents, in the execution of the following instructions (which will be referred to in the remaining part of the paper as *regular instructions*): TOUCH BLUE object ($Inst_{blue}^T$), TOUCH RED object ($Inst_{red}^T$), MOVE GREEN object ($Inst_{green}^M$), MOVE RED object ($Inst_{red}^M$), INDICATE BLUE object ($Inst_{blue}^I$), INDICATE GREEN object ($Inst_{green}^I$), and INDICATE RED

object ($Inst^I_{red}$, see also Table 1). TOUCH and MOVE require the agent to rotate S^1 and S^2 until S^2 collides with the target object. TOUCH requires an agent to remain in contact with the target object with the right side of S^2 (that is, by activating the touch sensor T^r) for an uninterrupted interval of 100 time steps. During this interval, S^1 must not rotate. MOVE requires an agent to rotate S^1 more than $35°$ while S^2 is touching the object with its right side. The rotation of S^1 while S^2 is touching the object determines the movement of the object. INDICATE requires an agent to rotate S^1 until the angular distance between S^1 and the object is less than $30°$. INDICATE is correctly executed only if S^1 remains at less than $30°$ from the target object for more than 100 time steps. During the execution of INDICATE, an agent must not collide with any object. During the execution of TOUCH and MOVE, an agent must not collide with the non target objects (i.e., the objects not mentioned in the current linguistic instruction).

The behaviour-recognition task consists, for the agents, in recognising and correctly labelling own behaviours perceived through sequences of α, β duplet. Each duplet corresponds to the angular rotation of the two segments of the arm. In particular, α corresponds to the normalised clockwise angle from S^1 to the axis from O to the lower end position of the blue object Init. sector. β corresponds to the normalised relative rotation of S^2 with respect to S^1 (see Fig. 1). The duplets are recorded during the successful execution of the behaviours at the behaviour-production task.

We run two different series of simulations (referred to as Exp. A and Exp. B) which differ in the training schema. In Exp. A, the agents are evaluated on the behaviour-recognition task only if they successfully perform all the regular instructions during the behaviour-production task. In Exp. B, each agent performs the behaviour-recognition task as soon as it successfully executes at least one regular instruction at the behaviour-production task. In this case, the behaviour-recognition task is limited only to those regular instructions successfully executed at the behaviour-production task. After training, all the agents are evaluated for their capability to access regular and non-regular linguistic instructions and to execute the corresponding behaviours and also for their capability to label behaviours corresponding to the execution of regular and non-regular instructions.

The agent controller and the evolutionary algorithm

The agent controller is composed of a continuous time recurrent neural network (CTRNN) of 22 sensor neurons, 8 inter-neurons and 10 output neurons (Beer and Gallagher, 1992). During the behaviour-production task, at each time step, sensor neurons from 1 to 20 are activated using an input vector I_i with $i \in [1, .., 20]$ corresponding to the sensors readings indicated in Fig. 2, and the input to sensor neuron 21 and 22 is set to 0. During the behaviour-recognition task,

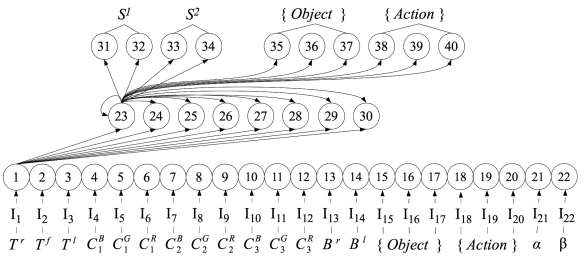


Figure 2: The neural network. Continuous line arrows indicate the efferent connections for the first neuron of each layer. Underneath the input layer, it is shown the correspondences between sensors/linguistic instructions, the notation used in equation 1a to refer to them, and the sensory neurons.

at each time step, the input to sensor neurons 1 to 20 is set to 0, and sensor neurons 21 and 22 are activated using an input vector I_i with $i \in [21, 22]$ corresponding to the α, β generated by successfully executing the linguistic instructions at the behaviour-production task.

The inter-neuron network is fully connected. Additionally, each inter-neuron receives one incoming synapse from each sensory neuron. Each output neuron receives one incoming synapse from each inter-neuron. There are no direct connections between sensory and output neurons. The states of the output neurons are used to control the movement of S^1 and S^2 as explained later. The states of the neurons are updated using the following equations:

$$\frac{\Delta y}{\Delta T} = \begin{cases} \Big(-y_i + gI_i \Big) \dfrac{1}{\Delta T}; & \text{(1a)} \\ \Big(-y_i + \displaystyle\sum_{j=1}^{30} \omega_{ji} \sigma(y_j + \beta_j) \Big) \dfrac{1}{\tau_i}; & \text{(1b)} \\ \Big(-y_i + \displaystyle\sum_{j=23}^{30} \omega_{ji} \sigma(y_j + \beta_j) \Big) \dfrac{1}{\Delta T}; & \text{(1c)} \end{cases}$$

for $i \in \{1, .., 22\}$ in eq. 1a, for $i \in \{23, ..., 30\}$ in eq. 1b, for $i \in \{31, .., 40\}$ in eq. 1c, and with $\sigma(x) = (1 + e^{-x})^{-1}$. In these equations, using terms derived from an analogy with real neurons, y_i represents the cell potential, τ_i the decay constant, g is a gain factor, I_i the intensity of the perturbation on sensory neuron i, ω_{ji} the strength of the synaptic connection from neuron j to neuron i, β_j the bias term, $\sigma(y_j + \beta_j)$ the firing rate (hereafter, f_i). All sensory neurons share the same bias (β^I), and the same holds for all output neurons (β^O). τ_i and β_i with $i \in \{23, ..., 30\}$, β^I, β^O, all the network connection weights ω_{ij}, and g are genetically specified networks' parameters. At each time step the angular movement of S^1 is $2.9H(f_{31} - 0.5)sgn(0.5 - f_{32})$ degrees and of S^2 is $2.9H(f_{33} - 0.5)sgn(0.5 - f_{34})$ degrees, where H is the Heaviside step function and sgn is the sign function.

A generational genetic algorithm is employed to set the parameters of the networks (Goldberg, 1989). The population contains 100 genotypes. Generations following the first one are produced by a combination of selection with elitism, recombination and mutation. For each new generation, the five highest scoring individuals from the previous generation are retained unchanged. The remainder of the new population is generated by fitness-proportional selection from the 70 best individuals of the old population. Each genotype is a vector comprising 340 real values. At the beginning of the evolutionary process, each gene is chosen randomly from a uniform distribution in the range $[0, 1]$. Cell potentials are set to 0 when the network is initialised or reset, and circuits are integrated using the forward Euler method with an integration time step $\Delta T = 0.1$.

The fitness function

During evolution, each genotype is translated into an arm controller and evaluated more than once for all the object-action regular instructions by varying the starting positions. The agent fitness is computed on both the behaviour-production task and the behaviour-recognition task ($F^{total} = F^{production} + F^{recognition}$, see below for details).

The behaviour-production task

During the behaviour-production task, the agents perceive regular instructions and they are required to execute the corresponding behaviours. Agents are evaluated 14 times initialised in the left and 14 times in the right initialisation area, for a total of 28 trials. For each initialisation area, an agent experiences 2 times all the regular linguistic instructions. The linguistic instructions $Inst^M_{blue}$ and $Inst^T_{green}$ are never experienced during the training phase. At the beginning of each trial, the agent is randomly initialised in one of the two initialisation area, and the state of the neural controller is reset. A trial lasts 12 simulated seconds ($T = 250$ time steps). A trial is terminated earlier in case the arm collides with a non target object.

In each trial k, an agent is rewarded by an evaluation function which seeks to assess its ability to execute the desired action on the target object. The final fitness $F^{production}$ attributed to an agent is the sum of two fitness components F^1_k and F^2_k. F^1_k rewards the agent for reducing the angular distance between S^1 and the target object. F^2_k rewards the agent for performing the required action on the target object.

$$F^{production} = \frac{1}{28}\sum_{k=1}^{28}(F^1_k + F^2_k); \qquad (2)$$

F^1_k and F^2_k are computed as follows:

$$F^1_k = \max\left(0, \frac{d^i - d^f}{d^i} \cdot P^1_k, \mathbb{1}_{d^f<4.6^\circ}\right); \qquad (3)$$

where d^i and d^f are respectively the initial (i.e., at $t = 0$) and final (i.e., at the end of the trail k) angular distances between S^1 and the target object and $\mathbb{1}_{d^f<4.6^\circ}$ is 1 if $d^f < 4.6^\circ$, 0 otherwise. P^1_k is the penalty factor, which is set to 0.6 if the agent collides with a non target object, to 1.0 otherwise. The angle between S^1 and the target object o can be measured *clockwise* (α^{clock}_o) or *anticlockwise* (α^{anti}_o). In equation 3, d^i and d^f are the minimum between the clockwise and anticlockwise distance, that is $d = \min\left(\alpha^{clock}_o, \alpha^{anti}_o\right)$.

$$F^2_k = \begin{cases} \dfrac{\textit{steps-on-target}}{\textit{max-steps-on-target}} \cdot N \cdot P^2_k; & \text{for TOUCH or INDICATE} \quad (4a) \\ \dfrac{\Delta\theta}{\textit{max-angular-offset}} \cdot N \cdot P^2_k; & \text{MOVE} \quad (4b) \end{cases}$$

where *max-steps-on-target* $= 100$, $P^2_k = 0$ if $F^1_k < 1$ otherwise $P^2_k = 1$, *max-angular-offset* $= 34.4^\circ$, $N = 2$ for TOUCH and MOVE, and $N = 1$ for INDICATE. For the action INDICATE, *steps-on-target* refers to the number of time steps during which $F^1_k = 1$, and S^2 does not touch the target object. For the action TOUCH, *steps-on-target* refers to the number of time steps during which $F^1_k = 1$, S^2 touches the target object by activating the touch sensor T^r, and S^1 does not change its angular position. $\Delta\theta$ is the angular displacement of the orientation of S^1 recorded while $F^1_k = 1$, and S^2 is touching the target object by activating the touch sensor T^r. A trial is terminated earlier if *steps-on-target* $=$ *max-steps-on-target* during the execution of INDICATE or TOUCH and when $\Delta\theta =$ *max-angular-offset* during the execution of MOVE.

The behaviour-recognition task

During the behaviour-recognition task, the agent is evaluated for labelling its behaviours corresponding to the successful execution of each of the regular instructions. That is, the arm of the agent is moved so as to display a behaviour previously exhibited during the behaviour-production task by the agent itself, and it is asked to produce the corresponding linguistic instruction (without receiving it as input).

In Exp. A, an agent moves on to the behaviour-recognition task only if it successfully completes all the trials of the behaviour-production task (i.e., $F^{production} > 2.57$) . In Exp. B, an agent moves on to the behaviour-recognition task as soon as it successfully completes at least one trial at the behaviour-production task (i.e., $\exists k|(F^1_k + F^2_k) > 2.57$). The behaviour-recognition task comprises only the trial/s successfully executed at the behaviour-production task. In other words, in Exp. A, the evolution of the mechanisms to accomplish the behaviour-recognition task follows the evolution of the mechanisms to successfully execute the behaviour-production task. In Exp. B, the evolution of the mechanisms for the behaviour-production task and the behaviour-recognition task evolve simultaneously,

since it suffices for an agent to successfully complete a single trial of the behaviour-production task to move on to the behaviour-recognition task.

In each trial k, the functions F_k^{obj} and F_k^{act} reward the agents for matching with the firing rate of the output neurons 35, 36, 37, 38, 39, and 40 the six digit regular instruction that triggered the currently experienced successful behaviour. F_k^{obj} and F_k^{act} are computed as follow:

$$F^{recognition} = \frac{1}{28}\sum_{k=1}^{K}(F_k^{obj} + F_k^{act});$$

$$F_k^3 = \frac{\sum_{t=T-5}^{T}\left(2^{-2\cdot rank_{k,t}} + \frac{2(1-f_{k,t}^r) + \sum_{i\in W_{k,t}} f_i}{4}\right)}{2\cdot 5}; \quad (5)$$

with $F_k^{obj} = F_k^3$ with $W_{k,t}$ the subset of output neurons defining the object label (i.e., neurons 35, 36, and 37) whose activation should be 1, $f_{k,t}^r$ the firing rate of the neuron defining the object label whose activation should be 0, $rank_{k,t}$ the rank of $f_{k,t}^r$ when the output neurons defining the object label are ranked in ascending firing rate order. F_k^{act} is computed as F_k^{obj} considering the output neurons defining the action label (i.e., neurons 38, 39, 40). $F_k^3 = 0$ if $(F_k^1 + F_k^2) < 2.57$ (i.e. if the behaviour at trial k has not been correctly executed).

Results

For each experimental condition (Exp. A, Exp. B), we run ten evolutionary simulations for 10000 generations, each using a different random initialisation. Recall that our objective is to generate agents that are capable of successfully performing both the behaviour-production task and the behaviour-recognition task. Moreover, we are interested in investigating whether successful agents develop semantic structures that are functionally compositional. Agents endowed with a functionally compositional semantics should be able to access and execute linguistic instructions never experienced during training (i.e., from non-regular instructions to the execution of the corresponding behaviours). They may also be able to linguistically describe a behaviour never performed/experienced during training (i.e., from the perception of behaviours never executed during training to the generation of non-regular instructions). We run two different series of simulations (i.e., Exp. A and Exp. B) to see whether a different training bears upon the development of functionally compositional neural structures.

The best agents of each generation in both experimental conditions have been post-evaluated by first running sets of 80 trials for each regular and non-regular linguistic instruction in which the agents are asked to perform the behaviour-production task. Hereafter, we refer to this first phase of the post-evaluation test as behaviour-production test. In half of the trials the agents are randomly initialised in the right initialisation area and half of the trials in the left one (see Fig 1). We considered those agents successful at the behaviour-production test (hereafter, referred to as *b-successful*) that manage to obtain a success rate higher than 80% in performing the behaviours corresponding to the execution of the regular linguistic instructions (i.e., those experienced during evolution). *b-successful* agents have been further classified into i) *b-non-compositional* agents, referring to those *b-successful* agents that proved to be less than 80% successful at performing the behaviour corresponding to the execution of both the non-regular instructions, $Inst_{blue}^M$ and $Inst_{green}^T$; ii) *b-partially-compositional* agents referring to those *b-successful* agents that proved to be more than 80% successful at performing the behaviour corresponding to the execution of only one of the two non-regular instructions, $Inst_{blue}^M$ or $Inst_{green}^T$; iii) *b-fully-compositional* agents re-

Table 2: Result of post-evaluation tests performed on the best agents of each generation for four runs of Exp. A, and for two runs of Exp. B. The tables show the number of successful agents at the behaviour-production task on regular linguistic instructions, and the percentage of them also successful on the non-regular instructions. The tables also show the number of successful agents at the behaviour-recognition task on regular linguistic instructions, and the percentage of them also successful on the non-regular instructions.

Exp. A				
run	1	8	9	10
Num. *b-successful*	5310	414	2079	6588
$Inst_{blue}^M$	0.00	0.00	6.54	37.66
$Inst_{green}^T$	0.00	0.00	33.57	0.00
$Inst_{blue}^M$ and $Inst_{green}^T$	0.00	0.00	0.00	0.00
Num. *l-successful*	0	0	0	0
$Inst_{blue}^M$	0.00	0.00	0.00	0.00
$Inst_{green}^T$	0.00	0.00	0.00	0.00
$Inst_{blue}^M$ and $Inst_{green}^T$	0.00	0.00	0.00	0.00

Exp. B		
run	5	7
Num. *b-successful*	3183	9613
$Inst_{blue}^M$	0.00	17.07
$Inst_{green}^T$	8.83	19.08
$Inst_{blue}^M$ and $Inst_{green}^T$	0.00	21.49
Num. *l-successful*	0	1753
$Inst_{blue}^M$	0.00	0.00
$Inst_{green}^T$	0.00	0.00
$Inst_{blue}^M$ and $Inst_{green}^T$	0.00	0.00

ferring to those *b-successful* agents that proved to be more than 80% successful at performing the behaviour corresponding to the execution of both the non-regular instructions, $Inst^M_{blue}$ and $Inst^T_{green}$.

During the second phase of the post-evaluation test, *b-successful* agents are asked to perform the behaviour-recognition task. That is, they are required to produce as output the regular and non-regular linguistic instructions that, during the behaviour-production test, triggered their successful behaviour. Hereafter, we refer to this second phase of the post-evaluation test as behaviour-recognition test. Recall that, behaviour-recognition test on non-regular instructions is performed only on *b-partially-* or *b-fully-compositional* agents. Moreover, recall that the agents perceive their successful behaviours through sequences of duplet α, β, recorded during successful post-evaluation trials of the behaviour-production test. As for the behaviour-production test, we considered those agents successful at the behaviour-recognition test (hereafter, referred to as *l-successful*) that manage to obtain a success rate higher that 80% in generating the regular linguistic instructions. Note that, the object label generated by the agent controller is considered "blue" if the neuron with the lowest firing rate is neuron 35, "green" if it is neuron 36, "red" if it is neuron 37. The action label generated by the agent controller is considered "touch" if the neuron with the lowest firing rate is neuron 38, "move" if it is neuron 39, "indicate" if it is neuron 40. *L-successful* agents have been further classified in i) *l-non-compositional* agents, referring to those *l-successful* agents that proved to be less than 80% successful at generating non-regular linguistic instructions, $Inst^M_{blue}$ and $Inst^T_{green}$; ii) *l-partially-compositional* agents referring to those *l-successful* agents that proved to be more than 80% successful at generating only one of the two non-regular instructions, $Inst^M_{blue}$ or $Inst^T_{green}$; iii) *l-fully-compositional* agents referring to those *l-successful* agents that proved to be more than 80% successful at generating both the non-regular instructions, $Inst^M_{blue}$ and $Inst^T_{green}$.

Table 2 shows the results of post-evaluation tests on those evolutionary runs in which we recorded the presence of *b-successful* agents. First, only four out of ten runs in Exp. A, and two out of ten runs in Exp. B produced *b-successful* agents. Second, only run 7 in Exp. B produced agents that are both *b-successful* and *l-successful*. This result indicates that, given our methodological setup, it is extremely difficult to design the mechanisms to allow autonomous agents to perform both the behaviour-production task and the behaviour-recognition task as described in previous Sections. The experimental condition in which the mechanisms to perform the behaviour-production task and the behaviour-recognition task co-adapt simultaneously (i.e., Exp. B) seems to contain the necessary "ingredients" to accomplish the objective of this study. However, the fact that only one out of ten runs produced both *b-successful* and *l-successful* agents suggests that there are elements that severely hindered the evolution from generating the neural structured required by the agents to accomplish their objective. What are these elements? At this stage of our investigation, we have evidence to claim that the number of hidden neurons of the neuro-controllers has a bearing on the evolution of *b-successful* agents. In a previous study described in (Tuci et al., 2010), we have evolved agents to perform only the behaviour-production task in evolutionary circumstances identical to those illustrated in this study. In (Tuci et al., 2010), agents were controlled by neural controllers with only three hidden neurons. Almost all the evolutionary runs generated *b-successful* agents. It seems that smaller neural controllers corresponding to a smaller evolutionary search space facilitates the evolution of the mechanisms to accomplish the behaviour-production task. However, when employed in this study, three-hidden-neuron controllers proved to be insufficient to perform both the behaviour-production task and the behaviour-recognition task. We had to progressively increase the number of hidden neurons from three to eight to generate *b-successful* and *l-successful* agents. Further tests are certainly required to isolate other elements of our model that may have a strong bearing on the capability to generate *b-successful* and *l-successful* agents.

Table 2 also shows the results concerning compositionality. Only run n. 7 in Exp B produced agents that turned out to be *b-fully-compositional*. *b-partially-compositional* agents can be found in run 9 and 10 of Exp. A, and in both runs of Exp. B. None of the runs produced *l-partially-compositional* or *l-fully-compositional* agents. It is worth noting that the mechanisms to access non-regular instructions and to generate the corresponding behaviours do not underpin the inverse process, that is, from the perception of behaviours never executed during training to the generation of the corresponding non-regular instructions. This suggests that linguistic skills related to the capability to comprehend and to generate linguistic instructions in *b-fully-compositional* and *l-successful* agents are underpinned by different neural mechanisms. The mechanisms concerning the capability to be *b-fully-compositional* work as a functionally compositional semantic structure. The mechanisms concerning the capability to be *l-successful* allow the agents to learn by rote the association between the perception of sequences of α, β duplet and regular instructions.

Figure 3 show several graphs which tell us more about the evolutionary dynamics which led to the emergence of *b-successful* and *l-successful* agents in run 7 of Exp. B. These graphs show for each best agent of each generation of run n. 7 the percentage of success for each instruction of the behaviour-recognition test (see dotted, dashed, and continuous lines in Figure 3) as well as the generations in which the agents turned out to be *b-successful*, and the generation at which the agents turned out to be *b-fully-compositional*

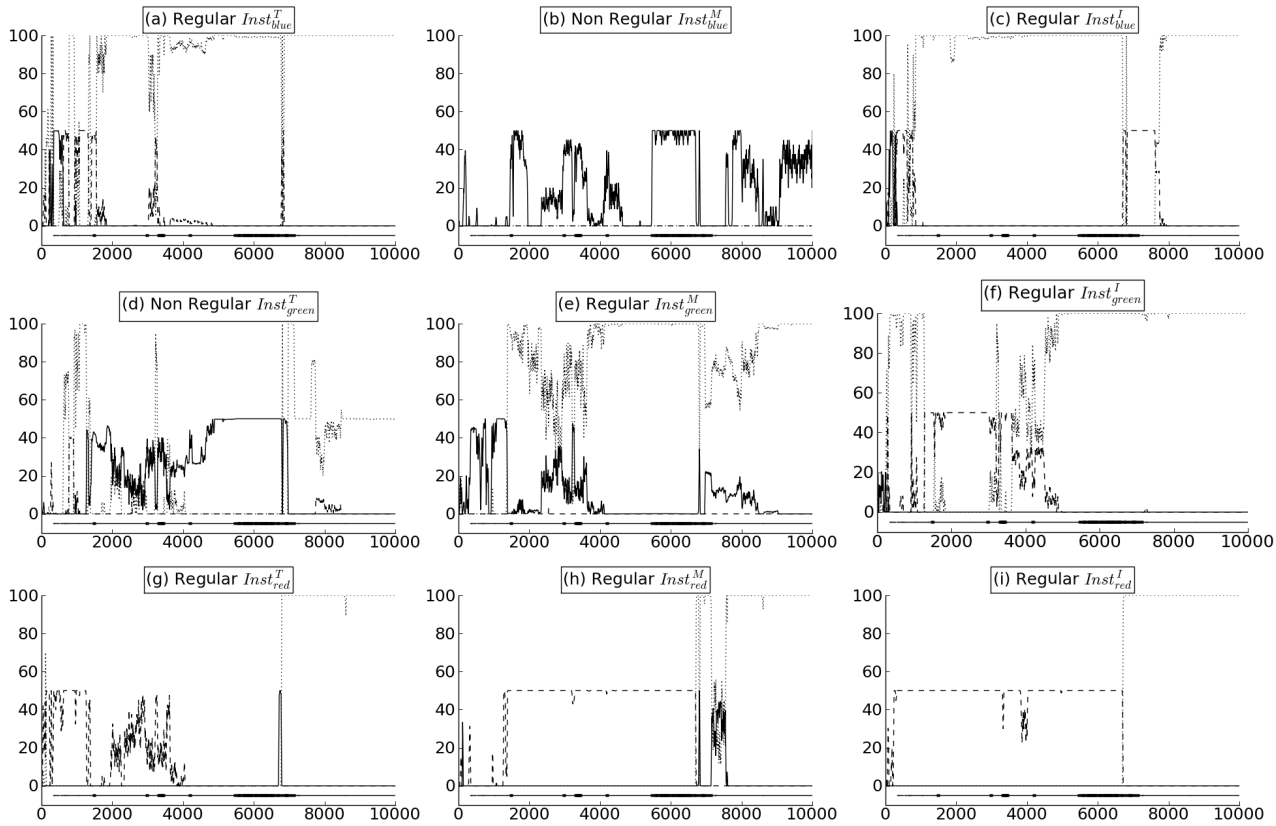


Figure 3: Graphs showing for each best agent of each generation of run n. 7 the percentage of success for each instruction of the behaviour-recognition test. Dotted lines refer to the percentage of success in generating the labels for both the object and the action. Continuous lines refer to the percentage of cases in generating the correct label for the object and the wrong one for the action. Dashed lines refer to the percentage of cases in generating the correct label for the action and the wrong one for the object. At the bottom of each graph, the thin horizontal continuous line indicates the generations in which the agents turned out to be *b-successful*. The tick horizontal line over-imposed on the thin one, indicates the generations in which the agents turned out to be *b-fully-compositional* (see text for details). Data are smoothed with a moving average of window size 20.

(see thin and thick horizontal lines below zero in Figure 3). First, we notice that *b-fully-compositional* agents keep on appearing and disappearing during evolution, while successful agents once generated, are almost never lost. These data suggest that compositionality is not automatically associated with, and is not a prerequisite for developing the capability of successfully performing the behaviour-production task. Second, *l-successful* agents appear very late in evolution. In particular, the agents seemed to have hard time to correctly label behaviours triggered by instructions concerning the red object (see continuous and dashed lines in Figure 3g, 3h, 3i). *l-successful* agents appear after generation 6000, definitely later than the appearance of *b-fully-compositional* agents (see dotted lines and the tick horizontal lines below zero in Figure 3a, 3c, 3e, 3f, 3g, 3h, 3i). This suggest that the emergence of a functionally compositional semantics is not determined by the evolution of the mechanisms to successfully perform the behaviour-recognition task. Third, the graphs concerning non-regular instructions tell us that the agents are not completely unable to deal with these circumstances. For example, as far it concerns $Inst^{M}_{blues}$ (see Figure 3b), several agents during evolution proved to be up to 50% successful in correctly labelling the object on which the action was performed. As far it concerns $Inst^{T}_{green}$ (see Figure 3d), up to generation 6000, the agents seemed to be more effective in labelling the object, while after generation 6000 they proved to be at least 50% effective in correctly labelling both the object and the action given the behaviour corresponding to the execution of this instruction.

Conclusions

We have described a set of simulations which generated autonomous agents, controlled by a single non *a priori* modularised neuro-controller, capable of successfully executing both a language comprehension and a language production task. Post-evaluation tests revealed that, successful agents

display a form of compositional semantics which allow them to access linguistic instructions not experienced during training and to execute the corresponding behaviours also no experienced during training. That is, we observed generalisation capabilities in the behaviour-production task. The same successful agents proved not capable of correctly labelling their own behaviours not experienced during training. That is, we did not observe generalisation capabilities in the behaviour-recognition task. Although at this stage we do not have enough empirical evidence to account for this result, we can definitely formulate a number of not mutually exclusive hypothesis that we will consider to identify future directions of work.

Why successful agents show generalisation capabilities at the behaviour-production task and no generalisation capabilities at the behaviour-recognition task? First, we can hypothesise that, the agents have enough computational resources (e.g., hidden neurons) to learn by rote the association between behaviours represented by sequences of α, β duplet and linguistic labels. Alternatively, it could be that the behaviour-recognition task did not produce sufficiently selective evolutionary pressures to generate the mechanisms required to shift from rote knowledge to a more flexible conceptual system. Second, from the agent point of view, the behaviour-production task and the behaviour-recognition task are mostly uncorrelated tasks. This becomes clear if we consider that the agent has two groups of input-output neurons: one (comprising input neurons 1 to 20 and output neurons 31 to 34) that is only used during the behaviour-production task; the other (comprising input neurons 21 and 22 and output neurons 35 to 40) that is only used during the behaviour-recognition task. Due to the different nature of the two input-output groups, the input received during the behaviour-recognition task is completely different from the motor output and from any other input experienced during the behaviour-production task. This may make it difficult for the agent to develop a coherent internal structure, common to the language comprehension and language production task. To try to cope with this problem we plan to explore two possibilities: one is to modify the agent body and neural architecture, the other is to slightly modify the task. As far as the agent is concerned, one possibility could be to change the way the output controlling the arm movement is encoded, so to have at least similar kinds of input and output signal. Another possibility could be to feed the α and β input neurons also during the behaviour-production task (as if the agent could "see" himself doing the task). On the task side, we plan to implement setups in which the two abilities have to be used together. For example, we could ask the agent to produce the correct linguistic instruction during the behaviour-production task. Even though this is a rather easy task (the correct instruction is already present in the input units), it could nonetheless favour the emergence of common structures underpinning both the language comprehension and language production task.

Acknowledgements

This research work was supported by the *ITALK* project (EU, ICT, Cognitive Systems and Robotics Integrating Project, grant n° 214668). The authors thank Andrew Szabados and Domenico Parisi for helpful comments during the preparation of the paper.

References

Beer, R. D. and Gallagher, J. C. (1992). Evolving dynamic neural networks for adaptive behavior. *Adaptive Behavior*, 1(1):91–122.

Cangelosi, A. and Parisi, D., editors (2002). *Simulating the evolution of language*. Springer Verlag, New York.

Cangelosi, A. and Riga, T. (2006). An embodied model for sensorimotor grounding and grounding transfer: Experiments with epigenetic robots. *Cognitive Science*, 30(4):673–689.

Gallese, V. (2008). Mirror neurons and the social nature of language: The neural exploitation hypothesis. *Social Neuroscience*, 3:317–333.

Glenberg, A. and Kaschak, M. (2002). Grounding language in action. *Psychonomic Bulletin & Review*, 9:558–565.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA.

Harnard, S. (1990). The symbol grounding problem. *Physica D*, 42:335–346.

Hutchins, E. and Johnson, C. (2009). Modelling the emergence of language as an embodied collective cognitive activity. *Topics in Cognitive Science*, 1:523–546.

Roy, D. (2002). Learning visually grounded wprds and syntax for a scene description task. *Computer Speech and Language*, pages 353–385.

Steels, L. (2002). Grounding symbols in through evolutionary language games. In Cangelosi, A. and Parisi, D., editors, *Simulating the evolution of language*, pages 211–226. Springer Verlag, London.

Sugita, Y. and Tani, J. (2005). Learning semantic combinatoriality from the intercation between linguistic and behvioral processes. *Adaptive Behavior*, 13(1):33–52.

Sugita, Y. and Tani, J. (2008). Acquiring a functionally compositional system of goal-directed actions of a simulated agent. In Asada, M., Hallam, J., Meyer, J.-A., and Tani, J., editors, *Proc. of the* 10^{th} *Int. Conf. on Simulation of Adaptive Behavior (SAB2008)*, pages 331–341. Springer Verlag.

Tuci, E., Ferrauto, T., Massera, G., and Nolfi, S. (2010). Co-development of linguistic and behavioural skills: compositional semantics and behaviour generalisation. In *Proceedings of the 11th International Conference on the Simulation of Adaptive Behaviour (SAB 2010)*, volume LNCS. Springer, Berlin, Germany. In Press.

Van Gelder, T. (1990). Compositionality: A connectionist variation on a classical theme. *Cognitive Science*, 14:355–384.

Two Agents Acting as One

Malte Harder[1], Daniel Polani[1] and Chrystopher L. Nehaniv[1]

[1]Adaptive Systems Research Group, University of Hertfordshire, Hatfield, UK
{m.harder,d.polani,c.l.nehaniv}@herts.ac.uk

Abstract

We consider two agents, each equipped with a controller. When they achieve a joint goal configuration, their coordination can be measured informationally. We show that the amount of *coordination* that two agents need to configure in a certain way depends on the amount of information they obtain from their environment. Furthermore the environment imposes a coordination pressure on the agents that depends on the size of the environment. In a second scenario we introduce a shared centralized controller which leads to a synchronisation of the agents' actions for suboptimal policies. However, in the optimal case this *intrinsic coordination* vanishes and the shared centralized controller can be split into two individual controllers.

Introduction

When one considers biology, many phenomena require that subentities perform actions in a coordinated way. This phenomenon is so prevalent that it requires pivotal treatment. It is seen in swarms, morphogenesis as well as in the actions of different parts of a single organism. We wish to study some principles behind this central phenomenon in an Artificial Life setting. In the sense of a 'life that could have been' (Langton, 1997) we are interested in what minimal assumptions have to be made to investigate coordination and autonomy within a collective of two agents. For this purpose we use the framework of information theory. We do not assume a particular metabolism and intrinsic dynamics but have the choice of certain limitations on information processing. This makes it possible to develop necessary and sufficient conditions for life-like scenarios and to find invariants for Artificial Life in any type of environment.

Nonetheless a physically consistent model can be plugged into the framework. Furthermore, studying coordination in a scenario that approximates nature has many applications: In ethology the understanding of collective tasks like foraging, flocking or group decision-making is active research (Deneubourg and Goss, 1989; Couzin et al., 2005; Nabet et al., 2009). Social interactions and coordination in robotics have been first studied by Walter (1950) and these issues in natural and artificial agents have received more attention lately (Dautenhahn, 1995, 1999; Ikegami and Iizuka, 2007; Di Paolo et al., 2008), for a review see (Goldstone and Janssen, 2005). Furthermore agent based and cellular models of morphogenesis have been studied with respect to coordination: Deneubourg et al. (1991) investigated the dynamics of ant-like agents that were not able to communicate directly but could pick up and drop objects of different types, leading to coordinated behaviour, called stigmergy, among the agents and clustering of objects of the same type. In an effort to understand morphogenesis of a certain slime mold, coordination between cells was modelled on a subcellular level, resulting in a simulation of the self-organised migration of the mold via an emergent level of photo- and thermotaxis (Marée and Hogeweg, 2001).

Stigmergy and local observation are common ways to model agent communication to get coordinated behaviour (Beckers et al., 1994; Castelfranchi, 2006). In both cases the communication is 'routed' through the environment, in the case of stigmergy in a very explicit way by altering the environment. In these models communication is spatially bound and limited by the amount of information that can be 'stored' in the environment.

When we talk about information, we specifically mean Shannon information (Shannon, 1948). The theory that comes with it allows to compare and quantify relations between random variables which can be used to model causal relationships in Bayesian graphs. Information theory gives a universal language to quantify conditions and invariants for a large class of models in very general way. Furthermore, this allows to compare quantities of models that are otherwise not directly comparable.

To study agent coordination from an information-theoretic perspective towards a predictive and quantitative theory of agent interactions we will look at embodied agents in a grid-world that is underlain by certain 'physical laws', like movement and blocking by other agents. To isolate the influences that a constraint of the agent's information processing capabilities has on the agents' coordination, we will neither impose an environmental constraint on the communication between them, nor a constraint on their sensors. The agents will have a shared controller, but we will limit their

information processing capabilities. Using the information-theoretic quantification of coordination, we will investigate how much they need to coordinate to achieve a given goal in the grid-world and compare this to the coordination in the case where the agents have independent controllers. Obviously the size of the environment has an impact on the amount of coordination as in large grid worlds with few agents there is a smaller chance of collision and less necessity to deal with this situation in an optimal way. For a shared controller we will investigate when the actions are coordinated in a way such that it is not possible to split the controller into two independent controllers which we interpret as both agents 'acting as one'.

Information theory has been successfully employed to models of embodied agents in a growing body of scientific literature starting from Ashby (1956). The idea that information is a main resource for organisms, but at the same time costly to process, is reflected in the evolution of sensors (Nehaniv et al., 2007) and affects the way information theoretic models of agents are investigated (Polani et al., 2007). Lately this idea received increased attention due to new techniques (Touchette and Lloyd, 2000; Klyubin et al., 2004a, 2007; Ay et al., 2008) and there are now broad applications of information theory to Artificial Life related fields (Linsker, 1988; Shalizi and Crutchfield, 2002). Recent results showed that information theoretic learning principles can lead to higher coordination between linked agents (Zahedi et al., 2009) though a different notion of coordination than in this paper is used. In the context of the Information Bottleneck (Tishby et al., 1999) the concept of *relevant information* was introduced by (Polani et al., 2001) and later extended to the perception-action loop (Polani et al., 2006). Here it will be set in relation to an information theoretic quantification of coordination as the mutual information between actions. Sperati et al. (2008) already used the mutual information between actions as a measure of coordination to evolve maximally coordinated agents.

When agents socially interact, or coordinate in an environment they sometimes seem to act as a single entity (e.g. bee hives, ant colonies, multicellular organism), at the same time they are individuals acting at a 'lower' level. In our experiment we will study under which constraints the agents can still be considered as autonomous with respect to the other agents and whether acting as a single entity helps to perform better to achieve a given configuration. Therefore we will introduce a measure of *intrinsic coordination* between two agents which vanishes if both agents have an independent controller and attains its maximum if the action of one agent is fully determined by the action of the other. We will then analyse how much intrinsic coordination is actually needed when acting optimally under an information processing constraint.

Information Theory

Information Theory was introduced by Shannon (1948). We will give a brief introduction: In information theory, *entropy* is given by $H(X) = -\sum_x p(x) \log p(x)$ where X denotes a finite-valued random variable with values in $\mathcal{X}$ and $p(x)$ the probability that X takes on the value $x \in \mathcal{X}$. Entropy measures the uncertainty of the outcome of a random variable. Given a second random variable Y the *conditional entropy* is

$$H(Y|X) = -\sum_{x,y} p(x)p(y|x) \log p(y|x)$$

and measures the uncertainty of Y knowing the outcome of X. To relate these, *mutual information* is defined by $I(X;Y) = H(Y) - H(Y|X)$. Hence, mutual information is a measure of how much the uncertainty of Y is reduced if we know the value of X. Again, this can be conditioned on a third random variable Z which gives the *conditional mutual information* $I(X;Y|Z) = H(Y|Z) - H(Y|X,Z)$. For a detailed account on information theory, see Cover and Thomas (2006).

Coordination

We propose measures of coordination that are independent of the topology of the environment and only depend on distributions of states and actions. Let S denote the random variable of the world states and A the random variable representing its actions where the actions only depend on the current state of the environment.

An important quantity in this context is Relevant Information: it is the minimal amount of information an agent needs to process to perform optimal actions (Polani et al., 2006), denoted by

$$I(S;A^*) = \min_{p(a|s):p(a|s)p(s)>0 \Rightarrow a \text{ optimal for } s} I(S;A).$$

This minimises the mutual information between states and actions but still requires that in each state with positive probability the optimal action is taken. Relevant information reflects, as mentioned in the introduction, the information parsimony principle that processing information has a metabolic cost (Polani et al., 2007) and complies with findings that certain neurons work at information limits, minimising the bandwidth to just maintain their function (Laughlin, 2001).

In theory the relevant information can be much lower than the bandwidth of the sensor, that is, different sensory inputs lead to the same distribution of actions. Moreover, one can ask the converse question: how well can a policy perform if $I(S;A)$ is limited? To do this a utility in terms of a reward structure will be used and the trade-off will be calculated with an algorithm introduced by (Polani et al., 2006).

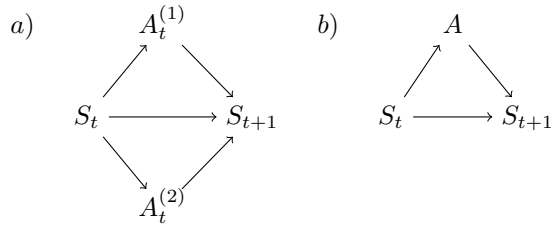


Figure 1: Bayesian network of the perception-action loop for a) independent actions b) joint actions. Here $A^{(1)}$ and $A^{(2)}$ denote the random variable of the action of each agent, A denotes the random variable of the joint action $(a^{(1)}, a^{(2)})$ and t is the time index. In both cases the actions are fully determined by the current state of the environment.

Suppose now there are two agents; the *coordination* is then defined as the mutual information between their actions $I(A^{(1)}; A^{(2)})$ where $A^{(1)}$ is the random variable representing the actions of the first agent and $A^{(2)}$ the random variable representing the actions of the second agent. In the case of independently embodied agents, that is, if $p(a^{(1)}, a^{(2)}|s) = p(a^{(1)}|s)p(a^{(2)}|s)$ the coordination is limited by the relevant information of each agent

$$I(A^{(1)}; A^{(2)}) \leq \min\{I(S; A^{(1)}), I(S; A^{(2)})\}.$$

This follows easily from the data processing inequality (Cover and Thomas, 2006, p. 34). If the agents however have a joint policy $p(a^{(1)}, a^{(2)}|s)$ the coordination is only limited by the entropy of the actions. See Figure 1 for the perception-action loop of the whole system in the case of a) independent controllers and b) one shared controller.

For such an agent pair that has one shared controller it is interesting to see whether there is any intrinsic coordination or whether the controller could be split into two independent controllers. We define *intrinsic coordination* as the conditional mutual information $I(A^{(1)}; A^{(2)}|S)$ which vanishes if $p(a^{(1)}, a^{(2)}|s) = p(a^{(1)}|s)p(a^{(2)}|s)$, that is, the agents come to independent decisions given the state of the environment. By definition intrinsic coordination can be higher or lower than the coordination. In the case that the actions are independent of the state, that is, $H(A^{(1)}|S) = H(A^{(1)})$ and $H(A^{(2)}|S) = H(A^{(2)})$, coordination equals intrinsic coordination, however, the converse is not always the case.

Experimental Setup

We want to study how much (intrinsic) coordination the agents have when they follow an optimal policy to achieve a particular goal configuration (under information processing constraints). Furthermore the amount of coordination will be compared to the coordination in the case where the agents have independent controllers.

The setup consists of two agents, determined by a joint state $s = (s^{(1)}, s^{(2)}) \in \mathcal{S}$ in the state space $\mathcal{S} = \mathcal{W} \times \mathcal{W} - \Delta$ where $\mathcal{W}$ is a $n \times m$ grid-world and $\Delta = \{(w, w)|w \in \mathcal{W}\}$ the diagonal. Hence only one agent is allowed to occupy a particular grid cell per time step. As before, the random variable representing the state of the environment is denoted by S. The goal is given by two particular adjacent cells in the centre of the grid-world and it is not relevant which agent occupies which goal cell, hence there are two goal states in the state space $\mathcal{S}$.

Each agent has five possible actions $\{N, S, W, E, H\}$, go to one of the four neighbouring cells or halt. The actions are denoted by the random variables $A^{(1)}, A^{(2)}$, and their joint action $a = (a^{(1)}, a^{(2)})$ by the random variable A. The distribution of the actions only depends on the location of the two agents. In this scenario the transitions to the next step are deterministic $p(s_{t+1}|a_t, s_t) \in \{0, 1\}$ and reflect the movement of the two agent in the grid-world, blocked by the walls and blocking each other symmetrically (see Figure 2). The agents are blocked if they try to move to the same field or if one agent moves to a field where the other agent stays.

For every step the agents get a reward that is determined by a reward function $r(s_{t+1}, a_t, s_t)$ which depends on the current state, the action taken and the state of the world after the action was executed. A negative reward is given unless both agents occupy a goal cell in which case no reward or penalty is given. Thus, a policy that maximises the expected reward over the lifetime of the agent is one that takes the shortest way to the goal configuration. This defines a Markov Decision Process (MDP), for which reinforcement learning can be used to find such a policy. Given the MDP we can define a *state value function* $V^\pi(s)$ that gives the expected future reward at some state s following the policy π and a *utility function* $U^\pi(s, a)$ that gives the expected reward incorporating the action chosen at state s and then following

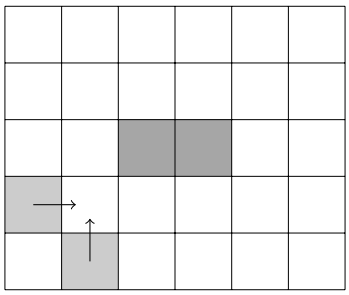


Figure 2: In this 6×5 grid-world, the two dark-grey rectangles show the goal configuration, the light-grey rectangles show a configuration where the agents block each other if they move in the directions of the arrows. This causes that the agents stay at their current position.

the policy π:

$$V^\pi(s) = \sum_a \pi(a|s) \sum_{s'} p(s'|a,s) \left(r(s',a,s) + V^\pi(s')\right),$$
$$U^\pi(s,a) = \sum_{s'} p(s'|a,s) \left(r(s',a,s) + V^\pi(s)\right).$$

The definition of the state value function is recursive and the correct value function is a fixed point of this equation. Iterating the recursive definition of the value function converges to the correct value function for a given policy. If the policy is updated to be greedy with respect to the current utility in every step, the iteration, called optimistic policy iteration, results in an optimal policy for the MDP (Sutton et al., 1999).

If the agents' actions are independent, that is, if $p(a^{(1)}, a^{(2)}|s) = p(a^{(1)}|s)p(a^{(2)}|s)$, the problem breaks down to two dependent MDPs that are not deterministic anymore but whose transition probabilities depend on a prediction of the other agent's action $\tilde{p}(a^{(i)}|s)$. For instance when agent i expects j to act according to $\tilde{p}(a^{(i)}|s)$, then the predictor for the transition of i is:

$$\tilde{p}(s_{t+1}|a_t^{(i)}, s_t) = \sum_{a^{(j)}} \tilde{p}(a^{(j)}|s_t) p(s_{t+1}|a^{(i)}, a_t^{(j)}, s_t),$$

where $i, j \in \{1, 2\}$ and $i \neq j$. In this paper we will update the predictor in every iteration to be the same as the policy of the other agent: $\tilde{p}(a^{(i)}|s) = \pi(a^{(i)}|s)$. That means the agents can do the best possible prediction of the action of the other agent in every step.

Given a scenario where agents do not know anything about each other, it is possible to set the predictor to a uniform distribution. But we want to study how the performance of a split controller compares to the shared controller and will use the policy of the other agent to make the best prediction about the action of the other agent as possible.

The performance of a policy π is measured by the expected utility over all state action pairs, denoted $\mathbb{E}[U^\pi(S, A)]$. To compare both cases a different reward is used in each case: For the shared controller a reward of -2 is given whenever the agents do not enter a goal state. For the independent controllers a reward of -1 is given to each of the agents if it does not enter a goal state, so in each case the summed reward per step is -2 if the goal is not reached. Using the current policy as the predictor $\tilde{p}$ gives another advantage: For the joint policy $\pi(a^{(1)}, a^{(2)}|s) = \pi(a^{(1)}|s)\pi(a^{(2)}|s)$, now the following holds

$$\mathbb{E}[U^\pi(S, A)] = \mathbb{E}[U^{\pi^1}(S, A^{(1)})] + \mathbb{E}[U^{\pi^2}(S, A^{(2)})],$$

where U^π is the utility consistent with the joint policy and U^{π^1}, U^{π^2} are the utilities consistent with the policies $\pi(a^{(1)}|s), \pi(a^{(2)}|s)$. Thus we have a common scale for the expected utilities.

Algorithm

As introduced before, the relevant information is the mutual information between sensor and actions, minimised over all optimal policies. Minimising mutual information under the constraint of a distortion measure can be done using the Blahut-Arimoto algorithm (Blahut, 1972). To obtain a policy that is optimal and minimising, Polani et al. (2006) used a Blahut-Arimoto iteration with the utility $U^\pi(s, a)$ as a distortion measure. The Blahut-Arimoto iteration is given by

$$\begin{aligned} \pi_{k+1}(a|s) &= \frac{p_k(a)}{Z_k(s,\beta)} \exp(\beta U^\pi(s,a)), \\ p_{k+1}(a) &= \sum_s p_k(s)\pi_k(a|s), \end{aligned}$$

where k denotes the iteration step, $Z_k(s, \beta)$ is a normalisation term and $\beta > 0$ a trade-of between optimality and relevant information. Now the iteration is alternated with an update of the state probabilities and a value iteration to get a consistent utility U^{π_k}.

The agents act only until they reach the goal configuration, the task is episodic. The probability to be in state s after t steps is given by

$$p(s|t) = \frac{1}{|\mathcal{S}|} \sum_{s'} P^t(s, s')$$

where P is the state transition probability matrix and a uniform distribution for $t = 0$ is assumed. Let s^{g_1}, s^{g_2} denote the two goal states. Now the probability that the agent is in state s and it has not reached the goal, denoted as living, is

$$p(s|\text{living}) = \lim_{T\to\infty} \frac{\sum_{t=0}^T \delta(s) p(s|t)}{\sum_{t=0}^T 1 - p(s^{g_1}|t) - p(s^{g_2}|t)},$$

where δ is zero if s is a goal state and one otherwise. Now we set $p(s) = p(s|\text{living})$. Updating the state probabilities is important as a correct state distribution is essential for good convergence of the algorithm.

For the whole iteration the iterations steps are then done in the following order

$$\pi_k \to p_k(s) \to V^{\pi_k} \to U^{\pi_k} \to \pi_{k+1}.$$

The algorithm then minimises the functional

$$\mathcal{L}[p(a|s)] = I(S; A) - \beta\, \mathbb{E}[U^\pi(S, A)].$$

As an optimal policy maximises the expected utility, the Lagrange multiplier β determines a trade-of between an optimal policy and limited relevant information. Iterating the algorithm for small β results in optimal policies given a limitation on the relevant information, which is of particular interest as many real world agents especially in collectives have very limited information processing capabilities. For

$\beta \to \infty$ the resulting policy is optimal and at the same time minimises the mutual information $I(S;A)$.

Recent work shows that extending relevant information to multiple steps, results in a similar algorithm that unifies the value iteration and the Blahut-Arimoto iteration and gives a new framework for minimising information quantities in Bayesian graphs under optimality constraints (Tishby and Polani, 2010). A proof of convergence for these algorithms is work in progress.

Having two agents with independent actions will change the algorithm. The iteration is now alternated between the two agents. For each agent a value iteration and a Blahut-Arimoto iteration is done using the current policy of the other agent as a predictor in the utility update. This gives the following scheme of iterations:

$$\begin{aligned} \pi^1_k, \pi^2_k \to p_k(s) &\to V^{\pi^1_k} \to U^{\pi^1_k} \to \pi^1_{k+1} \to ... \\ ... &\to V^{\pi^2_k} \to U^{\pi^2_k} \to \pi^2_{k+1}. \end{aligned}$$

First, we have the two policies for each agent from which the common environmental state distribution is calculated. This is followed by a value iteration step for the first policy and a Blahut-Arimoto update that gives the new policy for the first agent. Using this policy as a predictor the value iteration step for agent two is done, again followed by a Blahut-Arimoto step.

For most samples the algorithm converged very fast, but for certain values of β this is not the case, however, these values can be detected by taking a fine distribution of samples for β.

Results

Iterations were performed with different environment sizes (6×7,6×5,4×5, 4×3, 4×2 and $n \times 1$ with $n = 5, 6, 7, 8$). Samples were taken for different values of β ranging from 0.05 to 10.0 with steps ranging from 0.005 to 0.1, greater worlds required a larger step size due to computational limitations. Each value β leads to a policy and a state distribution, the performance of the policy can be plotted against the mutual information between actions and states (see Figure 3). At the upper limit of $\beta = 10.0$ the trade-of was already completely in favour of an optimal policy. For each sample the iteration was stopped when $\sum_s |V^\pi_{k+1}(s) - V^\pi_k(s)| < 10^{-6}$. In all runs the setup with a shared controller/policy outperforms the case where the actions are independent (see Figure 3). However the optimal ($\beta \to \infty$) shared controller shows almost no intrinsic coordination, that is $I(A^{(1)}; A^{(2)}|S)$ vanishes. Here the agents perform equally well with a shared controller as with independent controllers (see Figure 3 and 4). This suggests that in the optimal limit intrinsic coordination does not help to perform better. Similarly Zahedi et al. (2009) showed that for linked robots, those performed better that had split controllers for their motors, although this was in the context of maximising predictive information.

In the suboptimal region, especially small values of β, the shared controller performs better with the same amount of relevant information. In this region the coordination behaves differently depending on the kind of controller. With independent controllers the coordination tends to zero, as less relevant information is processed (see Figure 5). While this was expected due to coordination limited by relevant information, the coordination is not even close to the possible limit. The shared controller shows the opposite behaviour: the coordination increases as less relevant information is processed. This is also valid for the intrinsic coordination, which vanishes in the optimal limit (see Figure 4).

The maximum of coordination of the shared controller depends closely on the size and geometry of the world (see Figure 6). The spikes in the graph are due to convergence problems for certain values of β. For larger worlds the coordination still increases for $\beta \to 0$, but by a significantly smaller amount: In a 6×7 grid world the difference be-

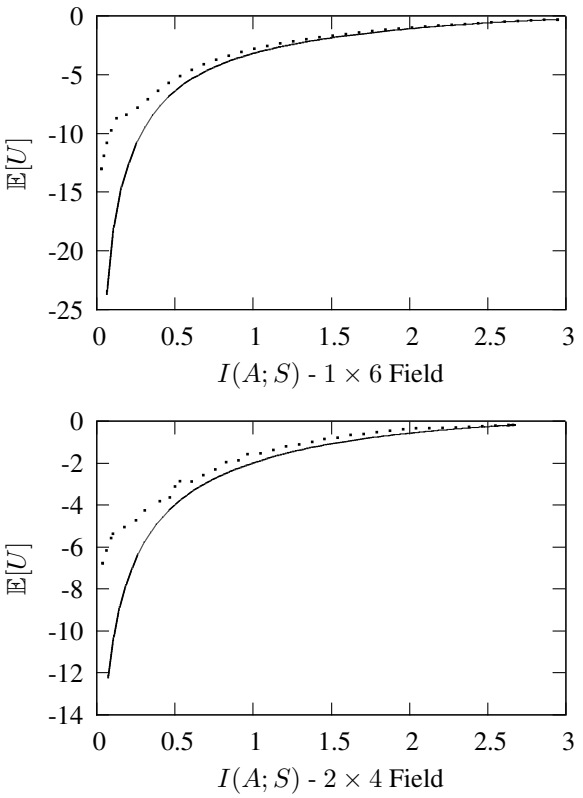


Figure 3: Performance of agents, dotted line – shared controller, solid line – individual controllers with summed expectation of utility per agent and relevant information for the joint distribution of $(a^{(1)}, a^{(2)})$. Both graphs show the same features but the scales differ.

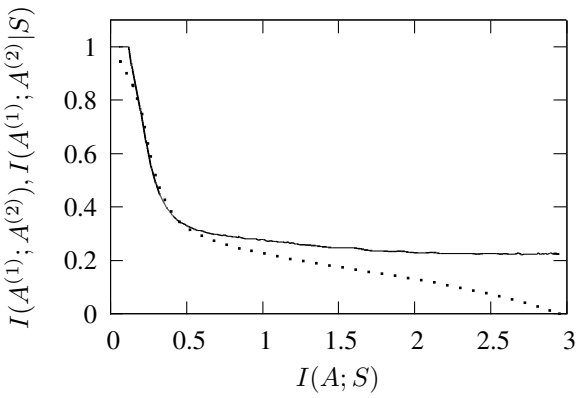


Figure 4: Coordination of agents with shared controller on a 6×1 field, dotted line – intrinsic coordination, solid line – coordination.

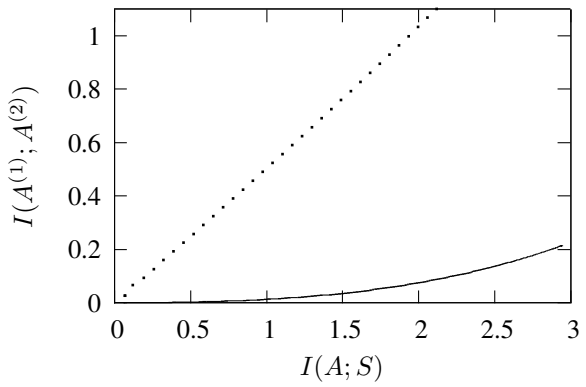


Figure 5: Solid line – coordination of agents with individual controllers on an 6×1 field, dotted line – limit given by each controllers relevant information.

tween the coordination for small and large values of β is only ≈ 0.05 bit whereas in a 4×5 world the difference is ≈ 1.54 bit. For very narrow worlds (size $n \times 1$) the coordination even reached its maximum $\max H(A^{(1)}) = \max H(A^{(2)}) = 1$ bit. It may seem unintuitive that this can happen while the relevant information is positive, as it means that one action fully determines the other and each of the two possible actions is chosen with probability $\frac{1}{2}$. However the coordination takes the expectation over all states: the actions can be totally synchronised, that is, $H(A_1|A_2) = 0$ while $H(A_1|S)$ is not maximal. Thus the distribution of the possible two synchronous actions is not uniform, but this effect can vanish when the expectation over all states is taken, which can also be seen by that fact that the intrinsic coordination does not equal the coordination and therefore the actions cannot be independent of the states.

The distribution of the states is not uniform and S has rather low entropy as the cells that are closer to the goal are visited more often by the agents. To ensure that the observed behaviour of coordination is prevalent over the whole state space and not just appearing close to the goal the resulting policies were also analysed assuming a uniform distribution of S, which resulted in insignificant differences.

Discussion

We introduced intrinsic coordination as a measure how much different agents' actions are correlated given the state of the environment. The setting we investigated is a grid world with two agents and a goal to configure in a certain way. As both agents have the same possible two goal states, they have to cooperate to reach the goal in an optimal way. The actions only depend on the current location of the agent (the agents are memoryless) thus the joint intent to move to the goal states is explicitly encoded in the controllers. Using an alternated fixed point iteration method we computed optimal policies for the agents under information processing constraints.

The results show that agents use intrinsic coordination to overcome limitations of their environment. This coordination is not needed in the optimal case where every agent can get all the relevant information from the environment that it needs to choose an optimal action. Though plausible, this is not entirely obvious a priori. One could think of various scenarios where the controllers are stochastic and the precise knowledge of the others agent action would lead to a better performance.

Now, large agent collectives will usually perform suboptimal policies as each agents' abilities will be limited: In real environments, the size of the agent and its supply of energy are just some limiting factors to information processing capabilities. Furthermore having many agents acting in the environment leads to spatial limitations that were here matched by the situation of narrow grid-worlds. In these cases intrinsic coordination performs better than just prediction of the other agents' behaviour: The shared controller cannot be split into two independent controllers, this is what we understand as 'acting as one'. The intrinsic coordination gives a measure of how strong this behaviour is. In the case of the 6×1 world and a small β the actions of the agents are always in the opposite direction, but with a small bias whether the agents move towards each other or away from each other. Despite being a feature of the controller the synchronisation does not depend on the state and there is no information needed to decide whether to act synchronised or not. The agents perform even better with this strategy. This could be interpreted as a kind of morphological computation (Pfeifer and Bongard, 2006) where the synchronisation is a feature of the embodiment of the agents used to perform better in reaching the goal configuration. Due to the symmetry of the present environment and the embodiment of the agents there is also a symmetry in the shared controller. However, intrinsic coordination does not specifically depend on symmetries

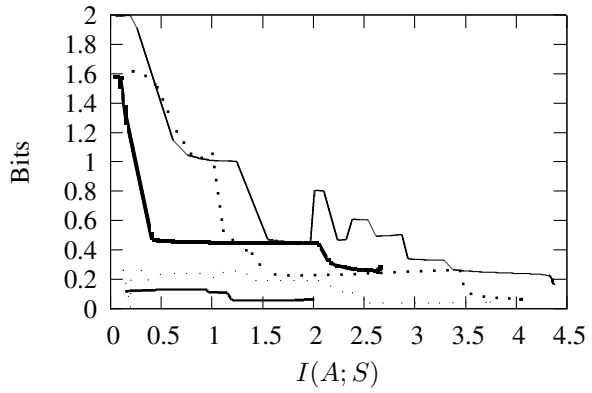


Figure 6: Coordination of agents with shared controllers on a,medium thick line – 6×7, thin dotted line – 6×5, thick dots 4×5, thick line 4×3, thin line – 4×2 field.

and can occur in any scenario within this formalism.

In the setup the intrinsic communication is not limited: the two agents share a common 'brain'. But often coordination is only 'routed' through the environment: In the case of stigmergy the environment takes the role of the communication channel (Klyubin et al., 2004b). Other ways of communication that have low interference with the environment like sound, dissolving molecules or radio signals qualify more to be modelled as intrinsic coordination, although their limited channel capacities must be considered. In our experiment intrinsic coordination was not modelled using directed communication and the agents came to a instantaneous joint decision. What we have not done here, but to what the formalism could be changed, is a dependence of $A^{(2)}$ on $A^{(1)}$, which would model connected controllers where the first agent can express an intent to which the second can react. This would be a more restrictive model than the shared controller. Moreover this framework can be further elaborated to take issues of time shifts and turn taking during the decision process into account. Examples where collectives of cells use molecular signalling, with almost no interference, to activate a certain behaviour in the whole collective (Marée and Hogeweg, 2001) could then be modelled as intrinsic coordination. One can argue that the molecular signalling should be modelled with each cell having an independent controller and a sensor for these molecules, but a model allowing intrinsic communication could lead to a simpler description and therefore be more preferable.

Furthermore it is not necessarily obvious whether a particular collective of agents is just a collection of individuals or acts as one individual. If there is a simpler model allowing intrinsic coordination does that automatically mean that it acts as a single entity? Ant colonies are sometimes called super-organisms (Theraulaz and Bonabeau, 1999) and were recently found to fulfil certain laws that apply for animals (Hou et al., 2010), melting the boundary between the individual and the collective. If two agents have the possibility of maximal intrinsic coordination they can hardly be viewed as individual agents as their actions are completely synchronised. Thus having non-maximal intrinsic coordination gives each agent a certain degree of freedom to decide for an action solely on its own perception of the environment. This means that a collective with a shared centralized controller still can undertake actions that conflict each other, especially in the suboptimal case, but intrinsic coordination can be used to avoid this to a certain degree. In the spirit of defining autonomy for a system in an information theoretic way (Bertschinger et al., 2008), intrinsic coordination could function as another measure of individuality or autonomy with respect to other agents.

References

Ashby, W. R. (1956). *An Introduction to Cybernetics*. Chapman & Hall Ltd.

Ay, N., Bertschinger, N., Der, R., and Güttler, F. (2008). Predictive information and explorative behavior of autonomous robots. *The European Physical Journal B - Condensed Matter and Complex Systems*, 63(3):329–339.

Beckers, R., Holland, O., and Deneubourg, J. (1994). From local actions to global tasks: Stigmergy and collective robotics. In *Artificial Life IV*, pages 181–189. MIT Press.

Bertschinger, N., Olbrich, E., Ay, N., and Jost, J. (2008). Autonomy: An information theoretic perspective. *Biosystems*, 91(2):331–345.

Blahut, R. (1972). Computation of Channel Capacity and Rate Distortion Functions. *IEEE Transactions on Information Theory*, 18(4):460–473.

Castelfranchi, C. (2006). Silent agents: From observation to tacit communication. *Advances in Artificial Intelligence-IBERAMIA-SBIA*, 4140:98–107.

Couzin, I. D., Krause, J., Franks, N. R., and Levin, S. a. (2005). Effective leadership and decision-making in animal groups on the move. *Nature*, 433(7025):513–6.

Cover, T. M. and Thomas, J. A. (2006). *Elements of Information Theory 2nd Edition*. Wiley Series in Telecommunications and Signal Processing. Wiley-Interscience.

Dautenhahn, K. (1995). Getting to know each other–artificial social intelligence for autonomous robots. *Robotics and Autonomous Systems*, 16:333–356.

Dautenhahn, K. (1999). *Embodiment and interaction in socially intelligent life-like agents*, pages 102–142. Springer Lecture Notes in Artificial Intelligence.

Deneubourg, J. and Goss, S. (1989). Collective patterns and decision-making. *Ethology, Ecology and Evolution 1*, pages 295–311.

Deneubourg, J., Goss, S., and Franks, N. (1991). The dynamics of collective sorting: Robot-like ants and ant-like robots. *Proceedings of the first international conference on simulation of adaptive behavior*.

Di Paolo, E., Rohde, M., and Iizuka, H. (2008). Sensitivity to social contingency or stability of interaction? Modelling the dynamics of perceptual crossing. *New Ideas in Psychology*, 26(2):278–294.

Goldstone, R. and Janssen, M. (2005). Computational models of collective behavior. *Trends in Cognitive Sciences*, 9(9):424–430.

Hou, C., Kaspari, M., Vander Zanden, H. B., and Gillooly, J. F. (2010). Energetic basis of colonial living in social insects. *Proceedings of the National Academy of Sciences*, 107(8):3634–8.

Ikegami, T. and Iizuka, H. (2007). Turn-taking interaction as a cooperative and co-creative process. *Infant Behavior and Development*, (30):278–288.

Klyubin, A. S., Polani, D., and Nehaniv, C. L. (2004a). Organization of the Information Flow in the Perception-Action Loop of Evolved Agents. In Zebulum, R. S., Gwaltney, D., Hornby, G., Keymeulen, D., Lohn, J., and Stoica, A., editors, *Proceedings of 2004 NASA/DoD Conference on Evolvable Hardware*, pages 177–180. IEEE Computer Society.

Klyubin, A. S., Polani, D., and Nehaniv, C. L. (2004b). Tracking Information Flow through the Environment: Simple Cases of Stigmergy. In Pollack, J., Bedau, M., Husbands, P., Ikegami, T., and Watson, R. A., editors, *Artificial Life IX: Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems*, pages 563–568, Cambridge, MA. MIT Press.

Klyubin, A. S., Polani, D., and Nehaniv, C. L. (2007). Representations of Space and Time in the Maximization of Information Flow in the Perception-Action Loop. *Neural Computation*, 19(9):2387–2432.

Langton, C. (1997). *Artificial life: An overview*. The MIT Press.

Laughlin, S. B. (2001). Energy as a constraint on the coding and processing of sensory information. *Current Opinion in Neurobiology*, 11(4):475–480.

Linsker, R. (1988). Self-Organization in a Perceptual Network. *Computer*, 21(3):105–117.

Marée, A. F. and Hogeweg, P. (2001). How amoeboids self-organize into a fruiting body: multicellular coordination in *Dictyostelium discoideum*. *Proceedings of the National Academy of Sciences of the United States of America*, 98(7):3879–83.

Nabet, B., Leonard, N., Couzin, I., and Levin, S. (2009). Dynamics of Decision Making in Animal Group Motion. *Journal of Nonlinear Science*, 19(4):399–435.

Nehaniv, C., Polani, D., and Olsson, L. (2007). *Information-Theoretic Modeling of Sensory Ecology: Channels of Organism-Specific Meaningful Information*, pages 241–282. MIT Press.

Pfeifer, R. and Bongard, J. (2006). *How the Body Shapes the Way We Think: A New View of Intelligence*. MIT Press, Cambridge, MA.

Polani, D., Martinetz, T., and Kim, J. T. (2001). An Information-Theoretic Approach for the Quantification of Relevance. In *ECAL '01: Proceedings of the 6th European Conference on Advances in Artificial Life*, pages 704–713, London, UK. Springer-Verlag.

Polani, D., Nehaniv, C. L., Martinetz, T., and Kim, J. T. (2006). Relevant Information in Optimized Persistence vs. Progeny Strategies. In *Artificial Life X : Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, pages 337–343. The MIT Press (Bradford Books).

Polani, D., Sporns, O., and Lungarella, M. (2007). How Information and Embodiment Shape Intelligent Information Processing. In *50 Years of Artificial Intelligence*, pages 99–111.

Shalizi, C. R. and Crutchfield, J. P. (2002). Information Bottlenecks, Causal States, and Statistical Relevance Bases: How to Represent Relevant Information in Memoryless Transduction. *Advances in Complex Systems*, 5:91.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423.

Sperati, V., Trianni, V., and Nolfi, S. (2008). Evolving coordinated group behaviours through maximisation of mean mutual information. *Swarm Intelligence*, 2(2-4):73–95.

Sutton, R., Precup, D., and Singh, S. (1999). A Framework for Temporal Abstraction in Reinforcement Learning. *Artificial intelligence*, 112(1-2):181–211.

Theraulaz, G. and Bonabeau, E. (1999). A brief history of stigmergy. In *Artificial Life V*, pages 97–116. MIT Press.

Tishby, N., Pereira, F. C., and Bialek, W. (1999). The information bottleneck method. In *The 37th annual Allerton Conference on Communication, Control, and Computing*, pages 368–377.

Tishby, N. and Polani, D. (2010). Information Theory of Decisions and Actions. In Cutsuridis, V., Hussain, A., and Taylor, J., editors, *Perception-Reason-Action Cycle: Models, Algorithms and Systems.* Springer. In Press.

Touchette, H. and Lloyd, S. (2000). Information-theoretic limits of control. *Physical Review Letters*, 84(6):1156–1159.

Walter, W. (1950). An imitation of life. *Scientific American*, (May):42–45.

Zahedi, K., Ay, N., and Der, R. (2009). Higher coordination with less control - A result of information maximisation in the sensori-motor loop. *CoRR*, abs/0910.2.

Multisensory Perceptual Discrimination in Evolved Networks and Agents

Marieke Rohde[1]

[1]Max Planck Institute for Biological Cybernetics, Tübingen, Germany
marieke.rohde@tuebingen.mpg.de

Abstract

The fact that humans and animals have several sensory modalities and use them together to make sense of the world imbues their behaviour with an immense richness and robustness. In this study, recurrent neural networks and minimal agents with active vision are evolved for a perceptual discrimination task (unimodal and bimodal). The purpose of this study is mainly exploratory: to test which of the characteristics of human perceptual discrimination evolve easily (with a focus on statistically optimal integration), how they are realised and what active perception does in this process. Whilst some of the systems evolved to perform perceptual discrimination well, they did not conform to the predictions from statistical optimality. Analyses of the systems point towards a number of relevant issues, noticeably towards the lack of a good account of 'unimodality' in existing models of multisensory perception.

Introduction

Humans and animals use several sensory modalities to make sense of the world and to judge on and distinguish objects in the environment. For instance, the size of an object can be judged both by touching the object or by looking at it, or by doing both at the same time. In humans, it could be shown that subjects, when estimating object size, integrate visual and tactile cues in a statistically optimal fashion to decrease uncertainty (Ernst and Banks, 2002). Similar findings were reported from other multisensory tasks, e.g., audio-visual sound localization (Alais and Burr, 2004).

These kinds of results are usually obtained using a psychophysics approach, where subjects are asked to perform perceptual judgments on stimuli that are varied systematically along a physical dimension. Comparing the human behaviour to that of an 'ideal observer' using maximum likelihood estimation (MLE), the mentioned findings of optimality are derived. This approach is *prima facie* behaviour-based; the underlying mechanisms of (optimal) multisensory integration are not yet well understood. Under the dominant representationalist paradigm, we would expect a dedicated internal neural mechanism to implement MLE. Accordingly, Knill and Pouget (2004) rephrase the problem of statistically optimal multisensory integration as follows: "(i) how do neurons, or rather populations of neurons, represent uncertainty, and (ii) what is the neural basis of statistical inferences?" and review candidate neural correlates.

By contrast, Artificial Life and dynamical approaches in cognitive science have repeatedly shown that efficient, robust or plausible models exist that do not rely on local computation but on agent morphology, contingencies in agent-environment interaction or on non-linear dynamics in neural control. Examples of such models in perception research include active vision to solve a non-Markovian visual discrimination task with feed-forward control (Floreano et al., 2004; Izquierdo-Torres and Di Paolo, 2005), agency detection by emergent behavioural coordination (Di Paolo et al., 2008) or olfactory perception through chaotic neural dynamics (Freeman, 1987). These models do not just point out alternatives, they also show that, if global dynamics are taken into consideration, many phenomena that appear complex emerge effortlessly.

For the study presented, recurrent neural network controllers and minimal agents with an active vision system were evolved to solve a size discrimination task. Such an evolutionary robotics (ER) approach has been argued to minimise prior assumptions about underlying mechanisms by outsourcing the design to an automated search procedure (Harvey et al., 2005). The purpose was mainly exploratory: if no constraints of optimality are imposed, which, if any of the hallmarks of MLE optimal integration evolve? How do the systems realize perceptual discrimination? How do they integrate their senses and how do they deal with varying levels of uncertainty? Comparing a disembodied network and an embodied agent, what are the differences and commonalities? Are there advantages associated with active perception in this task?

The results presented can be seen as work in progress. They point out issues that require a rethinking of the approach taken here. While some of these difficulties are of a more technical nature, others proved to be insightful with respect to the overarching question of (optimal) multisensory integration. In particular, the question of what unimodality means in a system with several sensory channels is of

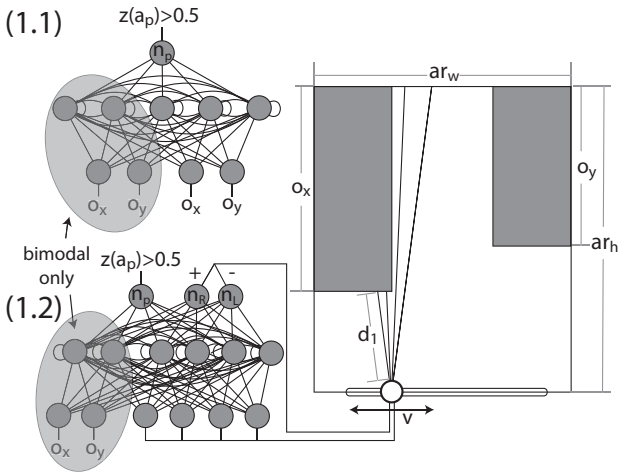


Figure 1: Evolved networks for the direct condition (1.1) and for the active vision condition (1.2).

potential importance for the study of multisensory integration in general. The results confirm that emphasizing the non-obvious is one of the key characteristics and merits of generative ER modelling.

Methods

Simulation and Genetic Algorithm

Continuous-time recurrent neural networks (CTRNNs; e.g., Beer, 2003) are evolved to solve a two-alternative forced-choice (2AFC) size discrimination task. The decision, which of two objects $o_x, o_y \in [1, 2.5]$ is larger is either generated by an agent controlled by a CTRNN or by a CTRNN directly. The dynamics of units in a CTRNN is governed by

$$\tau_i \frac{da_i(t)}{dt} = -a_i(t) + \sum_{j=1}^{N} w_{ij} z(a_j(t) + \theta_j) + I_i(t) \quad (1)$$

where $z(x)$ is the standard sigmoidal function $z(x) = 1/(1 + e^{-x})$, $a_i(t)$ is the activation of unit i at time t, θ_i is a bias term, τ_i is the activity decay constant, w_{ij} is the strength of a connection from unit j to unit i. The structure of the network is partially layered, network sizes vary between conditions (see Fig. 1). Neural and environmental dynamics were simulated using the forward Euler method with a time step of $h = 1ms$.

For all controllers, input signals are fed into input units n_i by $I_i(t) = Sg_i \cdot inp + \nu\epsilon$, where Sg_i is the evolved sensory gain, inp is the input signal, ϵ is a normally distributed random variable and $\nu \in [0, 3, 6, 9, 12]$ is the level of sensory noise that modulates channel reliability across trials. In the network condition, the inputs $inp = o_x, o_y$ are fed directly into the network (see Fig. 1, 1.1). The active vision agent, inspired by (Beer, 2003), can move left and right by $v = Mg \cdot (z(n_l) - z(n_r))\ units/s$ in an arena of random width $ar_w \in [3.5, 4]$ and depth $ar_d \in [4.5, 5]$ (see Fig. 1, 1.2). The agent has a vision system comprised of four rays with angles $[-7.5°, -2.5°, 2.5°, 7.5°]$ and perceives distance by $inp_i = d_i/5$ where d_i is the distance at which a ray i is intercepted. All controllers are evolved for both a 'unimodal' and a 'bimodal' condition. In the bimodal condition, controllers are given a redundant direct input channel and two additional hidden units (see Fig. 1).

An output unit n_p generates a perceptual estimate: $z(a_p) > 0.5$ means a perceived $o_x > o_y$ at the end of a trial. This leads to the following performance criterion for pairs of objects (o_x, o_y)

$$P(o_x, o_y) = \begin{cases} 1 & \text{if } (z(a_p) > 0.5) = (o_x > o_y) \\ 0 & \text{else} \end{cases} \quad (2)$$

Fitness for individual controllers is computed according to

$$F = \frac{(1 - RB)}{16} \sum_{i=0}^{16} P(o_x, o_y) \cdot P(o_y, o_x) \quad (3)$$

where $o_x, o_y \in [1, 2.5]$ are drawn from a uniform distribution. As pairs are presented in both orders for F, evaluation involves $2 \times 16 = 32$ trials. The response bias $RB \in [0, 1]$ is proportional to the amount by which $z(a_p) > 0.5$ has a bias stronger than 75% to either side. The multiplicative term and the punishment for response bias were included after piloting because evolved systems tended to be very accurate but strongly biased towards one side. Object presentation lasts $T \in [3000, 4000ms]$ for networks ($+t_{pre} \in [100, 500ms]$ without stimulus) and $T \in [16000, 18000ms]$ for agents. Networks are initialised randomly and agents are positioned on the mid point of the line along which they can move.

CTRNNs are evolved using a generational GA with a population of 30 and are selected using truncation selection $(1/3)$. Genes are real-valued $\in [0, 1]$ with vector mutation $r \in [0.3, 0.5]$ and reflection at gene boundaries. Evolved gene values are linearly mapped onto the target range for $w_{ij} \in [-8, 8], \theta_i \in [-3, 3]$ and exponentially for $Sg \in [0.1, 20], Mg \in [0.1, 100]$ and $\tau_i \in [30, 3000ms]$ (networks) or $\tau_i \in [30, 10000ms]$ (agents) respectively. For the hidden and output layer, $\theta_i = -0.5 \sum_{j=0}^{N} w_{ij}$ (center-crossing).

ν is drawn randomly each trial from the available range of noise levels. Evolution starts noiseless (ν=0) and the maximum level of noise is increased every time average top performance over 50 generation exceeds $\bar{F} = 0.5$ till the full range ($\nu \in [0, 3, 6, 9, 12]$) is reached. In the bimodal condition, two quarters of the trials were unimodal trials (one quarter for each channel) to avoid specialization. This means that one modality received no signal but instead strong noise with $\nu = 15$. Otherwise, noise in the first channel was random as in the unimodal condition, whereas noise in the second channel was fixed at $\nu = 6$.

Analysis

Perceptual discrimination and integration is analysed just as in human psychophysics (e.g., Ernst, 2005). Perceptual response probability is described as a cumulative probability function ('psychometric curve') of real differences in object sizes. Evaluation is performed presenting a standard stimulus $o_s = 1.75$ to one side and a comparison stimulus $o_c \in [0.3o_s, 1.7o_s]$ to the other side. Each measurement is repeated 20 times. This procedure is repeated for both sides and for all levels of noise ν. Cumulative Gaussians are fitted to the responses using the Matlab toolbox psignifit for maximum likelihood fitting (Hill, 2005). The 50% level of a psychometric curve is called the PSE (point of subjective equality) and corresponds to the mean of the fitted Gaussian. It indicates perceptual bias. The difference between the 50% and the 84% is called the JND (just-noticeable-difference) and corresponds to $\sqrt{2}\sigma$ of the underlying Gaussian. It indicates perceptual accuracy.

Optimal integration is assessed by comparing the evolved system's perceptual discrimination with an ideal observer model using MLE and an independent channel model. In such a model, a bimodal perceptual estimate S^* is generated as a weighted sum of unimodal estimates (i.e., $S^* = w_1 S_1 + w_2 S_2$) in a way that minimizes uncertainty. MLE generates the following testable predictions (cf. Ernst, 2005; Ernst and Banks, 2002):

$$w_1 + w_2 = 1 \quad w_i = \frac{1/\sigma_i^2}{1/\sigma_1^2 + 1/\sigma_2^2} \quad \sigma^{*2} = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2} \quad (4)$$

The first term indicates multisensory integration in general, whereas the second and third term are characteristic of optimal integration in particular. These criteria also clarify the significance of the noise level ν as the parameter that should modulate σ_i. According to the predictions, the weights w_i and σ^* should change with σ_i (in particular, bimodal discrimination should be more accurate than each of the unimodal discriminations).

To compute the weights, crossmodal conflicts $c \in [-.25o_s, .25o_s]$ are introduced during testing, i.e., for one modality $o_s^1 = o_s - 0.5c$ and for the other modality $o_s^2 = o_s + 0.5c$. Integration occurs if, in the presence of conflicts, PSEs are shifted along the $[o_s - 0.5c, o_s + 0.5c]$ interval according to the weights. σ_i can be computed by $JND = \sqrt{2}\sigma_i$.

Perceptual Discrimination in Recurrent Neural Networks

Evolving perceptual discrimination in recurrent neural networks is a less biased approach to the study of perceptual integration because it allows for the evolution of dynamically complex solutions and functional intertwinement: solutions evolved may not employ separate populations of neurons to perform different tasks, such as unimodal estimation, integration and measuring uncertainty. Also, given that the fitness function Eq. (3) does not require optimal integration, there is the possibility that optimality spontaneously emerges.

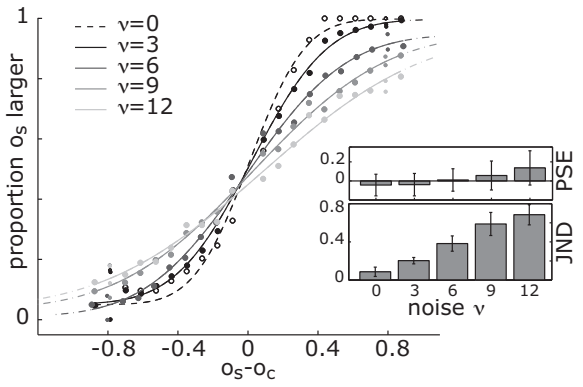


Figure 2: Unimodal networks. Psychometric curves for the different noise levels ν, data pooled from all 7 networks and both orders. Inlay: mean and s.e.m. for fitting parameters PSE (bias) and JND (accuracy) from individual fits (average of both stimulus orders; $N = 7$).

Unimodal Networks

The purpose of the unimodal condition was primarily to verify that the task is suitable for the study of perceptual discrimination. In order to allow the evolution of optimal integration, controllers have to perform perceptual discrimination sufficiently well. Their accuracy should decrease with the level of noise (JND should increase) to make it possible to test for statistically optimal integration.

CTRNNs were evolved in 20 evolutionary runs with 1000 generations. 7 of the 20 networks evolved performed sufficiently well according to these criteria. The main exclusion criterion pointed towards a very successful but trivial local maximum for this task (up to $F \approx 0.6$): 7 networks were excluded because they considered only one stimulus and judged if it is 'big or not', which means that performance is good during testing for the standard o_s on one side, but at chance level or substandard for the other side.

Figure 2 depicts the psychometric curves for the different noise levels ν for all 7 successful networks together, as well as the JNDs and PSEs from individual fits. Increase in ν leads to a clear increase in JND (1 factor ANOVA: $F(4, 2) = 7.55, p < 0.001$), while PSEs are not influenced by noise ($F(4, 2) = 0.25, p = 0.91$). The successfully evolved networks show that, given the task and the fitness criterion, artificial systems can evolve to generate behaviour and simulated data that can be compared to human data and that can be analysed the same way.

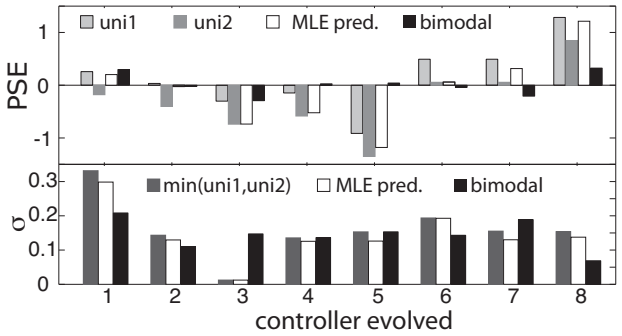


Figure 3: Unimodal, bimodal and predicted PSE (top) and σ (width of fitted Gaussians, bottom) for all networks evolved to perform (partial) bimodal discrimination.

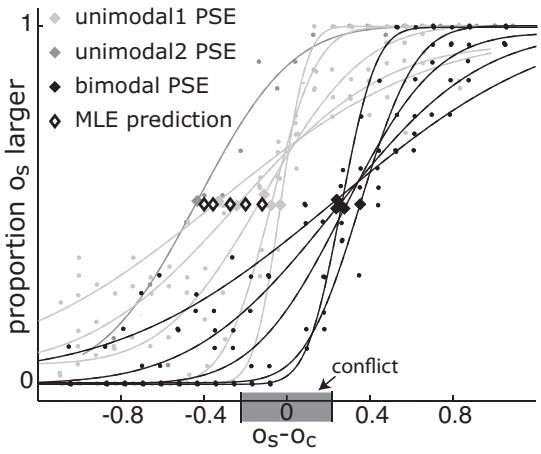


Figure 4: Example psychometric curves for the most successful network with $\nu = 0$ in the silent channel. $c = -0.25$, all noise levels ν. Data pooled for c_s left/right. Unimodal curves are shifted along the x-axis according to the conflict.

Bimodal Networks

In the bimodal condition, the emphasis is on the kind of integration behaviour that the networks exhibit and if it conforms to the predictions from MLE in Eq. (4).

Controllers for the bimodal condition were evolved in 20 evolutionary runs with 2000 generations. Only one network evolved to successfully discriminate between objects for all orders in both the unimodal and the bimodal conditions. The simulated data was fitted and analysed like in the previous simulation. When comparing the JND of the unimodal and the bimodal condition for the successfully evolved network, at first glance it appeared to exhibit the most important hallmark of MLE, i.e., that the probability distribution of bimodal estimates was more accurate than either of the unimodal estimates. However, testing the exact predictions from MLE (Eq. (4)) on this controller, the network proved to be *super-optimal*, i.e., the accuracy (in terms of σ of the fitted Gaussian) was dramatically better than expected from MLE (Fig. 3, bottom left).

7 of the other controllers evolved performed satisfactorily for both modalities if the standard o_s was presented to one side only. They were analysed and compared to the predictions of MLE as well. Even if lateral specialization is unsatisfactory concerning the main question, it involves some degree of integration. Figure 3 (bottom) depicts σ for the bimodal condition, averaged over noise levels ν, in comparison to the lower of the unimodal σ and the predicted σ using Eq. (4). All controllers were either grossly super-optimal or less accurate than the better of the uni-modal conditions, i.e., there was no evidence for optimal integration.

Why is it so easy to be 'better than optimal'? Is it because of the noise $\nu = 15$ of the inactive channel disturbs the network in the unimodal condition? Controllers were tested again with $\nu = 0$ in the unimodal condition to test this assumption. Contrary to the expectations, taking out the noise, in most cases (5 of the 8 networks), did not improve unimodal accuracy, but led to a complete break-down of unimodal discrimination. This indicates that the noise served a functional purpose in integration.

Defining the unimodal condition as noise with $\nu = 15$ and the absence of a signal had been an arbitrary design decision. However, as it is the case in biological evolution, the GA worked with what was there and thus incorporated this noise functionally into the solution, with surprising effects on perceptual accuracy across conditions. This result raises the question of what 'uni-modality' means in a multimodal system which will be picked up in the discussion. For those networks that also worked in the absence of noise, discrimination during unimodal trials became better than during bimodal case, eliminating the super-optimality. This result supports the hypothesis that noise in the silent channel is the reason for bimodal super-optimality.

Maybe more surprising still is the fact that the controllers did not evolve to integrate the two estimates. Introducing a cross-modal conflict, networks would be expected to generate PSEs in between the PSEs that the unimodal data predicts. Figure 3 (top) shows that, in the large majority of cases, the PSE of bimodal networks is far outside this range and, therefore, also far away from the PSE predicted from MLE. Figure 4 shows this behaviour for the most successful network (with $\nu = 0$ in the inactive channel): the discrimination is successful for all noise levels for both the unimodal and the bimodal stimuli. Accuracy for the bimodal trials is comparable to the unimodal trials. However, the PSE is far outside the range that would indicate integration. Rather than to integrate uni-modal estimates, the networks had evolved to perform a different and comparably viable way of discriminating size in the presence of redundant signals. The result indicates that multi-modal integration, as it is characteristic of humans, is not a process that simply

emerges as an epiphenomenon of the existence of redundant sensory channels but probably evolved due to more specific adaptive needs. The previously mentioned tendency of networks to evolve solutions with strong perceptual biases in this task is likely to also play a role in this result.

The solutions evolved do not make use of the dynamic complexity afforded by the recurrent network structure - they rely mainly on feed-forward principles. The passive open-loop nature of the task for disembodied recurrent networks does not encourage the use of dynamic complexity.

Perceptual Discrimination in Simple Agents

Living organisms are always in dynamic interaction with the environment. The surge of sensorimotor approaches in perception research (e.g., O'Regan and Noë, 2001) reflects an increasing awareness that such closed-loop dynamics afford alternative and clever ways of solving perceptual tasks. Existing models of optimal integration assume that integration, as well as estimation of channel certainty and weight adjustment are performed internally. The objective of evolving simple vision agents for this task was to explore if and how active perceptual strategies can play a role in multisensory integration and perceptual discrimination.

To bootstrap the evolution of active perceptual strategies, the performance criterion Eq. (2) was amended such that agents receive $P = 0.1$ if their visual system perceives both objects at least once, even if the wrong decision is made. If they do not move to see both objects, they receive $P = 0$, even if the right decision was made. In 20 evolutionary runs with 1000 generations, not one controller evolved that could reliably distinguish objects of different sizes for the whole problem space: local maxima, in most cases the mentioned solution to only pay attention to one of the stimuli, could not be overcome. Variations of the task were explored to mitigate this problem, including a punishment for lateral specialization and the administration of an extra position sensor, but performance never exceeded the stable local maximum, i.e., to focus just on one side. This suggests that a more radical change of fitness criterion/task may be necessary.

Controllers were also evolved for the bimodal condition in 16 runs for 2000 generations. The possibility exists that the presence of a direct sensory channel serves as a guidance for the evolution of active visual discrimination. Instead, the agents evolved rely heavily on their second (direct) input channel (see Fig. 1) and did not evolve to use their active sense according to demand. Where partially viable behaviour evolved, it replicates the general results from disembodied networks.

While these performance deficits mean that the predictions of the ideal observer model could not be tested, it is still interesting to test whether the partial solutions evolved exhibit sensorimotor strategies for sub-parts of the problem space. If agents evolve to base their decision on one input only, they could just evolve to move over to one side (passing the other side briefly to fulfill the revised performance criterion) and, otherwise, act as if they had a direct input channel. Instead, nearly all agents exploit their capacity to act in the closed-sensorimotor loop in order to make the 'big or not' strategy more effective. The remainder of this section presents examples of such active sub-strategies.

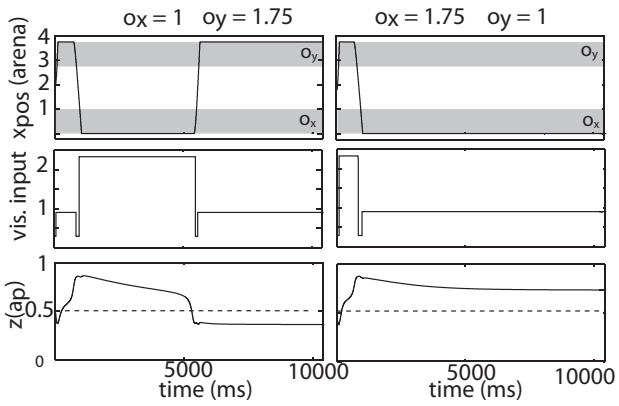


Figure 5: Selected variables across time from an agent presented with two pairs of objects with $o_x < o_y$ (left) and $o_x > o_y$ (right). Top: position. middle: sensory input from one input unit. bottom: decision output $z(n_p)$

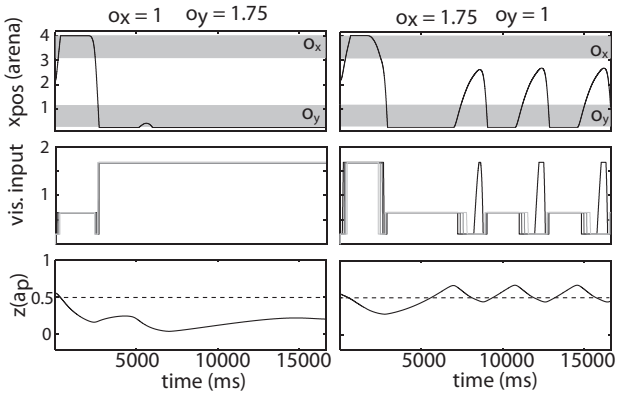


Figure 6: Selected variables across time from and agent presented with two pairs of objects with $o_x < o_y$ (left) and $o_x > o_y$ (right). Top: position middle: sensory inputs bottom: decision output $z(n_p)$

Active decision making. Figure 5 depicts the motion, inputs and decision output over time for an agent evolved. The agent evolved, under some circumstances, to steer towards the smaller of the two objects and to then make the decision contingent on the output velocity (using internal activation like an efference copy). This active decision making capacity is the most straight-forward one of the ones evolved and is an exception to the trend to pay attention to one input only.

*Active decision expression.*The agent depicted in Fig. 6

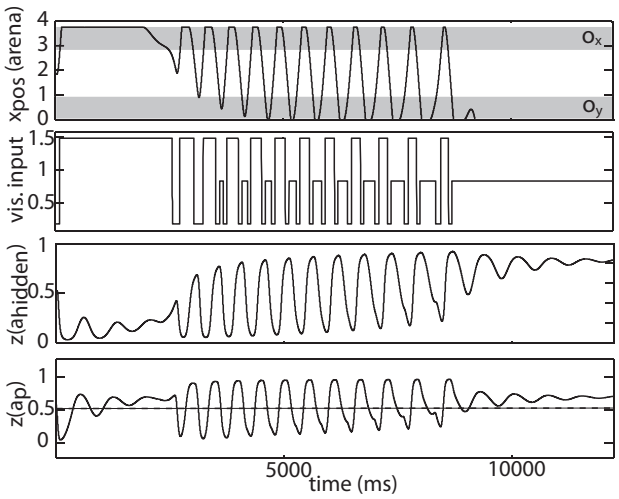


Figure 7: Selected variables across time from and agent presented with one pair of objects with $o_x > o_y$. Top: position. Second: sensory input from one unit. Third: $z(a)$ from a selected hidden unit. Bottom: decision output $z(n_p)$

evolved to only pay attention to the second input o_y. If the agent deems it large (Fig. 6 left), it comes to a halt and constantly outputs its decision ($z(a_p) < 0.5$). If, however, it deems the object small (Fig. 6 right), it initiates an oscillation towards and away from the object. Driven by this oscillation, the decision output starts oscillating around the decision boundary at $z(a_p) = 0.5$. This kind of behaviour evolved very frequently. It provides the agents with a way of expressing uncertainty: depending on when the trial ends, the same input would lead to different answers, and slight differences in object size may bias the proportion of such decisions by modulating the oscillations. Probably, such strategies evolved at least partially in response to the RB term in the fitness function Eq. (3) that punishes a strong response bias: if some of the decisions are random, it is unlikely that more than 75% of decisions would be of one kind.

Temporal decision making. Figure 7 depicts an agent's dynamics during the presentation of a single pair of objects. The agent's strategy makes active use of the time allocated for making a decision. One hidden unit (Fig. 7, third) controls the position of the agent: it decreases activity dramatically in the beginning (steering to the right) and then slowly increases. When it reaches a certain threshold, the agent starts moving to the left. Reaching the gap between the objects, the agent starts oscillating between the two objects, which is reflected in the activity of the hidden unit, too. The output unit always decides o_x is larger ($z(a_p) > 0.5$), unless the oscillations pull it below this threshold. Therefore, oscillation stands in correlation with the decision that o_x is smaller. The oscillation can only be stopped in time before the trial ends if the second object is small enough, otherwise it will go on indefinitely or at least till the end of the trial. In that sense, this controller can be seen as a variant of the o_y only strategy. The length of the oscillatory phase is, however, not just contingent on o_y. The size of o_x appears to take influence on the time of onset of the oscillations as well as its offset in ways that are not obvious.

These are just three examples of the ways in which agents used their motion capacities in their size discrimination activity, not all of which are easy to understand. In depth analysis of only partially functional agents is an endeavour of limited value. The fact that an abundance of active strategies evolved, however, is a result worth mentioning. In systems that discriminate stimuli exploiting the agent-environment interaction dynamics, processes of multisensory integration would rely on these closed-loop dynamics. How (optimal) integration could work in the absence of explicit representation of perceptual estimates remains an intriguing open question.

Discussion

Using ER for this kind of multisensory perceptual discrimination task is a novel approach and as such the research presented has mainly exploratory character. Both technical and conceptual difficulties were encountered. Most dramatically, minimal agents could not be evolved to perform perceptual discrimination and the predictions from MLE could not be tested for the second part of the project. ER simulation modelling serves as a tool for thinking, and as such, the simulation results here presented have pointed out a number of issues that are worth reporting.

Unimodality in a Bimodal System

Possibly the most important insight gained from the simulation models is that existing models of optimal integration have a gap to fill: as humans, it is obvious for us what a unimodal and what a bimodal stimulus is. It is, however, not clear how the MLE circuits proposed (e.g. Knill and Pouget, 2004; Ernst and Banks, 2002; Alais and Burr, 2004) or a localized brain area would be able to recognise the absence of a signal in one channel and what possible noise entering through that channel can do to the decision making process. MLE assumes independent channels and independent processes of unimodal estimation and multisensory integration (cf. Method section). How the same process of generating perceptual judgments in human observers can be indicative of either of the stages is not made clear in existing models. In the model presented, the administration of random noise in the silent channel led to the evolution of apparent 'super-optimality' in bimodal trials: not because networks accurately estimate the levels of noise present, but just because additional noise sources were absent during bimodal trials. The fact that performance breaks down in most controllers when the noise is removed shows that the definition

of what 'uni-modal' means in a system is not an arbitrary one. Existing models of optimal integration would benefit from making explicit the behaviour of the inactive channel during unimodal trials and incorporating mechanisms into their models that distinguish between multimodal and bimodal trials. Testing for their existence can then confirm that the reported increase in accuracy in bimodal trials is not due to the influence of the silent channel during 'unimodal' trials.

Perception vs. Perceptual Judgments

Unlike humans, the evolved systems were surprisingly incapable to integrate their senses in a coherent way. This problem may well be due to the fact that the controllers were evolved for a laboratory task. 2AFC perceptual discrimination tasks, like the size discrimination task used here, make it possible to measure perceptual accuracy, as well as perceptual bias. The fitness criterion Eq. (3) emphasises this accuracy component. Therefore, the systems evolved tend to favour being accurate over the absence of perceptual biases (as evident from the large and variable PSEs in Fig. 3) and are rewarded for this tendency. Humans, on the other hand, develop their perceptual skills not for this kind of psychophysics task, but in real-world situations, where perception has behavioural relevance. In many real-world contexts, strong or variable perceptual biases would be extremely disadvantageous. In future research, therefore, systems will not be evolved for 2AFC tasks exclusively, but for perceptual capacities more generally (e.g., the approach taken here can be combined with a magnitude estimation task or with a sensorimotor control task that involves perceptual decision making).

Ideal Observing vs. Active Sensing

Ideal Observer Models of perceptual integration strongly draw on the assumptions of the dominant representationalist paradigm in cognitive science: MLE is a dedicated process that combines unimodal estimates and noise estimates. Even though behavioural approaches (e.g. Ernst and Banks, 2002; Alais and Burr, 2004) are *prima facie* agnostic about the underlying mechanisms, it is easy to jump to conclusions and assume that internal dedicated neural process perform MLE, represent the noise, represent the unimodal estimates, etc. (e.g. Knill and Pouget, 2004). Evolving embodied agents to integrate their senses optimally (on a behavioural level) can potentially challenge such underlying assumptions (on the level of the underlying mechanism). The active vision agents presented here did not arrive at a level of behaviour that would allow drawing strong conclusions about multisensory integration. However, even superficial analysis of their behaviour revealed an abundance of active sensing in the accomplishment of aspects of perceptual discrimination, including but not limited to active decision making and the expression of uncertainty through motion patterns.Thinking of the human hand and the human eye as agents, it is not unlikely that active sensing principles are exploited in a task like visuo-haptic size estimation. It is by no means clear that the introduction of noise or the variation of physical parameters, like in psychophysics, would have the same impact on such embodied processes as they have on decoupled systems that are passively cruncing representations. Even though limited in their own significance, the present results provide a good incentive to proceed with a revised version of the research on perceptual discrimination in simulated agents.

Noise and Uncertainty

The question of noise estimation, independent noise sources and reduction of uncertainty is one of the cornerstones of optimal multisensory integration research. Given that no system evolved to confirm the predictions from MLE, this question could not be direclty addressed. The first simulation confirmed that the introduction of different levels of Gaussian noise led to the expected deterioration of perceptual accuracy (cf. Fig. 2). It is arguable if adding Gaussian noise at any time step to a signal that is then fed into a rate code neural network is the most suitable approach for the evolution of systems whose behaviour is contingent on levels of noise. As a lot of the noise is filtered directly by the neurons, that have a minimal time constant of $\tau = 30ms$, such systems may have a hard time to develop sensitivity to levels of noise. In future models, noise may instead be added to a physical stimulus, which, at least in theory, would allow agents to use active strategies not just to perform perceptual discrimination, but also to perform noise estimation. Generally, it was a long shot to expect that optimal integration would evolve in evolved systems by merely adding the requirement to be accurate in perceptual discrimination. Even if the outlined technical and conceptual problems can be solved in future research, it may be necessary as a next step to explicitly require agents to integrate optimally in order to tackle this question.

Conclusion

The ambitious goal to evolve optimal multisensory integration in networks and agents has not been met in the current research. However, the difficulties encountered were informative about hidden prior assumptions on several levels: about ideal observer models (what is 'unimodal' in a bimodal system? Can noise in the silent channel explain an increase in bimodal perceptual accuracy?), about using a psychophysics task for evolution (does success in a 2AFC task equal perceptual capacity?) and about the role of action in perceptual discrimination (if active sensing is beneficial for perceptual discrimination, how does it figure in multisensory integration?). Rather than answering one question, the study generated more digestible sub-questions, which is characteristic of generative ER models. The outlined avenues for future research will be pursued to further elucidate

the relevant question of (optimal) multisensory integration from an embodied and Artificial Life point of view.

Acknowledgements

This work was supported by the HFSP Research Grant (2006) on Mechanisms of associative learning in human perception.

References

Alais, D. and Burr, D. (2004). The ventriloquist effect results from near-optimal bimodal integration. *Current Biology*, 14:257–262.

Beer, R. D. (2003). The dynamics of active categorical perception in an evolved model agent. *Adaptive Behavior*, 11:209–243.

Di Paolo, E., Rohde, M., and Iizuka, H. (2008). Sensitivity to social contingency or stability of interaction? Modelling the dynamics of perceptual crossing. *New Ideas in Psychology*, 26:278–294.

Ernst, M. O. (2005). A Bayesian view on multimodal cue integration. In Knoblich, G., Grosjean, M., Thornton, I., and Shiffrar, M., editors, *Human body perception from the inside out*, pages 105–131. Oxford University Press, New York.

Ernst, M. O. and Banks, M. S. (2002). Humans integrate visual and haptic information in a statistically optimal fashion. *Nature*, 415:429–433.

Floreano, D., Kato, T., Marocco, D., and Sauser, E. (2004). Coevolution of active vision and feature selection. *Biological Cybernetics*, 90:218–228.

Freeman, W. J. (1987). Simulation of chaotic EEG patterns with a dynamic model of the olfactory system. *Biological Cybernetics*, 56.

Harvey, I., Di Paolo, E., Wood, R., Quinn, M., and Tuci, E. A. (2005). Evolutionary Robotics: A new scientific tool for studying cognition. *Artificial Life*, 11(1-2):79–98.

Hill, J. (2005). psignifit toolbox for Matlab 5 and up. http://www.bootstrap-software.org/psignifit/ retrieved 05.02.2010. Version 2.5.6 for Mac OSX.

Izquierdo-Torres, E. and Di Paolo, E. (2005). Is an embodied system ever purely reactive? In *Proceedings of the 6th European Conference of Artificial Life ECAL 2005*, pages 252–261.

Knill, D. C. and Pouget, A. (2004). The Bayesian brain: the role of uncertainty in neural coding and computation. *Trends in Neurosciences*, 27(12):712 – 719.

O'Regan, K. and Noë, A. (2001). A sensorimotor account of vision and visual consciousness. *Behavioral and Brain Sciences*, 24:939–1011.

Selection Pressures for a Theory-of-Mind Faculty in Artificial Agents

J. Noble, T. Hebbron, J. van der Horst, R. Mills, S. T. Powers, and R. A. Watson

Science and Engineering of Natural Systems group
School of Electronics and Computer Science
University of Southampton, UK
jn2@ecs.soton.ac.uk

Extended Abstract

To have a theory of mind (ToM) is to anticipate the behaviour of other agents by considering what they want and what they know. It requires a representation of the environment that includes the internal states (e.g., beliefs) of other agents. Adult humans generally possess a ToM ability, demonstrated by reasoning like "he did not see the chocolate being switched from the red box to the blue one, so I predict he will choose the red box." Note the distinction between what the speaker believes to be true, and what the speaker believes about the other agent's belief states. ToM is of interest in developmental psychology (when and how do children acquire it?) and primatology (do our near relatives possess it?).

In this project we ask: in an evolving population of social agents, under what circumstances would a ToM ability be selected for? Using simulation to identify the ecological niches that produce selection pressure for ToM should cast light on its origin in humans and on when we should expect to see it in other animals. We build on earlier work by Takano and Arita (2006).

To operationalize ToM we borrow a hierarchy of cognitive architectures from Dennett (1987). A zero-order intentional agent (often seen in ALife work) is purely reactive to its perceptual inputs. A first-order agent builds on this by including internal state that has a mapping relation with the environment, e.g., remembering where a predator was last seen. A second-order agent has basic ToM, i.e., it is equipped with a world-model that includes the internal states of other agents (e.g., "there's a predator behind that tree, but my friend hasn't seen it yet."). Third- and higher-order agents include a recursive aspect, i.e., a model of what I think he thinks I am thinking.

Low-order agents are logically prior, but the evolution of higher-order agents like ourselves is not inevitable. ALife and related work (Braitenberg, 1984) have shown that outwardly sophisticated behaviours can be produced by simple underlying mechanisms. The evolutionarily stable strategy will sometimes remain zero- or first-order and this will depend on aspects of the ecological niche, such as the nature of the payoff matrix for agent interactions and the degree of perceptual overlap between agents. We tested these ideas in simulation by constructing a range of different social environments and running invasion studies, in which a population of (n)-order agents is exposed to an infrequent (n+1)-order mutant. If the higher-order mutant is fitter and thus able to invade, this indicates selection pressure for more advanced ToM abilities.

Results confirm that fragmented perception (not all agents see the same things) and socially relevant payoff matrices (my payoff depends on both our actions) are necessary for ToM to evolve. More specifically, competitive rather than cooperative interactions produce greater selection pressure for ToM. This finding is a challenge for the common association between ToM and human language (Grice, 1969) as the latter requires a cooperative context. Something about the early human ecological niche must have combined cooperative and competitive contexts in a near-unique way.

References

Braitenberg, V. (1984). *Vehicles: Experiments in Synthetic Psychology*. MIT Press, Cambridge, MA.

Dennett, D. C. (1987). *The Intentional Stance*. MIT Press / Bradford Books, Cambridge, MA.

Grice, H. P. (1969). Utterer's meaning and intention. *Philosophical Review*, 68:147–177.

Takano, R. and Arita, T. (2006). Asymmetry between even and odd levels of recursion in a theory of mind. In Rocha, L. M., Yaeger, L. S., Bedau, M. A., Floreano, D., Goldstone, R. L., and Vespignani, A., editors, *Artificial Life X: Proceedings of the Tenth International Conference on Artificial Life*, pages 405–411. MIT Press, Cambridge, MA.

A Conscious-based Mind for an Artificial Creature

Ricardo Capitanio Martins da Silva[1], Ricardo Ribeiro Gudwin[2]
[1]DCA-FEEC-UNICAMP martins@dca.fee.unicamp.br
[2]DCA-FEEC-UNICAMP gudwin@dca.fee.unicamp.br

Abstract

This work describes the application of the Baars-Franklin Architecture (BFA), an artificial consciousness approach, to synthesize a mind (a control system) for an artificial creature. The BFA was reported in the literature as a successful control system to different kinds of agents: CMattie, IDA and CTS. In this paper, BFA is for the first time applied for controlling an artificial (virtual) creature. Firstly we introduce the theoretical foundations of this approach for the development of a conscious agent. Then we explain the architecture of our agent and at the end we discuss the results and first impressions of this approach.

Keywords: artificial consciousness, intelligent systems, autonomous vehicle, multi-agent systems

Introduction

In the last ten years there has been an intensive growth in the scientific study of consciousness (Atkinson et al., 2000; Blackmore, 2005). A technological offspring of these studies is the field of artificial consciousness (Aleksander, 2007; Bogner, 1999; Cardon, 2006; Chella and Manzotti, 2007; Gamez, 2008). In this work we concentrate in what we call here the Baars-Franklin architecture (BFA). The BFA is a computational architecture being developed by the group of Stan Franklin, at the University of Memphis (Franklin and Graesser, 1999; Bogner, 1999; Negatu and Franklin, 2002; Negatu, 2006), based on the model of consciousness given by Bernard Baars, called Global Workspace Theory (Baars, 1988).

The BFA has already been applied to many different kinds of software agents. The first application of BFA was CMattie (Franklin and Graesser, 1999; Bogner, 1999), an agent developed by the Cognitive Computing Research Group (CCRG) at the University of Memphis, whose main activities were to gather seminar information via email from humans, compose an announcement of the next week's seminars, and mail it to members of a mailing list. Through the interaction with human seminar organizers, CMattie could realize that there was missing information and ask it via email.

The overall BFA received major improvements with subsequent developments. One remarkable implementation of it was IDA (Intelligent Distribution Agent) (Franklin, 2005), an application developed for the US Navy to automate an entire set of tasks of human personnel agent who assigns sailors to new tours of duty. IDA is supposed to communicate with sailors via email and, in natural language, understand the content and produce life-like messages.

The BFA was also used outside of Franklin's group. Daniel Dubois from University of Quebec developed CTS (Conscious Tutoring System) (Dubois, 2007), a BFA-based autonomous agent to support the training on the manipulation of the International Space Station robotic control system called Canadarm2.

Nevertheless, up to our knowledge, BFA was never used to implement a mind (a control system) for an artificial virtual creature. Artificial Creatures are a special kind of agents, embodied autonomous agents which exists in a certain environment, moving itself in this environment and acting on it (Balkenius, 1995). Artificial creatures may be real or virtual. Examples of real artificial creatures are robots acting in the real environment. Virtual Artificial Creatures are software agents living in a virtual world, where they are able to sense and actuate by means of an avatar (a virtual body). One example of a virtual artificial creature is an intelligent opponent in a computer game, where an intelligent control system must decide the actions to be performed by the agent in order to foster a good entertainment to the system user, simulating with realism the behavior of a human opponent. Other examples of virtual artificial creatures include ethological simulation studies, in artificial life, where tasks such as foraging and sheltering are very common.

Virtual artificial creatures pose some interesting research problems when compared to other kinds of software agents where BFA has already been tested. In the original applications where BFA was tested, the perception system is based on the exchange of e-mail messages (the case of CMattie and IDA), and interactions in a HCI (human-computer interface), in the case of CTS. In a virtual artificial creature, perception must rely on remote (e.g. visual, sonar, etc)

and/or local (e.g. contact) sensors, capturing properties of the scenario and interpreting them in order to create a world model. The behavior generation module is also different, as the agent must act on itself (its body) and over things on the environment. The main motivation for the research reported in this work is though to investigate how the use of BFA may impact the control of a virtual artificial creature, and what are the benefits which can be expected.

In the next section, we introduce briefly Baars' theory of consciousness, Global Workspace Theory, and then we describe how we customized BFA in order to deal with virtual artificial agents. After that, we introduce CAV (Conscious Autonomous Vehicle), the artificial creature we used in our study and its environment, and a brief analysis of the results of our simulations using CAV.

Global Workspace Theory and BFA

Bernard Baars has developed the Global Workspace Theory (GWT) (Baars, 1988, 1997) inspired by psychology and based on empirical tests from cognitive and neural sciences. GWT is an unifying theory that puts together many previous hypothesis about the human mind and human consciousness.

Baars postulates that processes such as attention, action selection, automation, learning, meta-cognition, emotion, and most cognitive operations are carried out by a multitude of globally distributed unconscious specialized processors. Each processor is autonomous, efficient, and works in parallel and high speed. Nevertheless, in order to do its processing, each processor may need a set of resources (mostly information of a specific kind), and at the same time, will generate another set of resources after its processing. Specialized processors can cooperate to each other forming coalitions. This cooperation is by means of supplying to each other, the kinds of resources necessary for their processing. They exchange resources by writing in and reading from specific places in working memory. Coalitions may form large complex networks, where processors are able to exchange information to each other. But processors within a coalition do have only local information. There may be situations, where the required information is not available within the coalition. To deal with these situations, and allow global communication among all the processors, there is a global workspace, where processors are able to broadcast their requirements to all other processors. Likewise, there may be situations where some processor would like to advertise the resource it generates, as there may be other processors interested in them. They will also be interested in accessing the global workspace and broadcasting to all other processors. In the broadcast dynamics, only one coalition is allowed to be within the global workspace in a given instance of time. In order to decide which coalition will go to the global workspace in a given instant of time, a whole competition process is triggered. Each processor has an activation level, which expresses its urgency in getting some information or the importance of the information it generates. A coalition will also have an activation level which is the average of activation levels of its participants. At each time instant, the coalition with the highest activation level will win the access to the global workspace. Once a coalition is within the global workspace, all its processors will broadcast their requests and the information they generate. The broadcast mechanism do allow the formation of new coalitions, and also some change in working coalitions.

For Baars, consciousness is related to the working of this global workspace. Processors are usually unconscious, having access only to local information, but in some cases they may require or provide global information, in which case they request access to consciousness, where they will be able to broadcast to all other processors. This is the case when they have unusual, urgent, or particularly relevant information or demands. This mechanism supports integration among many independent functions of the brain and unconscious collections of knowledge. In this way, consciousness plays an integrative and mobilizing role. Moreover, consciousness can be useful too when automatized (unconscious) tasks are not being able to deal with some particular situation (e.g. they are not working as expected), and so a special problem solving is required. Executive coalitions, specialized in problem solving will be recruited then in order to deal with these special situations, delegating trivial problems to other unconscious coalitions. In this way, consciousness works like a filter, receiving only emergency or specially relevant information.

Inspired by Baars description of his theory of consciousness, and also by previous work in the computer science literature, Franklin proposed a framework for a software agent which realized Baars theory of consciousness, in terms of a computational architecture, constituting so what we are calling here the Baars-Franklin architecture. In specifying BFA, Franklin used the following theories as background, among others not detailed here: Selfridge's Pandemonium (Selfridge, 1958) and Jackson's extension to it (Jackson, 1987), Hofstadter and Mitchell Copycat (Hofstadter and Mitchell, 1994) and Maes' Behavior Network (Maes, 1989).

From Hofstadter's Copycat, Franklin borrowed the notion of a "Codelet" (and also the Slipnet, for perception). He noticed that these *codelets* were more or less the same thing as Selfridge's "demons" in Pandemonium theory and also a good computational version for Baars *processors*. Jackson's description of an arena of *demons* competing for selection will fit as well Baars description of processors competing in a *Playing Field* for access to consciousness. Using these similarities, Franklin set up the basis of BFA: cognitive functions are performed by coalitions of codelets working together unconsciously, reading and writing tagged information to a Working Memory. Each codelet has an activity level and a tagged information. A special mechanism, the *Coalition Manager* will manage coalitions and

calculate the activity level of each coalition. Another special mechanism, the *Spotlight Controller*, will be evaluating each coalition activity level, and defining the winning coalition. Also, the *Spotlight Controller* will be responsible for performing the broadcast of the tagged information of each codelet in the winning coalition, to all codelets in the system. The agent behavior is decided using a Behavior Network, whose propositions are related to the tagged information in the Working Memory.

Unfortunately, a full description of BFA is beyond the space available in this text. We refer the interested reader to (Bogner, 1999; Negatu, 2006; Dubois, 2007; da Silva, 2009), where a more detailed description of BFA is available. Some background in the auxiliary theories we mentioned above is provided next.

Pandemonium Theory

Selfridge's Pandemonium Theory is a connectionist architecture originally used for pattern recognition. Selfridge (Selfridge, 1958), influenced by the parallelism of human data processing, suggested a parallel architecture composed of multiple independent processes called *demons*. Each demon works simultaneously recognizing specific conditions (or a set of them). Demons have links that allows them to "call" other demons.

John Jackson extended the original Pandemonium theory of perception by creating the stadium metaphor, organizing demons in two different locations, the equivalent of stands and arena of a stadium. Jackson (Jackson, 1987) proposed a system consisted of a crowd of usually dormant demons located at the stands, from where a few demons could go down to the arena and start exciting the crowd. Some demons in the crowd gets more excited and starts to yell louder. If the activity of demons in the arena drops below a threshold they may return to the stands and the loudest demons in the crowd replace them. Besides the crowd getting excited watching the demons in the arena, the last ones can spread activation to the former through links. These connections between demons are created or strengthened according to the time they are together on the arena, following a Hebbian learning scheme.

Copycat Architecture

Copycat is a hybrid symbolic-connnectionist architecture that is intended to model analogy making along with recognition and categorization. It was developed by Hofstadter and Mitchell (Hofstadter and Mitchell, 1994) with the premise that analogy making is a process of high-level perception. Copycat makes and interprets analogies between situations in a predefined and fixed domain like letter-string analogy problems.

Those analogies emerge from the activity of many independent processes, called *codelets*, running in parallel, sometimes cooperating, sometimes competing with each other. Copycat starts with a fixed number of codelets in a codehack, predetermined by the designer.

Codelets count with an associative network (the *Slipnet*) that contains interrelated concept types (*nodes*) and links between them. Codelets look for specific words or parts of words and if they find them they activate some nodes of the Slipnet. Nodes can vary in their level of activation which is a measure of relevance to the current situation. They spread some activation to neighbors and lose activation by decay. The Slipnet is a long-term memory and represents what Copycat knows. It does not learn anything during execution.

Finally, Copycat has a working memory where perceptual structures are built and modified. At each moment the content of the working memory represents Copycat's current perception of the situation it is facing.

Behavior Network

Pattie Maes (Maes, 1989) developed a behavior-based action selection mechanism, built as a society of behaviors or competence modules in a distributed, recurrent, non-hierarchical network. This network is formed by four kinds of nodes. The first kind of node (and the most important) represents a low level behavior (e. g. approach food, drink water, walk around). The second kind of node represents propositions (or predicates e.g. glass-on-hand, glass-with-water-inside, glass-empty), which can be true or false. The third kind of node represents goals (or motivations). The fourth kind of node represents sensors from the environment.

Sensor nodes are linked to proposition nodes. Behavior nodes are input linked from *preconditions* propositions which must be true for the behavior to be executable. In its output, they are linked to two possible kinds of propositions: *add* propositions, which are expected to become true after the behavior is executed, and *delete* propositions, which should be set to false after the behavior is executed. For example, a behavior *"drink water"* could have the preconditions *glass-on-hand* and *glass-with-water-inside*. Its add list could contain *glass-empty* and the delete list would contain *glass-with-water-inside*. Goal nodes are linked to proposition nodes, which are backward linked to behavior nodes. See figures 3 and 4, further, for an example of the connection among links. In these figures, triangles are proposition nodes, ovals are behavior nodes, round squares are sensor nodes and pentagons are goal nodes.

The network executes as follows. Each behavior has an activation level, which is changed by two waves of spreading activation: one from sensor nodes forward and the other from goal nodes backwards. The first one spreads activation forward from sensor nodes to propositions which are evaluated (true or false) according to the environmental situation and from them forwards to behavior nodes which need these predicates to be true to be fired. The second spreads activation backwards from goal nodes to predicate nodes and

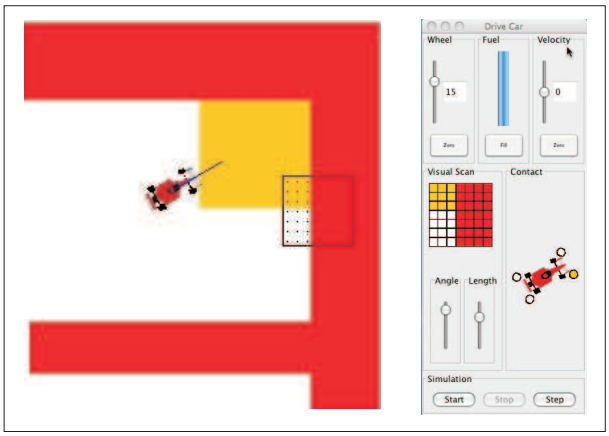


Figure 1: Sensory-motor structure of the creature

then to behaviors which can satisfy these goals. More details on the spreading mechanism can be found in (Maes, 1989; Negatu, 2006). At the end, after all the energy is spread-up, the behavior which remains with the highest activation level is chosen to be executed. Only one behavior is chosen to be executed at each operational cycle.

Our implementation of BFA

In our experiment, we developed an artificial mind (a control system), which we call CAV - *Conscious Autonomous Vehicle*, to control an artificial creature in a virtual environment (see figure 1). The creature and its environment were originally presented in (Gudwin, 1996) (where more details on its characteristics can be obtained) and were adapted for our current studies. In this environment, the creature is equipped with sensors and actuators, which enable it to navigate through an environment full of objects with different characteristics. An object can vary in its "color" and each color is linked to: a measure of "hardness" which is used in the dynamic model as a friction coefficient that can slow down the creature's movement (or completely block it), a "taste" which can be bad or good, and a feature related with "energy" which indicates that the object drains/supplies energy from/to the creature's internal rechargeable battery.

The creature connects to its mind through sockets. In this sense, the artificial mind is a completely separate process, which can be run even in a different machine. So, different minds can be attached to the creature and tested for the exact same situation.

When the simulation is started, the creature builds an incremental map of the environment based on the sensory information. Our agent adds landmarks to this map and uses them to generate movement plans. It has two main motivations: it should navigate from an initial point up to a target point, avoiding collisions with objects; and it should keep its energetic balance, taking care of the energy level in the internal batteries.

Our architecture (see figure 2) is essentially rooted in the

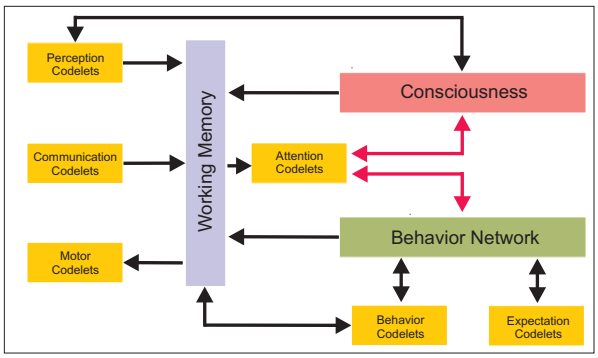


Figure 2: *CAV*'s Architecture

BFA implementation as in (Bogner, 1999) (consciousness) and (Negatu, 2006) (behavior network). *CAV* brings some modifications in the implementation related with the application domain, and the interaction among consciousness and behavior network. The following sections contain a brief description of CAV's modules.

Codelets

CAV is heavily dependent on small pieces of code running as separate threads called codelets (BFA borrows this name from Hofstadter's Copycat). Those codelets correspond pretty well to the specialized processors of global workspace theory or demons of Jackson and Selfridge.

BFA prescribes different kinds of codelets such as attention codelets, information codelets, perceptual codelets and behavior codelets. In addition to that, it is possible to create new types of codelets depending on the problem domain. CAV's domain does not require string processing as do most other BFA applications. Instead of that, the creature state is well divided in registers at the working memory. It is possible to have access to all variables anytime. Because of this, CAV does not use information codelets which in BFA are used to represent and transfer information. We have two kinds of behavioral codelets: the behavior codelets, linked with the nodes of the Behavior Network and responsible for "what to do", and motor codelets, which know "how to act" on the environment. With this in mind CAV has the taxonomy of codelets presented at Table 1.

Working Memory

The working memory consists of a set of registers which are responsible for keeping temporary information. The major part of the working memory is related to the creature status. The communication codelet constantly overwrites the registers like speed, wheel degree, sensory information and creature position. CAV's working memory works also as an interface among modules, for example, between consciousness and the behavior network. Some codelets, including attention codelets watch what is written in the working memory in order to find relevant, insistent or urgent situations.

Table 1: *CAV's Codelets Taxonomy*

Type	Role
Communication	Perform the communication with the simulator, bringing novel simulation information
Perception	Give an interpretation to what the agent senses from its environment
Attention	Monitor the working memory for relevant situations and bias information selection
Expectation	Check that expected results do happen
Behavior	Alter the parameter of the motor codelet
Motor	Act on the environment

When they find something, they react in order to compete for consciousness. Whenever one of then reaches consciousness, its information will influence the agent's actions.

Consciousness mechanism

The consciousness mechanism consists of a *Coalition Manager*, a *Spotlight Controller*, a *Broadcast Manager* and attention codelets which are responsible for bringing appropriate contents to "consciousness" (Bogner, 1999). In most of the cases, codelets are observing the working memory, looking for some relevant external situation (e.g. a low level of energy). But some codelets keep a watchful eye on the state of the behavior network for some particular occurrence, like having no plan to reach a target. More than one attention codelet can be excited due to a certain situation, causing a competition for the spotlight of consciousness. If a codelet is the winner of this competition, its content is then broadcast to the registered codelets in the broadcast manager. We have three main differences between standard BFA and CAV, related to this module. The first one is that we don't use information codelets. The second is that not all of the codelets are notified like in BFA, just the registered ones. Finally, some codelets can be active outside of the playing field. In this case their contents will never reach consciousness.

Behavior Network

CAV's behavior network is based on a version of Maes' architecture (Maes, 1989) modified by Negatu (Negatu, 2006). Negatu adapted Maes' behavior network so each behavior is performed by a collection of codelets. Negatu's implementation also divided the behavior network in *streams* of behavior nodes.

The behavior network works like a long-term procedural memory, a decision structure and a planning mechanism. It coordinates the behavior actions through an "unconscious" decision-making process. Even so it relies on conscious broadcasts to keep up-to-date about the current situation. This is called "consciously mediated action selection" (Negatu, 2006).

CAV uses two main behavioral streams, the *Target* stream and the *Energy* stream, as in figures 3 and 4.

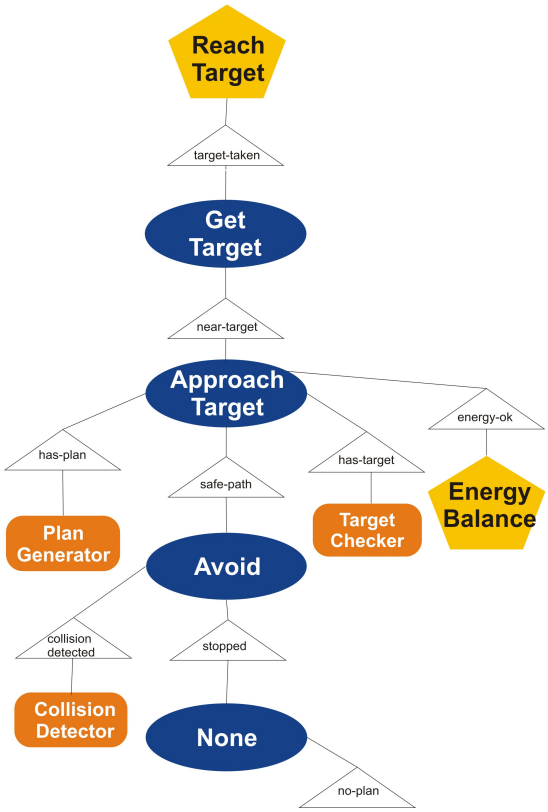


Figure 3: Behavior Network - Target Stream

Cognitive Cycle

In GWT, all codelets and the consciousness mechanism are asynchronous and parallel processes. In the first implementations of BFA, these were all implemented by completely asynchronous threads. Nevertheless, due to many synchronism problems among codelets, further implementations of BFA prescribed the creation of a *Cognitive Cycle*. This cycle imposes some synchronism points on codelets threads, and organizes the interaction among BFA's components in the form of an operational cycle. This solved synchronism issues of the multi-thread environment and made less difficult the computational implementation without detriment of the main ideas in GWT.

CAV's cognitive cycle (CCC) brings significant differ-

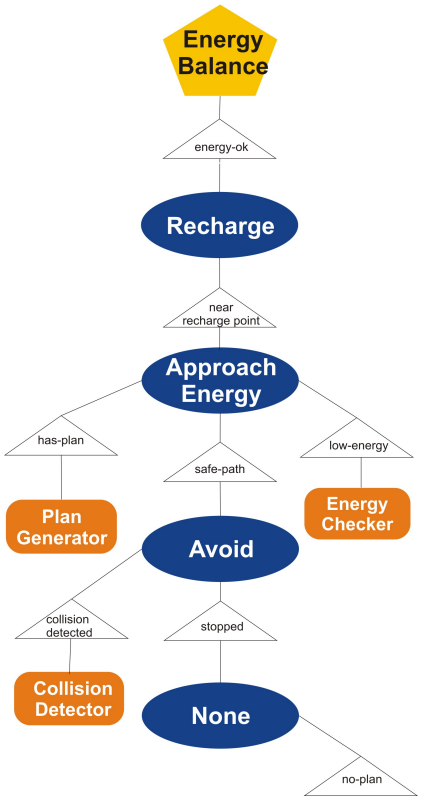


Figure 4: Behavior Network - Energy Stream

ences when compared to standard BFA's one. For a detailed account on how CCC is modified compared to the standard BFA cycle, see (da Silva, 2009).

For the standard BFA's cognitive cycle see (Baars and Franklin, 2003).

We removed the first three original steps: perception (interpretation of sensory stimuli), percept to preconscious buffer (the percept is stored in working memory), local associations (retrieve local associations from transient episodic memory (TEM) and long term associative memory (LTM)). This last one is quite obvious as CAV does not have an implementation of TEM or LTM. In the other cases, the removal of the two first steps is related to the problem domain. CAV does not process streams of characters like IDA. So CAV does not need a Slipnet. Moreover, the input data of CAV is well structured, as working memory's registers can be updated anytime. It guarantees that all codelets will handle the most possible up-to-date input data. The "recruitment of resources" step has also been removed, because the "answer" of all listening codelets happens in parallel with the cycle, not inside it.

The remaining CCC five steps are summarized below (adapted from (Baars and Franklin, 2003). We will indicate major accordances with standard BFA with sentences written in italics):

Competition for consciousness *Attention codelets, whose job is to bring relevant, urgent, or insistent events to consciousness, access working memory and the behavior network state. Some of them gather information and actively compete for access to consciousness. The competition may also include attention codelets from recent previous cycle.*

Conscious broadcast *A coalition of codelets (possibly with just a single codelet) gains access to the global workspace and has its contents broadcasted. This broadcast is hypothesized to correspond to phenomenal consciousness.* Not all CAV's codelets are registered at the Broadcast Manager (e.g. the behavior codelets). So the information between Behavior Network and consciousness pass through attention codelets when those codelets gain consciousness access (see figure 2). In doing so, the propositions added to the behavior network state by behavior codelets can be known by all registered codelets.

Setting goal context hierarchy At this stage CAV updates all the new propositions which were added since the last cycle and incorporates new and more accurate information to the behavior network. The goals are checked and updated. It is also possible to add or remove a goal following the current situation.

Action chosen *The behavior net chooses a single behavior. This choice is heavily affected by the update of the past stage. It is also affected by the current situation, external and internal conditions, by the relationship among behaviors and by the residual activation values of various behaviors.*

Action taken *The execution of a behavior results in the behavior codelets performing their specialized tasks, which may have external or internal consequences. The acting codelets also include an expectation codelet whose task is to monitor the action and bring to consciousness any failure in the expected results.* CCC does not wait for the running end of a behavior codelet. CAV keeps a list of active behavior codelets and, if some particular codelet is already running, it does not start another instance of it. But it can abort a running behavior codelet, if it is necessary. For example, if a new perception makes a plan unfeasible, during the execution of a behavior codelet (let's say the vehicle is going from a point A to a point B and a new obstacle is detected), then the behavior codelet is aborted, as a new plan must be generated.

A Brief Analysis of CAV's implementation

A running simulation of CAV's performance is illustrated in figure 5. The main experiment worked as expected. The creature was able to pursue its main objectives: to avoid collision with obstacles while exploring the environment, and

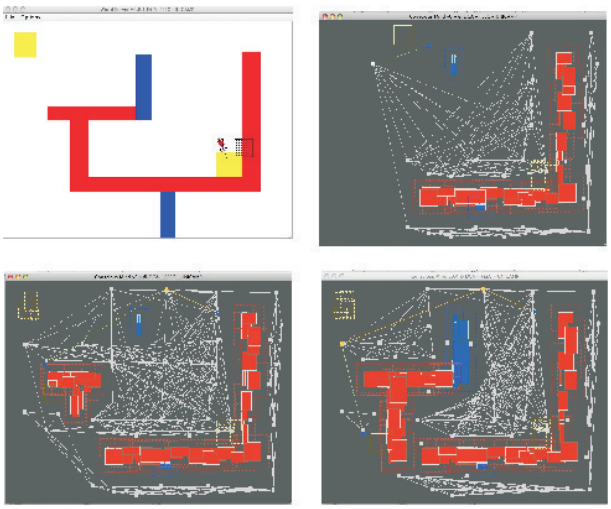


Figure 5: Example of Simulation

at the same time maintaining an energy balance. While exploring the environment, if the energy level decreased to a critic limit, CAV correctly postponed its exploratory behavior, looked for the closest source of energy and traced a route to it to feed itself. After refreshing its batteries, it returned to its exploratory behavior. As we said before, though, our main goal was not simply related to the achievement of these tasks (something which could be achieved by more traditional methods, as e.g. in (Gudwin, 1996)), but understanding how "consciousness" could be used in such an application.

By applying BFA to this application, we would like to evaluate the value of "consciousness" (as in BFA) to the construction of a new generation of cognitive architectures to control artificial creatures. Pragmatically, we would like to understand what exactly it is this "consciousness" technology, and what the benefits to expect while applying it as a mind to an artificial creature. This goal was also achieved while we had the experience of studying BFA and applying it to the current application. Our findings are summarized in the next subsections.

A Qualitative Analysis

Two important findings of our investigation are the qualitative understanding of what is "consciousness" (in BFA) and an abstraction of what may be its main benefits as a technology. The philosopher Daniel Dennet has already stated that: "Human consciousness (...) can best be understood as the operation of a "*Von Neumannesque*" virtual machine *implemented* in the parallel architecture of a brain". Even though Baars and Franklin do not explicitly point this out, this is what BFA provides. It implements a (virtual) serial machine on top of a parallel machine. The overall structure of codelets reading and writing on the *Working Memory* configures a fully parallel multi-agent system. The constraints of the *SpotlightController* and the broadcast mechanism implements on top of it the emergence of a serial stream which is the consciousness. But this serial stream is not just any serial stream. It focuses attention on the most important kind of information in each time step. It builds what Koch called an *executive summary* of information (Koch, 2004). This is one of the main advantages of this technology: to focus attention on what is most important and spreading this to all agents in the multi-agent system. Now, this interplay between a serial and parallel components opens a large set of opportunities to future research. Among other things, we envision the opportunity of new learning schemes (using the broadcast to form new connections among codelets) and many other enhancements.

A Quantitative Analysis

Some data related to the experiment can be viewed in figures 6, 7 and 8. Figure 6 shows the number of active threads at each instant of time. We can see that an average of 8 threads are working at the same time. Figure 7 shows the number of codelets running at the same time at the playing field. An average of 1 or 2 codelets were at the playing field at the same time. The maximum of codelets at the playing field at the same time was 3. Finally, figure 8 shows the different types of codelets accessing the consciousness at each time. We can see that most of the time the codelet *ObstacleRecorder* was at consciousness. The second more frequent was *PlanGenerator*. The other three, *TargetCarrier*, *Collision-*

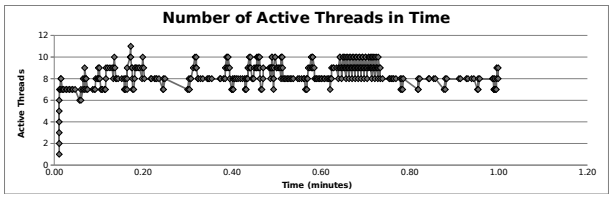


Figure 6: Number of Active Threads in Time

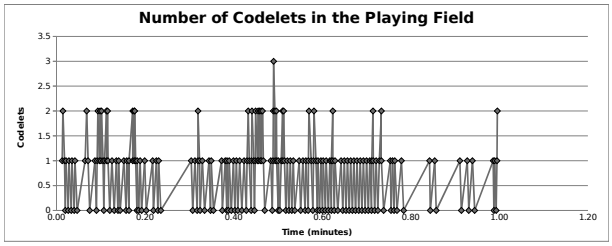
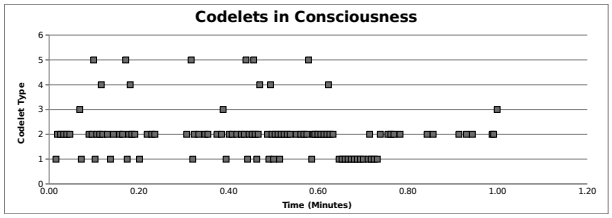


Figure 7: Number of Codelets in the Playing Field



1 - PlanGenerator 2 - ObstacleRecorder 3 - TargetCarrier
4 - CollisionDetector 5 - PathChecker

Figure 8: Types of Codelets in Consciousness

Detector and *PathChecker* were less frequently at the consciousness.

These data refer to 1 minute of simulation. The subsequent instants of time show more or less the same behavior. Other codelets, like e.g. *LowEnergy*, also appear from time to time, but they didn't appear in the time-frame shown in the figure.

Conclusion

BFA is shown to be a very flexible and scalable architecture, due to its consciousness and behavior network mechanisms implemented through independent codelets. Newer features can be easily included by means of newer codelets performing new roles. Consciousness mechanism makes possible a deliberation process that enables the perception of most relevant information for the current situation, building what Koch called an executive summary of perception. Much work remains to be done, especially related to a better model formalization and a better understanding of the overall role of coalitions. However, seen as an embryo of a conscious artificial creature, the first results of this study show the feasibility of such techniques, motivating our group to continue on this line of investigation.

Acknowledgments. R.G. acknowledges FAPESP for funding support.

References

Aleksander, I. (2007). Modeling consciousness in virtual computational machines. *Synthesis Philosophica*, 44(2):447–454.

Atkinson, A. P., Thomas, M. S. C., and Cleeremans, A. (2000). Consciousness: mapping the theoretical landscape. *Trends in Cognitive Sciences*, 4(10):372–382.

Baars, B. J. (1988). *A cognitive theory of consciousness*. Cambridge University Press.

Baars, B. J. (1997). *In the Theater of Consciousness: The Workspace of the Mind*. Oxford University Press.

Baars, B. J. and Franklin, S. (2003). How conscious experience and working memory interact. *Trends in Cognitive Sciences*, 7(4):166–172.

Balkenius, C. (1995). *Natural Intelligence in Artificial Creatures*. Lund Univ. Cognitive Studies 37.

Blackmore, S. (2005). *Consciousness - A very short introduction*. Oxford University Press.

Bogner, M. B. (1999). *Realizing "Consciousness" in Software Agents*. PhD thesis, The University of Memphis.

Cardon, A. (2006). Artificial consciousness, artificial emotions, and autonomous robots. *Cognition Process*, 7:245–267.

Chella, A. and Manzotti, R. (2007). *Artificial Consciousness*. Imprint Academic.

da Silva, R. C. M. (2009). Análise da Arquitetura Baars-Franklin de Consciência Artificial Aplicada a uma Criatura Virtual. Master's thesis, DCA-FEEC-UNICAMP.

Dubois, D. (2007). *Constructing an agent equipped with an artificial consciousness: application to an intelligent tutoring system*. PhD thesis, Université du Québec à Montréal.

Franklin, S. (2005). A "consciousness" based architecture for a functioning mind. In Davis, D. N., editor, *Visions of Mind: Architecture for Cognition and Affect*, chapter 8, pages 149–175. Idea Group Inc (IGI).

Franklin, S. and Graesser, A. (1999). A software agent model of consciousness. *Consciousness and Cognition*, 8:285–301.

Gamez, D. (2008). Progress in machine consciousness. *Consciousness and Cognition*, 17:887–910.

Gudwin, R. R. (1996). *Contribuições ao Estudo Matemático de Sistemas Inteligentes*. PhD thesis, Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas.

Hofstadter, D. R. and Mitchell, M. (1994). The copycat project: A model of mental fluidity and analogy-making. *In Holyoak, K.J & Barnden, J.A. (Eds.). Advances in connectionist and neural computation theory*, 2:31–112.

Jackson, J. V. (1987). Idea for a mind. *ACM SIGART Bulletin*, xx(101):23–26.

Koch, C. (2004). *The Quest for Consciousness - A Neurobiological Approach*. Roberts & Company Publishers.

Maes, P. (1989). How to do the right thing. *Connection Science Journal*, 1:3.

Negatu, A. S. (2006). *Cognitively Inspired Decision Making for Software Agents: Integrated Mechanisms for Action Selection, Expectation, Automatization and Non-Routine Problem Solving*. PhD thesis, The University of Memphis.

Negatu, A. S. and Franklin, S. (2002). An action selection mechanism for "conscious" software agents. *Cognitive Science Quarterly*, 2:363–386.

Selfridge, O. G. (1958). Pandemonium: a paradigm for learning. In *Mechanism of Thought Processes: Proceedings of a Symposium Held at the National Physical Laboratory*, pages 513–526, London: HMSO.

Self-Organization of Subjective Time and Sustainable Autonomy in Mind Time Machine

Takashi Ikegami and Yuta Ogai

Department of General Systems Studies,
The Graduate School of Arts and Sciences,
the University of Tokyo, 3-8-1 Komaba. Tokyo 153-8902
ikeg@sacral.c.u-tokyo.ac.jp, yuta@sacral.c.u-tokyo.ac.jp

Extended Abstract

It is time to bring artificial life in silico into the real world. Different from artificial or simulated environments, the real world presents many unexpected and complex encounters; and living systems essentially adapt to the real world's complexities. Any agent must deal simultaneously with various kinds of sensory flows while sustaining its own identity and autonomy. In this paper we introduce our recent project of making a special machine that self-organizes its own "subjective" timescape in an open environment.

We made a machine called MTM (Mind Time Machine), which runs in the real world all day long without losing its complex dynamics. As the result of this longtime sustainability, we argue that the system's own temporal structure is organized.

We presented this MTM for the first time at the Yamaguchi Center for Arts and Media in March, 2010. The machine consists of three screens: right, left and above, displayed at the corner of a cubic skeleton 5.400 meters per side. Fifteen cameras attached to each pole of the skeleton photograph things that happen in the venue. These images are decomposed into frames and chaotic neural dynamics control other macro processes that combine, reverse and superpose them to make new frames. We presented the MTM as artwork, but at the same time we recorded data from the system daily to monitor the diversity of the system's behavior.

The operating principle is to process timeframes of the visual inputs by combining chaotic instabilities from neural dynamics and optical feedback, in order to make autonomous "time-organizing" phenomena. Intake images from cameras were progressively embedded into the network's connections as a memory of the patterns. Visual images are taken in and re-played again and again with recursive modifications. The system itself is completely deterministic and uses no random numbers, but it shows different images depending on its inherent instabilities, environmental lighting conditions, movement of people coming into the venue and the system's stored memory.

This is not a large chaotic dynamical system that updates the visual inputs randomly. Different from the mere chaotic system, MTM is designed as life-like system since its dynamics are controlled by an environment and system has a short and long term memory to sustain its dynamics. Namely, we claim that MTM is "artificial life", since we design it to i) retrieve information from its environment, ii) memorize it in the form of the Hopfield type learning which tunes the parameters of the overall dynamics, iii) generate "episodic memory" , vi) change the network structure by the way of the Hebbian dynamics continuously and v) organize its overall dynamics as adaptation to the environmental changes.

At the conference, we will report how MTM's daily dynamics are varied by weather conditiosn and argue how it is difficult to sustain its autonomy, i.e. both sensitivity to the environment and inherent dynamics, for long periods of time.

References

Ikegami, T., Simulating Active Perception and Mental Imagery with Embodied Chaotic Itinerancy, Journal of Consciousness Studies Vol.14 (2007) pp.111-125.

Iizuka, H. and T.Ikegami, Simulated autonomous coupling in discrimination of light frequencies, Connection Science, 17 (2004) pp.283-299.

Ikegami, T., Morimoto, G. Chaotic Itinerancy in Coupled Dynamical Recognizers, Chaos 13 (2003) 1133-1147.

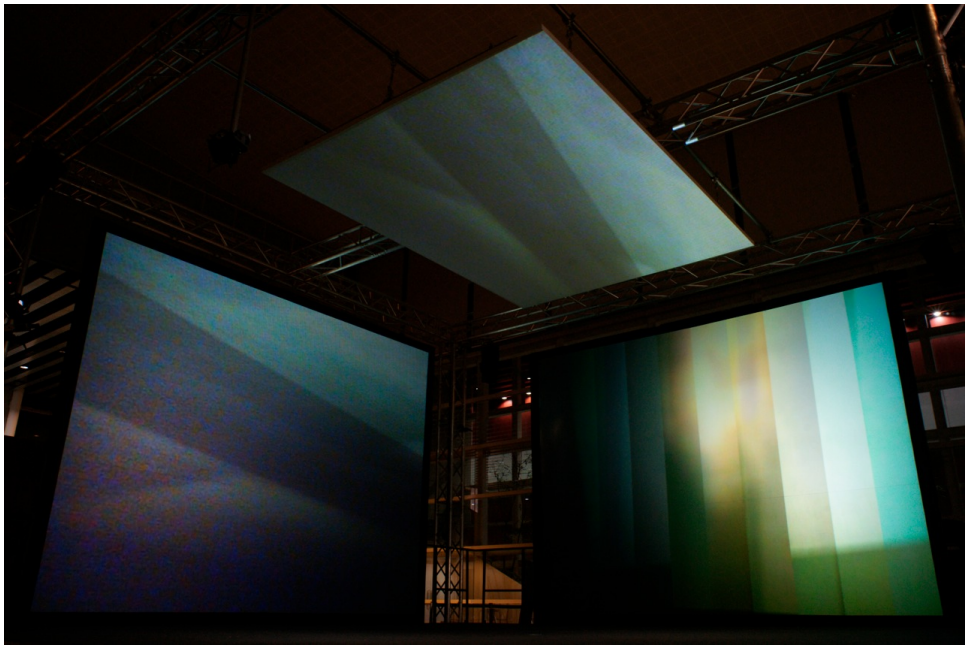


Figure 1: Outlook of MTM displayed at Yamaguchi Center for Art and Media, 2010. (Photo taken by Kenshu Shintsubo)

Bee Nest Site Selection as an Optimization Process

Konrad Diwold[1], Madeleine Beekman[2] and Martin Middendorf[1]

[1]Department of Computer Science, Universität Leipzig, D-04103 Leipzig, Germany
[2]School of Biological Sciences and Centre for Mathematical Biology, The University of Sydney, NSW 2006, Australia
kdiwold@informatik.uni-leipzig.de

Abstract

In recent years several bee inspired optimization techniques have been proposed. These methods are either based on the bees' foraging or mating behavior. Both foraging and mating regulate distributions outside (foraging) or within a colony (mating). Foraging determines the ratio of individuals that explore the surroundings for new food sources and those that exploit known food sources, while mating determines the distribution of genotypes within a colony. In contrast, nest-site selection is a processes that constitutes a decision-making process and enables a colony to identify and converge towards one best solution. We therefore propose to use the bees' nest-site selection behavior as the basis for developing new bee inspired optimization techniques. Using a model of the nest-site selection process of real bees, we empirically investigate its optimization potential. In particular, we determined if this model works in dynamic and noisy environments. Our results are promising and suggest that nest-site selection can be indeed useful in the context of optimization.

Introduction

Identifying and mimicking concepts underlying natural phenomena and applying them to solve problems in fields such as computer science, material science and engineering, has grown into a research field in itself. So-called nature inspired computation has given rise to computational concepts which are almost ubiquitous in computer science such as neural networks (Haykin (1999)), evolutionary computation (Eiben and Smith (2003)), and swarm intelligence (Bonabeau et al. (1999)).

Swarm intelligence tackles problems of various computational domains (e.g., robotics and optimization (Blum and Merkle (2008))) using the collective behavior of simple decentralized, self-organized systems. The result has been the emergence of several prominent meta-heuristics e.g., ant colony optimization (for an overview see Dorigo and Stützle (2004)) and particle swarm optimization (for an overview see Poli et al. (2007)).

Due to their decentralized collective behavior, honey bees have become an important model system in the field of swarm intelligence. Honey bee colonies tackle several complex tasks such as maintaining a constant hive temperature (Jones et al. (2004)), adapting to changing foraging conditions (Beekman et al. (2007)) or deciding on the best possible nest site available (Seeley and Buhrman (2001)). Several algorithms based on the honey bees' collective behavior have been developed and applied to various domains such as network routing, robotics, multi-agent systems, and optimization (see (Karaboga and Akay (2009)) for a recent review on bee inspired algorithms). Existing optimization algorithms based on principles of honey bee behavior usually mimic either foraging or mating behavior.

Mating-inspired optimization algorithms are closely related to methods found in evolutionary computation. They are based on the fact that genetic heterogeneity among workers typically increases a colony's fitness (Fuchs and Schade (1994)). In honey bees genetic heterogeneity is achieved via the queen mating with several males (polyandry). While some mating inspired methods constitute new operators for existing methods in evolutionary computation (e.g., Sato and Hagiwara (1997); Jung (2003); Karci (2004)), others try to mimic the mating flight both on a behavioral and genetic level (see, Abbass (2001)).

Foraging-inspired optimization algorithms make use of the bees' decentralized foraging behavior. During foraging honey bees balance the trade-off between exploiting known food sources and scouting for new food sources in a dynamic environment (Beekman et al. (2007)). Bees use a communication mechanism called the "waggle dance" which enables them to transfer information about found food sources to other colony members. The dance encodes the distance and direction to a food source as well as its quality. On the basis of available dances, bees entering the foraging process decide to become dedicated to a specific source (exploit) or to start searching for new sources (explore). Optimization algorithms based on the foraging concept consist of a number of agents, so-called artificial bees. As in nature, the purpose of the agents is twofold. On the one hand they search for new solutions (i.e., food sources) in problem space, on the other hand they try to improve (i.e., exploit) existing solutions using local search. The ratio between exploration and exploitation behavior depends on the number and quality of

available solutions. Several foraging based algorithms have been proposed such as the artificial bee colony optimization (ABC) (Karaboga (2005)), the bees algorithm (BA) (Pham et al. (2006)), the bee colony optimization (BCO) (Teodorovic and Dell'Orco (2005)) or the bee colony optimization algorithm (BCOA) (Chong et al. (2006)).

Here we introduce a third possible class of optimization algorithms which is based on the bees' nest-site selection behavior. After a colony produces new queens, the old queen will leave the nest with approximately a third of the colony members while a young queen perpetuates the old colony. The homeless swarm now has to find a new nest-site (detailed information on the underlying biological mechanisms are provided in the next Section). This is not an easy task as a swarm needs to select the best site out of many possible sites. While during foraging typically several resources are exploited simultaneously, nest-site selection constitutes a decision process, as a swarm has to decide on one nest site by solving the best-of-n-problem (Seeley and Buhrman (2001)).

Bees face a speed-accuracy trade-off when trying to find a new nest site. A decision needs to be made quickly as a swarm is vulnerable to predation and inclement weather, but not too fast which could lead to the swarm settling for a suboptimal nest site. Hence, the decision-making process has to account for temporal delays in nest site discoveries and needs to exhibit sufficient flexibility in order to incorporate late discovered nest sites into the decision-making process.

In terms of optimization, the principles underlying nest-site selection seem of particular interest for dynamic optimization problems, where the problem space changes during the optimization process. We use a biological model of nest-site selection to test the applicability of nest-site selection in the context of optimization. We do this by testing nest-site selection in situations innate to dynamic optimization problems. Additionally we will demonstrate how iterative nest-site selection can lead to function optimization.

This article is structured as follows. Section 2 briefly outlines the biological principles underlying nest-site selection in honeybees. In Section 3 we introduce a biological model of nest-site selection. Based on this model we present various experiments on the applicability of the nest-site selection process to optimization in Section 4. We finish with a summary and conclusions in Section 5.

Nest Site Selection in Honey Bees

One of the most impressive examples of decentralized decision-making in animals is how bees decide on a new home. When a bee colony reaches a certain size it will start to reproduce and rear new queens. Once the young queen is nearly mature, the old queen leaves the old nest in order to give way for her daughter queen (Winston (1987)).

After leaving the nest the homeless swarm temporarily settles on a branch of a tree or on an overhang forming a tight cluster around the queen. Scouts now leave the swarm to search for potential nest sites such as tree hollows or crevices in buildings. Only about 5% of the bees engage in the nest-site selection process while the rest will stay clustered around the queen (Seeley et al. (1979)). If a scout has found a suitable cavity, it will assess its quality (i.e., volume, height, aspect of the entrance, and entrance size) (Seeley and Morse (1978)).

If the site is of sufficient quality, the scout returns to the swarm cluster and performs a waggle dance to advertise the site. The dance encodes the direction and distance to the site. The number of dance circuits in the first dance performed by a returning scout is positively correlated with the scout's perception of the site's quality. By following a dance, bees can learn about the nest-site's location, visit it and then independently evaluate its quality.

After finishing its dance, the scout revisits the site for re-evaluation, which is again followed by returning to the cluster and advertising the site. The number of dances a scout performs for the same nest-site over consecutive visits decreases by around 16 dance circuits (Seeley and Visscher (2008)) per visit regardless of the site's quality (Seeley (2003)). This implies that sites of high quality will be advertised for longer than sites of poor quality due to the higher number of initial circuits. Thus over time more individuals are recruited to high quality sites compared to sites of lower quality.

While inspecting a potential nest site, a scout also assess how many other scouts are present at that site. A specific site is chosen if the number of scouts present exceeds a certain threshold ("quorum"). Scouts then return to the swarm and start "piping" on the swarm cluster. Piping constitutes an auditory signal produced by wing vibration (Seeley and Visscher (2003)), it informs the swarm members that a decision has been made and prepares them for lift off (Visscher and Seeley (2007)).

Once a swarm is airborne it will fly towards the chosen site. The exact mechanism underlying the guidance process is still debated. A well established hypothesis is that informed scouts guide the swarm towards a new location by flying rapidly through the swarm in the direction of the nest site (Schultz et al. (2008); Latty et al. (2009)). Finally after reaching the new nest-site the bees move in and establish a new colony.

Bee Nest Site Selection as an Optimization Process

This section introduces a model of the honeybees' nest-site selection process. It extends a previous model developed by Janson et al. (2007) by including spatial features of nest sites in the model. This extension allows studying the impact of different spatial nest-site distributions. We also introduced noise in the system that affects the scout's perception of the site's quality. We use our model to test the applica-

bility of nest-site selection to optimization problems. The reader should be aware that any observed optimization will be coarse and slow. This is because the presented model is intended for biological simulations and has not been adjusted for optimization. Nevertheless it will allow us to assess the optimization potential of the nest-site selection process.

The model only simulates a fraction of the swarm i.e., the bees involved in the decision-making process during nest-site selection. The model operates in discrete time-steps with each time step corresponding to 1 second of real time. As bees need to find potential nest sites in a spatial environment such a fine temporal resolution is crucial. Real bees are able to travel with a maximum speed of 5 meters per second (Beekman et al. (2006)), thus any coarser time resolution would lead to scouts missing potential nest sites by simply flying over it.

At every simulation-step each bee is in a behavioral state associated with nest-site selection and will act accordingly. Some states E have an associated specific mean duration time T_E. The exact duration is determined by $T(E) = \lambda \cdot T_E$, where $\lambda = \mu/10$ is a scalar factor, with μ being drawn from a chi-square distribution with mean value 10 ($\chi^2(10)$). Note that this leads to an expected value of 1 for λ. There are 8 possible behavioral states:

- REST: The bee is on the swarm but currently not involved in nest-site selection
- SEARCH: The bee is on the swarm and tries to find a dance to follow
- SCOUT: The bee searches the surroundings for potential nest sites
- ASSESS: The bee is at a potential nest site and assesses its quality
- DANCE: The bee is on the swarm and dances for its preferred site
- FOLLOW: The bee is on the swarm and has found a dance and follows it
- RECRUITED: The bee flies to the nest site advertised in the dance it followed
- MISS: The bee misread the dance and searches the surrounding of the swarm unsuccessfully before returning to the swarm

Figure 1 depicts a state diagram that outlines a bee's state transitions in the model. In the following the behavior that corresponds to the different states will be explained in more detail.

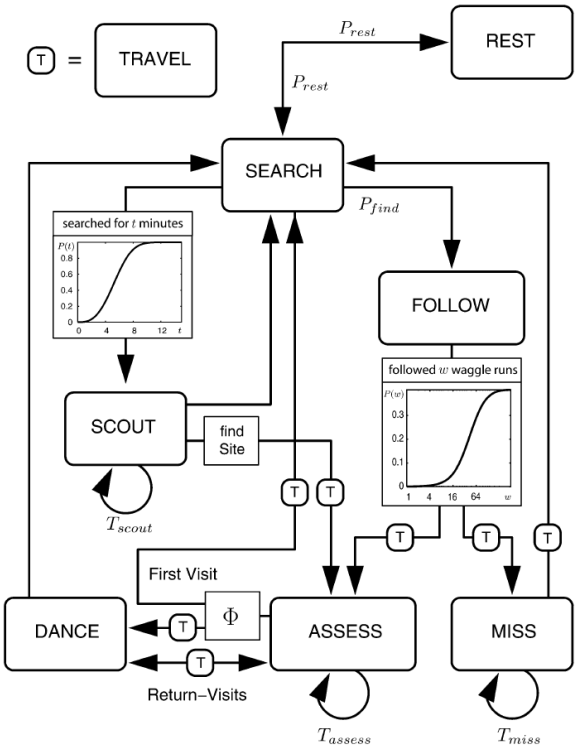


Figure 1: State diagram of individual behavior underlying nest-site selection. Reprint from Janson et al. (2007)

Resting A resting bee will engage in the nest-site selection process by starting to search for a dance to follow with a probability of $P_{rest} = 0.002$ per second (Beekman et al. (2007)). A searching bee will switch to the resting state with the same probability.

Searching The number of dances that are performed on the swarm for potential nest-sites affects the likelihood of a searching bee finding and joining a dance. Let D be the number of dances currently performed on the swarm. The probability that a searching bee will locate a dance is given by $P_{find} = 0.005 \cdot D$. If it is able to find a dance it is randomly assigned to one of the available dances. Experimental studies have shown that dances comprised a maximum of 7 followers. The probability that a bee will start to follow the dance it was assigned to is thus given by $P_{follow} = 0.2^{min\{2,f\}}$, with f denoting the number of bees already following the dance.

The longer a searching bee is unable to find and join a dance, the more likely it becomes that it will switch to proactive scouting behavior and try to find a suitable nest-site itself. The probability that a bee switches from searching to scouting behavior is given by $P_{scout}(t) = t^2/t^2 + \theta^2$ where t denotes the number of time steps of unsuccessful searching and $\theta = 4000$. Note that this switching mechanism modulates the exploration/exploitation rate of the swarm. Scout-

ing is very likely when only few or low quality nest-sites have been found and thus only a few dances are available. When many sites have been found and dances are abundant, a searching bee is likely to find a dance to follow and will become a recruit instead of a scout.

Scouting Lindauer observed that bees usually scout the surroundings for about 20 minutes before returning to the swarm (Lindauer (1955)). We thus used a mean scout duration time of $T_{scout} = 1200$. While scouting the virtual bees move through a 2-dimensional environment in search of potential nest sites. This is a major difference to the previous model where scouting was modeled probabilistic. The scouting process can be divided into two phases:

1. scouting: a bee will scout as long as it is able to be back at the swarm after T_{scout} time steps.

2. returning: if the remaining scouting time is smaller or equal to the time needed to return to the swarm a scout returns to the swarm.

In nature a bee can spot a target if the target subtends the bee's visual angle α_{min} which can range between five and fifteen degrees (Giurfa et al. (1996)). The diameter of nest boxes normally used in nest-site selection experiments is around 40cm. Given an assumed minimal angle of $\alpha_{min} = 8$ degrees, a scout can spot a nest site up to a distance of approximately 280cm. After a successful discovery a scout will immediately start to assess the site and thus change its state.

Scouting Strategy Please note that the exact way scouts search the environment is still unknown. Some studies suggest that bees search in a scale-free fashion (Reynolds et al. (2007)) but this is still debated (Benhamou (2008)). In this model the scouts' search strategy is realized as an intermittent search strategy (Benichou et al. (2005)). When starting to scout a bee will choose a random location within a search area that is defined by the range of locations that are reachable within one third of its available scouting time T_{scout}. After reaching the chosen location a scout will start to search the surrounding for potential nest-sites using a correlated random walk (CRW) (Bartumeus et al. (2005)) with a fixed movement length of 1m per step and a correlation parameter value of $\rho = 0.5$ resulting in slightly correlated movement steps.

Flying towards a destination Scouts fly towards a destination with a travel-speed of 5m/sec. A scout is placed on its destination (i.e., reaches it) when its distance to the destination is less than 5m. Angular noise from a uniform random distribution η_{fly} ($-22.5 \leq \eta_{fly} \leq 22.5$) was added to prevent bees from flying in straight lines.

Site assessment After locating a potential nest site a scout will immediately start to assess it. In nature nest-site assessment usually lasts for about 10 minutes Lindauer (1955) which corresponds to mean assessment duration time of $T_{assess} = 600$. In the model each nest site S is associated with a certain quality Q_S ($0 \leq Q_S \leq 100$). When assessing a nest site a bee will perceive the quality. Quality is always perceived with some noise, thus $Q(S) = Q_S + \delta$, with δ drawn from a normal distribution $N(0, \sigma^2)$ with a standard deviation of $\sigma = 10$. A bee will only dance for a given nest-site S if the perceived quality $Q(S)$ exceeds a bee's quality threshold Φ. Otherwise the bee will switch to search behavior after returning to the swarm. Here a uniform threshold value $\Phi = 50$ is used for all individuals.

Dancing If a bee discovered a suitable nest site S while scouting it will advertise it after returning to the swarm by means of a waggle dance. The number of waggle runs performed during a dance depends on the perceived quality of the site $Q(S)$ and the number of consecutive visits to the site. Based on empirical data (Seeley (2003)), the virtual bees perform $Q(S)$ waggle runs after their first visit to the site and $Q(S) - 16(k-1)$ after the kth return. Bees will stop promoting a site (i.e., stop dancing) and switch to searching and if $Q(S) - 16(k-1) \leq 0$.

A waggle run encodes the distance and the direction to the potential nest site. This has also been incorporated into the model's dance behavior. Based on empirical data (Gardner et al. (2008)) we assume that a waggle phase lasts 2.4sec per kilometer of distance to the potential nest site plus 1.5 sec for the return phase.

Following A bee following a dance will follow the dance until the dancer ceases dancing. If the follower had previously visited the advertised site, it will find that site again. Otherwise the probability of correctly locating the advertised site depends on the number of waggle runs w the bee followed. Based on experimental data (Mautz (1971)) the probability of finding a nest site is $P_{findSite}(w) = s(w)/1.5 \cdot u(w) + s(w)$ where w denotes the number of followed waggle runs, $u(w) = 1 - 1/\sqrt{(w+1)}$ represents the distribution of unsuccessful bees and $s(w) = w^2/(w^2 + \theta)$ with $\theta = 60$ represents the distribution of successful bees.

Successfully recruited to nest site A successfully recruited bee flies towards the proposed nest site and assesses its quality. If it finds its quality sufficient (i.e., $Q(S) > \Phi$), the bee will advertise the site after returning to the swarm. Otherwise it will search for new dances after its return to the swarm.

Missing the advertised nest site If a bee is not able to read a dance correctly it will not be able to find the advertised site. In such cases, the bee flies the same distance as the advertised site, but in a slightly wrong direction. In the

model this is achieved by adding a maximum of 5 degree noise drawn from a uniform random distribution to the actual direction towards the advertised nest site. After reaching the wrong location a bee searches the surroundings for 400sec.

Experiments

To investigate the optimization potential of the honeybee's nest-site selection process, we performed three experiments using the model described above. Unless stated otherwise we used the parameter values mentioned in the last section. We present the results as average values obtained from 10 independent runs. The number of individuals used in the experiments was set to $n = 500$, which corresponds to the number of bees involved in nest-site selection in real honey bees.

Experiment 1: Nest-site selection in a dynamic environment This experiment was performed to test how the nest-site selection process performs in a dynamic environment. While a change in a site's quality during the selection process is unlikely to occur in nature, changing or moving optima are ubiquitous in dynamic optimization problems.

The environment contains two potential nest sites $n1$, $n2$ that are located in opposite directions 150m away from the swarm's position. Initially site $n1$ is of good quality $q_{good} = 75$ while $n2$ is of bad quality $q_{bad} = 45$. The sites qualities however switch during the course of the simulation i.e., at every interval of 28800 simulation steps (i.e., every 8 hours) the qualities of the nest sites are swapped. A simulation runs for 32 hours corresponding to 115200 simulation steps and thus a total number of 3 quality switches occur during one run.

As the search process is performed in a spatial environment it is likely that a swarm only discovers one nest site or even none. Additionally a swarm might forget a low quality nest-site as dances might not sustain during the low quality period. In order to ensure that the swarm is aware of both sites each time a quality change occurs, a randomly chosen bee will start dancing for the nest site that was of low quality but switched to high quality.

Figure 2 depicts the time evolution of the number of bees at each nest site. As can be seen the swarm is able to quickly adapt to changes in nest-site quality. The number of bees at a given nest-site will not exceed ≈ 400 because a fraction of the swarm is resting, very few will still scout for different nest-sites and bees at a given nest-site will return to the swarm to promote it. In terms of optimization this process is still rather slow as it takes the swarm approximately 2 hours to adapt to the change in quality. Slow adaption is not necessarily a disadvantage as it makes a swarm resilient against noise. As pointed out before quality changes are unlikely to happen in nature, however discovering new sites in the course of the selection process constitutes a similar change

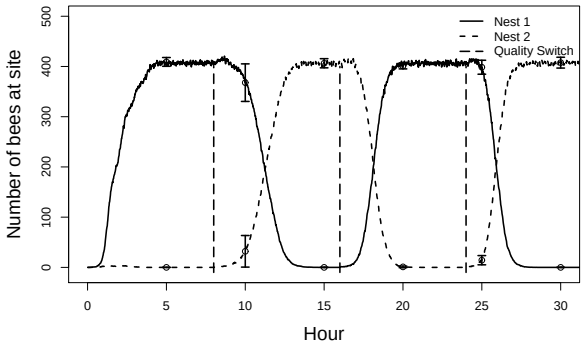


Figure 2: Time evolution of the number of bees assessing a nest site where the site qualities change occur every 28800 simulation steps. Error bars represent the standard deviation.

in the swarm's environment. Without the ability to react to changes in the environment, a swarm can get stuck in a sub-optimal solution if it finds a nest site of mediocre quality early in the decision-making process. In terms of optimization, adapting to a dynamic environment is an interesting aspect, as it can be applied to the detection of changing locations of the optima in problems with dynamic fitness functions.

Experiment 2: Nest-site selection in a noisy environment Here we tested whether the swarm is capable of selecting a stable mediocre quality nest site and disregard a site of sometimes high but very unstable quality.

The number of bees and the number and position of the potential nest sites is the same as in Experiment 1, however here the quality of nest site $n2$ is kept constant at mediocre level $q_{mediocre} = 55$ whereas the quality of site $n1$ changes at an interval of 1800 simulation steps (i.e., every 30 minutes) alternately between good $q_{good} = 75$ and very bad $q_{vbad} = 35$. A simulation again lasted for 115200 simulation steps corresponding to 32 hours. To ensure that the swarm is aware of both sites, a random bee starts dancing for each site in the first simulation step.

Figure 3 depicts the time evolution of number of bees at the two nest sites. Clearly the majority of the swarm selects the stable mediocre nest site. At the start of a simulation the number of bees builds up quickly at both nest sites, due to the fact that one bee starts to dance for each site at the first simulation step. However, over the course of revisiting the sites, more bees get recruited towards the mediocre stable site. The revisit behavior of honeybees plays a key role in that respect. Initially site $n1$ will be promoted stronger than site $n2$ due to the quality difference. The ongoing revisitation will cause recruited and dedicated bees to abandon the unstable site and choose the stable site as it makes it pos-

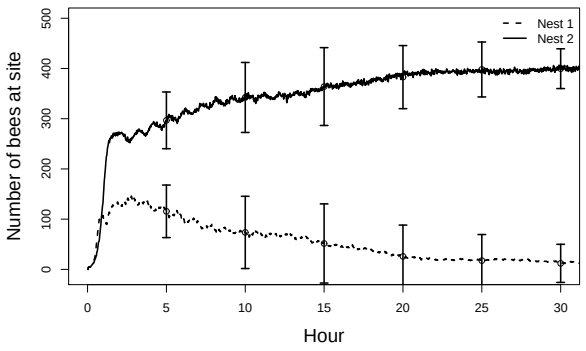


Figure 3: Time evolution of the average number of bees assessing a nest site when the nest site of high quality is very unstable. The quality of nest site $n1$ changes each 1800 simulation steps between $q_{good} = 75$ and $q_{vbad} = 35$, whereas the quality of nest site $n2$ is kept constant at $q_{mediocre} = 55$. Error bars represent the standard deviation.

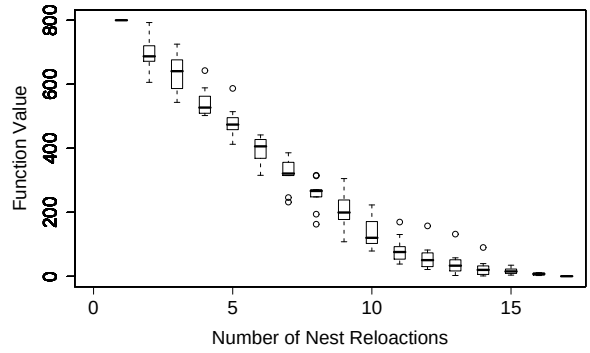
sible for the individuals to gain awareness of the changing quality. Site $n1$ will never be completely abandoned simply because some visiting bees will always experience it as a very good nest site and thus promote and revisit it. In general this experiment demonstrates that the nest-site selection mechanism is to some extent resilient towards noise.

Experiment 3: Function optimization via iterative nest-site selection The European honey bee *Apis mellifera* has very specific requirements regarding its nest site. This is because once a decision is made it is final (i.e., a swarm is very unlikely to relocate after moving into a new nest site). In contrast open nesting bee species such as the Asian Dwarf honey bee *Apis florea* are quite flexible and a swarm might relocate if its initial decision was suboptimal (Oldroyd et al. (2008)).

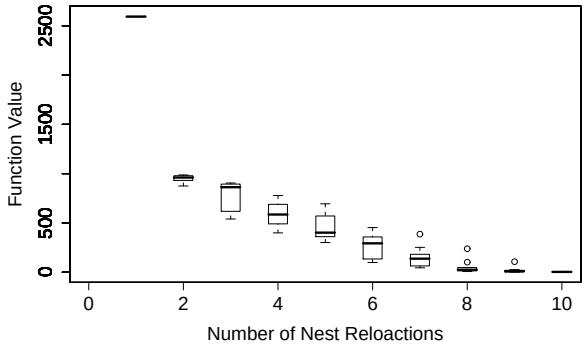
		R
Sphere	$f_{sp}(\vec{x}) = \sum_{i=1}^{n} x_i^2$	$[-25; 25]^n$
Booth	$f_{bt}(\vec{x}) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	$[-10; 10]^n$

Table 1: Test functions and domain space range (R). The dimension of each function is 2.

Such an iterative selection process as found in *Apis florea* can lead to an optimization in an environment with many potential nest sites. In this experiment it is assumed that the swarm's environment corresponds to the search space of a continuous function that needs to be minimized. Each position in the search space corresponds to a potential site, and its quality corresponds to a value of the function at that position. The test functions used in the experiment and their associated parameter values are given in Table 1. Initially the swarm is placed at position [-20,-20] for the Sphere function and [-10,-10] for the Booth function.



(a) Sphere



(b) Booth

Figure 4: Boxplots of the quality of the occupied nest site over several relocations for the two test functions.

For this experiment we changed the bees' scouting behavior because the first version of the extended model is modeled on the behavior of the European honey bee *Apis mellifera* where a scout assesses a nest site for a certain period of time before returning to the swarm. As each location corresponds to a potential nest site, scouts would immediately start to assess sites after a single scouting step. To overcome this, a scout will advertise the best position it found during its scouting period, if the quality of that position is better than quality of the swarms current location.

The quality of a newly discovered site depends on the quality difference regarding the current location of the swarm. If a scout discovers a nest site that is X% better than the swarm's current location this site is assigned quality X.

While recruits fly towards a site that was advertised by a dancing bee, they will actively monitor the quality of the

locations they fly over. If they encounter a better site on their way, they abandon their initial choice and become scouts. Recruits that fail to locate an advertised site also become scouts.

Nest sites are assessed by recruits and returning bees for a certain amount of time. During that time each assessing bee counts the number of other bees present at the site. If the number of bees at the site reaches a given quorum $q = 10$ the swarm is placed on this new site and the nest-site selection process is restarted. The parameter values used in this experiment are: step size $step = 0.1$, scouting time $T_{scout} = 100$, and assessment time $T_{assess} = 20$. A simulation run is stopped when a swarm does not relocate within 3600 simulation steps.

The changes in the quality of the found sites for both test functions over several nest-site relocations is depicted in Figure 4. The bees are able to iteratively optimize the position of the swarm within the search space (i.e., minimize the function value). However the optimization process is limited by several factors: as scout time T_{scout} and step size $step$ are fixed, scouts are only able to explore a certain range around the swarm's current location whereas a fixed step size prevents scouts from finding better solutions as they are likely to fly over them. This is critical when the swarm is close to the global optimum and scouts would need to search on a finer scale in order to find better positions. Another limiting factor is the quality assignment. As the quality difference between solutions decreases around the global optimum the model will always reach a point were better solutions are not selected any more as the quality difference between them is too low. The performance of the nest-site selection process in function optimization is yet by no means comparable to the performance of other optimization algorithms (e.g., Aderhold et al. (2010)). In order to use the nest-site selection paradigm in an algorithm for real optimization problems, the swarm needs to become more sensitive to small quality differences to identify better potential sites when the swarm comes closer to the location of an optimum.

The speed of the decision-making process depends on the quorum q used. The higher q the more bees are needed at a potential nest before the swarm changes its location and the slower the optimization process. The quorum mechanism can however also prove to be useful in terms of optimization, as the existence of a quorum prevents a premature convergence onto local minima, as it gives the bees time to find better sites. Another potential benefit of the quorum is that it requires bees to revisit and reassess a given site several times which is important for dynamic or noisy optimization functions.

Conclusion

Recently bee inspired optimization techniques have become popular within the optimization community but have been restricted to using the bees' foraging behavior and mating behavior. Here we proposed to use the bees' nest-site selection behavior for developing bee inspired optimization techniques. Nest-site selection involves the active discovery of potential sites by scout bees and a decision on the best site. In nature it enables bees to solve the best-of-n-problem (i.e., deciding on the best nest-site). Nest-site selection is thus a decision-making process that has a clear optimum which is in contrast to foraging which mainly regulates the distribution of foragers over available food sources.

We used a model of the nest-site selection process of real bees to investigate its optimization potential. Using this model, we performed three optimization experiments. Our results suggest that the nest-site selection process is able to make the best decision even in dynamic and noisy environments and that the process can detect and decide on the best stable solution even when better but noisier solutions are present. The final experiment demonstrated how an iterative application of the nest-site selection process could be used for function optimization.

Our results corroborate that the honey bee's nest-site selection process is indeed useful in the context of optimization. Future work will involve developing an bee inspired optimization scheme that is based on nest-site selection.

Acknowledgments

This work was supported by the Human Frontier Science Program Research Grant "Optimization in natural systems: ants, bees and slime moulds".

References

Abbass, H. A. (2001). Marriage in honeybees optimization (mbo): A haplometrosis polygynous swarming approach. In *Proceedings of the Congress on Evolutionary Computation*, pages 207–214.

Aderhold, A., Diwold, K., Scheidler, A., and Middendorf, M. (2010). *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, chapter Artificial Bee Colony Optimization: A New Selection Scheme and Its Performance, pages 283–294. Springer Berlin / Heidelberg.

Bartumeus, F., Luz, M. G. E. D., Viswanathan, G. M., and Catalan, J. (2005). Animal search strategies: a quantitative random–walk analysis. *Ecology*, 86(11):3078–3087.

Beekman, M., Fathke, R., and Seeley, T. (2006). How does an informed minority of scouts guide a honey bee swarm as it flies to its new home? *Animal Behavior*, 71(1):161–171.

Beekman, M., Gilchrist, A. L., Duncan, M., and Sumpter, D. J. T. (2007). What makes a honeybee scout? *Behavioral Ecology and Sociobiology*, 61:985–995.

Benhamou, S. (2008). How many animals really do the levy walk? *Ecology*, 89:2351–2352.

Benichou, O., Coppey, M., Moreau, M., Suet, P.-H., and Voituriez, R. (2005). Optimal search strategies for hidden targets. *Physical Review Letters*, 94:198101.

Blum, C. and Merkle, D., editors (2008). *Swarm Intelligence: Introduction and Applications*. Springer Berlin / Heidelberg.

Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm intelligence: from natural to artificial systems*. Oxford University Press.

Chong, C. S., Low, M. Y. H., Sivakumar, A. I., and Gay, K. L. (2006). A bee colony optimization algorithm to job shop scheduling. In *Proceedings of the 2006 Winter Simulation Conference*.

Dorigo, M. and Stützle, T. (2004). *Ant Colony Optimization*. MIT Press.

Eiben, A. E. and Smith, J. E. (2003). *Introduction to Evolutionary Computing*. Springer Berlin / Heidelberg.

Fuchs, S. and Schade, V. (1994). Lower performance in honeybee colonies of uniform paternity. *Apidologie*, 25:155–168.

Gardner, K. E., Seeley, T. D., and Calderone, N. W. (2008). Do honeybees have two discrete dances to advertise food sources? *Animal Behaviour*, 75:1291–1300.

Giurfa, M., Vorobyev, M., Kevan, P., and Menzel, R. (1996). Detection of coloured stimuli by honeybees: minimum visual angles and receptor specific contrasts. *Journal of Comparative Physiology A*, 178:699–709.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. Prentice Hall.

Janson, S., Middendorf, M., and Beekman, M. (2007). Searching for a new home – scouting behavior of honeybee swarms. *Behavioral Ecology*, 18:384–392.

Jones, J., Myerscough, M. R., Graham, S., and Oldroyd, B. P. (2004). Honey bee nest thermoregulation: diversity promotes stability. *Science*, 305:402–404.

Jung, S. H. (2003). Queen-bee evolution for genetic algorithms. *Electronics Letters*, 39(6):575–576.

Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.

Karaboga, D. and Akay, B. (2009). A survey: algorithms simulating bee swarm intelligence. *Artificial Intelligence Review*.

Karci, A. (2004). *PRICAI 2004: Trends in Artificial Intelligence*, chapter Imitation of Bee Reproduction as a Crossover Operator in Genetic Algorithms, pages 1015–1016. Springer Berlin / Heidelberg.

Latty, T., Duncan, M., and Beekman, M. (2009). High bee traffic disrupts transfer of directional information in flying honeybee swarms. *Animal Behaviour*, 78:117–121.

Lindauer, M. (1955). Schwarmbienen auf Wohnungssuche. *Zeitschrift für vergleichende Physiologie*, 37:263–324.

Mautz, D. (1971). Der Kommunikationseffekt der Schwänzeltänze bei *Apis mellifera carnica*. *Zeitschrift für vergleichende Physiologie*, 72:192–220.

Oldroyd, B. P., Gloag, R. S., Even, N., Wattanachaiyingcharoen, W., and Beekman, M. (2008). Nest-site selection in the open-nesting honey bee apis florea. *Behavioral Ecology and Sociobiology*, 62:1643–1653.

Pham, D., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., and Zaidi, M. (2006). The bees algorithm a novel tool for complex optimisation problems. In *Proceedings of IPROMS 2006 Conference*, pages 454–461.

Poli, R., Kennedy, J., and Blackwell, T. (2007). Particle swarm optimization: An overview. *Swam Intelligence*, 1:33–57.

Reynolds, A. M., Smith, A. D., Menzel, R., Greggers, U., Reynolds, D. R., and Riley, J. R. (2007). Displaced honey bees perform optimal scale-free search fights. *Ecology*, 88:1955–1961.

Sato, T. and Hagiwara, M. (1997). Bee system: finding solution by a concentrated search. In *IEEE International Conference on Systems, Man, and Cybernetics, 1997. 'Computational Cybernetics and Simulation'*.

Schultz, K. M., Passino, K. M., and Seeley, T. D. (2008). The mechanism of flight guidance in honeybee swarms: subtle guides or streaker bees? *Journal of Experimental Biology*, 211:3287–3295.

Seeley, T. (2003). Consensus building during nest–site selection in honey bee swarms: the expiration of dissent. *Behavioral Ecology and Sociobiology*, 53:417–424.

Seeley, T. and Buhrman, S. (2001). Nest–site selection in honey bees: how well do swarms implement the "best-of-n" decision rule. *Behavioral Ecology and Sociobiology*, 49:416–427.

Seeley, T. and Visscher, P. K. (2008). Sensory coding of nest-site value in honeybee swarms. *Journal of Experimental Biology*, 211:3691–3697.

Seeley, T. D. and Morse, R. A. (1978). Nest site selection by the honey bee, apis-mellifera. *Insectes Sociaux*, 25:323–337.

Seeley, T. D., Morse, R. A., and Visscher, P. K. . (1979). The natural history of the flight of honey bee swarms. *Psyche*, 86:103–113.

Seeley, T. D. and Visscher, P. K. (2003). Choosing a home: how the scouts in a honey bee swarm perceive the completion of their group decision making. *Behavioral Ecology and Sociobiology*, 54:511–520.

Teodorovic, D. and Dell'Orco (2005). Bee colony optimziation - a cooperative learning approach too complex transportation problems. In *Advanced OR and AI Methods in Transportation. Proceedings of the 10th Meeting of the EURO Working Group on Transportation*.

Visscher, P. K. and Seeley, T. D. (2007). Coordinating a group departure: who produces the piping signals on honeybee swarms? *Behavioral Ecology and Sociobiology*, 61:1615–1621.

Winston, M. L. (1987). *The Biology of the Honey Bee*. Harvard University Press.

Learning Games using a Single Developmental Neuron

Gul Muhammad Khan[1] and Julian F. Miller[2]

[1]NWFP UET Peshawar, Pakistan, gk502@nwfpuet.edu.pk
[2]University of York, UK
jfm7@ohm.york.ac.uk

Abstract

An agent controlled by a single developmental neuron is trained to play arcade game. Genetic programming is used to find the DNA of neuron such that it can learn and store the learned information in the form of development in its architecture and updates in chemical concentration. The developmental neuron consists of dendrites, axons, and synapses that can grow, change and die. The structure of this neuron complexify itself at runtime as a result of game scenarios. The network is tested in arcade game environment of checkers. The agent has to recognize the patterns of the board and use this information to learn how to play the game better. The network is evolved against a professional checker program for its capability to learn. Input from the board is provided using sensory neuron through synapses. The developmental neuron process these signals and send output to the motor neurons to make a move. The structure of the neuron is also modified during signal processing. The developmental neuron successfully defeated the professional minimax based checker program during evolution by a large margin. We also tested the agent against some other opponents (not seen during evolution) of various levels for its generality and it proves to outperform them.

Introduction

In this paper we present the idea of developmental neuron capable of learning and adaptation. We have adopted the view that the intelligent behaviour of human being is the consequence of the special DNA. It is the DNA that is responsible for development of human body and brain. DNA of humans are different from other organisms that is why human can interact with each others. We beleive if we somehow manage to identify the functionality of human DNA and provide it with a neuron like structure we will be able to produce intelligent behaviour. Learning in brain is the consequence of biological development thus if we somehow manage to identify the rules for development we would be able to produce a learning system. DNA does not in itself encode learned information. Recent results demonstrated that even a single neuron has the capability of learning and adaptation as evident from the experimental results on snail aplysia (Kandel et al. (2000)). We have used Cartesian Genetic Programming (CGP) to develop a neuron having branching structure (Miller and Thomson (2000)). CGP represent the genotype (DNA) inside neuron responsible for development and signal processing. We evolved genotypes that encode programs that *when executed* gives rise to a neuron with developmental structure that can play checkers at higher level. The developmental and signalling functions are distributed at various segments (soma, axon, dendrite) inside neuron similar to biology (Zubler and Douglas (2009)).

We have produced an artificial agent that used this developmental neuron as its computational system. The agent receive information from checkers board using sensory neurons. Sensory neuron has a number of axonal branches that are distributed in the vicinity of CGP neuron and provide signal to them by making synapse. Synaptic transformation of signal is done using a CGP program similar to the one inside DNA of CGP neuron. CGP neuron recieves the external information in the form of its dendrite branches potential updates. This signal is then processed by CGP neuron using its DNA and a decision signal is transferred in forward direction to the motor neuron having dendrite branches distributed in the vicinity of CGP neuron.

The genotype inside CGP developmental neuron is a set of computational functions that are inspired by various aspects of biological neurons. Each agent (player) has a genotype that grows a computational neural structure (phenotype). The initial genotype that gives rise to the dynamic neural structure is obtained through evolution. As the number of evolutionary generations increases the genotypes develop structure that allow the players to play checkers increasingly well.

We have used an indirect encoding scheme in which the rules of network (CGP Neuron) are evolved instead the network directly. When we run these evolved programs they can adjust the network indefinitely. This allows our network to learn while it develops during its lifetime. The network begins as small randomly defined structure of neuron with dendrites and axosynapses. The job of evolution is to come up with genotypes that encode *programs* that when *executed* develop into mature neural structures that learn through environmental interaction and continued development. So the

complexity of the evolved programs is independant of the complexity of the task. The network continue to develop and complexify itself based on the environmental conditions.

A number of indirect methods are used in ANNs that evolve the rules for development of the network. ANN although inspired by biological nervous system has only few notions of biological brain. Here we have extended the view and identified a number of other important features that need to be added to individual neuron structure. These features prove to be extremly important for learning and memory. Memory and learning in brain is caused by many other mechanisms. Synaptic weights are only responsible for extremely short term memory (Kleim et al. (1998)), long term memory is stored in the structure of the neuron (Terje (2003)). The network presented here is an inspiration of biology, not the implementation of biology.

We have evolved the genetic programs inside CGP neuron that develop during the course of the game playing against a fixed level minimax program that plays checkers. At the start, the genes of neuron were random so the neuron behaviour was not that good during the course of game. As evolution progresses, the genes started to develop the neuron from an initial random structure such that it can understand the pattern of the board and use this information to make various intelligent moves such that it can beat a human intelligent based computer program. The opponent make moves based on the intelligence of humans who developed the program whereas the CGP developmental neuron evolved the inteligent genes that can cause a developmental neural structure that is capable of understanding the pattern of the board and play a move. The agent with a single neuron make a number of intelligent moves before it beat the opponent. These results prove that it is possible to evolve the genes that can produce networks capable of learning and intelligent decision making. To date, not a single developmental system proved to be capable of learning behaviour. This is the first time in the history of computational evolution that learning genes are evolved. The neuron structure continue to develop and change during the game. The results presented in paper clearly demonstrate that the learning capability of the agents improves over the course of evolution.

Cartesian Genetic Programming (CGP)

CGP is a well established and effective form of Genetic Programming. It represents programs by directed acyclic graphs (Miller and Thomson (2000)). The genotype is a fixed length list of integers, which encode the function of nodes and the connections of a directed graph. Nodes can take their inputs from either the output of any previous node or from a program input (terminal). The phenotype is obtained by following the connected nodes from the program outputs to the inputs. We have used function nodes that are variants of binary if-statements known as 2 to 1 multiplexers (Miller and Thomson (2000)). Multiplexers can be considered as atomic in nature as they can be used to represent any logic function (Miller and Thomson (2000)).

In CGP an evolutionary strategy of the form $1 + \lambda$, with λ set to 4 is often used (Miller and Thomson (2000)). The parent, or elite, is preserved unaltered, whilst the offspring are generated by mutation of the parent. If two or more chromosomes achieve the highest fitness then *newest* (genetically) is always chosen.

Developmental Models of Neural Networks

A number of developmental techniques are introduced to capture the learning capabilities by having time dependent morphologies. Nolfi et al presented a model in which the genotype-phenotype mapping (i.e. ontogeny) takes place during the individual's lifetime and is influenced both by the genotype and by the external environment (Nolfi et al. (1994)).

Cangelosi proposed a related neural development model, which starts with a single cell undergoing a process of cell division and migration until a neural network is developed (Cangelosi et al. (1994)). The rules for cell division and migration is specified in genotype, for a related approach see (Gruau (1994)).

Rust and Adams devised a developmental model coupled with a genetic algorithm to evolve *parameters* that grow into artificial neurons with biologically-realistic morphologies. They also investigated activity dependent mechanisms so that neural activity would influence growing morphologies (Rust et al. (1997)).

Federici presented an indirect encoding scheme for development of a neuro-controller (Federici (2005)). The adaptive rules used were based on the correlation between post-synaptic electric activity and the local concentration of synaptic activity and refractory chemicals.

Roggen et al. devised a hardware cellular model of developmental spiking ANNs (Roggen et al. (2007)). Each cell can hold one of two types of fixed input weight neurons, excitatory or inhibitory each with one of 5 fixed possible connection arrangements to neighbouring neurons. In addition each neuron has a fixed weight external connection. The neuron integrates the weighted input signals and when it exceeds a certain membrane threshold it fires. This is followed by a short refractory period. They have a leakage which decrements membrane potentials over time.

In almost all previous work the internal functions of neurons were either fixed or only parameters were evolved. Connections between neurons are simple wires instead of complicated synaptic process. The model we propose is inspired by the characteristics of real neurons.

Key features and biological basis for the model

Features of biological neural systems that we think are important to include in our model(Cartesian Genetic Programming Developmental Neuron (CGPDN)) are synaptic trans-

mission, and synaptic and developmental plasticity. Signalling between biological neurons happens largely through synaptic transmission, where an action potential in the pre-synaptic neuron triggers a short lasting response in the post-synaptic neuron (Shepherd (1990)). In our model signals received by a neuron through its dendrites are processed and a decision is taken whether to fire an action potential or not.

Neurons in biological systems are in constant state of change, their internal processes and morphology change all the time based on the environmental signals. The development process of the brain is strongly affected by external environmental signals. This phenomenon is called Developmental Plasticity. Developmental plasticity usually occurs in the form of synaptic pruning (Van Ooyen and Pelt (1994)). This process eliminates weaker synaptic contacts, but preserves and strengthens stronger connections. More common experiences, which generate similar sensory inputs, determine which connections to keep and which to prune. More frequently activated connections are preserved. Neuronal death occurs through the process of apoptosis, in which inactive neurons become damaged and die. This plasticity enables the brain to adapt to its environment.

A form of developmental plasticity is incorporated in our model, branches can be pruned, and new branches can be formed. This process is under the control of a 'life cycle' chromosome (described in detail in section 6) which determines whether new branches should be produced or branches need to be pruned. Every time a branch is active, a life cycle program is run to establish whether the branch should be removed or should continue to take part in processing, or whether a new daughter branch should be introduced into the network.

Starting from a randomly connected network, we allow branches to navigate (Move from one grid square to other, make new connections) in the environment, according to the evolutionary rules. An initial random connectivity pattern is used to avoid evolution spending extra time in finding connections in the early phase of neural development.

Changes in the dendrite branch weight are analogous to the amplifications of a signal along the dendrite branch, whereas changes in the axon branch (or axo-synaptic) weight are analogous to changes at the pre-synaptic level and post-synaptic level (at synapse). Inclusion of a soma weight is justified by the observation that a fixed stimulus generates different responses in different neurones.

Through the introduction of a 'life cycle' chromosome, we have also incorporated developmental plasticity in our model. The branches can self-prune and can produce new branches to evolve an optimized network that depends on the complexity of the problem (Van Ooyen and Pelt (1994)).

The CGP Neuron

This section describes in detail the structure and processing inside the CGP Neuron and the way inputs and outputs are interfaced with it.

The CGP Neuron is placed at a random location in a two dimensional spatial grid (as shown in figure 1). It is initially allocated a random number of dendrites, dendrite branches, one axon and a random number of axon branches. Neurons receive information through dendrite branches, and transfer information through axon branches to neighbouring dendrite branches. The branches may grow or shrink and move from one grid point to another. They can produce new branches and can disappear. Axon branches transfer information only to dendrite branches in their proximity. Electrical potential is used for internal processing of neurons and communication between neuron and is represented by an integer.

Health, Resistance, Weight and Statefactor

Four variables are incorporated into the CGP Neuron, representing either fundamental properties of the neuron (*health*, *resistance*, *weight*) or as an aid to computational efficiency (*statefactor*). The values of these variables are adjusted by the CGP programs. The *health* variable is used to govern replication and/or death of dendritic and axonal connections. The *resistance* variable controls growth and/or shrinkage of dendrites and axons. The *weight* is used in calculating the potentials in the network. Each soma has only two variables: *health* and *weight*. The *statefactor* is used as a parameter to reduce computational burden, by keeping neuron and branches inactive for a number of cycles. Only when the *statefactor* is zero are the neuron and branches are considered to be active and their corresponding program is run. *Statefactor* is affected indirectly by CGP programs.

Inputs, Outputs and Information Processing

The signal is transferred to and taken from this neuron using virtual axon and dendrite branches by making synaptic connections.

The signal from the environment is applied to CGP neuron using virtual input axo-synaptic connections. There are also virtual output dendrite branches used as the output of the system. The virtual axo-synaptic branches are allowed to not only transfer signals to the dendrite branches of processing neuron (CGP Neuron) but also to the output virtual dendrite branches which is the output of the system. The CGP Neuron transfers signals to the virtual output dendrite branches using the program encoded in the axo-synaptic chromosome.

Information processing in the CGP Neuron starts by selecting the list of dendrites and running the electrical dendrite branch program. The updated signals from dendrites are averaged and applied to the soma program along with the soma potential. The soma program is executed to get the final value of soma potential, which decides whether a neuron should fire an action potential or not. If soma fires, an action potential is transferred in forward direction using axo-synaptic branch programs.

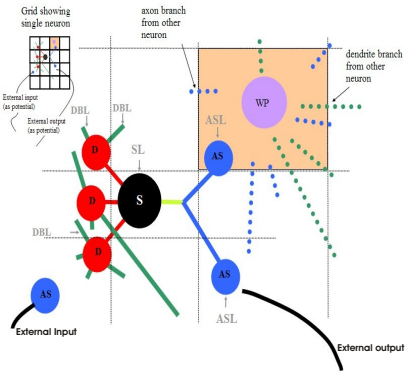


Figure 1: On the top left a grid is shown containing a single neuron. The rest of the figure is an exploded view of the neuron is given. Electrical processing parts containing dendrite (D), soma (S) and axo-synapse branch (AS) is shown as part of neuron. Developmental programs responsible for the 'life-cycle' of neural components (shown in grey) are also shown. They are dendrite branches (DBL), soma (SL) and axo-synaptic branches (ASL). The weight processing (WP) block shown is used to adjusts synaptic and dendritic weights.

Functionality of CGP Neuron

Neural functionality is divided into three major categories: electrical processing, life cycle and weight processing. These categories are described in detail below.

Electrical Processing

The electrical processing part is responsible for signal processing inside neuron and communication between neurons. It consists of dendrite branch, soma, and axo-synaptic branch electrical chromosomes.

The dendrite program D, handles the interaction of dendrite branches belonging to a dendrite. It take active dendrite branch potentials and soma potential as input and updates their values. The *Statefactor* is decreased if the update in potential is large and vice versa. If any of the branches are active (has its statefactor equal to zero), their life cycle program (DBL) is run, otherwise D continues processing the other dendrites.

The soma program S, determines the final value of soma potential after receiving signals from all the dendrites. The processed potential of the soma is then compared with the threshold potential of the soma, and a decision is made whether to fire an action potential or not. If it fires, it is kept inactive (refractory period) for a few cycles by changing its *statefactor*, the soma life cycle chromosome (SL) is run, and the firing potential is sent to the other neurons by running the AS programs in axon branches. AS updates neighbouring dendrite branch potentials and the axo-synaptic potential. The *statefactor* of the axosynaptic branch is also updated. If the axo-synaptic branch is active its life cycle program (ASL) is executed.

After this the weight processing program (WP) is run which updates the *Weights* of neighbouring (branches sharing same grid square) branches.

Life Cycle of Neuron

This part is responsible for replication, death, growth and migration of neurite branches. It consists of three life cycle chromosomes responsible for the neurites development. The two branch chromosomes update *Resistance* and *Health* of the branch. Change in *Resistance* of a neurite branch is used to decide whether it will grow, shrink, or stay at its current location. The updated value of neurite branch *Health* decides whether to produce offspring, to die, or remain as it was with an updated *Health* value. If the updated *Health* is above a certain threshold it is allowed to produce offspring and if below certain threshold, it is removed from the neurite. Producing offspring results in a new branch at the same grid square connected to the same neurite (axon or dendrite). The soma life cycle chromosome produces updated values of *Health* and *Weight* of the soma as output.

The Game of Checkers

Throughout the history of AI research, building computer programs that play games has been considered a worthwhile objective. Shannon developed the idea of using a game tree of a certain depth and advocated using a *board evaluation function* (Shannon (1950)) that allocates a numerical score according to how good a board position is for a player. The method for determining the best moves from these is called minimax (Dimand and Dimand (1996)). Samuel used this in his seminal paper on computer checkers (Samuel (1959)) in which he refined a board evaluation function. The current world champion at checkers is a computer program called Chinook (Schaeffer (1996)), which uses deep minimax search, a huge database of end game positions and a handcrafted board evaluation function based on human expertise.

More recently, board evaluations functions for various games including Checkers have been obtained through Artificial Neural Networks (ANNs) and often evolutionary techniques have been used to adjust the weights (Chellapilla and Fogel (2001)).

Although the history of research in computers playing games is full of highly effective methods (e.g. minimax, board evaluation function), it is highly arguable that human beings use such methods. Typically they consider relatively few potential board positions and evaluate the favourability of these boards in a highly intuitive and heuristic manner. They usually learn during a game, indeed this is how, generally, humans learn to be good at any game. So the question arises: How is this possible? In our work we are interested in how an *ability to learn* can arise and be encoded in a genotype that *when executed* gives rise to a neural structure that can play a game well.

Experimental Setup

The experiment is organized such that an agent is provided with CGPDN as its computational network. It is allowed to play against a minimax based checker program (MCP). The initial population of five agents, each starting with a small randomly generated initial network and randomly generated genotypes. The genotype corresponding to the agent with the highest fitness at the end of the game is selected as the parent for the new population. Four offspring formed by mutating the parent are created. Any learning behaviour that is acquired by an agent is obtained through the interaction and repeated running of program encoded by the seven chromosomes within the game scenario.

The MCP always plays the first move. The updated board is then applied to an agent's CGPDN. The potentials representing the state of the board are applied to CGPDN using the axo-synapse(AS) chromosome. The agent CGPDN is run which decide about its move. The game continues until it is stopped. It is stopped if all its or opponent players are taken, or if the agent or its opponent can not move anymore, or if the allotted number of moves allowed for the game have been taken.

Inputs and outputs of the System

Input is in the form of board values, which is an array of 32 elements, with each representing a playable board square. Each of the 32 inputs represents one of the following five different values depending on what is on the square of the board (represented by I). Zero means empty square. $I = M = 2^{32} - 1$ means a king, (3/4)M means a piece, (1/2)M an opposing piece and (1/4)M an opposing king.

The board inputs are applied in pairs to all the sixteen locations in the 4x4 CGPDN grid (i.e. two input axo-synapse branches in every grid square, one axo-synapse branch for each playable position) as the number of playable board positions are 32 as shown in figure 2. Figure 2 shows how the CGPDN is interfaced with the game board, input axo-synapse branches are allocated for each playable board position. These inputs run programs encoded in the axo-synapse electrical chromosome to provide input into CGPDN (i.e. the axo-synapse CGP updates the potential of neighbouring dendrite branches).

Input potentials of the two board positions and the neighbouring dendrite branches are applied to the axo-synapse chromosome. This chromosome produces the updated values of the dendrite branches in that particular CGPDN grid square. In each CGPDN grid square there are two branches for two board positions. The axo-synapse chromosome is run for each square one by one, starting from square one and finishing at sixteenth.

Output is in two forms, one of the outputs is used to select the piece to move, and second is used to decide where that piece should move. Each piece on the board has an output dendrite branch in the CGPDN grid. All pieces are assigned

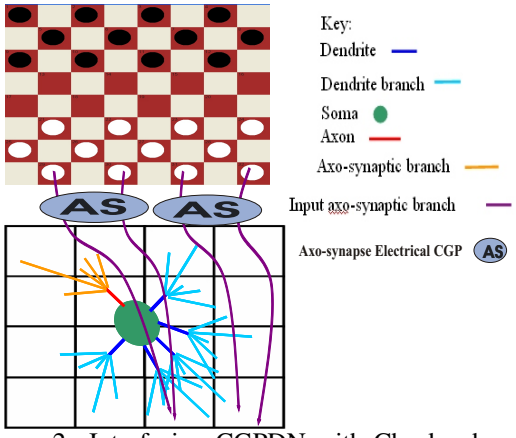


Figure 2: Interfacing CGPDN with Checker board. Four board positions are interfaced with the CGPDN such that board positions are applied in pair per square of CGPDN.

a unique ID, representing the CGPDN grid square where its branch is located. So the twelve pieces of each player are located at the first twelve grid squares. The player can only see its pieces, while processing a move and vice versa. Also the location of output dendrite branch does not change when a piece is moved, the ID of the piece represent the branch location not the piece location. Each of these branches has a potential, which is updated during CGPDN processing. The values of potentials determine the possibility of a piece to move, the piece that has the highest potential will be the one that is moved, however if any pieces are in a position to jump then the piece with the highest potential of those will move. In addition, there are also five output dendrite branches distributed at random locations in the CGPDN grid. The average value of these branch potentials determine the direction of movement for the piece. Whenever a piece is removed its dendrite branch is removed from the CGPDN grid.

CGP Developmental Neuron (CGPDN) Setup

The experiment parameters are arranged as follows. Each player CGPDN has a neuron with branches located in a 4x4 grid. Maximum number of dendrites is 5. Maximum number of dendrite are 200 and axon branches is 50. Maximum branch *statefactor* is 7. Maximum soma *statefactor* is 3. Mutation rate is 2%. Maximum number of nodes per chromosome is 200. Maximum number of moves is 20 for each player.

Fitness Calculation

The fitness of each agent is calculated at the end of the game using the following equation:

$$Fitness = A + 200(K_P - K_O) + 100(M_P - M_O) + N_M,$$

Where K_P represents the number of kings, and M_P represents number of men (normal pieces) of the player. K_O and M_O represent the number of kings and men of the opposing player. N_M represents the total number of moves

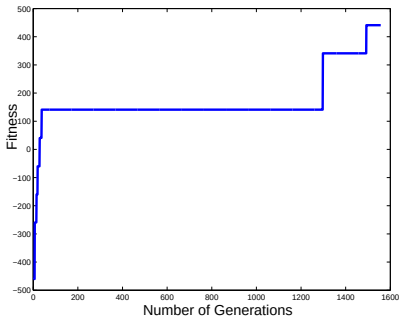


Figure 3: Fitness of CGPDN based player against MCP

played. A is 1000 for a win, and zero for a draw. To avoid spending much computational time assessing the abilities of poor game playing agents we have chosen a maximum number of moves. If this number of moves is reached before either of the agents win the game, then A =0, and the number of pieces and type of pieces decide the fitness value of the agent.

Results and Analysis

We have evolved agents against MCP in a number of evolutionary runs for 1500 generations and plotted it in figure 3. From the fitness graph, it is evident that the agent plays poorly at the early stage of evolution, but as the evolution progresses, the agent starts playing increasingly better and after 1250 generations, it begins to beat the opponent by three and four pieces margin. MCP is using minimax at ply level of 5. Agent plays with different strategy every time and finally manages to beat the opponent. It is worth mentioning here that the agent does not have any clue of what it is doing. It just receives signals from the board and produce moves accordingly, but as evolution progresses, the agent begins to understand the board and plays better. This is evident from the fitness graph shown in figure 3. Keeping in view that the agent is using a single neuron as a computational system and still manages to beat a program based on human (having trillion of neurons) intelligence is a big achievement demonstrated by any learning developmental system to date. Table 1 shows a game played between the well evolved agent and MCP. This is presented to demonstrate the level of play that the two players play. Figure 5 shows various stages of the game along with the corresponding neuron structure updated as a result of game scenario. Figure 4 shows the variation in the number of axon and dendrite branches of the CGP neuron during the game. Table 1 and figure 5 shows the complete game, the game start with black (MCP) making the first move by forwarding its piece from square 12 to 15. The updated board is applied as input to the CGPDN causing white(CGPDN) to forward a piece from square 21 to 17 as a result of signal received from CGPDN to motor neuron. Motor neuron receive signal using virtual dendrite branches distributed in the CGPDN Grid. Initially neuron has a small branching structure as evident from the first neuron image in figure 5 (Row-2, Column-1). Mutual exchange of pieces occur at various stages of the game and the neural branching structure continue to develop. The important break through occurs when black make a blunder at move-31 causing white to not only take two black pieces in one move but also becoming a king so that it can move both in forward and backward direction. Figure 5 show the move on the third row and last column. At this stage the CGPDN has the maximum dendrite branching structure so it can sense the signal from the board through its branches and act accordingly as evident from figure 5 and figure 4. The game continue until the aloted number of moves (40) are taken with white (CGPDN) having one king and a piece advantage over black(MCP).

Black Move	White Move
B1 12 - 15	W2 21 - 17
B3 10 - 13	W4 17 - 10
B5 5 - 14	W6 23 - 20
B7 1 - 5	W8 25 - 21
B9 14 - 19	W10 29 - 25
B11 5 - 10	W12 20 - 16
B13 10 - 13	W14 28 - 23
B15 19 - 28	W16 32 - 23
B17 13 - 17	W18 16 - 12
B19 7 - 16	W20 23 - 19
B21 15 - 20	W22 24 - 15
B23 11 - 20	W24 22 - 18
B25 8 - 12	W26 26 - 22
B27 17 - 26	W28 30 - 21
B29 9 - 13	W30 18 - 9
B31 2 - 5	W32 9 - 2 W33 2 - 11
B34 20 - 23	W35 27 - 20
B36 16 - 23	W37 22 - 18
B38 12 - 16	W39 11 - 14
B40 16 - 20	W41 19 - 15

Table 1: The first 41 moves of a game between a high evolved player (white) against MCP(black)

Generality

In order to test the generalization property of the agent, we have conducted a number of experiments by allowing the agent to play against five different opponents with various

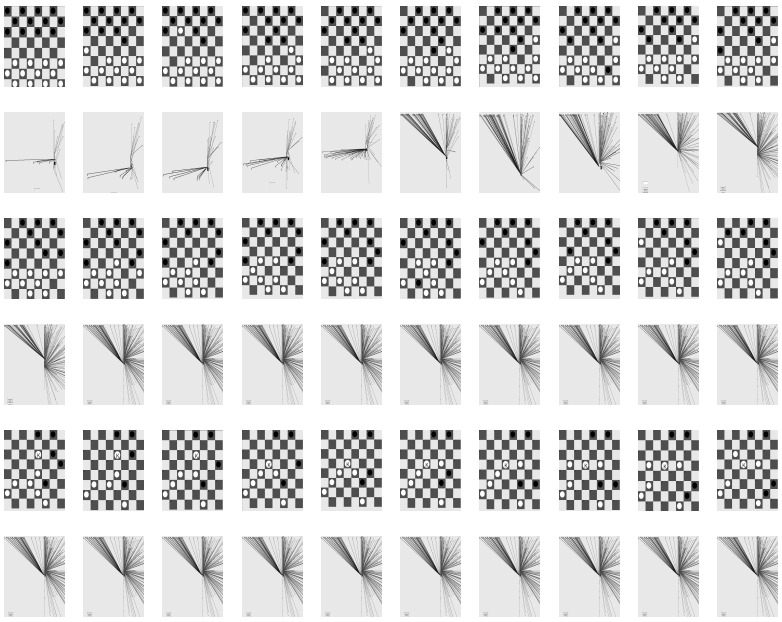


Figure 5: Various move played by CGP Neuron based agent and MCP with Agent playing white and MCP Black. Figure also shows the variation in neural structure during the game at various stages

Game Number	Winning Margin of CGPDN Agent	Level of opponent	Number of Moves to win
1	2 MEN and 1 King	50	76
2	2 MAN and 1 King	100	83
3	1 King	1000	111
4	1 MAN and 1 King	1200	120
5	lost by 1King and 4Men	1300	59

Table 2: Results of Evolved agent against various opponents not seen during evolution

playing levels. The neuron inside the agent starts with a random branching structure with the evolved genotype and continues to develop during the game. The agent was playing against completely new opponents that he has never played before during the course of evolution. Opponent's level of play is evident by the number of generations for which it is evolved. It beats the 50th generation agent by one King and two Men(normal peices) within 76 moves. An agent evolved for 100 generations also by one King and two Men but in 83 moves, the 1000th generation agent by one King in 111 moves and finally the 1200th generation by one Man and a King in 120 moves. In final case, the agent lost the game to a 1300 generations evolved player by one King and 4 Men in 59 Moves. It is worth mentioning that the agent was trained (evolved) to play forty moves. It never played a game beyond forty moves during evolution. From the re-

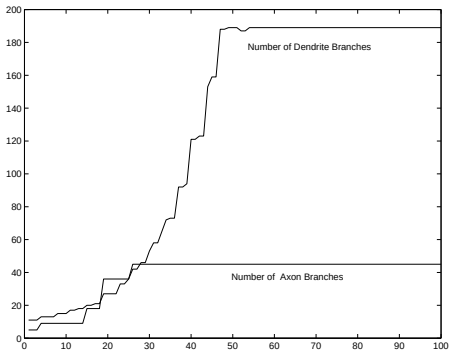


Figure 4: Variations in the number of Dendrite and Axon branches during the game

sults shown in table 2 it is evident that as the level of play of the opponent increases, the winning margin decreases, thus demonstrating clearly that we are able to obtain a DNA using CGP such that when used inside neuron produce a structure that can play game intelligently.

Conclusion

We have investigated the evolution of checkers playing agents that are controlled by a single developmental neuron. The development and signal processing inside neuron is controlled by a number of CGP programs working as DNA of the agent. The branching structure of neuron develops during the course of game. The agent demonstrated that it can play intelligently and beat a human intelligence based agent by a large margin. We have also tested a single neuron based agent for its generality. It beated the low level players with big margins in lesser time and tends to have problems beating high level players. From the results, it appears that we have successfully evolved CGP programs that encode *an ability* to learn 'how to play' checkers. In the future, we are planning to run the programs for longer, and against high level professional checkers agents to have more experience.

References

Cangelosi, A., Nolfi, S., and Parisi, D. (1994). Cell division and migration in a 'genotype' for neural networks. *Network-Computation in Neural Systems*, 5:497–515.

Chellapilla, K. and Fogel, D. B. (2001). Evolving an expert checkers playing program without using human expertise. In *IEEE Trans. on Evolutionary Computation*, volume 5, pages 422–428.

Dimand, R. W. and Dimand, M. A. (1996). A history of game theory: From the beginnings to 1945. *Routledge,Urbana*, 1.

Federici, D. (2005). Evolving developing spiking neural networks. In *Proceedings of CEC 2005 IEEE Congress on Evolutionary Computation*, pages 543–550.

Gruau, F. (1994). Automatic definition of modular neural networks. *Adaptive Behaviour*, 3:151–183.

Kandel, E. R., Schwartz, J. H., and Jessell, T. (2000). *Principles of Neural Science, 4rth Edition*. McGraw-Hill.

Kleim, J., Napper, R., Swain, R., Armstrong, K., Jones, T., and Greenough, W. (1998). Selective synaptic plasticity in the cerebellar cortex of the rat following complex motor learning. *Neurobiol. Learn. Mem.*, 69:274–289.

Miller, J. F. and Thomson, P. (2000). Cartesian genetic programming. In *Proc. of the 3rd European Conf. on Genetic Programming*, volume 1802, pages 121–132.

Nolfi, S., Miglino, O., and Parisi, D. (1994). Phenotypic plasticity in evolving neural networks. In *Proc. Int. Conf. from perception to action*. IEEE Press.

Roggen, D., Federici, D., and Floreano, D. (2007). Evolutionary morphogenesis for multi-cellular systems. *Journal of Genetic Programming and Evolvable Machines*, 8:61–96.

Rust, A. G., Adams, R., George, S., and Bolouri, H. (1997). Activity-based pruning in developmental artificial neural networks. In *Proc. of the European Conf. on Artificial Life (ECAL'97)*, pages 224–233. MIT Press.

Samuel, A. (1959). Some studies in machine learning using the game of checkers. *IBM J. Res. Dev.*, 3(3):210–219.

Schaeffer, J. (1996). *One Jump Ahead: Challenging Human Supremacy in Checkers*. Springer, Berlin.

Shannon, C. (1950). Programming a computer for playing chess. *Phil. Mag.*, 41:256–275.

Shepherd, G. (1990). *The synaptic organization of the brain*. Oxford Press.

Terje, L. (2003). The discovery of long-term potentiation. *Philos Trans R Soc Lond B Biol Sci*, 358(1432):617–20.

Van Ooyen, A. and Pelt, J. (1994). Activity-dependent outgrowth of neurons and overshoot phenomena in developing neural networks. *Journal of Theoretical Biology*, 167:27–43.

Zubler, F. and Douglas, R. (2009). A framework for modeling the growth and development of neurons and networks. *Frontiers in Computational Neuroscience*, 3.

Emergent Generalization in Bayesian Agents Using Iterated Learning

Giancarlo Schrementi and Michael Gasser

School of Informatics and Computing
Indiana University, Bloomington, IN 47405
gischrem@cs.indiana.edu

Abstract

The compositional nature of human language is a remarkable adaptation that solves the problem of generalizing our communications to novel experiences. The Iterated Learning Model of agent interaction has proven to be a useful tool for exploring the emergence of this phenomenon of generalization. Recently, a Bayesian interpretation of this model has been proposed and analyzed in the literature. The work here combines the Bayesian approach with the traditional goal of iterated learning, the emergence of compositional communication. Two methods of measuring language likelihood are investigated, one based on agent comprehension and the other on production scope. Calculating likelihood based on agent comprehension is shown to result in the emergence of significantly better generalization. The beneficial effect of a description-length based prior probability is also demonstrated.

Introduction

The ability to generalize our knowledge to novel experiences is a fundamental capability of the human mind. Nowhere has this faculty had more impact than on how we communicate. Our languages have developed to be massively compositional. As children, we learn a set of components and rules for combining those components in a way that allows us to express an infinite number of utterances. Likewise, we can understand those utterances by breaking them down into their components and rules. Thus the compositional nature of our languages has given us tremendous ability to generalize our communications.

In this paper, we look at how this compositional nature emerges through communicative interactions between agents that are finite-state transducers. In order to model these interactions, we use the Iterated Learning Model (ILM) of Kirby and colleagues (Kirby, 2001; Smith et al., 2003). ILM originated as way to model this kind of language emergence and evolution, but has since been used as a more general model of knowledge change in domains with a teacher and a learner (Kalish et al., 2007).

Iterated learning can involve many agents, but in its purest form involves a single teacher and a single learner. Initially, the teacher agent imparts some of its knowledge to a learner. Since the teacher is not revealing all of its knowledge, the learner must fill in the blanks according to some inference algorithm. Typically, the inference algorithm looks at the knowledge the learner already has and infers from that. The learner then becomes a teacher and instructs a new learner in the same fashion and this continues for many iterations. Eventually this process of knowledge transfer and self-organization converges to an equilibrium in a manner similar to the transfer and self-organization of genetic information in an artificial life simulation.

Language evolution models usually operate with a space of idealized meanings that agents need to communicate to each other. These meanings take the form of vectors of features, each having some range of values. The agents then turn these meanings into some form of signal, creating a meaning-signal mapping. In iterated learning models, the agents can be broadly defined to fall into two categories.

The first type of agent we will call grammatical inducers. These grammatical inducers keep track of any correlations between features in the meaning space and the received signals. These correlations are kept track of with a context-free grammar, neural network, or matrix. The agents induce a signal for a novel meaning by making use of any noticed correlations between the features of the meaning and portions of earlier signals. Those correlations are typically combined with a randomly generated signal portion that represents the rest of the uncorrelated features to create a final signal for the novel meaning. The success of these agents is judged by how compositional their signals are after a number of generations. Originally this was measured through subjective analysis of the signals (Batali, 1998), but more recently is often measured by $expressivity$ (Kirby, 2007; Brighton, 2005). Expressivity is defined as the number of meanings that can be distinctly expressed.

The second type of agent is the more recent Bayesian agent that was analyzed in detail by Griffiths and Kalish (2005, 2007). Griffiths recognized that the learner in ILM is essentially using a form of Bayesian inference to infer the language from the teacher's instruction. The learner considers many hypotheses about the language before picking

the one that it feels is most probable. The probability of a hypothesis is calculated based on how closely a hypothesis matches the data from the teacher, the likelihood, and by the agent's inductive biases, the prior. This relationship allows iterated learning to be formulated as a mathematical process that can be rigorously analyzed. One of the results of Griffiths' analysis is that over generations of iterated learning the posterior probability distribution converges to the prior probability distribution. Essentially, the languages the inductive biases favor are the languages that will emerge over the course of the process.

The convergence of the Bayesian agent form of ILM has been rigorously analyzed (Rafferty et al., 2009; Ferdinand and Zuidema, 2009). However, these studies have used arbitrary priors and were not looking for evidence of compositionality in agent signals. The work here combines the goals of the grammatical inducers with the method of the Bayesian inducers. To do this we need to characterize what information our prior is to use and how to calculate likelihood.

Bayesian inference, Equation (1), has long been known to be related to the mathematical model selection criterion of Rissanen (1978) called the Minimum Description Length Principle (MDL) and the closely related Minimum Message Length (MML) measurement of Wallace and Boulton (1968). A detailed discussion of this relationship is in Vitanyi and Li (2000), but we will discuss the nature of the correspondence here.

$$P(Model|Data) = \frac{P(Data|Model)P(Model)}{P(Data)} \quad (1)$$

Both MDL and MML measure the success of a mathematical model of data. A successful model is one that is simple and compactly expresses the data. By combining a measure of the size of the model and a measure of the size of the data as encoded by the model the total information load can be quantified. The essence of the relationship with Bayesian inference is that the amount of information can be viewed as the amount of Shannon entropy. A higher information load corresponds to a model with lower posterior probability, $P(Model|Data)$. The relationship extends to the two primary components of Bayesian inference, the likelihood and the prior. The likelihood, $P(Data|Model)$, corresponds to the size of the data as encoded by the model and the prior, $P(Model)$, corresponds to the complexity of the model.

The selective pressures of minimizing description length on a model are not very different from the selective pressures on a language. Language is a model that uses syntax to represent semantics. A successful language is one that can express everything we want to talk about but is also simple to learn and use. This correspondence provides us with a way to formulate the Bayesian inference components of our agents. The likelihood needs to measure how successful we are at expressing ourselves and the prior needs to measure how simple our manner of expression is.

This is not the first time MDL is used as a way to encourage to the emergence of generalization without directly selecting for it. Schrementi and Gasser (2010) used it as a fitness metric for a genetic algorithm. Brighton (2005, 2003, 2002) used description length as a hypothesis selection measure in an iterated learning model that used a modified form of transducers called finite-state unification transducers. Brighton's work was not specifically Bayesian and stayed close to the original formulation of the likelihood in MDL; that likelihood was the size of the data as encoded by the model. The focus of the work here is to investigate likelihood as a measure of the probability that a signal can be decoded to its original meaning. We investigate two methods of formulating likelihood as a probability, one based on expressivity and the other comprehension.

Iterated Learning Framework

Our implementation of the iterated learning model uses agents that are simple finite-state transducers. These transducers sequentially process input strings and encode them into output strings. Each edge between states in the transducer reads in an input character and writes an output character. This encoding process provides a simple way to model linguistic production, the translation of meaning into signal.

Notably, the same transducer can be used for the other half of a linguistic interaction, comprehension, by reversing what is read and what is written for each edge. This inverted transducer will be able to translate the output strings back into the original input strings, with an important caveat. The inversion process can introduce ambiguity in the transducer that didn't exist before. A state that has two edges leaving it that output the same character will after inversion have two edges leaving it that read the same character. This ambiguity results in a non-deterministic transducer that can have multiple paths that read the same input string.

The algorithm starts with a state-minimal finite-state automaton that recognizes the entire set of input training strings. A transducer recognizes a string if it finishes in an end state after reading the string. A state-minimal transducer is one that has been compressed to have the fewest states needed to recognize the input set and only the input set. Each edge in the automaton is then randomly assigned to write one of the output characters. This transducer is the first teacher in the iterated learning process. The learner starts out as an empty transducer, with just a start state and an end state.

The learning process begins with the teacher going through a random selection of the input training strings and producing an input-output pair. The learner adds each of these input-output pairs to its transducer, such that there is a path from the start state to the end state that reads the input string and writes the output string. Any remaining input

training strings are added to the learner but not paired with any output. The learner's transducer is then compressed to be state-minimal. This results in a learner that has the same transducer structure as the teacher but some of the edges may not write anything.

The edges that have no output form the basis for the invention part of the iterated learning model. Invention refers to the process of inferring outputs for inputs which were not presented to the agents as input-output pairs by their teacher. Our invention method uses Bayesian inference to select the output characters for the edges that lack them. The set of sets of possible outputs to fill in the blanks forms a search space whose size is determined by the number of blanks, n, and the size of the output character space. For each of the experiments in this paper, there are two possible output characters, resulting in a space of 2^n.

Each set of output characters in the search space is a hypothesis of the optimal language. This hypothesis coupled with the learned transducer completely specifies all the input-output mappings of the agent for the training strings. The transducer can now be further compressed following a compression criterion from Brighton (2002). The criterion is that any two states can be combined if the change doesn't affect the input-output mappings of the training strings. We have added two additional criteria. The first is that the two states don't have conflicting output edges, e.g. two edges reading the same character but writing a different character, which prevents production ambiguity. The second is that the two states to be combined must also be at the same depth from the initial state, in order to prevent cycles and to allow the compression to be done iteratively.

The further compressed transducer now recognizes and encodes additional strings beyond those that it was trained on. In essence, this compression allow the transducer to generalize its knowledge about the training set to a wider range of input strings. Each hypothesis results in a transducer that can be compressed in this way to different degrees. The size of this compressed transducer will form the basis of our calculation of the prior probability of a hypothesis. Additionally, we can now measure how well a given language, as specified by the transducer, generalizes to novel test strings.

The posterior probability of each hypothesis in the search space is calculated according to our formulation of its prior probability and likelihood, the specifics of which are discussed in the next section. The set of output characters with the highest posterior probability is selected by the learner to fill in its blanks. In case of a tie, the set that is closer to the teacher's edge outputs is chosen. After the learner completes this inference process, it is ready to become a teacher. A new learner agent is created and the cycle repeats with the old learner as the new teacher. This process continues for a set number of generations.

Bayesian Inference Formulation

Bayesian inference has two primary components, the prior probability of a hypothesis, and the likelihood of the hypothesis given the data. There is also a third component, the marginal probability of the data. However, this component is constant and in the interest of simplification we will drop it in our calculations.

Our investigation of methods of calculating likelihood looks at three different measures. The first is a control likelihood that is always one, Equation (2). The second is a likelihood measure based on expressivity. Expressivity makes a plausible likelihood measure because the more distinct signals a hypothesized transducer is able to make the more likely that its signals can be decoded back into the correct meaning. Our measurement of expressivity looks at the list of output strings produced for the training input strings and simply divides the number of different strings by the total number of strings, Equation (3).

The third likelihood calculation is based on comprehension; how likely a transducer is able to decode, when reversed, its encodings of the training set. A hypothesis that results in a transducer that has this internal consistency is considered more likely. Essentially, an agent checks whether a hypothesized language allows the agent to talk to itself as in Mirolli and Parisi (2006). The likelihood for a given input-output mapping is calculated by counting the number of paths through the reversed transducer that read the output characters and write the correct input characters divided by the total number of paths that read the output characters. The likelihood is never zero because there is always at least one path that will write the correct characters. The final likelihood for the hypothesis is the average over all of the input-output mappings drawn from the input training strings. Equation (4) shows this calculation, with R being the set of training strings and $|R|$ the size of the training set. Each input training string is equally likely, so the average is not weighted.

$$P(H|D) = 1 \quad (2)$$

$$P(H|D) = \frac{DifferentOutputs}{TotalOutputs} \quad (3)$$

$$P(H|D) = \frac{\sum_{s \in R} \frac{SuccessfulDecodingPaths_s}{TotalDecodingPaths_s}}{|R|} \quad (4)$$

$$DL(Transducer) = N_{Edges} * (2 * \lceil log_2(N_{States}) \rceil) \quad (5)$$

Our prior calculation weights hypotheses by how much the resulting transducer can be compressed. The size of the transducer is measured as description length in bits by calculating the cost of storing each edge based on the number of states, Equation (5). The compressed size, DL_c, is compared to the size of the transducer before compression,

DL_u. Equation (6) shows the formula that calculates the prior such that the more a transducer is able to be compressed the higher the probability. $DL_u + 1$ is used in the calculation to ensure that the prior is never zero. A second control prior that is always one is also used, Equation (7).

$$P(H) = \frac{(DL_u + 1) - DL_c}{DL_u + 1} \quad (6)$$

$$P(H) = 1 \quad (7)$$

Results

We demonstrate the results of two experiments that investigate the generalization performance of the likelihood and prior measures. For each experiment, the input and output alphabets are both of size two. The length of every input string is 8 and consequently the length of every output string is 8. Each experiment has a training set of a specified size and the test set is all 256 strings of length 8, so the training set is a subset of the test set. Generalization performance is measured using the expressivity metric across the entire test set, rather than just the training set as it is used in the learning process.

Experiment One

The first experiment uses a training set of 16 input strings with one of the strings randomly chosen each generation to not have its corresponding output conveyed to the learner. This results in average of 3.3 blanks, with a standard deviation of 2.2, to be inferred by the learner out of total of 53.74 edges on average. The results shown here are the average expressivity across 50 trials each with a different randomly chosen training set. The experiment runs for 200 generations of teacher-learner interactions.

Figure 1 shows a plot of the expressivity over time, with standard deviation bars, using the description-length prior and each of the three likelihood measures: flat, expressivity-based and comprehension-based. We see that all three measures start with similar levels of expressivity but the comprehension measure quickly jumps ahead of the other two measures. It continues this rapid ascent before plateauing at slightly over 90% expressivity. The expressivity-based measure also ascends but much more slowly and settles in slightly above 26%. This isn't bad considering that the training set is only 6.25% of the test set, but it falls well short of success of the comprehension-based measure. The flat measure establishes a baseline that ends around 15%.

Figure 2 compares the expressivity when using the description-length prior versus the flat prior under the comprehension-based likelihood. The description-length prior results in clear improvement in expressivity. But, the flat prior turns in a respectable performance that ends at almost 70%.

Experiment One shows that agents that try to maximize comprehension are much better at generalizing than agents

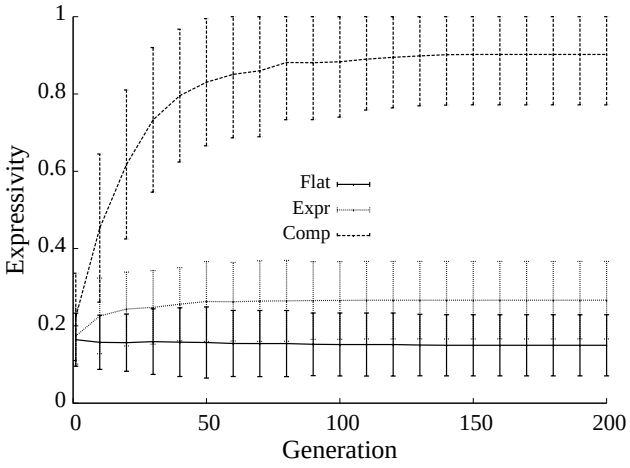


Figure 1: Likelihoods, 16 Training Strings

that try to maximize expressivity. The results from the analysis of the priors indicate that seeking to maximize compression in addition to maximizing comprehension results in even better generalization. The verdict on expressivity as a likelihood measure doesn't look good, but we want to make sure that the small training set isn't setting up expressivity to fail.

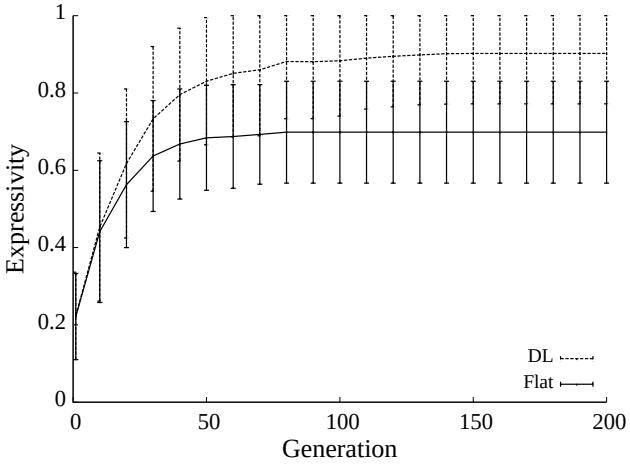


Figure 2: Priors, 16 Training Strings

Experiment Two

The second experiment uses a training set of 64 input strings, four times larger than the first experiment. Again, one of the strings is randomly chosen each generation to not have its corresponding output conveyed to the learner. The results shown here are the average expressivity across 50 trials each with a different randomly chosen training set.

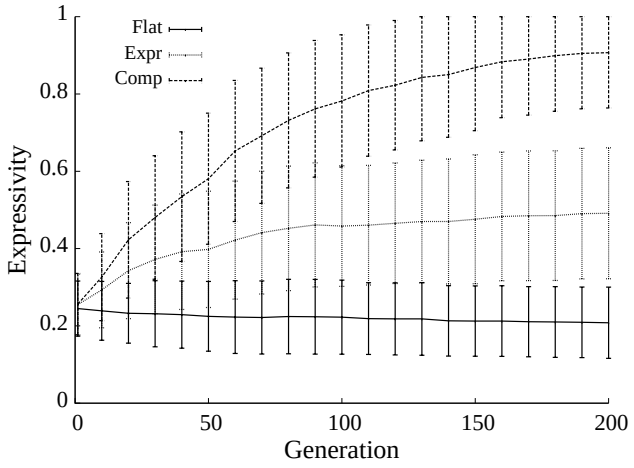


Figure 3: Likelihoods, 64 Training Strings

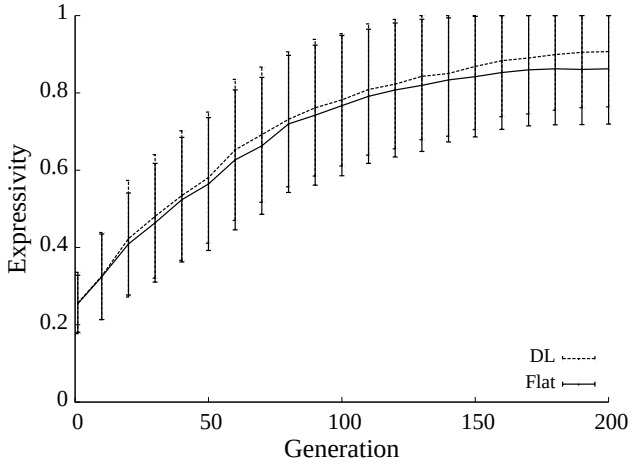


Figure 4: Priors, 64 Training Strings

Figure 3 shows the plot of the three likelihood measures. The comprehension-based measure is at the top ending again at slightly over 90%. The expressivity-based measure gets a boost from the larger training size and reaches slightly below 50%. The flat likelihood doesn't do much better than before settling in at around 20%. Once again, maximizing comprehension results in significantly better generalization than trying to maximize expressivity.

The analysis of the priors, using the comprehension-based likelihood, for Experiment Two is in Figure 4. Interestingly, we see that the that there isn't a significant benefit to maximizing compression with the larger training set. The training set is now large enough that seeking to maximize likelihood is sufficient to achieve high expressivity.

Conclusions

The capability to generalize is the hallmark of a compositional system. The Bayesian agents' ability to generalize their encodings to novel strings means that their communications are compositional. From the low expressivity at the start we can see that the compositionality emerges during training.

The success of the comprehension-based likelihood measure over the expressivity-based one demonstrates the value of including comprehension in the process. It is not sufficient to concentrate just on production and how many signals an agent can make. The pressure of being forced to actually decode those signals back into meanings is necessary to drive the emergence of a generalizable grammar.

The benefit of the description-length prior reaffirms the value of simplicity-based metrics like MDL. The added pressure to compress the grammar allowed the agents to express a large majority of the test set even with a very small training set. However, the prior's value decreases as the agents access more information. Large training sets mean that prior knowledge is no longer necessary to master the test set.

The iterated learning model again proves to be a powerful method of modeling the emergence of compositional grammars. The Bayesian version provides us with new ways of analyzing the process with the clear delineation of the role of the prior and the likelihood. The experiments here show that choosing a successful likelihood measure is not as simple as it might seem. A metric like expressivity seems like a good candidate but turns out to be rather poor. Likewise, the prior should be carefully chosen; a good prior can make the difference when knowledge is scarce. Finding two that work together, in this case the likelihood's pressure to be comprehensible and the prior's pressure to be simple, is the key to successful Bayesian inference and might be the key to our ability to generalize as well.

References

Batali, J. (1998). Computational simulations of the emergence of grammar. In Hurford, J., Studdert-Kennedy, M., and Knight, C., editors, *Approaches to the Evolution of Language: Social and Cognitive Bases*, pages 405–426. Cambridge University Press, Cambridge.

Brighton, H. (2002). Compositional syntax from cultural transmission. *Artificial Life*, 8(1):25–54.

Brighton, H. (2003). *Simplicity as a Driving Force in Linguistic Evolution*. PhD thesis, Theoretical and Applied Linguistics, The University of Edinburgh.

Brighton, H. (2005). Linguistic evolution and induction by minimum description length. In Werning, M. and Machery, E., editors, *The Compostionality of Concepts and Meanings: Applications to Linguistics, Psychology and Neuroscience*, pages 405–426. Ontos Verlag, Frankfurt.

Ferdinand, V. and Zuidema, W. (2009). Thomas' theorem meets bayes' rule: a model of the iterated learning of language. In *Proceedings of the 31th Annual Conference of the Cognitive Science Society*.

Griffiths, T. L. and Kalish, M. L. (2005). A bayesian view of language evolution by iterated learning. In *Proceedings of the 27th Annual Conference of the Cognitive Science Society*.

Griffiths, T. L. and Kalish, M. L. (2007). Language evolution by iterated learning with bayesian agents. *Cognitive Science*, 31(3):441–480.

Kalish, M. L., Griffiths, T. L., and Lewandowsky, S. (2007). Iterated learning: Intergenerational knowledge transmission reveals inductive biases. *Psychonomic Bulletin and Review*, 14(2):281–294.

Kirby, S. (2001). Spontaneous evolution of linguistic structure: an iterated learning model of the emergence of regularity and irregularity. *IEEE Transactions on Evolutionary Computation*, 5(2):102–110.

Kirby, S. (2007). The evolution of meaning-space structure through iterated learning. In Lyon, C., Nehaniv, C., and Cangelosi, A., editors, *Emergence of Communication and Language*, pages 253–268. Springer Verlag.

Mirolli, M. and Parisi, D. (2006). Talking to oneself as a selective pressure for the emergence of language. In *Proceedings of the 6th International Conference on the Evolution of Language*, pages 214–221.

Rafferty, A., Griffiths, T., and Klein, D. (2009). Convergence bounds for language evolution by iterated learning. In *Proceedings of the 31th Annual Conference of the Cognitive Science Society*.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, 14:465–471.

Schrementi, G. and Gasser, M. (Forthcoming, 2010). Minimum description length and generalization in the evolution of language. In *Proceedings of the 8th International Conference on the Evolution of Language*.

Smith, K., Kirby, S., and Brighton, H. (2003). Iterated learning: a framework for the emergence of language. *Artificial Life*, 9(4):371–386.

Vitanyi, P. M. B. and Li, M. (2000). Minimum description length induction, bayesianism and kolmogorov complexity. *IEEE Transactions on Information Theory*, 46(2):446–464.

Wallace, C. S. and Boulton, D. M. (1968). An information measure for classification. *Computer Journal*, 11(2):185–194.

Complex Taxis-Behavior in a Novel Bio-Inspired Robot Controller

Thomas Schmickl, Heiko Hamann, Jürgen Stradner, Ralf Mayet, and Karl Crailsheim
Artificial Life Lab of the Department of Zoology, Karl-Franzens University Graz, Universitätsplatz 2, A-8010 Graz, Austria
thomas.schmickl@uni-graz.at

Abstract

In swarm robotics, robots with only poor computational equipment are often used. Additionally, the precision of their actuators and sensors is rather poor. This causes a challenge in the construction of controllers able to achieve complex behaviors on such robotic systems. Here we describe a novel bio-inspired concept of a robot control paradigm, which is inspired by the information-processing of simple microorganisms. The basic idea is that we use a roughly abstracted model of inter-cellular signal emission and signal processing to control the movement behavior of a two-wheeled autonomous robot. Many unicellular organisms are able to perform taxis-behavior (phototaxis, chemotaxis, etc.) without having sophisticated sensor equipment and without possessing neuronal structures. Our Artificial Homeostatic Hormone System (AHHS) mimics primitive chemical signal networks and is able to achieve taxis-behavior with little computational cost. In this article the controller is analyzed in a simple mathematical model and additional tests are performed on a more sophisticated multi-agent simulation of robotic hardware and the controller is implemented on real robotic hardware.

INTRODUCTION

In swarm robotics (Beni, 2005; Şahin, 2005) simple and inexpensive robotic hardware is used frequently. Such robotic systems often have limited computational abilities and their sensors and actuators are rather imprecise. Also memory is often limited and therefore the minimal hardware equipment cannot easily be compensated by extensive software concepts such as data filtering, managing a world-model or by simultaneous localization and mapping (SLAM) of the environment. Thus, it is a challenging task to generate controllers that allow the generation of adaptable complex behaviors. In addition, evolutionary robotics (Floreano et al., 2008) is a concept to automatically design 'simple' robot controllers with algorithms of evolutionary computation, to explore the behavior space of the robots and to generate the desired behaviors.

Many microorganisms, that have only limited sensor precision and do not have neuronal systems to process information, show an impressive ability to perform complex and/or target-oriented behaviors (taxis). For example, a unicellular algae (Bound and Tollin, 1967) performs phototaxis with just one photo-sensitive eye-spot and just a single actuator (flagellum). Similar capabilities are found in many bacteria (Khan et al., 1995; Darnton et al., 2007). Also, multi-cellular aggregation (colonies) of simple cells are able to coordinate their joint motion to collectively approach the source of a stimulus (e.g., phototaxis in *Volvox*, Holmes (1903)).

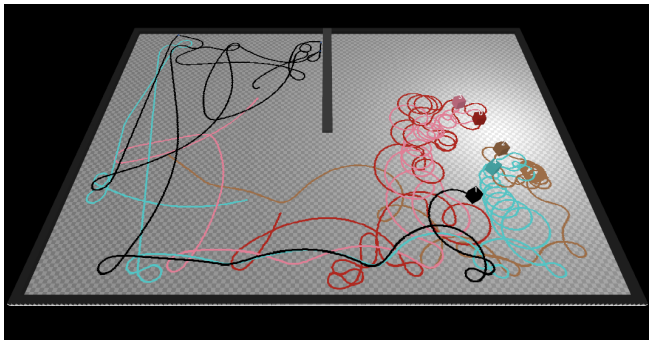


Figure 1: Five robots showing phototactic behavior with AHHS controller.

The internal processes of cells can be interpreted as computational processes as reported by Bray (2009). This 'non-cognitive' method (i.e., single cells have no neurons, hence, it is an anti-connectionist approach) of information processing was applied many decades ago by Grey Walter (Grey Walter, 1950, 1951) to control a simple robot. The behaviors reported in these papers are similar to this work, only we are modeling internal cell processes explicitly. In previous studies (Schmickl and Crailsheim, 2009; Hamann et al., 2010b,a), we suggested a simple difference-equation based model of the internal signal processing of uni-cellular organisms, which we call Artificial Homeostatic Hormone System (AHHS). In such systems, representing rough abstractions of biological physiological models, the difference-equation model controls the way of how a robot acts based on sensory input.

In the model we assume that the inner body of the robot is compartmentalized. Specific compartments are associated with certain components of the robot's real body. In

each compartment, the model tracks the dynamics of virtual chemical substances, which represent chemical cell signals in real organisms. These chemical substances can diffuse to neighboring compartments and they decay proportionally to their current concentration over time. Some of them are produced at constant rates as well, leading to homeostatic set points (equilibria) that are approached after an initial transient. Some of these signals affect actuators (e.g., wheels), leading to unstimulated behavioral modes of the organism. Sensor excitation by environmental stimuli result in local disturbances in the hormone equilibria, for example, by sudden secretion of one of the chemical signals (hormones). As hormone levels affect actuators, changing hormone concentrations may change the robot's behavior. This stimulated behavior lasts until the 'abnormal' sensor excitation has ceased and the hormone levels have approached the previous homeostatic settings again. A set of hormone-to-hormone interactions can enhance the behavioral repertoire of the robot by providing more complex forms of sensor-to-actuator linkage via the virtual hormone reaction networks.

To demonstrate the abilities of AHHS controllers in producing interesting behavioral patterns even with limited computational and with limited sensor equipment, we aimed to mimic taxis behavior that is found in very primitive lifeforms (e.g., some bacteria).

The bacterium *Esherichia coli* shows interesting behavior in finding attractive habitats by chemotaxis. The bacterium is propelled by several flagella (actuators), which have two modes of turning: clockwise (CW) and counter-clockwise (CCW). The CCW motion allows the organism to swim almost in straight trajectories and the CW motion of some flagella disturb the synchrony among the bundle of all flagella. This leads to a so called 'tumbling mode' of movement, where the organism almost randomly changes its direction (Khan et al., 1995; Darnton et al., 2007). Chemoreceptors that react to attractants in the environment suppress those cell-internal chemical signals which finally alter the rotation of flagella to the CW mode. In absence of these attractants, the CW mode is not suppressed that much, which leads to a higher probability and longer duration of the 'tumbling mode'.

This way, the organism is able to ascend in an attractive chemical gradient in a way that was found to be a very robust control mechanism (Alon et al., 1999; Yi et al., 2000). This approach of taxis is rather different from those approaches frequently used in mobile robotics, for example the famous Braitenberg vehicles (Braitenberg, 1984). For example, using just one single sensor is comparable to 'vehicle 1'. But the functionality of the taxis-behavior is not existent in 'vehicle 1', which rather speeds up or slows down depending on the current sensor intensity. When searching for the functionality of taxis, which is provided in our approach, a comparison with 'Braitenberg vehicles 2 and 3 (fear, aggression and love)' makes more sense. But here, the inner structure of the controller does not correspond. In contrast to these Braitenberg vehicles, our AHHS controller uses just one sensor, thus no gradient-ascent based on differences between parallel sensor values is used. Furthermore, there is no explicit implementation of any kind of 'seeking-behavior': Neither does the robot rotate with a directional sensor measuring light intensity until it finds a maximum in a particular direction that it then approaches directly, nor does it use any explicit memory storage of past sensor values or an explicit 'world model'. In contrast, we claim that in our solution, the robot, its position in the world (relative to the light optimum) and the trajectory itself serve as some kind of memory and as some kind of world model. This approach is rather unique.

In the study presented here, we investigate how an AHHS controller can be programmed to perform a comparably simple behavior with similar simple mechanisms. As most cheap robots are lacking real gas detectors (chemo-sensors) we wanted our focal robot to pursue a different but comparable task, that is phototaxis:

Our focal robot is equipped with two wheels and just one sensor on the right hand side of the robot. In this first controller example, this sensor is discrete and either passes a 1 (light perceived) or a 0 (no light perceived) to the controller. This 'binary' controller is able to detect whether it points towards the light or not, thus offering some directionality. In a second controller, we assumed that the sensor cannot determine this directionality, instead it can just report the local illuminance at the robot's current position. In contrast to the first controller, here the sensor reports a graduated output value proportionally to the current local illuminance. The task for the robot is to drive towards a light (phototaxis).

For a fixed topology with two wheels and a sensor on the right side of the robot, there are four potentially reasonable ways of programming a reactive agent: Without any sensory input the robot moves in right turns and sensory input either reduces the radius of the turns or it increases the radius. The other option is to let the robot move in left turns without sensory input and sensory input either reduces the radius of the turns or it increases the radius. The methods with standard right turns are gradient descends and the left turns lead to gradient ascends. The method of decreasing the turn radii leads to trajectories with many loops. We call this method *positive steering* because the robot steers by intensifying the standard turn direction. The method of increasing the turn radii or even changing the turn direction leads to waved or straight trajectories. We call this method *negative steering* because the robot steers by decreasing or inverting the standard turn.

AHHS controllers

In both of the reported controllers, we assumed that a basic 'forward-driving' hormone H^f is produced (in the following: forward hormone) in both compartments of the robot

at rate α. This hormone activates the motors. The main difference between these two controllers is the asymmetric production rate (α_l in the left compartment, $alpha_r$ in the right compartment, with $\alpha_l \neq \alpha_r$) in case of the first controller. The second controller has a symmetric production rate ($\alpha_l = \alpha_r = \alpha$). Thus, the levels of the forward hormone are equal in the 'normal' state, the robot basically drives in straight lines. Such an AHHS controller can easily be combined with a collision-avoidance system, as it was discussed in (Schmickl and Crailsheim, 2009; Stradner et al., 2009).

First AHHS controller

In our first AHHS controller, we assumed that the robot is equipped with a sensor that is able to determine whether it points towards the light source (within an angular threshold of $\pm 90^o$ around the sensor center). If this is the case, the sensor triggers the production of a light-dependent hormone H^l (in the following: light hormone). The light hormone interacts with the forward hormone H^f by blocking (decreasing) it. Thus, the hormone level in the compartment, that corresponds to the side of the light-sensor, is decreased by the light hormone and the robot starts to turn in curves towards this side. This first approach was inspired by the phototactic behavior of *Euglena gracilis* (Bound and Tollin, 1967), which rotates around its axis until a shading pigment shades the organism's eyespot. This is only the case, if the organism is oriented correctly towards the stimulus (light) source. In this case, all phobic responses disappear and the organism moves towards its target. In our case, also just one binary and directional sensor is available and the 'body' of the robot acts as a shading device.

We chose a system of difference equations to model the agent. It is assumed that the agent moves in a plane. The agent's position is given by $\mathbf{x}$ and updated by

$$\frac{\Delta \mathbf{x}}{\Delta t} = \begin{pmatrix} \cos\phi(t) \\ \sin\phi(t) \end{pmatrix} v, \tag{1}$$

for the agent's heading ϕ and a constant velocity $v > 0$. The change of the heading is defined by

$$\frac{\Delta \phi}{\Delta t} = \left((H_l^F(t) - H_l^L(t)) - (H_r^F(t) - H_r^L(t)) \right) \theta, \tag{2}$$

for the value of the forward hormone in the left compartment H_l^F (right compartment H_r^F), the value of the light hormone in the left compartment H_l^L (right compartment H_r^L), and a parameter θ called steering intensity that defines the intensity of the turns related to the difference in hormones in the tow compartments. The dynamics of the forward hormone H^F are given by

parameter	value
hormone production left α_l	0.11 [1/time unit]
hormone production right α_r	0.1 [1/time unit]
hormone decay β	0.04 [1/time unit]
hormone diffusion D	0.001
agent velocity v	0.01 [space unit/time unit]
sensor scale factor σ	0.03
steering intensity θ	0.1
sensor offset δ	45^o

Table 1: Standard parameters for the model of the first controller.

$$\frac{\Delta H_l^F}{\Delta t} = \alpha_l - \beta H_l^F(t) + D(H_r^F(t) - H_l^F(t)), \tag{3}$$

$$\frac{\Delta H_r^F}{\Delta t} = \alpha_r - \beta H_r^F(t) + D(H_l^F(t) - H_r^F(t)), \tag{4}$$

for hormone production rates α_l (left compartment) and α_r (right compartment), decay rate β, and diffusion constant D. The update rule of the light hormone H^L is

$$\frac{\Delta H_l^L}{\Delta t} = -\beta H_l^L(t) + D(H_r^L(t) - H_l^L(t)), \tag{5}$$

$$\frac{\Delta H_r^L}{\Delta t} = -\beta H_r^L(t) + \sigma S(t) + D(H_l^L(t) - H_r^L(t)), \tag{6}$$

for a sensor input $S(t)$ and the sensor scale factor σ.

The sensor returns a 1, if it points towards the light source (within an angular threshold of $\pm 90^o$ around the sensor center). Otherwise it returns a 0. This is defined by the scalar product:

$$S(t) = \begin{cases} 1 & \text{if } \left| \arccos\left(\|\mathbf{x}(t)\| \cdot \begin{pmatrix} \cos(\phi(t)+\delta) \\ \sin(\phi(t)+\delta) \end{pmatrix} \right) \right| > 90^o \\ 0 & \text{else.} \end{cases} \tag{7}$$

The standard parameters, that were used, if not stated explicitly, are given in Table . With this model we generated examples of trajectories by solving it numerically. Examples of three trajectories are shown in Fig. 2. These trajectories clearly show the two different strategies of positive and negative steering by changing the steering intensity parameter θ (a convoluted trajectory compared to waved and straight trajectories).

The model was also used to do extensive scans of parameter intervals. For example, an interesting behavior was found for the sensor scale factor σ that indicates complex behavior.

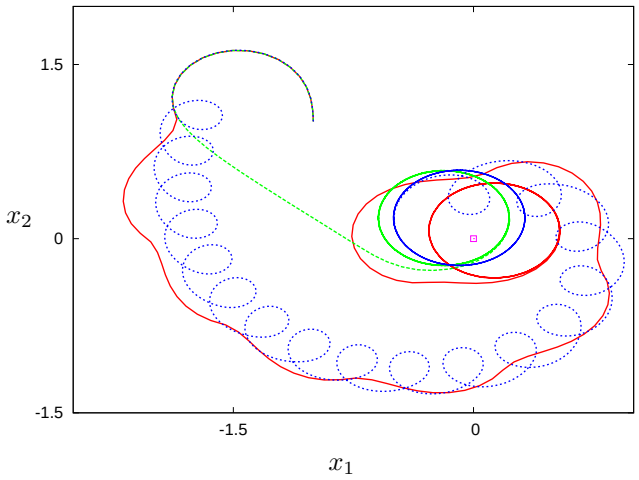


Figure 2: Examples of three trajectories with different parameter settings. The agent starts at $\mathbf{x} = (-1, 1)$ with heading $\phi = 90^o$ (north). The maximum of the light gradient is located at $(0, 0)$. The blue trajectory is an example of positive steering ($\theta = 0.1$). The green ($\sigma = 0.25$) and red ($\sigma = 0.01$) trajectories are examples of negative steering ($\theta = -0.1$).

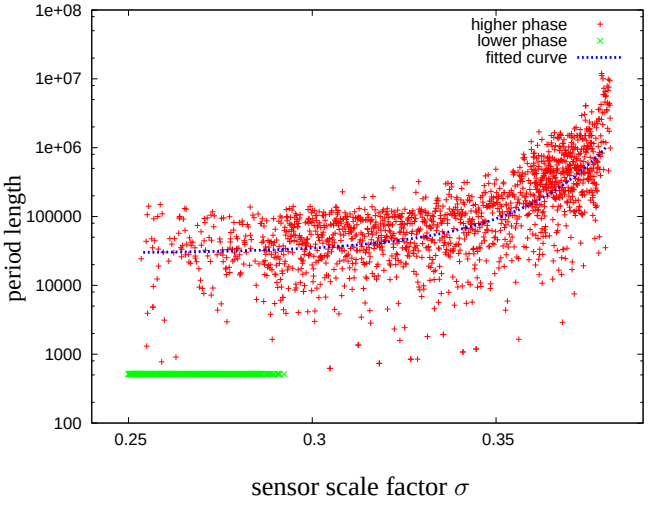


Figure 3: Scan over the sensor scale factor σ showing its influence on the length of the asymptotic period length. The green points correspond to the smallest possible period, red points correspond to rather complex periodic behaviors. The fitted blue curve is double exponential.

The sensor scale factor influences the radius of the circular behavior to which the robot converges to (i.e., the period length). Results are shown in Fig. 3 that indicate a complex relation (double exponential increase) between the sensor scale factor and the period length.

Second AHHS controller

In this example, we assumed a photo-receptor which is mounted on top of the robot, so that it has no directionality at all. It just can report the local luminance in a graduated manner: The higher the local luminance, the higher is the reported sensor value. This sensor value produces a light-dependent hormone in one of the two compartments of the AHHS controller, which breaks down the forward-driving hormone. As the sensor produces this hormone proportionally to the local illuminance, the forward-driving hormone level is lowered also in a proportional level, leading to smaller curve radii in higher illuminated areas. This rotation-behavior, changing the orientation of the robot frequently and decreasing the net movement speed of the robot, is inspired by the mechanisms of chemotaxis reported with *Escherichia coli*.

The agent's position update of this second controller is defined as in Eq. 1. The dynamics of heading ϕ is now given just by the difference of the forward hormone:

$$\frac{\Delta\phi}{\Delta t} = (H_l^F(t) - H_r^F(t))\theta. \tag{8}$$

The update rule of the forward hormone is similar to the definition above, except that now it is reduced by the light hormone H^l:

$$\frac{\Delta H_l^F}{\Delta t} = \alpha - \beta H_l^F(t) + D(H_r^F(t) - H_l^F(t)) - \gamma H_l^L(t), \tag{9}$$

$$\frac{\Delta H_r^F}{\Delta t} = \alpha - \beta H_r^F(t) + D(H_l^F(t) - H_r^F(t)) - \gamma H_r^L(t), \tag{10}$$

for production rate α (now symmetrically defined), diffusion constant D, decay rate β, and hormone-induced decay γ.

The update of the light hormone is defined as given by Eq. 6. The sensor input is now a continuous value which is a direct measurement of the local light intensity. The light gradient is simply defined by the reciprocal of the distance of the agent to the origin which is here the position of the light source:

$$S(t) = 1/\|\mathbf{x}(t)\|, \tag{11}$$

for agent position $\mathbf{x}$. The standard parameters, that were used, if not stated explicitly, are given in Table . An example of an agent's trajectory for this second controller is shown in Fig. 4.

We used this model to do extensive parameter interval scans. Such scans are the specialty of such abstract mathematical models due to the small computational cost of solving them. We just need a valid metric to (automatically) measure the performance of the parameter set. One possible

parameter	value
hormone production left α	0.1 [1/time unit]
hormone decay β	0.04 [1/time unit]
hormone diffusion D	0.03
agent velocity v	0.01 [space unit/time unit]
sensor scale factor σ	0.2
steering intensity θ	0.3
hormone-induced decay γ	0.003 [1/time unit]

Table 2: Standard parameters for the model of the second controller.

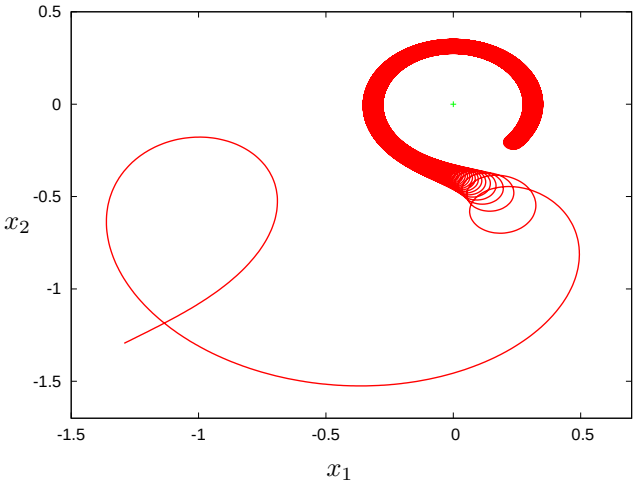
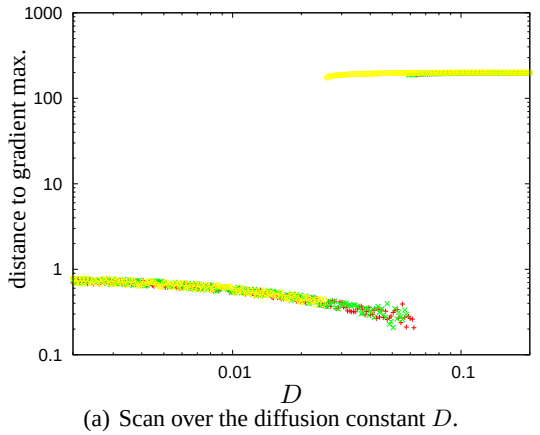
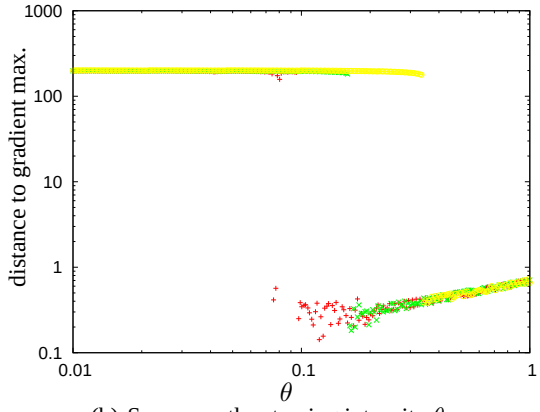


Figure 4: An example of an agent's trajectory for the second controller.

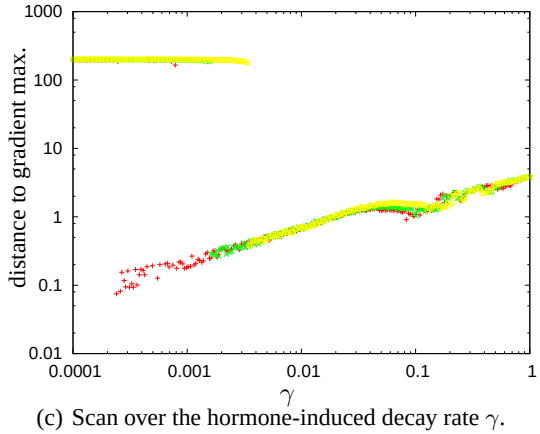
measure of the quality of the gradient ascent is the distance to the maximum during the asymptotic and periodic behavior of the agent. In Fig. 5 we present scans over the diffusion constant D, the steering intensity θ, and the hormone-induced decay rate γ for three different initializations of the agent position. For each parameter value six distances of the trajectory to the maximum of the light gradient during the last 3000 time steps are plotted (3000, 2500, ..., 500, 0 time steps before the numerical integration was stopped). Clearly two phases are detected. The distances above a distance of 100 correspond to the maximal possible distance that can be obtained by a robot (by driving in a straight line). Close to optimal parameter settings are found by choosing parameters with low distances. However, the parameters are not fully mutually independent.



(a) Scan over the diffusion constant D.



(b) Scan over the steering intensity θ.



(c) Scan over the hormone-induced decay rate γ.

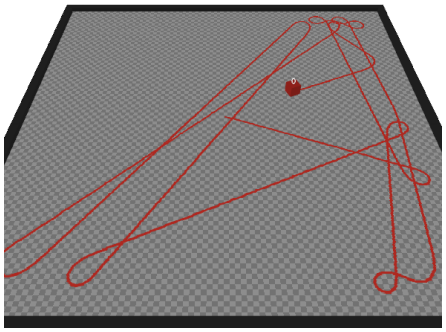
Figure 5: Scan over different parameters showing the distance to the maximum of the gradient of 6 time steps during the asymptotic behavior (3000, 2500, ..., 500, 0 steps before stopping to iterate) for three different initializations of the agent's positions (indicated by different colors). The distances above 100 correspond to the maximal possible distance that is obtained by driving in a straight line. Clearly two phases are detected.

Multi-agent implementation of the second AHHS controller

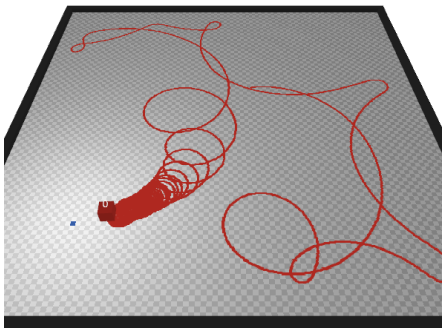
We tested the second AHHS controller in a multi-agent simulation of real robotic hardware, because we think that this controller is especially interesting for robotics: It allows a gradient ascent without any explicit memory of past sensor values and without any directionality of the used sensor. To test whether this concept works also in a more realistic environment (walls, obstacles, collision avoidance of robots) compared to the mathematical model described above, we implemented the AHHS controller in an individual based multi-agent simulation as well. In our multi-agent simulation, each robot can detect nearby obstacles through 2 IR sensors which are mounted laterally. These distance sensors emit a 'collision stress' hormone, which additionally activates the motor on the ipsi-lateral side. This leads to a turning away from the obstacle. This collision-avoidance behavior was implemented in an AHHS controller in (Schmickl and Crailsheim, 2009) where it is described in more detail. The focal questions for our experiment described here are: Will the collision-avoidance interfere with the phototactic behavior of our above-mentioned second AHHS controller? Will the phototactic behavior be adaptive to environmental fluctuations? Will sensor noise affect the system? To investigate these questions we tested the combined AHHS controller (collision avoidance and phototaxis) in a simulated robotic arena which was bound by an arena wall. All sensor data was affected by $\pm 25\%$ uniform random noise. To test the adaptability of the robots, we switched the position of the simulated light source to the other side of the arena, as soon as the robot approached the first optimum.

As can be seen in Fig. 6(a), the robot performs 'normal' collision avoidance behavior successfully when no light spot is present in the arena. As soon as the light spot is forming a gradient pointing towards the lower left corner of the arena, the robot starts to approach it with its characteristic phototactic behavior, see Fig. 6(b). After the robot approached the light spot, we switched the lightspot's position at a sudden and the robot changed its behavior and started to approach the new optimum, see Fig. 6(c). Fig. 7 shows the dynamics of the forward-driving hormone and of the light-induced hormone in the last two phases of the experiment. It is clearly visible how the robot maximizes the light hormone, thus it approaches the light spot, which, in turn, leads to a lowering of the forward-driving hormone.

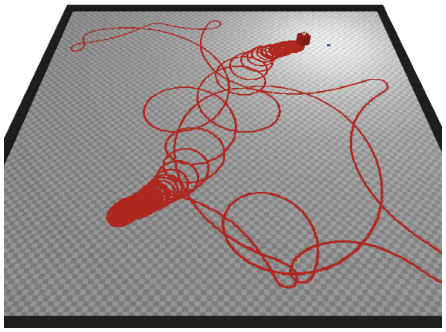
To perform a further test of this controller in the multi-agent simulator, we performed additionally a test run, which is shown in Fig. 1. In this run, the light spot was placed at the right side of a lengthy arena and 5 robots started simultaneously at the left side of the arena. A wall narrows down the possible paths from the left to right side of the arena and the robots had to avoid each other, as well as the surrounding outside wall. As the trajectories in Fig. 1 demonstrate the robots successfully managed to approach the light spot. The robot-to-robot interactions led to even more complex trajectories compared to those of the single-robot runs. We assume that such swarm effects can be exploited to kick robots out of circular trajectories that surround local optima. This will be tested in future studies.



(a) No light spot in the habitat.



(b) Light spot in the lower left corner.



(c) Light spot moved to the upper right spot.

Figure 6: Trajectories of robots in three phases of our 'disturbance' experiment. Without any light spot, the robot performs only collision avoidance. As soon as the light spot is in the left lower corner, the robot approaches it in the characteristic phototactic behavior. As soon as the light spot is shifted to the right upper arena corner, the robot changes its behavior and approaches the new optimum.

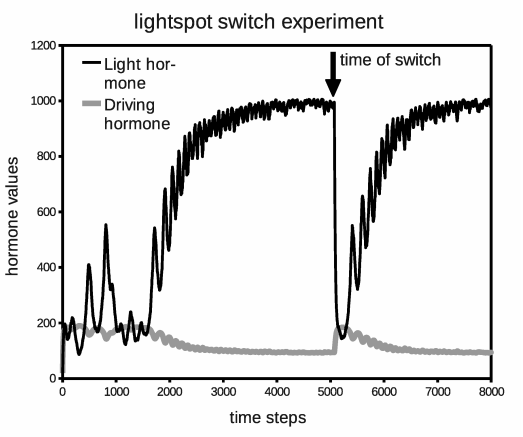


Figure 7: Hormone values in the AHHS controller that governed the robot's behavior in the 'disturbance' experiment, which is depicted in figure 6.

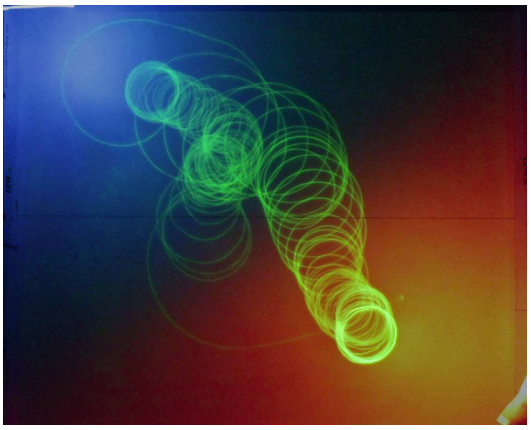


Figure 8: Composite image of the robotic implementation of the light-seeking AHHS in an e-puck robot. The two light emitters can be seen in the top left and bottom right corners. The robot trail, here captured using a phosphorescent paint, shows the spiral-way approach to the top left corner and the bottom right corner.

Implementation of the AHHS in robotic hardware

Based on the results we obtained from our simulation studies, we implemented the algorithm of the second AHHS controller (described above) on a robotic platform. We used an 'e-puck' robot (Mondada et al., 2009) for this experiment. The robot was equipped with only one light-sensor on top, pointing upwards. Therefore, the light sensor reports local luminance without any directional information. Also, the robot is equipped with two wheels (differential drive). The 'forward hormone' is steadily produced and decays proportionally, establishing an equilibrium that in turn determines the robots general forward speed. The 'light hormone' of the AHHS is emitted in response to light sensation, increasing the decay rate of the 'forward hormone' to slow the right wheel, thus inducing a curved trajectory. For the AHHS, we used the following parameterization: $\beta_1 = 0.04$, $\beta_2 = 0.04$, $D = 0.015$, $\alpha = 0.1$, $\gamma = 0.03$, and $\sigma = 0,055$. The light sensor reports sensor values between 0 (absolute darkness) and 1 (maximum luminance) with a noise factor of about 0.2. Because the arena was bounded by a wall, we implemented a collision-avoidance behavior based on the 8 IR proximity sensors of the e-puck robot. This behavior overruled the AHHS control when the robot approached a wall. In (Stradner et al., 2009), we showed that this kind of collision-avoidance behavior can also be built using an AHHS control.

For this experiment, we used an arena ($2.0\text{m} \times 1.8\text{m}$) with two light emitters in opposing corners (top left and bottom right). At first, only one emitter (top left) was switched on. The robot was placed directly under the other, switched off, light source with a heading pointing away from the light optimum. The robots objective was to navigate to the brightest spot in the arena, directly under the light emitter (top left). After the robot had reached the light spot, the light emitter was switched off, while the other emitter was switched on (bottom right). The robots task was now to locate and navigate to the new light optimum.

Figure 8 shows, that the robot (running the AHHS) performs the spiral-way target approach towards the light gradient successfully. It can be seen that the light sensor's noise is significantly reduced in both hormone levels, thus enabling the smooth spiraling movement of the robot.

CONCLUSIONS AND FUTURE WORKS

Conclusions

We have successfully demonstrated that a simple bio-inspired AHHS controller can be used to achieve phototactic behavior in autonomous robots. The controller is simple, so that it can be easily analyzed and studied with mathematical differential-equation models. Using this technique we analyzed the emerging phototactic behavior of two different controller setups, both based on different AHHS configurations. Both setups managed to perform phototaxis with just one single illuminance sensor, having a different sensor characteristic in each setup. Our mathematical analysis shows that the more interesting (and more complex) behavioral patterns can be produced with the second controller. This is especially interesting, because in this controller setup, the sensor offers no directionality and past information is never explicitly stored in a memory system. This means that the robot does not simply compare old and new sensor data and performs no memory-based gradient ascent. The behavior also differs significantly from classical Braitenberg vehicle approaches (Braitenberg, 1984).

One important aspect of simple mathematical models is that they allow exhaustive parameter sweeps in reasonable computational time. From our performed parameter sweeps

we conclude that the modeled AHHS controllers have a defined, but wide, range of parameters that lead to the desired phototactic behavior. The tests in the multi-agent simulation show, that this phototactic behavior can be performed, even with an underlying obstacle avoidance, with a more realistic robotic habitat and with a huge amount of sensor noise. And even multiple and frequent robot-to-robot interactions did not significantly impair the robot's ability to approach the desired target. In addition, the 'disturbance experiments' showed that the emerging phototactic behavior is stable on the one hand and flexible on the other hand. The AHHS controller has also been shown to work on real robotic hardware, in our case the e-puck robot. It performed a smooth spiral-way target approach similar to those in the multi-agent simulation. Furthermore it could adapt to the changing environment, when the light source switched places.

Future Works

In the future, we plan to use Evolutionary Computation to optimize parameter sets of our AHHS systems. We plan to implement a novel way of Artificial Evolution, so that evolutionary operators can 'create' new hormones and new sensor-to-hormones and hormones-to-actuator links. In addition, we plan to extend the system to multi-modular robotics, so the virtual hormones can be exchanged by linked robotic modules. This way, we plan to mimic the evolutionary step from uni-cellular to multi-cellular organism, like it happened several times in the natural evolution of life forms.

ACKNOWLEDGMENTS

We were supported by the following grants: EU-IST FET project 'I-SWARM', no. 507006 and the FWF research grant no. P19478-B16, EU-IST FET project 'SYMBRION', no. 216342, EU-ICT project 'REPLICATOR', no. 216240.

References

Alon, U., Surette, M. G., Barkai, N., and Leibler, S. (1999). Robustness in bacterial chemotaxis. *Nature*, 397:168–171.

Beni, G. (2005). From swarm intelligence to swarm robotics. In Şahin, E. and Spears, W. M., editors, *Swarm Robotics - SAB 2004 International Workshop*, LNCS, pages 1–9, Santa Monica, CA.

Bound, K. and Tollin, G. (1967). Phototactic response of *euglena gracilis* to polarized light. *Nature*, 216:1042–1044.

Braitenberg, V. (1984). *Vehicles: experiments in synthetic psychology*. MIT Press, Cambridge, MA.

Bray, D. (2009). *Wetware: A Computer in Every Living Cell*. Yale University Press.

Şahin, E. (2005). Swarm robotics: From sources of inspiration to domains of application. In Şahin, E. and Spears, W. M., editors, *Swarm Robotics - SAB 2004 International Workshop*, volume 3342 of *LNCS*, pages 10–20, Berlin, Germany. Springer-Verlag.

Darnton, N. C., Turner, L., Rojevsky, S., and Berg, H. C. (2007). On torque and tumbling in swimming *escherichia coli*. *Proceedings of the National Academy of Sciences*, 189:17561764.

Floreano, D., Husbands, P., and Nolfi, S. (2008). Evolutionary Robotics. In Siciliano, B. and Oussama, K., editors, *Handbook of Robotics*, chapter 61, pages 1423–1452. Springer, Berlin.

Grey Walter, W. (1950). An imitation of life. *Scientific American*, 182(5):42–45.

Grey Walter, W. (1951). A machine that learns. *Scientific American*, 185(2):60–63.

Hamann, H., Stradner, J., Schmickl, T., and Crailsheim, K. (2010a). Artificial hormone reaction networks: Towards higher evolvability in evolutionary multi-modular robotics. In *Alife XII*. MIT Press.

Hamann, H., Stradner, J., Schmickl, T., and Crailsheim, K. (2010b). A hormone-based controller for evolutionary multi-modular robotics: From single modules to gait learning. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'10)*. in press.

Holmes, S. (1903). Phototaxis in *volvox*. *Biological Bulletin*, 4(6):319–326.

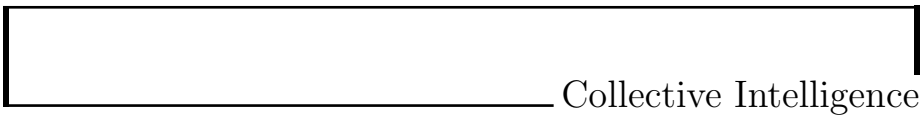
Khan, S., Spudich, J. L., McCray, J. A., and Trentham, D. R. (1995). Chemotactic signal integration in bacteria. *Proceedings of the National Academy of Sciences*, 92:9757–9761.

Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., Magnenat, S., Zufferey, J.-C., Floreano, D., and Martinoli, A. (2009). The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, volume 1, pages 59–65.

Schmickl, T. and Crailsheim, K. (2009). Modelling a hormone-based robot controller. In *MATHMOD 2009 - 6th Vienna International Conference on Mathematical Modelling*.

Stradner, J., Hamann, H., Schmickl, T., and Crailsheim, K. (2009). Analysis and implementation of an artificial homeostatic hormone system: A first case study in robotic hardware. In *The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'09)*, pages 595–600. IEEE Press.

Yi, T.-M., Huang, Y., Simon, M. I., and Doyle, J. (2000). Robust perfect adaptation in bacterial chemotaxis through integral feedback control. *Proceedings of the National Academy of Sciences*, 97(9):4649–4653.


Collective Intelligence

If You Can't Be With the One You Love, Love the One You're With: How Individual Habituation of Agent Interactions Improves Global Utility

Adam P. Davies[1], Richard A. Watson[1], Rob Mills[1], C. L. Buckley[2], Jason Noble[1]

[1]Natural Systems group, ECS, University of Southampton, U.K.
[2]CCNR, University of Sussex, U.K.
apd1e09@ecs.soton.ac.uk

Abstract

Simple distributed strategies that modify the behaviour of selfish individuals in a manner that enhances cooperation or global efficiency have proved difficult to identify. We consider a network of selfish agents who each optimise their individual utilities by coordinating (or anti-coordinating) with their neighbours, to maximise the pay-offs from randomly weighted pair-wise games. In general, agents will opt for the behaviour that is the best compromise (for them) of the many conflicting constraints created by their neighbours, but the attractors of the system as a whole will not maximise total utility. We then consider agents that act as 'creatures of habit' by increasing their preference to coordinate (anti-coordinate) with whichever neighbours they are coordinated (anti-coordinated) with at the present moment. These preferences change slowly while the system is repeatedly perturbed such that it settles to many different local attractors. We find that under these conditions, with each perturbation there is a progressively higher chance of the system settling to a configuration with high total utility. Eventually, only one attractor remains, and that attractor is very likely to maximise (or almost maximise) global utility. This counterintutitve result can be understood using theory from computational neuroscience; we show that this simple form of habituation is equivalent to Hebbian learning, and the improved optimisation of global utility that is observed results from well-known generalisation capabilities of associative memory acting at the network scale. This causes the system of selfish agents, each acting individually but habitually, to collectively identify configurations that maximise total utility.

Selfish Agents and Total Utility

This paper investigates the effect of a simple distributed strategy for increasing total utility in systems of selfishly optimising individuals. The broader topic concerns many different types of systems. For example, in technological systems, it is often convenient or necessary to devolve control to numerous autonomous components or agents that each, in a fairly simple manner, acts to optimise a global performance criterion: e.g. communications routing agents act to minimise calls dropped, or processing nodes in a grid computing system each act to maximise the number of jobs processed (1,2). However, since each component in the network acts individually, i.e., using only local information, constraints between individuals can remain unsatisfied, resulting in poorly optimised global performance. In an engineered system one could, in principle, mandate that all nodes act in accord with the globally optimal configuration of behaviours (assuming one knew what that was) – but this would defeat the scalability and robustness aims of complex adaptive systems. The question for engineered complex adaptive systems then, is the question of how to cause simple autonomous agents to act 'smarter' in a fully distributed manner such that they better satisfy constraints between agents and thereby better optimise global performance.

Meanwhile, in evolutionary biology it appears that in certain circumstances symbiotic species have formed collaborations that are adaptive at a higher level of organisation (3), but it has been difficult to integrate this perspective with the assumption that under natural selection such collaborations must be driven by the selfish interests of the organisms involved (4,5). In social network studies there is increasing interest in adaptive networks (6) where agents in a network can alter the structure of the connections in the network. Of particular interest is the possibility that by doing so they may increase the ability of the system to maintain high levels of cooperation (7,8). However, a general understanding of how agents on a network modify their interactions with others in a way that increases total cooperation is poorly understood. In each domain we are, at the broadest level, interested in understanding/identifying very simple mechanisms that might cause self-interested agents to modify their behaviour, or how their behaviours are affected by others, in a manner that increases adaptation or efficiency either globally or at a higher-level of organisation than the individual.

Taking an agent perspective, the obvious problem is this: If it is the case that agents collectively create adaptation that is not explained by the default selfish behaviours of individuals, then it must be the case that, on at least some occasions, agents take decisions that are detrimental to individual interests. If this were not the case then there is nothing to be explained over and above the selfish actions of individuals. But if it is the case, then this runs counter to any reasonable definition of a rational selfish agent. In what sense could it be self-consistent to suggest that a *selfish* agent has adopted a behaviour that *decreased* individual utility? One way to make sense of this is the possibility that, at the time that the agent

takes this action, it appears to them be the best thing for them – that the agent is no longer making decisions according to the true utility function but some distortion of it that alters their perception of the utility of that action. If somehow the perception of an agent were distorted in the right way, so that the action that it preferred, the one that it thought was best for it, was in fact the action that was globally optimal, then a rational agent with this distorted set of preferences could increase global efficiency even at the cost of personal utility. One might assume that this is easier said than done – but in this paper we suggest that the reverse is true; it is easier to do than to explain how it works. However, the general problem and the essence of the strategy we investigate is straightforwardly introduced by means of the following simple parable. Although this makes the concepts intuitively accessible it might tend to cast the model in a narrow interpretation – it is, of course, not really a model about scientists and their drinking habits, but a general model of interacting agents on a network with pair-wise constraints between binary behaviours.

Consider a community of individuals (e.g. researchers) in a social network. Each has an intrinsic symmetric compatibility, or 'complementarity', with every other individual that determines the productivity/pay-off of collaborating with them. Each evening all researchers attend one of two intrinsically equal public houses (or other such collaborative projects) initially at random. Individuals must decide which to attend based solely on who else attends that venue. Each individual seeks to maximise their scientific productivity by attending the pub that, on that night, maximises the sum of compatibilities with other researchers and minimises incompatibilities. Assessing the company they find at any moment, individuals therefore (one at a time in random order) may choose to switch pubs to maximise their productivity according to the locations of others. Since each individual has compatibilities and incompatibilities with all other individuals, each must choose the pub that offers the best compromise of these conflicting interests. Since compatibilities are symmetric, the researchers will quickly reach a configuration where no-one wants to change pubs (9), however, this configuration will not in general be the arrangement that is maximal in total productivity, but merely a locally optimal configuration.

This describes the basic behaviour of agents on the network. Our aim is to devise a simple individual strategy that causes researchers to make better decisions about when to change pubs such that total productivity is maximised. This will necessarily mean that some researchers, at some moments in time, must change pubs even though it decreases their individual productivity.

Surprisingly, we find that this can be achieved (over many evenings) by implementing a very simple rule – each individual must develop a preference for drinking with whichever other researchers they are drinking with right now. As Crosby, Stills and Nash put it "If you can't be with the one you love, honey, love the one you're with" (10). Since we already know the arrangements of researchers will be initially random and, most of the time, at best sub-optimal, this seems like a counter productive strategy. But, in fact we find that it is capable, given enough evenings and slowly developed preferences, of causing all researchers to develop preferences that cause them to make decisions that maximise total productivity reliably every evening.

The agents that we model are therefore not wholly selfish agents – they sometimes take actions that do not maximise individual utility, which is the point of the exercise after all. But neither are they overtly cooperative or altruistic agents. They are simply *habitual selfish agents*. In this paper we are not directly addressing *why* it might be that selfish agents act as creatures of habit, although we will discuss this briefly. But we suggest this type of distorted perception of a true utility function, one which agents come to prefer familiarity over otherwise obvious opportunities for personal gain, is one which does not require any teleological or, certainly, any centralised control and is therefore relevant to many domains.

In the next two sections we will detail an illustration of this strategy and the results we observe. In the Discussion section we will outline how this result can be interpreted in terms of adaptive network restructuring. Briefly: Initially, interactions between agents are governed by a network of intrinsic constraints (compatibilities), and latterly they are governed by a combination of these intrinsic constraints plus the interaction preferences that the agents have developed. The new behavioural dynamics of the agents caused by interaction preferences can therefore equally be understood as a result of changes to connection strengths in the effective interaction network. The increased global utility observed can then be explained using theory from computational neuroscience. In particular, we can understand how the system as a whole improves global adaptation via the observation that when each agent acts as a creature of habit it changes the effective dependencies in the network in a Hebbian manner (11,12). This means that through the simple distributed actions of each individual agent, the network as a whole behaves in a manner that is functionally equivalent to a simple form of learning neural network (13). In this case, the network is not being trained by an external training set, but instead is 'learning' its own attractor states, as we will explain. We discuss how a separation of the timescales for behaviours *on* the network and behaviours *of* the network (i.e. changes to network structure) is essential for this result.

Methods

Default agents

Our model involves N=100 agents playing two-player games on a fully connected network (Table 1). Specifically, for each game (i.e. each connection in the network), there is a single symmetric payoff matrix, U_{ij}, which defines for agents i and j either a coordination game (α=1, β=0) or anti-coordination game (α=0, β=1) with equal probability (Table 1).

		Player 2	
		A	B
Player 1	A	α,α	β,β
	B	β,β	α,α

Table 1: Payoff for (player 1, player 2).

Games are played in extensive form, i.e., initially all agents in the network are assigned a behaviour at random, and then each agent in random order is permitted to update its behaviour (to either A or B). Each agent does so according to a best response strategy, i.e., to adopt the behaviour (choose a pub) that maximises its utility, u_i (Eq.1) given the behaviours (pub choices) currently adopted by its neighbours:

$$u_{i(t)} = \sum_{j}^{N} U_{ij}(s_{i(t)}, s_{j(t)}) \qquad \textbf{(1)}$$

where $U_{ij}(x,y)$ is the payoff received by player i when player i plays strategy x and player j plays strategy y (according to Table 1 above), and $s_n(t)$ is the strategy currently played by agent n. Behaviours are updated in this manner repeatedly. Each agent is involved in many games but can adopt only one behaviour at any one time, thus coordinating with one neighbour may preclude coordinating with another, and so each agent must therefore adopt the behaviour that is the best compromise of these constraints. By using a symmetric game, $U_{ij}=U_{ji}$, we can ensure that the system will reach a stable fixed point (9), i.e. a configuration where no agent wants to change behaviour unilaterally (14). Moreover, this configuration will be a local optimum in the total or global utility, G, of the system which is simply the sum of individual utilities (9) (Eq.2).

$$G_{(t)} = \sum_{i}^{N} \sum_{j}^{N} U_{ij}(s_{i(t)}, s_{j(t)}) \qquad \textbf{(2)}$$

However, in general, the stable configuration reached from an arbitrary initial condition will not be globally maximal in total utility. If the system is repeatedly perturbed (reassigning random behaviours to all agents) at infrequent intervals (here every 1000 time steps = one evening), and thereby allowed to settle, or relax, to many different local equilibria (on different evenings), the behaviour of the system given these default agents can be described by the distribution of total utilities found at the end of each of these 'relaxations' (Fig. 1.c).

Creatures of habit

We seek a simple distributed strategy that causes agents to make different (hence unselfish) behavioural choices in particular contexts in such a manner that configurations of higher global utility are attained or high global utility configurations are attained with greater reliability (i.e. from a greater number of random initial conditions). To this end we investigate agents that act as 'creatures of habit' by increasing their preference to coordinate with whichever neighbours they are coordinated with at the present moment (regardless of whether this is presently contributing positively or negatively to their utility). Specifically, in addition to the 'true' utility matrix, U_{ij}, each agent also possesses a 'preference' matrix, P_{ij}, for each of its connections. These are used to modify the behaviour of the agent such that it chooses the behaviour that maximises its 'perceived utility', p_i, (Eq.3), instead of its true utility (Eq.2) alone:

$$p_{i(t)} = \sum_{j}^{N} \left[U_{ij}(s_{i(t)}, s_{j(t)}) + P_{ij(t)}(s_{i(t)}, s_{j(t)}) \right] \qquad \textbf{(3)}$$

where P_{ij} is a pay-off matrix that represents an agent's preference for the combination of behaviours s_i and s_j. The perceived utility is thus simply the sum of the true utility plus the agent's preferences. Each agent has a separate preference pay-off matrix for each other agent. All preference payoff matrices are initially set to zero, such that the initial dynamics of the agents are as per the default agents. But as the values in these matrices change over time they may come to collectively overpower the tendency to maximise true utility and thereby cause agents to make different decisions about which behaviour is best for them to adopt.

It should be clear that it is possible in principle, with knowledge of the globally optimal system configuration, to assign values to each of the P_{ij} matrices that will cause agents to adopt behaviours that maximise global system utility instead of choosing behaviours that maximise individual utility and thereby failing to maximise total utility. But our question then becomes how to enable agents to develop, via a simple distributed strategy (without knowledge of the global optimum, of course) such a perception of interactions with others that causes them to make these globally optimal decisions.

The strategy we investigate is very simple – we assert that each P_{ij} matrix is updated so as to increase the agent's perceived utility at the current moment. Specifically, whenever an agent's behaviour has just been updated (whether it changed behaviour or not), with probability r_p = 0.0001 all of its P_{ij} matrices will also be updated. To decide how to update each P_{ij} matrix, one of two possibilities is considered (chosen at random), either $P'_{ij} = P_{ij}(t)+A$ or $P'_{ij} = P_{ij}(t)-A$, where A is the adjustment matrix defined in Table 2. If $p_i(t)\_given\_P'_{ij} > p_i(t)\_given\_P_{ij}$ then $P_{ij}(t+1)= P'_{ij}$ else $P_{ij}(t+1)= P_{ij}$.

		Player 2	
		A	B
Player 1	A	r	$-r$
	B	$-r$	r

Table 2: adjustment matrix A (r=0.005)

This strategy has the effect of increasing agent i's preference for coordinating or anti-coordinating with agent j according to whether it is currently coordinating or anti-coordinating with agent j, respectively. Note that this preference is not sensitive to whether the interaction between these two agents is currently contributing positively to the utility of agent i; an agent increases its preference for the current combination of behaviours irrespective of whether $U_{ij}(s_{i(t)}, s_{j(t)}) > 0$. It is thereby simply reinforcing a preference for doing more of what it is currently doing with respect to coordinating with others (i.e. I'm in the same pub with them now, so change my preference so I like being in the same pub with them a little more or dislike it less). This is a counterintuitive strategy in the sense that it can increase the

preference for coordinating with other agents even when U_{ij} defines an anti-coordination game, and vice versa. Note that this habituation does not alter the independent preference for playing behaviour A or B, but instead alters the preference for coordinating behaviours with others.

Results

The system is run for 1000 relaxations, of 1000 time steps each, without habituation (i.e. default agents). Example trajectories of total utility for individual relaxations are shown in Fig1.a. The total utility at the end point of each relaxation is shown in Fig.1.b (first 1000 relaxations). The system is then run for 1000 relaxations with habituation (i.e. r=0.0005). As the preference utility matrices change over time the distribution of local optima found changes (Fig.1.b, relaxations 1001-2000). We see in these figures that the probability of finding the configurations with high total utility increases over time, such that the trajectories of the system after habituation (Fig.1.c) find high-utility configurations reliably. Histograms of the total utilities found before and after habituation are shown in Fig.1.d.

These results therefore show that habituation of agent interactions, created by developing a preference for whatever combination of behaviours is currently observed, has the effect of causing agents to adopt different behaviours in some situations (essentially because the resulting combination of behaviours has been experienced more often in the past). Specifically, since *without* habituation agents adopt behaviours that maximise their individual (true) utility, so the different behaviours adopted *with* habituation are therefore behaviours that (at least temporarily) decrease their true utility – otherwise the trajectories would not be different (neutral changes are very rare in this system). Over time agents therefore come to choose behaviours that decrease their individual utility in certain circumstances, but that allow the system to ultimately reach states of global utility higher than would have been otherwise possible. Accordingly, trajectories before and after habitation are different, but more specifically, the behavioural choices that agents make after habituation increase total system utility and are in this well-defined sense more cooperative.

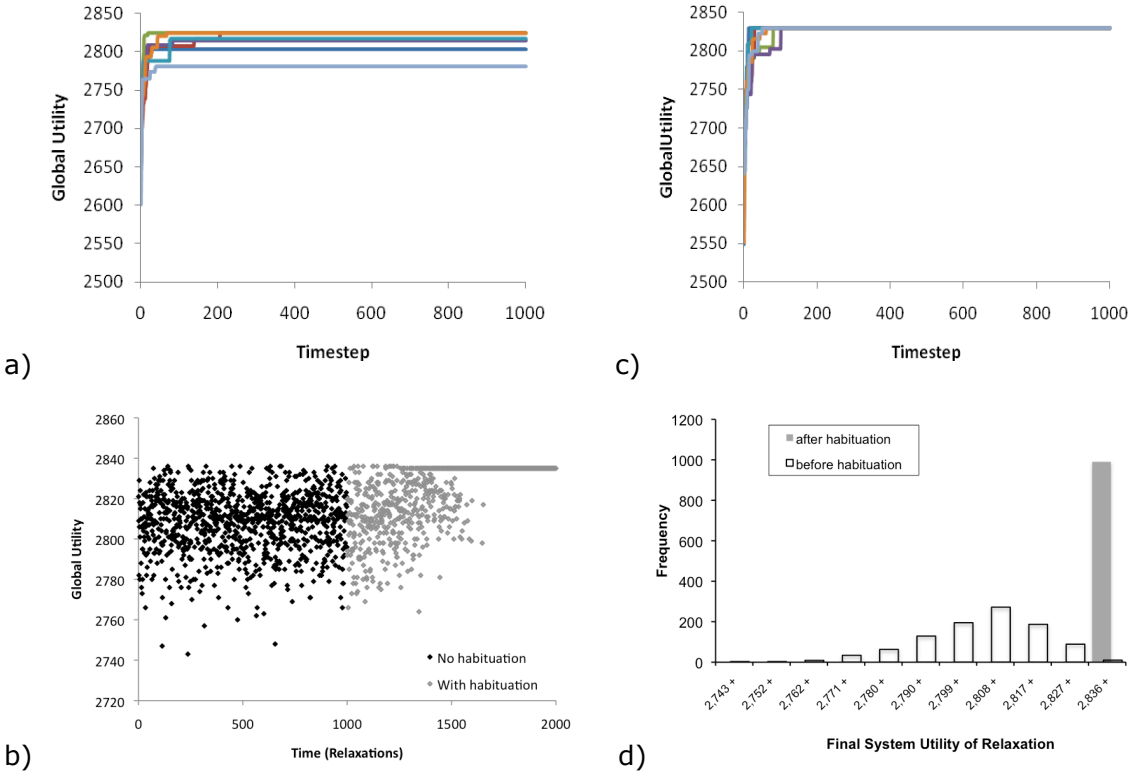


Fig.1. Behaviour of the system using default (no habituation) and habituating agents. a) Some example trajectories of system behaviour before habituation – each curve represents one relaxation (N=100, relaxation length 10N) – vertical axis is the total system utility (G, Eq 2); b) utilities of attractor states visited (i.e. end points of curves like those in (a)) without habituation (relaxations 1-1000) and during habituation (relaxations 1001-2000, r=0.0005); c) example trajectories after habituation; d) histogram of attractor utilities before habituation (relaxations 1-1000) and after habituation (relaxations 2001-3000), showing that after habituation the system reliably finds one of the highest total-utility configurations from any initial condition.

Results collected for 50 runs (each consisting of 1000 relaxations before habitation, 1000 relaxations during habituation and 1000 relaxations after habituation) show that with the current parameters, the global utility of system configurations found after habituation is on average in the 93rd percentile of global utilities of system configurations found before habituation. While this represents a considerable increase in the likelihood of finding a high utility system configuration, it is clear that with the current learning rate (r=0.0005) habituation will not always cause the system to ultimately settle at the global optimum. However, it is important to note that this is simply due to the learning rate used; with a sufficiently low learning rate, after habituation the system will only ever find the global optimum utility configuration (13,15).

Discussion

Adaptive networks

An agent system where actions are governed by a perceived utility (rather than the true utility) is formally equivalent to a system where actions are governed by a new network of constraints (rather than the original network of constraints) (28). Here we have been modelling a system that is fully connected with coordination and anti-coordination games played on the edges of that network. This is equivalent to a weighted network, where edges are weighted by $\omega_{ij}=\pm1$, and all games are coordination games ($\alpha=1$, $\beta=0$) with pay-off $\omega_{ij}U_{ij}$. (i.e. each of the table entries in U_{ij} is multiplied by the scalar ω_{ij}). The structure of the games defined by the pay-off matrices is thus converted into the connections of the network (with identical pay-off matrices). Further, the addition of a preference matrix (restricted to the limited form investigated here) is equivalent to an alteration of this weighting; specifically, $(\omega_{ij}+k_{ij}r)U_{ij}$, where r is the learning rate (as above) and k_{ij} is the number of times agents i and j have been coordinated in the past minus the number of times they have been anti-coordinated (note that k_{ij} will always equal k_{ji}, ensuring that the connections remain symmetric if they start symmetric). Thus, although conceptually contrasting, changing the perception of pay-offs for agent i via a preference matrix is functionally identical to altering the connection strengths between the agents. We chose not to introduce the model in these terms, in part because it is important to realise that although an agents' behaviours will be governed by the new connections, the effects on global 'true' utility that we are interested in must be measured using the original connection strengths (13) (it should be clear that if this were not the case it would be trivial for agents to alter connections in a manner that would make satisfying constraints easier for them and thereby increase total utility). Nonetheless, this perspective helps us to connect the current work with studies of adaptive networks (6) where agents on a network can alter the topology (here, connection strengths) of connections in the network. We can thereby understand the system we have illustrated to be an example of how agents on a network can 're-structure' the network in a manner that enhances the resolution of conflicting constraints and thereby global efficiency. Other works in this area include that of (7,8) where agents on a network, playing a variety of games, re-wire their links when their utility is low, but keep the local topology unchanged if their utility is high. Although there are several important technical differences with the current work, the basic intuition that agents should alter network topology to make themselves happier (or at least, alter it if they are unhappy) is common to both.

In essence, the form of habituation we model is a very simple form of re-structuring; it simply asserts that connections between agents increase or decrease in strength in a manner that reinforces the current combinations of behaviours observed. The effects of this habituation are put into context by considering the problem at hand: we are dealing with a limited form of global optimisation problem (16) in which local optima (and the global optimum) are created by the inability to resolve many overlapping low-order dependencies (17, 13). When using simple local search on this problem (i.e. agents without habituation), there is only a small probability of finding configurations with high global utility (Fig.1.a and b); however, they are found nonetheless. Habituation outcompetes local search, not by finding new configurations of absolute higher utility (although this may occur in some cases), but instead by progressively increasing the *probability* of finding high utility configurations, until only one configuration is ever found (which is very likely to be one of high utility). We can therefore view habituation as a mechanism that gradually transforms the search space of the problem from one with many varied local optima, to one with a single (and very likely high utility) optimum, which will always be reached; furthermore, it does so via a simple distributed strategy.

Specifically, although it is not immediately obvious from a static analysis of the connection matrix which connections should be increased and which decreased in order to cause selfish agents to solve the problem better, the necessary information is naturally revealed by allowing the system to repeatedly settle to local optima and reinforcing the correlations in behaviours so created. These correlations are created by the connections of the original network in an indirect manner. For example, a particular constraint may often remain unsatisfied in locally optimal configurations even though the direct connection defining this constraint states that it is just as valuable to satisfy it as any other connection. Then if a constraint is often easily satisfied its importance is strengthened, if it is equally often satisfied and unsatisfied it remains unchanged on average, and when agents are on average unable to satisfy it its importance is weakened and eventually its sign can be reversed. This causes the system to, gradually over time, pay more attention to the connections that can be simultaneously satisfied and weaken or soften the constraints that cannot be satisfied. One way to understand the result of this adaptive constraint relaxation/exaggeration is that agents become specialists, i.e. selectively attuned to some constraints more than others. That is, whereas the default agents are generalists who persist in trying to satisfy all constraints whether satisfiable or unsatisfiable, habituating

agents, through the self-organisation of the behaviours on the network, come to specialise in a manner that 'for their own comfort' (i.e. for the immediate increase of their perceived utility) fits together better with one another but thereby actually resolves more of the system constraints in total.

Self-structuring adaptive networks, neural network learning and associative memory

How this type of adaptive network, with very simple, local modification of connections, comes to maximise global utility can be explained formally using theory from computational neuroscience. Specifically, the behaviour of the network of default agents detailed above is identical to the behaviour of the discrete Hopfield network (9) (which is just a bit-flip hill-climber (15)) and when connections between nodes increase or decrease in strength in a manner that reinforces the current combinations of behaviours this is formally equivalent to *Hebbian* learning (13). Hebb's rule, in the context of neural network learning, is often represented by the slogan *neurons that fire together wire together*, meaning that synaptic connections between neurons that have correlated activation are strengthened. This learning rule has the effect of transforming correlated neural activations into causally linked neural activations, which from a dynamical systems perspective, has the effect of enlarging the basin of attraction for the current activation pattern/system configuration. This type of learning can be used to train a recurrent neural network to store a given set of training patterns (9) thus forming what is known as an 'associative memory' of these patterns. A network trained with an associative memory then has the ability to 'recall' the training pattern that is most similar to a partially specified or corrupted test pattern.

Formally, a common simplified form of Hebb's rule states that the change in a synaptic connection strength ω_{ij} is $\Delta\omega_{ij} = \delta s_i s_j$ where $\delta>0$ is a fixed parameter controlling the learning rate and s_n is the current activation of the n^{th} neuron. Here by changing the pay-off matrix of each individual by $k_{ij}(t)rU_{ij}$ where $k_{ij}(t)$ is the correlation of behaviours at time t, we are effecting exactly the same changes. Thus the habituating agents each modify their perceived utilities in a manner that effects Hebbian changes to connection strengths – which they must if these preferences are to mean that this behaviour combination is preferred more. This equivalence at the agent level has the consequence that the system of agents as a whole implements an associative memory. Since this is a self-organised network, not a network trained by some external experimenter, this is not an associative memory of any externally imposed training patterns. Rather this is an associative memory of the configuration patterns that are commonly experienced under the networks intrinsic dynamics – and given the perturbation and relaxation protocol we have adopted, which means that the system spends most of its time at locally optimal configurations, it is these configurations that the associative memory stores.

From a neural network learning point of view, a network that forms a memory of its own attractors is a peculiar idea (indeed, the converse is more familiar (18)). Forming an associative memory means that a system forms attractors that represent particular patterns or state configurations. For a network to form an associative memory of its own attractors therefore seems redundant; it will be forming attractors that represent attractors that it already has. However, in forming an associative memory of its own attractors the system will nonetheless alter its attractors; it does not alter their positions in state configuration space, but it does alter the size of their basins of attraction (i.e. the set of initial conditions that lead to a given attractor state via local energy minimisation).

Specifically, the more often a particular state configuration is visited the more its basin of attraction will be enlarged and the more it will be visited in future, and so on. Because every initial condition is in exactly one basin of attraction it must be the case that some attractor basins are enlarged at the expense of others. Accordingly, attractors that have initially small basins of attraction will be visited infrequently, and as the basins of other, more commonly visited attractors increase in size, so these infrequently visited attractors will decrease. Eventually, with continued positive feedback, one attractor will out-compete all others, resulting in there being only one attractor remaining in the system.

But what has this got to do with resolving the constraints that were defined in the original connections of the system? One might expect, given naïve positive feedback principles, that the one remaining attractor would have the mean or perhaps modal global utility of the attractor states in the original system; but this is not the case (Fig.1.d). In order to understand whether the competition between attractors in a self-modelling system enlarges attractors with especially high total utility or not, we need to understand the relationship between attractor basin size and the total utility of their attractor states. At first glance it might appear that there is no special reason why the largest attractor should be the 'best' (highest utility) attractor – after all, it is not generally true in optimisation problems that the basin of attraction for a locally optimal solution is proportional to its quality. But in fact, existing theory tells us that this is indeed the case (17) for systems that are additively composed of many low-order interactions. Specifically, in systems that are built from the superposition of many symmetric pair-wise interactions, the height (with respect to total utility) of an attractor basin is positively related to its width (the size of the basin of attraction), and the globally optimal attractor state has the largest basin of attraction. One must not conflate, however, the idea that the global optimum has the largest basin, with the idea that it is a significant proportion of the total configuration space and therefore easy to find: In particular, the global optimum may be unique, whereas there will generally be many more attractors that lead to inferior solutions, and importantly, the basins of these sub-optimal attractors will collectively occupy much more of the configuration space than the basin of the global optimum.

Given that high utility attractors have larger basins than low utility attractors, they are therefore visited more frequently and therefore out-compete low utility attractors in this self-modelling system. Thus, (in the limit of low learning rates such that the system can visit a sufficient sample of attractors) we expect that when a dynamical system forms an associative memory model of its own utility maximisation

behaviour it will produce a 'model' with ultimately only one attractor, and this attractor will correspond to the globally optimal minimisation of constraints between variables in the original system (13).

This is not an entirely satisfactory conclusion however. It implies that the system only fixes on the global optimum because the global optimum has already been visited many times in the past. But this is not the full story. A final part of the puzzle is provided by the well-known ability of Hebbian learning to generalise training patterns and create learned attractors that represent new combinations of common features from the training patterns rather than the training patterns per se. In associative memory research the creation of such 'spurious attractors' is generally considered to be a nuisance (18,19), but it in fact represents a simple form of generalisation that is important for our results. Producing new attractor states that are new combinations of features (sub-patterns) observed in the training patterns (20) enables the globally optimal attractor to be enlarged even though it has not yet been visited. Basically, this occurs because when Hebbian learning is applied to a training pattern it not only has the effect of enlarging the basin of attraction for this pattern, but also it enlarges the basin of attraction for all configurations in proportion to how many behaviour-pairs they share in common. The global optimum is, by definition, the configuration that has the most simultaneously satisfied constraints, and this ensures that, on average at least, it tends to share many behaviour combinations in common with locally optimal configurations that have many constraints simultaneously satisfied (but not as many as globally possible).

Lastly on this equivalence, it is essential to recognise how the separation of the timescales for behaviours *on* the network and behaviours *of* the network (i.e. changes to network structure) influence this result. Getting the timescale of the changes to network structure correct is equivalent to the problem of setting the learning rate correctly in a neural network. If connections are modified too slowly then learning is unnecessarily slow. And if learning happens too quickly the network will only learn the first local optimum it arrives at, or worse, if the learning rate is really high, the system could get stuck on some transient configuration that is not even locally optimal. More generally, if most learning happens at or near random initial conditions then the patterns learnt will be similarly random. It is therefore essential that the system is allowed to relax to local optima, and that most learning therefore happens at local optima, so that the patterns learned are better than random. But if the system is not perturbed frequently enough or vigorously enough, and consequently spends all of its time at one or a few local optima, the system will simply learn these attractor configurations and will not generalise correctly.

Limitations and further work

Why would agents be creatures of habit? In this paper we have mandated that (otherwise selfish) agents behave as creatures of habit and examined the consequences of this simple local mechanism on global system behaviour. But we are also interested in the question of whether selfish agents, given the opportunity to alter their preferences according to their own self-interest, would alter them in a Hebbian/habit forming manner. Intuitively, we suggest that this is indeed the case – that forming preferences for the *status quo* is a natural strategy for any agent that favours exploitation over exploration, as any non-teleological agent must.

There is some interesting subtlety involved here however. If an agent's perceptions only alter the *perceived* utility of its actions, and not its *true* utility, then an agent can only assess a proposed change in perception as having some real consequence for its utility if that change in perception causes it to change its behaviours and hence its true utility. Note that when the system is at a locally optimal configuration all changes to behaviours are deleterious, whereas Hebbian changes to preferences never cause a change in behaviour and are therefore neutral. This indicates a preference for Hebbian changes in a somewhat subtle sense. However, when behaviours are discrete (and deterministic) as in the current model, most changes to preferences, either Hebbian or non-Hebbian, will not cause a change in behaviour and will therefore be neutral.

Investigations using alternate behavioural models are therefore being developed elsewhere to address this question. This relates to work we are developing in the context of co-evolving species in an ecosystem where species may evolve the coefficients of a Lotka-Volterra system (21,22) or evolve symbiotic relationships (23). This connects the current work with concepts we refer to as 'social niche construction' (24,25,26,27).

Altruism in populations of self-interested individuals has been well researched (e.g. 29); however, very few previous studies investigate games on adaptive networks. Those that do (7,8) differ in a number of ways from the current model, in that here, we: a) only address one type of game (coordination/anti-coordination games), b) play games in normal form, and c) only allow strategies to be adopted to a best-response strategy, rather than by replication equations.

However, despite the novelty of the current model, there appears to be an important similarity between this and many other game theoretic models (network or otherwise) which observe flourishing altruism. Whether they do so by giving agents memory of their past games (30), allowing 'reputation' (31), rewiring links (7,8) or changing link weightings (15), all of these models promote altruism by giving the system a method of passing information from one game to the next, that is not available in the simple, non-altruistic case. This information passing effectively forms a distributed system-level memory, allowing optimisation over multiple games – a mechanism that unites these disparate mechanisms under a common theme.

Finally, it should be noted that the Hopfield model is not new (9), and its capabilities for Hebbian learning are well known (18). However, here we provide a reinterpretation of the system, staging it in a generic, game-theoretic network scenario. This opens up the possibility of reinterpretation of some of the analytically solved variants of the Hopfield model (e.g. 32,33).

Conclusions

This paper has investigated the effect of a simple distributed strategy for increasing total utility in systems of selfish agents. Specifically, habituating selfish agents develop a preference for coordinating behaviours with those they are coordinating with at the present moment, and henceforth adopt behaviours that maximise the sum of true utility and these preferences. We show that this causes agents to modify the dynamical attractors of the system as a whole in a manner that enlarges the basins of attraction for system configurations with high total utility. This means that after habituation, agents sometimes make decisions about their behaviour that may (at least temporarily) decrease their personal utility but that in the long run increases (the probability of arriving at configurations that maximise) global utility. We show that the habituating agents effectively restructure the connections in the network in a Hebbian manner and thus through the simple distributed actions of each individual agent, the network as a whole behaves in a manner that is functionally equivalent to a simple form of learning neural network. This network improves global adaptation by forming an associative memory of locally optimal configurations that, via the inherent generalisation properties of associative memory, enlarges the basin of attraction of the global optima. This work thereby helps us to understand self-organisation in networks of selfish agents and very simple processes that subtly deviate selfish agents in the direction that maximises global utility without overtly prescribing cooperation or using any form of centralised control.

Acknowledgments: Alex Penn, Simon Powers and Seth Bullock.

References

1. Heylighen, F., Gershenson, C., Staab, S., Flake, G.W., Pennock, D.M., Fain, D.C., De Roure, D., Aberer, K., Wei-Min Shen, Dousse, O. and Thiran, P. (2005). Neurons, viscose fluids, freshwater polyp hydra-and self-organizing information systems. Intelligent Systems, 18:4. 72-86. IEEE.
2. Nettleton, R.W. and Schloemer, G.R. (1997). Self-organizing channel assignment for wireless systems, Communications Magazine, IEEE 35:8. 46-51. IEEE.
3. Smith, J.M. and Szathmáry, E. (1997). The major transitions in evolution, Oxford University Press, USA.
4. Dawkins, R. (2006). The selfish gene, Oxford University Press, USA.
5. Michod, R.E. (2000). Darwinian dynamics: evolutionary transitions in fitness and individuality, Princeton University Press.
6. Newman, M. E. J., Barabasi, A. L. and Watts, D. J. editors (2006). The Structure and Dynamics of Networks. Princeton University Press.
7. Pacheco, J., Lenaerts, T. and Santos, F. (2007). Evolution of Cooperation in a Population of Selfish Adaptive Agents. Advances in Artificial Life, 535–544.
8. Santos, F.C., Pacheco, J.M. and Lenaerts, T. (2006). Cooperation prevails when individuals adjust their social ties. PLoS Comput Biol, 2(10), e140: 1284–1291
9. Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities, PNAS USA, 79 (8): 2554-2558.
10. Crosby, D., Stills, S. and Nash, G. (1971). Love the One You're With. 4 Way Street. Atlantic.
11. Hebb, D.O. (1949). The organization of behaviour. Wiley, New York.
12. Hinton, G.E. and Sejnowski, T.J. (1983). Analyzing Cooperative Computation. In Proceedings of the 5th Annual Congress of the Cognitive Science Society, Rochester, NY.
13. Watson R.A., Buckley, C.L. and Mills, R. (2010). Optimisation in 'Self-modelling' Complex Adaptive Systems, Technical Report, ECS, University of Southampton.
14. Nash, J.F. (1950). Equilibrium points in n-person games. Proceedings of the National Academy of Sciences of the United States of America, 36(1): 48–49.
15. Watson R.A., Buckley, C.L. and Mills, R. (2009). The Effect of Hebbian Learning on Optimisation in Hopfield Networks, Technical Report, ECS, University of Southampton.
16. Wolpert, D.H. and Macready, W.G. (1997). No Free Lunch Theorems for Optimization, IEEE Transactions on Evolutionary Computation 1, 67.
17. Kryzhanovsky, B., Kryzhanovsky, V. and Mikaelian A.L. (2007) Binary optimization: A relation between the depth of a local minimum and the probability of its detection. ICINCO-ICSO 2007: 5-10
18. Hopfield, J.J., Feinstein, D. and Palmer, R. (1983). 'Unlearning' has a Stabilizing Effect in Collective Memories. Nature, 304:158-159.
19. Gascuel, J.D., Moobed, B. and Weinfeld, M. (1994). An Internal Mechanism for Detecting Parasite Attractors in a Hopfield Network, Neural Computation 6:902-915.
20. Jang, J.S., Kim, M.W. and Lee, Y. (1992). A Conceptual Interpretation of Spurious Memories in the Hopfield-type Neural Network. Neural Networks, 1992. IJCNN., International Joint Conference on. 1:21-26
21. Lewis, M. (2009). An Investigation Into The Evolution Of Relationships Between Species In An Ecosystem. MSc Dissertation, ECS, University of Southampton.
22. Poderoso, F.C. and Fontanari, J.F. (2007). Model ecosystem with variable interspecies interactions. J. Phys. A: Math. Theor. 40:8723-8738
23. Watson, R.A., Palmius, N., Mills, R., Powers, S.T., and Penn, A.S. (2009). Can Selfish Symbioses Effect Higher-level Selection? European Conference on Artificial Life 2009.
24. Odling-Smee, F.J., Laland, K.N. and Feldman, M.W. (2003). Niche Construction. The Neglected Process in Evolution. Monographs in Population Biology. 37. Princeton University Press.
25. Penn, A. S. (2006). Ecosystem Selection: Simulation, Experiment and Theory. PhD thesis, University of Sussex.
26. Powers, S.T., Mills, R. Penn, A.S. and Watson, R.A. (2009). Social niche construction provides an adaptive explanation for new levels of individuality (ABSTRACT), Proceedings of Workshop on Levels of Selection and Individuality in Evolution, European Conference on Artificial Life 2009.
27. Powers, S.T., Penn, A.S. and Watson, R.A. (2007). Individual Selection for Cooperative Group Formation. Proceedings of European Conference on Artificial Life 2007. pp. 585-594.
28. Gross, T. & Blasius, B., (2008). Adaptive coevolutionary networks: a review. Journal of The Royal Society Interface, 5(20), 259.
29. West, S.A. and Gardner, A., (2010). Altruism, Spite, and Greenbeards. Science, 327(5971), 1341-1344.,
30. Axelrod, R., (1987). The evolution of strategies in the iterated prisoner's dilemma. Genetic algorithms and simulated annealing, 32–41.
31. Nowak, M.A., Page, K.M. and Sigmund, K., (2000). Fairness versus reason in the ultimatum game. Science, 289(5485): 1773.
32. Coolen, A.C.C. and Sherrington, D., (1993). Dynamics of fully connected attractor neural networks near saturation. Physical review letters, 71(23): 3886–3889.
33. Kryzhanovsky, B., Simkina, D. and Kryzhanovsky, V., (2009). A vector model of associative memory with clipped synapses. Pattern Recognition and Image Analysis, 19(2): 289-295.

Determining the public mood state by analysis of microblogging posts

Johan Bollen[1], Huina Mao[1], and Alberto Pepe[2]

[1]School of Informatics and Computing,
Indiana University, Bloomington, Indiana
[2]Center for Embedded Networked Sensing,
UCLA, Los Angeles, CA
jbollen@indiana.edu,huinmao@umail.iu.edu
apepe@ucla.edu

Extended Abstract

Microblogging is a form of online communication by which users broadcast brief text updates, also known as tweets, to the public or a selected circle of contacts. A variegated mosaic of microblogging uses has emerged since the launch of Twitter in 2006: daily chatter, conversation, information sharing, and news commentary, among others (Java et al, 2007). Regardless of their content and intended use, tweets often convey pertinent information about their authors mood status. As such, tweets can be regarded as temporally-authentic microscopic instantiations of public mood state (O'Connor et al, 2010). Here we perform a sentiment analysis of all public tweets broadcasted by Twitter users between August 1 and December 20, 2008. For every day in the timeline, we extract six dimensions of mood (tension, depression, anger, vigor, fatigue, confusion) using an extended version (Pepe and Bollen, 2008) of the Profile of Mood States (POMS), a well-established psychometric instrument (Norcross et al, 2006; McNair et al, 2003). We compare our results to fluctuations recorded by stock market and crude oil price indices and major events in media and popular culture, such as the U.S. Presidential Election of November 4, 2008 and Thanksgiving Day (see Fig. 1). We find that events in the social, political, cultural and economic sphere do have a significant, immediate and highly specific effect on the various dimensions of public mood. In addition, we found long-term changes in public mood that may reflect the cumulative effect of various underlying socio-economic indicators. With the present investigation (Bollen et al, 2010), we bring about the following methodological contributions: we argue that sentiment analysis of minute text corpora (such as tweets) is efficiently obained via a syntactic, term-based approach that requires no training or machine learning. Moreover, we stress the importance of measuring mood and emotion using well-established instruments rooted in decades of empirical psychometric research. Finally, we speculate that collective emotive trends can be modeled and predicted using large-scale analyses of user-generated content but results should be discussed in terms of the social, economic, and cultural spheres in which the users are embedded.

References

Johan Bollen, Alberto Pepe, and Huina Mao. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. *http://arxiv.org/abs/0911.1583*, 2010.

Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web mining and Social Network Analysis*, pages 56–65, New York, NY, USA, 2007. ACM.

D. McNair, J. P. Heuchert, and E. Shilony. *Profile of mood states. Bibliography 1964–2002.* Multi-Health Systems, 2003.

J. C. Norcross, E. Guadagnoli, and J. O. Prochaska. Factor structure of the profile of mood states (POMS): Two partial replications. *Journal of Clinical Psychology*, 40(5):1270 – 1277, 2006.

Brendan O'Connor, Ramnath Balasubramanyan, Bryan R. Routledge, and Noah A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*, May 2010.

Alberto Pepe and Johan Bollen. Between conjecture and memento: shaping a collective emotional perception of the future. In *Proceedings of the AAAI Spring Symposium on Emotion, Personality, and Social Behavior*, 2008.

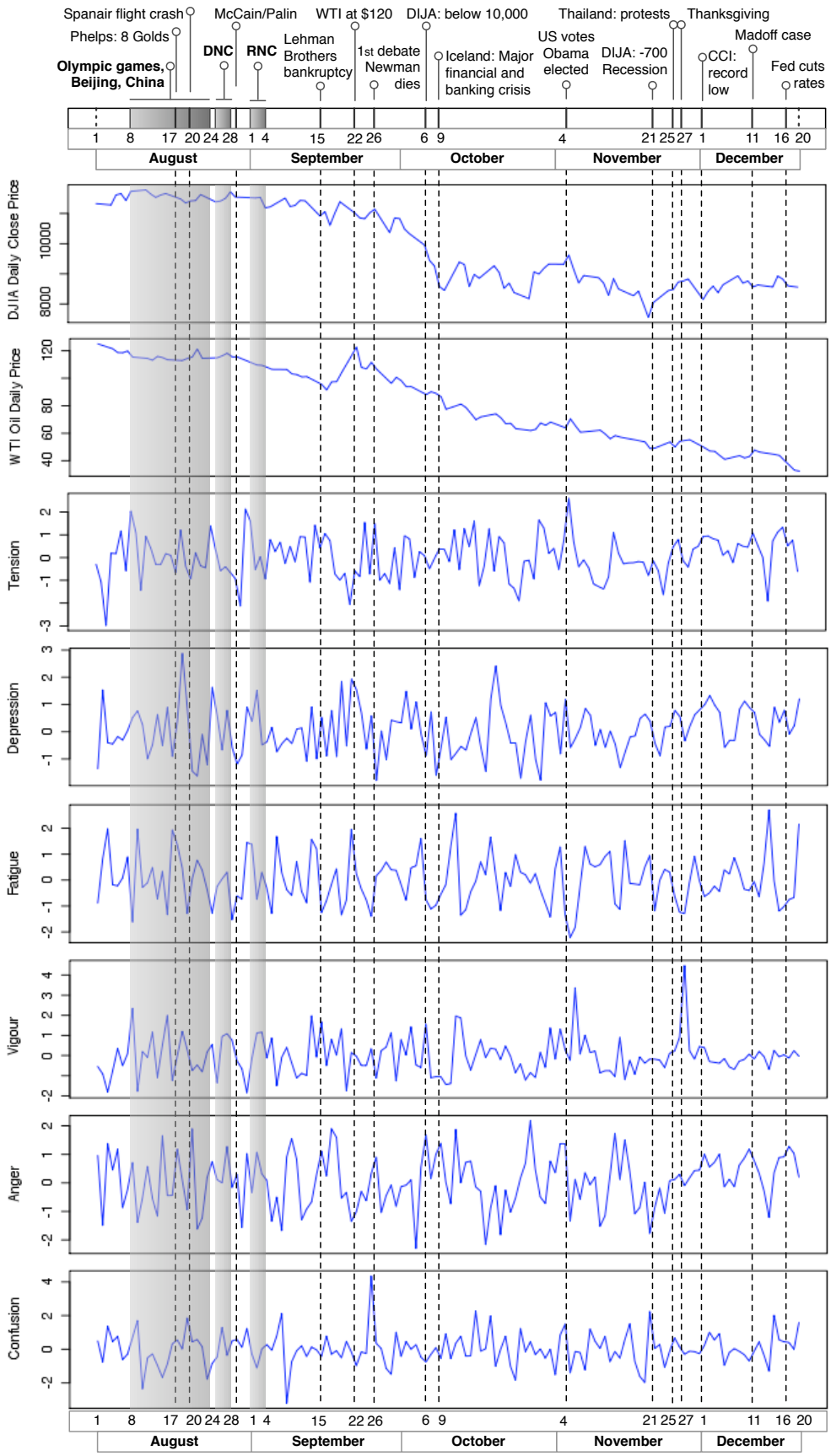


Figure 1: Twitter mood timeseries for 6 mood dimensions measured by extended Profile of Mood States test from August 1 to December 20, 2009. Major events marked in timeline above.

Fostering Creative Emergences in Artificial Cultures

Nicholas Gessler

ISIS (Information Science & Information Studies), Duke University
nick.gessler@duke.edu

Abstract

Empirically, culture is that complex whole which results from the interaction of a multitude of ideas, individuals, behaviors, groups, artifacts, workplaces and architectures, each distributed uniquely and differentially in space and time. *Artificial culture* is the program of describing, understanding and explaining such human complex systems in computer simulations. Several recent conferences in *evolutionary computation* (i.e. *dynamical hierarchies*, *computational synthesis*, and *dynamic ontology*) have focused on the problem of automatically creating novel and compounded emergences in natural and artificial worlds. This paper reviews current progress toward that goal from the perspective of an anthropologist.

Cultural Complexity

Each culture is as different as are its members. Moreover, the minds of individual members of a culture are often filled with different and competing thoughts. To further complicate matters, cognition is unevenly distributed not only among people, but also among their behaviors and the products of their technology. Culture is the totality that emerges, through complex webs of mutual causation at increasing levels of complexity, through *dynamical hierarchical synthesis*, from such seemingly dissimilar things:

> Ideas, and other atomic particles of human culture, often seem to have a life of their own – organization, mutation, reproduction, spreading, and dying. In spite of several bold attempts to construct theories of cultural evolution, an adequate theory remains elusive. The financial incentive to understand any patterns governing fads and fashion is enormous, and because cultural evolution has contributed so much to the uniqueness of human nature, the scientific motivation is equally great. (Taylor & Jefferson, quoted in Gessler, 2003).

> Culture shifts… with kaleidoscopic variety, and is characterized internally not by uniformity, but by diversity of both individuals and groups, many… in continuous and overt conflict in one sub-system and in active cooperation in another. (Wallace, 1961:28).

> Humans create their cognitive powers by creating the environments in which they exercise those powers. (Hutchins, 1995:xvi).

More formally, we might define culture as a complex network of activity through multidimensional multiagent webs of mutual causation, a computational process that is both massively parallel and simultaneous. Culture is the emergent product of the variety of beliefs held by a single individual and the variety of individual behaviors that constitute a society. Complexities of this kind are everywhere and everywhere they defy casual description. Although *complex adaptive systems* are largely intractable to traditional discursive and mathematical representations, the "new sciences of complexity" offer some fresh alternatives. Beginning about 1950, we created computational languages for describing, explaining and understanding these dynamic technicalities. *Artificial culture*[1] is a program that extends the trajectory that began with distributed *artificial intelligence* and grew from *artificial life* to *artificial society*, towards a new social scientific practice. Creative, critical, experimental and empirically informed, a*rtificial culture* is the project of describing the technical complexities of culture in computational terms. Much existing discursive and mathematical cultural theory may be amenable to translation; much may need to be completely reformulated. In short, we need to encode a population of agents, along with their social and physical environments, inside simulations. This enables us to begin to describe, understand and explain the complex causal web of biological and cultural evolutionary processes that distinguished us as humans from our primate ancestors. Experiments of this kind allow us to evaluate the nature of alternative counterfactual "what if" scenarios by observing the entailments of different initial patterns of similarity and difference, different constellations of individual and group (local and global) interaction and different degrees of ideational and material agency. Inspired by the epistemological convergence between evolution and computation (e.g. Rozenberg, et al. 2010), such investigations offer rich new insights into cultural complexity: the individual and society (local versus global), distributed cultural cognition (including the intermediation between humans and their technologies) and the coevolution of the unlimited variety of cultural *things-that-think*[2] and things-that-work. Vital to understanding the evolution of culture is understanding networks of trust, secrecy and deception, the human practice of judging the reliability of other individuals in exchanging matter and information, the practice that builds

[1] A term originally suggested by Michael Dyer.
[2] A phrase originated by the MIT Media Lab.

reputation. *Artificial culture* enables us to describe and experiment with the coevolution of seemingly disparate processes in natural culture and it suggests to us some new critical perspectives from which to evaluate our methods of anthropological inquiry.

Metaphors and Media

Although cultural evolution clearly outpaces genetic evolution in the natural world, it does so only to the degree in which it is freed from the constraints of biological materiality. Cultural change, considered as the reproductive cycle, takes place in seconds, minutes, days, years or decades, whereas human biological change takes at least a decade and a half. In the natural biological and cultural worlds the media of evolutionary transmission behave quite differently: genes reproduce slowly; ideas reproduce quickly. In the artificial world of the computer, whether modeled on a cultural or genetic metaphor, the medium through which evolution unfolds is essentially the same for both. The generations over which evolution unfolds are constrained by the same system clock. Although cultural evolution proceeds more quickly than biological evolution in the natural world, there is no *a priori* reason to believe that cultural processes will be quicker than genetic ones when evolution runs in simulation. Computational algorithms metaphorically modeled on culture may well run faster than those metaphorically modeled on biology, but even if we find this to be true, the argument that what holds true for the natural world must also hold true for the artificial world is simply unsupported (Gessler, 1998). Consequently, when we create simulations using artificial agents, we must critically question the representational analogies and metaphors we use.

Hierarchically synthesized emergences are likely to be more ephemeral and complex in culture than they are in physics, chemistry or biology, and certainly of a completely different nature. In those non-cultural domains, spatial and temporal proximity may be adequate for creating many emergent syntheses. The hierarchical two-fold emergences of monomers to polymers and polymers to micelles, spanning three levels of hierarchical complexity, may be readily visualized as aggregates of dots in three dimensions (Rasmussen, 2002). However, in cultural domains, although space and time may adequately define some features of human interaction (such as households, settlements and trading patterns) other emergent objects are more amorphous and atemporal. Cultural emergences are more difficult to circumscribe. How would a program automatically recognize, capture and repurpose the emergence of a concept such as trade, reciprocity or kinship in an evolutionary simulation? How would a programmer design a graphical user interface to visualize an emergent instance of an institution? In creating *artificial cultures* for social scientific research, one must be careful not to collapse the spatial, temporal and physical constraints of the real natural world into unrealistic artificial world representations. To exacerbate the problem, if one used natural or *artificial culture*s as inspiration for creating populations of synthetic artificial software agents interacting on the Internet, would those same spatial, temporal and physical constraints, that were so important to a science of culture, take on completely different meanings for so-called cultures of software agents? Can they really be "cultural agents" if they are so disembodied? To what extent can software agents be expected to behave like natural human agents? Should they even be modeled on human agents? Or might they better serve our purposes if freed to shape themselves according to their own natures?

Emergence

Among the goals of the "new sciences of complexity," if not of all the sciences in general, is the explanation of emergence in the natural world. In artificial worlds this translates to how to foster emergence in simulations. We often choose to talk about emergence, metaphorically, as levels in a hierarchy. Much research focuses on defining the primitive elements of a simulation at a "lower (local) level" and fostering emergences at a "higher (global) level" of system behavior. Several workshops and labs have focused on creating increasingly higher levels of emergence (Bilotta et al. 2003, Anonymous 2010).

> Given a particular framework, there is a tight correspondence between the complexity of the simple objects used and the system's ability to generate dynamical hierarchies…. The complex systems dogma encourages those studying dynamical hierarchies always to seek models with the simplest possible element. Our *ansatz*, by contrast, encourages us to add complexity to system elements to explore more levels of the hierarchy… Of course, we want to preserve the complex systems dogma to the extent that is possible; we want the simplest possible models of dynamical hierarchies. But we want to stress that the complex systems dogma should not block us from building simulations with enough object complexity to model multilevel dynamical hierarchies successfully. (Rasmussen, 2002:350).

The term "emergence" conflates at least two entangled, yet distinct, meanings. We may talk about it historically (diachronically), as the emergence of everything from the beginning of time to the present, and we may talk about it instantaneously (synchronically), as the structural foundation of the moment. Although the hierarchy of emergence, which we experience as the reality of this instant, may resemble the hierarchy of emergence, which historically enabled us to reach this point, they are qualitatively different. The hierarchy of emergence that we experience as the reality of this instant is in an instantaneous state of self-creation and self-maintenance. From the smallest quark up to the largest quasar, everything in the "now" is held together by emergence. Historically, if agriculture had not first emerged in Mesopotamia, it likely would have emerged somewhere else. We don't need to maintain every level of historical emergence in the present; it has passed. However, if at this instant, sub-atomic particles should change their nature, all instants now and in the future would change dramatically. Scenarios of the destruction of an emergent hierarchy in the "now" make good reading, such as the fictional account of the emergence of a seed crystal of "Ice-Nine," a new solid form of water that melts at 114 degrees (Vonnegut, 1963). However, such collapses at a human scale are common.

It is clear that in the natural world complexity evolves. The *big bang* was arguably simpler than the universe today, the planets more complex than dust from which the condensed and contemporary organisms more complex than cyanobacteria. Historical emergence builds the foundation for the instantaneous emergence of the "now." However, it is unclear to what extent both forms of emergence need to be represented in a simulation to produce persuasive results. My use of the adjective "creative" in the title refers to those emergences which serve as primitives for yet higher levels of emergence. They may perform this function autonomously as long as the causal infrastructure of their creation, from primitive to emergence, is maintained. Alternatively, they may perform it by proxy if their form and functionality can be captured in some other medium and the causal infrastructure of their creation is abandoned[3]. This is particularly likely if the maintenance of their proxy is less costly than the maintenance of the infrastructure of their creation, but other factors may come into play due to the different physical properties of that new medium. The evolution of an efficient route between A and B is replaced by its proxy: a well travelled path, a cleared path, a road. Mutually tolerated theft may lead to trade, a market, a designated market place. The relative advantages of autonomous emergences requiring high maintenance versus proxy emergences requiring low maintenance (as well as intermediate states) depend upon the circumstances in which they are embedded. Again it is interesting to look at science fiction to illustrate the point: Computist Paul Durham has created an artificial world called Elysium. Within it he has programmed two *artificial cultures*, Permutation City and the Autoverse. The inhabitants of Permutation City are modeled on their creators and called Copies. They resemble humans but are constructed of *ad hoc* rules and equations patched in at a high level, without the historical or instantaneous emergent structures that support their "originals" in the natural world. By contrast, inhabitants of the Autoverse, called Lambertians, evolved from a mutated artificial bacterium *in situ* and thus share their computational space with all the historical and instantaneous emergences that created them. Clocks for these two *artificial cultures* tick at different rates. Seven thousand years in Permutation City allow three billion years to pass in the Autoverse. The Autoverse, because of the thick richness of its emergences, evolves, while Permutation City, due to its thin superficiality, does not (Egan, 1994).

At the level of simulating living and human systems, maintaining representations of all the preceding and underlying levels of historical and instantaneous emergence is untenable. In this sense all our social science simulation models float, like Copies, upon a cloud of compromised reality. In creating increasingly immersive and compelling models, in suspending disbelief, we run the risk of ignoring this. In creating so-called "cultures" of software agents, we must be constantly aware that there is nothing underneath that cloud. Perhaps our scientific and commercial agents should be sustained by historical and instantaneous emergence from the bottom-up, evolved solely from the primitives in the computational universe that they inhabit. How might we best create an environment for their constructive coevolution with humans?

In The *Emergence of Everything*, 28 steps of historical emergence are identified (Morowitz, 2002). Little, if any, discussion is devoted to the emergence of the instant. However, it is useful to look at his last six steps to see the scope of the challenge for understanding culture:

- Hominization and Competitive Exclusion.
- Toolmaking.
- Language.
- Agriculture.
- Technology and Urbanization.
- Philosophy.

Culture

"Culture" is a term that has enjoyed a profound freedom in its use and meaning, dancing here and there to the tempo of political correctness and situational ethics. As a mark of status and distinction, it's a thing to which you might aspire to, or oppose. Culture in this sense is what is spoken of in circles of the arts, film, music, literature and fashion. It is the "culture" preserved in museums, galleries, heritage sites, and tourist brochures. In a world where political correctness demands that we respect cultural traditions and differences (c.f. Star Trek's *prime directive*), it is ironically only those things about an "other" people that we find interesting and worthy of preservation from our own perspective that we call "culture." Lightheartedly, "culture" is everything we've got that our primate ancestors and relatives don't. What is it then?

Heralded as "a monumental work of historical and critical analysis," two prominent anthropologists, Alfred Kroeber and Clyde Kluckhohn published *Culture – A Critical Review of Concepts and Definitions* (Kroeber, 1963). Finding the origin of the word in its anthropological and technical sense in 1871, they trace its slow disassociation from the concepts of cultivation and civilization and from this research extract taxonomy of meanings: the margins will therefore be as follows:

- Descriptive: enumeration of content.
- Historical: social heritage or tradition.
- Normative: rules, ways, ideas & values plus behaviors.
- Psychological: a problem solving device, learning, habit, attitudinal relationships among men.
- Structural: pattern and organization.
- Genetic: product or artifact, ideas, symbols, what distinguishes us from animals.

To those engaged in *artificial life* or *artificial societies* the term *artificial culture* evokes a scientific confrontation, the challenge of simulating emergence at the top of the scale of dynamical hierarchical synthesis. To many anthropologists, humanists and social scientists alike, largely unaware of the advances and potentials of *complex adaptive systems* and *evolutionary computation*, the term *artificial culture* stirs up apprehension. Some resent the intrusion of Western technology into the lives of "their people," promoting "cultural relativism," the privileging of "their peoples'"

[3] See Koza et al. 2005 on automatically defined functions, etc.

epistemological and ontological views of the world over that of "Western science." Others express amusement, derisively observing that culture is, by its very definition, artificial, and that the phrase is thus redundant and consequently sterile. Others in the "cultural studies of science" focus on narrative and discursive strategies of explanation. Some use traditional mindsets to study people who write and use simulations, but our goal is to use evolutionary and computational mindsets to study people by writing and using simulations. Opponents of a science of culture frequently call themselves *postmodernists*, not realizing that postmodernism originally did not discount scientific knowledge. A program of *artificial culture* is more closely allied to a *posthumanist* view (Hayles, 1999:2-3).

Reputation

Cognizers are those *things-that-think*, known or unknown, real or imagined, that occupy a person's head. They may also extended beyond a person's head to include observed behavior, material artifacts such as a tally stick, a knotted cord (quipu), an abacus or computer and the larger spatial architecture of a home or workplace. Without limiting the generality of the above, cognizers include beliefs, goals, plans, actions, images, algorithms, languages, observations, performances, desires, emotions, memories, dreams, fantasies, etc. Cognizers, such as those for reputation, and their referents change at different times and in different situations. How are reputations formed, mediated and communicated? How are they manipulated? Which are necessary and sufficient to explain the origins and maintenance of cooperation and competition in a scientific simulation?

The fitness (maintenance and origin) of any naturally or artificially synthesized dynamical hierarchy rests upon the fitness of the structure of its emergences and the fitness of the primitives that give rise to it. In the cultural domain these factors are likely to be widely variable and unevenly distributed in space and time. Cultural organization is conditional upon its individuals being recognized as "same" by one another, and the acquisition by each of information about others. Such information, arising from personal or exchanged experience, constitutes a database of trustworthiness, credibility or "reputation." The human operations of creating, maintaining, manipulating and leveling reputations are complex. But the human individual is not the only level at which reputation resides. Agency may be invoked at many levels in a cultural setting. Below the individual they might include agents in a cognitive society-of-mind. Above the individual they might include groups, their artifacts and behaviors. Reputation is an attribute of agents at all these levels. Thoughts and institutions have their reputations too. Reputation does not come free. Misinformation and disinformation mingle coadaptively with uncorrupted information flow. Reputation percolates through mazes of cognizers, individuals, groups, artifacts and behaviors. Consequently, we should not be surprised to find reputation represented in more than one cultural medium, each adapted to a different niche or competing for the same niche.

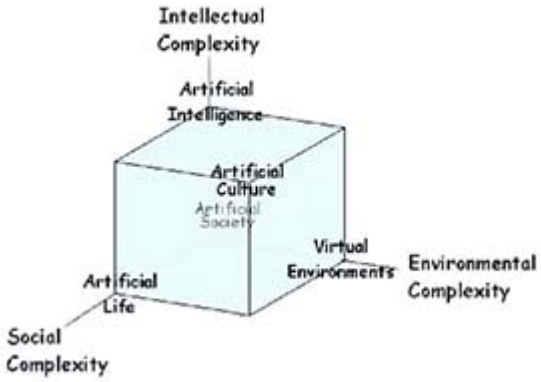
Cognitive reputational schemes, natural or artificial, embodied in the mind or in the material artifactual world, each have concomitant costs and benefits. The cognitive load (cost) of any particular medium of representing reputation is offset by its performance (benefit) in calculating fitness. The cognitive compression of reputation can be beneficial. But as much as cognitive compressions bring with them opportunities for creating yet more highly nested constellations of emergences, literally emergences of emergences, they have a down-side. In compressing, encapsulating and simplifying representations of reputation, they leave behind the mechanisms of their origin and maintenance, and may lose relevance in their new instantiations. Cognitive algorithms are emergent processes and are subject to the same caveats introduced in the previous discussion of historical versus instantaneous emergence. Each time an emergence is captured as a primitive for a higher level of emergence, it looses its infrastructure, and floats like a cloud in a thin atmosphere.

Growth in the new sciences of complexity relies on the intermediation of two lines of research. On the one hand, we must develop an effective means of representing complexity, describing it and calculating its entailments. On the other, we must examine the empirical world with freshly recalibrated eyes. The two are intimately intertwined, for without an adequate language of description and synthesis, complexity will always lie just outside our ken, and without direct confirmation from the real world, complexity will simply be an empty speculation. The psychology of perception implies that in the absence of a formal way of describing and talking about complexity one is likely not to recognize it in the world, and to settle for a simpler misperception. Empirically, things that we do not understand, we often do not see. Innovation in science requires new ways of looking at the world and new ways of looking at old theories and old data. Discovery is seeing what has not previously been seen.

Artificial Culture

Artificial culture has been outlined in several previous publications (Gessler, 1994, 1995, 1996, 2003). It would not only advance cultural theory in anthropology but also provide useful analogies and metaphors for research in *evolutionary computation* (Bäck, et al. 1997). It should provide *evolutionary computation* with new cultural metaphors and analogies which will broaden historical reliance on biological analogies to evolution. For anthropology, it should provide *cultural theory* with a realistic computational framework for describing, synthesizing, experimenting and assessing the entailments of a variety of human complex systems. It would answer the skeptic's taunt, "If you really know how culture works, then build me one!" Culture is technically complex. Should our explanations of it be less so? We can distinguish three major levels of cultural complexity. Within each human head we find a multiagent multimodularity of thoughts insightfully explored in *The Society of Mind* (Minsky, 1985) and *The Adapted Mind* (Borkow, 1992). Among human heads (among individuals) we find a *distributed cultural cognition* (Hutchins, 1995) dispersed among individuals, groups and institutions, as well as in their physical artifacts, workplaces, architectures and settlements. Cognition is rarely the entire picture, so the dynamics of work, matter and energy exchanges among individuals, groups and their technologies may be equally important. *Artificial culture* seeks a minimal

representation of objects and processes, a small core set of functionalities that are essential in explaining the desired aspects of the origins and evolution of culture. It builds upon the practices of *artificial life* and *artificial societies* by imbuing its primitives with a richer mix of intellectual, social and environmental primitives, necessary and sufficient to give rise to cultural complexity. It is useful to visualize *artificial culture* as the corner of a cube, situated in space equidistant from the major axes of *artificial intelligence*, *artificial life* and *virtual environments*. In this position, it distributes the computational load of simulation equally among those three schools of complexity.



Artificial culture can be an experimental vehicle for discovering what it minimally takes to build a culture, a desktop laboratory for evaluating theory against empirical observations by exploring alternative "what if" scenarios. I do not expect it to be predictive in fine detail, but I do expect that it will be insightful in helping us to separate those explanations that are viable from those that are not. If we can develop new approaches to social science theory by building leveraged computational models, models containing the minimal key features that produce maximal results, we can expect to advance both *evolutionary computation* and *cultural theory*.

Evolutionary computation is the convergence of a diverse collection of *evolutionary algorithms*. It embraces the historically separate trajectories of *genetic algorithms*, *evolutionary strategies*, *evolutionary programming*, *cultural algorithms* and *genetic programming* (Fogel, 1998) in a cooperative enterprise to automatically construct *dynamical hierarchies*. Under the rubric of a *computational synthesis*, it seeks, "formal algorithmic procedures that combine low-level building blocks or features to achieve given arbitrary high-level functionality" (Lipson, 2002). *Cultural theory* is an explicitly scientific enterprise in anthropology, a field that has traditionally had roots in both the sciences and the humanities. *Cultural theory* has made measured progress towards a *Science of Culture* (Harris, 1979). Anthropology has also been traditionally divided over the relative importance of cognition versus materiality in cultural causation. Two anthropologists have been particularly influential in articulating these relationships as "cultural materialism" (Harris, 1979 & 1998) and "culture processs" (Binford, 2001)[4]. A third expatriate anthropologist has extended cognition to the physicality of real-world artifacts. Material culture has too often been neglected.

> I hope to evoke… an ecology of thinking in which human cognition interacts with an environment rich in organizing resources… It is in real practice that culture is produced and reproduced… I hope to show that human cognition is not just influenced by culture and society, but that it is in a very fundamental sense a cultural and social process. To do this I will move the boundaries of the cognitive unit of analysis out beyond the skin of the individual person and treat (it) as a cognitive and computational system. (Hutchins, 1995:xiv).

The "holy grail" of *artificial life* research is arguably understanding the bottom-up and top-down exchanges between local and global levels of complex adaptive systems, as each provokes emergences and constraints upon the other. This is also the goal of simulation in sociology, economics, political science, and anthropology.

> (Multiagent systems) have attained a level of maturity where they can be useful tools for sociologists… (They) provide new perspectives on contemporary discussions of the micro-macro link in sociological theory, by focusing on three aspects of the micro-macro link: micro-to-macro emergence, macro-to-micro social causation, and the dialectic between emergence and social causation. (Sawyer, 2003).

Despite our tendency to speak about "the culture" of a people, culture is more than the often-cited "body of shared ideas and behaviors." That "sharedness" is not a sufficient explanation of cultural dynamics. Cross-cutting shared concepts are abundant divergences and disagreements that are often the animating factors in exchanges, negotiations and the flow and quality of goods and information. Culture has eloquently been described as the organization-of-diversity:

> Culture shifts in policy from generation to generation with kaleidoscopic variety, and is characterized internally not by uniformity, but by diversity of both individuals and groups, many of whom are in continuous and overt conflict in one sub-system and in active cooperation in another. (Wallace, 1961:28).

Fortunately, we are not fully enslaved by the languages, words, beliefs or categories that we generate and use to formulate our responses to the world. We recognize and distinguish many more differences in objects and behaviors than we have symbols to express them. In natural language metaphors and modifiers push and pull words in one direction or another to disambiguate their referents and meanings. Natural language is only one system of representation and reasoning, and although we accord it great respect, we must remember that each medium of representation has its distinctive costs and benefits. Each has its specificities and

[4] For opponents of these views see Geertz 1977 and Hodder 2001.

ambiguities, its own channel width and physical and energy requirements. Without pretending to understand how the mind speaks to itself, I think it is clear that thoughts also flow through images and diagrams, gestures and emotions, a gentle touch and a bop on the head. Science is a formalization of these more intuitive media of description and evaluation which grows by inventing new practices of representation and confirmation. Science has become the art of building increasingly reliable, comprehensive and economical representations of the world. Just as some modes of representation are more useful when confined to the mind of a human individual (e.g. meditation), others are more useful for exchanging information between individual minds (e.g. spoken discourse). Mathematics inhabits both our minds and our technologies. Computational simulation alone entrusts representations to the minds of our machines. The downside of the mind of the machine is that it is beyond the ken of those who do not reason with machines. "If you can't wrap your mind around it intuitively, if you can't understand it without a machine, how can you call it an explanation?" It is unlikely that this epistemological myopia will change. I won't attempt a rebuttal here, but will simply echo Jay Forrester's audacious claim:

> It is my basic theme that the human mind is not adapted to interpreting how social systems behave. Our social systems belong to the class called multi-loop nonlinear feedback systems. In the long history of evolution it has not been necessary for man to understand these systems until very recent historical times. Evolutionary processes have not given us the mental skill needed to properly interpret the dynamic behavior of the systems of which we have now become a part.
>
> In addition, the social sciences have fallen into some mistaken "scientific" practices which compound man's natural shortcomings. Computers are often being used for what the computer does poorly and the human mind does well. At the same time the human mind is being used for what the human mind does poorly and the computer does well. Even worse, impossible tasks are attempted while achievable and important goals are ignored. (Forrester, 1971:61).

Human cognition, whether biologically or culturally determined, is a composite of representations, a hall of mirrors, a set of nested Chinese boxes or Russian dolls. The connections among these representations are in a continual state of flux and intermediation. Computer scientists have proposed models of such complex cognitions. Marvin Minsky invokes a cultural (he calls it a "societal") metaphor of mental process. Mind, he says, is a microcosm of society itself, with mental agents vying for control over the individual. Consciousness, he and others assert, sits as an epiphenomenal observer arrogantly taking all the credit.

> We'll show that you can build a mind from many little parts, each mindless by itself. I'll call "Society of Mind" this scheme in which each mind is made of many smaller processes. These we'll call *agents*. Each mental agent by itself can only do some simple thing that needs no mind or thought at all. Yet when we join these agents in societies --- in certain very special ways --- this leads to true intelligence… One trouble is that these ideas have lots of cross-connections. My explanations rarely go in neat, straight lines from start to end. I wish I could have lined them up so that you could climb straight to the top, by mental stair-steps, one by one. Instead they're tied in tangled webs. (Minsky, 1985:17).

Rodney Brooks cogently argues that intelligence and representation are not necessary for purposeful action. He eats away at our conventional wisdom of what comprises intelligence:

> The so-called central systems of intelligence… (are) perhaps an unnecessary illusion… (Perhaps) all the power of intelligence (arises) from the coupling of perception and actuation systems. (Brooks, 1999:viii) The basic idea (of the first model) is that perception goes on by itself, autonomously producing world descriptions that are fed to a cognition box that does all the real *thinking* and instantiates the real *intelligence* of the system. The thinking box then tells the action box what to do, in some sort of high-level action description language. (The second model) completely turns the old approach to intelligence upside down. It denies that there is even a box that is devoted to cognitive tasks. Instead it posits both that the perception and action subsystems do all the work and that it is only an external observer that has anything to do with cognition, by way of attributing cognitive abilities to a system that works well in the world but has no explicit place where cognition is done. (Brooks, 1999:x).

Computational views of mind and culture offer new challenges to both social and computer science. The anthropologist may frame cultural explanations using advanced computational modeling. The evolutionary computist may invoke the complexities of culture in designing new algorithms for creativity and optimization.

Anthropology ambitiously makes claim to the entire domain of human cultural evolution, from our primate ancestors through small-group hunter-gatherers to civilized society and the global institutions of our present. It also often advocates a holistic view of culture. Consequently, anthropologists have repeatedly tried to transcend short-term historical particulars by contemplating the major factors that advanced our cultures to their present reflexive state of complexity (Boyd & Richerson, 1988, Johnson & Earle, 1988). A no less ambitious book attempting to find commonalities among all "Living Systems" was published a decade earlier. It won this praise from Margaret Mead:

> Scientists, from anthropologists to political scientists, and all students of living systems will find here a way of looking at changing scales, but comparable problems, which will enormously illuminate and simplify their attempts to relate one level of living system to another. (Miller, 1978: dustcover).

It seems appropriate that half-a-century after the popular acknowledgement of the "computist" and the "thinking machine" (Anon, 1950) and the recent publication of a

milestone book on an *artificial society* known as Sugarscape (Epstein & Axtell, 1996 and Gessler, 1996), we should finally begin to translate this limited discursive theorizing into robust computational models in an effort to create a fledgling *artificial culture*.

A Grand Challenge

Two conferences were recently held on the ontological and epistemological convergences between evolutionary and computational thought. The first was in connection with the Eighth International Conference on Artificial Life in Sydney, a workshop on "Computational Synthesis: From Basic Building Blocks to High Level Functionality". The second was in connection with the American Association for Artificial Intelligence Spring 2003 Symposium in Stanford, a workshop on "Modeling Dynamical Hierarchies in Artificial Life." Based upon discussions at these workshops, the challenge of *artificial culture* should be to explore models of dynamical hierarchical emergence in which selection is free to operate concurrently at different levels of complexity (cognitive agents, individuals and groups). This implies a connectedness between different informational media (ideational, behavioral and physical) as well as a fluid scheme for allocating the membership of agents to a variety of levels. Interactions need to be further mediated by space and time. Within this milieu of connections, reputations will be free to form and flow among individuals, and they will be captured (frozen with some loss of information about their formation) for subsequent reuse. In other words, the simulation must include functionality for the formulation of the reputation of each cognitive, individual and group agent by those same agents, as well as the reliability of that information. Individuals make their own choices of partners or groups with whom to cooperate, based upon their individual beliefs and perceptions of categories of group membership. Individuals are free to display informative, misinformative or disinformative cues about those affiliations and reputations, or not. It is important to explore the coevolution of cultural *things-that-think* and things-that-work: the cognitive, material and energetic exchanges that are the minimal elements of an *artificial culture*. How complex do simulation primitives need to be, how rich do embedded emergences need to be, in order to foster further hierarchical emergences? No one really knows.

A theoretical model is no better than the empirical observations that it attempts to explain. While detailed accurate, precise and repeatable prediction is too much to expect from a minimal *artificial culture*, prediction in the sense of building an insightful envelope of possibilities is a sufficient goal. Anticipating the criticism that such models are only "toy" explanations, I would ask how many of our discursive or mathematical models of social processes are any more than "toy?" The world is always much richer than simulations, and we must strike a balance between what is small and insightful and what is large and cumbersome. In short, our models must be guides to, not substitutes for, the empirical world:

> "That's another thing we've learned from your Nation," said Mein Herr, "map-making. But we've carried it much further than you. What do you consider the largest map that would be really useful?"
>
> *"About six inches to the mile."*
>
> "Only six inches!" exclaimed Mein Herr. "We very soon got to six yards to the mile. Then we tried a hundred yards to the mile. And then came the grandest idea of all! We actually made a map of the country, on the scale of a mile to the mile!"
>
> *"Have you used it much?" I enquired.*
>
> "It has never been spread out, yet," said Mein Herr: "the farmers objected: they said it would cover the whole country, and shut out the sunlight! So we now use the country itself, as its own map, and I assure you it does nearly as well. (Carroll, 1982:727).

After nearly two decades of archaeological, ethnohistorical and ethnographic research among the Haida hunter-fisher-gatherers of the Pacific Northwest Coast, I could find no adequate single-cause explanation of culture change. Various lines of empirical investigation show abundant evidence for complexly shifting factors coming into play from pre-European contact days (circa 1750) to the present, a period of 250 years of cultural evolution. Early records were limited in scope, and observers "spun" assorted biases into their observations, but there are many clear indications of tipping-points and small events leading to major structural changes. Historical specificities continually spawn irreversible emergences, echoing the properties of chaotic systems: sensitivity to initial (and subsequent) conditions.

Clearly, developing a program of *artificial culture* will not be an easy undertaking. No single implementation of a simulation is likely to address more than a few of the unique processes extant in cultural evolution. Nevertheless it is important to develop examples of how these processes build creative emergences culminating in the variety of complex cultural systems we see today. Although the origins of culture may be traced back to our hominid ancestors 4.4 million years ago and are beyond the scope of this paper, Lovejoy's articulation of an "emergent adaptive suite" of causally interrelated processes is prescient (Lovejoy, 2009) precisely because these processes break the boundaries among biology, behavior and technology, all arguably elements of proto-culture. In much the same way, the processes of culture and emergence that I have discussed form a culturally emergent adaptive suite. What we initially need are simulations which explore information processing and storage across media (intermediation), matter and energy processing and storage across industries (technology) and patterns and modalities of emergence across levels (creative emergence). Researchers in *evolutionary computation* will often tell you that breaking a problem into simpler modules precludes much of the potential for finding optimal solutions for the larger problem. Creativity and innovation in evolution often result from finding and exploiting unlikely coevolutionary interactions. A striking example is endosymbiosis, the evolution from prokaryotes to eukaryotes as the symbiotic inclusion of one species inside the body of another. When boundaries become permeable, causation may become complicated. Evolutionary computists Karl Sims, John Koza and David Fogel have casually characterized the code underlying successfully evolved complex entities as

unintelligible, incoherent and diffuse[5]. Perhaps culture is no less messy underneath.

The grand challenge is to synthesize a system rich in the physicality of its components, letting boundaries dynamically evolve with minimal human intervention. In order to accomplish this, a minimal *artificial culture* should be seeded with a population of individuals, each with the properties of age, sex and parentage, and situated in a physical environment with both space and time. Each should initially have four potentially competing goals: food, shelter, security and reproduction. Cooperative associations should be free to form among causal agents at the cognitive, individual and group levels. At each level a dynamically derived fitness value should be computed. As individuals and groups interact, hierarchical selection is likely to emerge, although it may be difficult to identify because of the shifting boundaries of the units of selection. Fitness advantages and disadvantages should accrue to each level of selection. Social structures would likely form around basic friendship and kinship-derived privileges and obligations, *theories of mind*, observed behaviors, as well as the accrued prestige, credit ratings and reputations of individuals and groups. Information acquired first-hand or second-hand from individuals should be tagged as such. Information about information, in expectation that the reputation of information will also be an important commodity, should also be kept. The perception of boundaries among associated cognitions, individuals, groups and artifacts are expected to be different for each individual.

I hope that incorporating many of these processes into simulations which exhibit limited historical and instantaneous emergence will help to foster proxied (intermediated) creative emergences, offering new rungs on which cultural theory may climb to look back upon the evolution and origins of culture.

References

Anonymous (1950). The Thinking Machine. *Time*, 55(4):54-60.

Anonymous (2010). Cornell Computational Synthesis Laboratory. Electronic document accessed 20 June 2010: http://ccsl.mae.cornell.edu/front

Bäck, T., Fogel. D. B. and Michalewicz, Z., editors (1997). *Handbook of Evolutionary Computation.* Oxford University Press, NY.

Barkow, J. H., Cosmides, L. and Tooby, J. editors (1992). *The Adapted Mind – Evolutionary Psychology and the Generation of Culture.* Oxford University Press, NY..

Bilotta, E., Groß, D., Smith, T. et al. (2003). Modelling Dynamical Hierarchies in Artificial Life. In *Alife VIII: Workshop Proceedings*. Electronic document accessed 20 June 2010: http://alife.org/alife8/workshops.html

Binford, L. R. (2001). *Constructing Frames of Reference – An Analytical Method for Archaeological Theory Building Using Ethnographic and Environmental Data Sets*. University of California Press, Berkeley, CA

Boyd, R. and Richerson, P. J. (1988). *Culture and the Evolutionary Process.* University of Chicago Press, IL.

Brooks, R. A. (1999). *Cambrian Intelligence. The Early History of the New AI.* MIT Press, Cambridge, MA.

Carroll, L. (1982) first published 1893. Sylvie and Bruno Concluded. In *The Complete Illustrated Works*. Gramercy Books, NY.

Egan, G. (1994). *Permutation City*. Millennium, London.

Epstein, J. M. and Axtell, R. (1996). *Growing Artificial Societies: Social Science from the Bottom Up.* MIT Press, Cambridge, MA.

Fogel, D. B. (1998). *Evolutionary Computation, The Fossil Record: Selected Readings on the History of Evolutionary Algorithms.* IEEE Press, NY.

Forrester, J. W. (1971). Counterintuitive Behavior of Social Systems. In *Simulation*, February: 61-73.

Geertz, C. (1977). *The Interpretation of Cultures*. Basic Books, NY.

Gessler, N. (1994). Artificial Culture. In Brooks, R. and Maes, P., editors, *Artificial Life IV*, pages 430-435. MIT Press, Cambridge, MA.

Gessler, N. (1995). Ethnography of Artificial Culture – Specifications, Prospects and Constraints. In McDonnell, J. R., Reynolds, R. and Fogel, D., editors, *Evolutionary Programming IV*, pages 319-331. MIT Press, Cambridge, MA.

Gessler, N. 1996. Book Review: Growing Artificial Societies by Joshua Epstein and Robert Axtell, MIT Press, Cambridge 1996. In *Artificial Life*, 3(3):237-242.

Gessler, N. (1998). Skeumorphs and Cultural Algorithms. In Porto, W. Saravanan, N. Waagen, D. and Eiben, A., editors, *Evolutionary Programming VII*, pages. 229-238. Springer-Verlag, Berlin.

Gessler, N. (2003). Evolving Cultural Things-That-Think. In *Computational Synthesis: From Basic Building Blocks to High Level Functionality*. Technical Report SS-03-02. AAAI Press, Menlo Park, CA.

Harris, M. (1979). *Cultural Materialism – The Struggle for a Science of Culture.* Random House, NY.

Harris, M. (1998). *Theories of Culture in Postmodern Times.* AltaMira Press, MD.

Hayles, N. K. (1999). *How We Became Posthuman – Virtual Bodies in Cybernetics, Literature, and Informatics.* University of Chicago Press, IL.

Hodder, I., editor (2001). Archaeological Theory Today. Polity Press, Oxford, UK.

Hutchins, E. (1995). *Cognition in the Wild.* MIT Press, Cambridge, MA.

Johnson, A. W. and Earle, T. (1988). *The Evolution of Human Societies: From Foraging Group to Agrarian State.* Stanford University Press, CA.

Koza, J., Keane, M. A., Streeter, M. J., et al. (2005). *Genetic Programming IV*. Springer, NY.

Kroeber, A. L. and Kluckhohn, C. (1963). Culture – *A Critical Review of Concepts and Definitions*. Vintage Books, NY.

Lipson, H., Antonsson, E. K. and Koza, J. R. (2002). Computational Synthesis – from Building Blocks to High Level Functionality. Electronic document accessed 20 June 2010: http://www.aaai.org/Library/Symposia/Spring/ss03-02.php

Lovejoy, C. O. (2009). Reexamining Human Origins in Light of *Ardipithecus ramidus*. In *Science*, 326:74 ff.

Miller, J. G. (1978). *Living Systems.* McGraw-Hill, NY.

Minsky, M. (1985). *The Society of Mind.* Simon and Schuster, NY.

Morowitz, H. J. (2002). *The Emergence of Everything – How the World Became Complex.* Oxford University Press.

Rasmussen, S., Baas, N., Mayer, B., Nilsson, M. and Olesen, M. (2002). Ansatz for Dynamical Hierarchies. In *Artificial Life* 7(4):329-353.

Rozenberg, G., Bäck, T. and Kok, J. N., editors (2010). *Handbook of Natural Computing*. Springer.

Sawyer, R. Keith. 2003. Artificial Societies – Multiagent Systems and the Micro-Macro Link in Sociological Theory. In *Sociological Methods and Research*, 31(3):325-363.

Vonnegut, K. (1963). *Cat's Cradle*. Holt, Rinehart and Winston, NY.

Wallace, A. F. C. (1961). *Culture and Personality.* Random House, NY.

[5] Personal communication 2007.

The Human Use of Living Technology: Toward a Cultural Understanding of Robots

Jean-Paul Peronard[1]

[1]Department of Marketing and Management
University of Southern Denmark
jpp@sam.sdu.dk

Abstract

This paper describes the concepts of concretization and social imaginary and argues that they provide helpful hints to a more advanced cultural understanding of robot technology, so the human system may exploit the full potential of such living technology.

Introduction

New advances in robot technology have made self-managed automated machines possible that not only do things for humans, but also affect the way we relate to the world, perceive ourselves and other people (Turkle, 2006). Such machines have been called "social robots" (Hegel, Muhl, and Wrede, 2009) and also viewed as being a part of the concept of "living technology" (Bedau, et al. 2010). Social robots are gradually finding their way into the healthcare sector where the prospects are fascinating, compelling, and controversial (Shibata and Wada, 2008; Dautenhahn, 2007). In this process, culture plays an important role in the selection and adoption of technological solutions (Steers, Meyer and Sanche, 2008).

The way a society responds to the potential use of robots in healthcare is influenced by norms, values, and symbols that often remain unquestioned by the parties involved even though they are profoundly influential. However, there is still not a proper theorization of robot technology and culture, although research in recent decades in various ways has revealed facets of this research. Concepts of *concretization* (Simondon, 1990) and *social imaginary* (Castoriadis, 1987) provide helpful hints toward a more advanced cultural understanding of robot technology. These concepts offer the first step to making a link between culture and non-human agents, so the human system may take full advantage of living technology.

The existence of technical objects

Gilbert Simondon is among the first to discuss the relation between culture and technical developments (Simondon, 1990). He argues that resistance to technology is embedded in culture. His thesis is that because technical objects evolve independently of the human they create tension between innovation and culture. This prevents some innovations from developing. To exploit the full potential of technology, Simondon believes it is necessary to reintegrate culture and technology.

In his theory *mekanology*, the technical item is socially autonomous as it undergoes a technical genesis, called the *concretization* process. It is a process in which a technical object goes through a series of stages, from the abstract to the concrete and thereby refines its functionality, independent of human interference. The genesis of a technical object is not to cover a specific need, but to create synergy and convergence between technical objects' functions. For example, a robot is an expression of such a technical process of creation (innovation) more than the result of specific human needs (Simondon, 1990).

Simondon divides the technical development into three levels: the *technical element*, the *technical individual*, and the *technical ensemble* (a network). The technical element constitutes the artifact and may be organized in relation to other elements. It is not a tool and does not have an associated environment. It can be compared to an organ in a body. However, contrary to an organ in the human body an element can be separated from the technical individual. Technical individuals are combined elements and associated environments. Examples of such individuals are a house, car or computer. The technical ensemble is a network of technical subjects (subsystems) which are arranged in relation to the outcome of their function. An example of a technical ensemble is a nursing home. It is on the ensemble level where the technical and economic are combined.

Following the theory of mekanology, living technology has reached a level, where it can be comprehended as something in-between a technical individual and a technical ensemble, but the final step, network integration, can only be achieved if it is constituted in the human sphere. With reference to Simondon this poses some important questions: When the technology becomes as complex, integrated, and far-reaching as living technology, what happens to human control and the impact technology has on the perception of control?

To elaborate on these points we can bring in Cornelius Castoriadis' perspective insofar that living technology is both

an outcome and product of culture, i.e. part of the social-historical imaginary.

The social imaginary of living technology

Living technology (e.g. robots) is considered highly *equivocal* (Weick, 1990), which means that the *valorisation* of such a new technology is neither fully given in advance nor once and for all, but is an on-going result of social interactions and discourses (e.g. Castoriadis, 1984). What will be considered "good" technologies, are those that meet the socially negotiated needs and not necessarily those technologies, which the experts see as the most suitable. In light of this, different technological solutions for various problems are related to the needs and wants of the people in a given culture, be it on societal, organizational, or local levels.

Not only is robot deployment negotiated socially among and between actors in healthcare, but it is also *recursive* because it both enables and constrains individual action as well as providing the precondition for the production of new types of technologies (Orlikowsky, 1992; Morin, 1986). At the same time, the social structure will be affected and reorganized within that circle of innovation.

Thus, technology should be seen as an institution of society that produces meaning in the same way that language does since both constitute the human as well as the real-rational world (Castoriadis, 1984, p. 240). This implies an imaginary dimension in the application and use of robot technology. According to Castoriadis *significations* that individuals and collectives use to make sense of reality both constitutes the physical world and organizes social life (Castoriadis, 1987, p. 146). The social imaginary significations are a form of self-creation and organizing that "have to confer meaning on everything" (Bouchet, 2007, p. 36) and are visible through the prevailing societal discourses (Castoriadis, 1987).

Conclusion

A cultural understanding of living technology may be seen as a function of the social imaginary. Meanings concerning living technology are generated by cultural images that reinforce these meanings. Uncertainty about the technology will stabilize through the negotiation of meanings, which aims at achieving rhetorical closure and societal consensus. This is a process that goes on parallel to the concretization process, but needs to be coupled together in order to get the full potential of living technology.

References

Bedau, M., McCaskill, J., Packard, N., and Rasmussen, S. (2010). Living Technology: Exploiting Life's Principles in Technology. *Artificial Life*, 16(1): 89-97.

Castoriadis, C. (1984). *Crossroads in the Labyrinth.* Cambridge, MA: MIT Press.

Castoriadis, C. (1987). *The Imaginary Institution of Society.* (K. Blamey, Trans.) Cambridge: Polity Press.

Dautenhahn, K. (2007). Socially intelligent robots: Dimensions of human-robot interaction. *Philosophical Transactions of the Royal Society: Biological Sciences*, 362(1480): 679-704.

Hegel, F., Muhl, C., Wrede, B., Hielscher-Fastabend, M., and Sagerer, G. (2009). Understanding Social Robots. *IEEE*: 169-174.

Morin, E. (1986). *La methode 3 La connaissance de la connaisance, Livre premier: Antropologie de la connaisance.* Paris: Seuil.

Orlikowsky, W. (1992). The duality of technology: Rethinking the concept of technology in organizations. *Organization Science*, 3: 398-426.

Shibata, T., and Wada, K. (2008). Robot therapy at elder care institutions: Effects of long-term interaction with seal robots. In A. Helal, M. Mokhtari, & B. Abdulrazak (Eds.), *The engineering handbook of smart technology for aging, disability, and independence* (pp. 405-418). New York: Wiley.

Simondon, G. (1990). *On the mode of Existance of Technical Objects.* (N. Mellamphy, Trans.): http://nsrnicek.googlepages.com/SimondonGilbert.OnTheModeOfExistence.pdf

Steers, R. M., Meyer, A. D., and Sanche, C. J. (2008, July). National culture and the adoption of new technologies. *Journal of World Business*, 43(3): 255-380.

Turkle, S. (2006). *A Nascent Robotics Culture: New Complicities for Companionship.* Boston: AAAI Technical Report.

Weick, K. (1990). Technology as Equivoque: Sensemaking in New Technologies. In P. Goodman, L. Sproull, & Associates (Eds.), *Technology and Organizations.* Jossey-Bass.

Crowdsourcing, Open Innovation and Collective Intelligence in the Scientific Method: A Research Agenda and Operational Framework

Thierry Buecheler[1], Jan Henrik Sieg[2], Rudolf M. Füchslin[1] and Rolf Pfeifer[1]

[1]Artificial Intelligence Laboratory, Department of Informatics, University of Zurich
[2]Chair for Strategic Management and Innovation, Department of Management, Technology and Economics, ETH Zurich
buecheler@ifi.uzh.ch

Abstract

The lonely researcher trying to crack a problem in her office still plays an important role in fundamental research. However, a vast exchange, often with participants from different fields is taking place in modern research activities and projects. In the "Research Value Chain" (a simplified depiction of the Scientific Method as a process used for the analyses in this paper), interactions between researchers and other individuals (intentional or not) within or outside their respective institutions can be regarded as occurrences of Collective Intelligence. "Crowdsourcing" (Howe 2006) is a special case of such Collective Intelligence. It leverages the wisdom of crowds (Surowiecki 2004) and is already changing the way groups of people produce knowledge, generate ideas and make them actionable. A very famous example of a Crowdsourcing outcome is the distributed encyclopedia „Wikipedia". Published research agendas are asking how techniques addressing "the crowd" can be applied to non-profit environments, namely universities, and fundamental research in general.

This paper discusses how the non-profit "Research Value Chain" can potentially benefit from Crowdsourcing. Further, a research agenda is proposed that investigates a) the applicability of Crowdsourcing to fundamental science and b) the impact of distributed agent principles from Artificial Intelligence research on the robustness of Crowdsourcing. Insights and methods from different research fields will be combined, such as complex networks, spatially embedded interacting agents or swarms and dynamic networks.

Although the ideas in this paper essentially outline a research agenda, preliminary data from two pilot studies show that non-scientists can support scientific projects with high quality contributions. Intrinsic motivators (such as "fun") are present, which suggests individuals are not (only) contributing to such projects with a view to large monetary rewards.

Introduction

The Scientific Method in empirical science is constantly being improved to investigate phenomena, acquire more knowledge, correct and/or integrate previous knowledge. Beyond a constant evolution, several researchers and meta-researchers (e.g., epistemologists and research philosophers) have tried to develop a process view of the main steps conducted in most forms of fundamental research, independent of discipline or other differentiating factors. In the context of this process, many interactions between groups of people and individuals are taking place: e.g., idea generation, formulation of hypotheses, evaluation and interpretation of gathered data, among many others. Furthermore, large project conglomerates (e.g., EU-funded research projects or projects funded through the Advanced Technology Program and others in the U.S., see Lee and Bozeman 2005, p.673ff.) increase the number of such interactions. In many cases, the scientist groups involved self-organize their work and contributions according to their individual strengths and skills (and other measures) to reach a common research goal, without a strong centralized body of control (Melin 2000, Stoehr and WHO 2003, Landry and Amara 1998). The interactions between these individuals and groups can be seen as instances of Collective Intelligence, including consensus decision making, mass communications, and other phenomena (see e.g., Hofstadter 1979).

In what follows, we will select examples of Collective Intelligence, which we base on the following broad definition (Malone et al. 2009, p.2): "groups of individuals doing things collectively that seem intelligent". Collective Intelligence involves groups of individuals collaborating to create synergy, something greater than the individual part (Castelluccio 2006).

Although we will mainly use the generic term "Collective Intelligence", or "CI", we will use an interpretation that is very close to "Crowdsourcing", because we are going beyond the traditional research collaborations (that, of course, are also a form of Collective Intelligence): Crowdsourcing, connoted as "Wikipedia for everything" by the inventor of the term (Howe 2006), has influenced several researchers and practitioners alike. It builds on the concept of User Innovation (von Hippel 1986) among others.

Although there are currently many definitions and similar concepts being discussed in the surrounding space (radical decentralization, wisdom of crowds, peer production, open innovation, mass innovation, wikinomics, and more (Malone 2004, Surowiecki 2004, Benkler 2006, Chesbrough 2003, Leadbeater and Powell 2009, Tapscott and Williams 2008), we will use the following definition of Crowdsourcing:

> "Crowdsourcing is the act of taking a job traditionally performed by a designated agent (usually an employee) and outsourcing it to an undefined, generally large group of people in the form of an open call." (Howe 2010)

For our purposes we understand "Crowdsourcing" as an umbrella term for the nuances indicated by the other terms

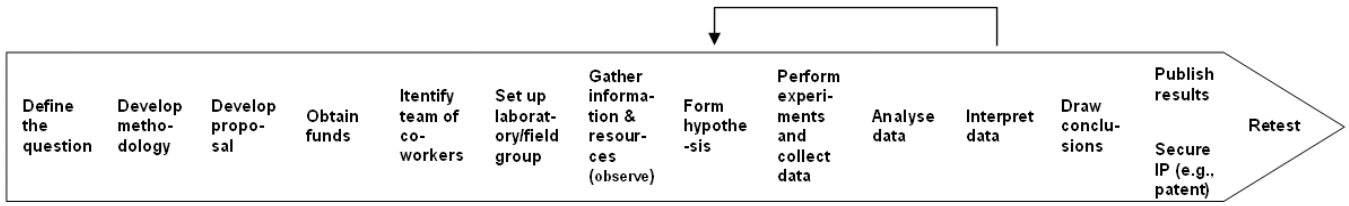


Figure 1 – Simplified Research Process, containing "tasks": the Research Value Chain

Crowdsourcing is a relevant construct for our research because it describes research collaboration that radically enlarges the pool of (potential) scientific collaborators. Research projects, such as NASA's Clickworkers and the "self-organized" research collaboration identifying the cause of the severe acute respiratory syndrome SARS (Stoehr and WHO 2003), go beyond traditional forms of collaboration by embracing electronic communication and cooperation between a very large group of scientists.

The applicability of Crowdsourcing approaches to the solution of scientific problems can be motivated by a simple probabilistic argument: a sufficiently large crowd of independent individuals will, in a majority yes/no vote, decide properly, with high probability, even if the individuals have only a slight bias towards the correct answer. Surowiecki (2004) shows by example that crowd based decision finding also works for questions with answers more complex than yes/no. Moreover, it is known that virtual stock exchanges estimating (betting on), e.g., results of elections deliver surprisingly precise predictions, even if the participants are subject to a broad variety of influences and cannot be regarded as independent. The implementation of Crowdsourcing in a scientific context first requires identifying the type of questions suitable to being answered by a crowd (e.g., strategic decisions that benefit from experience but for which no rational solution scheme exists) and second finding a balance for antagonistic system properties, such as, e.g., communication between agents vs. the independency of their respective decisions. Research areas that provide tools and insights for this optimization task include complex networks, spatially embedded interacting agents or swarms and dynamic networks.

In the following sections, we first propose a simplified process view of the Scientific Method that we use to investigate potential Crowdsourcing opportunities for fundamental research based on the above definitions. Second, we show how mass collaboration (including Crowdsourcing) is already changing the way parties interact in industry and connect this development to science. Third, we develop a framework for analyzing the tasks of the Scientific Method regarding their applicability for Crowdsourcing. After showing some examples from our preliminary analysis, we state important challenges and a research agenda, which investigates these challenges and the applicability empirically.

The Scientific Method as a process

Different fields of research have different approaches to conducting research as a process (see Amigoni et al. 2009 for an example comparing mobile robotics with other sciences). Paul Feyerabend and other well-known meta-scientists criticize every form of standardization, stating that any depiction has little relation to the ways science is actually practiced (see, e.g., Feyerabend 1993). There are, however, elements that are part of almost every research process (either explicitly or implicitly), such as characterizations, hypotheses, predictions, and experiments. We will use a simplified process for empirical science, based on Crawford and Stucki (1999) as a basis for this paper, which we call the "Research Value Chain" (see Figure 1). "Value" is not defined as economic value, but as an "addition to the body of reliable knowledge", rather a social value.

Not all tasks in our Research Value Chain are present in all research projects: After defining the (research) question at hand, a methodology is either developed or chosen. If necessary, a proposal is compiled to obtain funds or other resources. Potentially, a team of co-workers and a laboratory or field group is set up. Next, resources are gathered, hypotheses are formulated (sometimes implicitly), and subsequently experiments are performed which yield data. The data can then be analyzed and interpreted and conclusions may be drawn that may lead to new hypotheses, indicated by the small connecting arrow in Figure 1. The research piece is then published – or, in some cases, the resulting Intellectual Property (IP) is secured – in order to spread the insights, potentially appropriate the investment and enable other researchers to use it as a basis for their further thinking and testing.

Such a process is potentially subject to iterations, recursions, interleavings and orderings.

Why Crowdsourcing in the Scientific Method

Before answering this question, we need to put Crowdsourcing, a process that is described often in a business (or innovation) context, into a research context. Technological advance has often been subdivided into two categories: invention (a scientific breakthrough) and innovation (commercialization of the invention) - a distinction Nelson and Winter (Nelson and Winter 1982, p.263) attribute to Schumpeter (1934). For this purpose, we demonstrate an important development taking place throughout technologically advancing societies:

Industries are on the verge of a significant change in the way they innovate. Over the past decade, the Internet has enabled communities to connect and collaborate, creating a virtual world of Collective Intelligence (Malone et al. 2009, Lane 2010). Von Hippel (2005) states that for any group of users of a technology, a large number of them will come up with

innovative ideas. What began as a process in business is also being observed in science. Discussions on "Citizen Science" (Irwin 1995) and "Science 2.0" (Shneiderman 2008) suggest the same effects are relevant for fundamental research practices.
Chesbrough provides an example in the consumer sector where a form of Crowdsourcing (in this case, he calls it "open innovation") has proven successful and which seems to be applicable to fundamental research as well:

> "In 1999, Procter & Gamble decided to change its approach to innovation. The firm extended its internal R&D to the outside world through an initiative called Connect and Develop. This initiative emphasized the need for P&G to reach out to external parties for innovative ideas. The company's rationale is simple: Inside P&G are more than 8,600 scientists advancing the industrial knowledge that enables new P&G offerings; outside are 1.5 million." (Chesbrough 2003)

Schrage (2000) states innovation requires improvisation; it is not about following the rules of the game, but more about rigorously challenging and revising them, which is consistent with criticism of any standardization of the Scientific Method. An expert scientist (or an expert group) needs to manage (and perhaps improvise) the overall process and aggregate potential input from "the crowd". But the crowd doesn't necessarily have to be composed of experts.
(Maintained) diversity is an essential advantage of crowds. Scott E. Page has created a theoretical framework to explain why groups often outperform experts. The results of several experiments formed the basis for the "Diversity Trumps Ability" Theorem (Page 2008): Given certain conditions, a random selection of problem solvers outperforms a collection of the best individual expert problem solvers due to its homogeneity. The experts are better than the crowd, but at fewer things. Friedrich von Hayek stated in 1945 that nearly every individual "has some advantage over all others because he possesses unique information of which beneficial use might be made" (von Hayek 1945).
Although certain universities have been trending towards a more entrepreneurial model for more than two decades, (Etzkowitz 1983, Etzkowitz et al. 2000, Bok 2003) we still regard them as being in the not-for-profit field, interested in spreading knowledge throughout society. Crowdsourcing has been successfully used in the business environment for creating economic value. To our knowledge, there is no systematic study investigating the applicability of Crowdsourcing in not-for-profit basic research (as conducted in traditional universities).

This paper aims to help fill this gap by testing the use of Crowdsourcing in the Scientific Method in order to maximize the knowledge that can be gained and dispersed, reduce necessary resources, and other potential contributions to the fundamental research process. Crowdsourcing is regarded as a tool within the Scientific Method, not a substitute for it.

For the remaining sections of this paper, we will use the terms "Collective Intelligence" and "Crowdsourcing" interchangeably for "using a large group of individuals to solve a specified problem or collect useful ideas".

A Framework for integrating Collective Intelligence in the Scientific Method

We combine frameworks from prior research with our own thinking in order to systematically analyze the tasks comprising the Research Value Chain.
The first framework, drawn from MIT's Center for Collective Intelligence (Malone et al. 2009), uses the genome analogy to map the different elements of a Collective Intelligence task to 4 basic "genes": Who, Why, What, How.
These basic questions are further divided into subtypes that help structure the problem at hand in a mutually exclusive, collectively exhaustive manner with respect to Collective Intelligence.

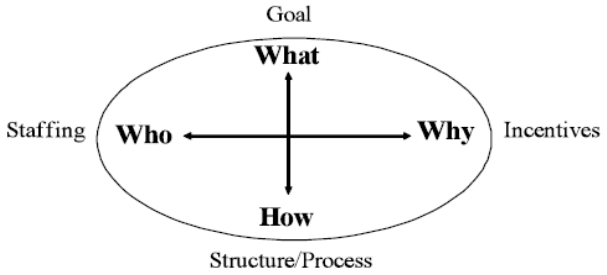


Figure 2 – MIT's Collective Intelligence genes (Malone et al. 2009)

The following list shows the hierarchy of the "genes". For a detailed description, please consult the original paper.

Who
- Crowd, Hierarchy

Why
- Money, Love, Glory

What, How
- Create
  - Collection, Contest, Collaboration
- Decide
  - Group Decision
    - Voting, Averaging, Consensus, Prediction Market
  - Individual Decisions
    - Market, Social network

However, before a task can be crowdsourced, it needs to be tested as to its suitability for Collective Intelligence. Here we use a design principle called the "Three-constituents principle" from Artificial Intelligence (see e.g., Pfeifer and Scheier 1999). It states that the ecological niche (environment), the tasks at hand and the agent must always be taken into account when investigating or modeling intelligent behavior. Therefore, for every task in our Research Value Chain, we analyze the environment (e.g., research institute location, funding situation), the agent (e.g., researchers'

tenure, culture, particularistic characteristics) and the task. To analyze the likelihood and potential success for collaboration given the environment and the agent, we use the moderating variables identified by Lee and Bozeman (2005).
The following list (not exhaustive) shows variables that moderate the relationship between scientific productivity (normal and fractional journal publications) and collaboration in a scientific setting (several of them backed by other studies):

Agent
Career age, Job satisfaction, Collaboration strategy, "Cosmopolitan scale" (collaborating with those outside the proximate work environment)

Environment
Log of current grants, Field/discipline, Number of collaborators

Figure 3 gives a schematic overview over all the relationships of the different elements of our framework.

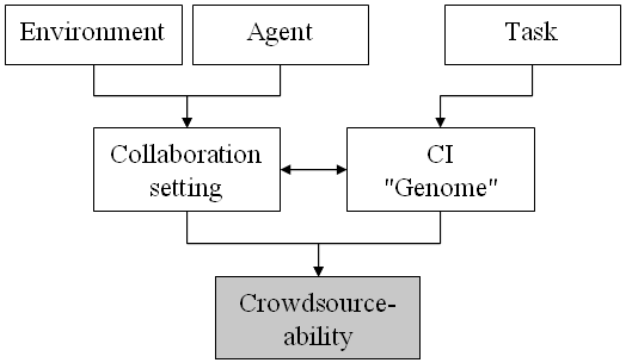


Figure 3 – Framework for assessing Crowdsource-ability of a task

In addition to the potential of crowdsourcing a certain task from the Research Value Chain, we assess its feasibility given limited resources (funding, apparatus, time).

In what follows, we offer a few intuitive examples of where we see untapped potential for Crowdsourcing in the Research Value Chain. We distinguish between "potential" and "feasibility".

Untapped potential for Crowdsourcing within the Scientific Method. Regarding untapped potential, we believe that the analysis of the collected data as well as the interpretation and drawing of conclusions have high potential for using the wisdom of the crowd or rather its intelligence. The crowd is particularly suited for recognizing patterns and important data points ("looking at the right spots"). In addition, the crowd might read data differently, draw additional conclusions and ideas, and thus complement the researcher or a small research team in its findings (evidence can be found in Kanefsky et al. 2001). Another good example for such a success is the "Goldcorp Challenge" (see e.g., Brabham 2008) The Canadian gold mining group Goldcorp made 400 megabytes of geological survey data on its Red Lake, Ontario, property available to the public over the Internet. They offered a $575,000 prize to anyone who could analyze the data and suggest places where gold could be found. The company claims that the contest produced 110 targets, over 80% of which proved productive; yielding 8 million ounces of gold, worth more than $3 billion. The prize was won by a small consultancy in Perth, Western Australia, called Fractal Graphics.
We see further potential in the formulation of hypotheses (similar to forecasting) from information collected. J. Scott Armstrong of Wharton School studied the prognoses of experts in several fields. In not a single instance could he find any clear advantage in having expertise in order to predict an outcome; "…expertise beyond a minimal level is of little value in forecasting change […].This is not to say that experts have no value, they can contribute in many ways. One particularly useful role of the expert seems to be in assessing a current situation." (Armstrong 1980). In the same paper he states several other studies that confirm this with respect to forecasting or hypothesizing. We also believe that the crowd can be especially useful in defining the (research) questions and in collecting relevant literature. As a positive side effect, consulting a crowd may also help overcome group biases like Groupthink (Janis 1972).

Feasibility of using Crowdsourcing within the Scientific Method. Regarding feasibility, the same steps are likely to be a target for Crowdsourcing: The questions can be discussed and exchanged through electronic channels (e.g., discussion boards, email) and literature collections can be remotely coordinated. Collected data can be posted on the Internet for analysis while interpretations can be discussed through application-sharing tools.

A pilot study was conducted during the "ShanghAI Lectures 2009", (see Hasler et al. 2009), a global lecture on Artificial Intelligence involving 48 universities from five continents – the 421 participating students could support one of four current scientific projects by contributing a paper stating their ideas on pre-defined open questions. The contest prize was a trip for the winning team to Zurich, Switzerland. Some of the solutions were rated "excellent", "well-elaborated" and "useful for the advancement of the project" by the scientists that headed the projects. We sent questionnaires to 372 participating students after the lectures and received 84 valid replies (23%). Although only 16% stated that they had prior theoretical or technical knowledge regarding the chosen subject, 32.1% of them indicated that they had much or very much fun participating in the contest and 15% agreed to participate in another contest while 29% answered "maybe" (although the workload was significant with several hours up to two weeks investment and the lecture was over). 22.6 % of all students (including those that did not participate in the contest) perceived a potential impact on current research if they participated in the contest.
However, the data collection was not thorough enough to analyze all the variables mentioned in our framework.

In addition, data gathered from the Crowdsourcing website "starmind.com" indicates that for 247 not-for-profit scientific questions posted between 1 January 2010 to 27 May 2010, 481 solutions have been submitted by question solvers, 368 of these have been viewed by the question posers with a resulting satisfaction of at least "good" for 267 (73%, on a scale "excellent", "good", "useful", "decline"). 66% of the problem solvers that contributed to a "good" rating are not "scientists" (self-assessed: PhD student, postdoctoral researcher, Professor). Starmind focuses on "small" questions. The rewards for answering a question start as low as EUR 3.-

Our research will analyze the tasks of the Research Value Chain according to the framework in much more depth, aiming to create a CI genome for each task of the Research Value Chain, where applicable. In addition, empirical data will be analyzed regarding the moderating variables to identify relevant sensitivities.

Challenges in Crowdsourcing and the Connection to AI Research

When dealing with any form of outsourcing of tasks (including Crowdsourcing), the risks are non-trivial. Especially for groups that are more distant, geographically and culturally, many situations arise that cannot be foreseen (see e.g., Nakatsu and Iacovou 2009). Crowdsourcing is an extreme case of dealing with the unknown, where emergence and the reactions to emerging behavior play an important role: The individuals of the "crowd" are a priori unknown and contingency plans for unexpected behavior of this interacting mass cannot be fully prepared beforehand. Moreover, in a Crowdsourcing scenario there are no pre-defined contracts between parties like in traditional outsourcing. Lane points out that risk is involved when using Crowdsourcing for decision making:

> "However, mechanisms also need to be in place to protect against competition sabotaging the crowd system. [...] Therefore, systems that leverage the crowd for creation decisions should ensure that the final decision passes through a governing body." (Lane 2010).

Roman (2009) states that there is an inherent weakness to Crowdsourcing that the difference between the "wisdom of crowds" and the "mob that rules" must be actively managed in order to manage correctness, accuracy and other elements that are relevant for valid fundamental research.
(For some further specific risks of Crowdsourcing, see e.g., Kazai and Milic-Frayling 2009).
There is, however, a fundamental consideration that justifies the trust in the wisdom of crowds: Assume that a decision problem has to be tackled. The members of the crowd have a certain intuition about the problem, which gives them a small bias towards the "correct" decision. It is easy to show that if a million individual agents decide independently and have a slight bias of 50.1 % towards taking the right decision (which is close to random guessing), a majority vote will lead to the correct decision with a probability of 97.7%. Even if there is a lack of expert knowledge, crowd decisions are rather robust. It is an open question to what extent the assumption about the independency of the decisions of individual agents is justified. Furthermore, independency also implies the absence of knowledge transfer between the agents, hardly a desired feature. Finding the optimal balance between communication and independency is therefore a relevant research topic.
Lakhani and Panetta (2007) state when comparing Open Source Software development (OSS) to traditional (business) management:

> "Brownian motion-based management" is not yet taught in any business schools. But the participation of commercial enterprises in OSS communities and other distributed innovation systems suggest that organizing principles for participation, collaboration, and self-organization can be distilled. Importantly, these systems are not "managed" in the traditional sense of the word, that is, "smart" managers are not recruiting staff, offering incentives for hard work, dividing tasks, integrating activities, and developing career paths. Rather, the locus of control and management lies with the individual participants who decide themselves the terms of interaction with each other.

Scholars in Artificial Intelligence (AI) research have developed (and are still developing) "design principles" that distill high-level principles for increasing the robustness of agents or groups of agents (see e.g., Pfeifer and Bongard 2007 or Pfeifer and Scheier 1999). These design principles specifically "prepare" the intelligent agents to deal with unexpected or unknown situations or to interact with unknown environments and large groups or known/unknown individuals.

Three examples of agent design principles

The following three example principles are stated here to make this idea more tangible. The first one deals with the importance of the way a problem is defined for Crowdsourcing, while the second example discusses the need for partial overlaps (redundancy). The third example puts the focus on local rules of interaction, thus shifting the focal point from a complex abstraction of "the crowd" to a better understandable, concrete set of small observations:

'Three Constituents' Principle. The ecological niche (environment), the tasks at hand and the agent must always be taken into account when investigating on or modeling intelligent behavior. This implies for Crowdsourcing, that not only processes or organizational structures (part of the environment) are relevant for success, but also the task (e.g., formulation of the problem at hand) and the socio-technical environment as well as the variables describing the agent (individual, group or other organization) in their interplay. AI research provides frameworks and tools in order to do this systematically. We have already incorporated this principle into our general analysis framework, above.

'Redundancy' Principle. Lean operations (Womack et al. 1991) and other optimizing paradigms are trying to eliminate redundancy in organizational processes. Current Artificial

Intelligence research shows that partial overlap of functionality is helpful and even necessary to build robust intelligent systems that are able to cope with the unexpected and new.

> In general, biological systems are extremely redundant because redundancy makes them more adaptive: if one part or process fails, another, similar part or process can take over. Brains also contain a lot of redundancy; they continue to function even if parts are destroyed. (Pfeifer and Bongard 2007)

Insights from AI research may help identify where redundancy is necessary to create robustness when crowdsourcing, and where it can be omitted for the sake of efficiency.

'Design For Emergence" Principle. This principle specifically aims at Collective Intelligence and states that when analyzing biological systems, the focus should be on the local rules of interaction that give rise to the global behavioral pattern that is studied:

> Because systems with emergent functionality rely on self-organizing processes that require less control, they tend to be not only more adaptive and robust but also cheaper. Emergent functionality requires us to think differently, for example, about social interaction, because much of what we may have thought would be under conscious control turns out to be the result of reflex-like local interactions. (Pfeifer and Bongard 2007)

The local rules of interaction for Crowdsourcing that produce desired input by the crowd are part of our ongoing research.
There are many more agent design principles dealing with different numbers of agents (e.g., single agents vs. groups of agents as in a Crowdsourcing situation) and different time scales (e.g. "here and now" vs. ontogenetic and phylogenetic time scales) that we will consider during the analysis that follows.

Making Crowdsourcing in Science more robust – towards a research agenda

In what follows we propose a research agenda that aims at three goals:
G1. Examine which forms (see e.g., Schenk and Guittard 2009) of Collective Intelligence in the large, or Crowdsourcing, and which incentives are suitable for use in fundamental research (based on the simplified "Research Value Chain" and our framework).
G2. Test the applicability of agent design principles in order to make collaboration based on Collective Intelligence more robust, with a special focus on Crowdsourcing in fundamental research.
G3. Identify local rules of interaction between agents in Collective Intelligence interactions (incl. Crowdsourcing) that lead to productive emerging phenomena. The definition of "productive" depends on the domain: In fundamental science it is measured by maximizing the contribution to the body of reliable knowledge.

Research Questions

The following questions will guide our research in the two branches:
G1-Q1. Which forms of Crowdsourcing (e.g., routine task vs. complex task vs. creative) are best suited to fundamental research?
G1-Q2. Are there best practices for Crowdsourcing in fundamental research that can be generalized for several disciplines?
G1-Q3. Which are the best incentive schemes for Crowdsourcing in fundamental research?
G1-Q4. How does the aim of protecting IP with a patent (or other instrument) change the above answers?
G2-Q5. Can the application of agent design principles (e.g., "frame of reference principle", "motivated complexity principle", "cumulative selection principle") to platforms and processes make Crowdsourcing interactions more successful in terms of useful input by "the crowd"?
G2-Q6. If the answer to Q5 is "yes", which design principles are best suited to which situation?
G2-Q7. Are there differences regarding Q6 in different disciplines?
G2-Q8. Decisions made by independent agents are highly robust, but communication offers other benefits. Is there a way to determine an optimal balance between robustness and interdependency/communication?
G3-Q9. Which local rules of interaction can be inferred in different tasks of the Research Value Chain?

Hypotheses

Given the limited data set so far, we state the following hypotheses in order to guide our empirical evidence finding.
These hypotheses form a basic collection of ideas that will be subsequently tested, expanded and detailed in a structured and systematic manner.

H1. The prerequisites for Crowdsourcing (see, e.g., Benkler 2006, Howe 2008, Kazman and Chen 2009) are present in academic settings.
H2. Scientists from different disciplines perceive Crowdsourcing as a useful tool for supporting fundamental research.
H3. By systematically applying agent design principles (Three Constituents, Complete Agent, Parallel, Loosely Coupled Processes, Sensory-Motor Coordination, Cheap Design, Redundancy, Ecological Balance, Value) to Crowdsourcing settings, the output of the community (in terms of "value" as judged by seeking scientists) can be significantly increased (compared to not applying principles).
H4. By systematically applying design principles for development (Integration of Time Scales, Development as an Incremental Process, Discovery, Social Interaction, Motivated Complexity) and insights from AI fields (e.g., Swarm Behavior, Complex Networks), the quality of a community can be improved over time in terms of efficiency and effectiveness in solving a crowdsourced task (compared to groups not applying principles).
H5. By systematically applying design principles for evolution (Population, Cumulative Selection and Self-Organization,

Brain-Body Coevolution, Scalable Complexity, Evolution as a Fluid Process, Minimal Designer Bias), a research group can increase its value creation (see above) from Crowdsourcing processes over time quicker than when not applying the principles.

H6. Crowdsourcing techniques allow Academic research groups to more successfully advance outputs from fundamental research to market maturity (technology transfer) than without Crowdsourcing.

H7. "Crowds" (in the sense of an active community in Crowdsourcing) involved in fundamental research are subject to guided self-organization (i.e., autonomous global self-organization with a few adjustable parameters, e.g., given by the environment or the platform).

Methods and Approaches

We will apply our framework to identify the sensitivities regarding moderating variables (environment and agent) when in a fundamental research setting. In addition, we will generate "CI genomes" for each task in the Research Value Chain, in order to better understand the applicability for Crowdsourcing. In parallel, we will collect more data regarding Crowdsourcing contributions to different steps in the "Research Value Chain":

The data gathering will consist of several Crowdsourcing contests treating current projects in fundamental research (at universities). Both the participants in the contests ("crowd") as well as the participating researchers will complete a set of questionnaires which include both closed- and open-ended questions on individual and team functioning (in case a contribution is made by a team) during these contests as well as self-assessed vs. outside-assessed ratings of the inputs they give. The questionnaires will be based on (Bartl 2006) and (Lakhani et al. 2006), but slightly adapted to better suit the non-profit context of universities.

One (or more) iteration(s) of the data gathering process will be used to (in)validate the insights gained from the data and test the application of agent design principles as stated above.

As a final measure, a Multiagent System (Weiss 2000, Wooldridge 2008) will be implemented in order to simulate stochastic behavior given the sensitivities and settings found in the data.

The inquiry will limit its focus to fields where the "Research Value Chain" is applicable and generally accepted as a guiding process for conducting fundamental research.

Conclusion

Based on the current success in several industries, we see indications that fundamental research potentially benefits from leveraging Collective Intelligence techniques (including Crowdsourcing). We hypothesize that there are "tasks" in the Scientific Method that can potentially benefit from Crowdsourcing and will test our hypotheses according to the stated research agenda.

In addition, we will test the applicability of agent design principles from Artificial Intelligence research to Crowdsourcing. In this paper, we have shown only a few examples of these principles, there are more stated in the current AI literature (The hypotheses H3 to H5 state some more principles that might be suitable for this context.) Although focusing on fundamental science, this research will potentially yield insights for making processes involving Collective Intelligence in the private sector more robust, too.

If you would like to be part of this research, please contact the corresponding author.

References

Amigoni, F., M. Reggiani, V. Schlaffonati. 2009. An insightful comparison between experiments in mobile robotics and in science. *Autonomos Robot*(27) 313–325.

Armstrong, J. S. 1980. The seer-sucker theory: The value of experts in forecasting. Technology Review 83: 18-24. *Technology Review* **83** 18–24.

Bartl, M. 2006. *Virtuelle Kundenintegration in die Neuproduktentwicklung.* Deutscher UniversitSts-Verlag | GWV Fachverlage GmbH Wiesbaden, Wiesbaden.

Benkler, Y. 2006. *The wealth of networks*. *How social production transforms markets and freedom.* Yale Univ. Press, New Haven.

Bok, D. C. 2003. *Universities in the marketplace*. *The commercialization of higher education.* Princeton Univ. Press, Princeton.

Brabham, D. C. 2008. Crowdsourcing as a Model for Problem Solving. *Convergence: The International Journal of Research Into New Media Technologies* **14**(1) 75–90.

Castelluccio, M. 2006. Collective Intelligence. Editor's comment. *Strategic Finance*(Tech Forum) 51–52.

Chesbrough, H. W. 2003. *Open innovation*. *The new imperative for creating and profiting from technology.* Harvard Business School Press, Boston.

Crawford, S., L. Stucki. 1999. Peer review and the changing research record. *Journal of the American Society for Information Science* **41**(3) 223–228.

David, P. A. 2004. Can "Open Science" be Protected from the Evolving Regime of IPR Protections? *Journal of Institutional and Theoretical Economics JITE* **160**(1) 9-34(26).

Etzkowitz, H. 1983. Entrepreneurial scientists and entrepreneurial universities in American academic science. *Minerva* **21**(2-3) 198–233.

Etzkowitz, H., A. Webster, C. Gebhardt, B. R. Cantisano Terra. 2000. The future of the university and the university of the future: evolution of ivory tower to entrepreneurial paradigm. *Research Policy* **29**(2) 313–330.

Feyerabend, P. K. 1993. *Against method,* 3rd ed. Verso, London.

Hasler, B. S., T. Buecheler, R. Pfeifer. 2009. Collaborative Work in 3D Virtual Environments: A Research Agenda and Operational Framework. A. A. Ozok, P. Zaphiris, eds. *Online communities and social computing*. *Third international conference, OCSC 2009, held as part of HCI International 2009, San Diego, CA, USA, July 19-24, 2009 ; proceedings*. Springer, Berlin, 23–32.

Hofstadter, D. R. 1979. *Gödel, Escher, Bach: An Eternal Golden Braid.* Basic Books.

Howe, J. 2006. The Rise of Crowdsourcing. *WIRED Magazine*(14.06), http://www.wired.com/wired/archive/14.06/crowds.html

Howe, J. 2008. *Crowdsourcing: Why the Power of the Crowd is Driving the Future of Business.* Crown Publishing, New York.

Howe, J. 2010. Crowdsourcing. Why the Power of the Crowd is Driving the Future of Business, http://www.crowdsourcing.com/ (April 3, 2010).

Irwin, A. 1995. *Citizen science. A study of people, expertise and sustainable development.* Routledge, London.

Janis, I. L. 1972. *Victims of groupthink. A psychological study of foreign-policy decisions and fiascoes*. Houghton Mifflin, Boston.

Kanefsky, B., N. G. Barlow, V. C. Gulick. 2001. Can Distributed Volunteers Accomplish Massive Data Analysis Tasks? *32nd Annual Lunar and Planetary Science Conference, Houston, Texas.*(1272).

Kazai, G., N. Milic-Frayling. 2009. On the Evaluation of the Quality of Relevance Assessments Collected through Crowdsourcing. *Proceedings of the ACM SIGIR 2009 workshop on the future of IR evaluation*.

Kazman, R., H.-M. Chen. 2009. The Metropolis Model: A New Logic for Development of Crowdsourced Systems, http://cacm.acm.org/magazines/2009/7/32094-the-metropolis-model-a-new-logic-for-development-of-crowdsourced-systems/fulltext (April 5, 2010).

Landry, R., N. Amara. 1998. The impact of transaction costs on the institutional structuration of collaborative academic research. Research Policy 27(9) 901–913.

Lakhani, K. R., L. B. Jeppesen, P. A. Lohse, J. A. Panetta. 2006. The Value of Openness in Scientific Problem Solving. *Harvard Business School*(HBS Working Paper Number: 07-050).

Lakhani, K. R., J. A. Panetta. 2007. The Principles of Distributed Innovation. *MIT innovations* **2**(3) 97–112.

Lane, S. 2010. Collective Intelligence for Competitive Advantage: Crowdsourcing and Open Innovation. *Applied Information Management Master's Capstone Projects and Papers*, https://scholarsbank.uoregon.edu/xmlui/handle/1794/10210.

Leadbeater, C., D. Powell. 2009. *We-think. mass innovation, not mass production.* Profile, London.

Lee, S., B. Bozeman. 2005. The Impact of Research Collaboration on Scientific Productivity. *Social Studies of Science* **35**(5) 673–702.

Malone, T. W. 2004. *The future of work. How the new order of business will shape your organization, your management style, and your life.* Harvard Business School Press, Boston.

Malone, T. W., R. Laubacher, C. Dellarocas. 2009. Harnessing Crowds: Mapping the Genome of Collective Intelligence. *MIT Sloan Research Paper*(4732-09).

Melin, G. 2000. Pragmatism and self-organization Research collaboration on the individual level. *Research Policy* **29**(1) 31–40.

Nakatsu, R. T., C. L. Iacovou. 2009. A comparative study of important risk factors involved in offshore and domestic outsourcing of software development projects: A two-panel Delphi study. *Information & Management* **46**(1) 57–68.

Nelson, R. R., S. G. Winter. 1982. *An evolutionary theory of economic change.* Belknap Press of Harvard Univ. Press, Cambridge.

Page, S. E. 2008. *The difference. How the power of diversity creates better groups, firms, schools, and societies,* 3rd ed. Princeton Univ. Press, Princeton, NJ.

Pfeifer, R., J. Bongard. 2007. *How the body shapes the way we think. A new view of intelligence.* MIT Press, Cambridge.

Pfeifer, R., C. Scheier. 1999. *Understanding intelligence,* 1st ed. MIT Press, Cambridge.

Roman, D. 2009. Crowdsourcing and the question of expertise. *Communications of the ACM* **52**(12) 12.

Schenk, E., C. Guittard. 2009. Crowdsourcing: What can be Outsourced to the Crowd, and Why? HAL: Sciences de l'Homme et de la Société. *halshs.archives-ouvertes.fr*.

Schrage, M. 2000. *Serious play. How the world's best companies simulate to innovate.* Harvard Business School Press, Boston.

Schumpeter, J. A. 1934. *The theory of economic development. An inquiry into profits, capital, credit, interest, and the business cycle.* Transaction Books, New Brunswick.

Shneiderman, B. 2008. Science 2.0. Copernican challenges face those who suggest that collaboration, not computation are the driving energy for socio-technical systems that characterize Web 2.0. *Science* **319** 1349–1350.

Stoehr, K., WHO. 2003. A multicentre collaboration to investigate the cause of severe acute respiratory syndrome. *The Lancet* **361**(9370) 1730–1733.

Surowiecki, J. 2004. *The wisdom of crowds. Why the many are smarter than the few and how collective wisdom shapes business, economies, societies, and nations,* 1st ed. Doubleday, New York.

Tapscott, D., A. D. Williams. 2008. *Wikinomics. How mass collaboration changes everything.* Portfolio, New York.

von Hayek, F. A. 1945. The Use of Knowledge in Society. *American Economic Review* 519–530.

von Hippel, E. 1986. Lead Users: A Source of Novel Product Concepts. *Management Science* **32**(7) 791–805.

von Hippel, E. 2005. *Democratizing innovation.* MIT Press, Cambridge.

Weiss, G., ed. 2000. *Multiagent systems. A modern approach to distributed artificial intelligence,* 2nd ed. MIT Press, Cambridge.

Womack, J. P., D. T. Jones, D. Roos. 1991. *The machine that changed the world. [the story of lean production],* 1st ed. Harper Perennial, New York.

Wooldridge, M. J. 2008. *An introduction to multiagent systems.* Wiley, Chichester.

SO-LOST: An Ant-Trail Algorithm for Multi-Robot Navigation with Active Interference Reduction

Seyed Abbas Sadat, Richard T. Vaughan

Autonomy Lab, School of Computing Science
Simon Fraser University, BC, Canada
{*sas21, vaughan*}*@sfu.ca*

Abstract

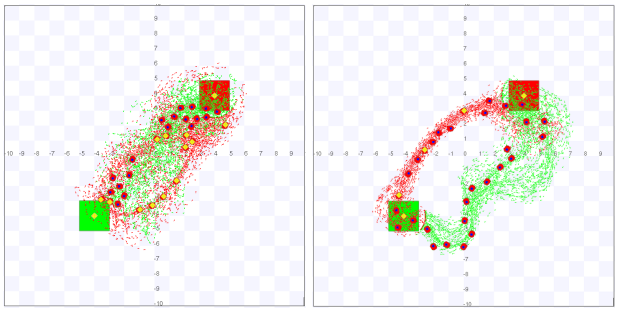
We consider a group of autonomous robots which perform the classical task of transporting resources from a source to home. The robots use ant-like emergent trail following to navigate between home and source. When trails lie close together, spatial interference between robots navigating in opposite directions reduces overall system performance. This paper proposes a navigation strategy which is effective in separating trails with different goals. The results of simulation experiments indicate that the performance of robots is usefully increased compared to original algorithm in constrained environments.

Introduction

This paper presents a navigation strategy to reduce interference in ant-inspired foraging-and-trail-following robot systems. We consider the classical *resource transportation* task, in which a team of robots works to transport resources in an initially unmapped environment. Robots start from a home position and search for a supply of resources. On reaching the source, they receive a unit of resource and must return home with it, then return to fetch more resource repeatedly for the length of a trial. Achieving this task reliably with robots will meet a real-world need. It is a canonical multi-robot task since the work is inherently parallelizable. The critical factor limiting scalability is mutual spatial interference between robots.

Our earlier work Vaughan et al. (2000, 2002) examined an implementation of ant-inspired trail following that is suitable for imperfectly-localized mobile robots. In our "localization-space trails" (LOST) algorithm, robots generate and share trail data structures composed of waypoints specified by reference to task-level features that are shared by all robots. The trails are continuously refined online, and maintain the ant-algorithm property Dorigo (1992) of converging to near-optimal paths from source to home.

Trails are labelled with their destination, and the trail to the current goal destination is followed. In previous work, the other trails were ignored during navigation. However, trails may overlap in space and robots navigating to different goals may interfere with each other's progress. We argued previously that an emergent property of LOST is that it can produce trails that are separated in space Vaughan et al. (2000) thus reducing interference. In this paper we describe a modification to LOST called Spread-Out LOST (SO-LOST) that greatly improves this effect, creating trails that share parts of the environment while being far enough apart to reduce interference. The result is superior performance in most of the cases we examine. The innovation is that the robots' trail-following behaviour is subtly modified to avoid competing trails, with the emergent effect that trails are iteratively spread out until intereference is largely avoided.



(a) LOST (b) SO-LOST

Figure 1: Trails formed in an obstacle-free "empty" environment using LOST and SO-LOST. SO-LOST has separated the trails, achieving better throughput due to reduced interference.

It is reported that some type of ant use repellent pheromone to mark unrewarding areas so that other ants avoid foraging that part of the environment (Robinson et al. (2008)). However, we do not know of any biological system that uses similar behaviour to tackle the spatial interference problem. The new navigation algorithm in this paper is a synthetic technique that improves the efficiency of a biologically-inspired path finding and sharing algorithm used in multi-robot systems. The advantage of these type of synthetic behaviors has been studied before (e.g. Heck and Ghosh (2002)).

Related Work

Various different robot implementations of ant-like trail following have been presented. Real chemical marks were first used to produce true stigmergic trail-following in Russell et al. (1994). Also recently, Fujisawa et al. Fujisawa et al. (2008) carried out a study out of communication in a swarm of robots using pheromone and proposed a behavior algorithm for robots to search for prey and attract other robots. They used ethanol as pheromone in their real robot experiments. The challenge of chemical and sensor engineering makes these methods often impractical. A more parsimonious method was invented by Payton et al. (2001) where virtual pheromone trails are implemented by directional infra-red messages transmitted from robot to robot. Robots echo received messages, incrementing a contained hop-count which is used to estimate the distance to the message source. In both chemical and IR-mediated methods, the local "gradient" is sensed directly from the environment. If robots are mutually localized, virtual trails can be created from global waypoints, which are communicated by wireless network. We showed that this scheme can be robust to large zero-mean localization error (Vaughan et al. (2000)), and admits a relaxed and practical definition of mutual localization (Vaughan et al. (2002)).

The diminishing-to-negative-returns effect of increasing the number of robots on performance has been studied in related contexts. In a mathematical model of robot foraging Lerman and Galstyan (2002), it was shown that adding more robots to the system improved the group performance while decreasing individual robot's performance. Based on that model, an optimal group size was found that maximizes the group performance. Explicit anti-interference strategies are studied in real robots in Zuluaga and Vaughan (2005), to increase performance in the transportation task. Congestion control in a dense multi-robot system is studied in Scheidler et al. (2008), where asymmetries that resolve conflicts are introduced by modifying either the environment or the robot controllers.

A related idea using occupancy grids to model multi-robot interaction is described in Zuluaga and Vaughan (2008). There, a global histogram of occupancy is constructed, and areas with high probability of co-location are identified and fed into an (unrelated) interference reduction method.

Localization-Space Trails (LOST) review

This section briefly reviews the generalized trail-following method formulated in Vaughan et al. (2002).

LOST generates trails between the locations of *Events*. An Event is defined as a task-relevant occurrence that may happen to any member of the team, and is locally but reliably perceived. For example, in our transportation task the relevant Events would be 'pick-up-resource' and 'drop-resource'. A robot must be able to recognize these events in order to switch between resource-seeking behavior and home-seeking behavior. When an Event occurs to a robot, its current pose in localization space is recorded to create an [Event,Pose] tuple called a *Place*. A robot can then express information about the world relative to the Places it has seen. Other robots that have position estimates for the same Events can interpret the coordinates in their own local frame of reference. Thus robots are mutually localized by the shared experience of the common task, rather than conventional global localization in some arbitrary coordinate system.

The purpose of LOST is to guide the robot to a Place currently of interest: the goal. The algorithm provides the robot controller with two pieces of information; (i) the *heading-hint* that is the local direction in which to travel to reach the goal; (ii) the *distance-hint* that is the estimated cost (usually in time) to reach the goal. These hints are extracted by examining a set of waypoints called *Crumbs* which are poses specified relative to a Place. The current set of Crumbs specified relative to a particular Place is a *Trail* to that place. A Crumb is a tuple $C = [P_c, L_c, d_c, t_c]$ containing the name of the Place P_c to which it refers, a localization space pose L_c, an estimate d_c of the distance (in some distance function) from L_c to P_c, and the time t_c when the Crumb was created.

Each robot maintains an initially empty temporary trail. Every S seconds, a robot inserts a new crumb to the temporary trail. The crumb contains the current location of the robot, the name of the most recent Event experienced by that robot, the distance from the last event, and the current time. When another event occurs to the robot (e.g., when a robot drops off its cargo), the temporary trail is broadcast to all robots, including itself, then deleted. A new temporary trail is then created for the recent Event.

Besides the temporary trail, each robot maintains a trail for each different Event it has learned about from the network. When a broadcast trail is received, the crumb poses are transformed into the local frame of reference by the rigid body transform defined by comparing the local and received poses of the trail's Place. The transformed crumbs are added to the local trail for this Place. All trails are periodically scanned and any Crumb with time stamp older than age threshold a seconds is discarded. Thus the trail is updated dynamically, and out-of-date information is expired. The dynamic response of the trail to changing environments is a function of a.

Suppose a robot at pose L_r has Place P_g as its goal, such as $Event(P_g)$ = 'drop-resource-at-home'. The robot searches the set of Crumbs with Place = P_g to find the set of crumbs that lie within its *field of view* (FOV) i.e., within radius d_f of L_r. From this set it finds the crumb C_L with the smallest distance-to-goal d_c. This distance is returned as the distance-hint. The heading-hint is the angle from the robot's pose L_r to $L_c = Pose(C_L)$. If the robot moves in the direction of the heading hint and repeats this process, it will encounter crumbs with decreasing distance to goal values,

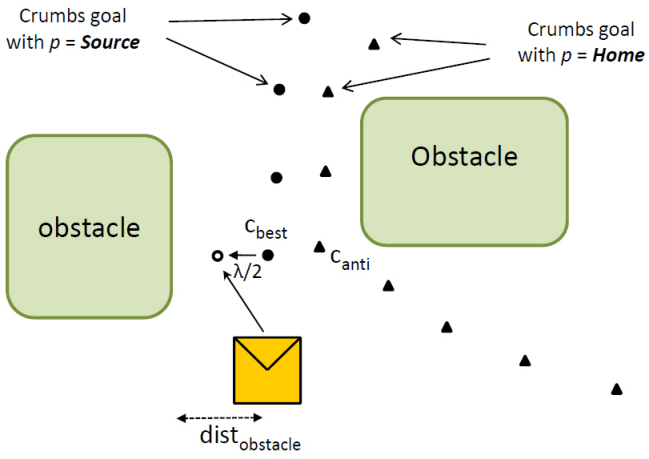


Figure 2: Sketch of the new LOST algorithm. While the robot was following the trail (filled circles), it sees a crumb with different goal (triangle) and thus changes its direction to a new point (the empty circle).

and eventually arrive at P_g.

The robot will take the shortest route so far discovered from that location. By following the Crumbs dropped by the whole population, each robot benefits from the others' exploration; robots will find a reasonable route much more quickly than they would alone. The larger the population size, the greater the probability of finding a good route and the more quickly a good route is found.

Spread-Out LOST

In the LOST algorithm, as the robots move they "lay" crumbs. The goal Place of these crumbs is the place that the robot has most recently visited. This means that in order to reinforce a trail, the robots should travel in the opposite direction that the crumbs are showing and consequently robots following a trail are very likely to interfere with robots laying (reinforcing) it. With few robots, this does not have much effect on performance and the "pick-up-resource" and "drop-resource" trails converge to one shortest discovered path. However as the robots' team size increases, these interferences damage the performance of the system.

To address this, we modify the LOST algorithm so that when a crumb is created, the P_c data field will be the goal of the robot rather than the recently visited place. With this modification, the robots have to perform two searches at the beginning; one for finding a path from home to source and another one for a path from source to home. We can avoid the need for the second search by copying the first discovered trail and changing the goal and reversing the distance hint along the trail.

When the environment in which the robots are working

Algorithm 1 The New Trail-Using Algorithm

Require: The distance $dist_{obstacle}$ from the robot to the nearest non-robot obstacle on the left side of the robot.
return the direction Dir_{robot} to which the robot should move

Θ = all the crumbs in the robot's FOV with positions relative to the robot;
$\Sigma = \{c | c \in \Theta \wedge (c.p_c = robot.goal)\}$;
$\Pi = \{c | c \in \Theta \wedge (c.p_c \neq robot.goal)\}$;

$\lambda = Min(crumb\_avoid, dist_{obstacle})$;

$c_{best} = c$ s.t. $(c \in \Sigma) \wedge (\nexists c' \in \Sigma$ s.t. $c.d_c > c'.d_c)$;

if $(\exists c_{anti} \in \Pi$ s.t. $dist(c_{anti}, robot) < crumb\_avoid) \wedge (c_{best}.d_c \leq 2s)$ **then**

$Dir_{robot} = \overrightarrow{(robot, c)} + \frac{\lambda}{2} \times \overrightarrow{(-1, 0)}$;

else

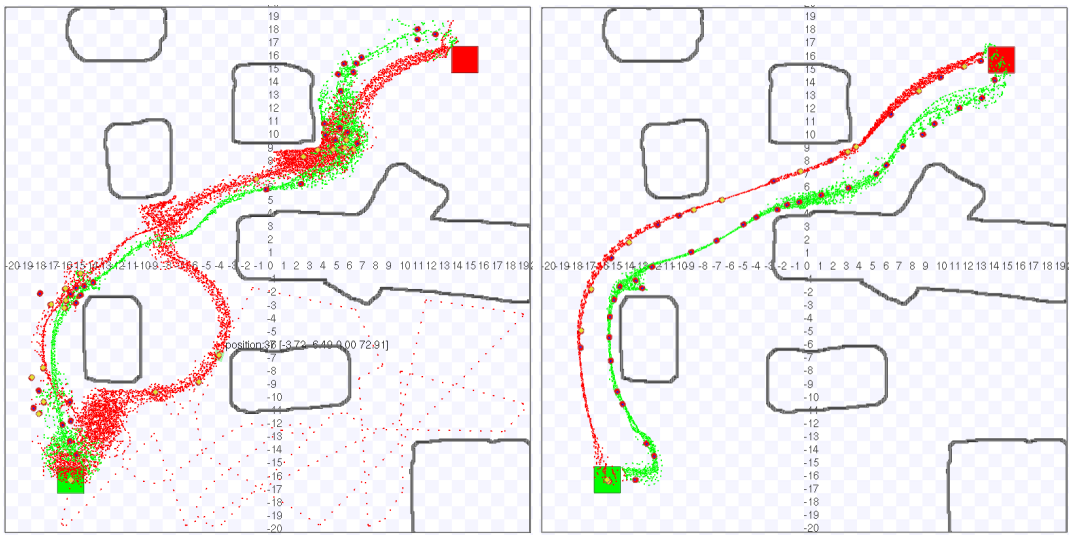
$Dir_{robot} = \overrightarrow{(robot, c)}$;

end if

is complicated and contains narrow corridors and doorways, or is very crowded, LOST may produce trails with different goals that are either very similar or have many parts in common. Figures 1(a),3(a),4(a) show this phenomenon in our trail-following robot system implemented in the well known simulator Stage (Vaughan (2008)). The trails formed between source and home are often very close to each other, leading to problematic interference between robots travelling in opposite directions. Since the crumb trail data structure does not contain any explicit information about the fixed obstacles in the environment, there is no way to directly process the trail data to avoid robot-robot interference without risking directing robots into fixed obstacles. Instead, we use a small modification to the robots' trail following control strategy that results in emergent trail separation.

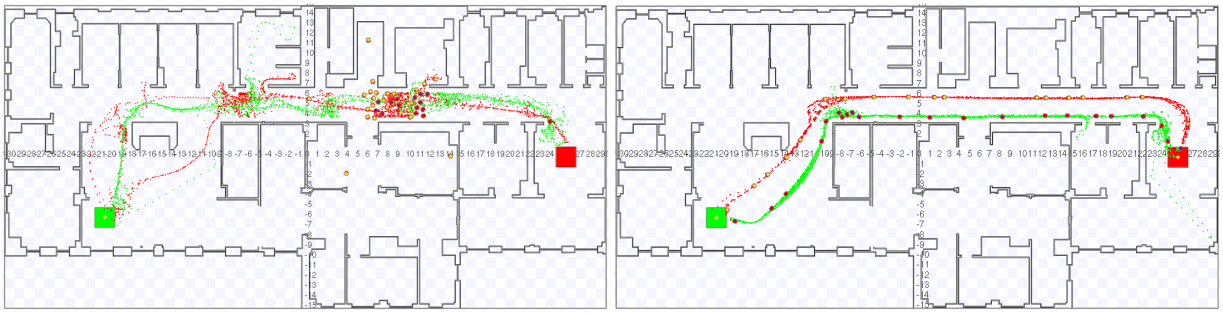
A robot following a trail to get to P_c, can interpret crumbs with goals other than P_c, as proxies for potentially interfering robots. If the robot follows the trail to P_c while slightly avoiding all other nearby crumbs, the new P_c crumbs it lays will tend to be slightly more distant from other crumbs than those just followed. This mechanism is essentially similar to the iterated corner-cutting that drives the ant-algorithm's ability to locally improve trail length. The resulting trails may be slightly longer but may reduce interference significantly, as suggested by the results below.

The new trail-using algorithm is presented in Algorithm 1. It first searches for the crumb c_{best} with minimum distance to goal that is located in the robot's FOV. Then if



(a) LOST (b) SO-LOST

Figure 3: Trails formed in the cave environment using the LOST and the new algorithm after 30 mins of simulation.



(a) LOST (b) SO-LOST

Figure 4: Trails formed in the hospital environment. using the LOST and the new algorithm.

there exists a crumb c_{anti} with different goal than the robot's goal and it was closer to the robot than a distance threshold ($crumb\_avoid$), the direction to which the robot moves will turn to the robot's left. This will change the angular velocity of the robot so that it keeps away from c_{anti}. The shift vector is orthogonal to the $\overrightarrow{(robot, c_{best})}$ vector. Also, the magnitude λ is calculated based on the obstacles near the robot such that the robot's target point does not lie inside an obstacle. Trails with different goals are necessarily very close to each other around source and home. Thus the shift vector is not applied when the robot is near the goals to prevent robot's circular trajectory in these areas.

Figure 2 illustrates how the behavior of the robot changes in presence of c_{anti}. The robot is following the small circles. On seeing the triangle crumbs, the robot's target point is changed from c_{best} to another point (the empty circle). This simple mechanism alters the robots movement so that different trails are gradually separated from each other. The divergent movement of trails continues until they are away enough from each other, if possible.

Experiments

Simulation Setup

We ran Stage simulations to evaluate the new algorithm in three different environment settings: $empty$ (Figure 1), $cave$ (Figure 3) and $hospital$ (Figure 4). The size of the $empty$, $cave$ and $hospital$ environments are 20x20m, 40x40m and 60x30m respectively, with robot length 0.45m. Robots are Stage's Pioneer 3DX and SICK LMS200 laser rangefinder models. The bottom left (green) square is the source; top right (red) square the sink of resources. In the screenshots, robots (red polygons) are shown with yellow diamonds to indicate they are carrying a unit of resource. Robots start every trial at the same randomly-chosen uniformly distributed positions, do not know the initial location of source and sink locations, and must find them by exploration at the start of

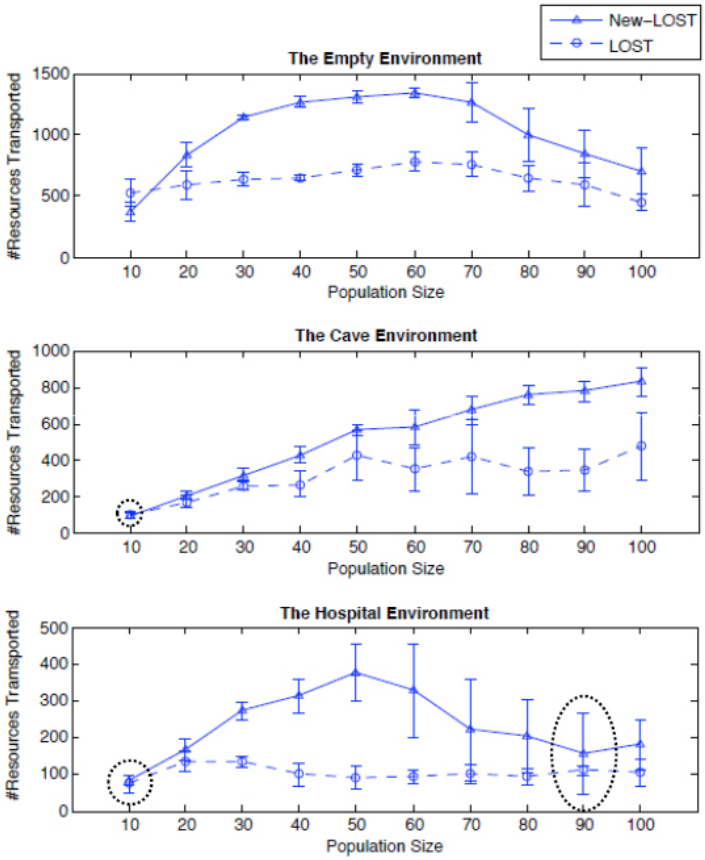


Figure 5: The result of the experiments in the 3 environments. The mean performance over 10 trials are shown with errorbars showing the standard deviation for both the original LOST and the new algorithm. The dotted line shows the data point for which the two algorithm do not show significant difference in distribution.

the trial. Each trial runs for 60 minutes, and the total number of resources delivered at the end of the trial is our performance metric. 10 trials are performed for each population size. LOST is deterministic but the local obstacle avoidance and searching is stochastic (for robustness), hence the need for repeated trials. For all experiments the $crumb\_avoid$ parameter is set to 2m.

Results

The results of the experiments are summarized in Figure 5, showing the mean and standard deviation of performance over 10 repeated trials plotted for each population size. The plot shows a marked improvement in many cases (in some cases 3 times better) in performance with the new algorithm.

As expected, with few robots (20), there is not much difference in performance since the interference among robots is small. In the empty environment with population size of 10, the LOST outperforms the new algorithm. This is because the benefit of interference reduction can not outweigh the penalty of increase in the length of the trails. As the population size increases and the environment becomes more constrained, improvement in performance gets bigger. This can be seen in the plot showing the results of the experiments in the hospital environment; For the smallest populations, the two methods perform about the same; however, since the hospital environment contains corridors and doorways (Figure 4(b)), there is a degradation in the LOST performance with more robots whereas the new algorithm improves the performance in some populations up to 3 times.

To verify that the performance results are significantly different for different algorithms, we performed hypothesis testing using a T-test. The P values for the hypothesis that the performance values for LOST and the new algorithm are from the same distribution are calcuated. For all population sizes, the test suggests that the distributions are significantly different ($P << 0.02$), except for the pairs identified in Figure 5 with dotted line.

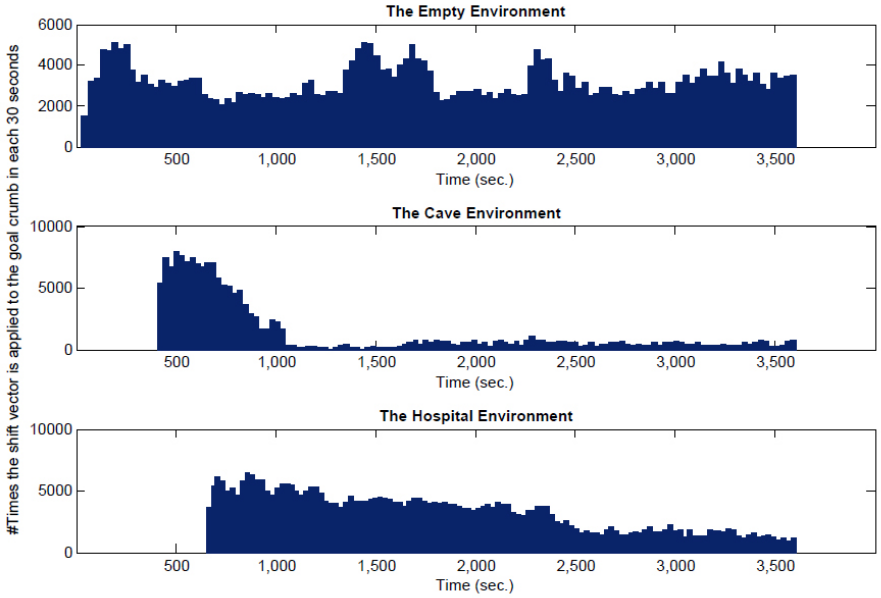


Figure 6: The histograms showing the number of times the goal crumb was shifted. The time bin is 30 seconds of simulation time. 30 robots are used in the empty environment and 50 robots are used in the other environments.

Discussion

The new algorithm is based on the idea that laying crumbs near other crumbs with different goals increases the probability of co-location among the robots performing different tasks. This is more clear in transportation task in which the trails for 'pick-up-resource' and 'drop-resource' tasks can be formed very close to each other. In the new algorithm robots follow the trails and also try to keep a distance from other crumbs and therefore new trails are laid at a safe distance from each other. Figures 1(b), 3(b), 4(b) show the trails formed with the new algorithm. It is visible that different trails are separated from each other and consequently robots do not approach the unattractive trails. The magnitude of the shift vector ($crumb\_avoid$) determines the distance of the trails from each other and should be large enough to keep robots away from each other.

In order to see if the trails converge to a stable state we plotted the number of simulation cycles in which the shift vector was applied in each 30 sec of simulation time (Figure 6). In the $cave$ and $hospital$ environments, after the trails are formed they are gradually separated from each other due to the high use of shift vector. After some time, the trails come into a relatively stable state. The shift vector is still applied occasionally since the trails in some narrow parts of the environment (like doorways) are at their maximum distance from each other and can not go farther away. For the $empty$ environment since the area is small and there is a short distance between source and sink, the robots tend to be pushed towards other trails which results in the high use of shift vector throughout the experiment.

We do not know of any biological system that uses a similar approach to reduce destructive effects of interference among individuals, but still we believe that these techniques can be used in systems inspired from animals and social insects to improve the efficiency of robots in performing a task.

Conclusion and Future Works

In this paper we presented SO-LOST, a new navigation strategy to reduce interference in ant-inspired foraging-and-trail-following robot systems. The method makes use of the different trails formed in the environment to prevent robots with different goals from getting in each other's way. It is quantitatively evaluated through simulation experiments and shown to be effective in relatively constrained environments. Qualitatively, the screenshots of simulation experiments show that distinct separate trails with different goals were formed while keeping a distance from each other hence reducing the interference.

In future work we will implement the new algorithm on real robots and run experiments to verify our findings in simulation. Also, we will investigate methods of congestion resolution in trail-following robot systems. The algorithm presented in this paper is used to avoid congestion and conflicts between robots. However, there is plenty of room for improvement in mutual robot-robot avoidance methods, and development here would have an impact in many multi-robot systems.

The LOST and SO-LOST framework allows us to add various kinds of meta-data to the crumb and trail data structures. Here we have allowed all nearby trails to influence the behaviour of a trail-follower. We expect that performance could be further improved by clever use of other meta-data embedded into crumbs, perhaps by gathering some global statistics. This would be unusual in ant-inspired systems, and perhaps powerful.

For now, we believe SO-LOST may be the most real-world practical trail-following algorithm yet described, since it explicitly manages the spatial interference that plagues real-world robots in any number.

Acknowledgements

This work was supported by NSERC and DRDC in Canada. Thanks to Greg Broten at DRDC, and to members of the Autonomy Lab for their constructive input.

Code publication

All source code, scripts, etc. used to produce the results reported in this paper are available online:
URI: http://www.cs.sfu.ca/ sas21/personal/abbas_alifeXII.tar.gz
SHA1: 9ea88a34d9971dffce26bc511f6ad2f875b8e44d

References

Dorigo, M. (1992). *Optimization, Learning and Natural Algorithms*. PhD thesis, Politecnico di Milano, Italy.

Fujisawa, R., Imamura, H., Hashimoto, T., and Matsuno, F. (2008). Communication using pheromone field for multiple robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1391–1396.

Heck, P. S. and Ghosh, S. (2002). The design and role of synthetic creative traits in artificial ant colonies. *J. Intell. Robotics Syst.*, 33(4):343–370.

Lerman, K. and Galstyan, A. (2002). Mathematical model of foraging in a group of robots: Effect of interference. *Auton. Robots*, 13(2):127–141.

Payton, D., Daily, M., Estowski, R., Howard, M., and Lee, C. (2001). Pheromone robotics. *Auton. Robots*, 11(3):319–324.

Robinson, E. J., Ratnieks, F. L., and Holcombe, M. (2008). An agent-based model to investigate the roles of attractive and repellent pheromones in ant decision making during foraging. *Journal of Theoretical Biology*, 255(2):250 – 258.

Russell, A., Thiel, D., Deveza, R., and Mackay-Sim, A. (1994). Sensing odour trails for mobile robot navigation. In *Proc. Int. Conf Robotics and Automation*, pages 2672–2677. IEEE.

Scheidler, A., Merkle, D., and Middendorf, M. (2008). Congestion control in ant like moving agent systems. In *Biologically-Inspired Collaborative Computing*, volume 268, pages 33–43.

Vaughan, R. T. (2008). Massively multi-robot simulations in stage. *Swarm Intelligence*, 2(2-4):189–208.

Vaughan, R. T., Støy, K., Howard, A., Sukhatme, G., and Matarić, M. J. (2002). Lost: Localization-space trails for robot teams. *IEEE Transactions on Robotics and Autonomous Systems*, 18(5):796–812.

Vaughan, R. T., Støy, K., Sukhatme, G. S., and Matarić, M. J. (2000). Whistling in the dark: cooperative trail following in uncertain localization space. In *Proceedings of the Fourth International Conference on Autonomous Agents*, pages 187–194, Barcelona, Spain.

Zuluaga, M. and Vaughan, R. (2005). Reducing spatial interference in robot teams by local-investment aggression. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Edmonton, Alberta.

Zuluaga, M. and Vaughan, R. T. (2008). Modeling multi-robot interaction using generalized occupancy grids, with application to reducing spatial interference. In *Proceedings of the IEEE International Conference on Robotics and Automation*.

Multi-modal Swarm Construction

Seth Bullock and Michael Kerby

University of Southampton, Southampton, SO17 1BJ, UK

sgb@ecs.soton.ac.uk

Abstract

Swarm construction involves a population of autonomous agents collaboratively organising material into useful persistent structures without recourse to central co-ordination or control. This approach to fabrication has significant potential within nanoscale domains, where explicit centralised control of building activity is prohibitive (e.g., Martel and Mohammadi, 2010). The ultimate value of swarm construction will be demonstrated in the real world with physical agents (or perhaps software agents working with real-world digital media). However, our interest is in exploring different possibilities for decentralised control of swarm construction in abstract simulated environments populated by idealised simplistic agents. The goal of such simulations is not to demonstrate solutions to specific realistic construction challenges, but to capture elements of the fundamental logic of decentralised control.

Here, we explore a population of simple simulated agents that combine information from two sensory modalities (one proximal and one distal) in order to overcome some of the limitations of two previously explored uni-modal schemes. Like the artificial paper wasps of Bonabeau et al. (2000), the agents simulated here are able to sense the configuration of building material in their immediate environment and use this proximal sensory information to trigger specific building activity via a set of *microrules*. In addition, like the simulated termites of Ladley and Bullock (2004, 2005), they are also able to sense simulated diffusing artificial pheromones deposited during building and movement, and use this distal sensory information to influence movement and release or inhibit building activity. Since both the proximal configuration of building material and the distal distribution of pheromone intensities in an agent's vicinity are themselves the consequence of prior agent building activity, the scheme is *stigmergic*—the environmental trace of agent activity guides subsequent agent behaviour.

Movement and building activity are constrained by a simple physics such that agents cannot pass through building material and must remain in contact with the ground or built structure. Moreover, new building material may only be deposited in locations with sufficient support. In contrast to Grushin and Reggia (2006), these constraints, while simplistic, do *not* prevent concave, hollow or over-hanging structures.

In principle, this swarm construction scheme is "universal" in that it is capable (given enough distinct types of building material) of generating *any* configuration of contiguous building material—a property inherited from Bonabeau et al. (2000)'s scheme. However, proofs of universality tell us nothing about what a scheme will in fact be useful for in practice (Bullock, 2006). Consequently, we concentrate here on exploring and describing the scheme's *generic* behaviour: what classes of structure are readily built and why; conversely, what kinds of structure require a prohibitively complex set of building materials, pheromones, microrules, etc.

Here, using hand-designed agents we are able to show that, unlike Ladley and Bullock's (2004, 2005) termites, the addition of proximal microrules enables agents to construct both simple conic *and* rectilinear structures such as domes, arches, pillars, cubes and frames (see figure 1 for examples of the latter), and that they are able to combine these structures relatively easily (see figure 2). Moreover, we are also able to show that, unlike Bonabeau et al's (2000) wasps, the addition of distal pheromone-mediated behaviour enables agents to construct architectures exhibiting long-range structure without recourse to a prohibitive number of block types (as required by, e.g., Howsman et al., 2004), and that these structures can be easily scaled in size through manipulation of pheromone parameters. However, complex structures still present challenges in terms of managing interactions between agents obeying different rule-sets, and timing issues related to the establishment of pheromone templates before the initiation of pheromone-template-mediated building activity.

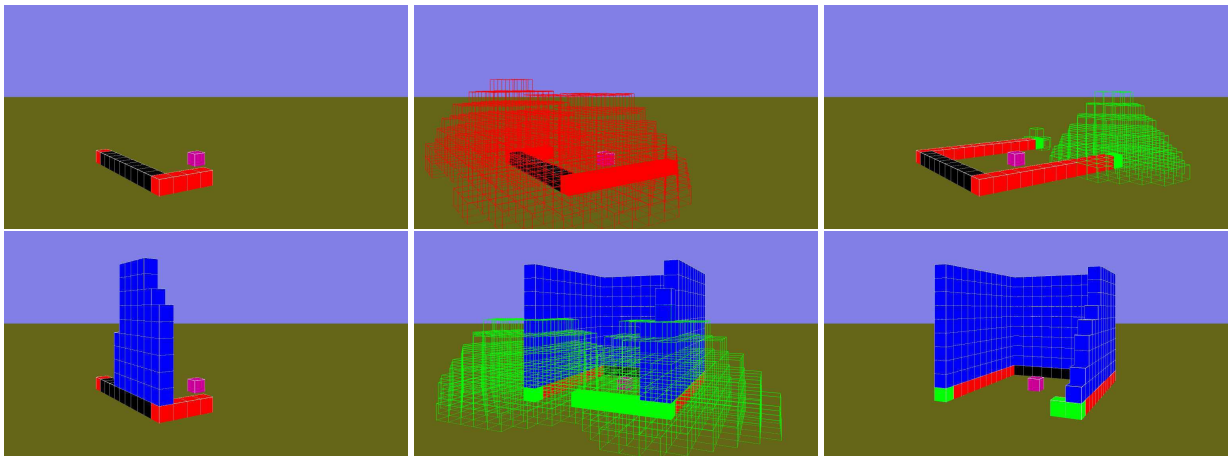


Figure 1: Stages in the formation of a square frame (top row), and a hollow cube (bottom row). In both cases building is initiated by the placement of a single block (depicted in magenta) in the centre of the ground plane. Distinct types of building material are represeted by solid cubes of different colours. Distributions of distinct types of pheromone are indicated by wire-frame cubes of different colours. Builder agents are not depicted.

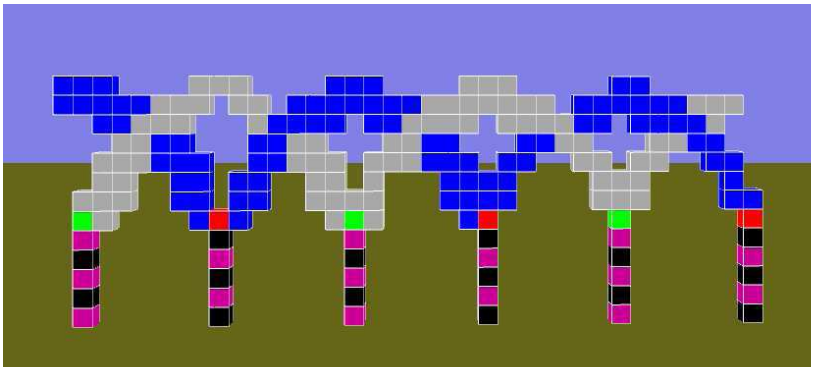


Figure 2: A series of interleaving arches mounted on a row of columns.

References

Bonabeau, E. W., Guérin, S., Snyers, D., Kuntz, P., and Théraulaz, G. (2000). Three-dimensional architectures grown by simple 'stigmergic' agents. *BioSystems*, 56:13–32.

Bullock, S. (2006). The fallacy of general purpose bio-inspired computing. In Rocha, L. M., Yaeger, L. S., Bedau, M. A., Floreano, D., Goldstone, R. L., and Vespignani, A., editors, *Proceedings of the Tenth International Conference on Artificial Life*, pages 540–545. MIT Press, Cambridge, MA.

Grushin, A. and Reggia, J. A. (2006). Stigmergic self-assembly of prespecified artificial structures in a constrained and continuous environment. *Integrated Computer-Aided Engineering*, 13(4):289–312.

Howsman, T. G., O'Neil, D., and Craft, M. A. (2004). A stigmergic cooperative multi-robot control architecture. In Pollock, J., Bedau, M., Husbands, P., Ikegami, T., and Watson, R. A., editors, *Proceedings of the Ninth International Conference on Artificial Life*, pages 88–93. MIT Press, Cambridge, MA.

Ladley, D. and Bullock, S. (2004). Logistic constraints on 3D termite construction. In Dorigo, M., editor, *Fourth International Workshop on Ant Colony Optimization and Swarm Intelligence*, pages 178–189. Springer-Verlag, Berlin.

Ladley, D. and Bullock, S. (2005). The role of logistic constraints in termite construction of chambers and tunnels. *Journal of Theoretical Biology*, 234:551–564.

Martel, S. and Mohammadi, M. (2010). Using a swarm of self-propelled natural microrobots in the form of flagellated bacteria to perform complex micro-assembly tasks. In *IEEE International Conference on Robotics and Automation.*

From Infotaxis to Boids-like Swarm Behaviour

Christoph Salge[1] and Daniel Polani[1]

[1]University of Hertfordshire, Hatfield, UK, AL10 9AB
c.salge@herts.ac.uk
d.polani@herts.ac.uk

Abstract

The three rules of alignment, separation and cohesion, introduced by Reynolds (1987) to recreate flocking behaviour have become a well known standard to create swarm behaviour. We aim to demonstrate that those three rules can emerge from the principle of information maximisation. We begin with a single agent looking for a specific location (i.e. a food source), its actions governed by a modified version of the Infotaxis behaviour introduced by Vergassola et al. (2007). Every action is selected to maximise the expected gain in information in the coming step. In Salge and Polani (2009, 2010) we demonstrated that this leads, without an explicit intent to communicate, to a "concentration" of "Relevant Information"(Polani et al. (2001, 2006)) in the agents' actions. In a multi-agent scenario it therefore becomes interesting, from an information theoretic (Shannon (1948)) perspective, to look at another agent's actions. We further demonstrated, that Bayes' Formula can be used to update the internal probability mapping of the food source using the other agents' actions, leading to an increase in agent performance and information gain per time.

So far, we only used the other agents' information when we encountered them incidentally. But it seems reasonable, as our behaviour is motivated by maximising the expected information gain, to include the expected position of other agents, and the expected gain of information from observing them, into our decision making process. Looking now at a multi-agent, grid-world scenario where all agents act with this new policy we can observe the emergence of some coordinated behaviour via local decision making of the agents. A closer analysis shows not only a further increase in performance, but also an increase in local agent density around the agent and an alignment of the overall direction the agents move in. Also, even though the agents are interested in being close to other agents to gain information from them, there is also some force that still separates them, since we rarely observe all agents congregating on one single spot and staying there.

Those measurements suggest that we are observing a behaviour that could - in spirit - also be created by the well-known three rules of "Boids" behaviour introduced by Reynolds (1987). The *cohesion* that makes agents move towards the average position of the local flock mates is recreated by the agent's motivation to have as many agents as possible in its sensor range, so it can profit from the information in their actions. The *separation* on the other hand, the aversion of the agents to get too close to others, is motivated by the lack of new environmental information around observed agents. Even though an agent's action is rich in information, it mostly provides information of its immediate surroundings. So, while some agent at the end of an agent A's sensor range would provide it with further information, an agent that is close to A can mostly display information that A has already acquired. Finally, *alignment* can be explained by realising that if an agent moves in a given direction, the goal is more likely to be there, and all else being equal, another agent should have a tendency to move in that direction as well.

References

Polani, D., Martinetz, T., and Kim, J. T. (2001). An information-theoretic approach for the quantification of relevance. In *ECAL '01: Proceedings of the 6th European Conference on Advances in Artificial Life*, pages 704–713, London, UK. Springer-Verlag.

Polani, D., Nehaniv, C. L., Martinetz, T., and Kim, J. T. (2006). Relevant information in optimized persistence vs. progeny strategies. In *Artificial Life X : Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, pages 337–343. The MIT Press (Bradford Books).

Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH Comput. Graph.*, 21(4):25–34.

Salge, C. and Polani, D. (2009). Information theoretic incentives for social interaction. Technical Report 495, University of Hertfordshire. presented at ECAL Workshop on Organisation, Cooperation and Emergence in Social Learning Agents.

Salge, C. and Polani, D. (2010). Digested information as an information theoretic motivation for social interaction. *Journal of Artificial Societies and Social Simulation (JASSS)*. accepted.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423.

Vergassola, M., Villermaux, E., and Shraiman, B. I. (2007). 'infotaxis' as a strategy for searching without gradients. *Nature*, 445(7126):406–409.

The Emergence of Complex Oscillatory Behaviour in *Physarum polycephalum* and its Particle Approximation

Soichiro Tsuda[1*] and Jeff Jones[1*]

[1]Unconventional Computing Group, Bristol Institute of Technology,
University of the West of England, BS16 1QY, United Kingdom
So.Tsuda@uwe.ac.uk

Abstract

The regeneration process of contractile oscillation in the plasmodium of *Physarum polycephalum* is investigated experimentally and modelled computationally. When placed in a well, the *Physarum* cell restructures the body (fusion of small granule-like cells) and shows various complex oscillation patterns. After it completed the restructuring and regained synchronised oscillation within the body, the cell shows bilateral oscillation or rotating wave pattern. This regeneration process did not depend on the well size and all the cases showed similar time course. A particle-based computational model was developed in order to model the emergence of oscillation patterns. Particles employing very simple and identical sensory and motor behaviours interacted with each other via the sensing and deposition of chemoattractants in a diffusive environment. From a random and almost homogeneous distribution, emergent domains of oscillatory activity emerged. By increasing the sensory radius the model simulated the regeneration process of the real plasmodium. In addition, the model replicated the rotating wave and bilateral oscillation pattern when the sensory radius was increased. The results suggest that complex emergent oscillatory behaviours (and thus the high-level systems which may utilise them, such as pumping and transport mechanisms) may be developed from simple materials inspired by *Physarum* slime mould.

Introduction

A plasmodium of true slime mould *Physarum polycephalum* is a multi-nuclear single-cellular organism. In the plasmodial state, the *Physarum* slime mould does not have any fixed shape and it lives as an amorphous amoeba-like organism. Being a single cell, it does not have any brains or neurons or any central controlling system. Nevertheless it is able to react to external stimuli by changing the body shape without losing control as a single cell. In other words, the *Physarum* plasmodium is an example of natural distributed computing system. Based on this fact, there has been a lot of research on the plasmodium from computational perspective. For example, it has been shown that the plasmodium can form an optimal tube network (Tero et al., 2010), compute planar proximity graphs (Adamatzky, 2008), and anticipate periodic events (Saigusa et al., 2008). The cell was also used to implement computational systems, such as basic logic gates (Tsuda et al., 2004), storage modification machine (Adamatzky, 2007), coupled oscillator system (Takamatsu et al., 2000b), and neural network system (Aono and Hara, 2007).

One of goals of these computational approaches to slime mould dynamics, termed as *Physarum* computing (Nakagaki, 2010), is to elucidate mechanisms of biological algorithm in the form that can be applied for bio-inspired computation, such as swarm intelligence (Bonabeau et al., 1999). A few approaches have already been taken towards this goal (e.g. Tero et al., 2006; Ishiguro et al., 2004).

So far it is known that the underlying mechanism which enables the primitive intelligent behaviour is intrinsic cellular oscillation. The *Physarum* plasmodium shows a cell thickness oscillation which period spans around 1-2 minutes. Any external stimuli impinging on the cell's behaviour (food, chemical, thermal, etc) are "encoded" as modulation of local oscillation rhythms. The local change in oscillation frequency propagates to other parts of the cell through protoplasmic streaming and is eventually "interpreted" as behavioural changes, such as locomotion towards food or shape changes (Miyake et al., 1996). Therefore, without the oscillation, the plasmodium is not able to perform any computations.

Our particular interests in this paper are two fold: (1) to experimentally investigate the generation of the contractile oscillation and (2) to develop a computational model that replicates the process. When innoculated onto an agar gel, a piece of *Physarum* plasmodium starts to reorganise the body structure in order to resume oscillating. Takagi and Ueda (2008) found that a small plasmodium cell shows various dynamic oscillation patterns in the course of body restructuring. As behaviours of the plasmodium is said to be size-invariant (Miura and Yano, 1998), we investigated the effect of size on the dynamic patterns and modelled them using a swarm-based particle model.

*These authors contributed equally to this work.

The Generation of Oscillatory Behaviour in the *Physarum* plasmodium

To observe how the plasmodium (re-)generates the contractile cell volume oscillation, a piece of the *Physarum* plasmodium is cut from one of growing tips of a larger culture and then placed in a well constructed with a 1.5 % agar gel and a transparency sheet (Fig. 1) in a Petri dish. The plasmodium tends to stay inside a well where agar gel is exposed because it prefers wet areas to dry ones. Immediately after placed in a well, the cell was placed under a microscope (Leica Zoom 2000, Germany) and illuminated from underneath with monochromatic light of wavelength 600 nm. The *Physarum* plasmodium is known to be insensitive to the wavelength of light in terms of the cellular oscillation activity (Nakagaki et al., 1996). A microscope camera image was taken every 3 seconds for over 5 hours. As the brightness level of a pixel in an image is inversely proportional to the thickness of the cell, the relative thickness oscillation can be calculated by image analysis. We tested 1.6, 3.2, and 4.5mm diameter wells and oscillation patterns of the plasmodium in those wells are compared.

Images were analysed with following process: First, each colour snapshot image was converted to a grey-scale image in which a pixel has a value corresponding to light intensity. Then a spatio-temporal moving average filter was applied over each snapshot image, which effectively works as a low-pass filter to reduce camera flicker noise. The window size used in this case was 41x41 pixels (spatial) and 5 images (temporal). Finally the relative thickness at time t was calculated as $\Delta s(t) = s(t) - s(t + \Delta t)$, where $\Delta s(t)$ is an image of extracted relative thickness at time t, and $s(t)$ and $s(t + \Delta t)$ are grey-scaled images at time t and $t + \Delta t$, respectively. $\Delta t = 7$ was chosen empirically.

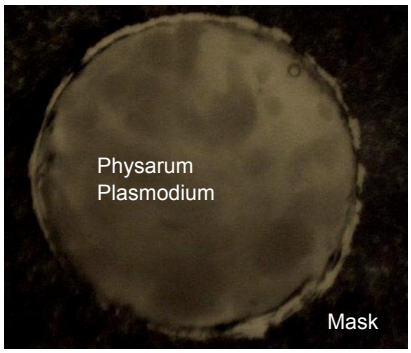


Figure 1: Picture of a *Physarum* plasmodium on 1.5 % agar gel. The cell is allowed to move only inside a circular well of a mask. The well diameter is 1.6mm in this example.

Results

A portion of *Physarum* cell in a well consists of small dark granules and transparent parts, as seen in Fig. 1. The transparent parts are considered to be extracellular material (slime) coating cell's body. A plasmodium in a larger well (e.g. 4.5mm) contains more granules in it.

A typical time course of the plasmodial contraction regeneration was as follows: Within 10 minutes, the plasmodium starts contractile oscillation. At this stage, each granule independently shows contractile oscillation within itself, but the oscillation rhythms appears to be unsynchronised to oscillations of other granules (Fig. 2a). Gradually small granules start to merge together with neighbouring granules and the independent oscillations start to synchronise accordingly (Fig. 2b). As a result, an area within which a synchronised oscillation is observed gradually extends over time until the whole cell in a well shows a synchronised oscillation (Fig. 2c). To illustrate this, *Physarum* thickness oscillation on a line (a grey arrow in Fig. 2b) is plotted against time (for 1 hour from the start of measurement), shown in Fig. 2d. This space-time plot shows how a globally synchronised pattern emerges in the plasmodium. As mentioned above, small granules oscillate independently at an early stage of the experiment. There are two oscillating granules on the line, one in the upper part and another small one in the bottom part of the plot, which are indicated as by gray rectangles in Fig. 2d. These two parts become larger and larger over time. This means the area exhibiting synchronous oscillation is gradually expanding. Approximately after 30 minutes from the start, the spatio-temporal pattern becomes somewhat chaotic (the period around (b)). Although various types of complex oscillating patterns can be observed in this period, there are a few areas where synchronised oscillation can be observed (In the case of Fig. 2b, roughly 3 synchronised areas can be found). This period can be interpreted as a "resetting" phase in which merged granules are reconstructing the whole body structure and preparing to become one single cell prior to the whole synchronised phase. Those areas eventually synchronise together and the whole cell shows a synchronise oscillation. After it reached the phase, there were mainly 2 types of oscillation patterns observed: bilateral oscillation (anti-phase oscillation between two halves of a well like Fig. 2c) and rotating wave pattern (clockwise or anti-clockwise). Only in the case of 4.5mm well, a convective pattern (two rotating wave colliding at the centre of a well) was observed. These oscillation patterns were constantly switching one to the other after a couple of cycles. Figure 3 illustrates such frequently changing patterns. In this case, it shows a bilateral oscillation at first (Fig. 3a). The pattern soon switches to a clockwise rotating wave (Fig. 3b), followed by an anti-clockwise rotating wave (Fig. 3c). During the period plotted, the cell has been already settled in a globally synchronised phase. Figure 3d is a spatio-temporal thickness oscillation plot along a circle indicated in Fig. 3bc. The bilateral pattern (Fig. 3a) is represented as checkerboard-like patterns where upper and lower halves show alternating stripe patterns (gray

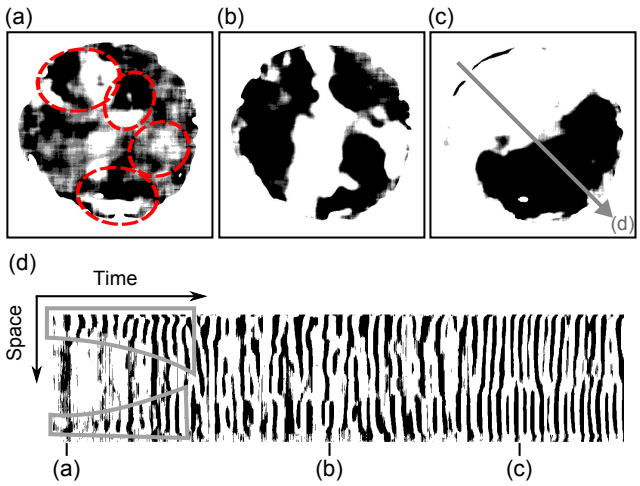


Figure 2: (a) An example of plasmodial oscillation pattern in 1.6mm well at an early stage. Black regions indicate thickness is increasing whereas white ones is decreasing. Synchronised areas are indicated by dotted circles. (b)(c) Snapshot images taken after 30 and 50 minutes, respectively. (d) space-time plot of *Physarum* oscillation along an arrow in (c). First 1 hours from the start is plotted. (a-c) in the plot corresponds to the above snapshots of *Physarum* thickness oscillation (a-c).

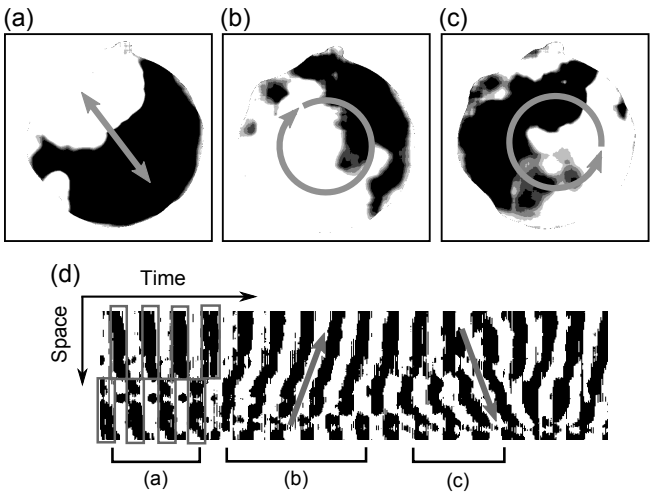


Figure 3: (a) space-time plot in (e) is plotted along the grey arrow, 360 points. (b) Bilateral oscillation (c) clockwise oscillation (d) anti-clockwise oscillation of a *Physarum* plasmodium in 4.5mm well. (e) space-time plot of *Physarum* oscillation. (b-d) in the plot corresponds to the period when above oscillation patterns were observed.

squares in Fig. 3d). The clockwise and anti-clockwise patterns are forwardslash and backslash stripe patterns (indicated by gray arrows).

All the 3 well sizes investigated here showed common time course of the oscillation regeneration as described above. However, in general, a plasmodium in a larger well took longer time to reach the whole synchronised phase due to the physical size. The average lengths to settle into a whole synchronised phase from the start were 0.95, 1.32, and 1.68 hours for 1.6, 3.2, and 4.5mm wells, respectively.

Takagi and Ueda (2008) observed oscillation patterns of unbounded *Physarum* cells (approximately 1.5mm diameter) during the regeneration of contractile oscillation and identified 4 distinctive patterns: standing wave, many drifting spirals, one or two stable spirals, and synchronous oscillation. Although their condition is similar to ours, in particular the case of 1.6mm well, our experiments did not confirm all the patterns they reported. Two possible reasons can be considered for this: First, they used *Physarum* plasmodia in liquid form obtained from protoplasmic veins, whereas ours are from growing tips. As a liquid plasmodium does not have any granule-like structures, it starts as a uniform cell to resume the contractile oscillation, which may leave out the granule fusing process in our observation. Another possible reason is the boundary for the cell. In (Takagi and Ueda, 2008), they observed plasmodial oscillation simply placed on a plain agar gel. On the other hand, in our setup cells are constrained to move only within a well. This may well have affected the way a plasmodium oscillates, as it is empirically known that the *Physarum* plasmodium shows a stable and sustained oscillation pattern when it is free to move and grow. This would partly explain the frequent pattern change observed in this paper. Because of surrounding walls, the movement of the plasmodium is constantly blocked and it may have led to the frequent pattern change in the globally synchronised phase. In fact, it has also been observed that *Physarum* cells in 3.2 and 4.5mm wells develop tiny mushroom-like 3D structures (pseudopods growing vertically) in the phase. This is possibly because the cell is not able to grow horizontally.

A Particle Approach to the Generation of Oscillatory Behaviour

To investigate and replicate the emergence of oscillatory behaviour within the plasmodium we employ and extend the particle model in (Jones, 2010b) which was used to generate dynamical emergent transport networks. The approach uses a population of mobile particles with very simple behaviours, residing within a 2D diffusive environment. The discrete 2D lattice (where the features of the environment are mapped to grey-scale values in a 2D image) stores particle positions and the concentration of a local factor which we refer to generically as chemoattractant. The 'chemoattractant' factor actually represents the hypothetical flux of sol within the plasmodium. Free particle movement represents the sol phase of the plasmodium. Particle positions

represent the fixed gel structure (i.e. global pattern) of the plasmodium. The particles act independently and iteration of the particle population is performed randomly to avoid any artifacts from sequential ordering. The behaviour of the particles occurs in two distinct stages, the sensory stage and the motor stage. In the sensory stage, the particles sample their local environment using three forward biased sensors whose angle from the forwards position (the sensor angle parameter, SA), and distance (sensor offset, SO) may be parametrically adjusted (Fig. 4a). The offset sensors represent the overlapping and intertwining filaments within the transport networks and plasmodium, generating local coupling of sensory inputs and movement (Fig. 4c,d). The SO distance is measured in pixels and a minimum distance of 3 pixels is required for strong local coupling to occur. During the sensory stage each particle changes its orientation to rotate (via the parameter rotation angle, RA) towards the strongest local source of chemoattractant (Fig. 4b). After the sensory stage, each particle executes the motor stage and attempts to move forwards in its current orientation (an angle from 0–360 degrees) by a single pixel forwards. Each lattice site may only store a single particle and—critically—particles deposit chemoattractant into the lattice only in the event of a successful forwards movement (Fig. 5a). If the next chosen site is already occupied by another particle the default (i.e. non-oscillatory) behaviour is to abandon the move and select a new random direction (Fig. 5b). Diffusion of the collective chemoattractant signal is achieved via a simple 3x3 mean filter kernel with a damping parameter (set to 0.07) to limit the diffusion distance of the chemoattractant.

The low level particle interactions result in complex pattern formation. The population spontaneously forms dynamic transport networks showing complex evolution and quasi-physical emergent properties, including closure of network lacunae, apparent surface tension effects and network minimisation. An exploration of the possible patterning parameterisation was presented in (Jones, 2010a).

Although the particle model is able to reproduce many of the network based behaviours seen in the *Physarum* plasmodium such as spontaneous network formation, shuttle streaming and network minimisation, the default behaviour does not exhibit oscillatory phenomena and inertial surging movement, as seen in the organism. This is because the default action when a particle is blocked (i.e. when the chosen site is already occupied) is to randomly select a new orientation—resulting in very fluid network evolution, resembling the relaxation evolution of soap films, and the lipid nanotube networks seen in (Lobovkina et al., 2008).

The oscillatory phenomena seen in the plasmodium are thought to be linked to the spontaneous assembly / disassembly of actin-myosin and cytoskeletal filament structures within the plasmodium which generate contractile forces on the protoplasm within the plasmodium. The resulting shifts between gel and sol phases prevent (gel phase) and promote

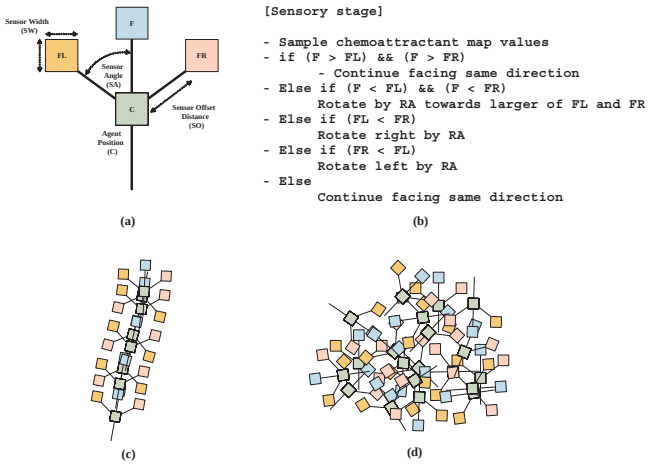


Figure 4: Particle morphology and schematic illustration of overlapping particle positions representing transport networks and plasmodium mesh. (a) Morphology showing agent position 'C' and sensor positions (FL, F, FR), (b) Algorithm for particle sensory stage, (c) Transport network formation, (d) Overlapping sensors representing plasmodium mesh.

(sol phase) cytoplasmic streaming within the plasmodium. To mimic this behaviour in the particle model requires only a simple change to the motor stage. Instead of randomly selecting a new direction if a move forward is blocked, the particle increments separate internal coordinates until the nearest cell directly in front of the particle is free. When a cell becomes free, the particle occupies this new cell and deposits chemoattractant into the lattice (Fig. 5c). The effect of this behaviour is to remove the fluidity of the default movement of the population. The result is a surging, inertial pattern of movement, dependent on population density (the population density specifies the initial amount of free movement within the population). The strength of the inertial effect can be damped by a parameter (pID, set to 0.05 for all experiments) which sets the probability of a particle resetting its internal position coordinates, lower values providing stronger inertial movement.

When this simple change in motor behaviour is initiated surging movements are seen and oscillatory domains of chemoattractant flux spontaneously appear within the virtual plasmodium showing characteristic behaviours: temporary blockages of particles (gel phase) collapse into sudden localised movement (solation) and vice versa. The oscillatory domains themselves undergo complex evolution including competition, phase changes and entrainment. We utilise these dynamics below to reproduce the oscillatory patterns seen in the *Physarum* plasmodium at different well sizes.

The particle lattice was configured to reflect the environment of a single well containing and confining the plasmod-

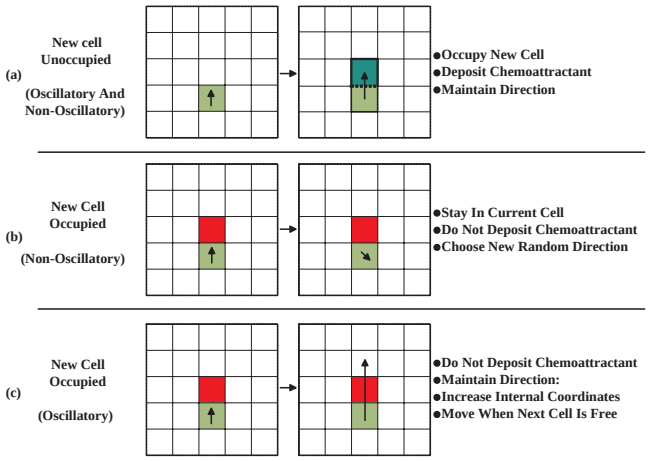


Figure 5: Particle motor behaviour in non-oscillatory and oscillatory modes. (a) Behaviour in both modes is identical when new site is unoccupied, (b) When the new site is occupied in non-oscillatory mode a new random direction is selected, (c) When the new site is occupied in oscillatory mode the particle increments an internal position counter at every subsequent motor step until a new site in the current direction becomes free.

ium. Movement was prevented outside this region (specifically, if the border region was encountered, a random change in direction was made). The population size was fixed at 90% of the well size, leaving 10% of the free space available for movement. No growth/shrinkage rules were implemented for these experiments. The results show patterns of the concentration of chemoattractant flux within the population. Areas of greater flux are shown as darker regions. Since deposition of chemoattractant only occurs when movement if successful the concentration relates to the amount of active transport caused by oscillations in plasmodium thickness. This is indirectly related to thickness changes of the plasmodium detected in laboratory conditions - there is a reciprocal relationship between contraction of the plasmodium in a local region and subsequent transport of material from that region, as noted by (Takamatsu et al., 2000a). Due to the complex evolution of the patterns the reader is encouraged to refer to the online supplementary recordings at (Jones, 2010c).

Results

Initial experiments with the sensory parameters SA and RA showed that a wide range of values yielded complex oscillatory patterns (see supplementary video recordings for examples). The differences in base pattern type at different SA-RA combinations were caused by differences between sensor arm angle and rotation angle. Whichever SA-RA was used there was a common evolution to all experiments.

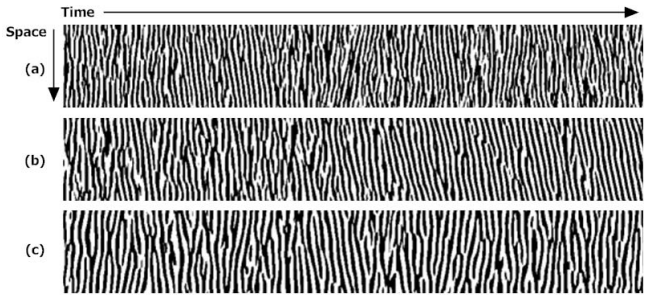


Figure 6: A constant SO parameter during an experimental run results in no significant changes in pattern type. Experiment iterated for 10,000 steps. Plots were sampled from a circular region at the centre of the well at half the well radius. Well size were all 200 pixels, SO for each run: (a) 9 pixels, (b) 21 pixels, (c) 41 pixels

There was an initial period where multiple foci of oscillating flux appeared. These small regions gradually exerted an influence upon each other and entrainment of patterns was seen. The size of the entrained regions depended upon both the SO parameter (sensory radius) and the well size. We selected a small sample from the parameter ranges (specifically SA 22.5 degrees and RA 45 degrees) in an attempt to explore the complex experimentally observed phase transitions. These SA-RA settings were used because, when considering the transport networks, they generated foraging-like behaviour (Jones, 2010a). Grey-scale output images from the model were saved every 10 iterations and a spatio-temporal moving average and thickness extraction for space-time plots were calculated as per the experimental method above.

When a fixed value was used for the Sensor Offset (SO) scale parameter, there was an initial period of chaotic interactions until a stable type of oscillatory pattern predominated. Occasionally the oscillatory behaviour was interrupted, however variations on this pattern were then observed throughout the time course of the simulation (Fig. 6). Although the fixed SO parameter was able to successfully generate emergent oscillatory behaviours, there was no predictable transition between the pattern types observed experimentally. When higher values of SO were used (with identical SA-RA) fewer independent foci of oscillations were seen. When the SO parameter increased significantly the type of oscillation pattern changed. This supports the idea that the independent domains in the plasmodium interact over an increasingly large scale.

To reproduce the experimental observation of the growth and fusion of oscillatory domains, and resultant change in pattern type, we gradually increased the SO parameter during the experiment. Beginning with an SO value of 3 pixels, the SO parameter for all particles was increased by 3

pixels every 500 iterations of the model. This resulted in a larger local sensory radius for each particle, causing the behaviour to be influenced by local particles at larger distances. An entrainment of movement was observed as the collective sensory coupling increased. The results showed clear transitions between different pattern types which were observed visually and in terms of the space-time plots (Fig. 7). The order of pattern transition tended to be: 1. Chaotic behaviour, 2. Interacting domains, 3. Rotational pattern, 4. Bilateral synchronisation, and 5. Pulsatile annular pattern. However, as with the experimental plasmodium, some reversion to earlier patterns was also observed. At the smallest well size (100 pixels) entrainment of the entire particle collective occurred relatively quickly (Fig. 7a). The rotational patterns within this small well were two rotating halves of the well. Larger wells produced 'propeller-like' rotational patterns, with increasing numbers of vanes as well sized increased. Synchronous oscillations (both bilateral and later with a pulsatile annular pattern) were observed some time after the rotational patterns. When larger well sizes were used, there was a longer time period before transition between pattern types. This can be seen from the phase plots in Fig. 7b and c, which show increasing delays before the onset of rotational patterns. The effect of the larger well size is also evidenced by the rather fragmented aspect to the phase plots which indicate a weaker initial coupling between different regions (Fig. 7d. Although the model was able to replicate the oscillatory patterns and transitions, there appeared to be some limitation on the maximum well size for entrainment of the particle population to completely occur. With the largest well size (400 pixels), the phase plots indicate the regions stay independent for much longer periods. When SO was very large (greater than 80 pixels) the large scale oscillations became frozen and the only flux of particles was along narrow domains within the collective. Whether this behaviour is a feature of the real plasmodium, or merely a modelling artifact, requires further investigation.

The phase plots of the regular periods of oscillation patterns seen with SA 22.5 and RA 45 (rotation, bilateral and annular synchronisation) can be seen in Fig. 8. Animated video recordings of the entire well phase patterns and transitions can be seen in the supplementary material. Experiments using other SA-RA settings produced other oscillatory patterns, including the convective oscillatory seen in the 4.5mm well experiments. Experiments with the particle model suggest that the causes of the changes in oscillatory regimes (and the reversion to previous patterns) may be the gradual increase in sensory influence. As the SO parameter increases previously separated oscillators begin to interact and some begin to predominate. The increase in sensory influence also appears to reduce the freedom of movement of the oscillatory patterns. From an informal observation the initially separate oscillatory bodies adopt spiral and circular paths. These independent circular paths then fuse into

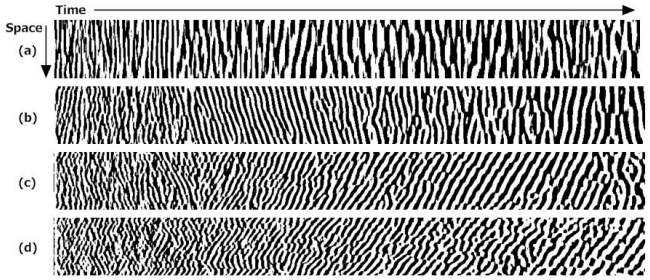


Figure 7: When increasing SO parameter during an experiment, well diameter affects pattern types, transitions and timing of transitions. Experiment iterated for 10,000 steps. Plots were sampled from a circular region centre in the middle of the well at a size half the well radius. Well sizes: (a) 100 pixels, (b) 200 pixels, (c) 300 pixels, d) 400 pixels

a single circumferential path (rotation pattern). The scope for movement is further reduced by the emergence of synchronous oscillations (movement is limited by the diameter of the well in bilateral oscillations, and to a radius distance with annular oscillations). This observation is difficult to quantify, however, and does not simply explain the reversion to previous patterns which possess greater freedom of movement. It is plausible that, just as there appears to be a mechanism within the plasmodium for increasing influence over distance, there may be another opposing mechanism which decreases influence over distance. The polymerisation/depolymerisation of actin filaments within the plasmodium could be one (speculative) mechanism of increasing/decreasing the region of influence.

Discussion and Conclusion

We experimentally investigated the regeneration process of the *Physarum* plasmodium in a well and computationally modelled oscillation patterns of the cell observed in the experiments using a particle model. It has been found that cells exhibited similar time course of oscillation regeneration independent of the well size. A granule-like cell works as an oscillator unit and by the fusion of granules the cell eventually reaches a state where all the parts in the cell are synchronised. Although the detailed synchronisation mechanism is yet to be investigated further, physiological findings of the cell suggest that there are two factors involved in the oscillation synchronisation (Kessler, 1982): Ectoplasmic local contraction and endoplasmic flow. The ectoplasm (gel phase) of the *Physarum* protoplasm contains actin in filamentous form (F-actin). This molecule is periodically polymerised or fragmented, which creates cell contraction and relaxation rhythm in a local part of the cell. The endoplasm (sol phase) flow generated by the contraction rhythm mediates the oscillation synchronisation between local parts, otherwise local rhythms do not synchronise at all (Yoshi-

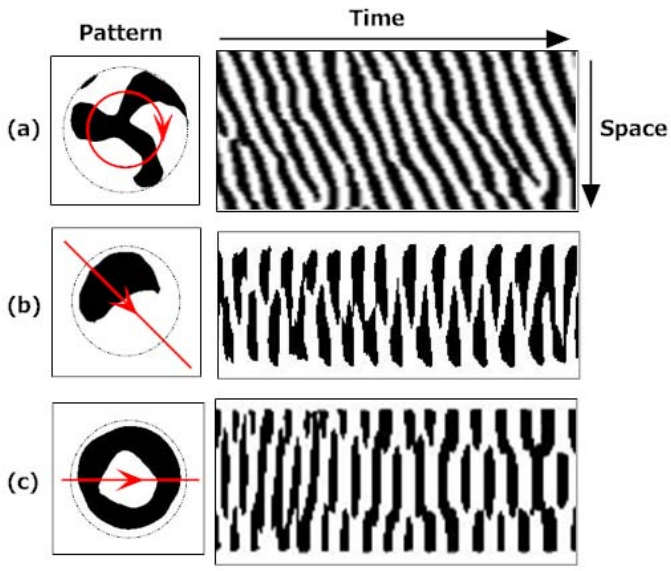


Figure 8: Characteristic oscillation patterns observed within the particle model. Left side indicates pattern type and sample through virtual plasmodium. Right side indicates space-time plot. (a) Rotating pattern observed in 200 pixel well, (b) Bilateral oscillation observed in 100 pixel well, (c) Synchronous annular pattern observed in 100 pixel well.

moto and Kamiya, 1978). In the experiments with real plasmodium cells, we observed that small granular cells showing independent oscillations in the beginning were gradually synchronised with time. Given the physiological findings above, our observation can be considered as a process of the endoplasmic flow network development, which coordinates the synchronisation between granular cells. In our simulation, we observed that the particle model replicates this process well when the Sensor Offset (SO) parameter was gradually increased. As the SO parameter determines the interaction range between particles, the whole system with large SO value acquired an (amorphous) interaction network, which effectively corresponds to the endoplasmic flow network in the plasmodium cell. The important factor to consider is that all the processes observed here (regarding both real and virtual slime moulds) emerged from the bottom-up local interactions between simple and identical components.

The amorphous nature of the *Physarum* plasmodium presents attractive possibilities from structural, computational and robotics perspectives. The plasmodium may be considered, on one hand, as a programmable material whose morphology may be specified and altered by +ve (chemoattractants, warmth) and -ve (chemorepellents, light etc.) stimuli. On the other hand, the material itself displays impressive and well documented computational properties which are also—to some degree—subject to external control. The computational possibilities of even small fragments of *Physarum* plasmodium arise from the same simple interactions and are distributed throughout the material, placing it in the category of programmable *and* functional bio-materials. Although there are numerous difficulties in trying to persuade the plasdmodium to adopt and indeed maintain the required structural and functional patterns, the simple low-level interactions which generate the emergent behaviours suggest that it may be possible to develop *Physarum*-like programmable-functional materials. Further work is in progress using the plasmodium and its oscillatory patterns for simple robotic devices and sub-components.

Acknowledgment

The research reported here is a part of the Leverhulme Trust funded project “Mould intelligence: biological amorphous robots”, directed by Prof. Andrew Adamatzky.

References

Adamatzky, A. (2007). Physarum machine: Implementation of a kolmogorov-uspensky machine on a biological substrate. *Parallel Processing Letters*, 17(4):455–467.

Adamatzky, A. (2008). Developing proximity graphs by physarum polycephalum: Does the plasmodium follow toussaint hierarchy? *Parallel Processing Letters*, 19:105–127.

Aono, M. and Hara, M. (2007). Amoeba-based nonequilibrium neurocomputer utilizing fluctuations and instability. In *6th International Conference, UC 2007*, volume 4618 of *LNCS*, pages 41–54, Kingston, Canada. Springer.

Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm intelligence: from natural to artificial systems*. Oxford University Press, Inc., New York, NY, USA.

Ishiguro, A., Shimizu, M., and Kawakatsu, T. (2004). Don’t try to control everything!: An emergent morphology control of a modular robot. In *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 981–985, Sendai, Japan.

Jones, J. (2010a). Characteristics of pattern formation and evolution in approximations of physarum transport networks. *Artificial Life*, 16:127–153.

Jones, J. (2010b). The emergence and dynamical evolution of complex transport networks from simple low-level behaviours. *International Journal of Unconventional Computing*, 6:125–144.

Jones, J. (2010c). http://uncomp.uwe.ac.uk/jeff/alifexii.htm.

Kessler, D. (1982). *Plasmodial structure and motility*, volume 1 of *Cell biology of physarum and didymium*, chapter 5. Academic Press.

Lobovkina, T., Dommersnes, P., Tiourine, S., Joanny, J., and Orwar, O. (2008). Shape optimization in lipid nanotube networks. *The European Physical Journal E*, 26:295–300.

Miura, H. and Yano, M. (1998). A model of organization of size invariant positional information in taxis of *Physarum* plasmodium. *Progress of Theoretical Physics*, 100(2):235–251.

Miyake, Y., Tabata, S., Murakami, H., Yano, M., and Shimizu, H. (1996). Environmental-dependent self-organization of positional information field in chemotaxis of *Physarum* plasmodium. *Journal of Theoretical Biology*, 178:341–353.

Nakagaki, T., editor (2010). *International Journal of Unconventional Computing. Special Issue: The Birth of Physarum Computing*. Old City Publishing.

Nakagaki, T., Uemura, S., Kakiuchi, Y., and Ueda, T. (1996). Action spectrum for sporulation and photoavoidance in the plasmodium of Physarum polycephalum, as modified differentially by temperature and starvation. *Photochemistry and Photobiology*, 64(5):859–862.

Saigusa, T., Tero, A., Nakagaki, T., and Kuramoto, Y. (2008). Amoebae anticipate periodic events. *Physical Review Letters*, 100(1).

Takagi, S. and Ueda, T. (2008). Emergence and transitions of dynamic patterns of thickness oscillation of the plasmodium of the true slime mold Physarum polycephalum. *Physica D*, 237:420–427.

Takamatsu, A., Fujii, T., and Endo, I. (2000a). Control of interaction strength in a network of the true slime mold by a microfabricated structure. *BioSystems*, 55:33–38.

Takamatsu, A., Fujii, T., Yokota, H., Hosokawa, K., Higuchi, T., and Endo, I. (2000b). Controlling the geometry and the coupling strength of the oscillator system in plasmodium of *Physarum polycephalum* by microfabricated structure. *Protoplasma*, 210:164–171.

Tero, A., Kobayashi, R., and Nakagaki, T. (2006). Physarum solver: A biologically inspired method of road-network navigation. *Physica A: Statistical Mechanics and its Applications*, 363(1):115–119.

Tero, A., Takagi, s., Saigusa, T., Ito, K., Bebber, D. P., Fricker, M. D., Yumiki, K., Kobayashi, R., and Nakagaki, T. (2010). Rules for biologically inspired adaptive network design. *Science*, 327(5964):439–442.

Tsuda, S., Aono, M., and Gunji, Y.-P. (2004). Robust and emergent *Physarum* logical-computing. *BioSystems*, 73:45–55.

Yoshimoto, Y. and Kamiya, N. (1978). Studies on contraction rhythm of the plasmodial strand iii. role of endoplasmic streaming in synchronization of local rhythms. *Protoplasma*, 95:111–121.

Collective Classification of Biomedical Articles using T-Cell Cross-regulation

Alaa Abi-Haidar[1,2] and Luis M. Rocha[1,2]

[1]School of Informatics and Computing, Indiana University, Bloomington IN 47401, USA
[2]FLAD Computational Biology Collaboratorium, Instituto Gulbenkian de Ciência, Oeiras, Portugal
{aabihaid,rocha}@indiana.edu

Abstract

We continue our investigation of a bio-inspired solution for binary classification of textual documents inspired by T-cell cross-regulation in the vertebrate adaptive immune system, which is a complex adaptive system of millions of cells interacting to distinguish between self and nonself substances. In analogy, automatic document classification assumes that the interaction and co-occurrence of thousands of words in text can be used to identify conceptually-related classes of documents—at a minimum, two classes with relevant and irrelevant documents for a given concept (e.g. articles with protein-protein interaction information). Our agent-based method for document classification expands the analytical model of Carneiro et al [5], by allowing us to deal simultaneously with many distinct populations of antigen-specific T-Cells and their collective dynamics. We have previously extended this model to produce a spam-detection system [2; 3]. We have also developed our agent-based model further to apply it to biomedical article classification [4], testing it on a dataset of biomedical articles provided by the BioCreative 2.5 challenge [17]. Here, we study the effect that the sequence of presentation of articles has on classification performance, as well as the robustness of the ensuing T-cell cross-regulation dynamics to initial biases of the proportions of effector and regulatory T-cells. We show that classification is improved when we preserve the original temporal order of biomedical articles, suggesting that our model is capable of tracking the natural conceptual drift of the relevant biomedical literature. We further show that initial biases in the proportions of T-cells are corrected by the dynamics of the model. Our results are useful for biomedical text mining, but they also help us understand T-cell cross-regulation as a potential general principle of classification available to collectives of molecules without a central controller. While there is still much to know about the specifics of T-cell cross-regulation in adaptive immunity, Artificial Life allows us to explore alternative emergent classification principles while producing useful bio-inspired tools.

Introduction

At least since the start of systematic genomic studies, there has been a tremendous growth of scientific publications in the life sciences [13]. Pubmed (`http://pubmed.gov`) now contains a growing collection of more than 19 million biomedical articles. Manually classifying these articles as relevant or irrelevant to a given topic of interest is very time consuming and inefficient for curation of new published articles [14]. Literature (or text) mining offers solutions for automatic biomedical document classification and information extraction from huge collections of text, as well as the linking of numerous biomedical databases and knowledge resources [14; 28]. Because it is very important to validate and assess the quality of proposed solutions, various community-wide competitions and challenges have been organized so that automatic systems can be evaluated against human annotated data sets (e.g. TREC Genomics [10]). One such effort is the BioCreative challenge, which aims to assess biomedical literature mining in real-world scenarios [11; 18; 17]. Machine learning has offered a plethora of solutions to this problem [14; 8], however, even the most sophisticated of solutions often overfit to the training data and do not perform as well on real-world scenarios such as that provided by BioCreative [1; 16]. One of the challenges of biomedical article classification in real-world scenarios is the presence of highly unbalanced classes; typically, there are many more irrelevant than relevant documents, without prior knowledge of class proportions. This was the case of the article classification data set in the Biocreative BC2.5 challenge [17]. While participating teams (including our own team [16]) did not enter bio-inspired solutions, the unbalanced nature of classes and the presence of conceptual drift, which we showed to occur between training to testing data sets [1; 16], may be a good scenario to test classifiers inspired by the vertebrate immune system—which must operate under class-imbalance with permanent drift in the populations of pathogens encountered. Therefore, here we explore the feasibility of using T-Cell cross-regulation dynamics to classify biomedical articles using the real-world scenario provided by the Biocreative 2.5. data set.

The immune system (IS) is a complex biological system made of millions of cells all interacting to distinguish between self and nonself substances, to ultimately attack the latter [12][1]. In analogy, relevant biomedical articles for a

[1]We use the terminology of self/nonself discrimination, though perhaps a more accurate description is classification of harmless

given concept need to be distinguished from irrelevant ones. To perform such a topical classification, we can use the occurrence and co-occurrence of thousands of words in a document. In this sense, words can be seen as interacting in a text in such a way as to allow us to distinguish between relevant and irrelevant documents—in analogy with the interactions among T-cells and antigens that lead to self/nonself discrimination in the immune system, as we describe below.

Our Artificial Life approach is based on the idea that the immune system is a distributed collection of molecular constituents with no central controller [25]. Therefore, its classification ability needs to result from a *collective classification* process, defined as the ability of decentralized systems of many components to classify situations that require global information or coordinated action [20]. Nature is full of examples of collective classification: the dynamics of stomata cells on leaf surfaces are known to be statistically indistinguishable from the dynamics of automata that are capable of performing nontrivial classification [21], biochemical intracellular signal transduction networks are capable of emergent classification [9], quorum sensing in bacteria [33] and social insects [23], etc. We can study collective classification in general models of complex systems such as Cellular Automata, namely by identifying regular patterns in the dynamics that store, transmit and process information [6; 24; 27]. Here, instead of looking at general models of complex systems, we focus on a specific immunological model of T-Cell cross-regulation dynamics [5]. We are are interested in exploring the collective dynamics of this model to: (1) build a novel bio-inspired machine learning solution for document classification, and (2) understand how well collections of T-Cells engaged in cross-regulation perform as a classifier. The first goal entails a bio-inspired approach to computational intelligence, and the second a computational biology experiment, but both are based on artificial life principles. It should be noted that recent work in artificial immune systems (AIS) [30] has lead to a few immune-inspired solutions to document classification in general [32], however, none to our knowledge has been applied to biomedical article classification nor do they employ T-cell cross-regulation dynamics.

We have already proposed an agent-based model of T-cell cross-regulation for spam detection [2; 3]. Our distributed model extendes the original analytical model of T-Cell cross-regulation dynamics [5] to be able to deal with many multiple features simultaneously, and therefore render the model applicable to real-world applications. Our results on spam-detection were comparable to state-of-art text classifiers [2; 3]. However, our initial agent-based implementation of cross-regulation dynamics did not explore important parameter configurations such as the death rate of T-cells or the best training strategies. It also lacked an extensive parameter search for optimized performance. More recently, we started addressing some of these issues on full-text biomedical data from BioCreative, and showed that T-cell death is important to obtain better classification [4]. This is an interesting result, showing that the loss of T-cells rather than hindering, can improve the collective classification of relevant documents. Therefore, the dynamics of T-cell cross-regulation as proposed by Carneiro et al. [5] can lead to the elimination of T-cells that are not useful for classification—even in our extended formulation which contains hundreds of distinct T-cells representing antigens or textual features. We also showed that training exclusively on relevant documents (or self antigens) leads to worse classification performance than training on both relevant and irrelevant documents [4]. This is interesting for tuning the algorithm in text mining settings, but also suggests that T-cell cross-regulation in the vertebrate adaptive immune system can improve from a "training" stage where it is presented with both self and nonself antigens.

Here, we study the importance of the original temporal sequence of bio-medical articles. Text mining classifiers do not typically depend on the sequence of documents they are trained with, but our model of T-cell cross-regulation dynamics does. Therefore, we are interested in ascertaining if the sequence-dependence of ensuing collective dynamics can be used to track the natural change in real-world textual corpora, i.e. concept drift [31]. We also study the effect of biases in the initial T-cell population. This more extensive study allows us to better understand the behavior of T-cell cross-regulation dynamics and establish its capability to classify sequential data. It also leads to a competitive, novel bio-inspired text classification algorithm.

The Immune System as Inspiration

The vertebrate adaptive immune system[2] (IS) is a complex network of cells that distinguishes between self and nonself substances or antigens—usually fragments of proteins that can be recognized by the immune system. When nonself antigens are discovered, an immune response to eliminate them is set in motion. Recognizing self antigens, which obviously should not lead to an (auto)immune response to eliminate them, is resolved by negative selection of T-cells which takes place in the thymus, and removes T-Cells that strongly bind to self antigens—after positive selection of T-Cells that are capable of binding with the major histocompatibility complex (MHC). It is in the thymus that T-cells develop and mature; only T-cells that have failed to bind to self antigens are released (as naive T-cells), while the rest of the T-cells is culled. Mature T-cells are allowed out of the thymus to detect nonself antigens. They do this by binding to

vs. harmful substances, because harmless can also include antigens from bacteria that are necessary for vertebrate bodies, and harmful can also include body's own tumor cells.

[2] A good, though already a bit dated, overview of the vertebrate immune system for the artificial life community is Hofmeyer's [12].

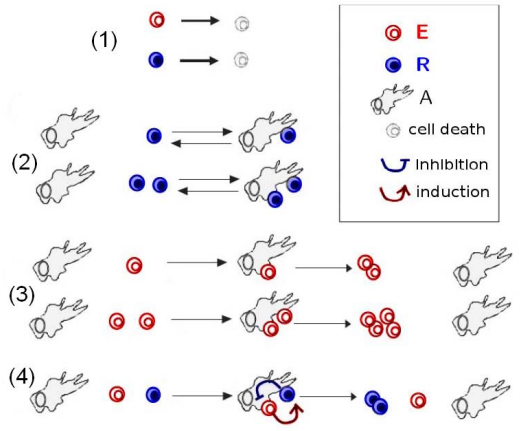


Figure 1: CRM interactions that define the dynamics of APC and E and R T-cells. The model assumes that APC can only form conjugates with a maximum of two T-cells. Adapted from [5].

antigen presenting cells (typically B-cells, macrophages and dendritic cells) that collect and present antigens via MHC after breaking them by lysosome. The specific T-cells that are able to bind to the presented antigens then stimulate B-cells that start a cascade of events leading to antibody production and the destruction of the pathogens or tumors linked to the antigens. However, it is possible that T-cells and B-cells, which are also trained in the thymus and bone marrow, mature before being exposed to all self antigens. Even more problematic is the somatic hypermutation that ensues in lymph nodes after the activation of B-cells. At this stage, it is possible to generate many mutated B-Cell clones that could bind to self antigens. Either situation can cause autoimmunity by generating T-cells capable of attacking self antigens. One way to deal with this problem is by a process called costimulation which involves the co-verification of self antigens by both T-cells and B-cells before the antigen is identified as associated with a nonself pathogen to be attacked. To further insure that the T-cells do not attack self, another type of T-cells known as regulatory T-cells, are formed in the thymus where they mature to avoid recognizing self antigens. These regulatory T-cells have the responsibility of preventing autoimmunity by down-regulating other T-cells that might bind and kill self antigens. Our model is based on this process of T-Cell cross-regulation.

Artificial Immune Systems (AIS) are artificial life tools, inspired by theories and components of the immune system, and applied towards solving computational problems, such as categorization, optimization and decision making [7]. Common AIS techniques are based on specific theoretical models explaining the behavior of the IS such as: Negative Selection, Clonal Selection, Immune Networks and Dendritic Cells [30]. AIS fall in categories: (1) mathematical and computational models to understand IS behavior and (2) engineering of adaptive machine learning algorithms. While our approach fits more immediately in the second category, our goal is also to use our classifier to test the prevailing model of T-cell cross-regulation and therefore also contribute to the first category of the study of AIS.

The Cross-Regulation Model

The *T-cell Cross-Regulation Model* (CRM) [5] is a dynamical system that aims to distinguish between self and nonself protein fragments (antigens) using only four possible interaction rules amongst three cell-types: *Effector T-cells* (E), *Regulatory T-cells* (R) and *Antigen Presenting Cells* (APC). As their name suggests, APC present antigens for the other two cell-types, E and R, to recognize and bind to them. Effector T-cells (E) proliferate upon binding to APC, unless adjacent to regulatory T-cells (R), which regulate E by inhibiting their proliferation. For simplicity, proliferation of cells is limited to duplication in quantity in contrast to having a proliferation rate. T-cells that do not bind to APC die off with a certain death rate. The dynamics of the CRM depend on four interaction rules defined by the following reactions (illustrated in Fig. 1):

$$E \xrightarrow{d_E} \{\} \text{ and } R \xrightarrow{d_R} \{\} \quad (1)$$

$$A + R \rightarrow A + R \quad (2)$$

$$A + E \rightarrow A + 2E \quad (3)$$

$$A + E + R \rightarrow A + E + 2R \quad (4)$$

Reaction (1) defines E and R apoptosis with the corresponding death rates d_E and d_R. The last three proliferation reactions define the maintenance of R (2), the duplication of E (3), and the maintenance of E and duplication of R (4) .

Carneiro et al [5] developed the analytical CRM to study the dynamics of a population of T-cells and APC that present a single antigen associated with a single T-cell population. In [2; 3], we extended the original CRM model to be able to deal with multiple populations of antigens and T-Cells using agent-based modeling. More recently, Sepulveda [26, pp 111-113] extended the original CRM to study analytically multiple populations of T-cells that recognize antigens presented by APC capable of presenting at most two distinct antigens. In our model, explained in detail in the next section, APC are capable of presenting hundreds of antigens to be recognized by T-cells of hundreds of different populations, using the same four interaction rules of the CRM.

The Agent-Based Cross-Regulation Model

In order to adapt CRM to an *Agent-Based Cross-Regulation Model* (ABCRM) for text classification, one has to think of documents as analogous to the organic substances that upon entering the body are broken into constituent pieces. These pieces, known as epitopes, are presented on the surface of Antigen Presenting Cells (APC) as antigens. In the ABCRM, antigens are textual features (e.g. words, bigrams,

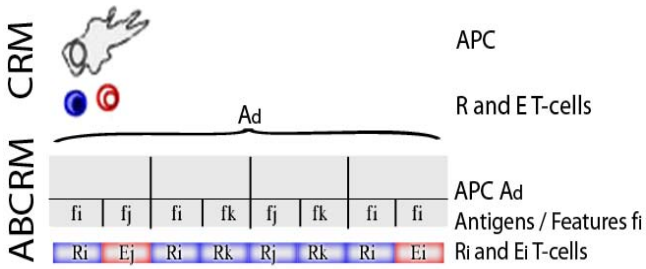


Figure 2: To illustrate the difference between the CRM and the ABCRM, the top part of the figure represents a single APC of the CRM which can bind to a maximum of two T-Cells. The lower part represents the APC for a document d in the ABCRM, which contains many pairs of antigen/feature "slots" where pairs of T-cells can bind. In this example, the first pair of slots of the APC A_d presents the features f_i and f_j; a regulatory T-cell R_i and an effector T-cell E_j bind to these slots, which will therefore interact according to reaction (4)—R_i inhibits E_j and in turn proliferates by doubling. The next pair of slots leads to the interaction of regulatory T-cells R_i,R_k that proliferate via reaction (2), etc.

titles, numbers) extracted from articles and presented by artificial APC such that they can be recognized by a number of artificial Effector T-cells (E) and artificial Regulatory T-cells (R). Individual E and R have receptors for a single, specific (textual) feature: they are *monospecific*. E proliferate upon binding to antigens presented by APC unless suppressed by R; R suppress E when binding in adjacent locations on APC. Individual APC present various document features: they are *polyspecific*. Each APC is produced when documents enter the artificial cellular dynamics, by breaking the former into constituent textual features. Therefore we can say that APC are representative of specific documents whereas E and R are representative of specific features.

A document d contains a set of features F_d; an artificial APC A_d that represents d, presents antigens/features $f_i \in F_d$ to artificial E and R T-cells. E_i and R_i bind to a specific feature f_i on any APC that contains it; if $f_i \in F_d$, then either E_i or R_i may bind to A_d as illustrated in figure 2. In biology, antigen recognition is a more complex process than mere polypeptide sequence matching, but for simplicity we limit our feature recognition to string matching. Once T-cells bind to an APC A_d, every pair of adjacent T-cells on A_d proliferates according to reaction rules (2-4). APC are organized as a sequence of pairs of "slots" of (textual) features, where T-cells, specific for those features, can bind. We use this antigen/feature presentation scheme of pairs of "slots" to simplify our algorithm. In future work we will study alternative feature presentation scenarios. In summary, each T-cell population is specific to and can bind to only one feature presented by any APC. Implementing the algorithm as an Agent-based model (ABM) allows us to deal with the recognition and co-recognition (co-occurrence in the same document/APC) of many features simultaneously, rather than a single one as the original CRM does.

The ABCRM uses incremental learning to first train on N labeled documents (relevant and irrelevant), which are ordered sequentially (typically by time signature) and then test on M unlabeled documents that follow in time order. The sequence in which documents are received affects the artificial cellular dynamics, as incoming APC and T-cells face a T-cell dynamics that depends on the specific documents previously encountered. Therefore, we use publication-time as the default ordering for incoming documents, but we study here if there is an advantage to preserving the original temporal sequence of articles (see below).

Carneiro et al [5] show that both E and R T-cells co-exist in healthy individuals assuming enough APC exist. R T-cells require adequate amounts of E T-cells to proliferate, but not too many that can out-compete R for the specific features presented by APC. "Healthy" T-cell dynamics is identified by observing the co-existence of both E and R features with $R \geq E$. "Unhealthy" T-cell dynamics is identified by observing $E \gg R$, and should result when encountering many irrelevant features in a document—in analogy with encountering many nonself antigens. In other words, features associated with relevant documents should have E and R T-cell representatives in comparable numbers in the artificial cellular dynamics (with slightly more R). In contrast, features associated with irrelevant documents should have many more E than R T-cells. Therefore, when a document d contains features F_d that bind mostly to E rather than R cells, we can classify it as irrelevant—and relevant in the opposite situation.

The ABCRM is controlled by 6 parameters:

- E_0 is the initial number of Effector T-cells generated for all new features
- R_0^- is the initial number of Regulatory T-cells generated for all new features in irrelevant and unlabeled (testing) documents
- R_0^+ is the initial number of Regulatory T-cells generated for all new features in relevant documents
- d_E is the death rate for Effector T-cells that do not bind to APC
- d_R is the death rate for Regulatory T-cells that do not bind to APC
- n_A is the number of slots in which each feature f_i is presented on APC

In the IS, millions of novel T-cells are randomly generated in the thymus every day to attempt to predict future antigens. In our algorithm, in contrast, we generate T-cells only for features (words) occurring in the relevant document corpus. This is reasonable because the space of meaningful words in a language are largely fixed and much smaller than the space of possible polypeptide epitopes in biology. When (textual) features are encountered for the first time, a fixed initial number of E_0 effector T-Cells and R_0 regulatory T-Cells is generated for every new feature f_i. These initial values of T-cells vary for relevant and irrelevant documents

in training and in testing stages. More Regulatory (R_0^+) than Effector T-cells are generated for features that occur for the first time in documents that are labeled relevant in the training stage ($R_0^+ > E_0$), while fewer Regulatory (R_0^-) than Effector T-cells are generated in the case of irrelevant documents ($R_0^- < E_0$). Features appearing in unlabeled documents for the first time during the testing stage are treated as features from irrelevant documents, assuming that new features are irrelevant (nonself) until neutralized by the collective dynamics given their co-occurrence with relevant ones.

Naturally, relevant features will occur in irrelevant documents and vice versa. However, the assumption is that relevant features tend to co-occur more frequently with other relevant features in relevant documents and similarly for irrelevant features. Therefore, the proliferation dynamics defined by the 4 reactions and guided by co-binding to APC slots is expected to correct the erroneous initial bias. But this self-correction has not been proven, and it is one of the issues we test in the present work, as detailed below. The pseudocode for the algorithm is shown below:

ABCRM:
(1) $\forall d$ generate a linear array A_d presenting each $f_i \in F_d$ at n_A arbitrary, randomly distributed slots
(2) Let C_t contain E_k and R_k T-cells for all features f_k in the cellular dynamics at time t.
(3) For an incoming document d, $\forall f_i \in F_d$, if $E_i \notin C_t$ and $R_i \notin C_t$ then,
(4) $E_i = E_0$ (generate E_0 Effector T-cells for f_i)
(5) if d is labeled relevant.
(6) $R_i = R_0^+$ (generate R_0^+ Regulatory T-cells for f_i)
(7) otherwise
(8) $R_i = R_0^-$ (generate R_0^- Regulatory T-cells for f_i)
(9) update C_t with E_i and Ri
(10) Let all E_i, R_i *bind specifically*[3] to matching f_i on A_d:
(11) $\forall$ pairs of adjacent (f_i, f_j) on A_d apply the interaction rules: $(R_i, R_j) \rightarrow R_i + R_j$ $(E_i, E_j) \rightarrow 2.E_i + 2.E_j$ $(E_i, R_j) \rightarrow E_i + 2.R_j$
(12) $\forall R_i$, E_i that bind to A_d, update total number of E_i, R_i
(13) $\forall R_k, E_k \in C_t$ that do not bind to A_d, cull E_k and R_k according to death rates d_E and d_R
(14) If d is unlabeled, Let $R(d) = \sum_{f_i \in F_d}(R_i)$ and $E(d) = \sum_{f_i \in F_d}(E_i)$
(15) Compute the normalized score $S(d) = \frac{R(d)-E(d)}{\sqrt{R^2(d)+E^2(d)}}$
(16) If $S(d) > 0$ then classify d as relevant, else irrelevant.

Data and Feature Selection

The BioCreative (BC) challenge aims to assess the quality of biomedical literature mining algorithms such as article classifiers. The article classification task of Biocreative 2.5 [17] was based on a training data set (T) comprised of 61 full-text articles relevant (P_T) to the topic of *protein-protein interaction* (PPI) and 558 irrelevant ones (N_T). The realistic imbalance between the relevant and irrelevant instances is very challenging for common machine learning techniques, since there are few instances of the topical category of interest to generalize from. Because we cannot predict how imbalanced the validation set will be, we first search for optimal ABCRM parameters on a smaller sample of the training that is balanced in the numbers of relevant and irrelevant documents. For this purpose, we chose the first 60 relevant and sampled 60 irrelevant articles that were published around the same date (uniform distribution between Jan and Dec 2008) as illustrated in figure 3. For final validation we used the entire Biocreative 2.5 testing data set (V) consisting of 63 full-text articles relevant to PPI (P_V) and 532 irrelevant ones (N_V) as also shown in figure 3. Furthermore, we compared our optimized algorithm with a Naive Bayes (NB) [19] and a support vector machine (SVM) classifier [15].

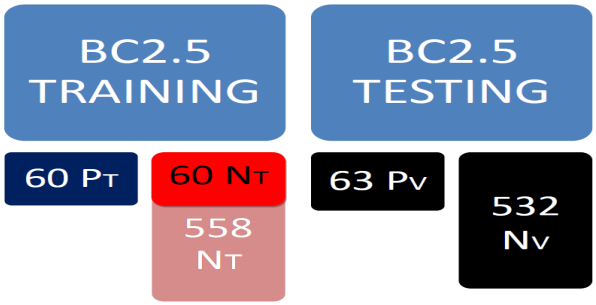


Figure 3: Numbers of relevant (P) and irrelevant (N) documents in the training (T) and testing (V) data sets of the Biocreative 2.5 challenge. In the parameter search stage, we use a balanced set of 60 P_T (blue) and 60 N_T (red) randomly selected articles from the training data set. In the testing stage we use the unbalanced validation set containing 63 P_V (black) and 532 N_V (black) documents. Notice that the validation data was provided to the participants in the classification task of Biocreative 2.5 unlabeled, therefore participants had no prior knowledge of class proportions.

We pre-processed all articles by filtering out common words[4] and porter stemming [22] the remaining words which are all the potential features. We then ranked words/features f extracted from training articles (T)[5] according to two scores: the first one is the average TF.IDF[6] [8], and the second one is the separation score $S(f) = |p_P(f) - p_N(f)|$ where p_P (p_N) is the probability of a feature occurring in a relevant (irrelevant) document of the training set T [1; 16]. The final rank $R(f_i)$ for every feature f_i is given by the product of the ranks obtained from both scores; we used only the 650 top ranked features according

[3]While the features f_i are arbitrarily distributed over the APC A_d, E_i and R_i that are specific to f_i, are sampled from C_t based on the proportions of E_i to R_i

[4]The list of common (stop) words includes 33 of the most common English words from which we manually excluded the word "with", as we know it to be of importance to PPI

[5]For feature extraction we used both the training data of Biocreative 2.5 and Biocreative 2 as described in [16]; all classifiers used the exact same feature set.

[6]TF.IDF is a common text weighting measure to evaluate the importance of a feature/word in a document in a corpus. TF stands for term frequency in a document and IDF for inverse document frequency in the corpus. [8]

Parameter	Range	Step
E_0	[1,7]	1
R_0^-	[3,12]	1
R_0^+	[3,12]	1
d_E	[0.0,0.4]	0.1
d_R	[0.0,0.4]	0.1
n_A	[2,22]	2

Table 1: Parameter ranges used for optimizing the ABCRM

to $R(f_i)$. Features such as "interact", "lysat" and "transfect" were ranked above others for their high ranks according to both scores. See [16] for more details about the feature extraction procedure.

Parameter Search and Robustness

We performed an exhaustive parameter search by training the ABCRM on 60 balanced full-text articles (30 P_T and 30 N_T from BC2.5 training) and testing it on the remaining 60 balanced ones (also 30 P_T and 30 N_T from BC2.5 Training) as illustrated in figure 3[7]. Each run corresponds to a unique configuration of the 6 parameters of the ABCRM. The explored parameter ranges are listed in table 1 which result in a total of 192500 unique parameter configurations for each experiment. Finally, the parameter configurations were sorted with respect to the resulting F-score measure of performance[8], which is a good measure between precision and recall when applied to balanced data [29].

We compiled the performance of the ABCRM on the entire parameter search space for two distinct experiments: (1) effect of **sequence order** of articles, and (2) effect of varying **initial T-cell counts**. In another publication [4] we showed that a positive T-Cell death ratio improves classification, whereas training exclusively on relevant documents lowers the performance. In both experiments, we choose the 50 configurations with highest F-score measure to study the ABCRM performance, because we are interested in identifying the experimental setups that lead to higher **robustness** to parameter changes. We compare experimental outcomes with the paired student t-test; the null hypothesis is that the two samples are drawn from the same distribution. A p-value < 0.01 rejects the null hypothesis, establishing a statistical distinction between the data drawn from two experimental setups—in our case, the data from each experiment are the top 50 F-score values obtained. Finally, we train on both relevant and irrelevant documents as this was shown to be advantageous [4], and search for optimal parameter configurations (including T-Cell death ratios).

The **first** experiment aims to establish how much the sequence order of processing documents impacts performance. In particular, we test if preserving the original temporal order of biomedical documents results in better performance, as this would indicate that the ABCRM can use its sequence-dependent dynamics to track the natural concept or topical drift and thus improve classification. Therefore, we compared the performance of the ABCRM when tested on a sequence of biomedical articles ordered by the original publication, against randomly shuffling the articles. We tested four distinct experimental setups in order to fully explore the influence of document order:

1. Ordered training set ⇒ ordered testing set
2. Ordered training set ⇒ shuffled testing set
3. Shuffled training set ⇒ shuffled testing set
4. Shuffled training set ⇒ ordered testing set

In the case of shuffled sets, we produced 8 runs with distinct random document orderings; in those cases, performance is represented by central tendency and variation.

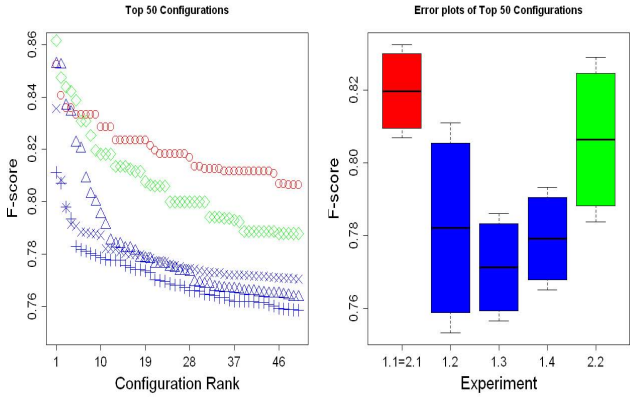


Figure 4: Left: top 50 parameter configurations ranked in terms of F-score for experimental setups 1.1/2.1 (red circles), 1.2 (blue triangles), 1.3 (blue pluses), 1.4 (blue crosses), and 2.2 (green diamonds). Right: mean (line), 95%CI (boxes), and standard deviation (whiskers) of F-scores for top 50 parameter configurations.

The results of this experiment are summarized in figure 4. The robustness of performance of the first experimental setup (preserving temporal order of articles) is significantly above the other setups. Using the paired student t-test as described above, we conclude that the ABCRM is sensitive to article order—i.e. if the articles are shuffled, the performance is worse. While the performance of the best classifier obtained via experimental setup 1.2 is equivalent to the best one obtained for experimental setup 1.2 (F-Score = 0.853, see table 2 and figure 4), that setup is very sensitive to parameter changes and the performance quickly and significantly decreases for subsequent best classifiers (see figure

[7]Notice that this parameter search on the provided labeled training data uses only the information available to the teams participating in Biocreative 2.5 challenge, and none of the testing data whose labels were revealed post-challenge.

[8]F-score = $\frac{2.Precision.Recall}{Precision+Recall}$ where Precision = $\frac{TP}{TP+FP}$ and Recall = $\frac{TP}{TP+FN}$. True Positives (TP) and False Positives (FP) are the classifier's correct and incorrect predictions for relevant documents, while True Negatives (TN) and False Negatives (FN) are the correct and incorrect predictions for irrelevant documents.

Exp.	F-Score	E_0	R_0^+	R_0^-	d_R	d_E	n_A
$1.1 = 2.1$	0.852	2	11	10	0.3	0.2	18
1.2	0.853	2	7	6	0.0	0.0	20
2.2	0.862	3	8	7	0.2	0.1	14

Table 2: Performance and parameters of top classifiers in experiments 1.1, 1,2, 2.1 and 2.2.

4). Indeed, the performance of the top 50 classifiers for experimental setups 1.2, 1.3, and 1.4 is statistically indistinguishable from each other, but is significantly lower than the performance of the top 50 classifiers for experimental setup 1.1. This means that there is indeed a conceptual drift in the Biocreative 2.5 article data stream, and the ABCRM can track it better (and in a more robust manner) when publication date is used as the sequence for processing articles than when the temporal order of articles is shuffled. This also suggests that the process of T-Cell cross-regulation in the IS, as modeled here, can track changing environments.

In the **second** experiment we test the effect of the initial biases introduced when features are first encountered. The initial biases of regulatory T-cells injected in the dynamics for a new feature f_i, depend on whether the first document d where the feature is encountered is labeled irrelevant/unknown (R_0^-) or relevant (R_0^+). Since features will occur in both relevant and irrelevant articles, this initial bias for a feature could be detrimental, as a feature most associated with one class could be first encountered on a document of the opposite class. Therefore, it is important to test if the dynamics of the four reactions and APC feature co-presentation that define the ABCRM can self-correct such erroneous biases. To perform this test, we altered the ABCRM algorithm such that T-cells are incremented appropriately every time a feature occurs in a document, and not just the first time the feature occurs (as the canonical algorithm does). Specifically, every time a feature f_i occurs in a document d, we increment $E_i = E_i + E_0$ and $R_i = R_i + R_0^+$ if d is labeled relevant and $R_i = R_i + R_0^-$ if d is labeled irrelevant or unknown.

The results of this experiment are also summarized in figure 4. The performance of top classifiers obtained for experimental setups 2.1 (same as 1.1) and 2.2 is shown in table 2. While the best overall classifier is obtained with experimental setup 2.2, the performance of both setups is statistically indistinguishable. Indeed, using the paired student t-test as described above, we conclude that this modification does not improve the performance of the ABCRM on the Biocreative data set, thus showing that the initial bias can be corrected by the ABCRM collective dynamics. Because features most associated with a given class tend to co-occur in text with other features most associated with the same class, they will also tend to be co-presented in APC and thus the relevant T-cells will proliferate with similar rates. Therefore, the dynamics of the ABCRM can self-correct initial erroneous biases from the natural textual co-occurrence of features. This shows that T-Cell cross-regulation as modeled here can self-correct initial antigen misclassification by the IS, assuming that antigens from one class (self/nonself) tend to co-occur with antigens from the same class.

Validation and Conclusions

To test the ABCRM on the full, unbalanced testing set of the Biocreative challenge (see figure 3), thus establishing its merit as a bio-inspired biomedical literature mining classifier, we adopted the best parameter configuration from the canonical ABCRM (experimental setup 1.1 and 2.1, see table 2) obtained from the parameter search described above. We compared the ABCRM classifier with the multinomial Naive Bayes (NB) with boolean attributes [19], and the publicly available SVM^{light} implementation of SVM applied to normalized feature counts [15]. All classifiers were tested on the same features obtained from the same data.

	ABCRM	NB	SVM	Mean	StDev.	Median
Precision	0.22	0.14	0.24	0.38		
Recall	0.65	0.71	0.94	0.68		
F-score	0.33	0.24	0.36	0.39	0.14	0.38
Accuracy	0.71	0.52	0.74	0.67	0.30	0.84
AUC	0.34	0.19	0.46	0.43	0.17	0.44
MCC	0.24	0.13	0.31	0.31	0.19	0.33

Table 3: F-Score, Accuracy, AUC and MCC performance of various classifiers when training on the balanced training set of articles and testing on the full unbalanced Biocreative 2.5 testing set. Also shown is the central tendency and variation of all systems submitted to Biocreative 2.5.

Since the F-score and Accuracy are not very reliable for evaluating unbalanced classification [29], we also use the Area Under the interpolated precision and recall Curve (AUC) and Matthew's Correlation Coefficient (MCC). The results are listed in table 3, which also includes the central tendency of the results of all systems submitted by all Biocreative 2.5 participating teams [17; 16]. It should be noted that the ABCRM, NB, and SVM classifiers we tested here, used only single-word features because we wish to establish the feasibility of the method. In contrast, most classifiers submitted to the Biocreative 2.5 challenge (including another method from our group which was one of the top-performing classifiers [16]) used more sophisticated features such as bigrams and problem-specific entities. Therefore, it is not surprising that these methods as tested here performed under the mean of the challenge. Our goal was to establish the ABCRM as a new bio-inspired text classifier to be further improved in the future with more sophisticated features. When we compare its performance to NB and SVM on the exact same single-word features, the results are encouraging. Indeed, based on the given measures, while SVM out-performed the ABCRM, the latter out-performed NB. Therefore, the dynamics T-Cell cross-regulation lead to a competitive collective classification of biomedical articles, which we intend to develop further.

In conclusion, we observed that our algorithm adapts to the initial bias of T-cell populations generated for new fea-

tures, and it performs best when tested on a sequence of articles ordered by publication date—showing that it can track concept drift in the biomedical literature. These properties of our Artificial Life model also show that T-Cell cross regulation is capable of efficient collective classification of nonself antigens and suggest that T-Cell cross-regulation can naturally respond to drift in the pathogen population. Therefore T-Cell cross-regulation defined by the 4 reaction rules and co-presentation of features in APC can be seen as an effective general principle of collective classification available to populations of cells. Clearly, there is still much to do to improve the model. For biomedical literature mining applications, we need to test it with more sophisticated features (as top classifiers in the field do). For our goal of understanding T-Cell cross-regulation in the IS, we need to understand better how memory is sustained in the collective cellular dynamics; for instance, how to sustain regulatory T-Cells, which keep memory of self, in the dynamics even in the presence of very unbalanced scenarios where there are many more nonself instances.

References

[1]Alaa Abi-Haidar, Jasleen Kaur, Ana Maguitman, Predrag Radivojac, Andreas Retchsteiner, Karin Verspoor, Zhiping Wang, and Luis M. Rocha. Uncovering protein interaction in abstracts and text using a novel linear model and word proximity networks. *Genome Biology*, page 9(Suppl 2):S11, 2008.

[2]Alaa Abi-Haidar and Luis M. Rocha. Adaptive spam detection inspired by a cross-regulation model of immune dynamics: A study of concept drift. In *Artificial Immune Systems (Proc. ICARIS)*, volume 5132 of *LNCS*, pages 36–47, 2008.

[3]Alaa Abi-Haidar and Luis M. Rocha. Adaptive spam detection inspired by the immune system. In S. Bullock, J. Noble, R. A. Watson, and M. A. Bedau, editors, *Artificial Life XI: 11th Int. Conf. on the Simulation and Synthesis of Living Systems*. MIT Press, 2008.

[4]Alaa Abi-Haidar and Luis M. Rocha. Biomedical article classification using an agent-based model of t-cell cross-regulation. In *ICARIS 2010: Proc. of the 8th Int. Conf. on Artificial Immune Systems*, LNCS, page In Press., 2010.

[5]J. Carneiro, K. Leon, I. Caramalho, C. van den Dool, R. Gardner, V. Oliveira, M.L. Bergman, N. Sepúlveda, T. Paixão, J. Faro, and J. Demengeot. When three is not a crowd: a crossregulation model of the dynamics and repertoire selection of regulatory cd4 t cells. *Immunological Reviews*, 216(1):48–68, 2007.

[6]James Crutchfield and Melanie Mitchell. The evolution of emergent computation. *PNAS*, 92(23), 1995.

[7]L.N. De Castro and J. Timmis. *Artificial immune systems: a new computational intelligence approach*. Springer Verlag, 2002.

[8]R. Feldman and J. Sanger. *The Text Mining Handbook: advanced approaches in analyzing unstructured data*. Cambridge University Press, 2006.

[9]Tomás Helikar, John Konvalina, Jack Heidel, and Jim A Rogers. Emergent decision-making in biological signal transduction networks. *Proc Natl Acad Sci U S A*, 105(6):1913–1918, Feb 2008.

[10]William Hersh, Ravi Teja Bhupatiraju, and Sarah Corley. Enhancing access to the bibliome: the trec genomics track. *Medinfo*, 11(Pt 2):773–777, 2004.

[11]Lynette Hirschman, Alexander Yeh, Christian Blaschke, and Alfonso Valencia. Overview of biocreative: critical assessment of information extraction for biology. *BMC Bioinformatics*, 6 Suppl 1:S1, 2005.

[12]S.A. Hofmeyr. An Interpretative Introduction to the Immune System. *Design Principles for the Immune System and Other Distributed Autonomous Systems*, 2001.

[13]L. Hunter and K.B. Cohen. Biomedical language processing: What's beyond pubmed? *Molecular Cell*, 21(5):589–594, 2006.

[14]L. Jensen, J. Saric, and P. Bork. Literature mining for the biologist: from information retrieval to biological discovery. *Nat Rev Genet*, 7(2):119–129, Feb 2006.

[15]T. Joachims. *Learning to classify text using support vector machines: methods, theory, and algorithms*. Kluwer Academic Publishers, 2002.

[16]A. Kolchinsky, A. Abi-Haidar, J. Kaur, A. Hamed, and L. M. Rocha. Classification of protein-protein interaction documents using text and citation network features. *IEEE/ACM Transactions on Computational Biology and Bioinformatics.*, page In Press, 2010.

[17]M Krallinger. The biocreative ii. 5 challenge overview. In *Proc. the BioCreative II.5 Workshop 2009 on Digital Annotations*, page 19, 2009.

[18]Martin Krallinger and Alfonso Valencia. Evaluating the detection and ranking of protein interaction relevant articles: the biocreative challenge interaction article sub-task (ias). In *Proc. 2nd Biocreative Challenge Evaluation Workshop*, 2007.

[19]V. Metsis, I. Androutsopoulos, and G. Paliouras. Spam Filtering with Naive Bayes–Which Naive Bayes? *Third Conf. on Email and Anti-Spam (CEAS)*, 2006.

[20]Melanie Mitchell. Complex systems: Network thinking. *Artificial Intelligence*, 170(18):1194–1212, 2006.

[21]David Peak, Jevin D. West, Susanna M. Messinger, and Keith A. Mott. Evidence for complex, collective dynamics and distributed emergent computation in plants. *PNAS*, 101(4):918–922, 2004.

[22]MF Porter. An algorithm for suffix stripping. *Program*, 13(3):130–137, 1980.

[23]Stephen C. Pratt. Quorum sensing by encounter rates in the ant temnothorax albipennis. *Behav. Ecol.*, 16(2):488–496, 2005.

[24]L.M. Rocha and W. Hordijk. Material representations: From the genetic code to the evolution of cellular automata. *Artificial Life*, 11(1-2):189–214, 2005.

[25]L.A. Segel and I. Cohen. *Design Principles for the Immune System and Other Distributed Autonomous Systems*. Oxford University Press, 2001.

[26]Nuno H. Sepulveda. *How is the T-cell repertoire shaped*. PhD thesis, Instituto Gulbenkian de Ciencia, 2009.

[27]Cosma Shalizi, Rob Haslinger, Jean-Baptiste Rouquier, Kristina Klinkner, and Cristopher Moore. Automatic filters for the detection of coherent structure in spatiotemporal systems. *Phys.Rev.E*, 73, 2006.

[28]Hagit Shatkay and Ronen Feldman. Mining the biomedical literature in the genomic era: An overview. *Journal of Computational Biology*, 10(6):821–856, 2003.

[29]M. Sokolova, N. Japkowicz, and S. Szpakowicz. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. *AI 2006: Advances in Artificial Intelligence*, pages 1015–1021, 2006.

[30]J. Timmis. Artificial immune systems today and tomorrow. *Natural Computing*, 6(1):1–18, 2007.

[31]Alexey Tsymbal. The problem of concept drift: definitions and related work. *Computer Science Department Trinity College Dublin*, 4(C):200415, 2004.

[32]J. Twycross and S. Cayzer. An immune system approach to document classification. *Master's thesis, COGS, University of Sussex, UK*, 2002.

[33]Matthew Walters and Vanessa Sperandio. Quorum sensing in escherichia coli and salmonella. *Int. Journal of Medical Microbiology*, 296(2-3):125 – 131, 2006.

Robotics

Evolving Amorphous Robots

Jonathan D. Hiller[1], Hod Lipson[1]

[1]Computational Synthesis Lab
Mechanical and Aerospace Engineering
Cornell University, Ithaca, NY, 14853
Hod.Lipson@cornell.edu

Abstract

Research in evolutionary robotics has traditionally been limited to morphologies comprising rigid and discrete components, such as links connected with rotational or linear joints and actuators. Here, we demonstrate the evolution of robots with continuous and amorphous morphologies composed of multiple materials. Actuation is accomplished by periodic volumetric expansion and contraction of one or more of these materials. The challenges of representing evolvable multi-material freeform shapes and evaluation (simulation) of the resulting soft bodies are discussed. Several genotypic representations are explored which use a level-set threshold to generate the material distribution in the phenotype. Soft body simulation of the robot is accomplished using a relaxation algorithm to model the dynamics of the resulting amorphous machines under the actuation material expansion, gravity forces, and non-linear ground friction. These results open the door to a new design space that more closely mimics the freeform, amorphous and continuous nature of biological systems.

Introduction

The field of evolutionary robotics has explored methods for generating interesting and functional robot morphologies (Nolfi and Floreano 2002). Ever since the early work in evolving virtual (Sims 1994) and physical (Lipson and Pollack 2000) creatures, many examples have been published of evolved walking, (Pollack, Lipson et al. 2001) running (Zykov, Bongard et al. 2004; Hornby, Takamura et al. 2005), and swimming (Ijspeert and Kodjabachian 1999) robots. These simulations all use rigid-body simulations of discrete components connected by rotational or linear joints. Many interesting biological forms of locomotion, however, are not modeled well by rigid links and joints – such as the earthworm (Quillin 1999) and the amoeba (Mast 1926). More recent work on morphogenetic robotics explores the development of more complex morphologies using many rigid links, but these bodies are still inherently discrete and relatively sparsely connected (Hornby, Lipson et al. 2001; Bongard 2003).

In this paper we focus on evolving fully amorphous soft robots. Material distributions take the place of discrete links, and volume changing materials replace discrete actuators. These material distributions are not constrained to any given topology or shape. This freedom removes fundamental constraints, thereby opening a vast new design space to explore.

Traditional computer aided design (CAD) tools are typically inappropriate for designing amorphous machines with continuous morphology and actuation. Such tools rely on feature-based modeling approaches that work well for well-defined geometric primitives made of a single material. However, the lack of constraints on the shape and material distribution of soft robots indicate that existing CAD programs would be ineffective in their ability to fully take advantage of the design space offered. Therefore, new higher level design tools are necessary to meet functional goals without geometric constraints.

As greater computing power becomes more readily available, design automation algorithms are becoming increasingly valuable for designing structures with freeform material distributions. Homogenization techniques (Bendsoe and Kikuchi 1988) are useful for designing single material structures such as 2D and 3D beams, and simple mechanisms(Nishiwaki, Frecker et al. 1998). However, homogenization techniques are limited in their ability to meet high level functional goals, such as specific beam deformations (Hiller and Lipson 2009) or locomotion. Here, we focus on the use of evolutionary algorithms to autonomously design locomoting amorphous soft robots.

We first briefly describe the field of soft robotics and the additive manufacturing technology that make amorphous robots possible. Next, we explore three representations that enable genetic algorithms to evolve functional three dimensional multi-material morphologies independent of topology. We then describe our soft body physics simulator used to evaluate potential solutions. Finally, the abilities of each representation are compared and several resulting amorphous, locomoting robots are shown for various scenarios.

Background

Soft Robotics

Robots are traditionally made of discrete rotary or linear actuators, connected by rigid links. This architecture is driven

by the manufacturing technology used to physically build the robots and the methods used to simulate and control them. The kinematics of these machines can be deterministically modeled and used to perform useful functions such as path planning and collision avoidance.

A new paradigm in robotics has recently emerged, inspired largely by the robustness and resilience of biological systems. These "soft" robots trade deterministic control for probabilistic models, but gain robustness (Rieffel, Trimmer et al. 2008). While several actuation methods have been explored for soft robots such as shape memory alloy (SMA), pneumatics, (Rieffel, Saunders et al. 2009) electroactive polymers, (Bar-Cohen 2001) and jamming (Mozeika, Steltz et al. 2009), these all place constraints on the geometry and the ways in which internal forces are applied. Here we consider pure volumetrically actuated materials in order to more accurately mimic distributed actuation and avoid imposing undue constraints.

Freeform Fabrication

The new design space of soft robots is characterized by a nearly complete freedom over the spatial distribution of materials (Beaman, Marcus et al. 1997). Physically, this is realized by novel additive manufacturing technologies (also known as solid freeform fabrication, rapid prototyping or 3D printing). This technology is currently capable of autonomously fabricating multi-material 3D objects in any desired shape, with any internal material distribution (Malone, Rasa et al. 2004; Malone and Lipson 2007; Hiller and Lipson 2009; Objet 2010). Materials that can be co-fabricated include rigid plastics and soft rubbers.

A significant missing link in soft robots becoming ubiquitous is the ability to print volumetric actuators. Many examples are present in literature of additively manufactured robots with actuators added after fabrication. However, these are limited to traditional rotational or linear actuators, (Pollack, Lipson et al. 2001) which would severely limit the generality and methods of actuation of an amorphous robot. Here in simulation we explore an ideal volumetric actuator, in which a given material expands isometrically (equally in all dimensions). A useful analogy is that we will be evolving robots with materials of varying thermal expansion coefficients (CTE), then "actuating" the robot by globally or locally varying the "temperature". Thus materials with a simulated CTE of zero will not change volume, whereas materials with a non-zero CTE will swell or contract isometrically as the temperature changes.

In these experiments, the temperature is assumed to vary sinusoidally over time, and slowly enough that actuation across the entire structure occurs simultaneously without heat diffusion effects. The period and amplitude of this temperature variation determine how dynamic the movement of the robot is. More complex actuation patterns including evolved brains will be examined in the future.

Methods

In this section we address the two main challenges of evolving amorpheous soft robots. First, we explore continuous representations of 3D multi-material objects unconstrained by topology, with the goal of maximizing interesting shapes and evolvability while minimizing the number of variables to be evolved. Each continuous representation is rendered to discrete voxel-space for simulation, which allows any suitable resolution to be used for the simulation process in order to balance computational efficiency with accuracy. Second, we outline our soft-body physics engine used to efficiently simulate the dynamics of amorphous robots with non-linear large deformations, volume-changing materials, and friction.

Representations

There are many possible representations for three dimensional freeform shapes for an evolutionary algorithm. Most prior examples use primitives (Sims 1994) but these are not conducive to creating smooth freeform shapes. We use a level-set class of representations that create a four-dimensional landscape, which is then threshholded to create a three dimensional solid (Sethian and Wiegmann 2000; Wang, Wang et al. 2003). A convenient analogy is to view the genotype as specifying a 3D density field, to which a threshold is applied. All the volume at a higher density is instantiated as part of the solid, whereas the rest is interpreted as empty space.

The level-set concept is versatile and useful for evolving shapes for several reasons. First, there is complete freedom in the topology of the object. More importantly, a continuous evolution path between different topologies exists since a phenotype's topology is derived, not prescribed. Moreover, this representation allows multiple materials to be seamlessly interspersed throughout the volume. A density field for each material is generated. Then the boundary of the volume is determined by thresholding the sum of the density fields of each material at each location. The material with the highest density at each location within the lattice is instantiated at that location. Alternatively, mixtures of materials could be described by blending materials in ratios proportional to their respective density fields.

We explore three different representations that create 3D density fields: (a) The Discrete Cosine Transform (DCT) representation (Hiller and Lipson 2009), (b) the Compositional Pattern Producing Network (CPPN) representation (Stanley 2007), and (c) the Gaussian Mixtures representation (Pernkopf and Bouchaffra 2005). Each of these was chosen to create smooth shapes of multiple materials. Each representation is also open ended in that it has the ability to increase the complexity of the resulting objects at the expense of the number of evolved parameters.

Discrete Cosine Transform (DCT). The discrete cosine transform is a special case of the discrete Fourier transform, in which boundary conditions favorable to creating amorphous morphology shapes are enforced. In the DCT representation (Hiller and Lipson 2009) the phenotype is a 3D matrix of frequency amplitudes, ranging from -1 to 1. To convert each phenotype to a genotype, the inverse DCT is applied to each row of each dimension of this matrix, converting from the

frequency domain to the spatial domain. Thus, each element in the frequency matrix scales a harmonic density field, where the number of modes in each of the three dimensions corresponds to its X, Y, and Z indices in the frequency matrix. A simple 1D example is shown in Figure 1. In this example, the 1D genotype matrix would be as follows:

$$[0.5 \quad -0.2 \quad -0.6 \quad 0.5 \quad -1.2]$$

The first element of this matrix scales the fundamental harmonic, the second element scales the second harmonic, and so on. These weighted harmonic functions are then summed to create a density field, which is thresholded at zero. In the 1D case, this results in a "freeform" 1D line segment, as shown in red in Figure 1. By extension, a 3D matrix of frequency components results in a freeform 3D solid.

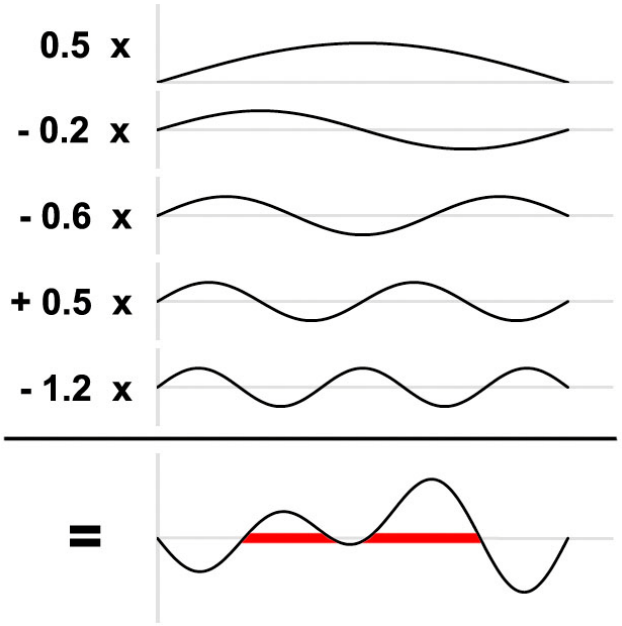


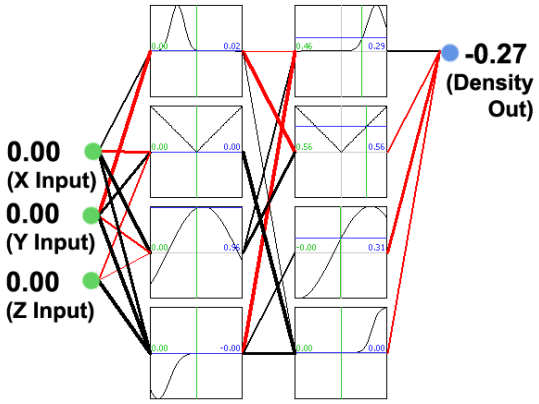
Figure 1: The inverse discrete Fourier transform representation sums weighted sinusoids, then thresholds them at zero as shown in this 1D example. The weights are the only evolved parameters.

The usefulness of the DCT representation for creating smooth, amorphous shapes is realized when the evolved frequency amplitude matrix is smaller than the rendered matrix of voxels in the spatial domain. Before the inverse DCT is applied, the frequency amplitude matrix of the genotype is simply padded with zeros to match the dimension of the number of voxels in the phenotype. Thus, smooth, freeform shapes are created.

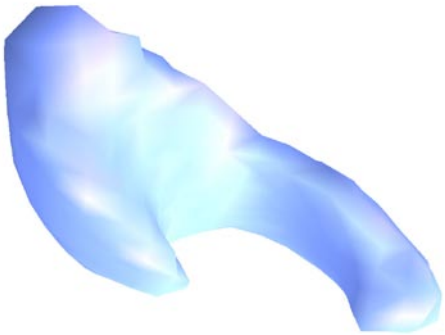
When evolving freeform amorphous morphologies using the DCT representation, mutation involves making small changes (up to 5%) in amplitude of these frequency components. A mutation rate of 20% was used. The crossover operation randomly selects each frequency component from either parent to create offspring.

Compositional Pattern Producing Network (CPPN). Compositional Pattern Producing Networks (CPPNs) (Stanley 2007) have been demonstrated to be useful for evolving two dimensional density fields (often interpreted as grey-scale images). Here, we introduce the third dimension to produce 3D density fields to threshold into amorphous morphologies. CPPNs are similar in concept to an artificial neural network (ANN), except that more geometrically-useful transfer functions are used instead of just sigmoids. A network of nodes (each containing a function) are connected by weighted paths. In order to create 3D amorphous morphologies, three coordinates (X, Y, and Z) that represent the position of a point in 3D space are used as inputs. The network has a single output, which represents the resulting density at that point. By sweeping through X, Y, and Z, the full 3D density field is obtained

Unlike ANNs, however, a variety of activation functions are used in a CPPN. Activation functions used here include traditional sigmoids and Gaussians, as well as sinusoids and the absolute value function for inducing repetition and symmetry, respectively. For each node (function), several parameters were evolved. These include the function type, offsets, and scaling. Additionally, a complexity measure was implemented to control minimum feature sizes, such that features were not being lost at a sub-voxel scale. Weights between nodes were also subject to evolution. An example CPPN and the resulting geometry is shown in Figure 2.



(a)

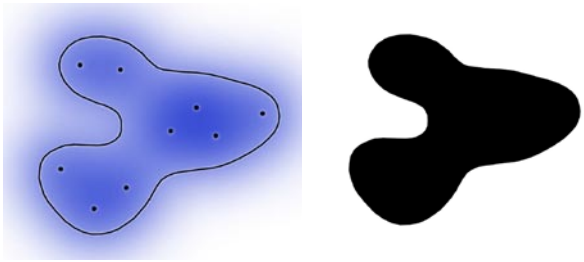


(b)

Figure 2: The Compositional Pattern Producing Network (CPPN) representation evolves a network of functions with three inputs (X, Y, and Z) and one output, which is the density at that location. The node functions and connecting weights (negative shown red, positive shown black) are evolved. After sweeping the inputs and thresholding, the network (a) produces a 3D freeform shape (b).

A CPPN has many evolvable parameters. Given a network that is m layers deep with n nodes per layer, there are a total of m×n nodes. Each node has an assigned activation function and three parameters that describe it (offset and scaling along the X axis, and scaling along the Y axis). There are also as many as $(m-1)\times n^2$ real-value weighted connections between the nodes, which can be either active or inactive. All these parameters are eligible for small changes upon mutation. However, the mutation rate is chosen such that on average only one of these values (in total) is adjusted. In the crossover operation, a rectangular "region" of nodes is selected from one parent, and the rest of the nodes are taken from the second parent. This region is chosen such that all nodes are equally likely to be in the selected region, not favoring the center nodes.

Gaussian Mixtures (GMX). The Gaussian mixtures representation also relies on the density field analogy of the level-set method common to all these representations. In the GMX representation, the density field is initialized with zero density everywhere. Then, points of density with Gaussian falloff are added within the spatial envelope. These points can have either positive or negative weights, which add or subtract from the density respectively. If only one Gaussian point was used, the resulting thresholded solids would always be spheres. However, with a relatively small number of Gaussian Mixtures, interesting and complex shapes and topologies can result. A simple 2D example with equal size and equally weighted Gaussian points is shown in Figure 3.



(a) (b)

Figure 3: In the Gaussian Mixtures representation, the locations and intensities of Gaussian distributions are evolved. In this simple 2D example, nine Gaussian point locations with equal sizes and weights (a) are threshholded to create a freeform 2D shape (b).

Mutating the Gaussian Mixtures representation involves making small changes to the location, density, and falloff (radius) of a Gaussian point, and occasionally adding or removing points. A mutation rate of 20% was used. Crossing over two individuals is accomplished by initializing a random plane that intersects the volume of the workspace. Points from one side of the plane are taken from one parent, while points from the other side of the plane are taken from the second parent. Here we used only spherical distributions though general Gaussians could be used as well by representing each distribution using a covariance matrix.

Soft Body Simulation

In order to efficiently evaluate amorphous soft robot morphologies with volumetric actuation, we developed a soft body simulator ab initio in C++. The main features of our simulation are:

1) **Speed:** With thousands of time steps per evaluation, and thousands of evaluations per evolutionary run, the feasibility of evolving amorphous robots depends on having an efficient simulation.
2) **Dynamics:** Full dynamics modeling with variable damping allows for realistic, 2nd order momentum effects in all translational and rotational degrees of freedom.
3) **Large deformation:** Shapes can be bent and twisted far past any linear small angle approximations without revoxelizing.
4) **Multi-material:** Any number of materials can be combined in any internal material distribution, each with varying stiffnesses and densities.
5) **Friction:** Nonlinear friction is incorporated with a static/dynamic friction model.
6) **Collision detection and handling:** Self intersection is calculated and enforced. With large deformation comes the need to avoid an object penetrating itself.

When a continuous amorphous robot object is imported into the simulation, it is first voxelized at an appropriate resolution. These voxels are then simulated according to the appropriate statics and dynamics, and the continuous mesh is drawn according to the deformation of the nearest voxel (Alec and Doug 2007). At each time step, the total force on each voxel is calculated. Then, the momentum (P) of each voxel is updated according to the length of the time step (Δt) and the total force.

$$P_t = P_{t-1} + \sum F \times \Delta t$$

Linear damping was modeled, which is consistent with the internal damping of most bulk materials. The loss factor (η) is normalized by the length of time step and determines how much energy (in the form of momentum) is lost at each time step.

$$P_{new} = P_{old} \times \eta$$

Finally, the momentum is numerically integrated to get the change in position (ΔX) of the voxel. The positions of each voxel are synchronously updated, and the process repeats.

$$\Delta X = \frac{P \times \Delta t}{m}$$

Although the equations above illustrate the translational degrees of freedom, the equivalent equations are used to model the rotational degrees of freedom. An example of the freeform, large-displacement nature of this soft-body simulation is shown in Figure 4.

The interaction between individual voxels is modeled by a standard flexible beam model. This allows all 6 relative forces and moments to be calculated based on the relative 3D position and rotation of the two voxels. For computational efficiency, each element is transformed to point in the positive X direction before the reactions are calculated. The reaction forces and moments then undergo the inverse transform to put

them back in the reference frame of the actual element. When considering two voxels of differing material properties (such as stiffness), the bond is assumed to have the composite stiffness of the two materials connected in series.

Choosing the optimal time step is critical to an efficient simulation. If the time step is too small, computation time is wasted with the extra time steps. However, a time step that is too large will result in diverging instability within the simulation. Calculating the optimal time step for an arbitrary geometry with varying stiffness in the material and non-linear interactions such as collisions and friction is non-trivial. To address this we experimentally determine the optimal time step upon importing an object to the simulation. This involves a series of short simulation runs, with steadily decreasing time steps. Since divergence is exponential, very few time steps are needed to determine if a given time step increment is unstable. Thus, the first simulation without a significant increase in strain energy by 100 time steps is assumed stable, and a safety factor of 5% is incorporated. For efficiency, very coarse steps are taken at first (one order of magnitude apart), then a second finer pass determines the optimal time step at a higher resolution.

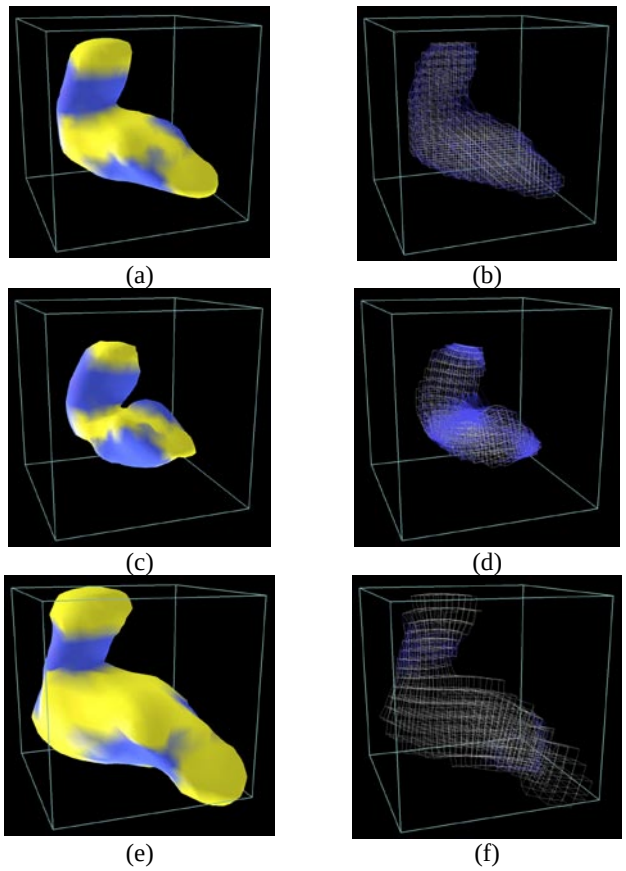


Figure 4: A randomly generated 3D object (a) is imported into the soft-body simulation, which voxelizes the object at an appropriate level of detail. (b) Potential self-intersection collisions are shown as blue lines. As the temperature varies, the volume of the yellow material shrinks (c & d) and swells (e & f) accordingly. The volume of the blue material remains constant.

Calculating self intersection is necessary in a soft-body simulation since the objects deform significantly. However, calculating collisions for a large number of lattice points is computationally intensive. This has an order of n^2 complexity, where the number of lattice points n can easily be in the thousands. This would dominate the computational time of the physical simulation itself, which is order n complexity. To address this, we made use of several useful simplifications. First, only the voxels on the exterior of the object need be considered for collisions. Second, an intermediate list of possible collisions between bonds can be maintained. Initially, each pair of voxels within an absolute distance 2.5 voxels but which are not touching within two links in the lattice are added to the list. Then, at each time step only the possible contact bonds on this list are considered. The list is periodically regenerated when the absolute displacement of any voxel in the lattice is enough to touch a voxel not on its list of possible interactions.

Several parameters of the simulation were chosen to be of interest for exploring further. The first is the level of dynamic response. This term refers to the importance of the momentum term of the material. An object with a high level of dynamic response could be a very dense, soft, rubbery object actuated near resonance, where movements can be significantly out of phase with the actuation. Conversely, an object with a low level of dynamic response would be light and stiff (or actuated very slowly), such that the static movement dominates the momentum terms. Several combinations of static and dynamic friction were also explored, ranging from realistic values to exaggerated stick-slip scenarios. For the bulk of experiments, the coefficient of dynamic friction was 0.3 and the coefficient of static friction was 1.0.

Each evaluation of an amorphous machine was broken into two segments. The relaxation segment settles the object under gravity and friction, allowing it to come to rest in a neutral position. In the movement segment, temperature oscillations begin. After 10 complete temperature cycles, the magnitude of change in position of the center of mass during the movement segment is returned as the fitness for a given individual.

Evolution parameters and performance

The solutions presented here were each evolved on a single quad-core desktop computer. Each solution was voxelized into a 20×20×20 workspace, which provided suitably accurate resolution while remaining computationally feasible. At a rate of approximately 3-15 seconds per evaluation (depending on number of instantiated voxels), 20,000 evaluations in a 24 hour day was typical. Deterministic crowding selection was used (Mahfoud 1996), in which an offspring replaces its most similar parent if it outperforms it. Small population sizes work well with this crowding method, so a population size of 20 was chosen for all experiments. The mutation rate was different for each representation as detailed above.

Results

The evolved behaviors of the amorphous robots generally took advantage of a combination of dynamics and non-linear friction to make forward progress. Two modes of movement in the desired direction were generally observed: Several

robots made significant progress by maximizing the distance traveled as they fell and flopped over. This movement was often aided by the actuation cycles gradually tipping the robot over, but this method of movement does not count as true locomotion because it cannot be sustained over an indefinite distance. The more successful mode of movement involved scooting, in which the robots expanded and contracted in specific ways, making and breaking contact friction selectively to make forward progress. Several observed solutions made use of a combination of flopping and scooting.

Two material results

In the first experiments, a palette of two equal stiffness and density materials was used. The first material (shown in blue) had a CTE of zero, signifying that it was not actuated. The second material (shown in yellow) had an arbitrary CTE of 0.01. The temperature was varied sinusoidally globally with an amplitude of 30 degrees, leading to a ±30% change in volume of the actuated material. The period of oscillation was 500 time steps.

Comparison of Representations

The three representations under consideration were all run three times for a total of nine evolutionary runs. Figure 6 displays the average and standard error of the best solution of each of the three runs. The GMX representation outperformed the other representations consistently. This may be a result of locality that preserves geometric novelties in the crossover process and can make small changes to specific areas of the robot through mutation.

The DCT representation fell behind and had a very large standard error, which means that the genetic algorithm was not able to consistently find good solutions. This is likely because each mutation in the genome has a global effect across the entire structure, a characteristic that couples the mutations and prohibits small, subtle changes.

The CPPN representation, as implemented, was not well suited to evolving freeform amorphous morphologies. Mutations drove the solution toward filling the entire workspace with material, a trend that significantly slowed the simulation down (since more elements needed to be simulated) and also led to fewer interesting geometries. Resulting amorphous robots generated by each representation are shown in Figure 5.

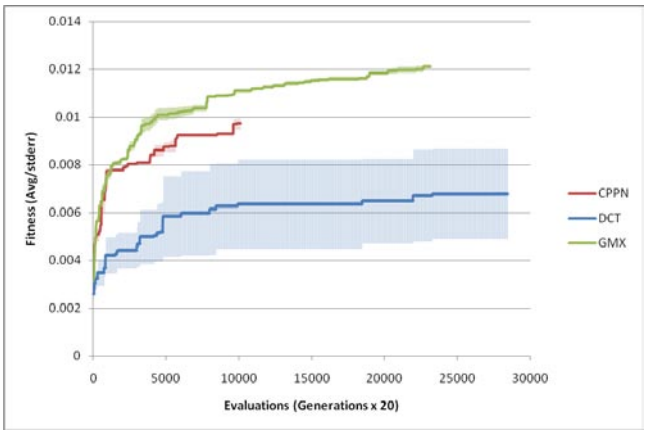


Figure 6: Three independent runs were completed for each of the three representations. The average and standard error of the three best solutions are plotted for each. The GMX representation outperformed the others.

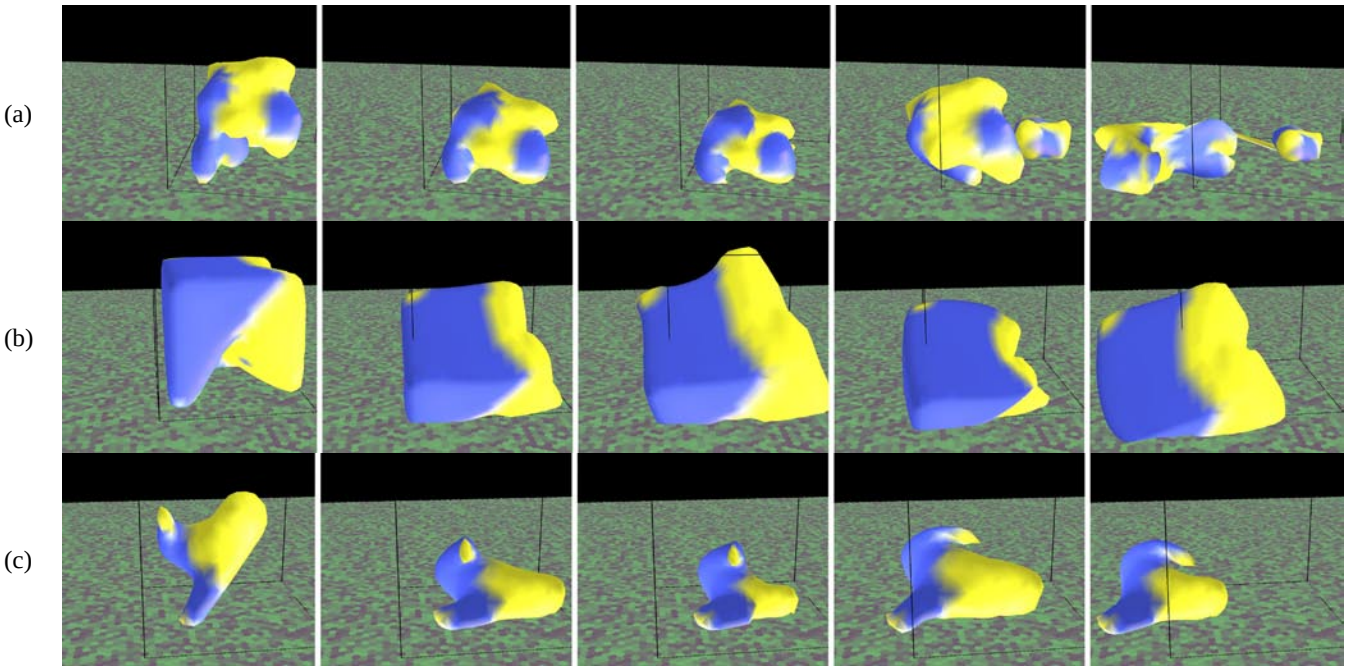


Figure 5: Evolved robots for the DCT (a), CPPN (b), and GMX (c) representations demonstrate successful locomotion. The blue material is passive, while the yellow material changes volume sinusoidally. The first frames for each show the initialized shape, the second frames show the settled result under gravity, and the following frames are snapshots of its motion. Direction of motion is to the left.

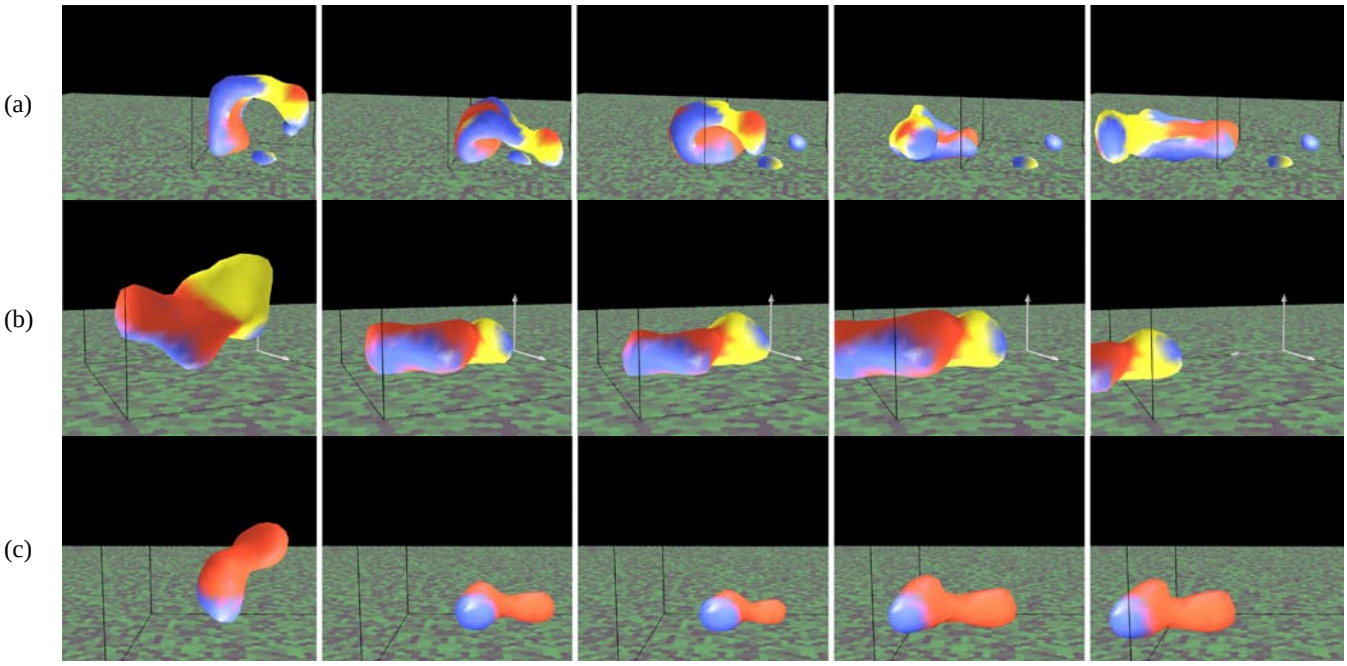


Figure 7: Evolved amorphous morphologies showing flopping (a), scooting (b), and dynamic bouncing (c) behaviors. The yellow and red materials are both sinusoidally actuated, but 90 degrees out of phase. Direction of motion is to the left.

Three material results

A second volumetrically actuated material was introduced to explore the possibilities enabled by multiple actuation modes. The second actuator material was simulated the same stiffness and density as the others, but with 90 degree phase lag in actuation. It was hypothesized that this would enable different modes of locomotion, such as continuous rolling or more interesting gaits. However, only the more primitive locomotion modes of flopping (Figure 7a) and scooting (Figure 7b) were observed.

Dynamic response. By varying the actuation speed of the temperature fluctuations and the internal material damping, the importance of momentum effects in the amorphous morphologies can be adjusted. The best solution of the dynamic runs ended up using only one actuator material, as shown in Figure 7c. Based on the size and mass of the optimized object, however, the dynamics were strongly exploited to bounce forward.

Friction. Different parameters for friction bias the solution towards different modes of locomotion. Experiments were run with very low dynamic friction and high static friction (0.1/5.0) and with dynamic and static friction values that were very close (0.4/0.5). The solutions with very high static friction tended to exhibit flopping/rolling over behavior, such as in Figure 7a, since the force to overcome static friction was extremely high. However, solutions with moderate static and dynamic friction tended towards the scooting locomotion, such as in Figure 7b.

Conclusions

We have demonstrated that evolutionary algorithms are suitable for designing the freeform material distribution of locomoting amorphous robots, which would be a difficult task to perform in traditional CAD software. This opens the door to a new design space of soft robotics, where the functionality of the robot is determined by the material distribution, not by rigid links. Sensing, actuation, and computation can all be distributed, potentially making the design of these robots even more difficult without the aid of design automation methods. Thus, with the exponentially expanding design space of robotics enabled by additive manufacturing of multiple materials, genetic algorithms and other design automation methods will play an increasingly important role in designing robots to directly meet high level functional goals.

Acknowledgements

This work was supported in part by a National Science Foundation Graduate Research Fellowship and NSF Creative-IT grant 0757478.

References

Alec, R. R. and L. J. Doug (2007). FastLSM: fast lattice shape matching for robust real-time deformation. ACM SIGGRAPH 2007 papers. San Diego, California, ACM.

Bar-Cohen, Y. (2001). Electroactive Polymer (EAP) Actuators as Artificial Muscles - Reality, Potential and Challenges, SPIE Press.

Beaman, J. J., H. L. Marcus, et al. (1997). Solid Freeform Fabrication: A New Direction in Manufacturing, Kluwer Academic Publishers.

Bendsoe, M. P. and N. Kikuchi (1988). "Generating optimal topologies in structural design using a homogenization method." Comput. Methods Appl. Mech. Eng. **71**(2): 197-224.
Bongard, J. (2003). Incremental Approaches to the Combined Evolution of a Robot's Body and Brain. Zurich, University of Zurich. **Ph.D.:** 193.
Hiller, J. and H. Lipson (2009). "Design and Analysis of Digital Materials for Physical 3D Voxel Printing." Rapid Prototyping Journal **15**(2): 137-149.
Hiller, J. and H. Lipson (2009). Multi Material Topological Optimization of Structures and Mechanism. Genetic and Evolutionary Computation Conference. Montreal, QC.
Hornby, G. S., H. Lipson, et al. (2001). Evolution of generative design systems for modular physical robots. Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on.
Hornby, G. S., S. Takamura, et al. (2005). "Autonomous evolution of dynamic gaits with two quadruped robots." Robotics, IEEE Transactions on **21**(3): 402-410.
Ijspeert, A. J. and J. r. m. Kodjabachian (1999). "Evolution and Development of a Central Pattern Generator for the Swimming of a Lamprey." Artificial Life **5**(3): 247-269.
Lipson, H. and J. B. Pollack (2000). "Automatic design and manufacture of robotic lifeforms." Nature **406**(6799): 974-978.
Mahfoud, S. (1996). "Niching methods for genetic algorithms."
Malone, E. and H. Lipson (2007). "Fab@Home: the personal desktop fabricator kit." Rapid Prototyping Journal **13**(4): 245-255.
Malone, E., K. Rasa, et al. (2004). "Freeform fabrication of zinc-air batteries and electromechanical assemblies." Rapid Prototyping Journal **10**(1): 58-69.
Mast, S. O. (1926). "Structure, movement, locomotion, and stimulation in amoeba." Journal of Morphology **41**(2): 347-425.
Mozeika, A., E. Steltz, et al. (2009). The First Steps of a Robot Based on Jamming Skin Enabled Locomotion. International Conference on Intelligent Robots and Systems (IROS). St. Louis, USA**:** 408-409.
Nishiwaki, S., M. I. Frecker, et al. (1998). "Topology optimization of compliant mechanisms using the homogenization method." International Journal for Numerical Methods in Engineering **42**(3): 535-559.
Nolfi, S. and D. Floreano (2002). "Synthesis of autonomous robots through evolution." Trends in Cognitive Sciences **6**(1): 31-37.
Objet. (2010). "Objet Geometries inc." Retrieved January 24, 2010, from http://www.objet.com.
Pernkopf, F. and D. Bouchaffra (2005). "Genetic-based EM algorithm for learning Gaussian mixture models." Pattern Analysis and Machine Intelligence, IEEE Transactions on **27**(8): 1344-1348.
Pollack, J. B., H. Lipson, et al. (2001). "Three Generations of Automatically Designed Robots." Artificial Life **7**(3): 215-223.
Quillin, K. J. (1999). "Kinematic scaling of locomotion by hydrostatic animals: ontogeny of peristaltic crawling by the earthworm lumbricus terrestris." J Exp Biol **202**(6): 661-674.
Rieffel, J., F. Saunders, et al. (2009). Evolving soft robotic locomotion in PhysX. Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers. Montreal, Quebec, Canada, ACM.
Rieffel, J., B. Trimmer, et al. (2008). Mechanism as Mind: What Tensegrities and Caterpillars Can Teach Us about Soft Robotics. Artificial Life (ALIFE) XI. Winchester, UK**:** 506-512.
Sethian, J. A. and A. Wiegmann (2000). "Structural Boundary Design via Level Set and Immersed Interface Methods." Journal of Computational Physics **163**: 489-528.
Sims, K. (1994). Evolving virtual creatures. SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques. Orlando, Florida, ACM**:** 15-22.
Stanley, K. (2007). "Compositional pattern producing networks: A novel abstraction of development." Genetic Programming and Evolvable Machines **8**(2): 131-162.
Wang, M. Y., X. Wang, et al. (2003). "A level set method for structural topology optimization." Computer Methods in Applied Mechanics and Engineering **192**(1-2): 227-246.
Zykov, V., J. Bongard, et al. (2004). Evolving Dynamic Gaits on a Physical Robot. GECCO '04.

Grounding Motivation in Energy Autonomy: A Study of Artificial Metabolism Constrained Robot Dynamics

Robert Lowe[1], Alberto Montebelli[1], Ioannis Ieropoulos [2], John Greenman [2], Chris Melhuish [2] and Tom Ziemke[1]

[1]University of Skövde, Cognition and Interaction Lab, Sweden
[2]University of West England, Bristol Robotics Lab, UK

Abstract

We present an evolutionary robotics investigation into the metabolism constrained homeostatic dynamics of a simulated robot. Unlike existing research that has focused on *either* energy *or* motivation autonomy the robot described here is considered in terms of *energy-motivation autonomy*. This stipulation is made according to a requirement of autonomous systems to spatiotemporally integrate environmental and physiological sensed information. In our experiment, the latter is generated by a simulated artificial metabolism (a microbial fuel cell batch) and its integration with the former is determined by an *E-GasNet*-active vision interface. The investigation centres on robot performance in a three-dimensional simulator on a stereotyped two-resource problem. Motivation-like states emerge according to periodic dynamics identifiable for two viable sensorimotor strategies. Robot adaptivity is found to be sensitive to experimenter-manipulated deviations from evolved metabolic constraints. Deviations detrimentally affect the viability of cognitive (anticipatory) capacities even where constraints are significantly lessened. These results support the hypothesis that *grounding* motivationally autonomous robots is critical to adaptivity and cognition.

Introduction

The pursuit of imbuing robots with levels of autonomy has resulted in recent emphasis on internal dynamics of robotic systems as they affect adaptive and cognitive behaviour (cf. Parisi 2004, Ziemke and Lowe 2009). McFarland (2008) has identified three core levels of autonomy – *energy*, *motivation* and *mental* levels and suggests: "Autonomy implies freedom from outside control. There are three main types of freedom relevant to robots. One is freedom from outside the control of energy supply. Most current robots run on batteries that must be replaced or recharged by people. Self-fuelling robots would have energy autonomy. Another is freedom of choice of activity. An automaton lacks such freedom, because either it follows a strict routine or it is entirely reactive. A robot that has alternative possible activities, and the freedom to decide which to do, has motivational autonomy. Thirdly, there is freedom of 'thought'. A robot that has the freedom to think up better ways of doing things may be said to have mental autonomy" (McFarland 2008, p.15).

Naturally, how the robot designer is to seamlessly integrate these levels of autonomy represents another challenge but inspiration can be derived from biology. A key feature of biological autonomous systems is homeostatic regulation. Drawing from Cannon (1915), the importance of bodily 'essential' variables to behavioural dynamics was identified in an artificial systems context by Ashby (1960). Ashby's *homeostat* produced feedback signals following deviation from a pre-set range of the essential variables (*EV*s). While Ashby's notion was deliberately abstract, biological evidence for the effects of *EV*s on regulation of behaviour has recently been found regarding feeding and drinking. Canabal et al. (2007) discovered that levels of extra cellular glucose in hypothalamus can impact on neural activity via slow diffusing nitric oxide (NO) molecules. NO emissions in glucose-sensitive cells correlate with feeding (cf. Morley et al., 1999) while 'osmoreceptor' cell NO emissions in hypothalamus correlate with drinking (cf. Yao et al. 2005).

Robot controllers have utilized bio-inspired mechanisms for 'brain-body' interfacing in the areas of: navigation (Vargas et al. 2009), foraging (McHale and Husbands 2006), two-resource problems (Avila-Garcìa and Cañamero 2004). This work has, however, invariably abstracted away details of the dynamic grounding of brain-body interfacing. Specifically, metabolic dynamics and their imposed behavioural constraints have been ignored. Instead, emphasis has been placed on motivation-like states (cf. McFarland and Spier 1997) as a function of *abstract* internal (essential) variable, and externally sensed, information. Such states are typically non-grounded either evolutionarily or metabolically. The resulting homeostatic expression of such robots may, therefore, be critically constrained regarding adaptive behaviour in spatial-temporal realistic environments.

Research into metabolic performance constraints has been carried out in recent years in the form of microbial fuel cell (MFC) robotics applications (cf. Melhuish et al. 2006, Ieropolous et al. 2007). MFCs can provide wheeled robots with (electrical) energy for driving motors as constrained by bio-chemical *EV* dynamics. MFC technology has the capacity to produce bioelectricity from virtually any unrefined

renewable biomass (e.g. wastewater sludge, ripe fruit, flies) using bacteria; thus, when used as the power source for actuation MFCs equip robots with a degree of energy autonomy concerning choice of 'recharging' resource. A limitation of artificial metabolism motored robots such as *EcoBot* (cf. Melhuish et al. 2006) given the present state of the art, however, is the energy requirement for actuating motors. Consequently, the robot may take as long as 15 minutes to move 15mm. This renders experimentation with new forms of homeostatic control and performance optimization challenging. A need is evident for simulations based scenarios for assessing the potential development of metabolism constrained robotic behaviour.

In the rest of this article we will describe an initial investigation into the dynamics of a robotic system that integrates two levels of autonomous control – *energy-motivation*. Two themes address the influence of simulated metabolic constraints on: 1) evolved sensor-motor resource acquiring strategies, 2) the emergence of affective ('motivational') dynamics. Spatiotemporal coherence between *internal* and *sensorimotor* domains is evaluated as it renders adaptive and cognitive behaviour. In the next section we introduce the components of the *energy-motivation autonomous robot* and our methodological approach. The results section evaluates themes 1) and 2) according to a comparative case study of best evolved controllers. The discussion section includes reference to present and future work.

Robot Architecture and Methodology

There are three architectural components: 1) a brain-body interface (*E-GasNet*) between 2) artificial metabolism (MFC model), and 3) sensorimotor (active vision) system. Below, each component is described in turn followed by the methodology used to assess the spatiotemporal coherent integration of the three components to adaptive behaviour.

Robot Architecture: The E-GasNet

The neurophysiological controller we propose is an extension of the GasNet (Husbands et al. 1998). The essential components comprise a standard neural network the activity of which is modulated by nitric oxide (NO) emissions enabling a spatiotemporal dynamic that when embedded in a wheeled robot tunes network performance to task requirements (cf. Smith et al. 2002). Work has been carried out utilizing GasNets according to evolutionary robotics investigations on bodily homeostasis (cf. Vargas et al. 2009) and energy constraints (cf. McHale and Husbands 2006). The focus, however, has not been on the incorporation of non-neural bodily states into GasNet 'nervous system' activity.

Based on the neuroscientific findings referred to in the previous section, we propose the *E-GasNet* ('Essential Variable Monitoring GasNet') as a type of GasNet developed according to an evolutionary robotics approach. The novel feature it incorporates is the use of *EV* level sensing nodes

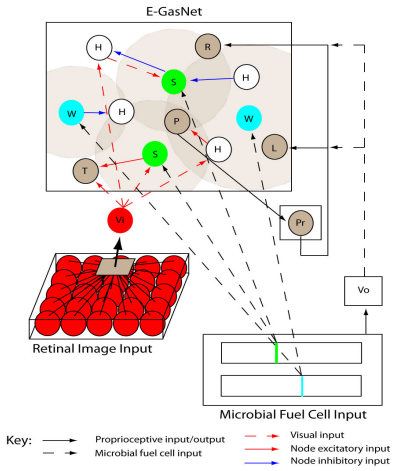


Figure 1: E-GasNet component of the complete energy-motivation autonomous robot architecture. Nodes: H = hidden, L = left motor, R = right motor, P = pan, T = tilt, V_i = Visual input, Pr = pan proprioception, W = water sensitive e-node, S = substrate sensitive e-node, V_o = MFC voltage input. Grey shaded circles depict potential e-node gas emissions. Green and blue coloured vertical lines provided by MFC represent substrate and water levels, respectively.

(for water and metabolizing-substrate) that emit gas contingent on the state of concomitant *EV*s. We term these nodes *e-nodes*. The E-GasNet (fig.1) represents the interface between artificial metabolic *EV* dynamics and actuators – (left and right) motors and active vision (pan and tilt) nodes. Depending on topological positioning on the two dimensional plane motor nodes in the network are modulated only by a retinal pan proprioception node and gas modulation – this simplified analysis concerning comparative sensorimotor activity. Pan and tilt nodes are modulated by electric input and gas. Electric input permits transient retinal image positioning on the camera. The position of nodes on the plane, the number of e-nodes and the sign and connectivity are determined by a genetic algorithm or GA (see methodology). An E-GasNet consists of four actuator nodes, four 'hidden' nodes and a variable number of e-nodes. E-nodes emit gas modulating the electric activity of neighbouring nodes (within a genetically specified radius) via affecting the gain of the output function. Gas emissions are dependent on a genetically determined e-node specific *EV* threshold. *EV* values are provided by the MFC dynamics (see next sub-section). Hidden nodes do not emit gas. Output from the MFC gates motor wheel activity while an output from a visual node provides a mean value of all cells on a 'retina' which inputs to the network as it pans and tilts across the camera image. The E-GasNet dynamic is governed by the same set of difference equations utilized by Husbands et al. (1998) and, where slightly adapted, Smith et al. (2002). It is to these papers that the reader is referred for details of electric output and gas emission dynamics.

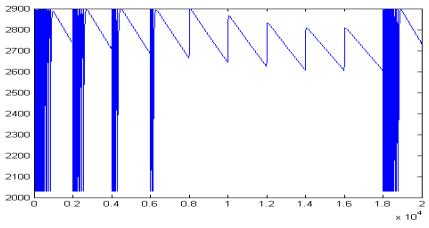


Figure 2: MFC model of system level (electric) energy output dynamics. The vertical axis provides output voltage (mV) where 2900 is the discharge level providing energy to the actuators (e.g. motors), the horizontal axis represents time in arbitrary units.

Robot Architecture: Artificial Metabolism

This component is comprised of the Microbial Fuel Cell (MFC) model of Montebelli et al. (2010a), designed at a level of abstraction purpose-made for autonomous robotics investigations. Critical to energy-motivation autonomous level integration is the charge-discharge electric output dynamics that gate motor wheel activation. An example of this dynamic is illustrated in figure 2 according to a substrate exhaustion cycle. At a threshold of electricity storage at the MFC capacitor bank (pre-set to 2900mV) energy is utilized by motors that indirectly contribute to the maintenance of the charge-discharge dynamic, i.e. through feeding/drinking. After a period without substrate, the charge is not arrived at in spite of periodic rehydration (every $0.2*10^4$ time units) at the cathode. Re-establishment of an efficient output dynamic owes to simulated fuel source provision at $1.8 * 10^4$ at the anode. This cycle demonstrates the requirement for *both* water and substrate (*EV*s) to be replenished for efficient system level energy to be produced. Reduced charge rate ensures less energy for the motors.

In the set-up used in our investigation, the robot produces a pulsing motor behaviour similar to 'EcoBot' (cf. Melhuish et al. 2006). This entails energy being made available to the motors for a short time window following the point of discharge. Where MFC performance degrades, motor pulses slow and in turn MFC performance continues to degrade as resource acquisition capacity is impaired. If the discharge threshold is not reached, motor output eventually ceases – no such constraint has been placed on visual sensing at present. For specific values used in our experimentation and an alternative application see Montebelli et al. (2010b).

The E-GasNet is evolved to track the level of the *EV*s in the MFC model – the GA may 'select' for e-nodes that 'monitor' the level of either substrate or water according to a genetically determined threshold value specific to the particular node. If the *EV* level falls below such a node-specific threshold, gas emission is initiated and linearly increments to an upper bound; only when *EV* values are re-established above threshold (as set by the GA) does the gas level linearly dissipate. In this Ashby-like manner, e-node activity

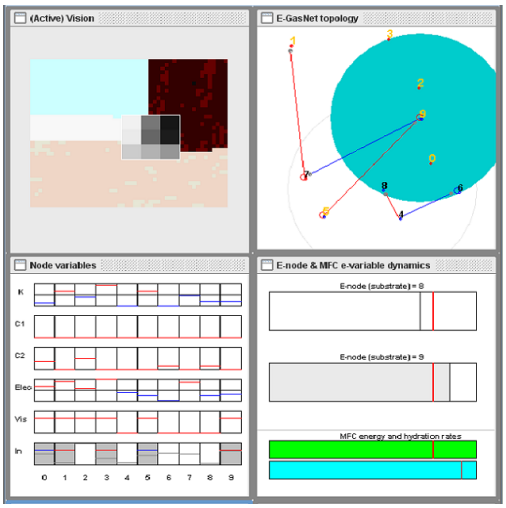


Figure 3: Controller dynamics top-left: retina superimposed on camera image, top-right: E-GasNet topology – blue circle represents (inhibitory) gas emission at node 9, bottom-left: E-GasNet parameters – K is gain level; $C1/C2$ are gas 1/2, respectively; $Elec$ is the electric output of each network node; vis is the scalar input from the retina (here 3*3 units); In is actual visual input, i.e. above noise threshold ensuring a differentiated node response to visual input, bottom-right: *EV* dynamics of the MFC including the same dynamics as they relate to e-node gas emission thresholds.

serves to anticipate the effect that *EV* depletion will have on the 'life-energy' output of the MFC providing a mode of embodied cognition. This occurs since MFC electric output cycles depend absolutely on efficient regulation of these two *EV*s. The e-node gas emission is the means by which body can interface with sensor-motor activity in order to pre-empt catastrophic performance degradation.

Robot Architecture: Sensor-Motor Morphology

An E-puck robot simulated in Webots (*Cyberbotics Ltd.* – *http://www.cyberbotics.com*) was used but any simple wheeled robot may be suitable. Our emphasis is on *integration* of sensorimotor capacities with neurophysiological dynamics. Sensor input consisted of a low dimensional grey-scaled retinal image superimposed on an e-puck camera image. The 'retina' is initialized for each evaluated robot controller in the centre of the camera image but may pan and tilt through 360 ° within the 2D bounds. Pan/tilt values (one node each) for the retina are modulated through: electric inputs from E-GasNet nodes, gas, a pan proprioception node. This permits a type of *active vision* similar to Floreano et al. (2004). A retinal scalar value inputs to GA-determined nodes in the network. Figure 3 provides a snapshot of the robot graphical interface for the retinal network (along with E-GasNet topology/activity and *EV* dynamics).

The equations that determine the active vision effects on robot dynamics are as follows: $P_o(t) = (C_x + R_w/2) - C_w/2$ and $P_r(t) = P_o(t-1)/(C_w - R_w)$ where $P_o(t)$ = pan

orientation at time t, C_x = x axis value of the robot camera image in [0,50], R_w = retina width genetically determined in [15,25], C_w = camera image width (50 pixels); $T_o(t) = (C_y + R_h/2) - C_h/2$ where $T_o(t)$ = tilt orientation at time t, C_y = y axis value of the robot camera image in [0,40], R_h = retina height genetically determined in [12,20], C_h = camera image height (40 pixels); $P_r(t)$ = pan proprioception at time t. Motor wheel output is determined by: $O_i(t) = b_r*(\alpha+P_r(t-1)*(V_i > V_ti)*(V_i-V_ti))$ where O = wheel output for node $i \in \{1,2\}$ at time t, b_r = 'burst' boolean, α = a constant set at 0.5, V = retina input in [0,1], V_t = a genetically determined node-specific threshold in [0,0.5].

Methodology: The Two Resource Problem

The *energy-motivation autonomous robot* was evaluated according to a two resource problem (McFarland and Spier 1997) where applicable resources are water and fuel substrate. The literal use of two resources (one of each type) serves as an initial benchmark control to facilitate identification of core principles and homeostatic dynamics. The two resource paradigm is a class of problem whereby *adaptive* sensorimotor activity enables a (quasi) optimal trade-off between two conflicting *EV* needs. Spier (1997) studied two-resource problems on 2D scenarios for agents utilizing an ethology-based *cue-deficit model* that states that likelihood of enacting a 'motivated' behaviour in animals is determined by the product of: 1) external stimuli, 2) related physiological need deficit. The realization of such a cue-deficit model in a 3D world is not obvious particularly if the robot sensors do not provide pre-given information with which to discriminate stimuli or/and implicitly provide information regarding stimulus distance/attainability. A stronger measure of autonomous capabilities is provided by robots remaining viable over long periods in partially human-known environments, possibly inhospitable to human habitation. Energy autonomous robots flexible in their means of refuelling are critical in this case. Realistic metabolic constraints impact on sensorimotor capabilities rendering high-level modelling approaches compromised regarding robot adaptivity to dynamic and challenging environments. Situated *integration* of internal and external sensing is therefore needed in order to enable motivational autonomous capabilities.

Evolved E-GasNet interfacing of metabolic and sensorimotor activity provides a spatiotemporally and metabolically situated cue-deficit model apt for 3D world robot performance where resource-specific sensory information concerning distance and type is not explicitly pre-given.

Methodology: An Evolutionary Robotics Approach

100 candidate controllers were evaluated over 50 generations via the distributed GA used by Husbands et al. (1998). Each evaluation consisted of a robot making 20 selections (one per trial) from the 2 available resources. Each trial is terminated either by successfully reaching a resource leading to instantaneous related *EV* replenishment, or if a resource is not reached by 500 cycles (basic timestep = 64ms). The latter time constraint ensured against inefficient/arbitrary approach behaviours. The metabolic constraints required the robot to 'switch' preference from one resource to the other at least twice ensuring against evolution of uninteresting dynamics. Agents viable after 20 trials were considered *survivors*. Both resources were within camera image scope at the beginning of each trial to limit potential bias towards one or other resource – test trials found no observable bias. Water and substrate resources were placed left and right of the robot trial starting position, respectively. This positioning – relative to the centre of the robot – was not varied in order to promote ease of analysis of the complex interactive dynamics of the system. Solutions were analyzed according to an independent variable (IV) – clamping, or not, of gas effects on motor node activity; the IV, thus, consisted of two values - a) Gas modulated motor output (*GM*), b) Non-gas modulated motor output (*NGM*). In a) motor output could be affected both by gas and the pan proprioception node; in b) motor output was modulated only by the pan proprioception node – this exerted evolutionary pressure for the emergence of 'active vision' strategies while purely electric inputs to the retina position otherwise ensured early stabilization. The only means by which robots can survive trials is by switching from one resource preference to the other over the 20 trials. This switching in the latter condition can, therefore, *only* be achieved via e-node gas modulation of pan-tilt activity. The emergence of e-node arbitration is therefore unsurprising. Our investigation instead focuses on exactly how such arbitration dynamically occurs.

The evaluation criteria consisted of 1) fitness, 2) no. *survivors*. Robot fitness is defined: $fit(t) = fit(t-1) + (subst(t) + wat(t))/2$ and $fit_\mu = t_{term} * (fit(t)/N_{tr})$ updated once per trial at time t, t_{term} is a boolean determining termination of the controller evaluation, i.e. at the end of N_{tr} = 20 trials. The fitness function captures physiological state at the time of resource acquisition while no assumptions concerning *ideal* state are made. Evolutionary parameters adhered to Husbands et al. (1998) but adopted the gaussian gas diffusion of Smith et al. (2002) and the connectivity schema of Jakobi (1998). Further parameters subject to the GA were: e-node no. (in [0,6]), e-node gas emission thresholds (in [0.0,1.0]), retina squared unit dimension size (in [3,5] where a unit = 5*4 pixels and camera dimensions are fixed at width = 50, height = 40). Finally, unlike the classical GasNet, left/right wheel (and pan/tilt) nodes' x, y coordinates were evolutionarily specified.

Results

Evolution and Evolvability of Strategies

Figure 4 illustrates fitness and survivor rate of all controllers over 10 runs. Evaluation of independent sample t-tests indicated that robots were significantly fitter in early generations

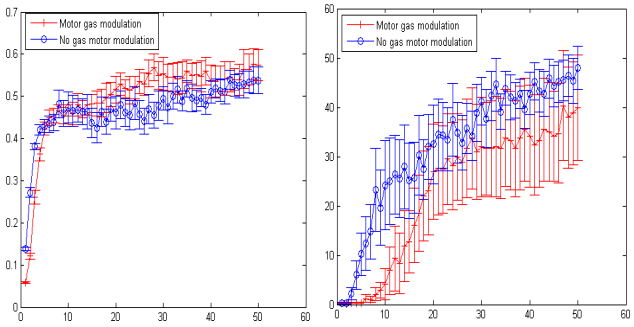


Figure 4: Left: Fitness (in [0.0,1.0]). Right: Survival rate. Mean values for agent popn. over 50 generations comparing gas modulating motor activity (*GM*) to non-gas motor modulating (*NGM*) runs; Error bars calculated as: $SE = \frac{sd}{\sqrt{N}}$ where N = 10 runs.

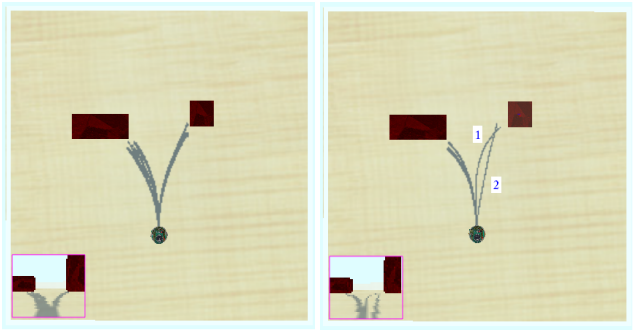


Figure 5: Inter-trial motor trajectories – inset camera images show in-trial perspectives. Left: *GMC* trajectories (20 trials). Right: *NGMC* trajectories (for visibility – trial 1 and 2, and a sub-set).

(1-4) of *NGM* runs than they were in *GM* runs. Comparison of performance of survivors uncovered that only generations 5, 7-9 produced significant differences with higher survivor rate in *NGM* runs. All tests were at $p < 0.05$ for two-tailed tests with d.o.f = 18. These results suggest a tendency, early in evolution, to favour higher performance in *NGM* runs suggesting greater evolvability. However, allowing motor nodes to be potentially modulated by gas emissions in *GM* runs ensured additional genome complexity possibly requiring more generations for adaptive strategies to manifest. The high survival variance in *GM* runs – 3/10 runs produced no survivors by generation 50 – compared to 10/10 in *NGM* runs producing > 30 survivors by generation 50 – and higher mean in *NGM* runs hints at *NGM* strategies differentiable from those found in *GM* runs.

MFC Constraints: A Comparative Case Study

An in-depth evaluation of individual controllers taken from the best runs of each condition furnished a case study comparison in order to unveil adaptive strategies. Owing to evolution converging on a common ancestor by generation 50 a given controller selected from the *genome candidate solution grid* (see Husbands et al. 1998) provided a typical evolved topology for the run. We compared only viable controllers, i.e. ones that enabled robots to 'survive' 20 trials.

Figure 5 depicts trial-by-trial motor trajectories for the two controllers. On the left of the figure is the *GM* controller (*GMC*). Typically, per trial, the robot followed an arced path towards the nearest edge of the approached resource which is energy-efficient. On the right of the figure is the *NGM* controller (*NGMC*) showing a similar pattern of approach for the water resource (left-side) but more varied trajectories for substrate approach (right-side). Substrate is acquired on 4/20 occasions (compared to 7/20 for the *GMC*). On trial one the robot retina is biased, by electric inputs to pan/tilt nodes, towards water resource saccade-fixation but pans to substrate subsequent to gas modulation effects. Figure 5 (right) depicts this initial movement towards the water which then arcs towards the substrate. On trial two, the robot decisively approaches the substrate where the retina remains fixated while the gas dissipates. Regarding *NGMC*, expression of 'opportunism' (trial 1) and 'persistence' (trial 2) is afforded by *active vision*. Such modes of flexible foraging activity have been posited as expressions of motivated behaviour in non-metabolically grounded architectures tested on two-resource problems (cf. Spier 1997, Avíla-García and Cañamero 2004). Opportunism entails ability to "change one's mind" concerning a preference while persistence entails behavioural resistance to alternative motivations. These behaviours accord with McFarland's (2008) *non-reactive* criterion for motivational autonomy. Such flexibility is afforded owing to fast saccade-fixation speed relative to inter-pulse wait time providing an example of how such system level energy constraints may be exploited sensorially given low, or, in the case of the robots here, zero, energy constraints to saccade. In essence, during the waiting period, the robot is able to saccade to the 'desired' resource affording *anticipatory* activity. Regarding *GMC*, the *orientation behaviour*, is more *reactive* – the tight coupling between metabolic and motor activity ensures behaviour is tied to present state (the inter-pulse wait time is not exploited – the retina remains, mostly, static). The comparative metabolic under-determination of sensor-motor activity in *NGMC* behaviour might permit us to label it cognitive (see Barandiaran and Moreno 2008). In spite of its cognitive utility, the emergence of active vision strategies appears to be stifled in the *GMC* condition and to no apparent advantage. This appears to owe to the relative ease of evolution to tap and fine-tune motor orientation-based solutions creating an obstacle for active vision evolutionary transition.

Internal and Sensorimotor Dynamics In order to provide a mechanistic explanation of how metabolism constrains sensorimotor strategies we investigated *sensorimotor* and *internal* dynamics as they affected resource selection. In figure 6 are displayed the evolved topologies for our study. In both cases, multiple gas-emitting e-nodes (grey-circled)

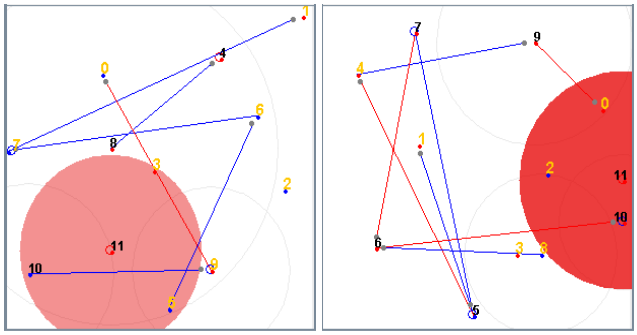


Figure 6: Topology of evolved controllers. Left: *GM* controller. Right: *NGM* controller.

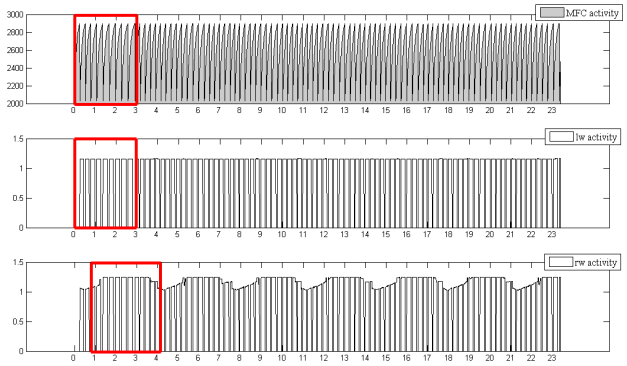


Figure 7: MFC-constrained sensor-motor activity for *GM* controller over 20 trials on a normalized time scale.

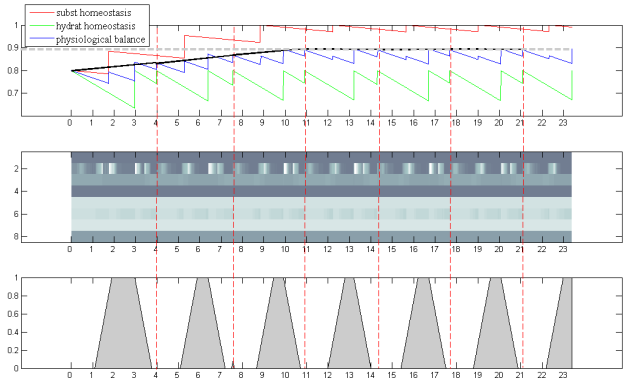


Figure 8: Internal activity for *GM* controller over 20 trials. Top: Physiological (*EV*) balance. Middle: E-GasNet electric activity (4 hidden nodes, 4 e-nodes). Bottom: E-node 11 gas dissipation.

evolved. However, via systematic 'clamping' of gas emission capability it was found that both controllers functionally depended on only a single e-node. The left side (*GMC*) shows that only the right wheel (node 3) is directly affected by gas (e-node 11). Left wheel (1) and pan (0) nodes are only indirectly gaseously affected while the tilt node (2) is affected only by sensory input (indicated by yellow figure colouration). The right side (*NGMC*) shows both pan and tilt nodes within the e-node (11) gas emission radius. This implicates gas as a motor switch mechanism where for *GMC* and *NGMC* the individual e-nodes are sensitive to water and substrate, respectively. The *GMC* was observed on individual re-runs not to use active vision. Figure 7 displays over the 20 trials *GMC* motor activity in [0,0.5] where a constant $C = 0.75$ was added to ensure forward movement (given sufficient MFC-supplied energy). The boxed windows capture a transient phase prior to a more regular periodic dynamic. MFC charge-discharge cycles slow during this period as does left and right wheel pulsed activity. The increased output of the right wheel captured in a time-lagged window reflects slow diffusing gas emission effects consistent with a water resource orienting response. The slow gas dissipation ensures 'commitment' in *GMC* accounting for a water-substrate acquisition ratio of 2:1 – the robot chooses water a second time even after acquisition brings the *EV* value above the e-node gas emission threshold.

Figure 8 displays *GMC internal dynamics* for: *EV*s (top), E-GasNet electric activity (middle), e-node gas output (bottom). Periodic activity for gas output at the e-node arises after the previously described transient phase. Vertical red dashed lines capture windows of resource acquisition dynamics comprised of 3 selections at the 2:1 ratio for water:substrate. The dashed horizontal grey line depicts the stable (mean) *EV* balance and it can be observed with reference to the skewed horizontal black line linking *EV* balance windows that stability occurs after 3 windows. During this period the robot's initial *EV* values become increasingly well regulated therefore. On the other hand, a salient periodic gas emission (and GasNet electric activation) dynamic appears prior to this – after the first window – in accordance with water acquisition dynamics. This happened in spite of the fact that resource distance from the invariant initial position of the robot was varied (to prevent strong sensor-motor dependencies – see Jakobi 1998). The duration of gas emission activity in the e-node observably correlates with the undulating right wheel activity responsible for 'behavioural switching' (fig.7) and dissipates at the point of water resource acquisition. Substrate approach, in the absence of gas effects, is the default behaviour – this is reversed for the *NGMC*. The gaseous 'thirst' signal is affective insofar as it is evolutionarily and metabolically grounded into the agent-environment dynamic and the product of embodied (e-node) anticipatory activity. In sum, the two controller strategies use gas for *EV*-relevant switching from a default resource-orientation response to spatiotemporally-tuned orientation towards the alternative resource. This 'tuning' is critical to sustaining the internal-sensorimotor cohesion of the robot. To better establish the relevance of metabolic grounding to this cohesion we varied inter-pulse wait time (MFC system level constraint) and re-assessed performance of the controllers.

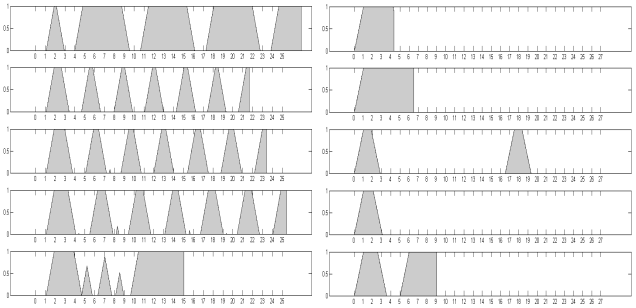


Figure 9: Gas emission as modulated by metabolic constraints over 20 trials according to $bC = \{0, 50, 75, 100, 150\}$ where constraints are represented top to bottom according to ascending strengths. Left: *GM* controller, Right: *NGM* controller.

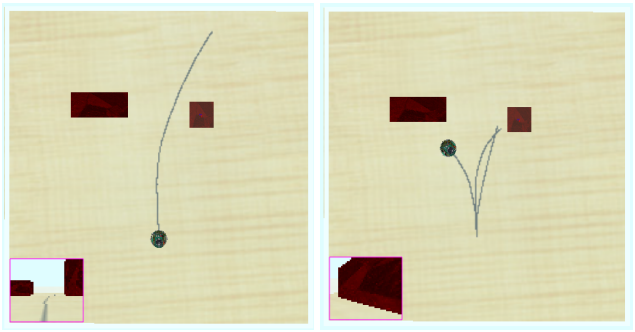


Figure 10: *NGMC* motor trajectories at different metabolic constraints. Left: low *intermediate* – the robot is not viable beyond one trial. Right: high *intermediate* – the robot stops moving (is non-viable) following two successful resource acquisitions.

Dynamic Robustness to MFC constraints The inter-pulse wait time is determined by a constant/parameter bC. All controllers were evolved according to $bC = 75$ steps. This was chosen as an apt challenge level following pre-trial testing. The two evolved controllers in the case study were tested against $bC\epsilon\{0, 50, 75, 100, 150\}$ providing *zero*, *intermediate* (50,75,100) and *high* (150) constraints, respectively. Figure 9 provides gas emission plots over all trials for the two evaluated controllers. It is observed for the *NGMC* (right-side) that only at the constraint value on which it was evolved does it remain viable – robots 'die' at the gas vertical 'cut-off' point and must emit at least twice – perform two switches – over all trials. Interestingly, at *zero* and low *intermediate* constraints the robot fairs badly but performs relatively better at high *intermediate* and *high* constraints. Figure 10 illustrates why this is the case. On the left-side (low *intermediate* constraint), saccade-fixation activity is now insufficiently fast relative to motor speed. The robot behaves 'opportunistically' but receives insufficient retinal stimulation to fixate on the substrate leading to 'dis-orientation'. On the right-side, the high inter-pulse wait time allows saccading to the substrate. This behaviour is more efficient than at the bC value on which the controller was evolved. However, owing to the strong constraint and requirement for regular rehydration the robot soon becomes unviable.

The internal dynamics of the *GMC* (fig.9 – left) are equivalent for all *intermediate* constraints with the same resource choice profile over the 20 trials. Interestingly, at the *zero* constraint the dynamic pattern of gas emissions bifurcates, relative to *intermediate* constraints, early in the trial set. This is an example of robot 'dithering' between the two resources leading to no resource acquisition on trial two which, following the initial transient, periodically recurs. This dynamic is viable but sub-optimal – the robot controller was evolved on $bC = 75$ and whilst robust to relatively minor bC *intermediate* shifts, dynamics are non-robust to extreme shifts of the metabolic constraint. The use of a sub-optimal strategy given a *zero* constraint is viable since the robot only 'dies' following a full trial of non-movement. MFC degradation is not sufficient for this to occur owing to the relatively unchallenging agent-environment dynamic.

In summary, we can say that the challenge level of the environment alone is an insufficient indicator of likely robot viability. It is more informative to consider the robot's spatiotemporal cohesion given internal and sensorimotor domains and evolved metabolic grounding. Specifically, 'dithering' in the *GMC* at *zero* metabolic constraint is an example of maladaptive behavioural performance not present at the evolved constraint. The above highlights the requirement for autonomous robotics architectures to account for metabolic grounding in order to shape adaptive and cognitive (anticipatory) capacities. Affective signals are critical for cohering body-brain dynamics and may be robust to perturbations in agent-environment coupling but are rendered ineffectual if the integration of internal and sensorimotor robotic domains is insufficient.

Discussion

This paper has described work towards an autonomous robotic system focused on the *integration* of energy and motivation autonomous levels as described by McFarland (2008). We suggest that top-down (e.g. ethological) models claiming to implement motivational autonomy in robots are limited as they: 1) ignore how metabolic constraints impact on sensorimotor activity, 2) require a priori environment knowledge. A major application for autonomous robots, however, is in their deployment in inhospitable and unknown environments where harmonious spatiotemporal agent-environment integration is crucial for long-term viability. Our work presents the first steps towards integrating levels of autonomy hinting at the potential for adaptive cognitive behaviour to emerge out of metabolic constraints. We summarize our findings as follows:

1. Two strategies evolutionarily emerged that spatiotemporally integrated metabolic and sensorimotor activity.

2. Strategy one – active vision – enabled robots to exploit the energy-constrained pulsed motor behaviour to produce:
   (a) Sensorial anticipatory behaviour,
   (b) Energy-efficient motor trajectories,
   (c) Adaptive opportunistic-persistent behaviours.
3. Strategy two – motor orientation – did not sensorially exploit its energy-constrained motor-pulsed behaviour.
4. *E-nodes*, via *EV*-level thresholded gas emission, anticipate metabolism constrained performance degradation.

The grounding of behaviour according to artificial metabolic constraints permitted the evolution of sensorial anticipatory behaviour in the form of simple pan/tilt active vision. Interfacing 'body' (MFC) and 'brain' (*E-GasNet*) entailed tuning gas emissions to enable this anticipatory sensorimotor response. Stable gas emission dynamics in functional nodes when metabolically situated constitutes motivation-like (thirst/hunger) signals. The existence per se of such signals precipitates orientation/saccade switching and is functional therein. The periodicity and duration of such signals are requisite to the agent-environment dynamic niche and functional therein. A significant change to this dynamic, e.g. severe modification of the metabolic constraint, renders the motivation-like signals non-adaptive even if the task challenge is effectively reduced.

We are currently investigating 'naturalistic' settings with dynamic resource configurations. Early findings hint at the emergence of distributed forms of e-node networks adapted to this more complex dynamic. A long term aim is to unveil robot controllers that exhibit *energy-motivation-mental autonomy* (see Ziemke and Lowe 2009) described using utility- and optimality-based ecological models.

Acknowledgements

This work has been supported by a 2006-2009 EC grant to the FP6 project "Integrating Cognition, Emotion and Autonomy" (ICEA, IST-027819, www.iceaproject.eu).

References

Ashby, W.R. (1960). *Design for a brain: The origin of adaptive behaviour*. Chapman and Hall.

Avíla-García, O. and Cañamero, L. (2004). Using hormonal feedback to modulate action selection in a competitive scenario. In *From Animal to Animats 8: Proc. of the 8th International Conference on Simulation of Adaptive Behaviour*. Cambridge, MA: MIT Press, 243-252.

Barandiaran, X. and Moreno, A. (2008). Adaptivity: From Metabolism to Behavior. *Adaptive Behavior*. 16: 325-344

Canabal, D.D., Song, Z., Potian, J.G., Beuve, A., McArdle, J.J. and Routh. V.H. (2007). Glucose, insulin, and leptin signaling pathways modulate nitric oxide synthesis in glucose-inhibited neurons in the ventromedial hypothalamus. *Am. J. Physiol. Regul. Integr. Comp. Physiol.* 292:1418-1428.

Cannon, W.B. (1915). *Bodily Changes in Pain, Hunger, Fear and Rage*. Appleton, New York.

Floreano, D., Kato, T., Marocco, D. and Sauser, E. (2004). Coevolution of Active Vision and Feature Selection. *Biological Cybernetics*. 90(3):218-228.

Husbands, P., Smith, T., Jakobi, N. and O'Shea, M. (1998). Better Living through Chemistry: Evolving GasNets for Robot Control. *Connection Science*. 10(3/4):185-210.

Ieropoulos, I., Melhuish, C. and Greenman, J. (2007). Artificial gills for robots: Mfc behaviour in water. *Bioinspiration and Biomimetics*. 2, S83-S93.

Jakobi, N. (1998). Minimal Simulations for Evolutionary Robotics. *DPhil thesis*, University of Sussex.

McFarland, D. (2008). *Guilty Robots, Happy Dogs*. Oxford University Press.

McFarland, D. and Spier, E. (1997). Basic cycles, utility and opportunism in self-sufficient robots. *Robotics and Autonomous Systems*. 20:179-190.

McHale, G. and Husbands, P. (2006). Incorporating Energy Expenditure into Evolutionary Robotics Fitness Measures. In L. Rocha et al. (Eds), *Proc. Alife X*, 206-212, MIT Press.

Melhuish, C., Ieropoulos, I., Greenman, J. and Horsfield, I. (2006). Energetically autonomous robots: food for thought. *Autonomous Robots*. 21:187-198.

Montebelli, A., Ieropoulos, I., Lowe, R., Ziemke, T., Melhuish, C. and Greenman, J. (2010a). Unplugged! A mathematical model of Microbial Fuel Cells for energetically self-sustainable simulated robotic agents. In preparation.

Montebelli, A., Lowe, R., Ieropoulos, I., Melhuish, C. Greenman, J., and Ziemke, T. (2010b). Exploring prospective hybrid life: Microbial Fuel Cell driven behavioral dynamics in robot simulations. *Artificial Life 12*, in press.

Morley, J.E., Alsaher, M.M., Farr, S.A., Flood, J.F. and Kumar, V.B. (1999). Leptin and neuropeptide Y (NPY) modulate nitric oxide synthase: further evidence for a role of nitric oxide in feeding. *Peptides*. 20:595-600.

Parisi, D. (2004). Internal robotics, *Conn. Sci.*, 16(4):325-38.

Smith, T., Husbands, P., Philippides, A. and O'Shea, M. (2002). Neuronal Plasticity and Temporal Adaptivity: GasNet Robot Control Networks. *Adaptive Behavior*. 10(3-4):161-183.

Spier, E. (1997). *From Reactive Behaviour to Adaptive Behaviour*. PhD Thesis. University of Sussex.

Vargas, P., Moioli, R.C., von Zuben, F.J. and Husbands, P. (2009). Homeostasis and evolution together dealing with novelties and managing disruptions. *International Journal of Intelligent Computing and Cybernetics*. 2(3):435-454.

Yao, S.T., Gouraud, S., Paton, J.F.R. and Murphy, D. (2005). Water Deprivation Increases the Expression of Neuronal Nitric Oxide Synthase (nNOS) but Not Orexin-A in the Lateral Hypothalamic Area of the Rat. *The Journal of Comparative Neurology*. 490:180-183.

Ziemke, T. and Lowe, R. (2009). On the Role of Emotion in Embodied Cognitive Architectures: From Organisms to Robots. *Cogn. Comp.*,1:104-117.

EcoBot-III: a robot with guts

Ioannis Ieropoulos[1], John Greenman[2], Chris Melhuish[1] and Ian Horsfield[1]

[1]Bristol Robotics Laboratory, University of the West of England, Bristol Business Park, Coldharbour Lane, Bristol, BS16 1QD, UK
[2]School of Life Sciences, Faculty of Health & Life Sciences, Frenchay Campus, University of the West of England, Coldharbour Lane, Bristol, BS16 1QY, UK
ioannis.ieropoulos@brl.ac.uk

Abstract

This paper describes the work carried out to develop EcoBot-III, which is a robot with an artificial digestion system. The robot is powered by Microbial Fuel Cells (MFCs) and it is designed to collect food and water from the environment, digest the collected food and at the end of the digestion cycle, egest the waste. EcoBot-III operated successfully for 7 days when fed with anaerobic or pasteurized sludge, before mechanical failure required human intervention. Work is ongoing to improve the mechanics and thus extend the artificial agent's operational lifetime.

Introduction

Autonomous behavior for artificial agents implies prolonged operational periods with minimum or no human intervention. This is important (and can also be considered as vital) for a variety of tasks/missions, generally categorized under 'remote area access'. Up until recently autonomous robotic behavior, was primarily seen as a computational challenge, where robots are developed with processing skills that allow action selection and decision making, but with the element of energy and energy collection taken for granted. Work by numerous groups has indicated that true autonomy needs to take into account the collection of energy from the environment (akin to biological agents) and build it in the robot's behavioral repertoire (McFarland, 1990; Steels and Brooks, 1995; McFarland and Spier, 1997; Spier and McFarland, 1997). Thus, over the recent years, energetic autonomy has received increased attention from the robotics community as a vital feature for autonomy and self-sustainability (Spier and McFarland, 1996; 1998; Melhuish and Kubo 2004; Ziemke 2008; Kubo et al. 2009). The robot pioneers Gastrobot, Slugbot and EcoBots have demonstrated how this notion may be realized, through the integration with real microorganisms living inside Microbial Fuel Cells (MFCs) (Gastrobot, EcoBots) and the collection of real food from the environment (Slugbot) (Kelly et al. 2000; Wilkinson, 2000; Greenman et al. 2003; Ieropoulos et al. 2003; Melhuish et al. 2006). This integration between biology and machines has been described as (artificially) symbiotic and has resulted in the introduction of a new class or robots known as Symbots (Melhuish et al. 2006).

The present study addressed the twin issues of energy autonomy and bio-regulation. Biologically inspired mechanisms and strategies were explored, to provide full energy autonomy to a new robot that *produced its own energy* from biological material (e.g. plant or insect material) which it collects and processes using MFCs. The work focused on the construction of a complete MFC-based self-regulating energy system which necessitated exploring mechanisms for (1) collecting, ingesting (eating) new substrate (2) removing waste material (3) maintaining internal homeostasis and (4) performing appropriate behavior for the foraging/ acquisition of food.

The work described in this paper, builds on EcoBots I and II and had the following main aims: **(i)** To build the individual prototype mechanisms for ingestion for EcoBot's artificial gut using MFC technology; **(ii)** To develop embedded low-power controllers capable of sensing and on-board actuation to maintain internal homeostasis; **(iii)** To design and build a novel egestion mechanism to allow the evacuation of waste material from both the MFCs and the digester unit; **(iv)** To design and build a system with which it will be possible for the robot to collect liquid food and water from the floor or wall of an arena (EcoWorld arena); **(v)** To integrate all components and systems to demonstrate self-sustainable operation of EcoBot-III. This demonstration will include ingestion of fresh food source, digestion and egestion of waste material in order to continue performing its assigned tasks.

The following sections describe the development of EcoBot-III - the third in a series of self-sustainable agents – with an artificial digestion system that collects its energy from the environment and 'lives' on microbial metabolism.

Materials and Methods

In the first phase of the study, the work focussed on the design and testing of engineered prototypes of sub assemblies for power production (MFC stacks), artificial gut circulation, food ingestion and their integration into a work bench demonstrator. The ingestion system needed to supply the anodic chambers with an organic substrate (food). It had to maintain appropriate separation between the stomach-like collecting pouch and the anodic chamber. Early experiments explored the possibility of designing a system that attracts insects (flies) using pheromone bait and traps the flies in a fluid reservoir. Later experiments focussed on using

alternative feed substrates (broths and pure substrates), which the robot accessed from a wall-mounted feedstock reservoir.

A biologically-inspired controller for homeostasis was also prototyped. This was used to model, in control-theoretic form, the biological negative feedback loops typical of regulatory mechanisms for homeostasis. Of particular importance to EcoBot, given its continuously low energy levels, was a model of the regulation of energy intake that takes into account the modulation of this system by internal temporal cycles for ingestion. The controller is generalized to regulate the internal parameters of the robot with electronic sensor boards for temperature and fluid levels (with option for pH or other sensor systems if they possess low power requirements).

Microbial Fuel Cells

MFCs are bioelectrochemical transducers that convert biochemical energy (generated by microbes) directly into electricity. They consist of two half-cells; an anode, which is the bacterial side and has negative polarity (electron generating) and a cathode, which is the oxidizing side and has positive polarity (electron accepting) and the two are separated by an proton selective membrane (PEM) (figure 1). Microbes in the anode chamber can be in either planktonic (suspended in liquid solution) and/or biofilm forms (attached to the electrode surface) and transfer electrons to the electrode either via electroactive metabolites naturally released by the microbes or direct conduction, via conductive pili (nanowires).

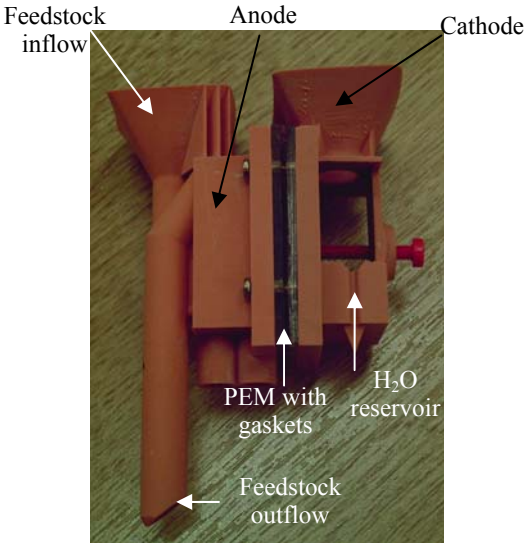


Figure 1: Photo of the terracotta colored (Nanocure® photo-polymer) final assembly of a MFC; labels show the various parts and features of the design. Inside the anode and cathode chambers (not shown in photo) are the carbon veil electrodes (67.5cm^2 total surface area for each electrode).

MFCs are a new technology, in the sense that only now can they produce sufficient power to make them drive useful applications. The open circuit voltage and maximum sustainable power output of a single MFC is approximately 0.7V and 50μW respectively, suggesting that a plurality of MFCs will be required to drive an application such as EcoBot. A related question is "can stacks of MFCs produce enough energy at a fast enough rate to drive a physical entity that could move and support the weight of its own energy generating system (MFC stacks, stomach, tubes, electronics, accumulators, motors and pumps). The weight onboard the robot had to be as low as possible and all actuators, motors and pumps had to function at the lowest possible power consumption. Earlier findings demonstrated that power density improves with decreasing size of individual MFCs (Ieropoulos et al. 2008). This formed the basis for EcoBot's final design.

A total of 48 MFCs were employed onboard EcoBot-III and they were configured in a circular fashion (figure 2). This was in order for the open-to-air oxygen-diffusion cathodes to be facing outwards in order to maximize oxygen (from free air) exposure. The 48 units were stacked in 2 tiers so that overflowing liquids (feedstock from the anodes and water from the cathodes) from the top tier could fall directly into the corresponding MFC units in the bottom tier.

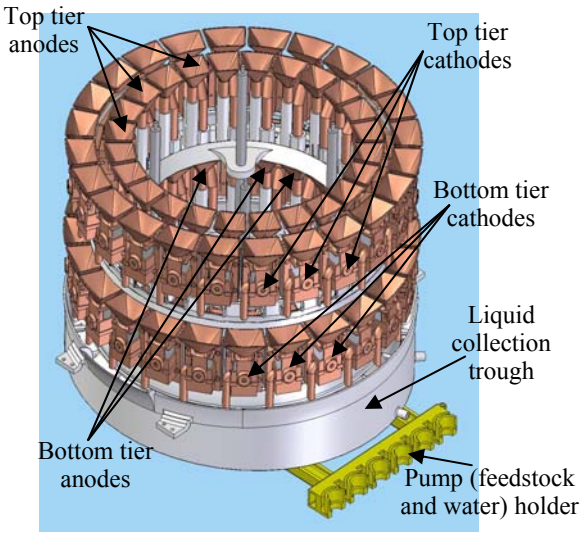


Figure 2: CAD snapshot of the MFC stack onboard the middle part of the EcoBot-III chassis.

Isolated liquid (feedstock and H_2O) distribution

When connected in stacks**,** MFCs behave like batteries and are thus prone to 'shorting' and system failure if brought in fluidic contact. This may be the result of (i) feeding multiple units from a common feedstock bottle, (ii) feeding one MFC unit directly from another in continuous flow or even (iii) if the structural material of MFCs is hygroscopic. This is particularly relevant when there are elements of the MFC network in series. Series connection is a pre-requisite since single units or units in parallel do not produce enough voltage (at max sustainable power) to drive electronic modules nor charge up accumulators. Energy at a voltage below 500mV is insufficient to be usefully harvested. It was therefore necessary to build-in to the EcoBot-III design a method of breaking this fluidic linkage and allowing the isolation between all functional units of the robot, whilst still being fed and/or hydrated from common sources. The problems of

common feeding have been previously identified (Ieropoulos et al. 2008). This was the main idea behind the introduction of a 'carousel' feeding mechanism, which distributes food and water in a sequential-isolated manner (see figure 4), which also alleviates the problems arising from feeding the bottom MFCs directly from the ones above.

Fluids (substrate feedstock to anodic chamber; water to the cathodes) had to be circulated on board the robot, with all the attendant challenges of "wet engineering". This meant that the overflowing fluids from the MFC stack were collected in a trough (see figure 3) and periodically recycled back into their respective reservoirs (food into stomach; water into distribution nozzle).

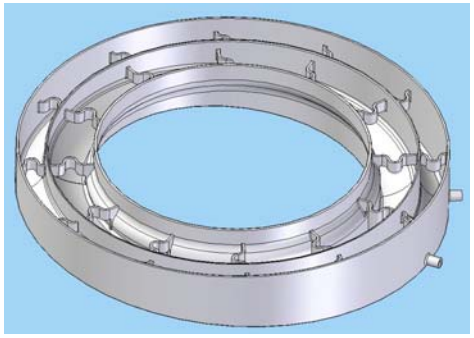


Figure 3: CAD image of the fluidic collection trough – inner channel (food), outer channel (water). Return ports are shown on the side. N.B. This is the bottom part of the image in figure 2.

Carousel feeding/distribution mechanism

As mentioned before, a sequential distributor was built-in to the EcoBot. This was a carousel-like mechanism which was motor-driven to increment its state by one position at a time so that all the MFCs can be fed and watered in an isolated manner (figure 4).

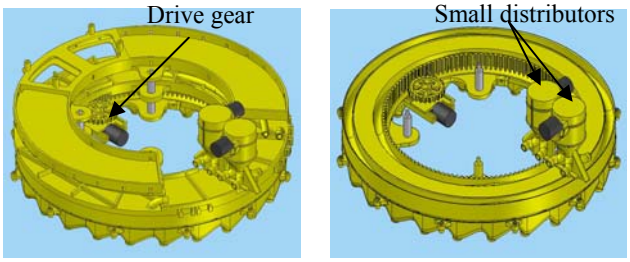


Figure 4: (Left) CAD snapshots of the carousel feeding mechanism; (right) the complete carousel feeding mechanism uncovered. Outside channel is for water and inside for feedstock. Funnels at the bottom of the part are the inlet nozzles for each MFC unit.

The carousel unit has additional smaller motor-driven distributors in order for food and water to be distributed over 4 outlet ports – in essence feeding 4 quartiles at the same time. The amount of fluid flowing per feed and water dose was intentionally superfluous so that the 4 MFCs on the top tier would overflow into the corresponding 4 MFCs on the bottom tier, during each feed or hydration.

Ingestion, digestion (stomach), fly-trapping and egestion of waste

One of the main objectives of this study was the design and development of mechanism(s) to allow the intake and processing of food and evacuation of the waste products e.g. recalcitrant and inorganic matter. To this effect a digestion unit was designed (figure 5) which incorporated a conical hat with added features (UV light, pheromone pocket, and liquid collection lip) to allow the ingestion of either liquid food or flies. In addition, the bottom part of this digestion unit was designed to allow the sedimentation of heavy-weight particles and was connected to a peristaltic pump, which allows the excretion of this material, in an effort to rid the microflora in this digestion unit from the accumulation of poisonous waste by-products, e.g. acid waste.

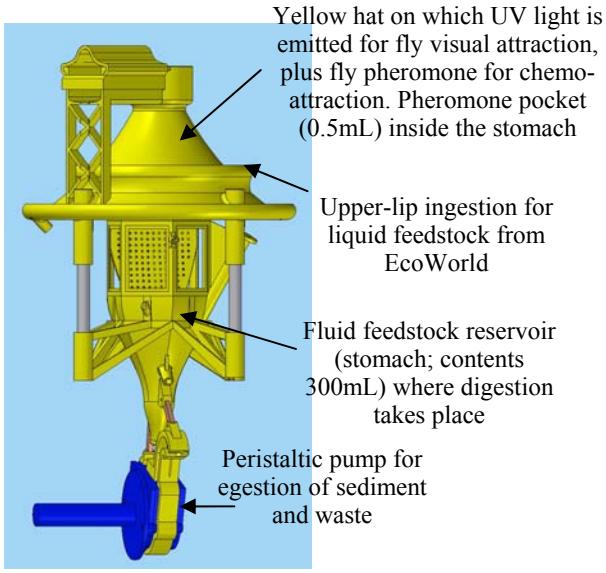


Figure 5: CAD image of the stomach unit with the ingestion, digestion and egestion features. The underside of the conical hat (not shown) is black and the stomach has transparent windows to ensure that flies remain trapped.

EcoBot-III was designed to operate on two feeding strategies; one attracting insects (flies) using pheromone bait and UV-light (as a visual stimulus), in order to lure and trap the flies in a fluid reservoir and the other collecting liquid food supply (complex broth or pure substrates) from a feeder mechanism from the side-wall of the test bed arena (see below). Visual attraction is by UV light LED's flashing periodically on the yellow surface of the stomach hat and chemical attraction is by using the fly sex pheromone Z-9 tricosene – only as a primer.

Onboard accumulator

However good the stacks of MFCs may be, power is still insufficient to run all actuators simultaneously and continuously. Energy storage, action selection processing and pulsed behavior patterns must be embedded. This was the core of the electronic circuitry which employed a capacitor bank acting as the energy accumulator.

Initially 0.408F capacitance was used (60 x 6800µF electrolytic capacitors 6.3V), which subsequently doubled to 0.816F (120 x 6800µF electrolytic capacitors 6.3V). The voltage operating range (V_{dis} = 2.96V; V_{ch} = 1.9V), was dictated by the symmetry around the intersection point between the actual capacitor charge curve and its first derivative.

Control architecture overview

Figure 6 below illustrates the actual embedded ultra-low power microcontrollers, *in situ*, for sensing and on-board actuation to maintain homeostasis. The main list of components is: microcontroller board (PIC46F20); startup isolator; 3.3V and 5V PSU board with onboard comparator; input board; output board; H-bridge board; level sensor board; pump driver board; photo eye boards; UV LED driver board.

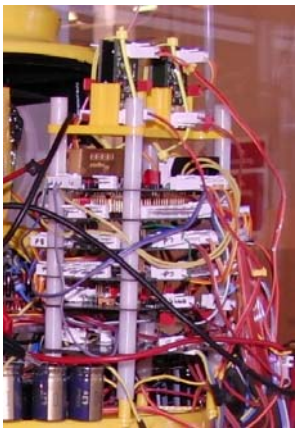


Figure 6: Control hardware onboard EcoBot-III, connected and running

EcoWorld (the robot arena)

The arena was constructed out of transparent Perspex and contained the robotic track and the water and liquid feedstock distribution mechanisms (figure 7).

The internal temperature was controlled by thermostatic fan heater to maintain the temperature at 30 ± 5 °C. The dimensions were 70cm x 100cm (floor area) x 67 cm height. Two microprocessor controlled feedstock distribution mechanisms (one for liquid nutrient, one for water) were designed and built, each with radio connectivity. The system distributes a fixed fluid volume on to the robot in response to the robot making contact with the micro switch.

Figure 7: EcoWorld finished with EcoBot-III on its robotic track inside. The external (arena) microcontroller is shown on the top, with water and liquid feedstock bottles shown on the left and right, respectively.

EcoBot-III

The final prototype EcoBot-III is shown in Figure 8. This is the resultant platform that integrates all the aforementioned functional units. The robot has the following physical characteristics: height, 63cm; diameter (outer), 29cm; weight (with full stomach, MFCs and trough), 5.88kg.

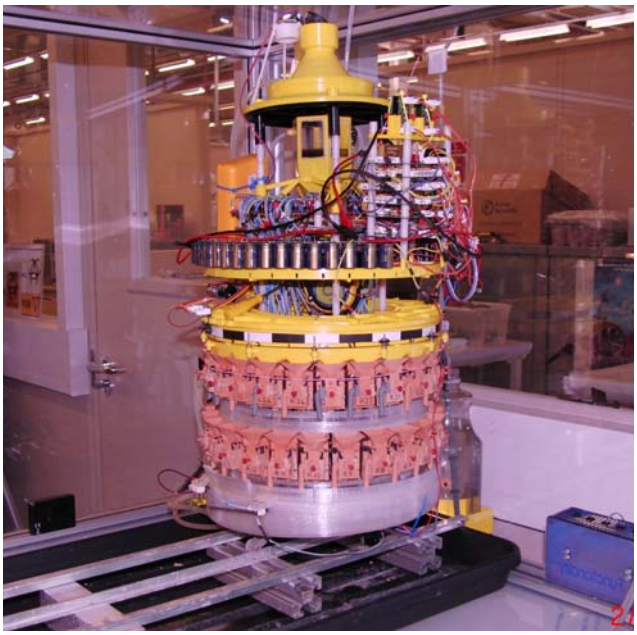


Figure 8: EcoBot-III in its final state and in the EcoWorld. The whole robot is made from 3 different rapid prototype materials: Nanocure® resin for the MFCs, yellow ABS for the more intricate parts due to its soluble scaffolding and polycarbonate (ISO) for the more 'heavy duty' parts.

As previously mentioned, EcoBot-III has been constructed in such a way, that there is only one waste evacuation mechanism. Microbial Fuel Cells have been developed with a continuous flow design, by which excess fluids (useful and useless) overflow to the outside and below. The current EcoBot consists of 2 tiers of MFCs. Fluid flows from the header tank (digester) into the MFCs of the first floor, which when full (6mL total volume) overflows directly into the MFCs of the level below. Overflow from the bottom MFC tier is collected into a trough, which loops back into the header tank, thereby allowing the re-circulation (and hence further utilization) of useful 'waste' that has overflowed from the MFCs. Eventually, undigested or indigestible waste will accumulate inside the digester unit, which has been designed with a central port for evacuation. This is located at the bottom of the digester, so that heavy weight particulates can settle. A heavy duty peristaltic pump has been modified and fitted at the bottom of the header tank, so that it can be periodically actuated to allow some of this semi-solid waste material to evacuate the digester in the form of a pellet. The solid (or semi-solid) waste evacuation is at the moment performed on a time basis (once every 24hrs). The semi-solid stomach contents may be periodically agitated (not part of the current design), using a high-speed dc motor with a flexible long shaft to bring solids into suspension and allow their re-distribution through the MFC network.

Results

EcoBot-III is designed to collect and utilize flies, however experiments in which live flies are introduced into the robot's arena (EcoWorld), in order to evaluate its autonomous behavior based on only 'insect-diet' are ongoing and have not been completed. The data presented herewith, are from the experiments in which EcoBot was manually fed with fly-juice (sludge that had been fed with flies) and also in which EcoBot successfully collected pasteurized sludge (artificial wastewater) from its environment.

Fly attraction

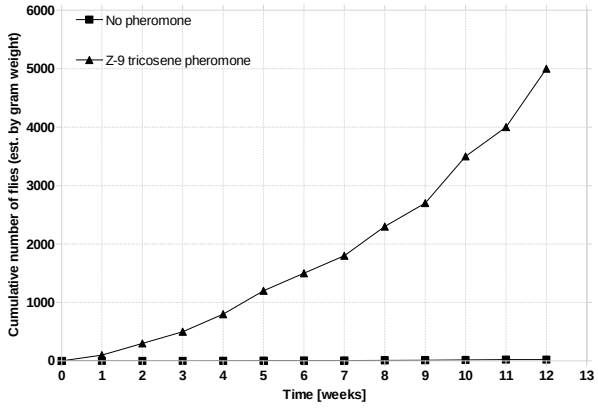


Figure 9: Comparison between fly-traps working with the chemo-attractant pheromone (triangle symbols) and without (square symbols; control).

Although live flies were not introduced in EcoWorld, the effectiveness of the Z-9 tricosene pheromone against a control was still of interest, since the stomach of EcoBot-III is designed to accommodate a small volume (0.5mL separate pocket inside a 300mL digester) of this chemical as a primer. Experiments using conventional fly-traps with the Z-9-tricosene pheromone (28mL in 2L) and without (control) have shown a remarkable difference (figure 9).

EcoBot-III telemetry data

EcoBot-III is designed to communicate with a basestation for reporting data such as time stamping, voltage of the onboard accumulator, task identity, fluid level status for the stomach and trough and also origin and destination in the arena. A snapshot of the telemetry data received from the real EcoBot-III experiments is shown below in figure 10.

```
0:7:53:11,E,52,LF,2.963,11010101
0:7:53:15,OFF,1.951,11010101
0:8:47:56,E,54,LF,2.950,11010101
0:8:48:0,OFF,1.932,11010111
0:9:41:42,E,54,LF,2.966,11010111
0:9:41:46,OFF,1.935,11010111
```

Figure 10: Exemplar of a string of telemetry data received from EcoBot-III when running in EcoWorld. In this particular example, the robot is moving towards the left feedstock distribution (looking at the arena from the front), and it is actuating every 54 minutes.

The incoming data (red boxed transmission) can be interpreted as follows (from left to right): **Days**: **Hours**: **Minutes**: **Seconds**, **Energy actuation** (as opposed to timer triggered actuation), **Time between actuations**, **Task** identification, **Capacitor Voltage**.

Binary data string (MSB→LSB): Arena right feedstock and H_2O distribution (1 = not there yet); Arena left feedstock and H_2O distribution (1 = not there yet); Stomach **low** fluid level (1=full, 0=empty); Stomach **high** fluid level (1=full, 0=empty); Trough feedstock **low** fluid level (1=full, 0=empty); Trough feedstock **high** fluid level (1=full, 0=empty); Trough H_2O **low** level (1=full, 0=empty); Trough H_2O **high** level (1=full, 0=empty).

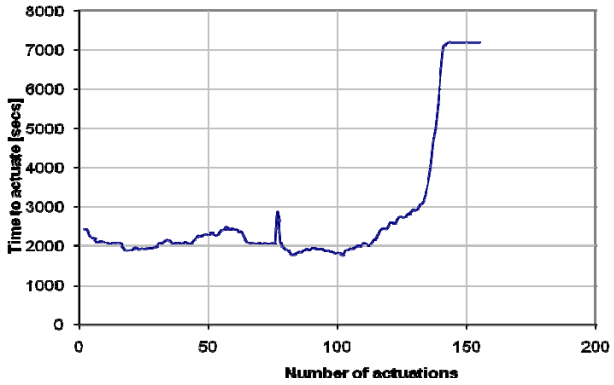


Figure 11: Time to fire vs. number of actuations for feedstock distribution via carousel mechanism. EcoBot-III operating for 5 days, feeding on anaerobic sludge that had been given dead flies.

Data from experiments completed using EcoBot-III are shown below in figure 11. This is the processed version of the telemetry data received from EcoBot during a 7-day experiment, when EcoBot was feeding on flies (>10 in 300mL of stomach contents). The data show that the robot was actuating (feeding the MFCs) every approx 30 minutes, until a mechanical failure occurred at the 111th actuation, at which point the time to fire increases exponentially.

EcoBot-III has a defense mechanism, by which it triggers actuation using a timer (after 2 hours) if during this period energy has not accumulated to the pre-set threshold at the correct rate (flat line at the end of the curve). All other actuations have been filtered out to show only those related with feeding – this could have been done for any of the actuations. In reality the total number of actuations (including hydration) was twice as many (309 firings) as shown in figure 12.

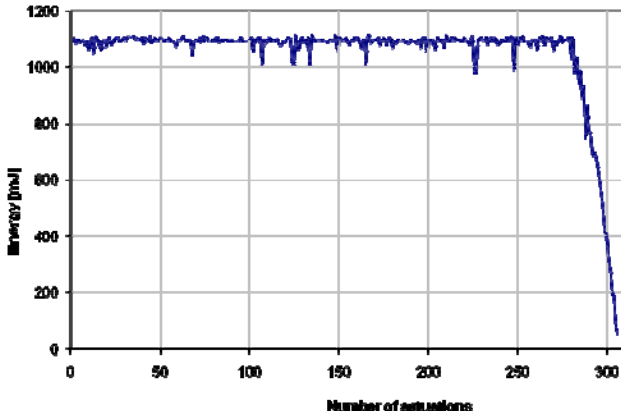


Figure 12: Total energy generated per actuation

Liquid feedstock (synthetic wastewater with 20mM sodium acetate)

In this experiment, EcoBot is employing the second feeding strategy, which is utilizing liquid food from the arena wall.

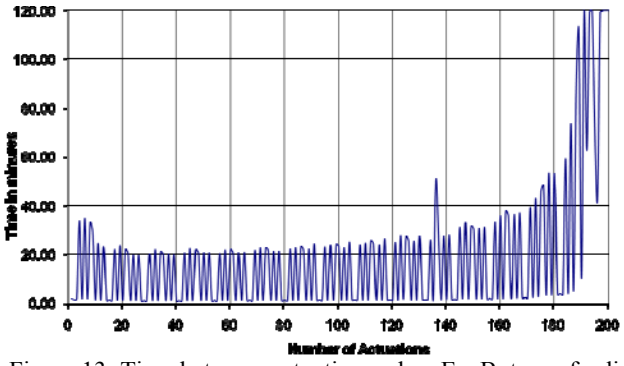


Figure 13: Time between actuations when EcoBot was feeding from the arena.

The liquid feedstock was artificial wastewater consisting of nutrients, minerals and carbon energy source (20mM acetate), but was deprived of any microbial growth that is found naturally in wastewater. This was in order to ensure that the energy is coming from this feedstock and not from exogenous (and newly introduced) microbes.

Figure 13 below shows the relationship between the number of actuations and the time (in minutes) it took for each actuation to fire.

As can be seen from the graph above, the time varies depending on the actuation, since different actuations use different amounts of energy and therefore take longer (or not) to occur. The increase in time between actuations is an indication that EcoBot is slowing down (MFC exhaustion; possible blockage; feedstock leakage due to blockage shorting MFCs out). The distribution of energy for each actuation is shown below in Figure 14.

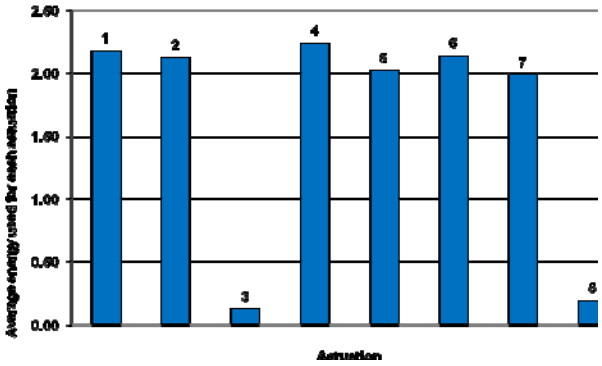
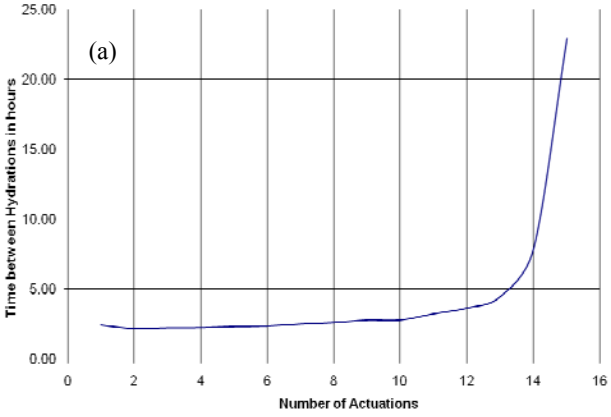


Figure 14: Energy usage per actuation; the numbered actuations are as follows: 1) water distribution (hydration) of cathodes; 2) feedstock distribution (feeding) of microbial anodes; 3) carousel indexing one position; 4) feedstock recycling into the stomach; 5) locomotion; 6) egestion; 7) UV light attractant; 8) single UV flash before each actuation.

As an exemplar of all actuations, onboard water distribution to the cathodes was further analyzed, as shown below in Figure 15.



The data in Figure 15, show a stable behaviour in terms of this particular actuation for the vast majority of hydration cycles, up until the point that the performance begins to slow down, at which point the time between actuations increases exponentially. Equally, the energy spent per hydration cycle is stable within ±10%, up until the system performance deteriorates. When EcoBot operates correctly, then the graphs for all actuations are constant.

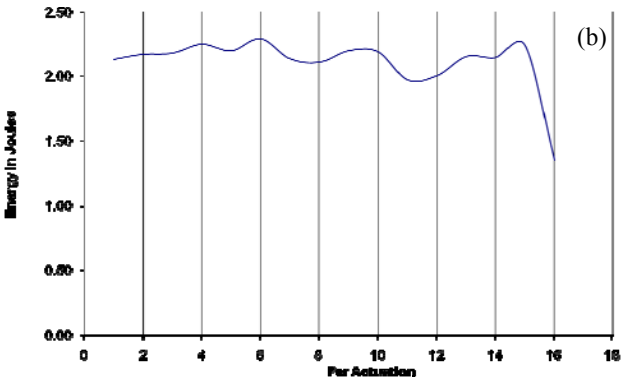


Figure 15: (a) Water distribution to MFC cathodes; (left) time between water distribution in hours; (b) average energy per water distribution actuation.

On this particular occasion, the EcoBot performance deteriorated due to the fact that the robot was dehydrated and did not make it to the water distribution mechanism on the side wall of the arena. The experiment started (intentionally) with an empty onboard water trough reservoir but with fully moistened MFC cathodes, to investigate whether it would make it to the water point. In addition, extra actuations were introduced (UV single flash before every actuation) and waste evacuation at the end of each actuation sequence. In reality, waste evacuation takes place only once in a day and there is no UV single flash before each actuation. These experiments are currently ongoing.

Discussion

Developments in energy-autonomous robots using microbial fuel cells (MFC) can be expected to be attractive to industry in two areas. Firstly, the MFC technology itself may eventually reach a development stage where it produces comparable energy densities with those of 'domestic' batteries and therefore provide an alternative, carbon-neutral, power source. This could lead to stand-alone appliances such as sensors, alarms, telecommunications, low energy lights, small pumps or actuators, small motorized systems (fans, robots) and trickle chargers for charging car batteries. Possibly the technology could be scaled sufficiently to generate energy from large 'reservoirs' of biomass such as those found in sewage treatment works. These fuel cells can also utilize waste products (such as acetate) from current fuel cells which are being employed to generate hydrogen thus improving the overall efficiency.

Autonomous robots powered from MFCs will have a wide range of applications and will be attractive to industry. The finding that MFCs can utilize waste (sludge) suggests that the technology can be considered as a useful novel method for tertiary wastewater treatment. Regarding their application into Symbots (i.e. EcoBot) provided their energy supply is sufficient for them to function and carry out their tasks, it may not matter that they are neither the most efficient nor the quickest; sufficient is all that matters. Therefore, it is easy to envisage energetically autonomous robots employed for monitoring of farm land and crops, sewers and also for marine exploration in non-sunlit waters.

Energy autonomy. It is clear from our work that as long as EcoBot is performing correctly within its working environment and is provided with food and water via the arena (EcoWorld), it continues to function well. It can gain sufficient electrical energy from organic food to continue motion on its track, to collect water and food when needed and distribute these to the MFCs. It has sufficient energy on board to also perform other exemplar tasks such as elimination of non-digestible components by controlled ejection of "waste", sensing (of temperature and light), data processing and radio transmission of logged data.

Bio-regulation. When mixed-culture "ecologies" are transplanted into EcoBot, they consist of a wide diversity of different groups and species of microorganism. Further groups of microbes may also be introduced, depending on the nature and source of the food – e.g. rotten fruits and vegetables and sludge carry with them their own microbes (essentially responsible for the rotting). The physicochemical environment within EcoBot (albeit different to the microbe's original natural environment) is nevertheless a suitably selective environment for the more robust microbes' survival and growth. The microbial community that finally adapts to this system, will still be sufficiently diverse to function. Clearly, some species that do not like the prevailing environment will diminish in population number (be selected against) whilst others that can adapt will be enriched. Electroactive species of microbe appear to be enriched as biofilms around the anodic electrodes. Within the stomach-digester (artificial gut) the main types of species (in a low dissolved oxygen environment) are likely to be strict and facultative anaerobes, and the main pathways by which they will gain energy will be via fermentation. Polymeric food molecules (starch, chitin, proteins, saccharides) are hydrolysed by microbial enzymes to give monomeric molecules that can be taken up by the cells. Fermentation produces organic acids as the main end-products of metabolism, including acetate, propionate, butyrate, lactate, formate, alcohols and carbon dioxide. The acids produced would normally be expected to reduce the pH. The organic acids (e.g. acetate) are circulated to the MFC units where electrogenic species utilise them by oxidation, through the abstraction of electrons (via the electrode) and producing carbon dioxide and more protons. However, the build-up of acids (and resulting low pH) does not appear to occur, possibly because of one or more of the following reasons: (i) the anaerobic sludge microbes forming into robust and stable biofilms, naturally buffering their surroundings (concomitant production of ammonia and other basic molecules at a rate which neutralises the pH); (ii) loss of acids through volatilization; (iii) effective removal of protons by the MFC cathodic system (PEM and cathode).

The latter mechanism appears to be the most important and the system maintains pH homeostasis throughout continuous operation. Alternative designs of cathode employ closed chambers with either chemical electrolytes, fast running water or aerated water. All these systems require high amounts of energy to remain operational and help catalyse the reaction: $O_2 + 4e^- + 4H^+ \leftrightarrow 2H_2O$ [+0.82]. In the cases where the chemical electrolyte is fully reduced, or the water/air stops

flowing, then the cathodic system no longer acts as the oxidising half-cell, and the H^+ ions generated in the anode (cations) cannot find their electrochemical path through to the cathode, thus accumulating to lethal levels for the microbes. The open to the air/periodically moistened cathode, might not be as efficient as the aforementioned alternatives at the initial stages of the MFC lifetime, however it continuously improves with time and eventually outperforms all other systems, especially in terms of longevity. It would be interesting to see (as part of future work) what happens if the robot is fed acid or alkaline mixtures of feedstock, or whether acid build-up does occur when the MFC are electrically disconnected.

Nutrient acquisition behavior. In the programming of EcoBot, nutrient acquisition is triggered by contact with the feed and water distribution mechanisms of the arena, at which point the behaviour changes so that the robot feeds and hydrates all MFCs, before it moves away to do other functions. Provision for different behavior patterns has been made so that the robot can move towards the feed/water distribution points when fluid levels are low or indeed when energy levels from the MFCs are low. This is what we would term as 'hunger' simulation.

Conclusions

As the development of MFCs continues (using smaller units which make for more powerful stacks), then the ability to utilise MFC-stacks on board robots will become more attractive and commonplace. This study shows the feasibility of the Symbot approach, albeit being far from fulfilled. It may not be a perfect system and still a proof-of-concept prototype, however, it is the authors' conclusion that EcoBot-III demonstrated energy autonomy, when fed with nutrient rich liquid feedstocks and within the boundaries of its environment.

To the best of the authors' knowledge, this is the first example of a robot, which integrates real life and machine in a symbiotic manner (Symbot) for digestion and autonomous operation as an exemplar of artificial life.

Acknowledgments

This work was funded by the European Union's Framework Programme 6 (FP-6) Scheme, grant no. IST 027819 ICEA.

References

McFarland, D. (1990). What it means for Robot Behaviour to be Adaptive. In From Animals to Animats 1, of the *Simulation of Adaptive Behaviour (SAB'90)*, pages 22-28, MIT Press, Cambridge. MA.

Steels, L. and Brooks, R. A. editors (1995), *The Artificial Life Route to Artificial Intelligence: Building Embodied Situated Agents*, Lawrence Erlbaum Associates, Hillsdale, NJ.

McFarland, D. and Spier, E. (1997). Basic cycles, utility and opportunism in self-sufficient robots. *Robotics and Autonomous Systems*, 20:179-190.

Spier, E. and McFarland, D. (1997). Possibly Optimal Decision Making Under Self-sufficiency and Autonomy. *Journal of Theoretical Biology*, 189:317-331.

Spier, E. and McFarland, D. (1996). A Finer-Grained Motivational Model of Behaviour Sequencing. In From Animals to Animats 4, of the *Simulation of Adaptive Behaviour (SAB'96)*, pages 255-63, MIT Press, Cambridge, MA.

Spier, E. and McFarland, D. (1998). Learning to do without cognition. In From Animals to Animats 5, of the *Simulation of Adaptive Behaviour (SAB'98)*, pages 38-47, MIT Press, Cambridge, MA.

Melhuish, C. and Kubo, M. (2004). Collective Energy Distribution: Maintaining the Energy balance in Distributed Autonomous Robots. In the proceedings of *7th International Symposium on Distributed Autonomous Robotic Systems*, Toulouse, France, pages 261-270.

Kubo, M., Sato, H., Matsubara, T. and Melhuish, C. (2009). High survivability of a large colony through a small-world relationship. *Artificial Life and Robotics*, 14:168-173.

Ziemke, T. (2008). On the role of emotion in biological and robotic autonomy. *BioSystems*, 91:401-408.

Kelly I. and Melhuish C. (2001) "Slugbot: A Robot Predator" In Proceedings of *European Conference on Artificial Life*, Prague, 2159, pages 519-528.

Wilkinson, S. 2000. "Gastronome" – A Pioneering Food Powered Mobile Robot. Proceedings of the 2000 *IASTED Int. Conference on Robotics and Applications*, Paper # 318-037.

Greenman G., Kelly I., Kendall K., McFarland D & Melhuish C. (2003). "Towards Robot Autonomy in the Natural World: A Robot in Predator's Clothing, *Journal of Mechatronics*, 13:195-228.

Ieropoulos, I, Melhuish C. and Greenman, J. (2003): Artificial Metabolism: Towards True Energetic Autonomy in Artificial Life, In Proceedings of the *7th European Conference in Artificial Life (ECAL 2003)*, Dortmund, Germany, pages 792-799.

Melhuish, C., Ieropoulos, I., Greenman, J., and Horsfield, I. (2006). Energetically Autonomous Robots: Food for Thought. *Autonomous Robots*, 21:187-198.

Ieropoulos, I., Greenman, J. and Melhuish, C. (2008). Microbial fuel cells based on carbon veil electrodes: Stack configuration and scalability. *International Journal of Energy Research*, 32:1228-1240.

What a Sunflower Can Teach a Robot? Efficient Robot Queueing by Reverse Phyllotaxis

Yaroslav Litus and Richard T. Vaughan

Simon Fraser University, Burnaby, BC, Canada
ylitus@sfu.ca

Abstract

We present a distributed multi-robot controller for forming spatially efficient queues of arbitrary numbers of robots. The method is formally analyzed and validated in a conventional robot simulation. This controller is based on sunflower phyllotaxis and inherits its efficient packing properties. Two measures of queue spatial efficiency are proposed and their upper bounds for the presented controller are found. The controller compares favorably with a simple line queueing and shows unexpectedly high tolerance to spatial interference between robots.

Introduction

Living systems have evolved remarkable properties that are very desirable to have in embodied artificial systems including robots. Biomimetic robotics focuses almost exclusively on animals and bacteria, which is natural since members of these kingdoms face locomotion-related tasks similar to those of robots. In this paper we show that useful inspiration can be obtained from plants as their growth can be viewed as movement. We describe a multi-robot system based on phyllotaxis, in particular the arrangement of seeds on a sunflower head. To our knowledge, this is the first robot controller inspired by plant morphogenesis.

The problem we are solving is in the context of our interest in the energetics of large-scale multirobot teams. We believe that the ability to manage its own energy is a key characteristic of artificial and biological living systems. Autonomous energy management poses a plethora of challenges one of which is sharing a single charging station between many robots. Here we focus on finding an efficient way for robots to organize themselves into a queue while waiting for the service at the station.

Specifically, we are looking for a queue organization that will allow a large group of hungry robots to queue for the station without creating a major obstacle for other robots and without spending too much energy on supporting the formation. Thus, we want the queue to be dense and not to extend far in any direction so it is easy to navigate around. Also, we want to decrease the additional distance a robot needs to travel in order to join and move in the queue. Though we focus on robots and recharging, our arguments could be applied to any type of service and any embodied artificial living agents.

The next section reviews related work which is followed by definition of Vogel's sunflower phyllotaxis model. We present our modification of this model and define the robot controller based on the modified model. We analyze this controller in terms of two measures of queue spatial efficiency and compare it with a simple line queueing solution. After that we describe an informal experimental demonstration of the system and conclude by summarizing the paper and offering directions for future work.

Related work

Biomimetic robotics is a vibrant and diverse field. An up to date exploration of biomimetic robot mechanisms was done by Vepa (2009), while Bar-Cohen and Breazeal (2003) provide wider survey of the field and discus both mechanisms and control. Biologically inspired robot navigation was reviewed by Franz and Mallot (2000).

A rare plant-motivated robotics work by Armour and Vincent (2006) describes robot morphology motivated by tumbleweed plant. Another unconventional non-animal design is a robot controlled by a slime mold (Tsuda et al., 2007).

Phyllotaxis has been studied extensively both by mathematicians and biologists. One of the most known works on phyllotaxis modelling was done by Vogel (1979). An accessible introduction to the topic is available at Prusinkiewicz and Lindenmayer (1990, Ch. 4), while a detailed review of the early work was done by Jean (1994). Embryogenic mechanisms involved in phyllotaxis are disused in Traas and Vernoux (2002). In a recent work Nisoli et al. (2009) give experimental and numerical evidence for emergence of phyllotaxis in a system of repulsive particles.

Work on autonomous robot recharging has traditionally focused on the engineering issues of the problem and used simplistic non-optimal recharging policies (Silverman et al., 2002; Oh and Zelinsky, 2000). Recently Wawerla and Vaughan (2007) described a near-optimal robot recharging control policy that mimics animal foraging. Couture-Beil

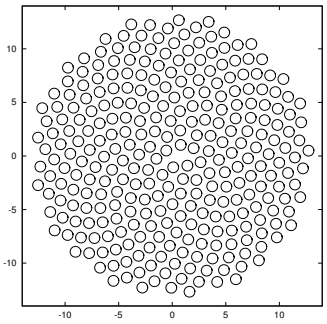


Figure 1: Sunflower head with 300 circular elements. Radius of an element is $r = 0.5$, Vogel constant $c = 0.75$.

and Vaughan (2009) developed an adaptive interference reduction strategy of placing recharging station and observed that the optimal location of the station is slightly off the path of working robots. A coordination mechanism for a large number of robots and multiple charging stations is presented by Drenner et al. (2009)

To our knowledge, no previous work has explicitly considered the cost of robot queues, either in terms of their direct navigation cost, or the indirect system cost due to the spatial interference they induce. Both are addressed here.

Efficient robot queueing by reverse phyllotaxis

Vogel's model

One of the best known models of sunflower phyllotaxis was proposed by Vogel (1979) in response to an early work by Mathai and Davis (1974). Vogel's model gives a constructive procedure for the shape of the mature sunflower head with the elements of equal sizes:

$$\rho = c\sqrt{n}, \tag{1}$$

$$\theta = ng, \tag{2}$$

where (ρ, θ) are the polar coordinates of the n-th element (r is the distance from the centre and θ is an angle between the element and a fixed axis passing through the centre), c is a scaling factor, and $g = \frac{2\pi}{\phi}$ is the golden angle (the smaller of the two angles produced by sectioning a circle circumference according to the golden ratio $\phi = \frac{1+\sqrt{5}}{2}$) so that the ratio of the full circumference to the larger arc is equal to the ratio of the larger arc to the smaller arc. A pattern produced by this model is shown on Fig. 1.

The elements in Vogel's model are arranged on a Fermat's (parabolic) spiral which has a general form $r^2 = a^2\theta$. Every turn of the spiral in the model contains on average $\phi + 1$ elements. Since Fermat's spiral crosses the annuli of equal areas in equal number of turns, equal areas on the head contain on average equal number of elements. The irrational angle between successive elements ensures that no two elements are located at the same angle. These two properties alone do not guarantee efficient packing of elements as locally the element packing density may differ significantly and large areas of unused space can be present.

However, the choice of the golden angle produces the theoretically most efficient packing of the elements among formations described by Eq. (1-2). Ridley (1982) proves that this angle will maximize the normalized packing efficiency defined as

$$\eta = A^{-1} \inf \left\{ |x - y|^2 : x, y \in X, x \neq y \right\},$$

where A is the average area occupied by each element (including its share of adjacent free space), and X is the set of elements. If η is high, then there are no areas where packing is too dense. Since elements are packed equally on average, having no overly dense areas ensures the absence of overly sparse areas with unused space.

This efficient packing and roughly circular shape of the sunflower head are appealing as a formation for a group of queueing robots. The head of the queue can be located at the centre of the sunflower and queuing robots can arrange themselves around it as if they were sunflower seeds. Simplicity of the model will transfer to the simplicity of a robot controller. Below we argue that this formation has a small diameter and allows for a low navigation overhead on joining and leaving the queue. However, first we need to provide a means for the robots to leave the queue once they were serviced.

Leaving the exit gap

Dense packing of the elements in Vogel's model makes the task of navigating from the centre of the formation outside very challenging. To minimize the interference and decrease the time spent on leaving the queue robots leave a gap from the centre of the formation to the periphery. This gap is located at a predefined angle and is wide enough for a robot to drive through (see Fig. 2). Assuming circular elements,

$$d(\rho, \theta) = |\rho \sin(\theta - \alpha)|, \tag{3}$$

$$V = \left\{ (\rho, \theta) | r = c\sqrt{n},\ \theta = gn \right\}, \tag{4}$$

$$G = \left\{ (\rho, \theta) \in V | (d(\rho, \theta) > s) \vee (\cos(\theta - \alpha) < 0) \right\}, \tag{5}$$

where $d(p)$ gives the distance of point p from the line passing through the centre of the exit gap, α is the direction angle of the gap, V is the set of element centres generated by Eq. (1-2), G is the set of element centres pruned of the elements that block the exit and s is the diameter of the element. The cosine condition in the generator for G is needed to restrict the blocking elements to the half plane in which the exit gap is located. As element locations are generated sequentially using Vogel's model, blocking elements can be skipped.

Therefore, leaving the queue amounts to simply going along the gap. Leaving the gap constantly open will increase

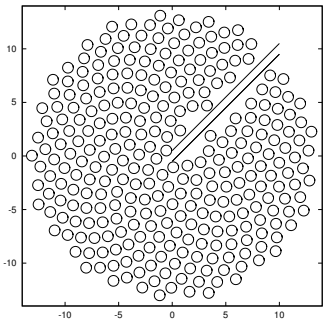


Figure 2: Sunflower head with 300 circular elements and an exit gap. Radius of an element is 0.5, Vogel constant $c = 0.75$. Exit direction $\alpha = \pi/4$.

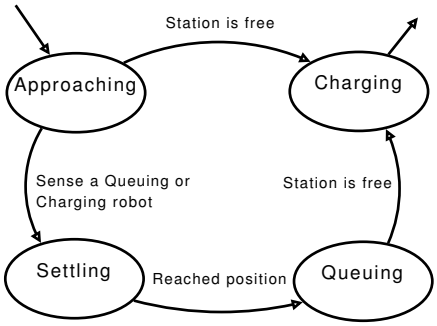


Figure 3: State diagram of the robot controller.

the diameter of the formation. However, it will eliminate the need for the queueing robots to move while letting a leaving robot through the formation. In the theoretical analysis section we explain why we believe this trade-off is reasonable. Also leaving a constant exit gap will be beneficial if the service rate is so high that a serviced robot starts leaving formation before another one finishes exiting.

Below we will use the terms "sunflower formation" and "sunflower queue" to refer to the formation with an exit gap unless specified otherwise.

Controller definition

We assume that robots are localized relative to the service stations. Every robot is equipped with a short range sensor capable of sensing the relative position of other robots. Possible choices for such a sensor include a stereo vision system and a laser-ranger-based fiducial finder. During the queueing routine a robot can be in one of four states. For simplicity we assume that if robots A senses robot B, it receives both relative position of B and its state. However, since the state of the robot can be deduced from its position and velocity, the state sensing is redundant. Sensors are subject to occlusions, so a robot can not sense through other robots.

Figure 3 describes the state diagram of the controller. When robot needs to get service, it switches into *Approaching* state. In this state robot drives straight to the charging station. If the station is free, it reaches it and switches to the *Charging* mode. Once recharged, the robot vacates the station and leaves along the predefined exit direction. If the station is busy, or the robot senses another robot in *Queueing* state, it switches into *Settling* state and calculates its position in queue based on the position of the furthest robot from the station observed so far. The Settling robot orbits around the queue and stops when it finds its position. Once there, the robot switches to Queueing state. If a robot in Queueing state is the closest one to the free charging station, it moves there and switches to Charging state. Other robots close to station sense this movement and move themselves closer to the station. This movement propagates through the whole queue and every robot moves closer to the station.

Below we provide a more formal description of behaviour in Settling, and Queueing states. All coordinates are polar with the origin located at the charging station. The currently assumed position of self on the formation is n, (ρ_d, θ_d) denotes the currently desired position, (ρ, θ) is the current actual position, ρ_i is the distance to station of the observed robot i (this can be calculated from the robot's own global position and the observed relative position of robot i), o is the orbiting offset, c is the Vogel constant, and g is the golden angle.

Settling A robot switches to the Settling state once it detects the presence of the queue by discovering that the charging station is occupied or sensing another robot in a Queueing state. The robot initializes its queue position to zero (line 1 of the Algorithm) and then processes positional information about the robot it senses.

The global position of a sensed robot is calculated as a sum of a global position of self and relative position of the sensed robot. If the robot observes a Queueing robot i the formation position of which is greater than the robots assumed position (line 4), the robot will chose the next position on a sunflower to occupy(line 5). If this position blocks the exit, the robot skips this position and chooses the next one (lines 6-8).

The robot calculates his desired coordinates from his selected position in the formation (lines 9-10) and navigates to that position in the following way. First, he moves from his current position to an orbit which is slightly above his desired distance from the charging station (line 13). Once on the orbit it moves on that orbit toward its desired angle (line 14). Once it successfully reaches this angle, it moves down from the orbit to the desired distance (lines 15-17).

The loop (2-18) ensures that the orbiting robot recomputes its desired position if it discovers a Queueing robot that occupies the desired position or even is further from it. At most one full turn around the formation will provide the Settling robot with a correct position in a formation. Once

the robot reaches its desired position, it switches to Queueing state(line 20).

Algorithm 1 Settling state controller

1: $n \leftarrow 0$
2: **repeat**
3: **for all** sensed robots i in $Queuing$ state **do**
4: **if** $\rho_i^2/c^2 \geq n$ **then**
5: $n \leftarrow \rho_i^2/c^2 + 1$
6: **if** $(c\sqrt{n}, ng)$ blocks the exit **then**
7: $n \leftarrow n + 1$
8: **end if**
9: $\rho_d \leftarrow c\sqrt{n}$
10: $\theta_d \leftarrow ng$
11: **end if**
12: **end for**
13: go to orbit $\rho_d + o$
14: move some distance along the circular orbit toward angle θ_d
15: **if** $\theta = \theta_d$ **then**
16: go to radius ρ_d
17: **end if**
18: **until** $p_c = p_d$
19: stop
20: $state \leftarrow Queuing$

Queueing A robot switches to Queueing state only from Settling state once the robot reaches its proper position in the formation. While in Queueing state, a robot finds the nearest robot it can sense that is closer to the charging station than itself and remembers the radius of this robot (line 1). If the robot senses a free charging station and is closer to the station than all queueing robots it senses to the station, then it is next to be charged and it proceeds to the station (lines 3-5). Once at the station the robot switches to Charging state.

The robot repeatedly finds the current value of the radius of the nearest sensed robot closer to the charging station (step 8). A change in the value means that a robot left the charging station, another robot occupied it and the queue moves closer to the station in response. This movement propagates from the centre of the formation to the periphery. The robot does not change its angle, but moves to the previous radius in the Vogel's model. Relocation happens once the new position is free (steps 9-11). If the new position blocks the exit the robot moves instead along the exit gap restoring the distance to the closest robot. When the robot reaches its new position, it updates the distance to the nearest robot which is closer to the charging station (step 16).

The positional update on steps 10-14 ensures that the order of recharging will correspond to the order of queueing. After the update the formation will remain a Vogel's formation with a gap. Such an update can be thought of as an inverse phyllotaxis during which the elements move inwards toward the centre instead of moving outwards.

Algorithm 2 Queueing state controller

1: $\rho_f \leftarrow \max_{\{\text{sensed Queuing } i|\rho_i<\rho\}} \rho_i$
2: **repeat**
3: $\rho_{\min} \leftarrow \min_{\{\text{sensed Queuing } i\}} \rho_i$
4: **if** $\rho < \rho_{\min}$ and charging station is free **then**
5: Move to charging station
6: $state \leftarrow Charging$
7: **else**
8: $\rho_c \leftarrow \max_{\{\text{sensed Queuing } i|\rho_i<\rho\}} \rho_i$
9: **if** $\rho_c < \rho_f$ and $(c\sqrt{(n-1)}, \theta)$ is free **then**
10: $n \leftarrow n - 1$
11: **if** $(c\sqrt{n}, \theta)$ is not blocking the exit **then**
12: Move to $(c\sqrt{n}, \theta)$
13: **else**
14: Move along the exit gap until $\rho_c = \rho_f$
15: **end if**
16: $\rho_f \leftarrow \max_{\{\text{sensed Queuing } i|\rho_i<\rho\}} \rho_i$
17: **end if**
18: **end if**
19: **until** $state = Charging$

Analysis

Our goal is to optimize two performance characteristics : the diameter of the queueing formation and the locomotion overhead on queueing. In this section we analyze the sunflower formation and provide theoretical guarantees of diameter and locomotion overhead. We will not make any assumptions about the initial spatial distribution of robots and service rate of charging station and derive instead the upper bounds of the performance characteristics. Moreover, since queueing overhead depends on the angle of the approach of the robot, worst case analysis allows to avoid complexities of parameterizing the result on that angle. Our primary interest is how performance characteristics change as the number of formation members grows.

Diameter Formation diameter is the maximum distance between the elements of the formation. Decreasing formation diameter is beneficial as it in general reduces the cost of non-participating robots to navigate around the formation.

Definition 1. *For a formation* $V = \{p|p \in \mathbb{R}^2\}$ *diameter* $d(V) = \max \|p_i - p_j\|, p_i \in V,\ p_j \in V$.

Lemma 1. *If* n *robots* s *are in the sunflower formation* $S(n)$ *with an exit gap,* $d(S(n)) \leq 2c\sqrt{2n} + s$*, where* c *is Vogel's constant and* s *is the size of the robot.*

Proof. Construction of formation with gap places elements according to the Eq.(1-2) skipping elements that block the exit. In Settling algorithm this skipping happens at steps

6-8. For a single-element-wide gap no two successive elements can block the exit, so at most every other position is skipped. Therefore, n-th element in the formation will be placed at most at radius $c\sqrt{2n}$. By construction, all previous elements are placed at smaller radii. Hence, the centres of all formation elements fit into a circle with diameter $2c\sqrt{2n}$. Since an element fits into the circle with diameter s, the maximum distance between points on formation surface is less than $2c\sqrt{2n} + s$. □

Locomotion overhead Queueing requires a robot to move into its position in the formation and then move in the queue until the robot reaches the charging station. This will usually require more locomotion than in the case where the station is free and robot can go straight for it. Locomotion overhead measures additional travelled distance caused by queueing.

Definition 2. *Locomotion overhead $P_o = P_r - P_s$, where $P_s = \|a - l\|$ is the distance between the point a at which robot detects the queue and starts a queueing manoeuvre, l is the location of charging stations, and P_r is the length of the robot trajectory from point a until it reaches the charging station while in queue* [1].

Lemma 2. *n-th robot in the queue has locomotion overhead $P_o(n) < (2\pi+2g)(c\sqrt{2n}+o)+c\sqrt{2n}-c\sqrt{2n-2}+o+2s$.*

Proof. Assume a robot detected a queue at point a. Its trajectory from that point to the charging station is comprised from three components (i) getting to the settling orbit, (ii) orbiting to the position in a queue, and (iii) moving toward the station while in a queue. By the argument used in the proof of Lemma 1 we conclude that n-th robot in a queue will settle at radius $c\sqrt{2n}$. The longest possible orbiting path for a robot n will result from detecting robot $n - 1$ only after one almost full turn around the queue and then skipping the next position on a spiral because it blocks the exit. Therefore, a robot will settle in less than one full turn and two golden angles on a circumference of circles of the radius less than $c\sqrt{2n} + o$, where o is the orbiting offset. Hence, component (ii) of the trajectory has an upper bound of $(2\pi + 2g)(c\sqrt{2n} + o)$.

Once settled and in a queue a robot moves only toward the charging station as it would do in the absence of a queue. Therefore, the only part of components (i) and (iii) that will contribute to the overhead is the travel from radius of point a to $c\sqrt{2n} + o$ and back. Because of the tight packing of the sunflower formation an approaching robot can travel at most one robot size s away from the outermost located robots before detecting the queue. That outermost located robot has number at least $2n - 2$. Therefore, the total contribution of (i) and (iii) is less than $s+o+c(\sqrt{2n}-\sqrt{2n-2})$ for a robot that is not encountering the exit gap on its straight path in a queue to the charging station.

For a case when robot has to follow the exit gap and depart from the straight path to the station a simple geometric argument shows that the increase in the path can not be greater than s. Hence, contribution of (i) and (iii) is bounded by $2s + o + c(\sqrt{2n} - \sqrt{2n-2})$ □

Comparison with the linear queueing There is no conventional robot queueing formation to serve as a benchmark for new queueing strategies. We will compare the sunflower formation with a simple and natural line queueing strategy. In this strategy a robot goes directly toward the charging station. If the charging station is occupied, the robot queues in a straight line that goes to the prespecified direction. To do this the robot follows the queue away from the station until it finds a free spot on a line. It is easy to argue, that diameter of this formation is $d(n) = ns$, where n is the number of robots in the queue and s is the size of the robot. Also, the locomotion overhead of n-th robot in line queueing is $P_o(n) = 2ns$ since the robot has to travel exactly two queue diameters before it reaches the charging station.

It seems that the linear queueing strategy has a lot of room for immediate improvement. For example, instead of going straight to the station, the robot can align itself with the queueing direction and then follow it to the station. If there is a queue, the robot detects it before reaching the station and can possibly reduce locomotion overhead by decreasing its travel to the tail of the queue. On the other hand, in case of no queue or a short queue this strategy will actually increase the overhead. A careful consideration shows that robot can make a correct decision on where to go only if he has an estimate of the current queue size beforehand. However, since the system described in this paper can also improve its performance by using a priori queue size information we will keep the comparison fair by using the simple uninformed linear queueing.

Linear queue diameters and the diameter bound of the sunflower queue differ in their rate of growth. The diameter of a linear queue grows linearly with queue cardinality, $d_l(n) = O(n)$, while the upper bound of a sunflower with a gap formation diameter grows at a slower square root rate $d_s(n) = O(\sqrt{n})$. Therefore, for any size of the robot and any Vogel's constant c the sunflower queue is guaranteed to eventually outperform the linear queue as the size of the queue grows, though for small queue this might not be the case.

Fig.4 compares the queue diameter of the simple linear queue with an upper bound of queue diameter of a sunflower formation for robots with size $s = 0.5$ and Vogel's constant

[1] It may be argued that the overhead should include leaving the formation and even returning to the original line of approach. However, it is not easy to define a standard way to measure these components of the trajectory across different formations. In any case, accounting for these components do not change the rate of growth of performance measure and qualitative comparison results we obtain.

$c = 0.6$. For queues with less than 13 robots a linear queue may have a smaller diameter, but for larger queues sunflower formation is guaranteed to outperform a linear queue. For 40 robots the sunflower queue already has half the linear queue diameter. The margin between the measures of two queues grows linearly as the queue size grows.

The locomotion overhead of the linear queue and the overhead bound of the sunflower queue relate similarly. The locomotion overhead of a linear queue grows linearly, while the overhead bound of a sunflower queue grows at a square root rate. Again, because of this difference in growth rates for any set of parameters there is a robot position for which the sunflower queue guarantees smaller overhead than the linear queue. For the larger robot positions sunflower queue will keep outperforming the linear queue and the margin between the measures will grow linearly with the robot position.

Fig.5 compares the locomotion overhead of the linear queue and the overhead upper bound of the sunflower queue for robots with size $s = 0.5$, Vogel's constant $c = 0.55$ and orbiting offset $o = 0.7$. For robot positions below 76 the linear queue can perform better, but for larger values the sunflower queue is guaranteed to have a smaller locomotion overhead.

Justification of leaving the exit gap Performance functions growth considerations can be also used to explain our choice of a leaving strategy for a recharged robot. Keeping the formation tight without a gap will lead to a constant factor improvement in the queue diameter, however robots will need to move and create an opening for every leaving recharged robot. Since the formation is tight, all robots will need to move whenever somebody leaves the formation from its centre. The last member of a queue with n members will need to move $O(n)$ times, therefore increasing the growth factor of the locomotion overhead from square root to linear. As we are interested in the efficient strategies for large number of robots, we prefer to leave a gap in a formation and suffer a constant factor increase in diameter but keep the growth rate of the locomotion overhead sublinear.

Demonstration

We implemented our queueing controller in the conventional robot simulator Stage (Vaughan, 2008). We simulate a team of 30 Pioneer robots in an 10m by 10m arena. Robots are equipped with short-range fiducial sensors capable of sensing bearing and distance to other robots, and a global positioning system. Robots do not communicate between themselves or with a charging station. Robots can collide, they have non-holonomic driving, and speed restriction and their fiducial sensors can be occluded by other robots. Parameters of simulation are given in Table 1.

We employ a simple orthodox reactive collision avoidance algorithm that uses range-finder readings. If there is an obstacle closer than a certain distance d_{stop}, the robot stops. If there is an obstacle which is at closer than a certain distance $d_{\text{avoid}} > d_{\text{stop}}$ then the direction that gave the smallest distance reading is found. If smallest reading came from the direction to the right of the robot bearing, a collision avoidance manoeuvre with a duration randomly selected in a certain interval is performed. The robot starts to turn left with a fixed turning speed and driving speed. Otherwise, the robot performs a right turn manoeuvre. If smallest reading came from the left, a right turn manoeuvre is performed. Once the collision avoidance manoeuvre is over, the robot continues to set the speed as prescribed by the main controller.

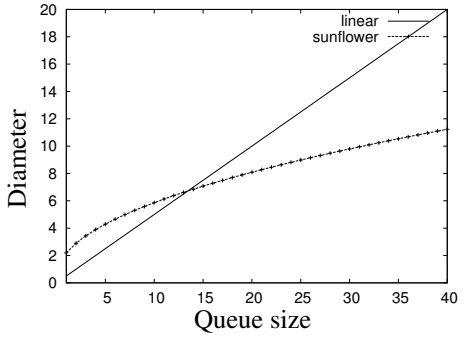


Figure 4: Diameters of the linear queue and the sunflower formation with a gap (vertical axis) plotted against number of robots in a queue (horizontal axis). Robot size $s = 0.5$, Vogel's constant $c = 0.6$.

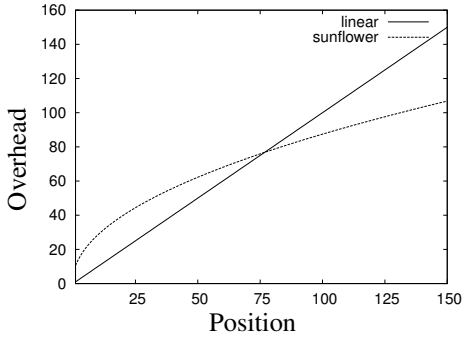


Figure 5: Locomotion overheads of the linear queue and the sunflower formation with a gap (vertical axis) plotted against robot position in queue (horizontal axis). Robot size $s = 0.5$, Vogel's constant $c = 0.55$, orbiting offset $o = 0.7$.

1: In the first set of simulations robots join the queue one by one with enough delay to let the previous robot settle in a queue and not create interference between settling robots.

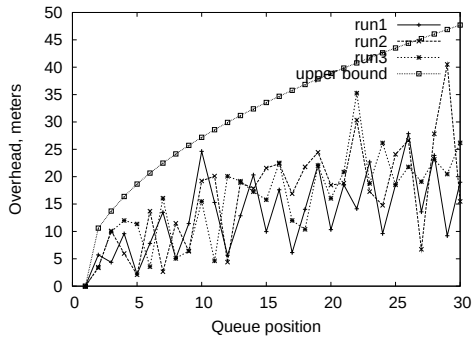


Figure 6: Locomotion overhead data from 3 experimental runs and a theoretical upper bound (vertical axis) plotted against robot position in queue (horizontal axis). Vogel's constant $c = 0.55$.

Locomotion overhead is measured as robots settle and move in queue. Once all robots join the queue, recharging start and robot recharge one by one until the queue is empty. The simulation stops once all robots are recharged. For various settings of Vogel's constant c and different initial approach of robots we observed the successful organization of sunflower formation with a gap and queue position updates after recharged robots depart.

Figure 6 shows the observed navigation overhead plotted against robot queue position from some representative example runs. All measured locomotion overheads were below the theoretically predicted upper bound (Lemma 2), which is also plotted. The angle of approach to the queue determined how closely the measured value approached the upper bound. If the approaching robot had the previously settled robot on the opposite side of the orbiting direction and beyond its sensor range, than an almost full turn around the formation was performed before the settling robot was able to sense it and calculate the position in formation. In this case the measured value of overhead was close to the theoretical upper bound. If the angle of approach allowed the robot to detect the last previously settled robot more quickly, then the measured value of overhead was significantly lower, than the predicted upper limits.

2: In the second set of simulations we tested how the system would cope with multiple robots approaching an empty queue at the same time. In this case they interfered with each other and a reactive collision avoidance algorithm took over control of the robots that came too close to other robots. The system handled interference unexpectedly well. For a small number of simultaneously approaching robots (between two and five) the system reliably created the formation albeit with a delay caused by repeated interference avoidance. For larger number of simultaneously approaching robots occasional collisions were observed as the collision avoidance was not able to handle large number of robots in close proximity to each other. However, most of the collisions were resolved by the emergent "helping" behaviour of other robots that approached stuck robots and triggered collision avoidance that separated them. Even for a very large number of robots successful formation creation was possible.

Maximum speed	0.4 m/s
Collision avoidance speed	0.05 m/s
Collision avoidance turning speed	0.5 rad/s
Collision avoidance initiation distance	0.6 m
Minimum front stopping distance	0.5 m
Collision avoidance duration interval	[1,2]s
Fiducial finder range	2 m
Orbiting offset	0.7 m
Position settling precision	0.05 m

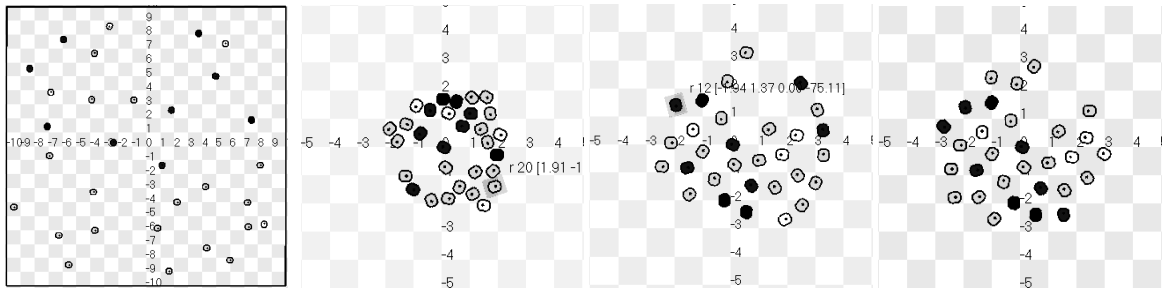
Table 1: Parameters used in Stage simulation

Fig. 7 illustrates successful creation of the formation by the group of 30 robots. Fig. 7(a) shows the initial positions of the robots. As they all simultaneously drive for the charging station a lot of interference occurs and robots spend most of the time in collision avoidance mode (See Fig. 7(b) with two robots in position and the rest interfering with each other). Eventually robots succeed in settling in positions and formation starts to grow (see Fig. 7(c) with 8 robots still settling). Fig. 7(d) shows the final formation.

Observe the group of robots following each other on the orbit in the right side of Fig. 7(c). This emergent "train formation" behaviour results from the interaction of orbiting part of the settling algorithm and the collision avoidance mechanism that randomizes the collision avoidance manoeuvre duration thus spreading robots in time. We believe that this emergent behaviour explains the tolerance of the system to spatial interference. As the queue forms, the system is capable of handling increasing numbers of simultaneously joining robots as the orbit circumference increases.

Conclusion

The focus of this paper was on autonomous creation of spatially efficient queues by a group of robots. We described a novel distributed decentralized queue formation algorithm inspired by the plant phyllotaxis, which we call the *sunflower formation*. To our knowledge this is the first robot control algorithm inspired by phyllotaxis. We defined two measures of spatial efficiency for robot queues and proved upper bounds of these measures for the sunflower formation algorithm. Our algorithm compares favourably with a simple linear queueing algorithm showing superior asymptotic behaviour of both measures. The controller was successfully



(a) (b) (c) (d)

Figure 7: 30 robots simultaneously attempting to join an empty queue. Vogel's constant $c = 0.55$ Exit direction $\alpha = \pi/3$.

demonstrated in a conventional multi-robot simulation and showed an unexpectedly high spatial interference tolerance.

This work can be extended in many directions. The first is the extension of the algorithm to create efficient queue formation in three dimensions with potential application in aerial, space and underwater robotics. A second direction is looking for ways to improve queueing as a system component, for example integrating it with a custom collision avoidance algorithm that would favour its emergent properties and allow it to successfully manage larger number of simultaneously approaching robots. Also, it may be possible to eliminate the need for global localization by using the relative poses of sensed queueing robots in addition to their relative positions. Finally, other queueing formation like zig-zag queue and theoretically optimal hexagonal packing should be investigated.

A very interesting direction is looking for ways to base the controller on models of emergent phyllotaxis instead of the constructive model employed here. Finally, we believe that plant kingdom has a lot of hidden potential for biomimetic robotics that is waiting to be discovered and exploited.

References

Armour, R. H. and Vincent, J. F. (2006). Rolling in nature and robotics: A review. *Journal of Bionic Engineering*, 3(4):195 – 208.

Bar-Cohen, Y. and Breazeal, C., editors (2003). *Biologically-Inspired Intelligent Robots*. SPIE- International Society for Optical Engineering.

Couture-Beil, A. and Vaughan, R. T. (2009). Adaptive mobile charging stations for multi-robot systems. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS'09)*, St. Loius, MO.

Drenner, A., Janssen, M., and Papanikolopoulos, N. (2009). Coordinating recharging of large scale robotic teams. In *IROS'09: Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems*, pages 1357–1362, Piscataway, NJ, USA. IEEE Press.

Franz, M. and Mallot, H. A. (2000). Biomimetic robot navigation. *Robotics and autonomous Systems*, 30:133–153.

Jean, R. V. (1994). *Phyllotaxis: A Systemic Study in Plant Morphogenesis*. Cambridge University Press.

Mathai, A. and Davis, T. (1974). Constructing the sunflower head. *Mathematical Biosciences*, 20(1-2):117 – 133.

Nisoli, C., Gabor, N. M., Lammert, P. E., Maynard, J. D., and Crespi, V. H. (2009). Static and dynamical phyllotaxis in a magnetic cactus. *Phys. Rev. Lett.*, 102(18):186103.

Oh, S. and Zelinsky, A. (2000). Autonomous battery recharging for indoor mobile robots. In *Proceedings of the Australian Conference on Robotics and Automation*.

Prusinkiewicz, P. and Lindenmayer, A. (1990). *The algorithmic beauty of plants*. Springer-Verlag New York, Inc., New York, NY, USA.

Ridley, J. (1982). Packing efficiency in sunflower heads. *Mathematical Biosciences*, 58(1):129 – 139.

Silverman, M., Nies, D. M., Jung, B., and Sukhatme, G. S. (2002). Staying alive: A docking station for autonomous robot recharging. In *IEEE International Conference on Robotics and Automation*, pages 1050–1055, Washington D.C.

Traas, J. and Vernoux, T. (2002). The shoot apical meristem: the dynamics of a stable structure. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 357(1422):737–747.

Tsuda, S., Zauner, K.-P., and Gunji, Y.-P. (2007). Robot control with biological cells. *Biosystems*, 87(2-3):215 – 223. Papers presented at the Sixth International Workshop on Information Processing in Cells and Tissues, York, UK, 2005 - IPCAT 2005, Information Processing in Cells and Tissues.

Vaughan, R. T. (2008). Massively multi-robot simulations in Stage. *Swarm Intelligence*, 2(2-4):189–208.

Vepa, R. (2009). *Biomimetic Robotics: Mechanisms and Control*. Cambridge University Press.

Vogel, H. (1979). A better way to construct the sunflower head. *Mathematical Biosciences*, 44(3-4):179 – 189.

Wawerla, J. and Vaughan, R. T. (2007). Near-optimal mobile robot recharging with the rate-maximizing forager. In *Proceedings of the European Conference on Artificial Life (ECAL)*, pages 776–785, Lisbon, Portugal.

Microbial Fuel Cell Driven Behavioral Dynamics in Robot Simulations

Alberto Montebelli[1], Robert Lowe[1], Ioannis Ieropoulos[2], Chris Melhuish[2], John Greenman[3] and Tom Ziemke[1]

[1]Cognition & Interaction Lab, University of Skövde, Sweden
[2]Bristol Robotics Laboratory, University of Bristol and University of the West of England, UK
[3]Microbiology Research Lab, University of the West of England, UK
alberto.montebelli@his.se

Abstract

With the present study we report the first application of a recently proposed model for realistic microbial fuel cells (MFCs) energy generation dynamics, suitable for robotic simulations with minimal and extremely limited computational overhead. A simulated agent was adapted in order to engage in a viable interaction with its environment. It achieved energy autonomy by maintaining viable levels of the critical variables of MFCs, namely cathodic hydration and anodic substrate biochemical energy. After unsupervised adaptation by genetic algorithm, these crucial variables modulate the behavioral dynamics expressed by viable robots in their interaction with the environment. The analysis of this physically rooted and self-organized dynamic action selection mechanism constitutes a novel practical contribution of this work. We also compare two different viable strategies, a self-organized continuous and a pulsed behavior, in order to foresee the possible cognitive implications of such biological-mechatronics hybrid symbionts in a novel scenario of *ecologically grounded* energy and motivational autonomy.

Introduction

Over the past decade, the perspective on what constitutes adaptive behavior in living organisms and robots has evolved from one of embodiment entailing solely the study of sensorimotor activity to one that incorporates internal bodily dynamics (e.g. Pfeifer and Scheier, 1999; Wilson, 2002; Ziemke, 2003). This century, the increased emphasis on internal dynamics to behavior has led some researchers to suggest that non-neural activity – of the type that is substantially affected by whole organism interaction with an external environment – is indispensable for garnering further insights into the nature of adaptive behavior (cf. Parisi, 2004; Ziemke, 2008; Ziemke and Lowe, 2009). Furthermore, the integration between non-neural internal components and sensorimotor activity may be at the heart of related concepts such as autonomy, emotion and agency.

The importance of non-neural internal (bodily) variables to behavioral dynamics was well appreciated by Ashby (1960). A leading figure in the British cybernetics movement in the 40s and 50s, Ashby emphasized the importance of feedback to control systems and, drawing on the work of Cannon (1915), applied the biological notion of homeostasis to an engineered artifact, the *homeostat*. The essential cognitive feature of the homeostat is that it purportedly provides a demonstration of what makes a system truly adaptive, or *ultrastable*. According to Ashby, a requisite feature of adaptive living and artificial organisms is that their behavior is governed not just by a first order reactive sensorimotor loop but also by a second order loop. In the case where environmental changes occur such that the value of a set of *essential variables* (e.g. blood glucose level) deviate from an ideal/viable bounded region, the 2nd order loop may be enacted. This 2nd order loop entails random changes in some of the system parameters that affect organism-environment interactive coupling, i.e. inducing a remapping of the sensorimotor activity. Only when the reconfiguration of the parameter values, altering the sensorimotor activity, permits essential variable values to be re-established within their ideal bounds, the stable/viable organism-environment interactive coupling will be likewise re-established.

Robotics investigations and research into adaptive simulated agents has been increasingly embracing the role of bodily dynamics regarding autonomous and adaptive behavior. Robot controllers utilizing homeostatic and non-neural modulatory mechanisms for cognitive shaping have been applied to navigation problems (Moioli et al., 2008, – neuroendocrine control), foraging (McHale and Husbands, 2006, – system-level energy constraints), competitive two-resource problems (Avila-García and Cañamero, 2004, – synthetic hormones). Other minimalist and dynamic systems centred approaches have investigated the effects of 'energy' or 'essential variables' that link agent viability to adaptive environmental interactions in terms of: action selection and anticipation (Montebelli et al., 2008, 2009), environment-contingent 'bodily' monitoring (Saglimbeni and Parisi, 2009), internal expression in resource competitive scenarios (Lowe et al., 2005) and also with regard to a minimal cognitive robotics interpretation of Ashby's ultrastability concept (Di Paolo, 2003). This whole body of work, relevant to system level energy constraints and neuro-physiological homeostatic control, has invariably

assumed abstract (or even arbitrary) metabolic dynamics. The homeostatic dynamics and their impact on robot behavior is rooted in designer-specified requirements and means of fulfillment, rather than on any bio-chemical reality.

A real-world instantiation of 'artificial metabolism', that can provide wheeled robots with (electrical) energy for behavioral performance as constrained by actual bio-chemical essential variable dynamics, exists in the form of Microbial Fuel Cell (MFC) technology (cf. Melhuish et al., 2006; Ieropoulos et al., 2007; Logan et al., 2006). MFC technology has the capacity to produce bioelectricity from virtually any unrefined renewable biomass (e.g. wastewater sludge, ripe fruit, flies, green plants) using bacteria. This provides robots with a degree of energy autonomy concerning choice of (non-battery) 'energy recharging' resource. Individual cells consist of anode and cathode compartments. Owing to the need for persistent rehydration of the electrode in the cathode compartment and the provision of substrate to be 'metabolized' in the anode compartment, the MFC electric energy wielding power can be said to depend on the dynamics of biochemical energy and water, essential variables of the system. Ongoing work in this area has led to generations of this MFC-powered robot demonstrating increasing independence from outside (human) control. The present incarnation EcoBot-III, for example, is able to circulate water and substrate intake according to a number of actuators (pumps) that also require a modicum of electric energy 'overhead'. Given the present state of the art, a critical limitation of this robot, motored by a biological-mechatronic symbiotic metabolism, is energy requirement. Individual robots are required to wait long-intervals between bursts of motor activity. Many minutes may be required for relatively little movement. Simulations based scenarios offer a means to overcome such performance constraints whilst simultaneously providing a tool for offering new insights and future direction. Moreover, the application of a (simulated) physically constrained metabolic dynamic on robotic behavioral competences, offers opportunities for investigating the significance of forms of homeostatic dynamics, provisioning adaptive behavior as it emerges from sensorimotor, internal and agent-environment interactions.

In the remainder of this article we will firstly present a MFC model pitted at a level of abstraction suitable for relative robotic platform independence and mathematically described. Secondly, we describe an abstract experimental scenario, and methodological approach used, in which a simulated robot is required to balance its MFC essential variable levels in order to remain viable. Thirdly, we report results from this experiment according to the evolutionary emergence of sensorimotor strategies tightly coupled to essential variable needs and environmental resource availability. Finally, we provide a discussion on the potential for simulations-based MFC-robotics applications to uncovering new breakthroughs in the physical domain.

Method

The MFC model

The core element of our experimental setup is constituted by the model of MFC recently reported by Montebelli et al. (2010a). The model has been derived from real experimental data generated by EcoBot-II, a prototype robot developed at the Bristol Robotics Lab and described in detail in Melhuish et al. (2006). The MFCs implemented for this robotic setup were characterized by oxygen-diffusion based cathodes. This choice critically constrainted the maximum energy performance. Nevertheless, it was fundamental to provide the robots with a long term self-sustainable energy source, thus promoting the conditions for genuine energy autonomy. With respect to other MFC models currently available in the scientific literature, e.g. in Picioreanu et al. (2007) and Marcus et al. (2007), our model was intentionally built at a high level of abstraction. This allows us to capture the characteristic energy generation dynamic of a MFC without the burden of details that would be non-crucial for our robotic simulations. In its simple formulation, the model works as a plug-in that can be easily implemented on any robot platform in simulation, and can endow robotic agents with realistic MFC energy production dynamics with minimal and extremely limited computational overhead.

As we direct the reader to the exhaustive description of the model in Montebelli et al. (2010a), we will here specify the details for its full implementation. We essentially developed a simple *resistance-capacitance* (RC) model (Fig. 1). Two of its physical parameters, namely the electromotive force (V) and internal resistance (R_i) of the MFC, fully characterize the MFC as an electric generator. These parameters crucially depend on the level of hydration of the cathode and on the chemical energy available in the substrate biomass of the anodic chamber. This dependency was extracted using system identification techniques from the experimental data. Therefore, once provided with the current level of hydration and of substrate richness, the model simulates realistic MFC energy generation dynamics, quantitatively similar to the ones produced by 8 MFCs connected in series. With reference to Fig. 1, the electromotive force V generates the electric current that through the internal resistance R_i buffers energy in the external capacitor C. The presence of this latter element is an arbitrary choice of the robot designers at the BRL to endow the system with an energy reservoir. This gives a partial solution to the strong electric constraints deriving from the low power rates that typically emerge from a MFC. This part of the circuit, fully platform-independent, describes the *energy generation* process and is specifically addressed by the MFC model. As soon as the difference potential across the capacitor reaches an upper threshold ($Vc_{max} = 2.9V$) the electronic switch (S) is triggered and the energy stored in the capacitor is mobilized towards the robot sensors/actuators and to its control electronics. This second part of the circuit, described by the

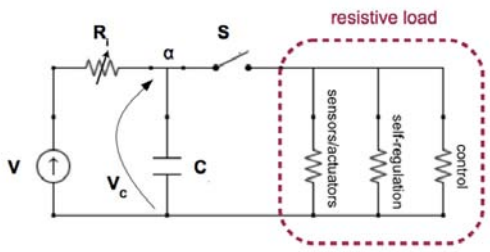


Figure 1: Electric schema of our model of energy generation in MFCs. The electromotive force (V) and the internal resistance (R_i) of the MFC depend on the current level of cathode hydration and on the biochemical energy in the substrate. This determines the dynamic of energy generation, buffered on the external capacitance (C). The dashed rectangle highlights the platform-dependent *resistive load*.

resistive load in Fig. 1, constitutes the *energy distribution* process and is completely platform-dependent. It cannot be addressed in general terms and must be tailored to the specific robot design. When the difference potential across the capacitor falls below a lower threshold ($Vc_{min} = 2.03V$) then the switch S is opened and the capacitor is recharged up to its upper threshold. This event closes the logical loop of the charge/discharge hysteresis cycle.

Using elementary electromagnetism we can describe the model in more analytical terms. The starting point is the first order linearly differential equation representing the electric current balance at node α in Fig. 1:

$$\frac{V - V_C}{R_i} = C\frac{dV_C}{dt} + I_M \tag{1}$$

where I_M represents the current drainage of the resistive load, while the meaning of all the other symbols has already been introduced. As anticipated, the quantity I_M, being platform-dependent, will be specified in the next section together with the other details regarding the specific robotic setup.

Under normal operating conditions, oxygen-diffusion cathode based MFCs are subject to water evaporation. Concurrently, although slower in time, the concentration of biochemical energy in the anodic substrate decays as a result of the bacterial activity. Linear laws describe the relations between: 1) the current level of hydration (hyd) and the time from the last full cathode hydration (t_h); 2) the chemical energy of the substrate ($subst$) and the time from the last anode replenishment with fresh substrate (t_s):

$$hyd = -\frac{t_h}{\tau_h} + 1 \tag{2}$$

$$subst = -\frac{t_s}{\tau_s} + 1 \tag{3}$$

where τ_h and τ_s (with $\tau_h << \tau_s$) respectively determine the time scales of the cathode dehydration and of the substrate biochemical energy decay.

The dependence of V and R_i with t_h is summarized by the following equations:

$$R_i = R_{i0} + k_{Ri}t_h \tag{4}$$

$$V = V_0 + k_V t_h \tag{5}$$

The effect of t_s is expressed by:

$$R_{i0} = q_R + m_R t_s \tag{6}$$

$$k_{Ri} = a_2 t_s^2 + a_1 t_s^1 + a_0 \tag{7}$$

$$V_0 = q_V + m_V t_s \tag{8}$$

The dynamic of R_{i0} is limited to values above 450. Numeric values for all the remaining symbols are: $C = 0.0282$, $k_V = -0.14$, $q_R = 642$, $m_R = -0.022$, $a_2 = 2.41e-8$, $a_1 = -1.1036e-4$, $a_0 = 0.1207$, $q_V = 3117V$, $m_V = -0.0166$, $\tau_h = 2500$, $\tau_s = 7000$ [1].

Finally, the energy currently stored in the capacitor (ε) can be easily derived from the current tension of the capacitor (V_C):

$$\varepsilon = \frac{1}{2}CV_C^2. \tag{9}$$

In conclusion, the differential equation 1, and equations 4–9 specify the model. Equations 2 and 3 allow the (equivalent) descriptions of the system in terms of time domain or as a function of the current levels of cathode hydration and substrate biochemical energy. According to this model, well hydrated MFC with fresh substrate can generate energy at a significantly higher rate than in dehydrated and 'starving' conditions. The system is particularly sensitive to the hydration level. A serious dehydration as well as an exhausted substrate determine the disruption of the charge-discharge cycle previously described and the energy generation mechanism collapses.

The robotic setup

In our experiments, a commercial *e-puck* robot simulated with the program Evorobot* (Nolfi and Gigliotta, 2010) could freely move in a square arena (measuring 1000 mm x 1000 mm), bound by opaque walls all around its perimeter (Fig. 2, central panel). Centrally located in the arena were two circular recharging areas (radius 120 mm). Upon entering in the lower circle, in whose center is placed a light source, the robot instantaneously received full cathode hydration (i.e. water was injected so to fill the capacity of its

[1] In order to limit the duration of each trial, we anticipated the kick in of the substrate effect by reducing the physical value of τ_s by a factor 3. Refer to Montebelli et al. (2010a) for details about the appropriate physical dimensions.

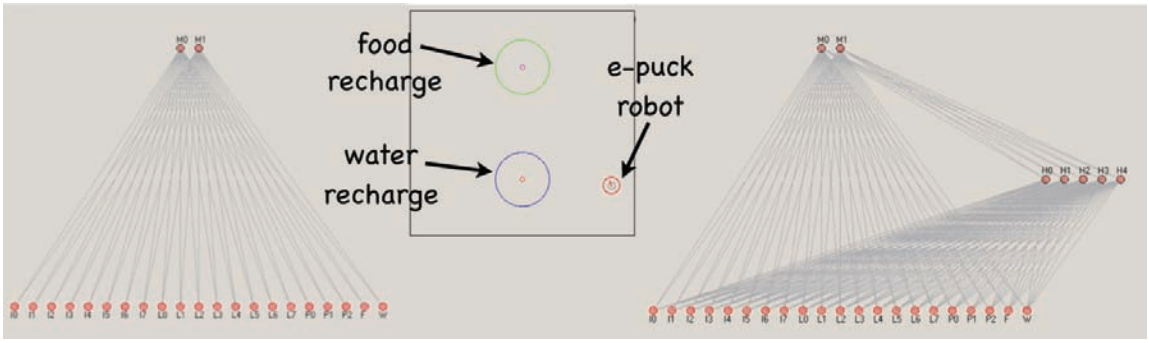


Figure 2: **Central panel-** Representation of the simulated arena. Upon entering the upper/lower circle (respectively, food/water recharging areas) the e-puck robot was fed with fresh substrate or fully rehydrated. **Left panel-** Feedforward ANN controller with no hidden layers. The ANN receives inputs from the robot's infrared and light sensors (I0-7 and L0-7), from its microphones (P0-2) and from the food and water level sensors (F and W). It outputs the motor activation signals of the robot's left and right motors (M0-1). **Right panel-** Feedforward ANN controller 5 hidden neurons and direct input-output connections.

cathode). On entering of the upper circle, landmarked by a continuous sound source, the robot received a complete and instantaneous refill of its anodic chamber with fresh substrate.

The simulated e-puck robot was provided with its standard 8 infrared sensors, 8 light sensors (activated by the light source) and 3 microphones (reacting to the sound source with an intensity that is inversely proportional to the square distance of the microphone from the sound source). A small quantity of noise was injected in the system. Customized water and food level sensors were included in the robot's sensory capabilities, providing information about the current level of cathode hydration and of the chemical energy stored in the anodic substrate.

The robot's motors were controlled by the activation of an artificial neural network (ANN). We tested several different standard architectures of discrete time ANNs, but in this report we will refer to only two of them for reasons of space. The first (Fig. 2, left panel) was a feedforward ANN with no hidden layer. The second (Fig. 2, right panel) was a feedforward ANN with five hidden neurons and direct input-output connections. In our setup, the robot's motor activation directly determined the energy drainage through the resistive load. The current I_M, i.e. the leakage term in equation 1, can be estimated as a function of the motor activation based on the robot's motor data sheets. Quantitatively:

$$I_M = 0.36|m_{act}| \tag{10}$$

where m_{act} is the current level of activation for each of the two motors, with values in the interval [-0.5 0.5], as imposed by the controlling ANN.

The energy production took place continuously (i.e. in any instant an electric current was flowing from the MFC to node α in Fig. 1) as long as the MFC was sufficiently hydrated and provided with fresh substrate. On the other hand, the energy distribution took the form of a hysteresis cycle. When the tension across the capacitor, V_C, reached its upper threshold an electric current flowed to power the robot's motors. When V_C fell below its lower threshold, the motor activity was suddenly inhibited and the robot remained still until V_C would be recharged above its upper threshold again. Accordingly, the current hydration level and the chemical energy of the substrate represent, in Ashby's terminology, the essential variables of the system.

We chose to boost the rate of energy generation characteristic of a series of 8 MFCs (the configuration that we used in order to identify the parameters of our MFC model) by a factor 100. That means that we considered a parallel electric connection of 100 elements constituted by 8 MFCs connected in series. Comments about this choice are left for the following discussion.

The free parameters of the ANN controller (synaptic weights and biases) were adapted in order to allow the robot to viably cope with its environment using a standard genetic algorithm (Goldberg, 1989) implemented in the Evorobot* simulator. We ran 10 replications of the evolutionary process, over 1500 generations with elitist selection. Each individual was on trial for 1000 simulated seconds (10000 time steps), and tested on 4 different trials from random starting position. The fitness function was intentionally rather generic: it integrated at each time step the absolute value of the current level of activation of the two motors, but only outside the recharging area. The rationale behind this choice was that we wanted the robot to consume the energy accumulated on its capacitor by demonstrating movement. On the other hand, similar to previous experiments by Floreano and Mondada (1996) and Montebelli et al. (2007, 2008), we wanted to avoid the affordance of clues about the existence of the light and sound sources, their relation to the recharging areas, their critical relations with the robot's hydration and food sensors, implicitly with the robot's energy generation rate and hence with its own overall viability.

We conclude this section with a few comments. Firstly, we emphasize the simplicity of our setup. A minimal setup focuses our attention on the object under study and allows a deeper mathematical exploration of the properties of the system. Secondly, in such a simple scenario a viable behavior might be imposed on the system by explicit design. Nevertheless, of all the options our choice was to adapt ANNs by using an evolutionary algorithm. The reason for our preference was twofold. On the one hand, we consider this alternative more liable to scaling up to more complex and less predictable circumstances (e.g. dynamically changing environments). On the other hand, we reckon on the flexibility of the fitness functions in evolutionary techniques for unsupervised adaptation, compared to other machine learning methods. This is functional to our focus on versatile robot autonomy within general and unpredictable environments, rather than on domain specific optimization.

Results

Continuous behavior

All of the considered ANN architectures managed to evolve viable behaviors for this simple task. In all cases the evolutionary process was liable to failures. Nevertheless, several classes of viable strategies were created during the evolutionary process for the best evolved individuals.

In the present and following sections we report the evolved behavior of the simplest control architecture that we considered, the feedforward ANN with no hidden layer sketched in Fig. 2, left panel. The *continuous* behavior of the best individual is shown in Fig. 3, left panel. The robot could move without sudden stops, as it would maintain a stable balance between the energy income from the MFC generator and the energy drained by its own motors (i.e. only seldom V_C fell below its lower threshold). The onboard capacitor provided a little energy buffer, but only episodically the robot had to stop and wait for its recharge.

During the initial transient period, the robot navigated in the environment, looking for a direct engagement with the water recharging area. Once reached its initial goal (Fig. 3, left panel), it maintained its engagement, looping around the water recharging area (associated with the light source) and systematically entering in it for hydration. After three loops around the light source, a fourth, larger loop would also encapsulate the food recharge area (marked by the sound source), entering which would instantaneously replenish the robot with fresh anodic substrate. This resulted in a stable and viable behavior: its timing maintained both essential variables within ideal bounds.

Essential variables as dynamic neuromodulators

By using a neuroscience-inspired clamp technique, similarly to Montebelli et al. (2008, 2009), we emphasized how the activation of the robot's water and food sensors was crucial for the emergence of the behavior. We clamped the values

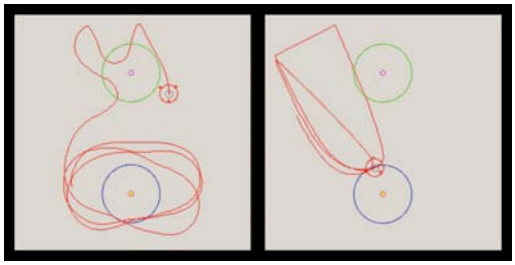


Figure 3: Examples of viable behaviors. After exhaustion of the initial transient, the robots enter in a stable, although not stereotypical loop, constituted of several passages across the water recharging area followed by one passage through the food area. **Left panel-** In the case of the continuos behavior generated by the ANN with no hidden layer (Fig. 2, left panel) the ratio between water and food access is $4:1$. **Right panel-** For the pulsed behavior of the ANN with hidden layer (Fig. 2, right panel) it is $3:1$. In both cases the trajectory of the robot is plotted for 1200 time steps.

of the two inputs F and W to arbitrary levels for the whole trial (i.e. we nullified the whole energy mechanism: the water and food levels remained constant at the selected value and the two recharging areas had no effect on the system). By systematically exploring different combinations of the clamped levels of hydration and substrate biochemical energy, we discovered that (after exhaustion of the transient period) these two essential variables, statistically determined the ratio between the numbers of accesses to water and food resources in the robot trajectories (*W:F ratio*). Ratios between $5:1$ and $1:1$ were observed (Fig. 4), and once mapped as a function of the values of the essential variables they showed a significant regularity (Fig. 5). In a tiny region of the essential variable state space, characterized by very high values of both F and W (both around 0.98), the system manifested bistability. The robot kept looping around either one or the other recharging area (Fig. 4, top and central right panels), depending on its starting position and on the integrated effects of noise. Behavioral transitions from one basin of attraction to the other were observed, although statistically rare (Fig. 4, bottom right panel). This persistence rapidly faded for different values of F and W, that modulated the height of the separation between the two different basins of attraction and the relative depth of the basins. For high values of F with subcritical levels of W (e.g. around 0.65) we noticed a maximal bias towards water, and accordingly a higher W:F ratio. Finally, in the vast area where the ratio is mapped to 0, we observed nonviable monostable behaviors, i.e. the robot would remain on a single behavioral attractor, without systematically entering any of the two recharging areas.

Detailing how the two essential variables (directly related to realistic MFC dynamics) modulated the behavioral dy-

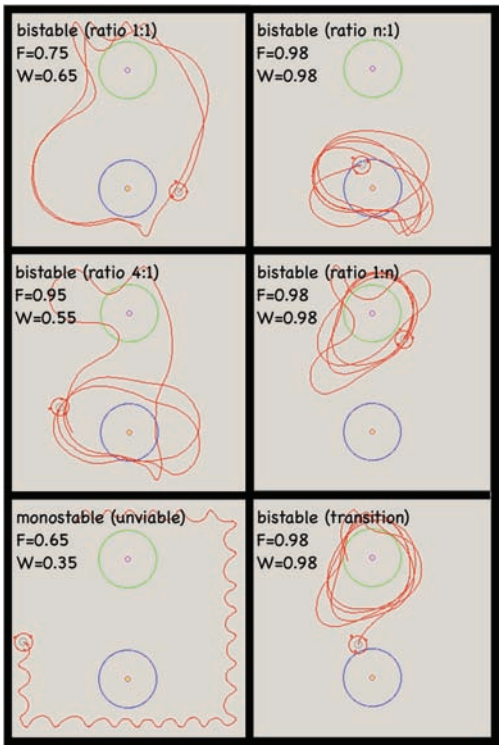


Figure 4: Examples of robot trajectories (behavioral attractors), for different clamped values of inputs W and F as specified on each panel, demonstrate different water to food access ratios. **Top and central left panels-** Examples of ratio $1:1$ and $4:1$. **Bottom left panel-** Unviable behaviors dominate lower levels of activation of the W and F sensors. **Top and central right panels-** Local behavioral attractors in the bistable regime. **Bottom right panel-** Random transition from one behavioral attractor to the other.

namics of this simple and purely reactive neurocontroller constitutes the main and novel practical contribution of this work. During normal interactions with its environment (the evolved task) the system relies on a *dynamic action selection mechanism*, self-organized during evolution without any hardwired rule.

Continuous vs. pulsed behavior

The behavior of the robot analyzed in the previous sections will here be compared to a qualitatively different *pulsed* behavior observed in the case of the feedforward ANN with 5 hidden neurons and direct input/output connections (Fig. 2, right panel). The robot always moved at its maximal speed, thus draining more energy than instantaneously provided by the MFC generator. Therefore, it systematically exhausted the energy stored on the capacitor and exploited the energy distribution hysteresis cycle previously described.

As in the previous case, the best evolved individual moved towards the water recharging area first. Once its stable behavior is reached, the robot engaged in regular loops from the water recharging area to the wall on the left side of the arena and back to the recharging area (Fig. 3, right panel). Every two loops, a third loop would emerge with a broadened width encapsulating the food recharging area. The robot apparently acted by integrating the information from all its sensory modalities. This behavior also qualifies as stable and viable, actually performing across the different trials equally well as the continuous behavior.

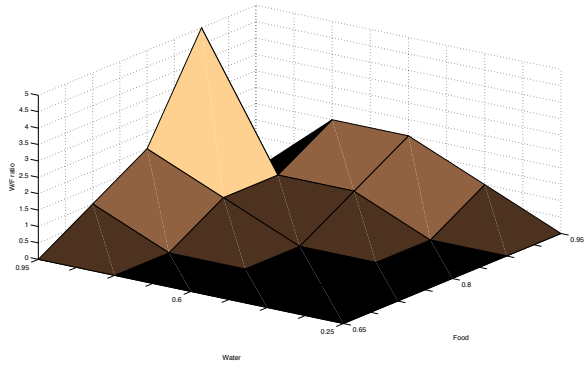


Figure 5: Water to food-access ratio (*W:F ratio*) as a function of the essential variables W and F. The area hidden under the highest peak is a region of bistability characterized by rare transitions between the two attractors. The dark area with 0 ratio represents dysfunctional behaviors: the robot cannot maintain its essential variables within a viable region.

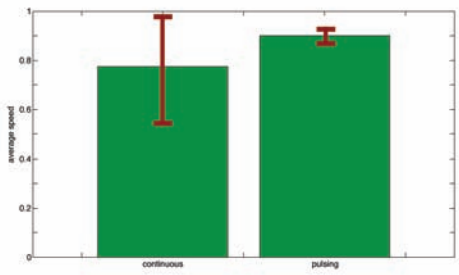


Figure 6: Average and standard deviation for the absolute value of the motor activation during continuous and pulsed behavior. Data from 2000 time steps of actual movement.

Fig. 6 quantitatively demonstrates the different nature of the continuous and pulsing behaviors. The continuously moving agent had its motors activated at about 77% of their maximal speed, with high variability, as demonstrated by the plot of the standard deviation. On the other hand, considering only the time intervals during which the robot was actually moving, the pulsing behavior was performed at 91% of the motor speed maximum, with a very low standard deviation.

Discussion

One of the most intriguing properties of computer simulations is the possibility to anticipate the forcefully slow pace of technological progress. As such, it should be used with full awareness and attention. In our study we multiplied by a factor 100 the basic electric performances of the modeled MFC energy generator. There are at least two important justifications for this choice. The first is experimental: preliminary studies (Ieropoulos et al., 2008) produced significant evidence that smaller MFCs might generate energy more efficiently, i.e. with a higher level of energy density. The second is theoretical, as it has been argued that the implementation of micron-level biofuel cells is possible in principle, and prototypes have been implemented (Kim et al., 2003). Although more research is necessary, the progressive miniaturization of MFCs seems to suggests an extremely alluring future scenario. With our choice of the multiplicative factor we anticipated the possibility to carry on board of our simple robot 800 single MFCs. The state of the art prototype of MFC powered robot, EcoBot-III, is currently endowed with a stack configuration of 48 basic MFCs. This number, limited for obvious practical reasons, is nevertheless destined to grow. Following these considerations, the factor 20 between the current physical implementation and our simulation seems appropriate.

This said, the selected multiplicative factor endowed our work with the power to foresee a crucial bifurcation in the development of MFC technology for robotic applications. The prospective historical period on which our investigation resides is the moment of transition from pulsed to continuos behaviors in MFCs powered robots. In other words, the moment when enough power is generated in order to support a sub-maximal motor activation in continuous mode. This is not to rule out the possibility of interesting pulsed behaviors. As already mentioned in Melhuish et al. (2006), for more complex cognitive architectures and environments, the intervals of stillness during energy recharge might be the perfect place to start dealing with cognition in terms of planning for thoughtful action selection, where 'mental activity' might be energetically less demanding than actual overt behavior. A similar approach, although still at a larval phase of development has been considered by Lowe et al. (2010). In this novel work, during the idle motor intervals, the robot can capitalize on active 'sensing' by executing energetically inexpensive visual saccades, rather than actual physical navigation.

Finally, why should we abandon the engineering perspective of robots that could turn to virtually unlimited sources of energy (in form of power sockets or batteries), a perspective largely inherited by cognitive roboticists? As a matter of fact, we just analyzed a not even too futuristic scenario where MFCs will converge towards offering the MFC powered robots the option of continuous action, simply considering appropriate stack configurations of basic miniaturized MFCs. Furthermore, if pragmatic results will support the theoretical expectations, MFC miniaturization might create a sort of limit situation, allowing a fully distributed energy generation system reminiscent of biological cellular energy generation strategies, where energy constrains would be crucially relaxed. A serious answer to this question has to do with our idea of autonomy. Future MFC powered robotic agents, through the development of a viable behavior in their environment, will be ecologically rooted in their environmental context. They will depend on food and water resources that are available as long as the robots can live in a sustainable and meaningful 'ecological relation' to their environment. This property, novel and original in robotics, represents an exciting new scenario for future research.

Conclusions

This work, jointly with the mentioned paper by Lowe et al. (2010), represents the first effort aimed to put to the test the MFC model for robotic simulations presented in Montebelli et al. (2010a). Its aim, beyond the mere demonstration, is to ground previous work related to the dynamic neuromodulatory role of non-neural internal variables (Montebelli et al., 2007, 2008) in a realistic simulation of physical energy constraints. The robot is energetically autonomous insofar as it can sustain a viable interaction with its environment by maintaining its essential variables. Within this tight agent-environment interaction, our analysis emphasized the neuromodulatory role played by the essential variables for dynamic action selection with no hardcoded rules. We also pointed to the possible coexistence of several viable strategies, different both in qualitative and quantitative terms and to their possible cognitive implications in a novel scenario of 'ecologically grounded' energy and motivational autonomy.

In future work we will further investigate these findings. The *2 resource problem* has been characterized in McFarland and Spier (1997), where a robot was expected to negotiate between an environmental resource critical to its survival (fuel) and the execution of a task that some external supervisor considered useful (work). We are extending our experimental setup for a fully fledged *3 resource problem*, where the exploitation of food and water will be functional to the execution of physical work in a dedicated area. In addition the experimental setup appears suitable for a deeper exploration of the concept of *embodied anticipation* (i.e. the capacity to profit from the non-neural neuromodulatory characteristics achieved during evolutionary and ontogenetic adaptation in order to perform swift readaptation to novel situations) as proposed in Montebelli et al. (2009, 2010b).

Acknowledgements

This work has been supported by a European Commission grant to the project *"Integrating Cognition, Emotion and Autonomy"* (www.iceaproject.eu, IST-027819 - 2006-2009).

References

Ashby, W. R. (1960). *Design for a Brain: The Origin of Adaptive Behavior*. John Wiley & Sons Inc., New York, NY.

Avila-García, O. and Cañamero, L. (2004). Using hormonal feedback to modulate action selection in a competitive scenario. In Schaal, S., Ijspeert, A., Billard, S., and Vijayakumar, J., editors, *Proceedings of the Eighth International Conference on Simulation of Adaptive Behaviour*, 243–252, Cambridge, MA. MIT Press.

Cannon, W. B. (1915). *Bodily Changes in Pain, Hunger, Fear and Rage*. Appleton, New York.

Di Paolo, E. (2003). Organismically-inspired robotics. In Murase, K. and Asakura, T., editors, *Dynamical Systems Approach to Embodiment and Sociality*, 19–42. Advanced Knowledge International, Adelaide.

Floreano, D. and Mondada, F. (1996). Evolution of homing navigation in a real mobile robot. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 26(3):396–407.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional.

Ieropoulos, I., Greenman, J., and Melhuish, C. (2008). Microbial fuel cells based on carbon veil electrodes. *International Journal of Energy Research*, 32:1228–1240.

Ieropoulos, I., Melhuish, C., and Greenman, J. (2007). Artificial gills for robots. *Bioinspiration and Biomimetics*, 2:S83–S93.

Kim, H.-H., Mano, N., Zhang, Y., and Heller, A. (2003). A miniature membrane-less biofuel cell operating under physiological conditions at 0.5v. *Journal of the Electrochemical Society*, 150:A209/A213.

Logan, B. E., Hamelers, B., Rozendal, R., Schröder, U., Keller, J., Freguia, S., Aelterman, P., Verstraete, W., and Rabaey, K. (2006). Microbial fuel cells: Methodology and technology. *Environmental Science and Technology*, 40(17):5181–5192.

Lowe, R., Montebelli, A., Ieropoulos, I., Greenman, J., Melhuish, C., and Ziemke, T. (2010). Grounding Motivation in Energy Autonomy: A Study of Artificial Metabolism Constrained Robot Dynamics. In *Proceedings of the 12th International Conference on the Synthesis and Simulation of Living Systems*, in press.

Lowe, R., Nehaniv, C. L., Polani, D., and Cañamero, L. (2005). The degree of potential damage in agonistic contests and its effects on social aggression, territoriality and display evolution. In *Proceeding of the Congress on Evolutionary Computation - IEEE CEC '05*, vol 1, 351–358.

Marcus, A. K., Torres, C. I., and Rittmann, B. E. (2007). Conduction-based modeling of the biofilm anode of a microbial fuel cell. *Biotechnology and Bioengineering*, 98(6):1171–1182.

McFarland, D. and Spier, E. (1997). Basic cycles, utility and opportunism in self-sufficient robots. *Robotics and Autonomous Systems*, 20:179–190.

McHale, G. and Husbands, P. (2006). Incorporating energy expenditure into evolutionary robotics fitness measures. In et al., L. M. R., editor, *Proc. Alife X*, 206–212. MIT Press.

Melhuish, C., Ieropoulos, I., Greenman, J., and Horsfield, I. (2006). Energetically autonomous robots: food for thought. *Autonomous Robots*, 21:187–198.

Moioli, R. C., Vargas, P. A., von Zuben, F. J. V., and Husbands, P. (2008). Towards the evolution of an artificial homeostatic system. In *IEEE Congress on Evol. Comput.*, 4024–4031.

Montebelli, A., Herrera, C., and Ziemke, T. (2007). An analysis of behavioral attractor dynamics. In Almeida e Costa, F., editor, *Advances in Artificial Life: Proceedings of the 9th European Conference on Artificial Life*, 213–222, Berlin. Springer.

Montebelli, A., Herrera, C., and Ziemke, T. (2008). On cognition as dynamical coupling: An analysis of behavioral attractor dynamics. *Adaptive Behavior*, 16(2-3):182–195.

Montebelli, A., Ieropoulos, I., Lowe, R., Melhuish, C., Greenman, J., and Ziemke, T. (2010a). Unplugged! a mathematical model of microbial fuel cells for energetically self-sustainable simulated robotic agents. *In preparation*.

Montebelli, A., Lowe, R., and Ziemke, T. (2009). The cognitive body: from dynamic modulation to anticipation. In Pezzulo, G., Butz, M. V., Sigaud, O., and Baldassarre, G., editors, *Anticipatory Behavior in Adaptive Learning Systems*, 132–151. Springer, Berlin, Heidelberg.

Montebelli, A., Lowe, R., and Ziemke, T. (2010b). More from the body: Embodied anticipation for swift re-adaptation in neurocomputational cognitive architectures for robotic agents. In Gray, J. and Nefti-Meziani, S., editors, *Advances in Cognitive Systems*, in press. IET.

Nolfi, S. and Gigliotta, O. (2010). Evorobot*. In Nolfi, S. and Mirolli, M., editors, *Evolution of Communication and Language in Embodied Agents*, 297–302. Springer-Verlag, Berlin, Heidelberg.

Parisi, D. (2004). Internal robotics. *Connection Science*, 16(4):325–338.

Pfeifer, R. and Scheier, C. (1999). *Understanding Intelligence*. MIT Press, Cambridge, MA.

Picioreanu, C., Head, I. M., Katuri, K. P., van Loosdrecht, M. C. M., and Scott, K. (2007). A computational model for biofilm-based microbial fuel cells. *Water Research*, 41:2921–2940.

Saglimbeni, F. and Parisi, D. (2009). Input from the external environment and input from within the body. In Kampis, G. and Szathmary, E., editors, *The 10th European Conference of Artificial Life*, Berlin. Springer.

Wilson, M. (2002). Six views of embodied cognition. *Psychonomic Bulletin and Review*, 9(4):625–636.

Ziemke, T. (2003). What's that thing called embodiment? In Alterman, R. and Kirsh, D., editors, *Proc. of the 25th annual conference of the Cognitive Science Society*, 1305-1310. Lawrence Erlbaum.

Ziemke, T. (2008). On the role of emotion in biological and robotic autonomy. *BioSystems*, 91:401–408.

Ziemke, T. and Lowe, R. (2009). On the role of emotion in embodied cognitive architectures: From organisms to robots. *Cognitive computation*, 1(1):104–117.

Chaotic Search of Emergent Locomotion Patterns for a Bodily Coupled Robotic System

Yoonsik Shim[1] and Phil Husbands[1]

[1]Centre for Computational Neuroscience and Robotics, University of Sussex, Falmer, Brighton, UK
Y.S.Shim@sussex.ac.uk

Abstract

We study a novel deterministic online process for the exploration and capture of possible locomotion patterns of a simulated articulated robot with an arbitrary morphology in an unknown physical environment. The robot controller is modelled as a network of neural oscillators which are coupled indirectly through physical embodiment. Goal directed exploration of coordinated motor patterns is achieved by a chaotic search method using adaptive bifurcation. The phase space of the indirectly coupled neural-body-environment system contains multiple phase-locked states each of which is a candidate for driving efficient locomotion. By varying the chaoticity of the system as a function of evaluation signal, it is able to chaotically wander through various phase-locked states and stabilise on one of the states matching the given criteria. The nature of the weak coupling through physical embodiment ensures that only physically stable locomotion patterns emerge as coherent states, which implies the emergent pattern is well suited for open-loop control with little or no sensory inputs.

Introduction

Properly coordinated rhythmic motor behaviours are ubiquitous in animals. From insects to humans, locomotive ability is one of the most fundamental survival mechanisms to have evolved. As has been increasingly pointed out over the past few years (Pfeifer and Iida, 2004), studying neural circuitry underlying the generation of rhythmic motor behaviour in isolation ignores the considerable advantage that can be obtained from incorporating the the physical body and its environment - an approach that can significantly reduce the amount of information needed to develop successful motor patterns.

This naturally led to efforts to exploit ready-made functionality provided by the given physical properties of an embodied system for the automatic generation of motor movement. One such line of enquiry involves using frequency adaptive oscillators that can be entrained to the resonant frequency of the mechanical system (Buchli et al., 2006), including the use of chaotic frequency scaling (Raftery et al., 2008). Although frequency adaptation to a given physical body accounts for a major part of the properties of locomotion, we believe that, in general, the appropriate phase relationship between each limb should take priority among other aspects when dealing with the creation of new motor patterns. One of the seminal works from this perspective is the exploration and acquisition of motor primitives, for a simple robot, using a mechanism which is embodied as a coupled chaotic field (Kuniyoshi and Suzuki, 2004). Those researchers modelled an extreme version of embodied coupling that had no electrical connection between neural units at all: they were only coupled indirectly through body-environment interactions. The neural oscillators were implemented using a simple logistic map with chaotic behaviour, and the system dynamics rapidly developed to a stable, coherent rhythmic motion by using mutual entrainment between the neural circuit and the body-environment interactions. The process was completely deterministic. Later work (Kuniyoshi and Sangawa, 2006) dealt with a more biologically plausible system in which a realistic musculo-skeletal model was employed and the neural control circuit consisted of a model CPG. While these previous studies have developed detailed biological models that have significant implications for the understanding of motor development, concrete general methodologies for applying such techniques to the automatic generation of desired motor patterns for autonomous robots remains a challenge.

In this paper we build on the prior work outlined above, extending and generalising it as we attempt to develop a generally applicable methodology for neural-body-environment coupled systems, based around self-organisation through chaotic dynamics. We present a study of goal directed online exploration of rhythmic motor patterns in a oscillator system coupled through physical embodiment, specifically generating forward locomotion behaviours without prior knowledge of the body morphology or its physical environment. This is explored in the context of a simulated limbed robot. In an important departure from the previous work outlined above, in order to explore and drive system dynamics toward a desired state, we employ the concept of Chaotic Mode Transition with external feedback (Davis, 1990), which exploits the intrinsic chaoticity of a system orbit as a perturbation force to explore multiple synchronised states of the system,

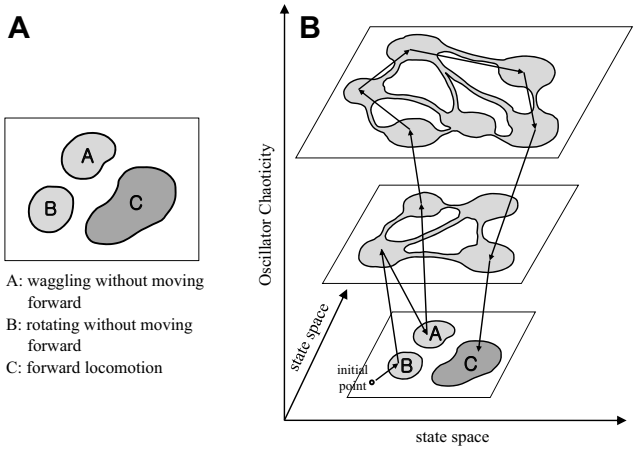


Figure 1: (A) A conceptual illustration of the state space of a neuro-body-environment system coupled through physical embodiment, which consists of three basins of attraction (A,B,C) with different performances. (B) An exploration process to find the desired attractor, C, by varying the complexity of the state space landscape. Lump spaces and narrow passages in the landscapes of higher complexities represent quasi-attractors and itinerant pathways respectively.

and stabilises the orbit by decreasing its chaoticity according to a feedback signal that evaluates the behaviour. This enables the system to perform a deterministic search guided by a global feedback signal from the physical system, which facilitates an active exploration toward a desired behaviour. This research is intended to open up new directions in the exploitation of chaos as a self-organising principle in embodied autonomous systems, as well as to potentially shed light on its role in biological systems.

Chaoticity as Perturbation Strength

Conventional optimisation strategies generally use stochastic perturbations on system parameters for search space exploration. However, a few studies address the effectiveness of chaotic dynamics as behaving like a stochastic source (Ott et al., 1994), and have found that a deterministic chaotic generator outperforms a stochastic random explorer (Morihiro et al., 2008). In these cases, the chaotic dynamics acts as an external module generating perturbations that cause system parameters to wander in parameter space. However, as we shall see, adaptive chaotic search methods using bifurcation to chaos can directly drive the phase orbit of a bodily coupled system for exploration because of the endogenous existence of chaotic dynamics in the system itself.

The general idea of applying a chaotic search method which uses adaptive parametric feedback control had been previously presented in the field of optical sciences (Aida and Davis, 1994) and for memory search (Nara and Davis, 1992). It has been argued that this method should be generally applicable when the target device is capable of supporting a variety of stable modes, with chaotic transitions existing between them, which interact with their environment and give a feedback signal evaluating whether the mode is suitable or not. Chaotic transitions allow the system to try each of the modes sequentially, and the mode which is evaluated as suitable is selected and stabilised by changing a device parameter to take it into a multistable regime. An indirectly coupled neuro-body-environmental system, such as the one used in this paper, has the required characteristics of such a device, including multiple coordinated oscillation modes. It is known that a properly designed coupled oscillator system can have multiple synchronised states which exhibit stable oscillations (Feudel and Grebogi, 1997), and the structure of emergent behaviour in these systems often reflect the spatial distribution of coupling strengths (Kaneko, 1994). Accordingly, a network of oscillators coupled through physical embodiment forms multiple synchronised states which reflects the body schema and its interaction with the environment.

A conceptual description of the chaotic search process is briefly illustrated in Fig. 1. The goal of the system can be regarded as finding and becoming entrained in the basin of a particular attractor which has high performance (denoted by C) while escaping from the low performing attractors (A and B) regardless of the initial point in the state space. The idea is to 'open' a new pathway which connects those isolated basins through use of an additional dimension afforded by changing the system dynamics through tuning the chaoticity according to the evaluation signal. The orbit will visit and evaluate each of the attractor (A,B,C) systematically yet chaotically by adaptively varying the bifurcation parameter of the system according to the feedback signal until it reaches the basin of the desired attractor. The process can be interpreted as a deterministic version of trial-and-error search which exploits the chaotic behaviour of system. For the first time, this study attempts to implement and integrate these concepts into an autonomous neuro-body-environment system, making use of a continuous-time dynamical system framework.

Method

The architecture of the neural part of the generic system developed is based on (Kuniyoshi and Sangawa, 2006), but with a more compact and modular configuration for each joint of the limbed robot. It is intended to be applicable to a wide range of robotic systems. The architecture consists of a number of identical control modules connected to each of the body parts in their environment. Each neuro-motor-joint system which receives afferent sensory input and gives motor output can be encapsulated as a single *motor unit*, and the whole system consists of identical motor units whose number is the same as the number of degrees of freedom of the robot (Fig. 2). The signal from the sensor of a motor unit (in most case a mechanosensory information) is fed, with opposite signs, to both of the pair of electrically unconnected oscillators that each motor unit contains. This configura-

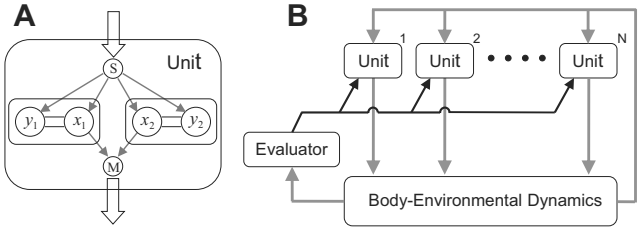


Figure 2: (A) A *motor unit* for a single degree of freedom in the joint-motor system. A unit consists of two electrically disconnected oscillators, which receive indirect integrated information of other oscillators in the system from the sensor (S), via environmental coupling, and give a control signal to the motor (M). (B) A neural-body-environment system whose body has N degrees of freedom. The complexities of all units are altered according to a global evaluation signal.

tion eliminates muscle redundancies by constraining joint-motors to be operated only by an antagonistic actuator pair, thus giving more weight to inter-limb interactions.

The control signals for the basic motor patterns are generated by central pattern generators (CPG), which are composed of a collection of neurons that produces an oscillatory signal for various locomotor patterns by synchronisation with the movement of the physical systems. The model consists of coupled Bonhoeffer-van der Pol (BVP, or Fitzhugh-Nagumo) oscillators which are widely studied as models of pacemaking cells and interlimb coordination. A particularly interesting feature of coupled BVP equations, that allows adjustment of the complexity of the system orbit, had been presented by (Asai et al., 2003). A pair of coupled BVP oscillators generates a stable limit cycle when the two control inputs are the same, but a quasiperiodic/chaotic orbit otherwise. Another interesting feature of the BVP model is flexible phase locking (Ohgane et al., 2009), where the phase relationship between CPG activity and body motion can be flexibly locked according to a loop delay. This is a beneficial feature for covering a range of sensorimotor delays originated from different body-environment configurations. A pair of oscillators for a motor unit i, dealing with its sensory input, is described by the following equations:

$$\tau\frac{dx_{1,i}}{dt} = c(x_{1,i} - \frac{x_{1,i}^3}{3} - y_{1,i} + z_1) + \delta(I_1(s_i) - x_{1,i}) \quad (1)$$

$$\tau\frac{dy_{1,i}}{dt} = \frac{1}{c}(x_{1,i} - by_{1,i} + a) + \varepsilon I_1(s_i) \quad (2)$$

$$\tau\frac{dx_{2,i}}{dt} = c(x_{2,i} - \frac{x_{2,i}^3}{3} - y_{2,i} + z_2) + \delta(I_2(s_i) - x_{2,i}) \quad (3)$$

$$\tau\frac{dy_{2,i}}{dt} = \frac{1}{c}(x_{2,i} - by_{2,i} + a) + \varepsilon I_2(s_i) \quad (4)$$

where τ is a time constant, and a=0.7, b=0.675, c=1.75 are the fixed parameters of the oscillator. δ=0.013 and ε=0.022 are coupling strength for afferent input $I(s)$ which is a function of the actual sensor value s. The time constant, which represents the frequency of the oscillator, was set to τ=0.8 throughout this work, as this was found to be an appropriate value. z_1 and z_2 are control parameters for adjusting the chaoticity of the motor unit. Their difference (z_2-z_1) changes identically in all motor units as a function of the evaluation signal, which will act as the bifurcation parameter for the chaotic search with adaptive feedback. In the stable regime where z_1 and z_2 are symmetric, (Asai et al., 2003) found that the two coupled BVP equations exhibit bistable phase locking of their oscillations in a parameter range of $0.6 < z_1 = z_2 < 0.88$. From the observation of a number of experiments on the oscillator dynamics, to ensure a higher probability of multistability of the system, we chose to fix $z_2 = 0.73$ and to vary z_1.

Evaluation and Feedback

The coherent integration of a performance evaluation signal that is able to control the chaoticity of the system is an important contribution of the current work. In the experiments to be described next, the performance evaluation signal E is measured by the forward speed of the robot. Since the system has no prior knowledge of the body morphology of the robot, it does not have direct access to the direction of movement nor of information on body orientation. In order to facilitate steady movement in one direction without gyrating in a small radius, a temporal integration of the velocity of the center of mass was formulated as an evaluation function. The center of mass velocity of a robot is continuously averaged over a certain time window and its magnitude was used as the performance of system. The performance signal E at any time instance can be calculated by applying a leaky integrator equation to the velocity vector as

$$E(t) = |\bar{\mathbf{v}}|, \quad \tau_E\frac{d\bar{\mathbf{v}}}{dt} = -\bar{\mathbf{v}} + \mathbf{v} \quad (5)$$

τ_E is the time scale of integration which is larger than that of an oscillator (slower than the oscillator period), but typically not exceeding it by more than an order of magnitude.

A global feedback signal determines the degree of chaoticity of an oscillator network. The bifurcation parameter for feedback control is continuously modified by an amount governed by the evaluation signal. If the current entrained state is not satisfactory, parameter μ is increased to where the orbit will follow quasiperiodic or chaotic dynamics, and when a satisfactory pattern appears, μ is decreased so that the satisfactory mode becomes stable. The adaptive control parameter μ ($= z_2 - z_1$) is described as follows:

$$\tau_\mu\frac{d\mu}{dt} = -\mu + G(E) \quad (6)$$

$$G(E) = \frac{\mu_c}{1 + e^{P(E)}}, \quad P(E) = \frac{16E}{E_d} - 8 \quad (7)$$

As described in the last section, z_2 (Equation 3) if fixed, hence z_1 (Equation 1) varies as μ changes. $G(E)$ is a monotonically decreasing sigmoidal function of locomotion performance E (Fig. 3). τ_μ determines the time scale of the

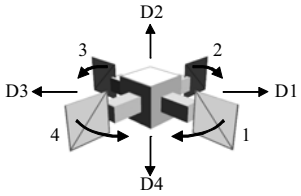
change of μ and is normally set faster ($\tau_\mu < T$) than the oscillation period (T) of the controller. If its value is too high, stabilisation of the system dynamics is significantly delayed which results in a partition mismatch (Aida and Davis, 1994). If it is too low, μ tends to fluctuate according to the undulation of the robot movement which acts as a disturbance for stabilisation, or the system can become locked in a ring of undesirable patterns in a regime of intermediate chaoticity. We used $\tau_\mu = 0.5T$ throughout this work. The evaluation function $G(E)$ determines the level of chaoticity by varying μ in the range [0,μ_c] where μ_c is the maximum level of chaoticity of the system. From the analysis of a single BVP oscillator it is well known that it shows Hopf bifurcation with the increase of the parameter z (Nomura et al., 1993). An analytically estimated critical value of z for equations (1) and (2), without its coupling term, is $z \approx 0.38247$, which indicates that the maximum possible value of μ_c is $0.73 - 0.38247 = 0.34753$. However, because the situation is different from the dynamics of a single oscillator, experiments on the robotic system presented here revealed that the actual behavioural criticality of z varies slightly among different body and environmental settings. Hence we chose $\mu_c = 0.35$, taking into consideration the asymptotic characteristic of the sigmoidal function G. E_d indicates the desired locomotion performance of a given robot. However we do not have prior knowledge of how much performance it can achieve. Hence the dynamics of E_d is modelled using the idea of a goal setting strategy (Barlas and Yasarcan, 2006). With this concept the expectation of a desired goal is influenced by the history of the actual performance experienced in the past. When the robot has already achieved high performance during a certain period in the past, the performance expectation increases. The performance expectation decays if it is not being met by the actual performance. We integrate this strategy in terms of simple continuous dynamics for E_d, which slowly decays toward the current performance. This can be simply described by a leaky-integrator equation:

$$\tau_d \frac{dE_d}{dt} = -E_d + E \tag{8}$$

where τ_d is set larger than τ_E. E_d functions as a temporal average of E for a certain time window. Since E_d continuously decays toward E, the changing speed of control parameter μ depends both on E and τ_d. Therefore τ_d determines the depth and the duration of chaotic wandering.

Experiments and Results

Initial experiments with the framework described above used the simple simulated robot shown in Fig. 3: a four-armed aquatic swimmer with fins at the end of each arm placed in a simulated hydrodynamic planar environment. The robot was modelled using ODE (Smith, 1998). A joint-motor of the robot was modelled using a pair of servo motors which generate torques in opposite direction. These motors are used as effectors for the neuronal output by varying their desired angular speed according to the simulated muscle force used by (Ekeberg, 1993). The functional structure of bodily coupling between motor units is formed by the transmission of hydraulic reaction forces of one limb to the others through body articulation. Each fin was modelled as a nonlinear torsional spring and its bending angle (θ) was fed to the corresponding motor unit. The fin angle implements the stretch receptor at each side of fin, so the afferent input I in the equations (1) and (3) were defined as: $I_1(\theta) = H(k\theta)$ and $I_2(\theta) = H(-k\theta)$ where k ($= 2.5$) is input gain and $H(x)$ is heaviside function. Joint axes and motor unit arrangements were set to be bilaterally symmetric which is a dominant feature throughout the animal world. The radial symmetry of the body morphology ensures that possible locomotion behaviours are not restricted to longitudinal directions. The radially symmetric shape in a 2D underwater environment is interesting because it makes generating continuous asymmetric propulsion forces challenging; in other words forward locomotion is non-trivial. The agent will not be able to move in a single direction unless the movements of all four arms are successfully synchronised with appropriate phase differences. The other parameters used for the search process was $\mu_c = 0.35$, $\tau_E = 5T$ and $\tau_d = 5\tau_E$



4-Fin Swimmer	
torso dimension (m)	0.2× 0.2× 0.2
arm dimension (m)	0.075× 0.075× 0.15
torso weight (Kg)	1.6
arm weight (Kg)	0.34 (× 4)
joint range (rad)	± 1.0
fin dimension (m)	0.2× 0.2
fin weight (Kg)	0.005
fin stiffness (N/m)	0.1
fin damping (Ns/m)	0.045
fluid density (Kg/m^3)	1000.0

Figure 3: The 4-Fin Swimmer and its parameters. The arrows at each joint describe the direction of rotation. Arrows D1-D4 represent the possible directions of movement.

Observation of Emergent Behaviours

First, we fixed the control parameter to a target value ($\mu = 0$, no chaotic search) and ran the simulation to see what kinds of behaviours emerged from various initial states. Numerous test was done in order to observe and categorise the behaviours that emerged from the system. Basic movement behaviours were categorised into motion in four directions (along the body axes D1,D2,D3 and D4 as shown in Fig. 4) which met expectations given the symmetric shape of the swimmer. For each direction of movement, four different behaviours were observed and classified according to the locomotion performance. These are straight movement, moving in medium radius circles, moving in small radius circles, and moving in/out in a spiral. Each circling locomotion can be either clockwise or counterclockwise. Also there were non-locomotion movements such as rotation or vibrating at a fixed position, and completely symmetric leg movements

Pattern	# of variations	Avrg E
1. straight (ST)	4 body orientations	0.45
2. medium R (MR)	8 (4×(CW/CCW))	0.25
3. small R (SR)	8 (4×(CW/CCW))	0.2
4. spiral (SP)	8 (4×(CW/CCW))	0.02-0.3
5. rotate (RO)	2 (CW/CCW)	0.03
6. vibration (VB)	2 (D1-D3 / D2-D4)	0.03
7. bound antiphase (BA)	1	0.0

Table 1: Categories of emergent behaviours. The variations of straight swimming are in 4 different body orientations. Circular movements (pattern 2,3,4) have 8 variations by including two circling directions. Vibration has 2 variations which are in direction of D1-D3 and D2-D4.

resulting in no movement (bound antiphase). The categories of emergent behaviours of the swimmer robot and their average performances are shown in Table 1, which indicates that the total number of movement patterns is 33.

In order to quantify an emergent pattern and its temporal dynamics we developed a method we call a *Feature Index* (FI) plot which is inspired by multivariable data binning techniques. A feature index is a scalar value which is calculated from the powered sum of the bin indices of the phase differences between each DoF. Therefore, a feature index can uniquely represent a given motor coordination. Since the phase difference alone cannot capture the difference of motor amplitudes we used two feature indices: one for the phase relationship and one for the amplitude relationship. If we define N phase differences of the limb movements, the feature index F can be written as:

$$F = \sum_{i=1}^{N} k_i B^{i-1}, \quad k_i \in \mathbb{Z} \tag{9}$$

$$k_i = (d_i - d_{min}) \text{ div } \{(d_{max} - d_{min})/B\} \tag{10}$$

where w is the width of a bin, B is the number of bins, and d_i is the ith wrapped phase difference which has the range $[d_{min}, d_{max}]$. The feature index for the amplitude relationship uses the phase differences between two antagonistic motor commands for d_i. The range of wrapped phase difference were $[-\pi,\pi]$ for the phase index and $[0,2\pi]$ for the amplitude index which indicates the phase difference of π between antagonistic motor signal is producing maximum amplitude. The FI plots of four different straight locomotions and the other behaviours are depicted in Fig. 4 using the following four phase differences: leg 1-4, leg 2-3, leg 1-2 and leg 4-3.

Dynamics of Chaotic Search

The stable dynamics of the system begins to fluctuate as μ increases, exhibiting a series of transient dynamics from quasiperiodicity to chaos. Fig. 5 shows the chaoticity of the system with different control parameters. In the higher chaotic regime complex transitory dynamics similar

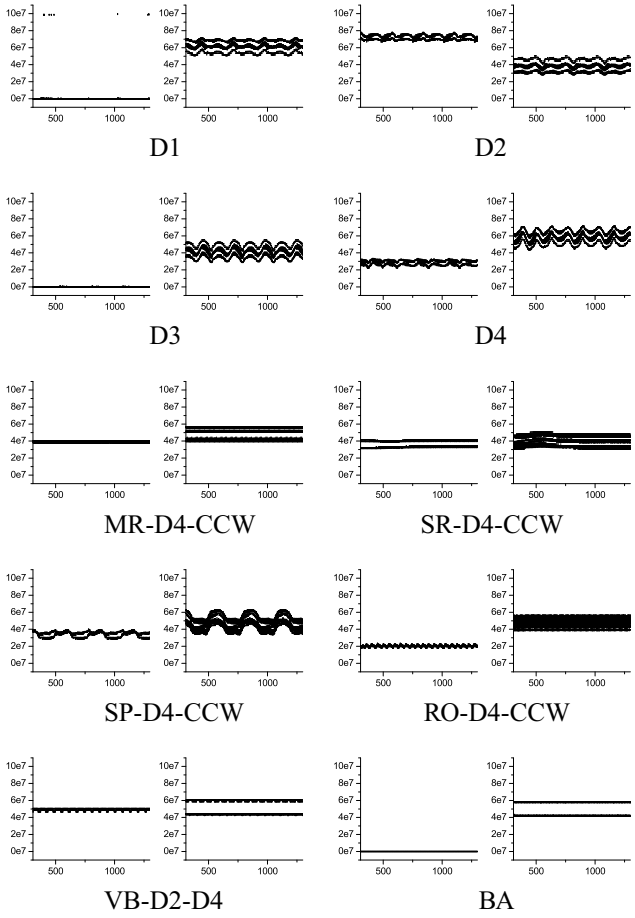


Figure 4: Limb cycle number vs. feature indices. In each pair of plots, a phase plot is on the left and an amplitude plot is on the right. From D1 to D4 are plots of straight locomotion in each direction. The next four plots are from the circular movements whose body orientation are D4 and rotating direction are counterclockwise. The last two plots are for vibration and bound antiphase.

to chaotic intermittency occurs which drives the system to briskly explore the phase space. To see the effect of chaotic search, the distributions of visits to each of the behaviour identified in Table 1 was investigated under the presence and absence of chaotic search. 100 simulations were performed for each case and the visiting counts of seven major behaviours were recorded. Fig. 6 shows a clear difference between the visiting ratios of the two cases, suggesting the effectiveness of chaotic search (B and C) which tended to settle on effective straight motion. In the search with fixed desired performance (Fig. 6B) any pattern below the criteria did not appear while the case of flexible E_d (Fig. 6C) shows a wider range of behaviours although the highest performing patterns still dominate. During the search process all variables and control parameters vary continuously as parts of the neuro-body-environment system, and the time evolution plots (Fig. 7) show that the stabilisation and destabilisation of the system occurs repeatedly in a trial-and-error manner

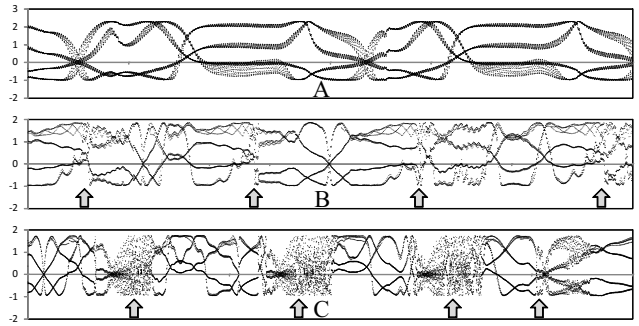


Figure 5: Poincare plots of the output of oscillators which correspond to the flexor neurons for legs 0 and 1 with different value of μ ((A) 0.2, (B) 0.34, (C) 0.346). We can see weak and strong chaotic intermittencies (the regions indicated by arrows) in high μ (B,C) while there is smooth and periodic transition of phase relationships in A.

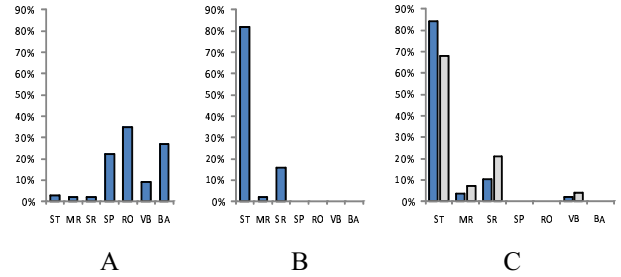


Figure 6: Visiting ratio distribution. (A) No chaotic search. (B) Search with $E_d = 0.2$. (C) Search with adaptive E_d as in Equation 11. Lighter shaded bars indicate visiting ratios in exclusion of ST-D4 pattern through the deep-path (see text).

until it settles on an effective form of locomotion.

Bad-Lock and Deep-Path

Although the system exploits chaotic dynamics for the exploration of motor patterns, unwanted synchronisation between chaotic movements of limbs, resulting in low performance (*bad-lock*), can arise from some initial conditions. In the case of fixed E_d, a local minimum was occasionally observed in which the system dynamics are locked in a narrow range of phase differences while the precise values of variables vary chaotically (Fig. 8). Although this is undesirable for the purpose of this work, it should be noted that this phenomena is observed in real biological systems (e.g. in walking and heartbeat rhythms). The bad-lock phenomena occurred more frequently if we set μ_c below the onset of chaos, indicating that the system has less exploratory 'perturbation force' when using low chaoticity.

Adaptive E_d was successful in enabling the goal seeking strategy for the unknown robotic system, as well as suppressing the bad-lock local minima outlined above by introducing an additional slow variable to the system. However another kind of deficiency, so called *deep-path*, was sometimes observed in this case. This involves the orbit becoming entrained in some periodic state for a long time before it eventually reaches the desired state (Fig. 9). This is due to the time spent in the chaotic regime becomes very short because the difference between E and E_d is too small, resulting in the system taking a long time to escape from the local minimum. The possibilities of bad-lock and deep-path always exist because the system is fully deterministic without stochastic sources, but it should be possible to reduce them by using more sophisticated goal seeking strategies.

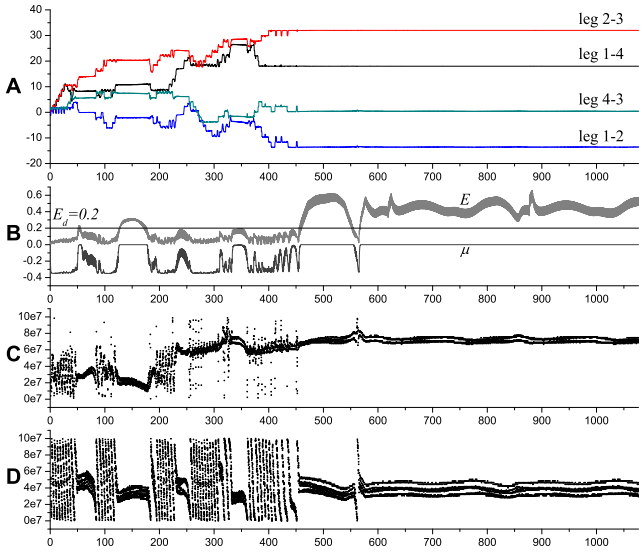


Figure 7: Time evolution of the search process. (A) Unwrapped phase differences between legs. (B) Performance and control parameters. (C,D) Phase and amplitude feature indices.

Physical Stability for Open Loop Control

Previous work on embodied coupling (Pitti et al., 2009) showed that the causal information flow between the controller and physical system is highly biased toward the sensor-to-motor direction, suggesting the controller strongly exploits the body-environment dynamics. Since the neuro-body-environment system used in the current paper is weakly coupled only through physical embodiment it can be inferred that the emergence of movement patterns is highly influenced by the dynamic stability of locomotion. Therefore we hypothesise that the more dynamically stable movement patterns remain longer as coherent states. A previous study (Iida and Pfeifer, 2004) provide the evidence that the intrinsic body dynamics of a properly designed controller-body system can self-stabilise into a periodic locomotion pattern without any sensory input. From the experiment in our study, we have shown that chaotic search of locomotion using a bodily coupled system is capable of naturally finding such stable patterns. This feature, together with the ready-built servo controller means the robot should be able to perform stable locomotion in an open-loop manner without any sensory information. This accomplishes "cheap" locomo-

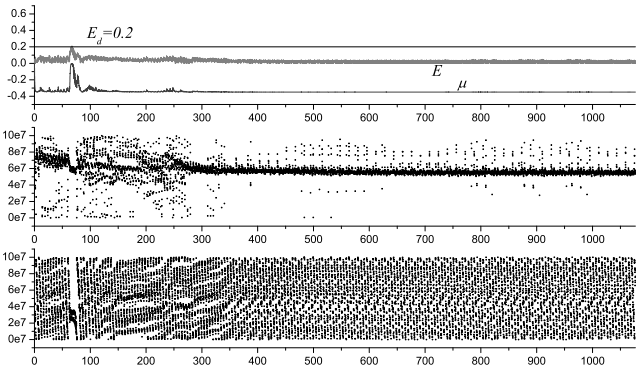


Figure 8: Local minima with fixed $E_d = 0.2$. The phase feature index plot (middle) indicates that the behaviour is locked around the vibrating (VB) pattern while the amplitudes fluctuate periodically.

tion, meaning that we should be able to readily capture a wide range of useful transient patterns which appear during the search process without being stabilised.

We tested this using a 'damaged' version of the robot by removing one of its fins, where there is no stable pattern when $\mu = 0$ but there exist a series of useful transient patterns. The chaotic search process was run for the 3-fin swimmer, and if some high performing pattern appeared the sensory input was gradually decayed to zero. We call this process *pattern capturing* for open-loop control rather than *acquiring* because it does not deal with the cortical memorisation of discovered patterns. The time course of the search process of the damaged robot (Fig. 10a) shows multiple transient patterns appear for a while, with high performing patterns among them. After the sensory inputs are removed the captured pattern is stably retained, providing fast locomotion; successful open-loop control is achieved. In order to see the dynamic stability of the captured behaviour, an external perturbation was applied by exerting random forces to each of fins (Fig. 10c). The stability of locomotion was remarkable, as the robot maintained a good locomotion performance even when the perturbation strength was over 200% of the average hydraulic force the fin receives during normal locomotion.

Discussion

We have modelled and investigated the emergent behaviours of a neuro-body-environment system coupled indirectly through physical embodiment and have shown the efficacy of exploring useful motor patterns by applying a novel chaotic search method. The whole system is treated as a single high dimensional continuous dynamical system containing intrinsic chaos as a necessary driving force for the exploration of its own dynamics. The search process was completely deterministic, and was able to selectively entrain the system orbit to one of the patterns by imposing goal directedness toward a desired behaviour. The emergent locomotion behaviours involved inherently stable physical dynamics, enabling stable open-loop control without a need for sensory information.

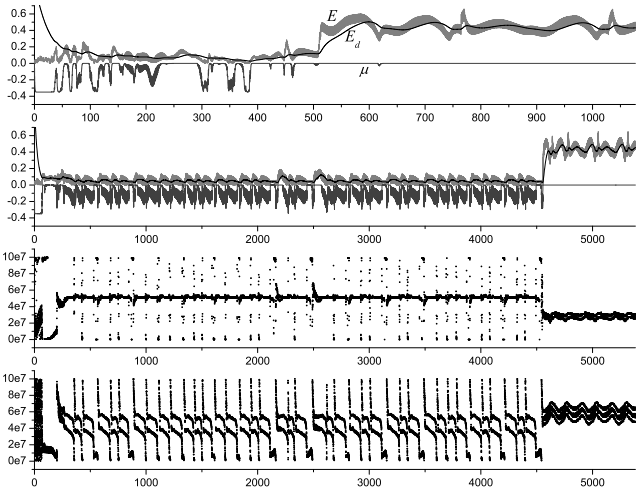


Figure 9: Deep path in the search process with adaptive E_d. The uppermost graph is an example of the typical search process, and the lower three graphs show the deep-path. The system is locked in a periodic state for a long time (see the time length) with very short duration of chaotic perturbation then eventually stabilises on the straight locomotion.

The method has been tested with a simple underwater robot, but it is generally applicable to a wide range of different robot morphologies and physical environments. However, further analysis is necessary in order to determine the optimum values of various parameters used in the search process. For example, the time scales of slow dynamics such as evaluation, goal seeking and feedback bifurcation (τ_E,τ_d,τ_μ) influence the search performance as well as the probability of being trapped in a local minima. Preliminary results of investigating the effect of different time scales revealed that the ratio between the time scales for evaluation and goal seeking determines the balance between the 'memorising' and 'forgetting' of patterns during the search process, implying there might be an optimal ratio which allows the system to stay in the chaotic regime for an optimal duration enabling fast search with less local minima. Another crucial factor which influences the system is the amount of bandwidth resulting from the design of body-environment interactions. In the case of the 4-fin swimmer presented here, the functional coupling strength between motor units varies with the body mass. Increased body mass will result in an increased moment of inertia which causes less transmission of the hydraulic force on one leg to the others, and vice versa. Similar effect will be caused by decreasing the density of the surrounding fluid or by increasing fin stiffness.

Our method is also applicable to terrestrial robots where a torque sensor is used for the sensory information. A few examples of initial results of our method applied to other

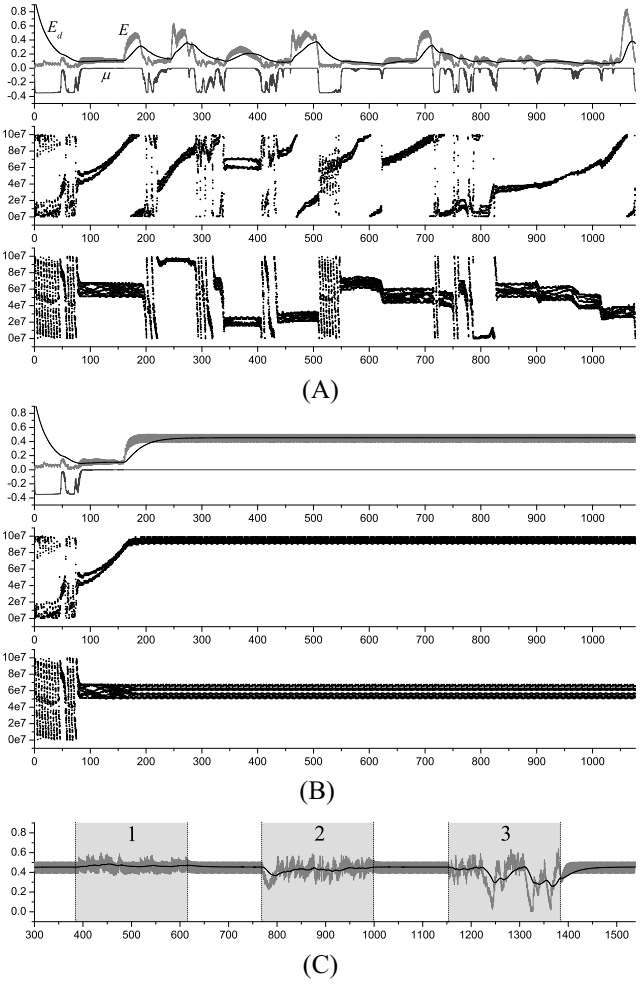


Figure 10: Capturing a transient locomotion pattern. (A) Normal behaviours of damaged robot. (B) Captured pattern by cutting sensory inputs. The initial condition is same as (A). (C) Stability of captured locomotion under perturbations. Over three equal time intervals random force vectors (N) whose strength were in ranges (1)[-0.1, 0.1], (2)[-0.5,0.5], (3)[-1,1] were exerted on each fin. The typical hydraulic force that a fin receives is around $\pm 0.3N$.

kinds of robots can be found in supplementary movie clips (http://email.kebi.com/∼necromax/explore.html). Although the movement patterns produced by our work can deviate from perfect patterns for highly adaptive locomotion, we believe it can make an important contribution as a basic exploratory element in more complex robotic system - such as providing supervisory pattern for the learning of locomotor CPGs.

References

Aida, T. and Davis, P. (1994). Oscillation mode selection using bifurcation of chaotic mode transitions in a nonlinear ring resonator. *IEEE Transactions on Quantum Electronics*, 30(12):2986–2997.

Asai, Y., Nomura, T., Sato, S., Tamaki, A., Matsuo, Y., Mizukura, I., and Abe, K. (2003). A coupled oscillator model of disordered interlimb coordination in patients with parkinson's disease. *Biological Cybernetics*, 88:152–162.

Barlas, Y. and Yasarcan, H. (2006). Goal setting, evaluation, learning and revision: A dynamic modeling approach. *Evaluation and Program Planning*, 29(1):79–87.

Buchli, J., Righetti, L., and Ijspeert, A. (2006). Engineering entrainment and adaptation in limit cycle systems. *Biological Cybernetics*, 95(6):645–664.

Davis, P. (1990). Application of optical chaos to temporal pattem search in a nonlinear optical resonator. *Japanese Journal of Applied Physics*, 29:L1238–L1240.

Ekeberg, O. (1993). A combined neuronal and mechanical model of fish swimming. *Biological Cybernetics*, 69:363–374.

Feudel, U. and Grebogi, C. (1997). Multistability and the control of complexity. *Chaos*, 7:597–603.

Iida, F. and Pfeifer, R. (2004). "cheap" rapid locomotion of a quadruped robot: Self-stabilization of bounding gait. In et al., F. G., editor, *Intelligent Autonomous Systems 8*, pages 642–649. IOS Press.

Kaneko, K. (1994). Relevance of dynamic clustering to biological networks. *Physica D*, 75(3):55–73.

Kuniyoshi, Y. and Sangawa, S. (2006). Early motor development from partially ordered neural-body dynamics: Experiments with a cortico-spinal-musculo-skeletal model. *Biological Cybernetics*, 95:589–605.

Kuniyoshi, Y. and Suzuki, S. (2004). Dynamic emergence and adaptation of behavior through embodiment as coupled chaotic field. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, pages 2042–2049.

Morihiro, K., Isokawa, T., Matsui, N., and Nishimura, H. (2008). Effects of chaotic exploration on reinforcement learning in target capturing task. *International Journal of Knowledge-based and Intelligent Engineering Systems*, 12(5):369–377.

Nara, S. and Davis, P. (1992). Chaotic wandering and search in a cycle-memory neural network. *Progress of Theoretical Physics*, 88(5):845–855.

Ohgane, K., Ei, S., and Mahara, H. (2009). Neuron phase shift adaptive to time delay in locomotor control. *Applied Mathematical Modelling*, 33(2):797–811.

Ott, E., Sauer, T., and Yorke, J. (1994). *Coping with chaos: Analysis of chaotic data and the exploitation of chaotic systems*. John Wiley & Sons, Inc.

Pfeifer, R. and Iida, F. (2004). Embodied artificial intelligence: Trends and challenges. *Embodied Artificial Intelligence*, 3139:1–26.

Pitti, A., Lungarella, M., and Kuniyoshi, Y. (2009). Generating spatiotemporal joint torque patterns from dynamical synchronization of distributed pattern generators. *Frontiers in Neuro-Robotics*, 3(2):1–14.

Raftery, A., Cusumano, J., and Sternad, D. (2008). Chaotic frequency scaling in a coupled oscillator model for free rhythmic actions. *Neural Computation*, 20(1):205–226.

Smith, R. (1998). Open dynamics engine (ode). http://ode.org/.

A Morphogenetic Approach to Self-Reconfigurable Modular Robots using a Hybrid Hierarchical Gene Regulatory Network

Yan Meng[1], Yuyang Zhang[1] and Yaochu Jin[2]

[1]Department of Electrical and Computer Engineering
Stevens Institute of Technology, Hoboken, NJ 07030, USA
[2]Department of Computing, University of Surrey, Guildford, Surrey GU2 7XH, UK
yan.meng@stevens.edu, yzhang14@stevens.edu, yaochu.jin@surrey.ac.uk

Abstract

In this paper, we present a morphogenetic approach to self-reconfiguration of a lattice-based simulated modular robot, CrossCube, under dynamic environments. A hybrid hierarchical controller inspired by the embryonic development of multi-cellular organisms is proposed to form different patterns for modular robots to adapt to environmental changes. The first layer is a rule-based controller to generate a number of appropriate target patterns (i.e. configurations) for various environments. The second layer is a gene regulatory network (GRN) based controller to coordinate the modules of CrossCube to transform from its current pattern to the target pattern. This hybrid hierarchical control framework is distributed in the sense that each module makes its own decisions based on its local perception. The global behavior of modular robots emerges from the local interactions with the environment and between the modules. The simulation results demonstrate that the proposed system is efficient and robust in adaptively reconfiguring modular robots to adapt to the changing environment.

Introduction

Self-reconfigurable modular robots are autonomous robots with a variable morphology, where they are able to deliberately change their own shapes by reorganizing the connectivity of their modules to adapt to new environments, perform new tasks, or recover from damages. Each module is an independent unit that is able to connect it to or disconnect it from other units to form various structures/patterns dynamically. Compared with conventional robotic systems, self-reconfigurable robots are potentially more robust and more adaptive under dynamic environments.

Modular robots can be generally classified into two groups according to their geometric arrangements of the modules: the chain/tree-based architectures [16] [19] [21] [22] and the lattice-based architectures [5] [7] [8] [10] [13] [14] [17] [24] [25]. In the chain/tree-based architectures, the modules are connected in a topology of a chain or a tree, where the motion controls of the modules are executed sequentially. It is relatively easier to design and implement this kind of architectures. In the lattice-based architecture, robot modules are usually arranged and connected in 3D patterns, such as a cubical or hexagonal grid, and the motion control of modules are carried out in parallel. Therefore, compared to the chain/tree-based architectures, the lattice-based architectures are more flexible and efficient to form complex structures although the design and implementation of this kind of architectures are more difficult. From this point of view, lattice-based modular robots are more suitable for dynamic environments. However, most available lattice-based modular robotic systems only have basic locomotion controllers to reconfigure the modular robots to a few predefined patterns by following predefined sequences or rules which have been optimized by human operators as a global controller. These predefined rules or sequences cannot predict all the possible situations that may occur for modular robots under dynamic environments. Although self-reconfiguration is believed to be the most important feature of modular robots, the ability to adapt their configuration autonomously under environmental changes remains to be demonstrated.

Generally, centralized high-level controllers for lattice-based modular robots are vulnerable to system failures or malfunctions of robot modules. On the other side, decentralized controllers are more robust and flexible under dynamic and uncertain environments. However, the main challenge for distributed systems is that it is difficult to predict the emerging behaviors only from local interactions of individual agents; neither is it easy to design rules for local interactions to generate desired global behaviors. Therefore, the major challenge in developing a decentralized controller for self-reconfigurable modular robots is how to coordinate local behaviors of multiple modules to achieve the desired global patterns to adapt to current environmental situations. To this end, we turn our attention to biological systems. Biological systems, from macroscopic swarm systems of social insects to microscopic cellular systems, can generate robust and complex emerging behaviors through relatively simple local interactions subject to various kinds of uncertainties [9]. We are more interested in the morphogenesis procedure in multi-cellular organisms. During the morphogenesis, genes in each cell are expressed, resulting in various cellular functions. The expression of the genes is regulated by their own protein products as well as proteins produced by other genes in the same cell or neighboring cells through intracellular and intercellular diffusion, forming a gene regulatory network that can be described by a set of coupled ordinary differential equations.

The connection between reconfigurable modular robots and multi-cellular organisms appears straightforward. Each unit in modular robots can be seen as a cell, and there are similarities

in control, communication and physical interactions between cells in multi-cellular organisms and modules in modular robots. For example, control in both modular robots and multi-cellular organisms are decentralized. In addition, global behaviors of both modular robots and multi-cellular organisms emerge through local interactions of the units, which include mechanic, magnetic and electronic mechanisms in modular robots, and chemical diffusion and cellular physical interactions such as adhesion in multi-cellular organisms. Therefore, it is a natural idea to develop control algorithms for self-reconfigurable modular robots using biological morphogenetic mechanisms.

Inspired by the embryonic development of multi-cellular organisms [24], in this paper, we propose a morphogenetic approach to self-reconfiguration of a lattice-based simulated modular robot, CrossCube. Basically, each module of CrossCube has a flexible single cubic shape like Molecube [25] [15], which does not require much free space for modules to move around, similar to the mechanics of SUPERBOT [14] [4] and MTRAN [10] [11] [20]. In the high-level controller, a two-layer morphogenetic architecture is proposed. Layer 1 is pattern generation layer, which is a rule-based controller to generate appropriate patterns represented by look-up tables. Layer 2 is a gene regulatory network (GRN) based controller to reconfigure modules automatically to the target patterns generated from layer 1.

Recently, Stoy proposed cellular automata to control reconfiguration [17]. Both our method and Stoy's method used cellular mechanism to reconfigure the modular robots. However, there are some major differences between our work and his work. Our method is two-layer hierarchical method. In [17], one-layer approach was proposed which corresponds to layer 2 in our model. Layer 2 of our model uses priorities to assign the importance of the positions of the target pattern, which help to improve the balance of target formation. In addition, our proposed method can solve the dead-lock situations of the modules while [17] cannot.

The major contributions of this paper are listed as follows. (1) The mechanics of CrossCube enables highly flexible locomotion compared to that in existing lattice-based modular robots. (2) A hybrid hierarchical morphogenetic controller is proposed, which is a decentralized approach where each module makes its own decisions based on its local perceptions on the environment and interactions with its immediate neighboring modules. (3) The modular robots can autonomously choose an appropriate pattern based on the current environment and then automatically self-reconfigure itself to the target pattern. (4) The proposed system is very robust to system failures.

The rest of the paper is organized as follows. The basic mechanics and locomotion design of CrossCube are described at first, followed by a brief introduction to biological morphogenesis. Then the proposed morphogenetic approach to self-organization of modular robots is presented. Various simulation results on evaluating the proposed morphogenetic approach to modular robots under dynamic environments are described. The paper concludes with a short summary of the current results and future work.

CrossCube – A Simulated Modular Robot

CrossCube is a simulated modular robot we developed in a robot simulator using a real time physics engine PhysX. The detailed information on the simulator will be discussed in the simulation section. CrossCube adopts a lattice-based cube design. Each module is a cubical structure having its own computing and communication resource and actuation capability. Like all modular robots, the connection part of the modules can easily be attached to or detached from other modules. Each module can perceive its local environment and communicate with its neighboring modules using on-board sensors.

Each CrossCube module consists of a core and a shell as shown in Fig. 1(a). The core is a cube with six universal joints. Their default heading directions are bottom, up, right, left, front, and back, respectively. Each joint can attach to or detach from the joints of its neighbor modules. The axis of each joint can actively rotate, extend, and return to its default direction and length.

The cross-concaves on each side of the shell restrict the movement trajectory of the joints, as show in Fig. 1(a). The borders of each module can actively be locked or unlocked with the borders of other modules, as shown in Fig. 1(b).

Basic motions of modules in CrossCube include rotation, climbing and parallel motion. Fig. 1(c) illustrates a rotation movement of two modules. Parallel motion means that a module moves to a next position which is parallel to its current position. During a parallel motion, a module moves from its current position to a parallel position Climbing motion means that a module moves to a diagonal neighboring position. Parallel motion and climbing motion allow a module of CrossCube to move to any position within the modular robot as long as the modules are connected. Since the major focus of this paper is the self-reconfiguration control algorithm, the detailed mechanical design of CrossCube is skipped here.

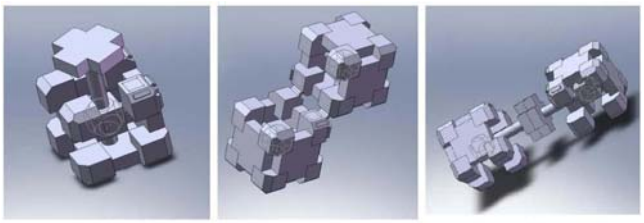


Figure 1: Mechanical demonstration of CrossCube. (a) The joints; (b) The locks on the boundaries of the modules. (c) Rotation and extension of the joints of the modules.

Morphogenetic Approach

Multi-Cellular Morphogenesis

Multi-cellular morphogenesis is under the control of gene regulatory networks. When a gene is expressed, information stored in the genome is transcribed into mRNA and then translated into proteins. Some of these proteins are transcription factors that can regulate the expression of their own or other genes, thus resulting in a complex network of

interacting genes termed as a gene regulatory network (GRN). To understand the emergent morphology resulting from the interactions of genes in a regulatory network, reconstruction of gene regulatory pathways using a computational model has become popular in systems biology [1]. A large number of computational models for GRNs have been suggested [2], [3], which can largely be divided into discrete models, such as random Boolean networks and Markovian models, and continuous models, such as ordinary differential equations and partial differential equations. Sometimes, GRN models also distinguish themselves as deterministic models and stochastic models according to their ability to describe stochasticity in gene expression. Note that in artificial life, a few high-level abstraction models have also been used for modeling development, such as the L-systems [12] and grammar trees [6].

The Hierarchical Framework

The metaphor between reconfigurable modular robots and multi-cellular organisms is straightforward. We can treat each module in modular robots as a single cell. And the similarities in control, communication and physical interactions between cells in multi-cellular organisms and modules in modular robots are obvious. For example, the control in both modular robots and multi-cellular organisms in decentralized. Furthermore, the global behaviors of both modular robots and multi-cellular organisms emerge through local interactions of the units, which include mechanic, magnetic and electronic mechanisms in modular robots, and chemical diffusion and cellular physical interactions such as adhesion in multi-cellular organisms.

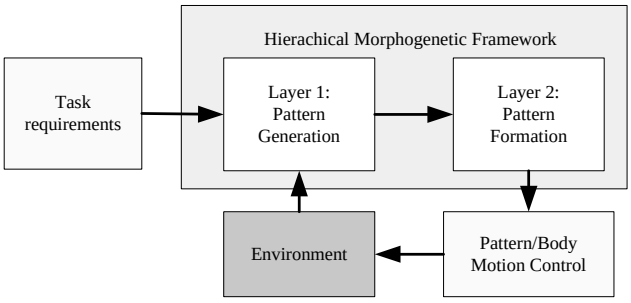


Figure 2: The block diagram of the hierarchical framework for the morphogenetic approach.

Based on this metaphor, a hybrid hierarchical morphogenetic approach is developed in this paper for self-reconfiguration of modular robots. First, the target pattern (i.e. final configuration) that a modular robot needs to form has to be generated automatically based on the current environments and mission at hand using some heuristic rules, which is the layer 1 controller of the hierarchical framework. Then, the modules in a modular robot need to self-organize themselves to form the target pattern generated by layer 1 using a GRN-based controller, which is the layer 2 controller. Fig. 2 shows the block diagram of this hierarchical GRN framework. Each unit of the modular robots contains a chromosome consisting of several genes that can produce different proteins. The local communications between the modules can be setup by diffusing the proteins into neighboring modules. The concentration of the diffused proteins decays over time and distance.

Layer 1: Pattern Generation

Adaptation to environmental changes is of paramount importance in reconfigurable modular robots. A mechanism is needed to define and modify the target configuration of the modular robot adaptively. Adaptation of the global configuration of the modular robot, i.e., change in morphogen values, can be triggered by local sensory feedback. For such tasks, it is assumed that each module is equipped with a sensor to detect the distance(s) between the module and obstacle(s) in the environment. Once a module receives such sensory feedback, this information will be passed on to its neighbors through local communication. In this way, a global change in configuration can be achieved.

The target pattern of the modular robot is defined by morphogen values of each grid. Grids are discretized from the space in which the modular robot is located. Each grid has the same size of with a robot module. The morphogen value can be either positive or negative. A positive morphogen value means that the grid should be occupied by a module, while a negative gradient suggests that the module in the grid, if any, should be removed. A higher value of morphogen value indicates a higher priority for the grid to be filled by a module.

For the sake of simplicity, a number of basic configurations for different environments can be represented in terms of a look-up-table for a given mission, for instance locomotion. An example of defining the configuration of a vehicle is provided in Table 1. In the table, x, y, and z are 3D coordinates of grid positions, MG denotes morphogen level and PID stands for position identification. Additionally, we define some joints' behaviors to enable the vehicle to move once the configuration is completed. Joints can be identified by its PID and RD means joint rotate direction.

Then the question is how to generate the look-up-table and decide the morphogen value for each position of a pattern under current environmental situations. A rule-based controller is developed for this purpose. In this paper, we only focus on the generation of some specific vehicle patterns to explain the basic ideas. We will investigate a more generic controller for different patterns in the future.

It is assumed that initially all robot modules know the heading direction of the vehicle pattern. When a robot needs to traverse a path whose width is narrower than that of the robot, the width of the front row will be first adapted to fit in the path. The remaining rows of the vehicle will be adapted row by row in a decentralized manner through local communication. The basic rules for this procedure can be summarized as follows:

- Rule 1: Once a module in the front row detects obstacle(s), it passes this information through local communication to its neighboring modules until all the modules are reset to the unstable state for initialization. Refer to the next section for a definition of different states of the robot modules.
- Rule 2: If some of the modules in the first row detect an obstacle, they will estimate whether the robot need to reconfigure itself to avoid the obstacle. If yes, these modules will estimate how many modules need to be removed and this information is passed to other modules in

the same row through local communication. Therefore, the mophogen gradients of these need-to-remove positions are set up as negative values while others as positive values. As a result, those positions with positive mophogen values are head of the new vehicle pattern.

- Rule3: After the GRN-based pattern formation controller finishes the reorganization of the modules in one row, the states of these modules are set to be 'stable'. If a row of a vehicle pattern is filled in by stable modules, these modules can set the positive morphogen values for the position in the next row. One exception is that if the module is used as a wheel for the vehicle pattern, the morphogen value of its next position should be set as negative because two neighboring wheel modules causes fault pattern.
- Rule4: The pattern generation procedure stops when all the modules change to the stable state.

Layer 2: Pattern Formation

By setting any single module as the origin, all other modules can figure out their relative positions to this origin easily through local communications. Based on the relative positions and the information on the target pattern, each module can produce different types of proteins to attract other modules to fill in its neighboring positions with positive morphogen values, or repel its neighbor modules from positions with negative morphgen values.

Finite States of Modules

The attraction and repellent behaviors of the modules are regulated by a GRN-based controller, which can adaptively set the state of the modules to one of the following five states, namely, 'stable', 'unstable', 'attracting', 'repelling', and 'repelled'. The transition relationships between the five states of modules are given in Figure 3.

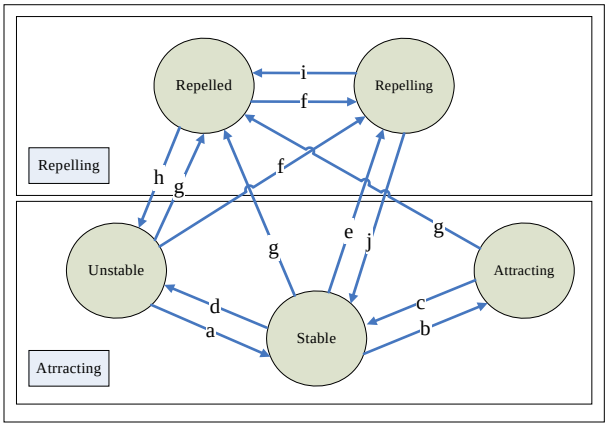


Figure 3: State transition of each module in CrossCube.

The "stable" state means the final state of the module. The "attracting" state means the module can attract other modules to fill in some of its neighboring positions. The "unstable" state means the module can respond to attractions. The "repelling" state means the module can repel specific neighboring modules away. The "repelled" state means that the module responds to repelling requests and move away from the current position.

When an 'unstable' module arrives at the destination position (grid), it changes its state to "stable" (arrow *a* in Figure 3). A 'stable' module can change its state to 'attracting' (arrow *b* in Figure 3) if it has neighboring positions with a positive morphogen value. When those neighboring positions are occupied by modules, the 'attracting' module returns to the 'stable' state (arrow *c* in Figure 3). A 'stable' module may also give up its current position so that it can fill in some more important positions in the pattern (with a high positive gradient) by turning its state to 'unstable' (arrow *d* in Fig. 2).

When the 'repelled' module moves away from its current position it switches its state to 'unstable' (arrow *h* in Figure 3). A module can be triggered to be 'repelling' state under two situations. First one is when a 'stable' module finds out that some of its neighboring modules are located in the positions with negative morphogen value, it changes its state to 'repelling' (arrow *e* in Figure 3) and switches the state of those neighbors to be 'repelled' (arrow *g* in Figure 3). When all the 'repelled' modules have left, the 'repelling' module returns to the 'stable' state (arrow *j* in Figure 3). The second situation is a deadlock situation. A deadlock happens when a module is blocked by its neighboring modules. To resolve this deadlock, the blocked module switches its state to be 'repelling' (arrow *f* in Figure 3), and trying to change the state of all its neighbors to be 'repelled' (arrow *g* in Figure 3). This removes some of its neighboring modules to make room for the blocked module to move away. Then the 'repelling' module turns back to the 'repelled' state (arrow *i* in Figure 3).

The state transitions are controlled by a GRN-based model having two gene-protein pairs: an attracting gene-protein pair (g_A, p_A) and a repelling gene-protein pair (g_p, p_p). We assume that the repellent states always have a higher priority than the attracting states. As a result, all the states triggered by the attracting behaviors can be overwritten by the states triggered by the repelling behaviors. The reason for this assumption is that the positions with a repelling (negative) morphogen value should be kept empty as long as migration modules are still in need during reconfiguration.

Gene-Protein Pair for Attraction

The attracting gene-protein pair (g_A, p_A) is used to control the transitions between 'attracting', 'stable' and 'unstable' states in Figure 3. Basically the expression level of g_A affects the state as shown in (1). And protein p_A will regulate g_A's expression level.

$$\text{state} = \begin{cases} \text{'unstable'} & \text{if } g_A < G_{A\_L} \\ \text{'stable'} & \text{if } G_{A\_L} < g_A < G_{A\_H} \\ \text{'attracting'} & \text{if } g_A > G_{A\_H} \end{cases} \quad (1)$$

where G_{A_L} is a negative threshold and G_{A_H} is a positive threshold.

At the initial stage of pattern formation, all modules are set as 'unstable'. After they are initialized with the target pattern and the relative position information to the origin, modules that are located in the grids with a positive morphogen value become 'stable'. A new 'stable' module initializes the gene expression level of its attracting gene g_A to zero.

Each 'stable' module generates attracting protein p_A for all of the empty neighboring grids having a positive morphogen value. The local generated p_A and received p_A from other modules will regulate the expression level of g_A. When g_A is high enough to trigger the module to be 'attracting', the local generated p_A will be diffused to other modules. During diffusion, the concentration of p_A are weaken by a fix rate each time when it enters a cell. Here, p_A is defined as

$$p_A^{ij} = \{AP^{ij}, M_A^{ij}\} \quad (2)$$

where p_A^{ij} is the attracting protein generated by *i*-th module for its *j*-th neighbor position. AP^{ij} is the position, and M_A^{ij} is the concentration of the protein p_A^{ij}, which is discounted from the morphogen value of AP^{ij} defined by layer 1 of the control framework.

The dynamics of regulation can be described by the following GRN model:

$$\frac{dg_A^i(t)}{dt} = -k_1 \cdot g_A^i(t) + k_2 \cdot \sum_j p_A^{ij} - k_3 \cdot \sum p_{A\_received} \quad (3)$$

where $g_A^i(t)$ is g_A's concentration of the *i*-th module. The first term indicates that $g_A^i(t)$ will decay over time. The second term represents the sum of all locally generated p_A by grid *i*. The more proteins a module (which is associated with grid *i*) generates for its empty neighboring grid, the higher the g_A expression level this grid will be, which means it will have better chance to change its state from "stable" to "attracting". Meanwhile, $g_A(t)$ will decrease if it receives p_A from other modules. The module may turn to 'unstable' if outer attraction is strong enough. $k_1, k_2,$ and k_3 are constant coefficients. Unstable modules choose the attracting position with the highest P_A from all the received attracting proteins to fill in, and move to the destination by following morphogen gradient. Once a module reaches its destination, it will become stable.

To summarize, the gene-protein pair (g_A, p_A) can regulate each other according to the GRN model described in Eqns. (1) and (3). More specifically, p_A can regulate g_A through Eqn. (3). Meanwhile, p_A can diffuse only if g_A is greater than G_{A_H} based on Eqn. (1).

Gene-Protein Pair for Repelling

The 'repelling' states are controlled by the repelling gene-protein pair (g_p, p_p). The repelling modules produce p_p, which is defined as

$$p_P^{ij} = \{RP^{ij}, M_P^{ij}\} \quad (4)$$

where p_P^{ij} is the repellent protein generated by *i*-th module for its *j*-th neighbor. RP^{ij} is the *j*-th repellent grid around *i*-th module, and M_P^{ij} is the concentration of the protein p_P^{ij}, which equals to a predefined positive constant. Each module has repelling gene whose concentration affects whether the module should change to 'repelled' state, that is, to respond to a 'repelling' module. The gene expression level of g_P is initialized as 0 and can be regulated by p_P through Eqn. (5)

$$\frac{dg_P^i(t)}{dt} = -k_4 \cdot g_P^i(t) - k_5 \cdot \sum p_{P\_rec} \quad (5)$$

state = repelled when $g_P < -MG^i$

where $g_p(t)$ is the gene expression level of the repellent gene at time *t*. p_{P_rec} is the concentration of the received repellent protein. MG^i is the morphogen value of the current position. k_4 and k_5 are constant coefficients. The first item denotes $g_p(t)$ will decays to zero along time. The second term indicates that when a module receives p_P, the concentration of g_P is reduced. Obviously modules with a lower morphogen value are more likely to be repelled.

To summarize, p_P can regulate g_P through Equation (5). g_P can produce p_P under the condition that g_P is below MG^i and the module is blocked.

Simulation Results

To evaluate the efficiency and robustness of the morphogenetic approach to the self-reconfiguration of CrossCube, several case studies have been conducted in a robot simulator, as shown in Figure 4. This simulator is used to simulate the behaviors and interaction of CrossCube with a physical world using C++ and the PhysX engine from nVidia (http://en.wikipedia.org/wiki/PhysX). In the following experiments, the system parameters are setup as follows: $k_1 = 0.7$, $k_2 = 1$, $k_3 = 1$, $k_4 = 0.5$, $k_5 = 2$, G_{A_L} = -1, G_{A_H} = 1, $G_{P\_L} = -2$, C_P^{ij} = 0.7. Protein concentration decays to 80% of its previous level when it diffuses into a neighbor module.

Case Study 1: Pattern Formation

To evaluate the performance of the GRN-based controller for pattern formation layer, first, we can predefine a fixed target pattern using a look-up table. For example a vehicle pattern, can be defined as Table 1.

Positions (X, Y, Z, MG, PID)		**Joints (PID1, PID2, RD)**
(0, 0, 0, 10, 0)	(1, 0, 3, 10, 10)	(0, 1, 0)
(1, 0, 0, 10, 1)	(2, 0, 3, 10, 11)	(2, 3, 1)
(2, 0, 0, 10,2)	(0, 0, 4, 10,12)	(6, 7, 0)
(3, 0, 0, 10, 3)	(1, 0, 4, 10, 13)	(8, 9, 1)
(1, 0, 1, 10, 4)	(2, 0, 4, 10, 14)	(12, 13, 0)
(2, 0, 1, 10, 5)	(3, 0, 4, 10, 15)	(14, 15, 1)
(0, 0, 2, 10, 6)	(0, 0, 1, -1, 16)	
(1, 0, 2, 10, 7)	(3, 0, 1, -1, 17)	
(2, 0, 2, 10, 8)	(0, 0, 3, -1, 18)	
(3, 0, 2, 10, 9)	(3, 0, 3, -1, 19)	

Table 1: Definition of a vehicle pattern for case study 1. In the table, x, y, and z are 3D coordinates of grid positions, MG denotes morphogen level and PID stands for position identification.

Based on this predefined target pattern, the modules of CrossCube modules need to autonomously configure themselves to form the target pattern using the GRN-based controller in layer 2. A set of snapshots of this pattern

formation procedure in the experiment is depicted in Figure 4. From Figure 4, we can see that the CrossCube can automatically form a given target pattern through self-reconfiguration using the proposed GRN-based controller.

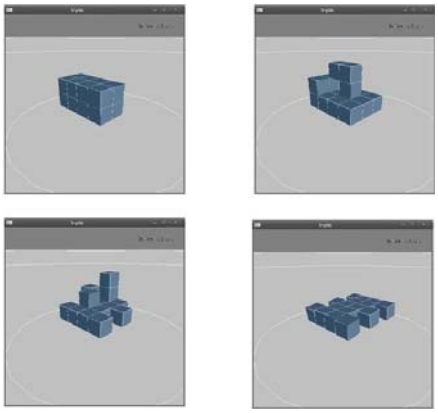


Figure 4: Autonomous reconfiguration of a CrossCube from a rectangle to a vehicle using the GRN-based model.

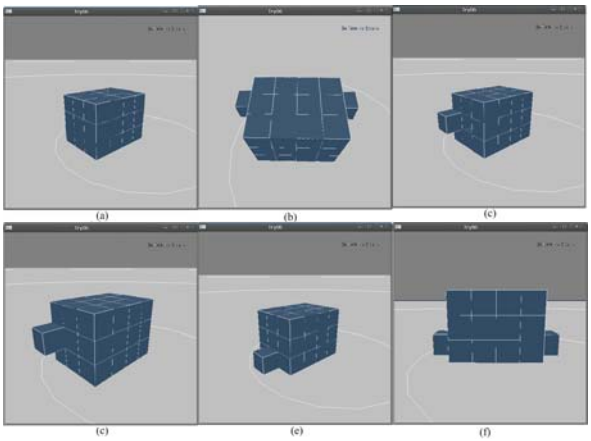


Figure 5: A set of snapshots for the simulation using the repelling feature of the GRN-based controller to resolve a deadlock problem. (a) The original pattern of the robot. (b)(c) Two modules are repelled by the central modules. (d)(e) The central modules move away from blocked positions. (f) The target pattern is finished.

Case Study 2: Resolving Deadlock

In this case study, a deadlock problem is resolved using the repelling function of the GRN-based controller in layer 2. Robot modules are initialized in a 4x3x3 solid cube, starting at (0, 0, 0) and ending at (3, 2, 2). The target pattern is predefined in Table 2 which is a center-empty box plus two additional modules at sides. To build the pattern, the modules in the center of the solid cube should move out the module that is blocked by the modules on surface. Then the GRN-based controller of layer 2 is conducted to solve the deadlock problem to form the target pattern. Figure 5 shows the successful procedure of solving this deadlock problem using this morphogenetic approach on CrossCube simulator. It is shown that the modules with lower morphogen value are repelled which is consistent with our design.

Positions (x, y, z)	**Morphogen value**
(-1, 0, 1), (4, 0, 1), (0, 1, 1), (3, 1, 1)	2
(1, 1, 1), (2, 1, 1)	-10
Other positions	10

Table 2 Definition of a vehicle pattern for case study 2

Case Study 3: Self-Repairing

One important feature of a reconfigurable modular robot is being able to dynamically self-repair itself from the malfunctions of modules or damaged modules. For example, if some of the modules are damaged, the remaining modules will release new attracting proteins to repel those damaged modules and attract existing modules in the positions with a low morphogen value to fill in the positions of the damaged modules. In other words, modules that are located in less important positions of the target pattern will automatically migrate to the positions originally occupied by the damaged modules with a higher morphogen value. To evaluate the self-repairing performance of the GRN-based control in layer 2, another experiment is conducted here. First, the look-up table for the target pattern (i.e., a vehicle patter here) is given in Table 3 as a fixed predefined layer 2. The bottom modules (y equals to 0) are functional modules in the vehicle pattern. The top modules (y equals to 1) are backup modules, which are used to repair the malfunctioned parts of the vehicle pattern. Therefore, the backup modules have a lower morphogen value than that of the functional modules.

When the vehicle is moving, an "explosion" occurs and some functional modules are blown away. The backup modules then automatically move to fill in the damaged modules. Figure 6 shows a snapshot of this self-repairing procedure using the proposed hierarchical framework on CrossCube modules. This experiment demonstrates that the proposed approach is efficient for self-repair of a modular robot in the presence of some failed modules.

Positions (X, Y, Z, MG, PID)		**Joints (PID1, PID2, RD)**
(0, 0, 0, 10, 0)	(0, 0, 4, 10,12)	(0, 1, 0)
(1, 0, 0, 10, 1)	(1, 0, 4, 10, 13)	(2, 3, 1)
(2, 0, 0, 10,2)	(2, 0, 4, 10, 14)	(6, 7, 0)
(3, 0, 0, 10, 3)	(3, 0, 4, 10, 15)	(8, 9, 1)
(1, 0, 1, 10, 4)	(0, 0, 1, -1, 16)	(12, 13, 0)
(2, 0, 1, 10, 5)	(3, 0, 1, -1, 17)	(14, 15, 1)
(0, 0, 2, 10, 6)	(0, 0, 3, -1, 18)	
(1, 0, 2, 10, 7)	(3, 0, 3, -1, 19)	
(2, 0, 2, 10, 8)	(1, 1, 1, 1, 20)	
(3, 0, 2, 10, 9)	(2, 1, 1, 1, 21)	
(1, 0, 3, 10, 10)	(1, 1, 2, 1, 22)	
(2, 0, 3, 10, 11)	(2, 1, 2, 1, 23)	

Table 3 Definition of a vehicle pattern for case study 3

Pattern Adaptation in a Changing Environment

To verify the efficiency and robustness of the rule-based controller for pattern generation, a transformable vehicle is

developed. During the pattern generation process, the positive morphogen value is set as 10 and the negative morphogen value is -10.

A set of snapshots showing the adaptation of the vehicle pattern to environmental changes is provided in Figure 7. First the pattern generation controller generates a vehicle pattern based on the width of path it needs to traverse using the rule-based method. As the vehicle is moving forward, a narrower path is detected. Consequently, a new vehicle pattern that can fit in this narrower tunnel is generated. Then steps are detected in front of the robot, new target patterns are dynamically generated to allow the robots to climb the steps, and eventually a new vehicle pattern is generated to continue its locomotion task after finishing the climbing. During this procedure, the GRN-based controller for pattern formation layer would automatically reconfigure the modules to form the new target patterns.

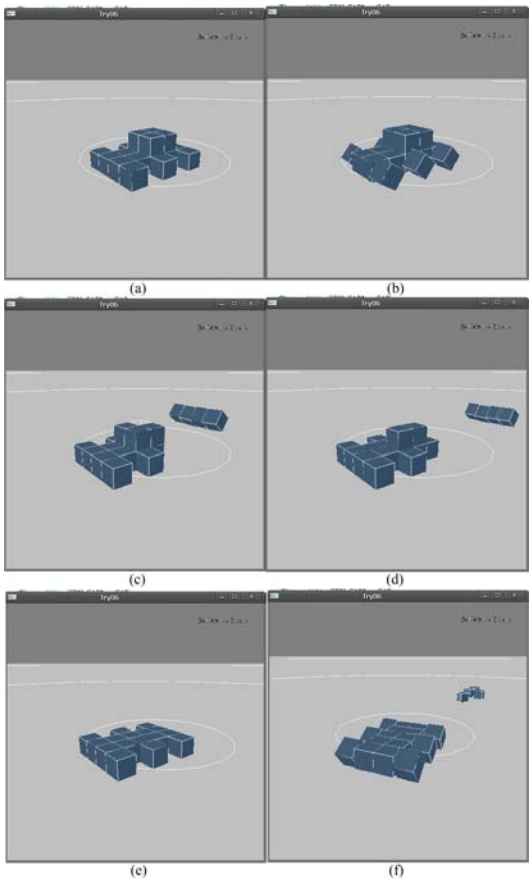


Figure 6: A set of snapshots of the self-repairing of CrossCube using the GRN-based controller. (a) A vehicle pattern is formed. (b) The vehicle pattern moves forward. (c) Some modules are blown off when the explosion happens. (d) The failed part is filled up by the backup modules. (e) The vehicle is repaired. (f) The repaired vehicle continues moving.

Conclusion and Future Work

In this paper, we presented a hybrid hierarchical approach to self-reconfiguration of a simulated modular robot, CrossCube, which is inspired by multi-cellular morphogenesis. First layer defines the desired configuration of the modular robots while the other layer organizes the modules autonomously to achieve the desired configuration. Such a hierarchical structure makes it possible to separate the control mechanisms for defining a target configuration from those for realizing it, similar to biological gene regulatory networks. In response to the environment changes, the layer for defining the robot configuration is able to adapt the target configuration, based on which the second layer can re-organize the modules autonomously to realize the target configuration.

The current system is only based on simulated modular robots with considerations of physical constraints. In the future, we will develop the real modular robots based on the current mechanical design. Furthermore, since the current design of the first layer is a heuristic rule-based method, it has some limitations to generate various patterns for dynamic environments, only some simply patterns are possible. In the future, we will investigate a more general approach for the design of layer 1 so that more general patterns can be automatically generated to adapt to various dynamic environmental changes.

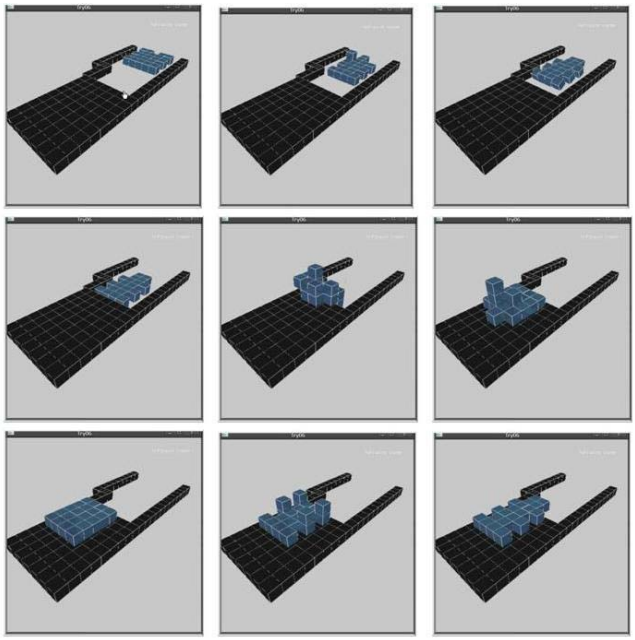


Figure 7. A set of snapshots demonstrating a series of reconfigurable processes during locomotion and climbing. The robot first adapted its width to the narrow path, then changed its configuration for climbing up a step, and finally reconfigured itself into a vehicle again to move forward.

Acknowledgements

This work was supported in part by Honda Research Institute Europe. The work was done while Yaochu Jin was with Honda Research Institute Europe.

References

[1] Alon, U. (2007). Network motifs: theory and experimental approaches, *Nature Review Genetics*, vol. 8, pp. 450–461.

[2] DeJong, H. (2002). Modeling and simulation of genetic regulatory systems: A literature review, *Journal of Computational Biology*, vol. 9, pp. 67–103.

[3] Endy D., and Brent, R. (2001). Modeling cellular behavior, *Nature*, vol. 409,pp. 391–395.

[4] Everist, J., Hou, F., and Shen, W. (2006). Transformation of Control in Congruent Self-Reconfigurable Robot Topologies, in Proc. of *International Conference on Intelligent Robots and Systems*.

[5] Gilpin, K., Kotay, K. and Rus, D. (2007). Miche: Modular Shape Formation by Self-Dissasembly, in Proc. of *IEEE International Conference on Robotics and Automation*.

[6] Gruau, F. (1993). Genetic synthesis of modular neural networks, in International Conferences on Neural Networks. *Morgan Kaufmann*, pp. 318–325.

[7] Jorgensen, M. W., Ostergaard, E. H., and Lund, H. H. (2004). Modular ATRON: Moduels for a self-reconfigurable robot, in Proc. of *IEEE International Conference on Intelligent Robots and Systems*.

[8] Kamimura, A., Kurokawa, H., Yoshida, E., Murata, S., Tomita, K., and Kokaji, S. (2004). Distributed adaptive locomotion by a modular robotic system, M-TRAN II, in Proc. of *IEEE International Conference on Intelligent Robots and Systems*.

[9] Kelly, K. (1994). Out of Control – The New Biology of machines, *Social Systems and Economic World. Basic Books.*

[10] Kurokawa, H., Tomita, K., Kamimura, A., Kokaji, S., Hasuo, T., and Murata, S. (2008). Distributed self-reconfiguration of M-TRAN III modular, *The International Journal of Robotics Research*, 27(3-4):373–386.

[11] Kurokawa, H., Tomita, K., Kamimura, A., Yoshida, E., Kokaji, S., and Murata, S. (2005). Distributed Self-reconfiguration Control of Modular Robot M-TRAN, in Proc. of *International Conference on Mechatronics and Automation.*

[12] Lindenmayer, A. (1968). Mathematical models for cellular interaction indevelopmental. Parts I and II, *Journal of Theoretical Biology*, vol. 18, pp. 280–315.

[13] Murata, S., Yoshida, E., Kurokawa, H., Tomita, K., and Kokaji, S. (2001). Self-repairing mechanical systems, *Autonomous Robots*, vol. 10, pp. 7–21.

[14] Salemi, B., Moll, M., and Shen, W. (2006). SUPERBOT: A Deployable, Multi-Functional, and Modular Self-Reconfigurable Robotic System, in Proc. of *IEEE International Conference on Intelligent Robots and Systems*.

[15] Moll, M., Will, P., Krivokon, M., and Shen, W. (2006). Distributed Control of the Center of Mass of a Modular Robot, in Proc. of *IEEE International Conference on Intelligent Robots and Systems.*

[16] Shen, W., Krivokon, M., Chiu, H., Everist, J., Rubenstein, M., and Venkatesh, J. (2006). Multimode locomotion for reconfigurable robots, *Autonomous Robots*, vol. 20, no. 2, pp. 165-177.

[17] Stoy, K. (2006), Using cellular automata and gradients to control self-reconfiguration, *Robotics and Autonomous Systems*, vol. 54, pp.135-141.

[18] Unsal, C., Killiccote, H., and Kholsa, P.K. (2001). A Modular Self-Reconfigurable Bipartite Robotic System: Implementation and Motion Planning, *Autonomous Robots*, vol. 10, pp.23–40.

[19] Yim, M., Eldershaw, C., Zhang, Y., and Duff, D. G. (2004). Limbless conforming gaits with modular robots, in Proc. of *International Symposium on Experimental Robotics.*

[20] Yoshida, E., Kurokawa, H., Kamimura, A., Tomita, K., Kokaji, A., and Murata, S. (2004). Planning Behaviors of a Modular Robot: an Approach Applying a Randomized Planner to Coherent Structure, in Proc. *IEEE International Conference on Intelligent Robots and Systems.*

[21] Yu, C.H., Haller, K., Ingber, D., and Nagpal, R. (2009). Morpho: A selfdeformable modular robot inspired by cellular structure, in Proc. of *International Conference on Robotics and Automation*.

[22] Yu, C., and Nagpal, R. (2009). Self-Adapting Modular Robotics: A Generalized Distributed Consensus Framework, in Proc. of *International Conference on Robotics and Automation.*

[23] White, P., Zykov, V., Bongard, J., and Lipson, H. (2006). Three Dimensional Stochastic Reconfiguration of Modular Robots, in Proc. of *Robotics: Science and Systems Conference*, pp. 161–168.

[24] Wolpert. L. (2002). *Principles of Development*. Oxford University Press.

[25] Zykov, V., Chan, A., and Lipson, H. (2007). Molecubes: An Open-Source Modular Robotics Kit, in Proc. of *International Conference on Robotics and Automation.*

Artificial Hormone Reaction Networks: Towards Higher Evolvability in Evolutionary Multi-Modular Robotics

Heiko Hamann, Jürgen Stradner, Thomas Schmickl, and Karl Crailsheim
Artificial Life Lab of the Department of Zoology, Karl-Franzens University Graz, Universitätsplatz 2, A-8010 Graz, Austria
heiko.hamann@uni-graz.at

Abstract

The semi-automatic or automatic synthesis of robot controller software is both desirable and challenging. Synthesis of rather simple behaviors such as collision avoidance by applying artificial evolution has been shown multiple times. However, the difficulty of this synthesis increases heavily with increasing complexity of the task that should be performed by the robot. We try to tackle this problem of complexity with Artificial Homeostatic Hormone Systems (AHHS), which provide both intrinsic, homeostatic processes and (transient) intrinsic, variant behavior. By using AHHS the need for predefined controller topologies or information about the field of application is minimized. We investigate how the principal design of the controller and the hormone network size affects the overall performance of the artificial evolution (i.e., evolvability). This is done by comparing two variants of AHHS that show different effects when mutated. We evolve a controller for a robot built from five autonomous, cooperating modules. The desired behavior is a form of gait resulting in fast locomotion by using the modules' main hinges.

Introduction

The (semi-)automatic synthesis of robot controllers with artificial evolution belongs to the software section of evolutionary robotics (Cliff et al., 1993). The main challenge in this field is the curse of complexity because an increase in the difficulty of the desired behavior results in a significantly super-linear increase in the complexity of its evolution. This is partially documented by the absence of complex tasks in the literature (Nelson et al., 2009). Additionally, in evolutionary robotics the cost of the fitness evaluation is rather high even in case of simulations, if the application of a physics engine (simulation of friction, inertia etc.) cannot be avoided. Another challenge is the appropriate choice of a genetic encoding (Matarić and Cliff, 1996) and the basic principle of the controller design as they define the designable fraction of the search space and the fitness landscape (non-designable fractions are induced, for example, by the environment or the task itself). While the search space should be kept small, the fitness landscape should be smooth with a minimum number of local optima. Experience shows that these two criteria are contradicting. We summarize this complex of challenges by the aim to 'strive for high evolvability'.

Concerning the problem of finding appropriate controller designs a pleasant trend can be observed in recent literature. The most prominent candidate is presumably the HyperNEAT design (Stanley et al., 2009; Clune et al., 2009). It is based on artificial neural networks (ANN) but combines the 'search for appropriate network weights with complexification of the network structure' (Stanley and Miikkulainen, 2004) through the generation of connectivity patterns. It has proven to have good evolvability combined with an adequate range of applications. Other promising, recent approaches tend to be more inspired by biology, in particular by unicellular organisms and endocrine systems. Examples showing good evolvability are the reaction-diffusion controller by Dale and Husbands (2010) and homeostasis and hormone systems based on GasNets (Vargas et al., 2009) and ANNs (Neal and Timmis, 2003). They indicate homeostasis as a prominent feature in successful adaptation to dynamic environments.

In this paper, we analyze a controller design called Artificial Homeostatic Hormone Systems (AHHS) that is based on hormones only and was introduced before (Hamann et al., 2010; Schmickl et al., 2010; Schmickl and Crailsheim, 2009; Stradner et al., 2010, 2009). AHHS is a reaction-diffusion approach. Sensory stimuli are converted into hormone secretions that, in turn, control the actuators. In addition, hormones interact linearly and non-linearly comparable to the hidden layer of ANN. The topology of this hormone-reaction network is not predefined. Such systems show homeostatic processes because they typically converge to trivial equilibria for constant sensor input. The sensory stimuli are basically integrated in form of hormone concentrations (a form of memory) and decomposed over time (oblivion). However, during a limited period of time (transient) after a stimulus they show also variant behavior, especially, if non-linear hormone-to-hormone interactions are applied. This way, explorative behavior of the robot is implemented that allows for the testing of many sensory-motor configurations. The concept of AHHS is related to

gene regulatory networks. However, here each edge has its own activation threshold and redundant edges with different activations between two hormones are allowed.

The desired main application of AHHS is multi-modular robotics (SYMBRION, 2010; REPLICATOR, 2010). In this field, autonomous robotic modules are studied, that are able to physically connect to each other, and can also establish a communication and energy connection. Hence, they form a super-robot called 'organism', that is able to reconfigure its body shape, see for example, Shen et al. (2006) or Murata et al. (2008). Therefore, the underlying idea of diffusion in our reaction-diffusion system is that hormones diffuse from robot module to robot module and establish a low-level communication. Following our maxim of trying to reach a maximum of plasticity we use identical controllers in each module independent of their position within the robot organism, so there is neither a controller nor a module specialization. This concept implements the focus of evolutionary robotics on modularity (among others) in terms of hardware and software (Nolfi and Floreano, 2004). Although we evolve cooperative behaviors by evolving a kind of self-organized role selection, there is no co-evolution.

In general, our approach is more organic in contrast to the typical symbolic approach (direct encoding of pitch, roll, yaw angles, use of pattern generators using Gaussian functions etc.). The biological inspiration is not practiced as an end in itself but rather introduces more robustness in computations and it allows the diffusion of such values from module to module (implementing implicit communication).

One focus of our current research track is to design fitness landscapes by using appropriate controller designs. We investigate possibilities of smoothing the fitness landscape by a sophisticated interaction between the controller design and the mutation operator. We test whether it is useful to maximize the causality of the mutation operator (i.e., small causes have small effects) by reducing the maximal impact to the organism's behavior. However, whether high causality is really desirable, is questionable (e.g., cf. Chouard (2010)).

The investigated scenario is a modular-robotics variant of gait learning in simulation. Initially, we connect five modules in a simple chain formation as the body formation itself is not yet in our focus. The task is to move as far as possible by utilizing the hinge in each module only (no wheels).

Artificial Homeostatic Hormone Systems

In AHHS, sensors trigger hormone secretions, which increase hormone concentrations in the robot. These hormones diffuse, integrate, decay, interact and finally, affect actuators. We have analyzed AHHS controllers in single robots before (Schmickl et al., 2010; Schmickl and Crailsheim, 2009; Stradner et al., 2010, 2009). In these cases, the robot's body was virtually divided into compartments that hold hormones and between which hormones diffuse. These compartments create a spatial context (embodiment) by associating sensors and actuators with explicit compartments (e.g., left proximity sensor and left wheel actuator are associated with the left compartment and hence depend only on hormone concentrations of this compartment). In the case of modular robotics, the subdivision of the robot organism is naturally defined by the modules themselves. A virtual compartmentalization is not necessary and hormones diffuse from module to module (see Fig. 1). A first small case study with organisms built from three modules was reported in (Hamann et al., 2010).

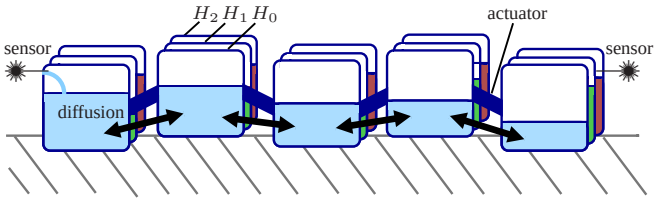


Figure 1: Sketch of the hormone dynamics and diffusion processes in an organism. Each module holds different hormones with different concentrations, hormones diffuse through the organism based on a diffusion coefficient evolved individually for each hormone, module locations (e.g., elevation) are not relevant for diffusion; sensor settings simplified, actually four proximity sensors per module.

AHHS1

We call the AHHS, initially presented in (Schmickl et al., 2010; Schmickl and Crailsheim, 2009), AHHS1. An AHHS consists of a set of hormones and a set of rules. On the one hand, it defines production/decay rates and diffusion coefficients for each hormone. On the other hand, it defines by rules the production through sensors and interaction of hormones as well as their influence on actuators. There are four types of rules. Sensor rules define the production of hormone through sensor input. Actuator rules define the control of actuators through hormone concentrations. Hormone rules define the interaction between hormones, that is, one hormone triggers the production of another hormone (or itself). Additionally, there is an idle rule to allow a direct deactivation of rules through mutations. Rules are triggered at runtime, if a certain threshold is reached (sensor values in case of sensor rules or hormone concentrations in case of hormone rules). The amount of produced hormone or the actuator control value are linearly depending on the controlling sensor or hormone respectively ('$\lambda x + \kappa$'). For more details see Schmickl et al. (2010).

AHHS2

Based on AHHS1 we designed an improved variant called AHHS2. The guiding principle of this improved controller design was to gain higher evolvability by creating smoother fitness landscapes. There were three main changes.

First, we introduced an additional rule type that implements nonlinear hormone-to-hormone interactions in the general form of $\Delta x/\Delta t = xy$, where x is the considered hormone concentration and y is the hormone concentration of the influencing hormone that triggers the considered rule. The idea is to increase the intrinsic dynamics (basically transient behavior before equilibria are reached) of the hormone network even without significant sensor input.
Second, a rule is not just triggered by exceeding or falling below a threshold but is linearly weighted within a trigger window (i.e., a tent function with a maximum of 1, defined by a center and a width, see eq. 2 below).
Third, the mutation of rule types in the form of discrete switches seemed to be too radical. This was overcome by introducing a concept of weights for rule types. Now, each rule can operate as any rule type at the same time. Each rule has a weight for each of the five rule types summing up to one (see Fig. 2). The influence of a rule type is proportional to its weight, for example, the sensor-rule aspect of a rule with a weight of 0.1 will produce only 10% of the hormone it would produce, if its weight would be 1, see $w_{\mathcal{L}}$ in eq. 1 below. A mutation will now only change two rule weights by reducing one by w and adding w to the other weight. In a well adapted controller we would expect that the weights of a rule are mainly concentrated on one or at most two rule types. Other weight distributions should be transitional only because specialization allows for better optimization.
The mathematical closed-form of this concept using the example of a linear hormone rule type is

$$\mathcal{L}(t) = w_{\mathcal{L}}\theta(H_k(t))(\lambda H_k + \kappa), \tag{1}$$

where $\mathcal{L}(t)$ is the hormone amount that is to be added to the considered hormone at time t, $w_{\mathcal{L}}$ is the weight of the linear hormone rule (see Fig. 2), k is the index of the input hormone and H_k is its concentration, λ is the dependent dose, κ the fixed dose. θ is called trigger function and defined by

$$\theta(x) = \begin{cases} \frac{1}{\eta}(\eta - |x-\zeta|) & \text{if } |x-\zeta| < \eta \\ 0 & \text{else} \end{cases}, \tag{2}$$

for trigger window center ζ and trigger window width η. For a more detailed introduction of AHHS2 and for a comparison of the AHHS approach to the standard ANN approach, see Hamann et al. (2010).
Note that the rule parameters (fixed dose, input hormone, trigger window etc.) are correlated via the rule types. For example, the input hormone is used for both the linear and the nonlinear hormone rule. If we would allow independent parameters for each rule type the genome (encoding of the controller) size would be increased by a factor of about three. This is a tradeoff in the complexity of the genome and, for example, a difficulty when analyzing the results. This is related to the completeness-vs-compactness challenge (Matarić and Cliff, 1996).

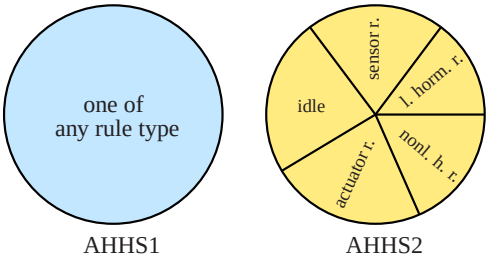


Figure 2: Rule type weights of the AHHS2 approach compared to AHHS1 (abbreviations: sensor rule, linear hormone rule, nonlinear hormone rule, actuator rule).

Investigated scenarios

Our main focus is on the field of modular robotics and our main concern is whether we are able to evolve fast locomotion in the gait learning task. Still, we tested the AHHS approach also in an inverted pendulum task as well, due to its lower computational complexity.

Inverted pendulum

In addition to the gait learning task, we tested the AHHS approach in a task that is easier to handle: balancing the inverted pendulum (see Fig. 3). The computational demand of the gait learning task is very high due to the sophisticated simulation of physics. We satisfy the need for a simulation of lower computational complexity by introducing the inverted pendulum task. Higher statistical significance of the results can be reached within reasonable time of computation. The original inverted pendulum is only slightly related to a real robotic task. Therefore, we adapted it to our requirements. The sensors are noisy (equally distributed and uncorrelated in time, $\pm 2.3\%$) and sampling rates of sensors are low which is documented by the relation between the cycle length τ and the maximal angular velocity of $0.05\pi[1/\tau] = 9°[1/\tau]$. The pendulum can move up to $9°$ between two calls of the controller. The controller has little time to adapt to new configurations. Furthermore, the sensors do not deliver actual angles and positions directly but partitioned onto several sensors and also relative rather than absolute (distance to wall instead of the crab's position etc.). The AHHS controls two outputs, left actuator A_0 and right actuator A_1, while the speed control of the crab is determined by their difference. The pendulum is started in the lower equilibrium position, so the nonlinear up-swinging phase is included. Combined with the sensor noise it is impossible for the controller to balance the pendulum in the upper equilibrium position. So the task stays dynamic and the controller is exposed to new situations constantly. The fitness function is the summation over all time steps of the angular distance to the top position in radians.

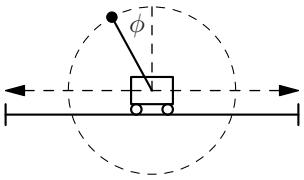


Figure 3: Inverted pendulum, pendulum free to move full 360° mounted on the crab that moves in one dimension (left/right) bounded by walls.

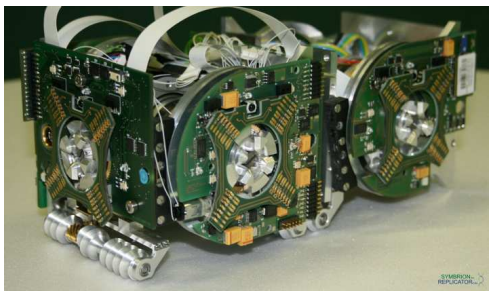


Figure 4: Two connected prototypes of the projects SYMBRION (2010) and REPLICATOR (2010).

Gait learning in multi-modular robotics

Gait learning in legged robotics is a commonly studied task in evolutionary robotics as reported by Nelson et al. (2009). However, here we investigate gait learning in multi-modular robotics. Each module consists of one hinge and we connect five modules. These five hinges are controlled decentrally although the modules have a low-level communication channel by means of diffusing hormones.

In contrast to the standard tasks of gait learning and collision avoidance, the challenge of gait learning in multi-modular robotics is more complex. The resulting gait is emergent due to the decentral and cooperative control of the actuators. In addition, there are several conceptionally different solutions, that is, different techniques of locomotion with good performance (e.g., caterpillar-like, erected walk, small jumps).

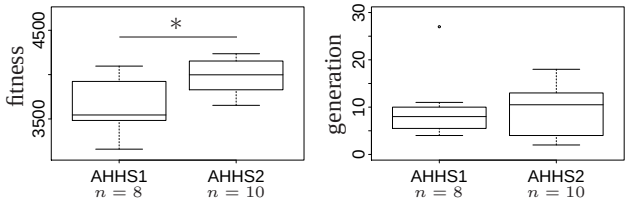
In each module the same controller is executed. Therefore, the gait learning task includes several sub-tasks. The organism has to break the symmetry (head and tail), synchronize through collective cooperation, and start moving into a common direction. This synchronization aspect is similar to the gait learning task for a legged robot with HyperNEAT by Clune et al. (2009).

All of this work is based on simulations as the actual hardware is not yet available (see Fig. 4 for a current prototype of Symbrion and Replicator (SYMBRION, 2010; REPLICATOR, 2010)). We use the simulation environment Symbricator3D by Winkler and Wörn (2009) that was developed for these projects. We use the current prototype design in the simulation (imported CAD data) as described in (Levi and Kernbach, 2010). However, we simplified the sensor setting to four proximity sensors (equally distributed around the robot shifted by 90 degrees: upwards, forwards, downwards, backwards). Symbricator3D is based on the game engine Delta-3D and currently uses the Open Dynamics Engine for the simulation of dynamics. The simulation of friction and momentum is important because the evolved gait behaviors rely on them. A drawback is that high computational complexity limits the number of evaluations in our evolutionary runs. We are interested in systems that evolve useful behaviors within a few hundred generations and with small populations (order of 10).

We have tested the AHHS controllers with two variants of the simulation framework. In the first version, the forces in the joints, that connect the modules, were damped and small displacements of the modules at the joints were allowed (i.e., simulation reacts moderately to big forces). It turned out that caterpillar-like locomotion was favored because the damped joints support wave motion. In the second version, the joints were fully fixed. In this version of the simulation the evolution of locomotion is more difficult which will be reflected by the best fitnesses in the following.

We start the scenario with five robot modules which are simply connected in a chain. Initially this robotic organism is placed in the center of the arena. In order to increase the complexity of the gait learning task, the central area is surrounded by a low wall forming a square (its height is about half the height of a robot module). Outside the wall several cubes are placed that could only be sidestepped by the organism. An identical robot controller is uploaded to the memory of all five modules. The robot modules have to figure out their position (their role within the configuration), that is, they have to break the symmetry of the configuration in order to generate a coordinated gait. This is, for example, possible because of different outputs of proximity sensors depending on the modules' positions. There are three classes of modules defined by their characteristic sensor inputs: front module, back module, and modules in between. We use identical controllers because we want to apply them to dynamic body shapes in our future work and also a single module should have all functionality. Hence, uploading heterogeneous controllers with predefined roles would not be an option. In addition, using self-organized role assignment will allow for high scalability (using the same controller for different body sizes), plasticity (reorganization of roles in changing body shapes), and new role types might emerge that were unthought of by the human designer.

The fitness is defined by the covered distance of the organism. It is an aggregate fitness function (Nelson et al., 2009) that evaluates the organism's performance as a whole. Although the organisms might achieve advancements early in the evolutionary run, there is a bootstrapping problem. For example, the downward proximity sensors will not give significant input until the organism has figured out how to erect the modules in the middle. In addition, controllers cannot



(a) fitness (Wilcoxon $p < 0.05$) (b) generation

Figure 5: Inverted pendulum, AHHS1 with 60 rules, AHHS2 with 15 rules, comparison of fitness and evolution speed (generation when 75% of max. fitness was reached).

evolve special techniques to climb the wall before they have actually managed to move the organism there to explore it.

Results and discussion

Inverted pendulum

The evolutionary runs of the inverted pendulum were performed with a population of 200 randomly initialized controllers. The AHHS was set to 15 hormones. For AHHS1 60 rules were used and 15 for AHHS2. The runs were stopped after 200 generations. Linear proportional selection was used and elitism was set to one. The mutation rate was 0.15 per gene with a maximal, absolute change of range 0.1. The recombination (two-point crossover) rate was 0.05.

For this task we configured AHHS with a left and a right compartment. The left compartment incorporates the left actuator A_0, the left proximity sensor, the sensors giving the angles of the pendulum when it is in the left half etc. and for the right compartment respectively.

The comparison of the best controllers of each run is shown in Fig. 5(a). In this scenario, AHHS2 performs significantly better than AHHS1 although in terms of evolution speed there is no significant difference (see Fig. 5(b)). The AHHS2 design is the better choice in this task. The cause of the advantage of AHHS2 over AHHS1 in this task compared to the indistinct situation in the gait learning task is unclear. In future studies we will investigate whether this trend will also be observed in more complex tasks from the domain of multi-modular, evolutionary robotics.

One of the best evolved AHHS2 controllers showing interesting behavior is analyzed in the following[1]. While it is not possible to keep the pendulum in the upper equilibrium for longer time due to noise, the controller still tries to maximize the time the pendulum is close to the upper equilibrium mostly by small displacements of the crab. The controller is mainly based on one hormone (H_0), and four rules (see Fig. 6). Sensor S_0 reaches its maximum, if the pendulum approaches $\phi = 0$ (top position) from the left. It triggers small displacements of the crab to the right, a behavior that keeps the pendulum turning counterclockwise with slow passes at

[1] `http://heikohamann.de/pub/hamannEtAlAlife2010pend.mpg`

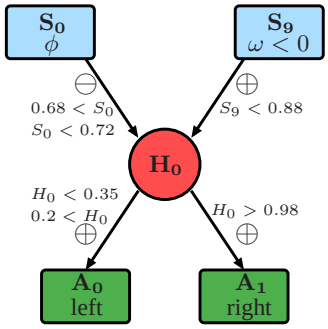


Figure 6: Inverted pendulum, analysis of one of the best evolved AHHS2 controllers; only most relevant rules of the evolved behavior are shown.

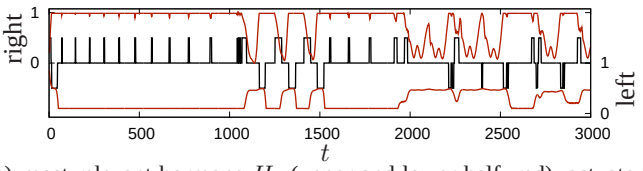
the top position. Sensor S_9 gives the intensity of negative angular velocities of the pendulum (clockwise turns) and triggers moves of the crab to the left. The proximity sensors are not used at all. The walls are avoided by the crab movements depending on position and turning direction of the pendulum. Hence, the position of the crab is virtually encoded in the motion of the pendulum.

See Fig. 7 for the sensor, hormone, and actuator dynamics. This sample run begins with an initial ($t < 50$) move of the crab from the center to the outer left due to transient dynamics of H_0 in the left compartment (see Fig. 7(a)). This motion implements the up-swinging of the pendulum and is followed by ten small displacements of the crab to the right to keep the pendulum swinging counterclockwise. At $t = 1093$ the turning direction of the pendulum changes (see Fig. 7(b)). A sequence of right-left movements is initiated to reestablish the counterclockwise turning. Later at $t = 1933$ a phase of low angular velocity is reached which causes irregular movements of the crab that hold the pendulum close to the top position.

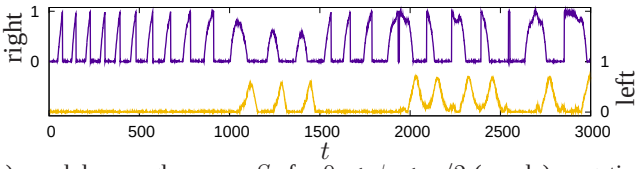
Gait learning

The evolutionary runs of the gait learning task were performed with a population of 20 randomly initialized controllers. The configuration of the AHHS was set to 5 hormones. The number of rules was varied between 20 and 300. The runs were stopped after 200 generations. Linear proportional selection was used and elitism was set to one. The mutation rate was 0.15 per gene (rule or hormone, with a maximal, absolute change of range 0.1). The recombination (two-point crossover) rate was 0.05. One run of the evolution (full 200 generations) took about 28 hours of CPU time (on a single core of a standard, up-to-date desktop PC).

In the first version of the simulation (damped joints), the evolved behaviors reach high fitness values for all investigated settings of the AHHS (see Fig. 8). Directly approaching the wall yields a fitness of about 0.7, getting one half of the modules over the wall yields a fitness of 0.8, and a fitness

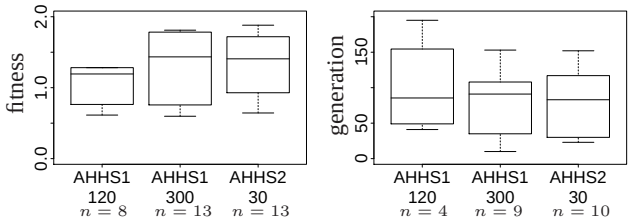


(a) most relevant hormone H_0 (upper and lower half, red), actuator left A_0 (upper half, black), right A_1 (lower half, black)



(b) pendulum angle sensor S_0 for $0 < \phi < \pi/2$ (purple), negative angular velocity sensor S_9 (lower half, yellow)

Figure 7: Inverted pendulum, most relevant hormone, sensors, and both actuator control values for both compartments (left and right) of the evolved behavior.

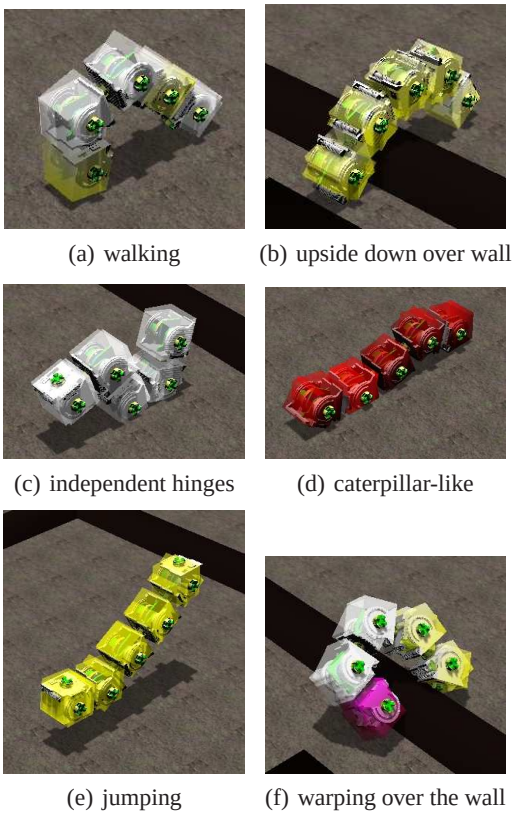


(a) fitness (b) generation

Figure 8: 5-module gait learning with damped joints, comparison of fitness and evolution speed, which is indicated by the generation in which 75% of the overall max. fitness ($1.41 = 0.75 \times 1.88$) was reached (if at all).

of above 1 is reached, if the wall is overcome. Typically the evolved behaviors rely on two or three of the five provided hormones only and make use of less than ten rules. However, a too low number of rules results in too little exploration of the behavior space. Based on preliminary tests we decided to use 30 rules for AHHS2. One AHHS2 rule is potentially active for each rule type, which corresponds to four active AHHS1 rules. However, AHHS2 cannot optimize the parameters for each rule type individually. Still, we tested the AHHS1 with 120 rules and also with a much higher number of 300 rules. The results show no statistical significant differences but show in a trend that the AHHS1 does not reach comparable results as AHHS2 with corresponding rule numbers. In addition, the behaviors evolved by AHHS1 show high variance depending on the deterministic chaos through the complex system (simulation of physics).

Using the second version of the simulation (fixed joints), we have tested smaller differences in the number of rules between AHHS1 and AHHS2. The results show that the more realistic simulation of the joints complicates the evolution of fast locomotion. However, the favoring of caterpillar-like locomotion is reduced significantly and especially in case of AHHS2 an unexpected vast diversity[2] of different locomotion paradigms is observed (see Fig. 9 for a short collection). Basically we observed three classes of locomotion: erected walking behavior, caterpillar-like locomotion, and locomotion through jumps. The behaviors evolved using AHHS1 were less diverse. Quantifying these differences will be the focus of future studies.



(a) walking (b) upside down over wall

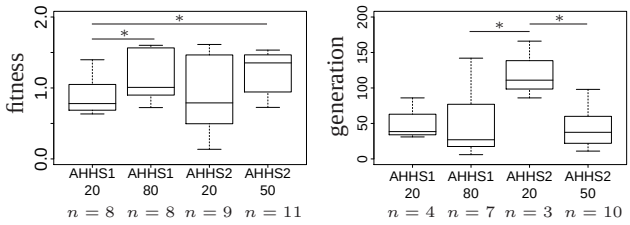
(c) independent hinges (d) caterpillar-like

(e) jumping (f) warping over the wall

Figure 9: Screenshots showing the diversity of evolved locomotion paradigms (colors represent three selected hormones in the primary colors according to the RGB color model).

The comparison of the best evolved behaviors is shown in Fig. 10(a) and the speed of evolution is shown in Fig. 10(b). 55% of the AHHS2-runs with 50 rules and 38% of the AHHS1-runs with 80 rules reach a best fitness that is within 80% of the theoretical maximum fitness of about 1.7. Significant results are only reached for AHHS1 with 20 rules compared to both AHHS1 with 80 rules and to AHHS2 with 50 rules. Noticeable is the bad performance of AHHS2 with just 20 rules both in terms of final best fitness and speed of evolution. From our observations we speculate that the initial exploration (during few of the early generations) of the search space (basically the sensory-motor configurations) is a relevant feature. Identifying the actual shortcoming of AHHS2 in this context is part of our future research.

[2] `http://heikohamann.de/pub/hamannEtAlAlife2010.mpg`



(a) fitness (Wilcoxon $p < 0.05$) (b) generation (Wilc. $p < 0.05$)

Figure 10: 5-module gait learning with fixed joints, comparison of fitness and evolution speed, which is indicated by the generation in which 75% of the overall max. fitness was reached (if at all).

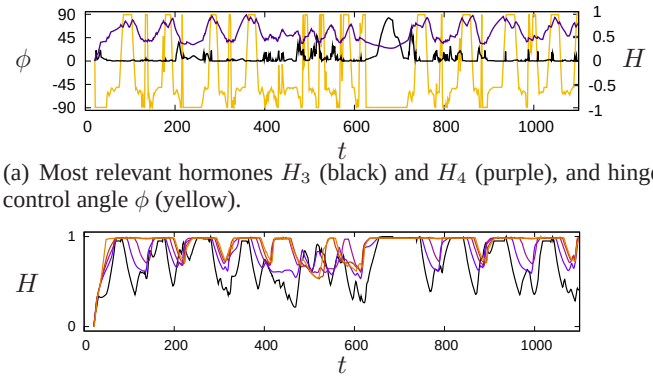
One important aspect in the differences between the two controller types seems to be the different triggering of rules in AHHS1 and AHHS2. The behaviors of AHHS1 clearly show more fast-paced movements. With damped joints this seems to be a disadvantage as smooth movements are less likely. Using the fixed joints this sometimes results in fast locomotion through little jumps.

The evolved structures are complex and the underlying processes are often counter-intuitive. The in-depth analysis of individual behaviors is alleviated by considering the number of steps a rule has been active (triggered). Typically, about one third of the rules trigger never or very seldom.

Post-evaluation and analysis

We have investigated the behavior of one of the best evolved AHHS2 controllers in the second version of the simulator. It shows a dynamic caterpillar-like motion[3]. It is noticeable that the rules show characteristics of specialization and optimization. For example, often the (floating) index of the output hormone is close to an integer (i.e., the rule's effect is mostly limited to one hormone) and often a rule weights are above 0.5 showing the specialization of those rules. For the investigated controller we have identified three most relevant hormones: H_2, H_3, and H_4. The angle of the hinge is mainly controlled by hormones H_3 and H_4 (see Fig. 11(a). High values of H_4 turn the hinge towards $+90°$ while any value of $H_3 > 0$ turns the hinge towards $-90°$. As a reinforcing effect there is a hormone rule that decreases H_4, if $H_3 > 0$. H_2 shows the influence by diffusion of hormones through the organism (see Fig. 11(b). A decreasing concentration in the back module is consequently followed by a decrease in the second last, middle, and second first module, hence, forming a hormone wave that is propagating through the organism. Finally, we investigated the influence of mutations. The leading design paradigm of AHHS2 was to improve the causality of the mutation operator (small changes in genome result in small changes in the behavior). This was done exemplarily by taking an evolved controller

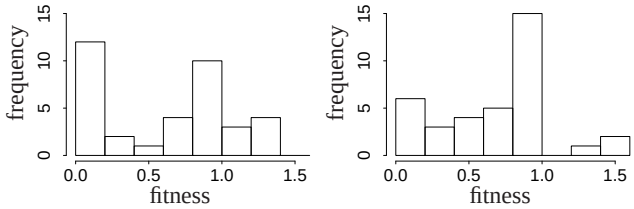
[3] http://heikohamann.de/pub/hamannEtAlAlife2010ind.mpg



(a) Most relevant hormones H_3 (black) and H_4 (purple), and hinge control angle ϕ (yellow).

(b) Hormone H_2 in all five modules, demonstrating the effect of diffusion (from front module to back: light to dark).

Figure 11: 5-module gait learning with fixed joints, analysis of the evolved behavior.



(a) AHHS1 (b) AHHS2

Figure 12: Fitness landscape neighborhood, fitness histogram of 35 samples of mutated controllers, fitness of the original controller is for AHHS1: 0.84, for AHHS2: 0.81.

from each type. For both we produced 35 controllers by applying the mutation operator once for each. The evaluated fitnesses of these 35 controllers are shown as a histogram in Fig. 12. For AHHS1 the majority of mutated controllers had a fitness of less than 0.2. For AHHS2 the majority of mutated controllers reached about the original fitness. For both types some controllers reached higher fitness due variance introduced by deterministic chaos in the simulated physics.

Conclusion and Outlook

We have reported the application of our hormone control approach to the domain of evolutionary modular robotics. The automatic synthesis of controllers, that facilitate locomotion of organisms built from five robot modules, has been effective in a majority of the evolutionary runs. Almost all evolved controllers are able to generate a form of locomotion that takes the organism at least to the wall. A majority of the evolved controllers were able to overcome the wall. An unexpected vast diversity of locomotion paradigms was evolved especially in the second version of the simulation. On the one hand, this shows the complexity of the gait learning task in modular robotics because there are many solutions of similar utility. On the other hand, it shows the diver-

sity of behaviors representable by AHHS controllers.
Whether the redesigned controller AHHS2 is generally superior to the original AHHS1 design is still an open question. However, in case of the inverted pendulum it performs significantly better. In the gait learning scenario AHHS2 shows a higher diversity and behaviors with smoother movements resulting in more reliable locomotion.
There are many open issues and this research track is rather at its beginning. Our future research will include the following. The different possibilities of initializations need to be investigated extensively. For example, the controllers could be initialized with specialized sensor, hormone, and actuator rules (i.e., weights of 1). Scalability and more complex tasks from the domain of modular robotics will be investigated (e.g., organisms with more modules). We plan to use environmental incremental evolution (e.g., steadily increasing heights of walls) as reported by Nakamura et al. (2000). The dynamic adaptation of rule numbers by evolution will be investigated. Hence, we will evolve hormone reaction networks through complexification similar to (Stanley and Miikkulainen, 2004). Finally, we plan to check the controllers' exploration of the sensory-motor space, especially, during the initial generations to get a better understanding of what facilitates a high diversity of solutions.

Acknowledgments

This work is supported by: EU-IST-FET project 'SYMBRION', no. 216342; EU-ICT project 'REPLICATOR', no. 216240.

References

Chouard, T. (2010). Revenge of the hopeful monster. *Nature*, 463:864–867.

Cliff, D., Harvey, I., and Husbands, P. (1993). Explorations in evolutionary robotics. *Adaptive Behavior*, 2.

Clune, J., Beckmann, B. E., Ofria, C., and Pennock, R. T. (2009). Evolving coordinated quadruped gaits with the hyperneat generative encoding. In *Proceedings of the 2009 IEEE Congress on Evolutionary Computation (CEC)*. IEEE.

Dale, K. and Husbands, P. (2010). The evolution of reaction-diffusion controllers for minimally cognitive agents. *Artificial Life*, 16(1):1–19.

Hamann, H., Stradner, J., Schmickl, T., and Crailsheim, K. (2010). A hormone-based controller for evolutionary multi-modular robotics: From single modules to gait learning. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'10)*. in press.

Levi, P. and Kernbach, S., editors (2010). *Symbiotic Multi-Robot Organisms: Reliability, Adaptability, Evolution*. Springer-Verlag.

Matarić, M. J. and Cliff, D. (1996). Challenges in evolving controllers for physical robots. *Robotics and Autonomous Systems*, 19(1):67–83.

Murata, S., Kakomura, K., and Kurokawa, H. (2008). Toward a scalable modular robotic system - navigation, docking, and integration of m-tran. *IEEE Robotics & Automation Magazine*, 14(4):56–63.

Nakamura, H., Ishiguro, A., and Uchilkawa, Y. (2000). Evolutionary construction of behavior arbitration mechanisms based on dynamically-rearranging neural networks. In *Proceedings of the 2000 Congress on Evolutionary Computation*, volume 1, pages 158–165. IEEE.

Neal, M. and Timmis, J. (2003). Timidity: A useful emotional mechanism for robot control? *Informatica*, 27(4):197–204.

Nelson, A. L., Barlow, G. J., and Doitsidis, L. (2009). Fitness functions in evolutionary robotics: A survey and analysis. *Robotics and Autonomous Systems*, 57:345–370.

Nolfi, S. and Floreano, D. (2004). *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press.

REPLICATOR (2010). Project website. http://www.replicators.eu.

Schmickl, T. and Crailsheim, K. (2009). Modelling a hormone-based robot controller. In *MATHMOD 2009 - 6th Vienna International Conference on Mathematical Modelling*.

Schmickl, T., Hamann, H., Stradner, J., and Crailsheim, K. (2010). Hormone-based control for multi-modular robotics. In Levi, P. and Kernbach, S., editors, *Symbiotic Multi-Robot Organisms: Reliability, Adaptability, Evolution*. Springer.

Shen, W.-M., Krivokon, M., Chiu, H., Everist, J., Rubenstein, M., and Venkatesh, J. (2006). Multimode locomotion via superbot reconfigurable robots. *Autonomous Robots*, 20(2):165–177.

Stanley, K. O., D'Ambrosio, D. B., and Gauci, J. (2009). A hypercube-based encoding for evolving large-scale neural networks. *Artificial Life*, 15(2):185–212.

Stanley, K. O. and Miikkulainen, R. (2004). Competitive coevolution through evolutionary complexification. *Journal of Artificial Intelligence Research*, 21(1):63–100.

Stradner, J., Hamann, H., Schmickl, T., and Crailsheim, K. (2009). Analysis and implementation of an artificial homeostatic hormone system: A first case study in robotic hardware. In *The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'09)*, pages 595–600. IEEE Press.

Stradner, J., Hamann, H., Schmickl, T., Thenius, R., and Crailsheim, K. (2010). Evolving a novel bio-inspired controller in reconfigurable robots. In *10th European Conference on Artificial Life (ECAL'09)*, LNCS. Springer. (in press).

SYMBRION (2010). Project website. http://www.symbrion.eu.

Vargas, P. A., Moioli, R. C., von Zuben, F. J., and Husbands, P. (2009). Homeostasis and evolution together dealing with novelties and managing disruptions. *International Journal of Intelligent Computing and Cybernetics*, 2(3).

Winkler, L. and Wörn, H. (2009). Symbricator3D - A distributed simulation environment for modular robots. In Xie, M., Xiong, Y., Xiong, C., Liu, H., and Hu, Z., editors, *ICIRA*, volume 5928 of *LNCS*, pages 1266–1277. Springer.

Adaptive Action Selection Mechanisms for Evolutionary Multimodular Robotics

Serge Kernbach[1], Thomas Schmickl[2], Heiko Hamann[2], Jürgen Stradner[2], Christopher S. F. Schwarzer[3], Florian Schlachter[1], Alan F.T. Winfield[4], Rene Matthias[5]

[1]University of Stuttgart, Germany; [2]Karl-Franzens-University Graz, Austria; [3]University of Tübingen, Germany; [4]Bristol Robotics Laboratory (BRL), UWE Bristol, UK; [5]Karlsruhe Institute of Technology, Germany

Abstract

This paper focuses on the well-known problem in behavioral robotics – "what to do next". The problem addressed here lies in the selection of one activity to be executed from multiple regulative, homeostatic and developmental processes running onboard a reconfigurable multi-robot organism. We consider adaptive hardware and software frameworks and argue the non-triviality of action selection for evolutionary robotics. The paper overviews several deliberative, evolutionary and bio-inspired approaches for such an adaptive action selection mechanism.

Introduction

Evolutionary robotics is a well-established research field, which combines several such areas as robotics, evolutionary computation, bio-inspired and developmental systems (Nolfi and Floreano, 2000). This field is characterized by multiple challenges related to platform development, onboard fitness evaluation, running time of evaluation cycles and other issues (Levi and Kernbach, 2010). Synergies between reconfigurable robotics and evolutionary computation are of special interest, because here the high developmental plasticity of the hardware platform can be exploited to realize the goal of adaptivity and reliability.

Modern reconfigurable multi-robot systems possess very high computational power and extended communication for performing evolutionary operations on-board and on-line. These hardware capabilities allow us to extend the software framework to include the whole regulative, homeostatic and evolutionary functionality for achieving long-term autonomous behavior of artificial organisms (Levi and Kernbach, 2010). In this work we focus on the issues of running multiple control processes on board the robot. These processes are created by evolutionary development, homeostasis and self-organizing control, learning, and middle- and low-level management of software and hardware. Some of these processes will have a protective role in preventing the mechatronic platform from harm during the evaluation phases. We expect that regulative and developmental processes will, in some situations, contradict each other and thus come into conflict. Multiple difficulties with action selection mechanisms are well-known in robotics (Prescott, 2008). When applied to evolutionary robotics these create problems related to, for instance, credit assignment (Whitacre et al., 2006), self-organization and fitness evaluation (Floreano and Urzelai, 2000), and robustness of behavioral and reconfiguration strategies (Andersen et al., 2009).

More generally, action selection is a fundamental problem in artificial systems targeting long-term autonomous and adaptive behavior in complex environments, especially when such a behavior is expected to be evolved (Gomez and Miikkulainen, 1997). Current thinking and experience suggests that several architectures, e.g. subsumption, reactive, insect-based or others (Brooks, 1986), need to be considered as a framework around bio-inspired and evolutionary paradigms for complex behaviorial systems.

This work is an overview paper, which introduces the problem of action selection in evolutionary modular robotics and considers a combination of behavioral, bio-inspired and evolutionary approaches for its solution. Firstly, the field of morphogenetic robotics is outlined in Sec. II, then the high complexity of the regulatory framework is underlined in Sec. III. Sec. IV reviews a number of approaches to action selection, from the literature. Secs. V and VI present several evolutionary and bio-inspired approaches, based on a combination of fixed, self-organized and evolvable controllers and hormone-based regulation. Sec. VII concludes this work.

Morphogenetic Robotics

Artificial developmental systems, in particular developmental (epigenetic) robotics (Lungarella et al., 2003), is a new and emerging field across several research areas – neuroscience; developmental psychology; biological disciplines such as embryogenetics; evolutionary biology or ecology; and engineering sciences such as mechatronics, on-chip-reconfigurable systems or cognitive robotics (Asada et al., 2009). The whole research area is devoted to ontogenetic development of an organism, i.e. from one cell to multicellular adult systems (Spencer et al., 2008).

A closely related field is evolutionary robotics (Nolfi and Floreano, 2000), which uses the methodology of evolutionary computation to evolve regulative structures of organisms over time. Evolutionary robotics tries to mimic biological processes of evolution (Elfwing et al., 2008), but also faces challenges of embodiment, the reality gap, adaptation or running on-line and on-board a smart microcontroller device (Baele et al., 2009).

In several aspects developmental and evolutionary methodologies differ from each other:

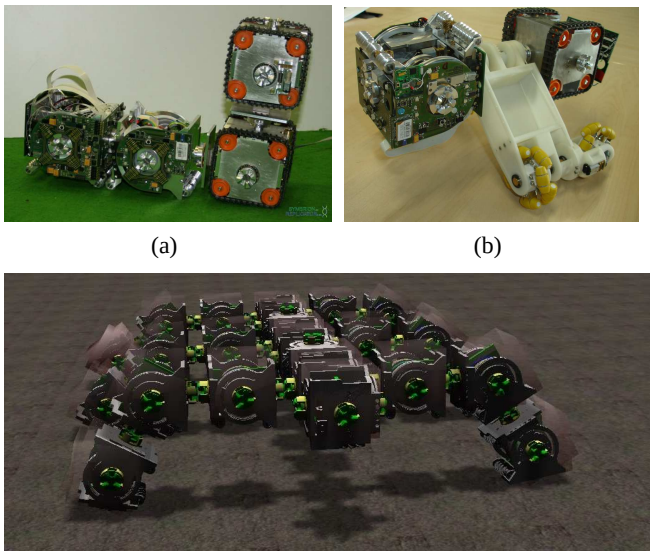
- "*... should try to endow the [developmental] system with an appropriate set of basic mechanisms for the system to develop, learn and behave in a way that appears intelligent to an external observer. As many others before us, we advocate the reliance on the principles of emergent functionality and self-organization...*" (Lungarella et al., 2003);

- "*evolutionary robotics is a new technique for the automatic creation of autonomous robots. Inspired by the Darwinian principle of selective reproduction of the fittest, it views robots as autonomous artificial organisms that develop their own skills in close interaction with the environment and without human intervention*" (Nolfi and Floreano, 2000).

Despite differences, evolutionary and developmental approaches share not only common problems, but also some ways to solve them, it seems that both are merging into one large area of self-developmental systems (Levi and Kernbach, 2010).

Both developmental and evolutionary methodologies impose a set of prerequisites on a system; one of the most important is that it should possess a high degree of *developmental plasticity*. Only then can an organism be developed or evolved. Developmental plasticity requires a specific flexible regulative, homeostatic, functional and structural organization – in this respect evolutionary/developmental systems differ from other branches of robotics. Since collective systems, due to their high flexibility and cellular-like organization, can provide such a versatile and re-configurable organization – collective robotics is a suitable subject for application of evolving and developmental approaches.

The approach used in our work is based on modularity and reconfigurability of the robot platform, as shown in Fig. 1. Individual modules possess different functionality and can dock to each other. Changing how they are connected, an aggregated multi-robot system (organism) possesses many degrees of structural and functional freedom. With a self-assembly capability, robots have control over their own structure and functionality; in this way different "self-*" features, such as self-healing, self-monitoring or self-repairing can emerge. These self-* features are related in many aspects to adaptability and evolve-ability, to emergence of behavior and to controllability of long-term developmental processes. The self-issues are investigated in manufacturing processes (Frei et al., 2008), distributed systems (Berns and Ghosh, 2009), control (Brukman and Dolev, 2008), complex information systems (Babaoglu et al., 2005) and cognitive sensor networks (Boonma and Suzuki, 2008).



(a) (b)

(c)

Figure 1: **(a), (b)** Real prototypes of aggregated robots from the SYMBRION/REPLICATOR projects; **(c)** Image of the simulated multi-robot organism.

The platform, shown in Fig. 1 is a complex mechatronic system. Each module includes the main CPU, intended for behavioral tasks that require high-computational power. This CPU is a Blackfin double-core microprocessor with DSP functionality, which can run with up to 550MHz core clock and supports a μCLinux kernel. It possesses an efficient power management system and in its current version the main CPU can utilize 64Mb SDRAM. Peripheral tasks, e.g. sensor-data processing, control of brushless motors, power management and others are executed by several ARM Cortex and low-power MSP microcontrollers. Each module has an energy source with a capacity of about 35Wh. All of them are connected through Ethernet and a power sharing bus. In the next section we briefly discuss a framework of software controllers, developed for this system and introduce the problem of action selection.

Controller Framework, Middleware Architecture and the Need for Action Selection

In robotics, there are several well-known control architectures, for example subsumption/reactive architectures (Brooks, 1986), insect-based schemes (Chiel

et al., 1992) or structural, synchronous/asynchronous schemes (Simmons, 1991). An overview of these and other architectures can be found in (Siciliano and Khatib, 2008). Recently, multiple bio-inspired and swarm-optimized control architectures have appeared e.g., (Kernbach et al., 2009b). In designing the general control architecture, we face several key challenges:

- **Multiple processes.** Artificial organisms execute many different processes, such as evolutionary development, homeostasis and self-organizing control, learning, middle- and low-level management of software and hardware structures. Several of these processes require simultaneous access to hardware or should be executed under real-time conditions.

- **Distributed execution.** As mentioned, the hardware provides several low-power and high-power microcontrollers and microprocessors in one robot module. Moreover, all modules communicate via a high-speed bus. Thus, the multiprocessor distributed system of an artificial organism provides essential computational resources, however their synchronization and management are a challenge.

- **Multiple fitness.** Although fitness evaluation using local sensors is mentioned in the literature, here we need to stress the problem of credit assignment related to the identification of a responsible controller, see e.g. (Whitacre et al., 2006)). Since many different controllers are simultaneously running on-board, the problem of credit assignment as well as *interference between controllers* is critical.

- **Hardware protection.** Since several controllers use the trial-and-error principle, the hardware of the robot platform should be protected from possible damage caused during the controllers' evolution.

Corresponding to the hardware architecture, the general controller framework is shown in Fig. 2. This structure follows the design principles, originating from *hybrid deliberative/reactive systems*, see e.g. (Arkin and Mackenzie, 1994). It includes rule-based control schemes, e.g. (Li et al., 2006), as well as multiple adaptive components. The advantage of the hybrid architecture is that it combines evolvability and the high adaptive potential of reactive controllers with deliberative controllers. The latter provide planning and reasoning approaches that are required for the complex activities of an artificial organism.

Meeting the challenges above raises the issue of choosing a suitable underlying middleware with an adequate architecture. As mentioned, a dual-core DSP with a μCLinux will be used as the main CPU. This approach provides much flexibility and facilitates rapid development, for instance in the use of shared standard libraries (e.g., STL, Boost and others). Although the DSP is relatively powerful computationally (given its power consumption), it nevertheless imposes some restrictions that need to be addressed.

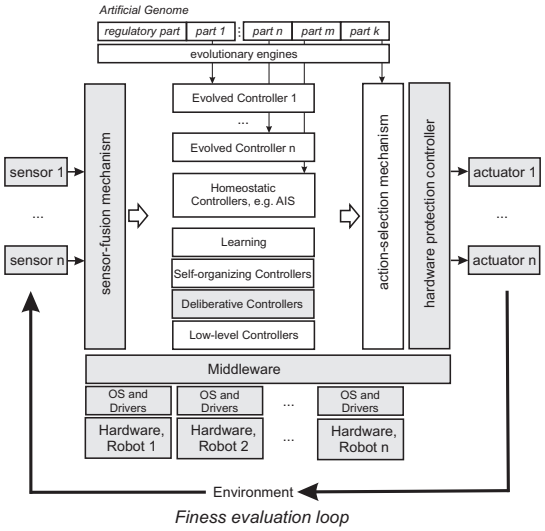


Figure 2: General controller framework. All controllers/processes are distributed in the computational system of an artificial organism, OS – operating system. Structure of controllers utilizes hybrid deliberative/reactive principle.

The most important limitation may be the fact that there is no hardware memory management unit (MMU). Due to the way the μCLinux software MMU works, we decided to design the controller framework as a set of competing applications; an approach that is quite common for UNIX environments (Tanenbaum and van Steen, 2008). For communication within the controller framework a message based middleware system has been implemented. This provides the necessary flexibility needed to implement an event-driven system without having to determine all of the timing constraints in advance (Tanenbaum and van Steen, 2008). Sockets serve as the only mechanism for inter-process communication. Although this may appear to be a disadvantage it yields some very important benefits. First, there is only one standard communications interface defined in advance, with attendant benefits in parallel development across multiple teams. Second, and with regard the robustness of the system; if, for example, a certain controller crashes, the impact of that crash is limited to a single process within the system. All the other applications remain functional and the system may even restart the crashed process later on. The same applies to the middleware itself, as it conforms to the same rules. The approach assures that connections once established are not harmed even if, for example, the addressing module itself is faulty and, therefore cancelled by the operating system (the only limitation here will be the creation of new connections as this is impossible without addressing modules). For connections to other robot modules via

Ethernet the same socket mechanism is used, as for standard Ethernet communications. With this framework we are able to create several controllers which use, for example, evolutionary engines with a structure encoded in an artificial genome. It is assumed that there are also a few task-specific controllers placed hierarchically above other controllers. These task-specific controllers are in charge of the macroscopic control of an artificial organism. They may, for instance, use deliberative architectures with different planning approaches, e.g. see (Weiss, 1999).

Finally, a hardware protection controller closes the fitness evaluation loop for the evolvable part of the controllers (Kernbach et al., 2009a). This controller has a reactive character and monitors activities between the action selection mechanism and actuators as well as exceptional events from the middleware. It prevents actions that might immediately lead to damage to the platform (e.g., by mechanical collisions).

The action selection mechanism is one of the most complex elements of the general controller framework. This mechanism reflects a common problem of intelligent systems, i.e. "what to do next", (Bratman, 1987). This problem is especially challenging in evolutionary robotics for several reasons. Firstly, the fitness evaluation loop will include a combination of different controllers, so it may be difficult to find a unique correlation between a specific evolved controller and its own fitness value. Secondly, several controllers on different levels will be simultaneously evolved, so that some co-evolutionary effects may appear. Among other problems, we should also mention the multiple co-dependencies between fixed, self-organized and evolving controllers.

Action Selection Mechanism

Formally, action selection is defined as follows: "given an agent with a repertoire of available actions ... the task is to decide what action (or action sequence) to perform in order for that agent to best achieve its goals" (Prescott, 2008). Within the context of the projects general controller framework shown in Fig. 2, the role of the action selection mechanism is to determine which controller(s) are driving the actuators at any given time. At one level the action selection mechanism can be thought of as a switch, selecting which of the controllers is connected to the actuators; however a simple switch would fail to provide for, firstly, smooth motor transitions from one controller to another and, secondly, the fact that in this hybrid deliberative/reactive architecture some controllers will need to be prioritized for short time periods (e.g., for obstacle avoidance) whereas others need periods of control over longer time spans (perhaps subsuming low-level reactive elements) to achieve high level goals. In practice, therefore, the action selection mechanism will need to combine some or all of the following elements:

- prioritization of low-level reactive controllers so that they are given control with very low latency;
- vector summation or smoothing between some controller outputs in order to achieve jerk free motor transitions on controller switching, and
- a time multiplexing scheme to ensure that different controllers are granted control with a frequency and for time periods appropriate to achieving their goals.

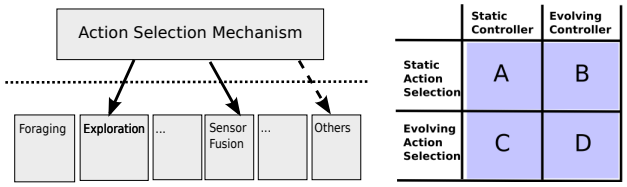
Action selection mechanisms have been the subject of research in both artificial and natural systems for some years, see for instance (Maes, 1990; Hexmoor et al., 1997; Prescott et al., 2007). However, in a recent review Bryson suggests that no widely accepted general-purpose architecture for action selection yet exists (Bryson, 2007). Relevant to the present work is a review of compromise strategies for action selection (Crabbe, 2007). A compromise strategy is one in which instead of selecting a single controller, the action selection mechanism combines several controller outputs in such a way as to achieve a compromise between their (otherwise conflicting) goals; (Crabbe, 2007) suggests that a compromise strategy is more beneficial for high-level than low-level goals.

It is important to note that the action selection mechanism embeds and encodes design rules which will critically influence the overall behavior of the robot. In order to arbitrate between, possibly conflicting, controller goals the action selection mechanism will certainly need to access internal state data for the robot (i.e. from the homeostatic controllers), and may need to access external sensor data. Furthermore, given that those action selection design rules and their parameters may be difficult to determine at design time, we are likely to require an evolutionary approach; hence the connection between the genome structure/evolutionary engine and the action selection mechanism shown in Fig. 2. We may, for instance, evolve the weights which determine the relative priority of controllers as in (González et al., 2006), or co-evolve both controllers and action selection parameters (González, 2007).

Evolution and Action Selection

The action selection mechanism can be seen as a two-tiered architecture of the robot controller (Fig. 3(a)). On the lower tier are activities like elementary actions (e.g. turn right), behavior routines (e.g. random walk) or sub-controllers (e.g. sensor fusion). The upper tier is the action selection mechanism, that controls which activities are running at the moment.

The adaptiveness of the entire robot control can be increased by applying evolutionary approaches to the different tiers of the architecture (Fig. 3(b)): **(A)** Neither the controller nor the action selection module adapts. **(B)** The action selection is static and the activities evolve. **(C)** The action selection mechanism evolves and the activities are



(a) (b)

Figure 3: **(a)** Two-tier architecture with action selection and activities. **(b)** Evolution at the different tiers of the architecture.

static. **(D)** Both action selection mechanism and activities evolve.

One concept for approach **(B)** is a static planning system where a plan to achieve a goal is formulated as a series of activities described by fitness functions. At each step of the plan, the actual controller for the corresponding activity is evolved by online evolutionary algorithms using the fitness function. In this way, the overarching plan does not adapt but the execution of the individual steps evolves. Examples for activities of such a plan can be "Sense Energy Source" or "Robot Aggregation".

An extreme example for approach **(C)** is a large monolithic evolving neural network as the action selection mechanism. The activities are direct sensor and actuator actions, like reading sensor values and setting motor velocities. An increase in complexity of the activities allows a reduction in the action selection mechanism. For example, instead of direct commands, activities can be small controllers such as collision avoidance or sensor fusion. With very complex activities that control complete behaviors, like foraging, resting or exploring for example, the action selection mechanism can degenerate into a simple priority management system that checks for which "needs" are the most urgent. While a complex neural network can be difficult to evolve efficiently, a priority system can be evolved easily by parameter evolution of the weights or thresholds of different needs and motivations.

Approach **(D)** offers the most flexibility and adaptiveness of the controller architecture. This could possibly be a simple combination of **(B)** and **(C)**. It is conceivable that the action selection adapts to a changing environment by changing priorities of preferences for subordinated activities. In case no matching controller is available for an operation, the action selection can define new fitness functions and evolve new activities to suit the current needs, or evolve existing activities for extended tasks.

In the next section a hormone based controller for approach **(D)** is presented.

Biologically Inspired Mechanism

Artificial Hormone Control

Within the scope of the SYMBRION/REPLICATOR projects, we follow a bio-inspired approach of decentralized coordination of action selection which is distributed across the robot modules: On the one hand, all robot modules, that form the organism, act as autonomous units which have a repertoire of behavioral programs available (actions/controllers). A localized action selection mechanism is needed, which decides within each single unit which action has to be selected. On the other hand, the whole organism has to decide "as a whole", which action it will perform based on its current status, on its past experience, on its current goals, and on the current set of sensor information. To achieve this difficult task, we developed the Artificial Homeostatic Hormone System (AHHS) which mimics the spread of cellular signals (chemicals, hormones) within multicellular (metazoan) organisms (Schmickl and Crailsheim, 2009; Stradner et al., 2009). This set of controllers, often called "hormone controllers" allow cells (robot modules) to specialize within the robot organisms and to reflect specific physiological states by a simple physiological model that mimics excretion, dilution, diffusion, (chemical) interaction, and degradation of hormones. Within the robotic organism, gradients of hormones emerge over time, reflecting not only the modules' positions in the organism but also important status information, such as the current energy level. In a hierarchical approach, the globally influenced hormone status within a robot module can help to select an optimal local controller. In turn, the execution of local controllers can significantly alter the hormone system, thus, via diffusion to neighboring modules, alter the behavior of controllers in nearby modules. This way, the AHHS controller allows not only decentralized action selection, but also inter-modular communication between different sub-controllers, hardware abstraction, and sensor integration. See Fig. 4 for a graphical representation of the AHHS design as described above. The concept of AHHS is related to gene regulatory networks (Bongard, 2002). However, here each edge has its own activation threshold and redundant edges with different activations between two hormones are allowed.

Action selection is not only about choosing the right action but also about how selected actions integrate to low-level motor commands in a robot (see Öztürk, 2009). The AHHS allows multiple hormones to affect the actuators of the robots in parallel by integrating various chemical stimulations (see Fig.5 for a schematic of this process).

In the following we present results of a simplified scenario to demonstrate the principles of action selection in a hormone controller. We restrict ourselves to a single robot module and we use the AHHS directly to control the robot without having sub-controllers as described in the general concept above. However, the robot's body is virtually separated into two compartments (c.f. Fig. 5a) between which

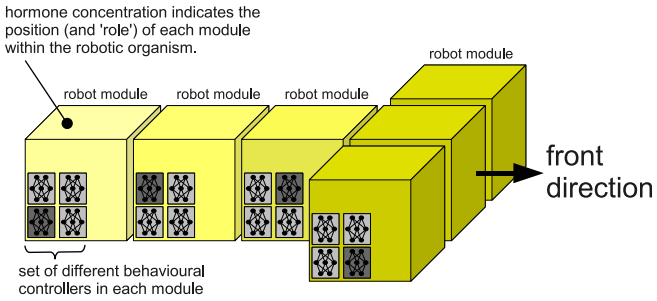


Figure 4: Schematic representation of decentralized action selection that is provided in various 'body parts' of the organism by the AHHS robot control.

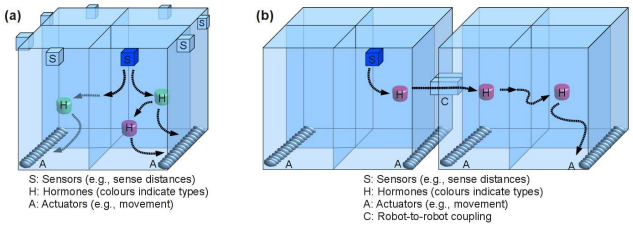
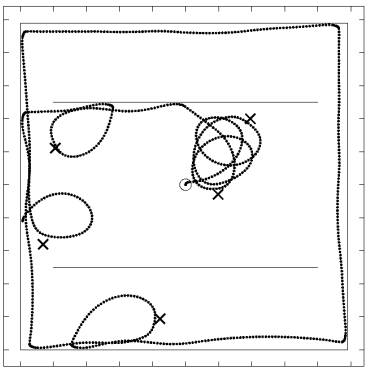
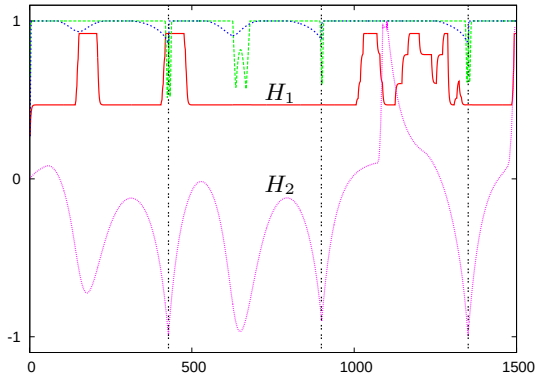


Figure 5: In the AHHS, actuators are influenced by various hormone states in parallel, this way allowing signal integration to produce "mixed" or blended actions.



(a) circle is initial pos., crosses show sequence of emitters



(b) The three vertical lines indicate the time at which emitters were reached; note local minima of H_2 at $t = 175$ and $t = 647$ showing the misses in approaching the emitter.

Figure 6: Robot's trajectory using AHHS controller and the dynamics of five hormones responsible for action selection.

hormones diffuse. Each compartment is associated with one half of the robot. The left compartment contains the left wheel and all proximity sensors of the left half (similar for the right half).

The task of the hormone controller is to control a robot module in a 2-D arena, to catch light emitters, and to explore the arena. Thus, basically two actions are needed to succeed in this task: exploration/wall avoidance and a gradient ascent behavior. The arena consists of surrounding and additional walls in the upper and lower third (see Fig. 6(a)). In addition, it includes one randomly positioned emitter. Both, the walls and the emitter, are perceived by the robot, if they are within range of the sensors (range of light sensors about 50% and range of proximity sensors about 10% of the arena width). The intensity of the sensor signal depends on the distance to the walls and the emitter, respectively. If the robot reaches the emitter (distance $<$ robot diameter) the emitter is erased and reappears at a random position.

The fitness function, that is applied in the artificial evolution, rewards the successful locating of the light emitter, but also – at smaller scale – the exploration of the arena. Thus, the robot has to switch between the action of exploration, if no emitter is detected (i.e., it is too far away to have any significant impact on the sensors), and the gradient ascent, if the emitter is detected. The trajectory of the best individual of the 1000th generation is plotted in Fig. 6(a).

The evolved hormone reaction network of the best evolved controller is complex. We restrict ourselves to a description of the most prominent features. In the hormone network we identify two major hormone interactions that represent the actions: exploration/wall avoidance and gradient ascent. Without any significant input the robot drives in wide right turns forming spirals. If it approaches a wall it avoids collisions because of two controller rules. First, the production of hormone H_1 (see Fig 6(b)) is triggered by the proximity sensor that points 45 degrees to the right (the closer the wall the higher the hormone production). Second, another rule controls the right wheel depending on hormone H_1. With increasing value of this hormone the wheel is accelerated resulting in a turn to the left. Hence, a wall following behavior emerges during which the robot keeps the wall to the right. A question concerning action selection is when to stop the wall following action and continuing the gradient ascent in order to reach the light emitter. This is controlled by hormone H_2. Its value is reduced with increasing input of the left light sensor (bright light results in low H_2). A second rule controls the left wheel which is decelerated mainly for values of $H_2 \in [-0.2, -0.6]$. This slowdown of the left wheel results in a left turn. Hence, the robot

interrupts its wall following behavior and turns towards the light (which is always to the left as the robot follows the wall counterclockwise). Hence, we have identified the relevant trigger (hormone H_2) for the action selection mechanism in this hormone network. Obviously, this is a simplified application of AHHS and in future applications we will aim for much more complex tasks of multi-modular robotics.

Adapting Hormone Control

The hormone controllers mentioned in the previous section are subject to evolutionary adaptation. A data structure called "genome" contains rule descriptions and other parameters, which describe some physicochemical properties of the simulated hormones (production rates, decay rates, diffusion rates). In addition, these data describe how one hormone can influence the dynamics of the concentrations of other hormones. The genome is modified by a process of artificial evolution, which allows the embedded action selection to adapt over time to a given body shape or to changes in the environmental conditions. In our evolutionary approach, the fitness of the system reflects multiple levels of adaptation: The whole organism level (e.g., efficiency of shapes and gait patterns) but also on the individual module level (e.g., energetic efficiency of singular modules within the organism).

Conclusion

In this paper we have briefly presented hardware and software frameworks for a reconfigurable multi-robot system. The mechatronic platform provides a high hardware plasticity in terms of structural reconfiguration, changeable locomotion and actuation, and sharing and distribution of power and information. Because of the complexity of regulative, homeostatic and evolutionary mechanisms there are multiple processes that require simultaneous access to actuators. Based on preliminary experiments these processes are expected to display contradictory characteristics. For example, the homeostatic system can require minimization of energy consumption, whereas the evolutionary system may require more energy for performing evaluation runs.

The problem of action selection considered in this paper is highly non-trivial in this context. It is not only related to the classical problem of action selection, well-known in robotics, but also has new aspects related to fitness estimation, credit assignment, evolving of multiple controllers and other issues. The problem of action selection requires a complex deliberative framework and specific controller architectures.

In this paper we have considered a hybrid controller framework, which has reactive and deliberative components. The evolutionary part, which consists of genome, evolutionary engines and evolvable controllers, represents in fact only a small part of the whole framework. It seems that evolving all regulatory structures of real robots from scratch is not feasible because of technology limitations, very specific sensor-actor systems and complexity. Furthermore, it is not fully clear whether this is a general property or is related only to technological artefacts.

Beside the hybrid framework, this paper has proposed evolutionary and bio-inspired solutions to the problem of action selection. The evolutionary approach combines fixed, self-organized and evolvable controllers; moreover the action selection mechanism can also be integrated into the evolutionary loop. The bio-inspired approach is guided by the hormone systems and based on the distribution of hormonal intensity (and between different hormones) in different compartments of a robot, and across robots in a multi-robot organism.

Acknowledgement

The "SYMBRION" project is funded by the European Commission within the work programme "Future and Emergent Technologies Proactive" under the grant agreement no. 216342. The "REPLICATOR" project is funded within the work programme "Cognitive Systems, Interaction, Robotics" under the grant agreement no. 216240. The authors also wish to acknowledge the contribution of all members of both projects.

References

Andersen, T., Newman, R., and Otter, T. (2009). Shape homeostasis in virtual embryos. *Artificial life*, 15(2):161–183.

Arkin, R. and Mackenzie, D. (1994). Planning to behave: A hybrid deliberative/reactive robot control architecture for mobile manipulation. In *Proc. of the Fifth International Symposium on Robotics and Manufacturing*, volume 5, pages 5–12, Maui, Hawaii.

Asada, M., Hosoda, K., Kuniyoshi, Y., Ishiguro, H., Inui, T., Yoshikawa, Y., Ogino, M., and Yoshida, C. (2009). Cognitive developmental robotics: A survey. *IEEE Transactions on Autonomous Mental Development*, 1(1):12–34.

Babaoglu, O., Jelasity, M., Montresor, A., Fetzer, C., Leonardi, S., van Moorsel, A., and van Steen, M. (2005). *Self-star Properties in Complex Information Systems: Conceptual and Practical Foundations (Lecture Notes in Computer Science)*. Springer-Verlag New York, Inc.

Baele, G., Bredeche, N., Haasdijk, E., Maere, S., Michiels, N., Van de Peer, Y., Schmickl, T., Schwarzer, C., and Thenius, R. (2009). Open-ended on-board evolutionary robotics for robot swarms. In Tyrrell, A., editor, *Proc. of the IEEE Congress on Evolutionary Computation (IEEE CEC-09)*, Trondheim, Norway. IEEE Press.

Berns, A. and Ghosh, S. (2009). Dissecting self-* properties. *Self-Adaptive and Self-Organizing Systems, International Conference on*, 0:10–19.

Bongard, J. (2002). Evolving modular genetic regulatory networks. In *Proceedings of the 2002 Congress on Evolutionary Computation (CEC'02)*, pages 17–21.

Boonma, P. and Suzuki, J. (2008). Exploring self-star properties in cognitive sensor networking. In *Proc. of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS-2008)*, pages 36–43.

Bratman, M. E. (1987). *Intention, Plans, and Practical Reason*. Harvard University Press, Cambridge, MA.

Brooks, R. A. (1986). A robust layered control system for a mobile robot. *Journal of Robotics and Automation*, 2(1):14–23.

Brukman, O. and Dolev, S. (2008). Self-* programming: Runtime parallel control search for reflection box. *Self-Adaptive and Self-Organizing Systems, International Conference on*, 0:481–482.

Bryson, J. (2007). Special issue (edited by J.J. Bryson): Mechanisms of action selection. *Adaptive behavior*, 15(1).

Chiel, H., Beer, R., Quinn, R., and Espenschied, K. (1992). Robustness of a distributed neural network controller for locomotion in a hexapod robot. *IEEE Transactions on Robotics and Automation*, 8(3):293–303.

Crabbe, F. L. (2007). Compromise strategies for action selection. *Philosophical Transactions of the Royal Society. B.*, 362:1559–1571.

Elfwing, S., Uchibe, E., Doya, K., and Christensen, H. (2008). Biologically inspired embodied evolution of survival. In Michalewicz, Z. and Reynolds, R. G., editors, *Proc. of the 2008 IEEE Congress on Evolutionary Computation IEEE Congress on Evolutionary Computation*, volume 3, pages 2210–2216, Hong Kong. IEEE Press.

Floreano, D. and Urzelai, J. (2000). Evolutionary robots with on-line self-organization and behavioral fitness. *Neural Networks*, 13(4-5):431–443.

Frei, R., Serugendo, G. D. M., and Barata, J. (2008). Designing self-organization for evolvable assembly systems. In *Proc. of the Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO-2008)*, pages 97–106.

Gomez, F. and Miikkulainen, R. (1997). Incremental evolution of complex general behavior. *Adaptive Behavior*, 5:317–342.

González, F. (2007). The coevolution of robot behavior and central action selection. In *Nature Inspired Problem-Solving Methods in Knowledge Engineering*, volume 4528 of *Lecture Notes in Computer Science*, pages 439–448. Springer-Verlag.

González, F., Reyes, J., and Ríos-Figueroa, H. (2006). Integration of evolution with a robot action selection model. In Gelbukh, A. and García, C. R., editors, *Proc. of the 5th Mexican International Conference on Artificial Intelligence (MICAI-06)*, volume 4293 of *Lecture Notes in Computer Science*, pages 1160–1170, Apizaco, Mexico. Springer-Verlag.

Hexmoor, H., Horswill, I., and (Eds.), D. K. (1997). Special issue: Software architectures for hardware agents. *Journal of Experimental and Theoretical Artificial Intelligence*, 9(2/3).

Kernbach, S., Meister, E., Scholz, O., Humza, R., Liedke, J., Ricotti, L., Jemai, J., Havlik, J., and Liu, W. (2009a). Evolutionary robotics: The next-generation-platform for on-line and on-board artificial evolution. In Tyrrell, A., editor, *Proc. of the IEEE Congress on Evolutionary Computation (IEEE CEC-2009)*, Trondheim, Norway. IEEE Press.

Kernbach, S., Thenius, R., Kernbach, O., and Schmickl, T. (2009b). Re-embodiment of honeybee aggregation behavior in artificial micro-robotic system. *Adaptive Behavior*, 17(3):237–259.

Levi, P. and Kernbach, S., editors (2010). *Symbiotic Multi-Robot Organisms: Reliability, Adaptability, Evolution*. Springer-Verlag.

Li, G., Lin, K.-C., and Xia, Z. (2006). Rule-based control of collaborative robots. In *Proc. of the 16th International Conference on Artificial Reality and Telexistence*, pages 68–72, Los Alamitos, CA, USA. IEEE Computer Society.

Lungarella, M., Metta, G., Pfeifer, R., and Sandini, G. (2003). Developmental robotics: a survey. *Connect. Sci.*, 15(4):151–190.

Maes, P. (1990). How to do the right thing. *Connect. Sci. J. Spec. Issue Hybrid Systems*, 1:291–323.

Nolfi, S. and Floreano, D. (2000). *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. The MIT Press, Cambridge, MA. / London.

Öztürk, P. (2009). Levels and types of action selection: the action selection soup. *Adaptive Behavior*, 17(6):537–554.

Prescott, T. (2008). Action selection. *Scholarpedia*, 3(2):2705.

Prescott, T., Bryson, J., and Seth(Eds), A. (2007). Theme issue on models of natural action selection. *Philosophical Transactions of the Royal Society. B.*, 362(1485).

Schmickl, T. and Crailsheim, K. (2009). Modelling a hormone-based robot controller. In *MATHMOD 2009 - 6th Vienna International Conference on Mathematical Modelling*.

Siciliano, B. and Khatib, O., editors (2008). *Springer Handbook of Robotics*. Springer-Verlag.

Simmons, R. (1991). Coordinating planning, perception, and action for mobile robots. *SIGART Bull.*, 2(4):156–159.

Spencer, J., Thomas, M., and McClelland, J. (2008). *Toward a Unified Theory of Development: Connectionism and Dynamic Systems Theory Re-Considered*. Oxford University Press.

Stradner, J., Hamann, H., Schmickl, T., and Crailsheim, K. (2009). Analysis and implementation of an artificial homeostatic hormone system: A first case study in robotic hardware. In *Proc. of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'09)*. IEEE Press.

Tanenbaum, A. and van Steen, M. (2008). *Verteilte Systeme - Prinzipien und Paradigmen*. Pearson Studium, München, Germany, 2nd edition.

Weiss, G. (1999). *Multiagent systems. A modern approach to distributed artificial intelligence*. MIT Press.

Whitacre, J., Pham, T., and Sarker, R. (2006). Credit assignment in adaptive evolutionary algorithms. In *Proc. of the 8th annual conference on Genetic and evolutionary computation (GECCO-06)*, New York, NY, USA. ACM.

Online Robot Task Switching Under Diminishing Returns

Jens Wawerla, and Richard T. Vaughan

Simon Fraser University, Burnaby, BC, Canada
{jwawerla, vaughan}@sfu.ca

Abstract

We investigate the task switching problem of a robot maximizing its long-term average rate of return on work performed. We propose an online method to maximize the average gain rate based on only past experience. For that we alter the formulation from optimal foraging theory and recursively include estimates of global task qualities. We demonstrate and analyze our method on a puck-foraging example. In simulation experiments under a variety of conditions we show that our method performs well compared to results obtained by brute force method using post-processed foraging data.

Introduction

Many robot applications require a robot to make task switching decisions in order to maximize its reward. Often this reward is a diminishing function of the time spent performing the task. These diminishing returns can either be caused by (i) exhausting a given task, for example having delivered all mail in a given building or by (ii) increasing difficulty to perform the task, e.g. it will be more and more difficult for a vacuum cleaning robot[1] to remove dirt as it cleans the floor. In fact it will be virtually impossible for a vacuum cleaning robot to remove all dirt particles and thus this task has no well defined intrinsic end point.

In both situations the robot has to decide when it is profitable to terminate the current task, pay a switching cost, and start a new task that yields higher rewards. The switching cost can come in form of an opportunity cost or an actual cost such as energy expenditure, transit toll or task acquisition cost. In other words the robot has to decide when to switch tasks in order maximize its long-term average reward rate. This decision depends on a number of factors: how good is the current task, how high is the switching cost and what is the average payoff function for tasks in the robot's environment?

In an earlier paper (Wawerla and Vaughan, 2009) we proposed a task switching policy based on the Marginal-Value Theorem (MVT) (see Sec. Marginal-Value Theorem). This policy required the robot to perform exploration steps in order to evaluate the average quality of the available tasks. We showed that the performance of the proposed policy was about 80% of that obtained by a near optimal policy discovered by brute force search.

In this paper we propose a recursive task switching policy based on locally available information only, hence no explicit exploration phase and thus no exploration/exploitation trade-off is required.

The policy is applicable to other task switching situations that exhibit diminishing returns. We choose foraging as an example task, since it is a canonical task in autonomous robotics (Cao et al., 1997). Robot foraging often means multi-agent *central place foraging* (Stephens et al., 2007), where foraged items are delivered to single privileged location. In contrast in this paper and our previous work (Wawerla and Vaughan, 2009) we use solitary, instant-consumption foraging in a patchy environment: a single robot immediately consumes items once they are encountered obtaining a reward without the need to deliver them to a centralized location. Items to be foraged are not distributed uniformly, but in patches defined for Behavioural Ecology as "*an homogeneous resource containing area separated from others by areas containing little or not resources*" (Danchin et al., 2008).

Marginal-Value Theorem

In behavioural ecology the task switching problem is often discussed in terms of optimal foraging theory (Stephens and Krebs, 1986) as a patch leaving decision. In this context patches are subject to diminishing returns and thus require the forager to make decisions about changing patches. In this case the task switching cost the inter-patch travel cost. An important result of optimal foraging theory is the Marginal-Value Theorem (MVT). Charnov and Orians (1973); Charnov (1976) proposed the MVT to model foraging decisions made by animals. His key result is the following patch leaving rule: "*when the intake rate in any patch drops to the average rate for the habitat, the animal should move on to another patch*" (Charnov and Orians, 1973). As

[1] We assume the robot gets rewarded for the amount of dirt collected and not for time spent vacuuming.

a consequence an optimal forager should exploit patches for a longer time as the inter-patch travel time increases and for a shorter time as the entire environment becomes more profitable. The simplicity of this rule makes it very appealing as a task-switching rule for robots, but the theorem and its validity has been widely and controversially discussed, for example by Green (1984); McNamara (1982); Stephens and Krebs (1986). Some of these issues make an implementation of the MVT as a robot task switching policy impossible. The main problems are:

- How to measure the marginal gain rate (the derivative of the gain rate) if the reward comes in discrete lumps. Andrews et al. (2007) suggest calculating the slope of the gain function between the last gain function change and the one two changes prior. In our tests (not shown) this method proved ineffective due to the stochastic nature of puck encounter during random foraging in patches with randomly placed pucks. In previous work (Wawerla and Vaughan, 2009) we used the expected value of a beta distribution over time-steps in which the robot found a puck and those in which it did not, as a proxy for the instantaneous rate. While we were able to build a task switching policy around this estimated gain rate, it is not the instantaneous gain rate. Thus leaving a patch once this estimated gain rate equals the long-term average rate does not maximize the long-term gain rate.

- The true long-term average gain rate for a given environment is usually unknown to the forager: all it can know is the average gain rate it experiences. This experience is a result of the foragers behaviour, yet the MVT requires the forager to base it's patch leaving decision on the obtainable long-term average gain rate. This circular dependency necessitates that the forager explores the action space in order to find the maximum long-term average gain rate. Previously (Wawerla and Vaughan, 2009) we used this circular dependency and turned the foraging task into a multi-armed bandit problem and applied standard ε-greedy methods (Sutton and Barto, 1998) to tackle the exploration-exploitation trade-off.

Stephens and Krebs (1986) summarize these problems as "*The MVT survives not as a rule for foragers to implement, but as a technique that finds the rate-maximizing rule from a known set of rules*". Since the MVT does not provide an implemetable policy, behavioural ecologists proposed other patch-leaving rules. (1) **number rule**, "leave after catching *n* items" (Gibb, 1958); (2) **fixed residence time rule** "leave after being in a patch for *t* time" (Krebs, 1973); (3) **give up time rule** "leave after *t* time has elapsed since the last encounter" (Krebs et al., 1974); (4) **rate rule** "leave when the instantaneous intake rate drops to a critical value *r*" (McNamara, 1982). Rules 1-3 have the advantage that the decision is based on values that are easily measurable by the forager.

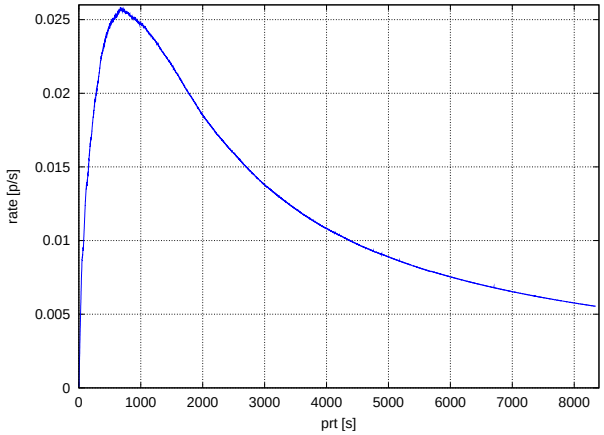


Figure 1: Average gain rate for a fixed patch residence time. Series of 100 patches with initially 50 pucks and a patch switching time of 500 seconds.

The rate rule is an extension of the MVT in that it copes with variance in patch sub-types, but it does not address the two issues mentioned above. None of these rules address the question of how to obtain the magic number on which the decision is based.

To illustrate the difficulty of this task-switching problem we conducted a brief simulation experiment. For this experiment we generated 100 constant size patches, each with initially 50 pucks. Next we had the robot forage in each patch until it was completely exhausted. For each time step we recorded the number of pucks gained from the current patch. From the recorded data we then calculated the average long-term gain rate as a function of patch residence time. In other words we forced the robot to leave each patch in a 100 patch series after a fixed time. By sweeping over patch residence times from 10 to 8000 seconds we obtained Fig. 1. This graph shows the long-term gain rate for a given patch residence time for this particular patch configuration and switching cost. The curve is interesting because it shows how large an error (i.e. reduction on average reward gain rate) a task-switching robot can make if switching too early or too late. It is worth pointing out that a robot is not actually able to measure this curve and exploit a patch optimally at the same time. Fortunately the robot only needs to find the maximum of the long-term gain rate and not determine the function per se.

Having described the optimization problem, in the following we present a new online adaptive solution that is grounded in the robot's perception and achieves foraging results comparable to an idealized forager that bases its decisions on global, unknowable environmental averages.

Marginal Gain Rate Task Switching

To derive the MVT Charnov (1976) argued that an optimal forager should maximize

$$R = \frac{\sum \lambda_j \cdot g_j(t_j) - \tau \cdot E}{\tau + \sum \lambda_j \cdot t_j} \quad (1)$$

where λ_j is the proportion of visited patches that are of type j, $g_j(t_j)$ is the net gain function for a patch of type j, τ is the average inter-patch travel time, E the rate of energy expended while switching patches and t_j is the time spent in a patch of type j. The objective of a forager is to select all patch residence times t_j such that R is maximized.

Without loss of generality we ignore the energetic cost of travel $\tau \cdot E$, since it is independent of the decision variables, so Eq. 1 reduces to

$$R = \frac{\sum \lambda_j \cdot g_j(t_j)}{\tau + \sum \lambda_i \cdot T_j} \quad (2)$$

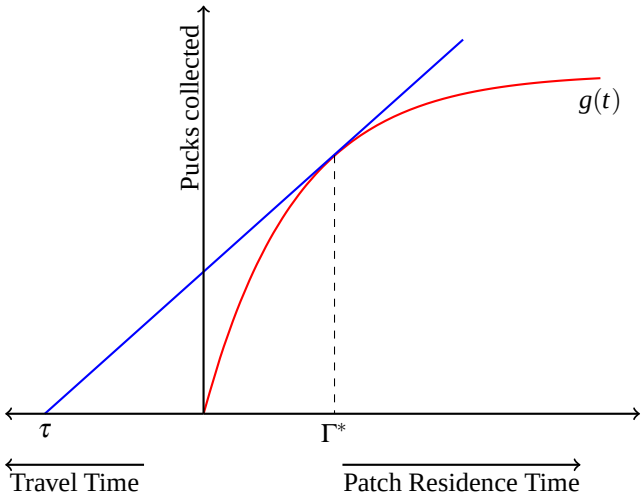


Figure 2: Typical MVT plot with two quantities on the abscissa: travel time increasing to the left, and patch residence time increasing to the right. The optimal patch residence time Γ^* is found by constructing a tangent to the gain function $g(t)$ that begins at the patch switching time τ on the travel time axis.

Charnov showed that R is maximized if $\frac{\partial g_j(t_j)}{\partial t_j} = R$. Graphically this is easy to do. As Fig. 2 shows, the optimal patch residence time T_j is found by constructing a tangent to the gain function that begins at the patch switching time τ on the travel time axis (see Stephens and Krebs (1986) for details).

The gain function $g(t)$ depends on (i) the actual patch quality, which varies from patch type to patch type but can also be variable within a patch type, for example if the pucks are placed randomly and (ii) on the robot environment interaction, e.g. sensor range, search strategy, motor control

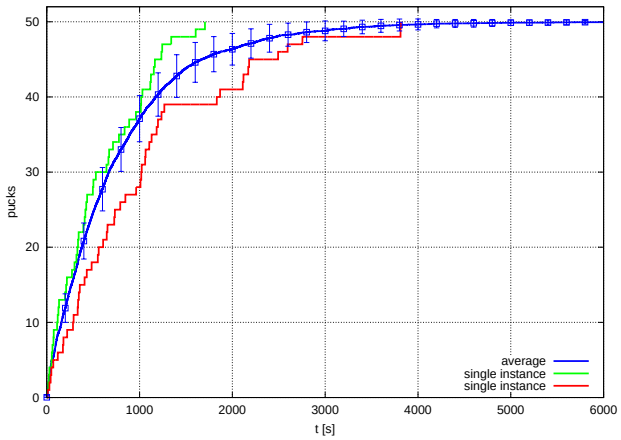


Figure 3: Average gain function (thin line) for random foraging in a 50 puck patch, error bars depict the standard deviation. Two instance of the gain function (thick lines) for patches with the same initial number of pucks.

etc. Thus foraging in two equally sized patches, initially containing the same number of pucks, that is patches with the same puck density, may result in two totally different gain functions and there is no way a forager can predict the gain function of a particular patch before entering the patch. Fig. 3 shows two exemplar gain functions and the average gain function over 100 patches (each patch with initially 50 pucks). Thus as McNamara (1982) argues, the subpatch type variance has to be considered. This immediately raises the question how does the forager determine the type of patch in which she is currently foraging ? In some scenarios the patch type might be detectable by an external cue, but in general it is not and the forager is required to forage in the patch in order to obtain information about the patch. This adds a patch discrimination problem to the decision process.

To overcome these issues, we suggest dropping the notion of patch types and treating each patch as its own type. (In the following we still use the phrase "patch type" to mean patches with the same initial number of pucks (same puck density), but we do not perform any form of rate maximization based on the notion of patch types.) For unique patches the long-term average gain rate is

$$R = \frac{\frac{1}{n}\sum_i^n g_i(t_i)}{\tau + \frac{1}{n}\sum_i^n t_i} \quad (3)$$

We replaced the patch type index j with index i referring to unique patches. The advantage of not having to distinguish patch types and not having to deal with patch subtype variance comes at the disadvantage of having a possibly very large planning horizon of n timesteps. In fact the planning horizon is the lifetime of the robot. Since the robot cannot predict the future, we avoid the large planning horizon by recursively maximizing Eq. 3 based on only past experiences

and ignoring possible future changes. Then our approximation of the long-term average gain rate while foraging in patch i, based on observations from previously encountered patches $0..i-1$ is

$$\tilde{R}_i = \frac{g_i(t_i) + G_i}{t_i + \tau + T_i} \quad (4)$$

Where G_i is the sum of collected pucks and T_i the total time (patch residence plus travel time) from all previous patches $0..i-1$. G_0 and T_0 can be used as a prior that provides the robot with an initial estimate of the average patch quality. Both G_i and T_i are a simple model of the average patch quality of the environment. This information (except the prior) is gained by the forager during exploitation. Hence a forager encountering only one patch type will actually maximize Eq. 2. But a forager first encountering a series of only low quality patches and then a series of high quality patches will maximize a very different average gain rate function than an omniscient forager. But an uninformed forager maximizing Eq. 4, will do as well as possible given the limited available information.

Robot Controller

The core of our task switching method is to maximize Eq. 4. This is done by numerically estimating the derivative of R_i at every time step and leaving the patch once the the derivative becomes zero. Since the gain function is assumed to be negatively accelerated, a maximum is found this way.

Algorithm 1 summarizes our task switching method. The robot forages for one time-step, if it collected a puck the local gain function $g(t)$ is incremented (line 10-15). Next we calculate an approximation of the long-term gain rate based on the experience from previous patches (G_i, T_i), an estimate of the travel time $\tilde{\tau}$ and the value of local gain function at the current time. Because of the stochastic and noisy nature of the gain function the estimate of the long-term gain rate has to be smoothed. In our implementation we use a low-pass filter (line 17-21). Other methods maybe substituted, however it performs well enough for our purpose. As mentioned earlier the patch leaving decision is based on checking if the derivative of the long-term gain rate is equal to zero. Again because of the stochasticity of the gain function we might experience a local region of zero or negative gradient, which could be interpreted as a local maximum. A simple counting step helps to overcome those undesired local maxima (line 22-27). As with the low-pass filter, any suitable method may substituted. The actual patch leaving decision is made in line 27. A patch is left once a maximum is found and a minimum amount of time has been spent in the patch. This minimum patch residence time is helpful during the initial time in a patch, since until the first puck is found $g(t) = 0$ would cause the robot to leave the patch immediately.

Once the robot leaves the patch it travels to the next patch. This travel takes τ_i time. Before starting to forage in the new patch the estimates for the environment quality G and T and the estimate of the switching time $\tilde{\tau}$ are updated (line 29-30).

```
1  Algorithm:patchMax
2  init G_0, T_0, τ̃, k_1, k_2, k_3, k_4
3  i = 1
4  forall patches do
5      enter patch i
6      t = 0
7      g(0) = 0
8      repeat
9          t = t + 1
10         randomly forage for one time-step
11         if puck collected then
12             g(t) = g(t−1) + 1
13         else
14             g(t) = g(t−1)
15         end
16         r(t) = (g(t)+G_i)/(t+τ̃+T_i)
17         if t == 1 then
18             r_filt(t) = r(t)
19         else
20             r_filt(t) = (1−k_3) r_filt(t−1) + k_3 r(t)
21         end
22         if r_filt(t) − r_filt(t−1) ≤ 0 then
23             c = c + 1
24         else
25             c = 0
26         end
27     until c > k_1 and t > k_2
28     move to next patch in τ_i time
29     G_{i+1} = G_i + g(t)
30     T_{i+1} = T_i + t + τ_i
31     τ̃ = τ̃ + k_4 (τ_i − τ̃)
32     i = i + 1
33 end
```

Algorithm 1: Task switching algorithm

Experiments

To investigate the effectiveness of our approach, we conducted a series of simulation experiments consisting of two phases (i) generate foraging data and (ii) test our task (patch) switching policy on the generated data (see Sec. Experimental Data). To generate the foraging data we used a generic mobile robot model in the well known simulator Stage (Vaughan, 2008). The robot is equipped with a short-range colour blob tracker to sense 'pucks', our unit of resources, in its vicinity. The robot knows (or equivalently can detect) the boundaries of a puck patch. Patches are 620 times the size of the robot, and contain initially 10, 30, 50, 100, 200 or 300 pucks placed uniformly at random. A min-

imum distance between pucks is enforced to avoid overlap. To exploit a patch, the robot randomly forages for pucks, by driving straight until it comes to the patch boundary, where it chooses a new heading that brings it back into the patch, at random. When a puck is detected, the robot servos towards the closest puck and collects it. Collecting a puck takes one simulation time step, so there is virtually no handling time. At each simulation time step we record how many pucks the robot has collected so far in the current patch: this is the gain function.

As mentioned earlier the gain function is not only dependent on the initial number of pucks per patch but also on the robot/environment interaction. To get a good sample of the distribution of gain functions, we randomly generate 100 patches of each of the six patch types and record the gain functions from the robot foraging in those patches. Note that at this point in the experiment no patch leaving decisions are made. The robot simply forages until the patch is exhausted and the simulation is terminated. Testing our approach on this recorded data set rather than during the robot simulation allows us to compare approaches on exactly the same data and it makes it feasible to determine a near-optimal solution by brute force solution search.

As a baseline for comparison we need to find a t_i for each patch such that the long-term gain rate is maximized. No closed form solution is known to this problem, and the gain functions are available as data points only. So we employ a brute force search. Since each patch is unique this technically requires us to solve Eq. 3 for all possible combinations of patch residence times. Because this is computationally prohibitive we resort to calculating the average gain function over all 100 instances of a patch type. Then we find the best patch residence time by solving Eq. 3 for all possible t ($0 \leq t \leq T_{patch\_exhaused}$) and selecting the t that maximizes the average gain rate. In case of multiple patch types we calculate the long-term gain rate for each combination of residence times on the average gain function. This is only feasible since the number of patch types considered is small.

In all of the following experiments we used the obtained long-term average gain rate as a metric for comparison. All algorithm parameters required were set manually and kept constant without any attempt to optimize them. The priors G_0 and T_0 were set to zero. To investigate our task switching method under a wide range of conditions we altered the task (patch) switching time τ from very short 10 seconds to very long 5×10^6 seconds (≈6 days). To put this in perspective we report the mean and standard deviation of observed times required to exhaustively forage patches in Table 1. The spectrum reaches from almost no switching cost to a switching cost about 200 times the average time required to exhaust a patch.

	initial pucks per patch					
	10	30	50	100	200	300
μ [s]	1858	2909	3631	4556	5171	5475
σ	825	1184	1271	1337	1206	1208

Table 1: Mean and standard deviation of the time required to exhaustively forage patches

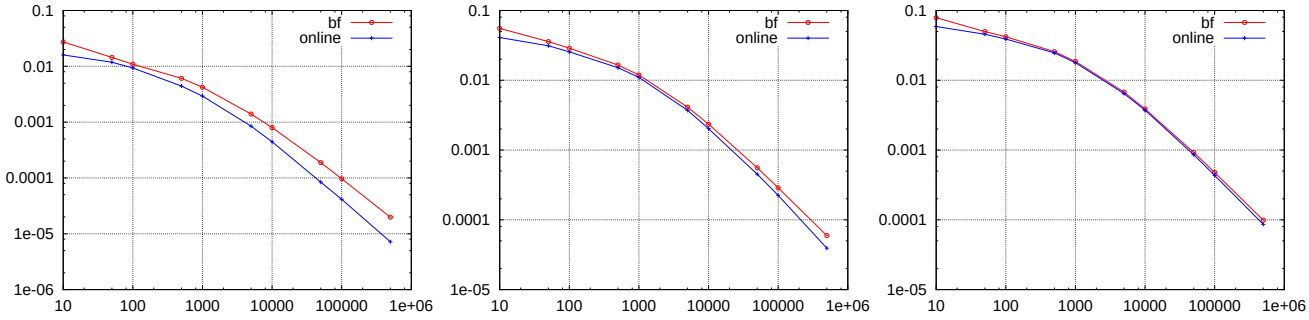
Single Patch Type

In a first experiment we had the robot forage in a series of 100 patches with the same initial number of pucks. Figure 4(a)-(f) shows the achieved long-term average gain rate for each patch type over a variety of switching times compared to the brute force solution. From the graphs we can draw three conclusions. (i) If the task switching times are short (i.e. much lower than the patch residence times) the performance of our method is in general lower than that of the near-optimal brute force method. The MVT predicts short patch residence times in situations where patch switching is cheap. But because of the various filters (filter parameters kept constant for all experiments over all conditions) our method's responsiveness is too slow in these short residence time situations. We say the performance is lower, but it is still above 78% (except in the 10 puck patches, where the performance drops to 50%). (ii) Under low patch quality situations (10 pucks, 30 pucks) our method performs less well than the brute force method. Again the reason is in the choice of parameters. The filters are too slow for the optimal, short patch residence time. (iii) The method described in this paper achieves similar long-term rates as the brute force method in all other cases examined. Recall that it uses only locally obtained information, in contrast to the omniscient brute force method.

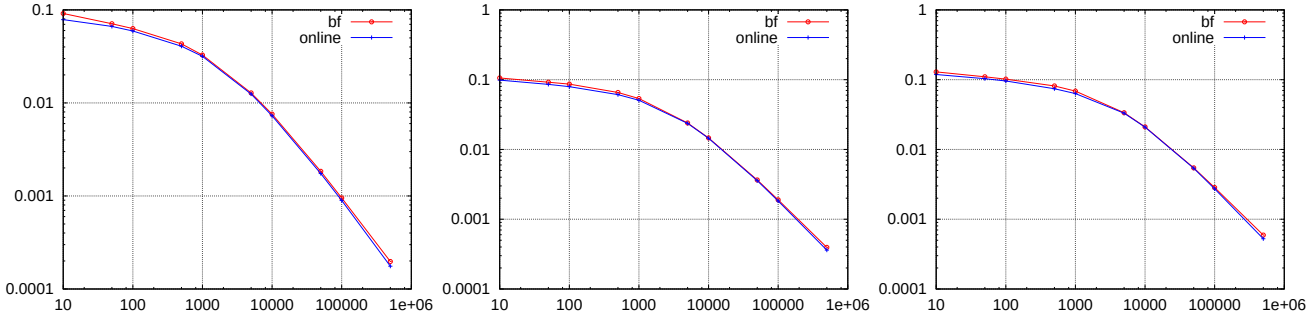
Multiple Patch Types

A more challenging problem is the case where patches of very different quality are encountered. As the MVT predicts the patch leaving decision is not only dependent on the quality of a given patch but on the global quality. To illustrate the difficulty of this decision we give a brief example. Let t_h be the optimal patch residence time if a forager only encounters patches of a fixed, high quality. If the same forager now encounters a mixture of high and low quality patches, t_h is no longer the optimal patch residence time for the high quality patches. The reason is that the cost of lost opportunity has increased due to the patches of low quality. As a consequence the forager should increase t_h under these circumstances.

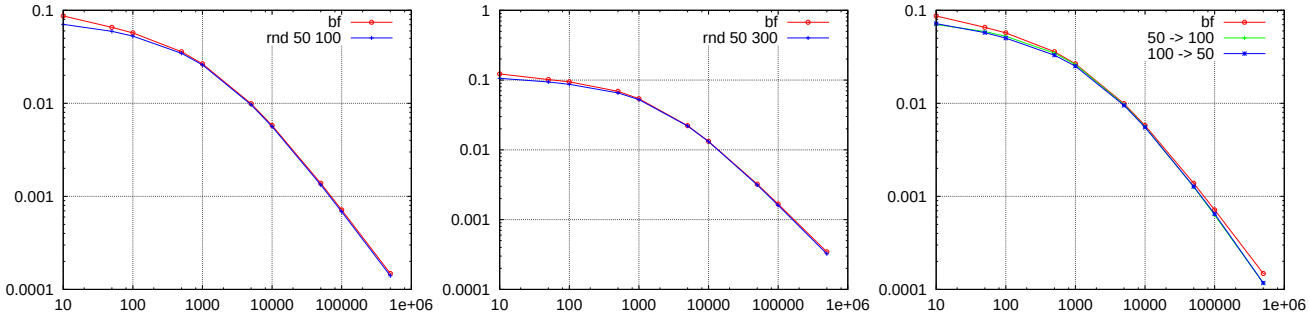
To investigate our system under these conditions we conducted a series of experiments. In a first experiment we had the robot encounter 100 patches of type A and 100 patches of type B in a random order. Figure 4(g) and 4(h) show the averaged results over 20 trials for patch configurations 50:100



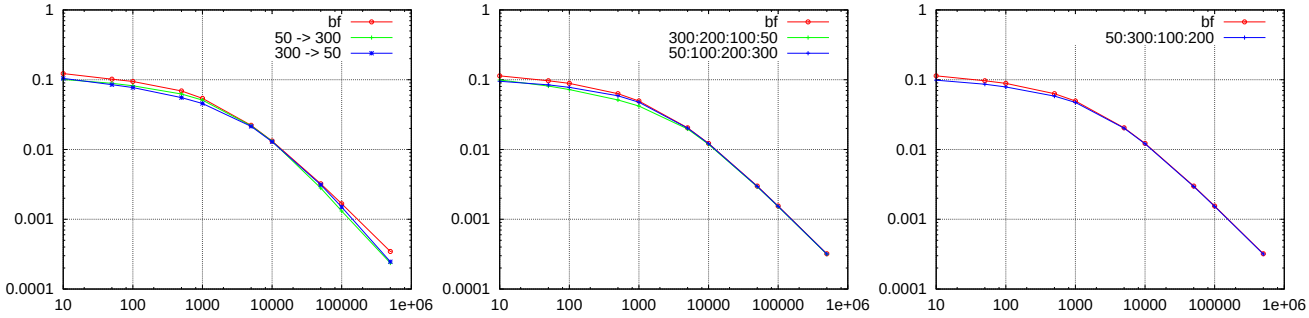
(a) Patch type 10 pucks (b) Patch type 30 pucks (c) Patch type 50 pucks



(d) Patch type 100 pucks (e) Patch type 200 pucks (f) Patch type 300 pucks



(g) Random sequence of patches with 50 and 100 pucks (h) Random sequence of patches with 50 and 300 pucks (i) Stepwise sequence of patches with 50 and 100 pucks



(j) Stepwise sequence of patches with 50 and 300 pucks (k) Stepwise sequence of patches with 50, 100, 200 and 300 pucks (l) Stepwise sequence of patches with 50, 300, 100 and 200 pucks

Figure 4: Long-term average gain rates achived by the bruteforce method (red line with circle) and our online method (green line with cross, blue with asterix). Inter patch travel time τ in seconds on the x-axis and long-term gain rate in pucks per seconds on the y-axis. More details in the text.

pucks and 50:300 pucks respectively. Errorbars were omitted because of the small standard deviation. As in the single patch type experiments and for the same reasons, the performance is somewhat lower under short switching time conditions, but in the general the graphs show that our method copes well with randomly encountered patches of different qualities.

An even harder problem is to encounter a longer series of patches of type A followed by a series of patches of type B, where the forager does not know anything about type B patches while it forages in type A patches. On encountering type B patches, the robot has built a strong prior expecting type A patches. In this experiment the robot was faced with a series of 100 patch of one type followed by 100 patch of a different type. The results for 50:100 and 50:300 patches with a stepwise change in both directions is shown in Fig. 4(i) and 4(j) respectively. Here the brute force method is at a significant advantage because the patch leaving decisions are derived with full knowledge of the future patch change. Our method does not/can not anticipate the patch quality change and thus for the first 100 patches acts under the "assumption" of a constant environment. The error resulting from this "assumption" grows with the difference in patch qualities. That is why the performance difference in the 50:300 scenario (Fig. 4(j)) is larger than in the 50:100 case (Fig. 4(i)).

Figure 4(k) shows the results for a stepwise sequence of 50:100:200:300 puck patches and the reverse ordering. The results are qualitatively very similar to those discussed previously. In one last experiment of this type we choose step wise patch encounter with larger step sizes. The ordering chosen was 50:300:100:200. Results are shown in Fig. 4(l). The performance results are again qualitatively similar, suggesting the our method handles this type of variance well.

Variable Switching Cost

So far we tested different switching costs but kept them constant in the single patch type as well as multi patch type experiments. To investigate varying inter-patch travel time, we conducted an experiment in which the travel time between patches was drawn from a normal distribution with mean 1000 seconds and standard deviation 100, 500 and 700 seconds respectively. Table 2 shows the results in percent compared to the long-term gain rate of the brute force solutions. Because of the computational complexity the brute force solution was only calculated using the mean and not the actual randomly drawn travel times. As in the previous experiments we see generally good performance and the usual drop in situations with low patch quality.

	initial pucks per patch					
σ	10	30	50	100	200	300
100	74.0	92.2	96.0	96.4	95.3	92.2
500	76.3	90.2	93.9	94.5	89.7	92.3
700	67.8	89.5	96.9	92.6	88.7	90.1

Table 2: Percent performance for variable patch switching time with mean 1000 sec. and standard deviation $\sigma = \{100, 500, 700\}$

Discussion

Task switching under diminishing returns is daily routine for many animals and important for many conceivable autonomous robots. Maximizing the long-term average gain or reward rate under these conditions requires the robot to have knowledge of future gain functions. This is not achievable by a robot relying solely on information obtained by its own actions. To the best of our knowledge no solution to this problem is known. In this paper we have argued that the MVT is not implementable because an instantaneous gain rate is meaningless in the case of rewards obtained in chunks. It also requires a continuous exploration phase in order to find the global maximum rate, but the MVT itself does not explore the action space.

Instead we proposed a task switching method that bases its decision only on previously obtained information, well aware that we therefore maximize a different function. Thus we may make suboptimal task switching decisions, but these decisions are as good as possible given no information about the future.

An important issue to discuss is how large the time window of past experiences should be, that are considered in the task-switching decision. In this paper we simply included all past foraging experiences when modelling the global patch quality. This is reasonable as long as the past is a good predictor for the future. On the other hand in situations where the future strongly deviates from the past, forgetting or a short memory can be beneficial. The memory size is also interesting from a behavioural ecology point of view, because it might explain why animals often appear to maximize the short-term and not the long-term intake rate (Real et al., 1990). In future it would be interesting to investigate what influence the memory size has on the rate maximization of a robot and what the optimal size is.

We draw a lot of insight from behavioural ecology, but we make no claims about mechanisms employed by animals.

Acknowledgements

The authors thank Yaroslav Litus for very valuable discussions on the work that led to this paper. This work was supported by NSERC and DRDC in Canada.

Experimental Data

In accordance with the Autonomy Lab's policy on code publication, the foraging data and the implementation of the experiments are made available online at `git://github.com/jwawerla/tsw_`

`experiment.git`. The exact data that led to the presented results can be accessed via the commit hash `4f84a82d09f2c181df57ab5d7faa2e53cc3348f3`.

References

Andrews, B. W., Passino, K. M., and Waite, T. A. (2007). Foraging theory for autonomous vehicle decision-making system design. *Journal of Intelligent and Robotic Systems*, 49:39–65.

Cao, Y. U., Fukunaga, A. S., and Kahng, A. B. (1997). Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4:226–234.

Charnov, E. L. (1976). Optimal foraging: Attack strategy of a mantid. *The American Naturalist*, 110:141–151.

Charnov, E. L. and Orians, G. H. (1973). Optimal foraging: Some theoretical explorations. Unpublished manuscript `http://hdl.handle.net/1928/1649`.

Danchin, É., Giraldeau, L.-A., and Cézilly, F., editors (2008). *Behavioural Ecology*. Oxford University Press.

Gibb, J. A. (1958). Predation by tits and squirrels on the eucosmid *Ernarmonia conicolana*. *Anim. Ecol.*, 27:375–396.

Green, R. F. (1984). Stopping rules for optimal foragers. *The American Naturalist*, 123(1):30–43.

Krebs, J., Ryan, J., and Charnov, E. (1974). Hunting by expectation or optimal foraging: A study of patch use by chickadees. *Animal Behaviour*, 22:953–964.

Krebs, J. R. (1973). *Perspectives in Ethology*, volume 1, chapter Behavioral aspects of predation, pages 73–111. Plenum New York.

McNamara, J. M. (1982). Optimal patch use in a stochastic environment. *Theoretical Population Biology*, 21(2):269–288.

Real, L., Ellner, S., and Harder, L. D. (1990). Short-term energy maximization and risk-aversion in bumble bees: A reply to Possingham et al. *Ecology*, 71(4):1625–1628.

Stephens, D. W., Brown, J. S., and Ydenberg, R. C., editors (2007). *Foraging - Behavior and Ecology*. University of Chicago Press.

Stephens, D. W. and Krebs, J. R. (1986). *Foraging Theory*. Princeton University Press.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement learning: an introduction*. MIT Press.

Vaughan, R. T. (2008). Massively multi-robot simulations in Stage. *Swarm Intelligence*, 2(2-4):189–208.

Wawerla, J. and Vaughan, R. T. (2009). Robot task switching under diminishing returns. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5033–5038.

Untethered Hovering Flapping Flight of a 3D-Printed Mechanical Insect

Charles Richter, Hod Lipson

Computational Synthesis Laboratory
Mechanical & Aerospace Engineering
Cornell University, Ithaca NY 14853, USA
+1 (607) 255 1686
hod.lipson@cornell.edu

Abstract

This project focuses on developing a flapping-wing hovering insect using 3D printed wings and mechanical parts. The use of 3D printing technology has greatly expanded the possibilities for wing design, allowing wing shapes to replicate those of real insects or virtually any other shape. It has also reduced the time of a wing design cycle to a matter of minutes. An ornithopter with a mass of 3.89g has been constructed using the 3D printing technique and has demonstrated an 85-second passively stable untethered hovering flight. This flight exhibits the functional utility of printed materials for flapping wing experimentation and ornithopter construction and for understanding the mechanical principles underlying insect flight and control.

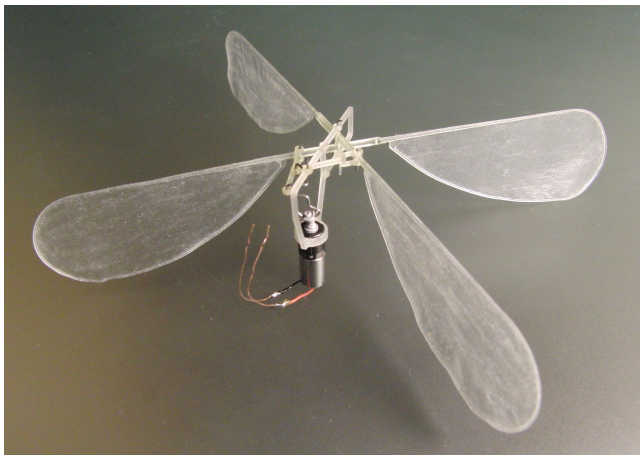


Figure 1: 3D-Printed elements of flapping-hovering insect.

Introduction

Hovering flapping flight of insects and birds has long fascinated scientists and engineers, but only in the last decade has it been successfully demonstrated by man-made flying machines. Unlike forward flight, hovering flapping flight poses several special challenges. First, there has yet to emerge an established body of theoretical and experimental work on the unsteady aerodynamics of flapping wing flight for the purposes of wing design. Second, flapping hovering flight of insects and birds is generally unstable and requires a sophisticated solution to maintain an upright flying position (Taylor and Thomas, 2003; Sun and Xiong, 2005). Third, the energy density of batteries was insufficient for the power demands of hovering flight until small lithium-based batteries became widely available. However with the improvement of electrical power solutions, a number of successful hovering ornithopters have been developed with a variety of wing designs. This project utilizes existing solutions to the power and stability problems and uses 3D printing as a novel approach to designing and manufacturing the key aerodynamic component: the wings.

Thus far, producing effective flapping wings for research and ornithopter construction has been a time consuming and delicate process taking days or longer to complete. The 3D printing technique allows wings to be produced in a matter of minutes, dramatically reducing the time of each design cycle. Overcoming this barrier to experimentation will allow a comprehensive study of lift production for a wide variety of wing shapes including those replicating real insect wings.

A comprehensive understanding of flapping wing aerodynamics and hovering flight will become increasingly important as ornithopters shrink to the scale of real insects where some advantages of flapping wing flight are realized (Ellington, 1999). These advantages include efficiency and maneuverability improvements over fixed and rotary wing aircraft at low Reynolds numbers as well as the suitability of micro-scale actuators to producing vibrating motion for flapping rather than rotary motion for traditional propellers (Pesavento and Wang, 2009; Woods et al., 2001). Maneuverable, low-power micro air vehicles have a wide range of applications including mapping, surveillance and search-and-rescue operations where these properties of small size and ability to maneuver in tight spaces are vital, or in thin extraterrestrial atmospheres where low Reynolds numbers occur (Michelson and Naqvi, 2003). Micro air vehicles also present a challenging synthesis of many areas of engineering, including materials, actuators, electronics, control, vision, guidance, and others (Floreano et al., 2010; Karpel-

Design	Year	Mass (g)	Span (cm)	Wings	Hover Time (s)	Features
Mentor (Zdunich, 2007)	2002	580	36	4	> 60	Nitromethane Fuel
DelFly II (DelFly, 2010)	2006	16.07	28	4	480	Camera, R/C
van Breugel (van Breugel et al., 2008)	2007	24.2	45	8	33	Passively Stable
Chronister (Chronister, 2010)	2007	3.3	15	4	Unknown	R/C
Wood (Wood, 2008)	2007	0.060	3	2	N/A	Piezoelectric Power
DelFly Micro (DelFly, 2010)	2008	3.07	10	4	N/A	Camera, R/C
NAV (AeroVironment, 2009)	2009	10 (est.)	7.5 (est.)	2	20	Active Wing Pitching
Richter (this paper)	2010	3.89	14.3	4	85	3D Printed Parts/Wings

Table 1: Characteristics of existing ornithopter designs.

son et al., 2008). This project has demonstrated the viability of 3D printed aerodynamic components for experimentation and for use in a real ornithopter on the size scale of the smallest current designs.

Review of Existing Work

The existing work that has influenced this project includes a variety of successful ornithopter designs and some research on the dynamics and control of insect flight. This project is effectively a continuation of an earlier ornithopter design project by Floris van Breugel of the Cornell Computational Synthesis Laboratory. Van Breugel's design used four motors to drive eight wings and featured passively stable flight dynamics using a set of damping sails above and below the body of the aircraft. This model had a mass of 24g and demonstrated stable hovering flight of over 30 seconds in 2007. Broad goals for the current project were to achieve a comparable flight time using this system of passive stability in a vehicle under 10g.

Several other successful designs currently exist, including the series of DelFly ornithopters, which are radio controlled using tail configurations resembling fixed-wing aircraft and the AeroVironment Nano Air Vehicle, which achieves control using active wing control. The Harvard Microrobotics Laboratory has also produced ornithopters weighing 60 mg using piezoelectric actuators and insect-like passive wing pitching, but require a tether for power and stability.

There have also been recent developments in the understanding of insect flight (Dickinson et al., 1999; Wang, 2005; Bergou et al., 2007; Ristoph et al., 2009). These studies have explored one mechanism of passive wing deflection in insect flight that is essential to the simplicity of some ornithopter designs. They have shown that some insect wings deflect to an angle of incidence of 45 degrees, which is thought to be optimal for lift production of a flat plate wing. These findings have also given rise to hypotheses explaining forward thrust, flight maneuvers and disturbance rejection, and experiments have been designed to examine these hypotheses using the ornithopter as a test bed.

Methods

One primary goal of this project was to produce a hovering ornithopter with as many 3D printed components as possible. An Objet EDEN260V printer and the Objet FullCure 720 material were used to produce all printed components. This material costs roughly 0.22 USD per gram and the EDEN 260V prints with a resolution of 42 μm on the x- and y-axes and 16 μm on the z-axis. At first, only the fuselage, hinges and pushrods were printed, however a method of printing entire one-piece wings was soon developed.

First attempts at wing construction were aimed at recre-

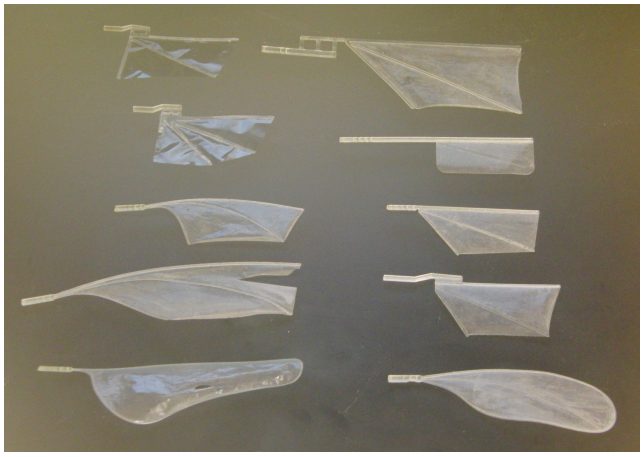


Figure 2: A variety of wing shapes for experimentation.

Figure 3: The most successful wing design during testing.

ating the wings of the van Breugel design, using a carbon fiber rod as the main strut, polyethylene terephthalate (PET) stiffening ribs and a Mylar film wing surface. Two examples of this early printed type can be seen in the upper left corner of Fig. 2. The carbon fiber rod was to extend out of a 3D printed hinge, but after several design iterations, the hinge, strut and stiffening ribs were combined into a single printed piece. When further experimentation revealed that a durable thin film could be printed using only two layers of printed material, this film was used instead of Mylar as the wing surface and the first one-piece printed wings were made. Fig. 2 shows many conventional and biologically inspired printed wings.

Printed Wing Construction

The printed wings of the ornithopter are comprised of three functional elements: the central beam, the surrounding frame, and the thin film wing surface. Fig. 4 shows the parts of the dual-wing used in the full ornithopter design.

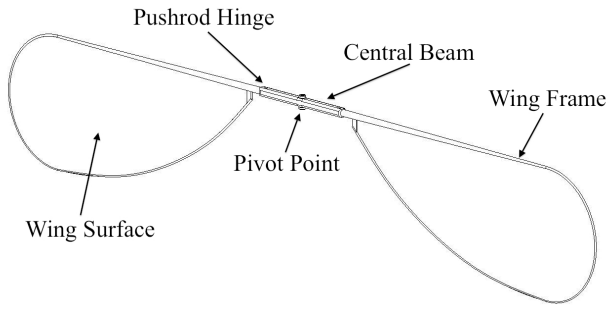


Figure 4: Parts of the one-piece printed wing.

The central beam is the most rigid portion of the wing and contains the pivot point as well as the attachment holes for the connecting rods. Whereas some designs require a bushing or dedicated hinge, 3D printing allows the hinge to be incorporated into the main beam design. Furthermore, the FullCure 720 material features relatively low friction against the stainless steel 0.5 mm piano wire hinge pins when lubricated with a drop of medium-viscosity oil. The holes for the pivot points were designed with a 0.6 mm diameter to provide an adequate gap for low-friction operation. This technique eliminates the need for a heavy bushing or complex assembly.

The outer frames of the wings are attached to the ends of the beam. The outer frames determine the flexibility of the wings and the deflection properties during flapping. The outer frames were defined in the CAD model as lofted curves connecting circular cross sections. By varying the radius of the circular cross sections at various points along the frame, the overall stiffness and flexibility patterns of the wing could be tuned.

The thin wing surface is a flexible film that extends through the area inside the outer frame. The surface has a thickness of 40μm, which is achieved by depositing two layers of material. The ability of the printer to print such a thin flexible film is the development that made a one-piece printed wing possible. While it is possible to print a thinner film using a single layer, wings constructed with a single layer surface are extremely delicate and tend to tear upon vigorous flapping. Chamfers were used to counter the tendency of the wing film to tear at points of discontinuous geometry, such as the edge where the film joins the frame.

One practical element of 3D printing technology is the use of a gelatinous material to support the structure during printing. Therefore, removing the support material is an important step in the manufacturing process, especially with delicate features such as the thin wing surface. Common methods used to remove support material include dissolving it in sodium hydroxide and spraying it off with pressurized water. However, both of these methods have limitations due to the delicacy of the thin film. When a printed wing is soaked in liquid for any period of time, it tends to curl up or become warped, which can be partially corrected by pressing it flat and allowing it to dry. However, the moisture tends to leave some permanent warping of the wing shape. The method of spraying pressurized water is also difficult because extreme care must be taken to avoid tearing the wing film. Again, the moisture tends to warp the wing shape. The best method thus far has been to place the wing on a clean surface with some elasticity such as a dense rubber mat and scrape the support material away using a dull blade. Any residual material can be removed by wiping with a cloth moistened with water or rubbing alcohol. This is the fastest and most successful method for removing support material from the thin wing film.

Wing Design

At the beginning of the project, the wing design process focused on narrowing the vast design space to a size scale that was appropriate for the motors available and desired weight of the vehicle. During initial testing, key wing design features were identified that helped produce the ideal shapes and deflections when flapping. Testing of a wide variety of wing shapes, sizes and structures was carried out by powering them with a small DC gear-motor using a DC power source. The lift of each wing was measured using a custom attachment for a digital lab scale and flapping behavior was analyzed using a high-speed camera capturing 1000 frames per second. Fig. 5 shows the experimental apparatus.

The wing size partially determines several important variables, including mass and surface area, which in turn determine how fast the wings can flap for a given power input. For the motor chosen for this project (GM 15 gear motor available from Solarbotics.com with 25:1 gear reduction) and the power expected from a pair of Lithium- Polymer bat-

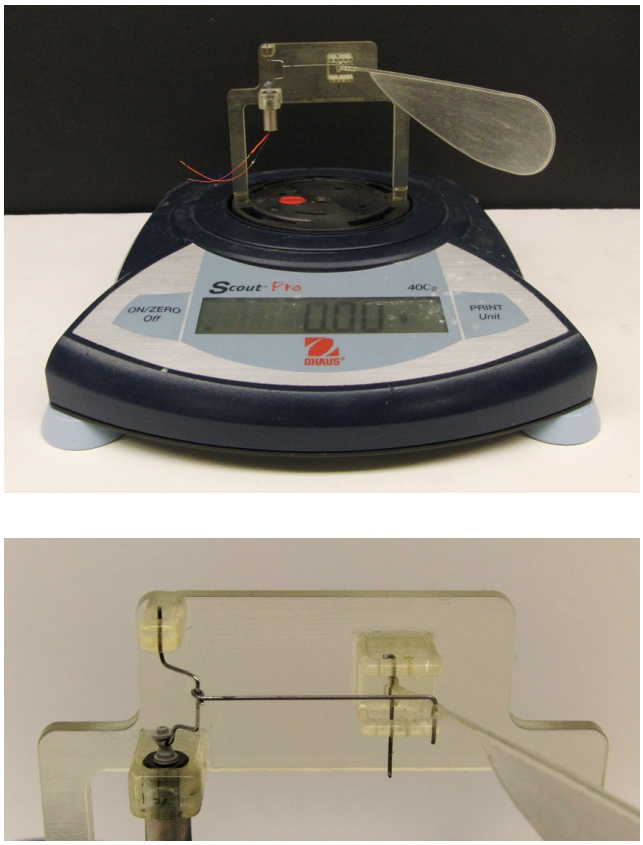


Figure 5: Experimental test setup on the lab scale (above); Close-up of mechanism (below).

teries (7.4V , 200mA), the best performing single wing of all wing designs tested had a length of 80 mm and a maximum chord of 30 mm. The overall weight of the wing was approximately 0.3g and the thickness of the wing film was 40 μm. This wing flapped at approximately 30 Hz through an angle of 110 degrees and produced a maximum lift force of 2.92g. This wing design is shown in Fig. 3.

The wing structure is important to proper deflection and wing shape during flapping. For maximum lift, the wing should deflect to an angle of attack of roughly 45 degrees at the middle of the stroke. This angle of attack can be tuned by adjusting the flexibility of the main wing strut and the ribs that stiffen the interior of the wing. Thus far, successful wing designs have been created with and without wing ribs.

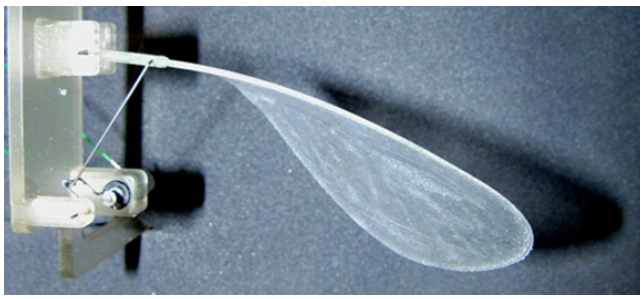
One major problem associated with simple deflecting wings is that they do not deflect as flat plates. Instead, the leading edge tends to remain vertical rather than flexing torsionally, while the wing surface bends away underneath it. This behavior creates an inverted camber shape that is undesirable. Several methods were explored to overcome this problem. The most effective solution was to extend the wing frame all the way around the tip of the wing. This design forced the leading edge to twist when the wing deflected, thus maintaining a roughly continuous slope across the chord of the wing near the tip. In other words, the tip of the wing behaved more like a flat plate with the entire wing deflecting to the proper angle, rather than just the lower half.

Wing ribs have also been used to control the deflection patterns and add stiffness in certain directions. Various rib designs were tested, featuring rectilinear patterns as well as curved patterns inspired by the wings of dragonflies and other insects. However, the current design does not feature stiffening ribs. Fig. 6 shows a top-down view of a wing deflecting during flapping tests on the experimental setup.

This general wing design, while not optimal, was deemed satisfactory for use in the challenge of building a full ornithopter using 3D printed wings. A new double-ended version of this wing shape was produced for use in the full ornithopter.

Full Ornithopter Design

Once a satisfactory wing design was obtained, it was implemented in the four-wing vehicle. The wing chosen for this purpose was the rib-less design that produced the greatest lift. A fuselage was designed to hold the motor, crank, and wing hinge. Care was taken to place the motor as close as possible to the wing pivot point to center the mass.



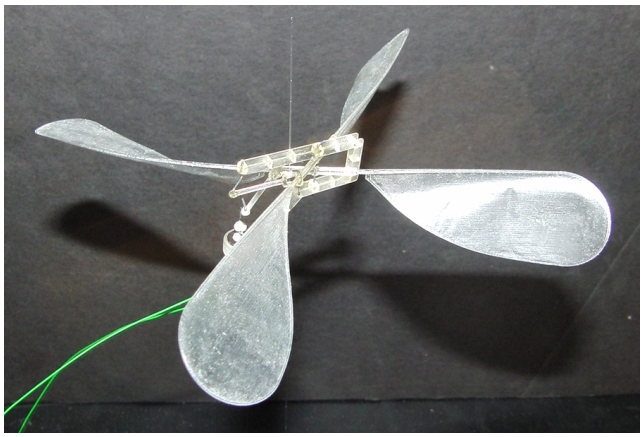


Figure 6: Flash photos showing deflection while flapping (above); wing deflection in a tethered flight test (below).

The wings are driven by a crankshaft connected to the motors gearbox. In order to drive the wings in a roughly symmetrical motion, the crankshaft includes two attachment points for the connecting rods powering the left and right wing. These two attachment points are roughly 30 degrees out of phase from each other to compensate for the asymmetry of the crank position at any given point in the stroke. Fig. 7 shows a top view of this offset-crank mechanism, which is similar to the DelFly I design (de Croon et al., 2009) and many toy ornithopters.

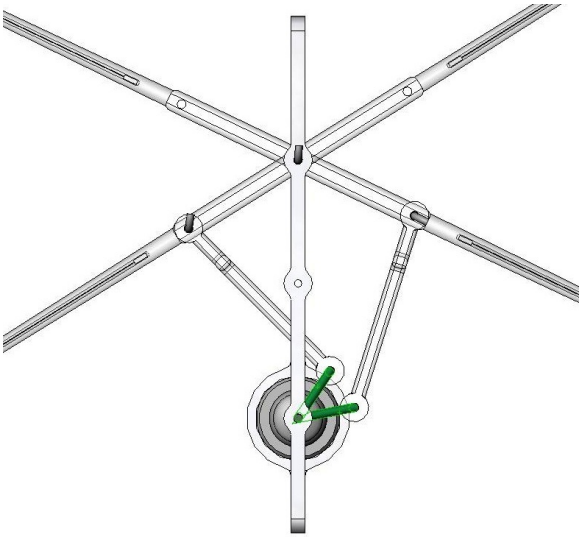


Figure 7: Top view of ornithopter with offset-crank in green.

The ornithopter was tested first using a DC power source and a fishing line tether to verify proper operation of the crank mechanism and proper flapping behavior of the wings. The crank is designed to flap each of the four wings through roughly 80 degrees, and when the flexibility of the wings is included, this angle is enough to allow the wings to clap and fling at the end of each stroke. The clap and fling phenomenon may aid in lift production (Lehmann et al., 2005). Fig. 6 shows a photo of a tethered flight test showing ideal wing deflection of roughly 45 degrees. In this test configuration, the ornithopter was able to lift up to 1.5g of payload, which is roughly equivalent to the mass of batteries required for flight.

Once the ornithopter was able to support a payload while flying on the tether, it was outfitted with batteries and untethered flight tests began. Two 10mAh Lithium Polymer batteries were used to power the motor and were attached on the opposite side to the motor to balance the mass. The other feature required for untethered flight is a set of thin foam damping sails attached to a thin carbon fiber rod above and below the fuselage to maintain an upright flying position. This method of achieving passive stability was developed by van Breugel and is replicated here (van Breugel et al., 2008).

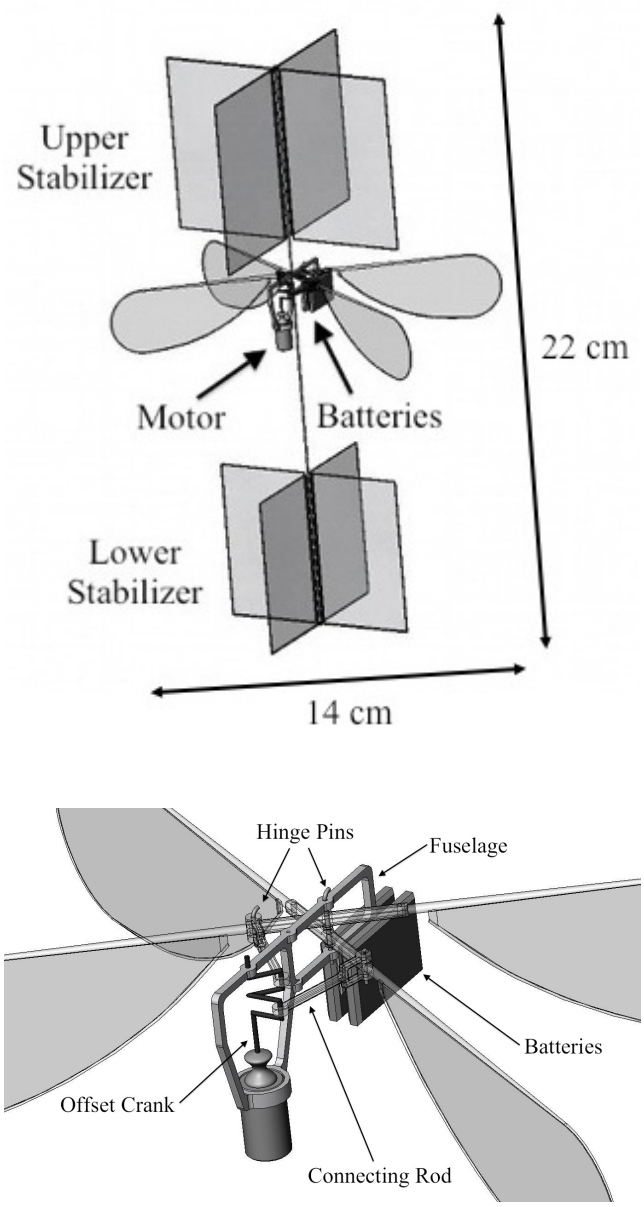


Figure 8: Final configuration and large view of mechanism.

Passive Stability

The sails employed to maintain stability help keep the ornithopter upright. Without sails, the ornithopter tends to tip over, causing a loss of upward lift. However, when the sails are attached, the larger top sail acts as a damper on the tendency to tip over, which allows the bottom sail to swing back under the fuselage, righting the ornithopter. The bottom sail is just large enough to dampen any oscillation when it swings. If launched upside down, the ornithopter will right itself, demonstrating the robustness of the design.

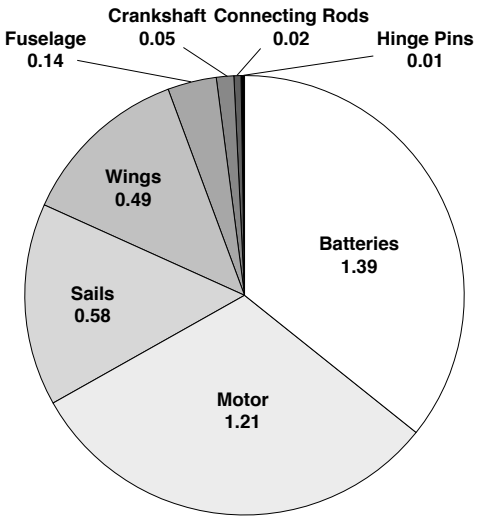


Figure 9: Breakdown of total mass (3.89g).

Conclusions

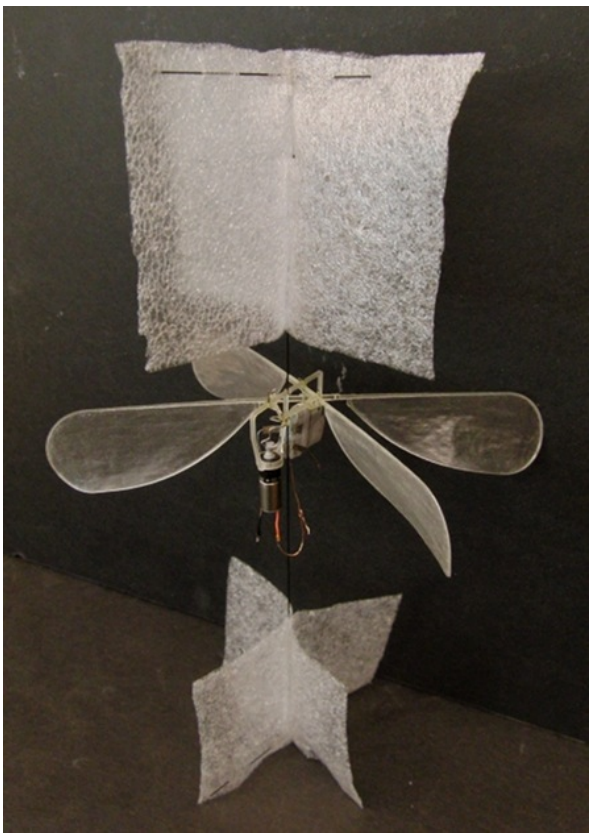
This project has yielded several significant results thus far. First, wing tests and the hovering demonstration have validated the concept of a printed ornithopter. This method of construction has greatly accelerated the design cycle, since a set of wings can be printed in less than 30 minutes and a complete set of ornithopter parts can be printed in 60 minutes. Thus, several design iterations can be tested per day.

The Objet FullCure 720 material has some limitations, particularly in its mechanical properties. It is not as light or as stiff as carbon fiber or balsa wood, which are the main alternative options for wing struts. Therefore, printed wings do not store as much energy when they flex and energy is lost to friction during each wing stroke. Different strut cross sections will be tested to improve stiffness per volume of material.

Other limitations of the 3D printed material include a tendency of thin wings to curl up after a period of days, rendering them useless. This problem can be corrected by storing wings between flat plates or in the pages of a book, which requires disassembly. Thin wings also tend to develop small tears after minutes of vigorous flapping, however this problem can be partially prevented with chamfered edges along the wing frame to avoid discontinuous geometry.

Experimentation with wing designs has begun to uncover some of the features and parameters of successful wings for this size and power scale. The GM15 motor seems to be well matched to wings that are approximately 80-100 mm long from base to tip with a chord length of 30-40 mm when it is running at a power of 1.5W (typical power consumption during flight). If the wing strut is extended further, then the drag of the wing acts along a longer lever arm, slowing down the rate of flapping and reducing lift.

One very successful design feature is the wing frame that extends around the wingtip. This feature helps maintain a continuous wing slope at the tip of the wing and helps approximate the flat-plate airfoil cross section of many hovering insects. The continuous wingtip frame was a design borrowed from the structure of dragonfly wings, which exhibit ideal shape and deflection at the wingtips. Overall, the use of 3D printing to create flexible wings that are aerodynamically functional is the main accomplishment of this project and will be one area for future improvement.



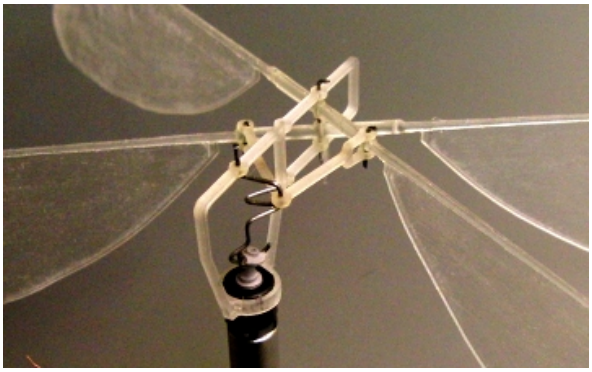


Figure 10: Final design with sails and mechanism close-up.

Figure 11: Ornithopter taking flight and hovering.

Future Work

A long-term project utilizing a hovering ornithopter will be to test hypotheses of insect propulsion and control. This project will be carried out by building wings with a nominal bias of several degrees built into the angle of incidence to produce forward thrust or turning maneuvers. If successful, these principles could form the basis of hovering ornithopter control.

Another project planned for the future is to perform a detailed study using 3D printed wings to develop analytical models predicting wing performance. The lift of many different wing designs will be measured to identify relationships between the major variables involved in lift production such as wing length, chord, surface area, flapping frequency, parameterized shape, etc. This data will then be mined for analytical relationships using the Eureqa software (Schmidt and Lipson, 2009). These laws will then be compared with current designs to evaluate the model and ultimately produce the best possible wings.

Finally, another ornithopter will be designed using 3D printed wings and other parts that is still smaller and lighter and is composed of an even greater proportion of printed components.

Acknowledgements

This work was supported in part by the U.S. National Science Foundation (NSF) Grant ECCS 0941561 on Cyber-enabled Discovery and Innovation (CDI). We thank Leif Ristroph and Itai Cohen for useful discussions.

References

AeroVironment (2009). Darpa awards aerovironment phase ii contract extension for nano air vehicle development program [online]. Accessed 4/9/10. http://www.avinc.com/downloads/NAVPRLongDARPAV4.doc.pdf.

Bergou, A. J., Xu, S., and Wang, Z. J. (2007). Passive wing pitch reversal in insect flight. *J. Fluid Mech.*, 591:321–337.

Chronister, N. (2010). Micro air vehicle ornithopters [online]. Accessed 4/9/10. http://www.ornithopter.org/history.mav.shtml.

de Croon, G. C. H. E., de Clerg, K. M. E., Ruijsink, R., Remes, B., and de Wagter, C. (2009). Design, aerodynamics, and vision-based control of the delfly. *Int. J. of Micro Air Vehicles*, 1(2):71–97.

DelFly (2010). Accessed 4/9/10. http://www.delfly.nl/.

Dickinson, M. H., Lehmann, F., and Sanjay, P. (1999). Wing rotation and the aerodynamic basis of insect flight. *Science*, 284:1954–1960.

Ellington, C. P. (1999). The novel aerodynamics of insect flight: Applications to micro-air vehicles. *J. Exp. Biol.*, 202(23):3439–3448.

Floreano, D., Zufferey, J. C., Srinivasan, M., and Ellington, C., editors (2010). *Flying Insects and Robots*. Springer.

A Navigating Rat Animat

David Ball[1], Scott Heath[1], Michael Milford[2], Gordon Wyeth[2] and Janet Wiles[1]

[1]The University of Queensland, Australia
[2]Queensland University of Technology, Australia
{dball, wiles}@itee.uq.edu.au
{gordon.wyeth, michael.milford}@qut.edu.au

Abstract

This paper presents a new rat animat, a rat-sized bio-inspired robot platform currently being developed for embodied cognition and neuroscience research. The rodent animat is 150mm x 80mm x 70mm and has a differential drive, visual, proximity, and odometry sensors, x86 PC, and LCD interface. The rat animat has a bio-inspired rodent navigation and mapping system called RatSLAM which demonstrates the capabilities of the platform and framework. A case study is presented of the robot's ability to learn the spatial layout of a figure of eight laboratory environment, including its ability to close physical loops based on visual input and odometry. A firing field plot similar to rodent 'non-conjunctive grid cells' is shown by plotting the activity of an internal network. Having a rodent animat the size of a real rat allows exploration of embodiment issues such as how the robot's sensori-motor systems and cognitive abilities interact. The initial observations concern the limitations of the design as well as its strengths. For example, the visual sensor has a narrower field of view and is located much closer to the ground than for other robots in the lab, which alters the salience of visual cues and the effectiveness of different visual filtering techniques. The small size of the robot relative to corridors and open areas impacts on the possible trajectories of the robot. These perspective and size issues affect the formation and use of the cognitive map, and hence the navigation abilities of the rat animat.

Introduction

Brains are evolved to control bodies, which have characteristic sizes, and live in specific environments. One approach to studying embodiment is to develop animats (Wilson, 1991), which are robots that mimic specific animals that enable the study of the integrated system formed by brain, body and environment (Beer, 2008; Beer & Williams, 2009). Animats also enable comparisons with the behavior of the corresponding animal on similar tasks, which can lead to the co-development of animats with animal laboratory studies. No animat perfectly mimics their biological counterpart, and priorities need to be established for the animat design.

Bio-inspired robotics is a growing field that draws insights from nature's solutions for interacting with real-world environments. A major research question in bio-inspired robotics is the design and evaluation of effective algorithms for embodied learning and action. In particular, rodents have been well-studied both biologically and for bio-inspired technologies. Rodents have excellent mobility, and interactions are particularly important for survival both within peripersonal space (the space within reach of the animal) and wider aspects of navigation in geopersonal space (the space that the agent can move through beyond its current location). Rodents have proved an effective match between embodied ability, brain complexity and current state-of-the-art in neuroscience. Embodiment itself can reduce the complexity of control architectures and improve energy efficiency (Brooks, 1991).

Bio-mimicry is often used as a more targeted term to develop engineering solutions that not only develop algorithms based on animal morphology and behaviour, but also that aim to preserve a high fidelity with the target system. This research focuses on bio-mimicry which has the potential to benefit biology as well as engineering, as discussed in detail in the extensive article and commentaries in (Webb, 2000, 2001).

In robotics, a significant engineering design aspect is the tradeoff between size and capabilities. Capabilities include sensing, actuation and computation. For a rat animat the size is given by the real animal. However, it is not always possible to integrate the desired capabilities into an animat the size of the real animal. The robot can be designed with only those capabilities that fit into the size of the real animal, or the robot's size can be increased to accommodate the full complement of desired capabilities. Setting the first design requirement to be a match between the size of the robot and the animal enables the study of aspects of embodiment and the physical context that are not possible in larger animats.

Body size places strong constraints on an animat, just as it does on an animal's abilities, including its navigational abilities and the range of its behavior. Size is rarely given precedence in design criteria in embodied systems, but to test the rat animat on the same laboratory tasks as real rats, size becomes a defining feature in our research. Physical size places strong constraints on power available for movement and computational abilities. Size also impacts on possible physical sensori-motor configuration. For example, with respect to sensors, the visual field perspective is impacted by the height of the camera, and for motor control, the power of the motors and size of the wheels impact on the range and terrain that the robot can cover.

Existing robot rats can be broadly categorized from an engineering point of view into two categories: those with computational capacity equivalent to a standard PC but larger than a rat, and those the size of rat but with reduced or custom

computational capacity. The recent availability of small x86 platforms (that allow a standard Windows or Linux OS) has allowed for a reduction in the size of robots without compromising on computational capacity. This paper describes a new rat animat that takes advantage of the recent miniaturization of PC equivalent computational parts to build a rat sized robot platform.

RatSLAM is a bio-inspired navigation system based on the rodent hippocampus, which uses visual appearance as the primary mechanism for localization (Milford & Wyeth, 2009). Previous studies have been performed on a robot where the visual sensor is approximately 0.5m from the ground. The rat's eyes are an order of magnitude lower at a height of approximately 0.05m above the ground. The nature and quality of information in different parts of the visual field is impacted by the location of the camera, and hence the perspective of the robot.

The next section reviews existing rodent animat platforms and rodent inspired navigation system. The following section describes the new rodent animat platform and the RatSLAM system. Then the paper describes the focus study for this paper where the rat animat maps a figure of eight environment. Then the results of the navigation studies, including the resultant topological map and 'place fields' are described. The final section provides discussion, including directions for future work, before the paper concludes.

Background

Robot rat studies to date have developed many components for building a rat-like robot, but either the size is much larger than a real rat, or the computational capabilities have limited low fidelity bio-mimicry. The AMouse (Fend, 2004) has two whisker arrays and an omnidirectional camera. The robot uses whiskers to ensure robust obstacle navigation in changing light conditions integrated into a subsumption architecture. The camera and whisker were separate modules added to the Khepera robot platform.

Psikharpax is a rat animat, with sensors, actuators and control architectures closely inspired by the rat (Meyera et al., 2005). Mechanically, the rat is 500mm long and has two wheels that allow a maximum speed of 0.3m/s. Psikharpax can rear and grasp objects with its foreleg and can move its head and eyes. The sensors include two visual sensors, an auditory system and a 32 whisker haptic system. A bio-mimetic chip capable of low-level real time signal processing for sensor fusion is under design. Recently an omni-directional visual system has been added (Lacheze, Benosman, & Meyer, 2008).

Alternatively, the Cyber Rodent project has less emphasis on physical bio-mimicry, rather taking its inspiration from neuromodulation (in particular dopamine, serotonin, acetylcholine and noradrenaline), and uses self-preservation and self-reproduction in a reinforcement learning framework to understand the biological reward system (Doya & Uchibe, 2005). The robot is larger than a typical rodent, 220mm long and weighs 1.75kg and has two wheels that allow a maximum speed of 1.3m/s. Sensors include a camera, range and proximity sensors, gyros and accelerometers, microphones. For communication the robot has a speaker and tri-color LED. Computationally, it has custom embedded hardware for on-robot learning.

There are a number of robot rats that are focuses on the embodiment of the whisker system (Fend, Bovet, & Pfeifer, 2006; Fox, Mitchinson, Pearson, Pipe, & Prescott, 2009; Pearson, Pipe, Melhuish, Mitchinson, & Prescott, 2007). These robots explore vibrissal sensory processing for texture discrimination, obstacle detection and wall following. A number of different sensors, whisker materials, whisker actuation methods and computational processing techniques have been explored.

Robot rats also interact with real rodents in a laboratory. Waseda Mouse-No.2 (WM-2) (1998) has a similar size and mass to rat, uses a fur coat to achieve a similar appearance and uses wheels for mobility. An embedded microcontroller handles sensors, motors and communication with the host computer over an IR link. They demonstrated that a real rat recognized the movement of WM-2, and that the robot influenced the rat's behavior, helping it to learn response to stimulations. WM-6 added arms at the front for interacting with levers (2006). WM-6 uses Bluetooth to communicate wirelessly with the host computer. Patanè, Mattoli et al. (2007) has increased the complexity of the interaction possible by using a legged robot rat. The host computer is responsible for autonomous control of the robot via overhead vision. The robot successfully taught the rat a lever pushing task to get food.

Rodent bio-inspired navigation

There has been extensive work investigating how animals navigate, in particular towards the goal of understanding how the rodent's hippocampus and associated regions work to localize, map and navigate an environment. These biological studies have formed the basis for many rodent-inspired robot navigation systems. Cells with a range of specific functions have been found including head-direction cells (Ranck Jr, 1984), place cells (O'Keefe & Conway, 1978), and grid-cells (Hafting, Fyhn, Molden, Moser, & Moser, 2005). There are several approaches to apply these insights to robot navigation ranging from those that try and mimic the biological studies as closely as possible to those that use them as inspiration but apply an engineering approach.

Early work by Mataric (1991) used a layers-of-competence subsumption architecture on a custom robot with sonar sensors. Burgess and Donnett et al. (1997) developed a simulation of neuronal place cells and "goal" cells to create mapping and navigation abilities on a K-Team Khepera robot. Meyer, Guillota et al. (2005) base their navigation system on place cells and behavioral system and are applying it to their large rat animat, Psikharpax, described previously. Alternatively, Arleo and Gerstner's (2000) approach more closely emulates biological place cells and was demonstrated using a K-Team Khepera robot in a small environment with artificial textures. Barrera and Weitzenfeld et al. (2008) demonstrated their biologically inspired spatially cognitive work in a typical wet lab experimental setting using a Sony AIBO. Milford and Wyeth (2009) focused on using place cell biology as an inspiration to engineer a complete robot navigation solution on an ActiveMedia Pioneer robot.

RoboRat platform

Given the research to date on rodent animats, there is an opportunity to integrate many of the existing ideas, extending them where necessary, and develop a robot rat-mimic which has the size and navigation abilities to operate in the same environments as real rats, challenged with the same tasks, and controlled by neural-inspired algorithms. Such a rat animat could be used to study embodiment issues in robotics, test theories of the neural basis of mammalian navigation, and also has the potential to open new areas of behavioral study through interaction with real rats. In this paper, we address the first goal, that of developing a rat-size robot to use as an integrated development platform.

A (real) rat is incredibly mobile and uses its legs, spine, head and tail to traverse complex environments. As shown in Fig 1 the prototype robot is approximately the size and mass of a large rat and mechanically simple using wheels for mobility. The robot's dimensions are 150mm long, 80mm wide, and 70mm high, not including the Wi-Fi antenna with a mass of 0.5kg approximating those of a real rat. Note that the cream colored body shown in the figure is designed to allow for evaluation of sensors and their locations and will be designed to incorporate aspects of the rat's body shape in subsequent development.

A real rat digests food for energy. The robot has a battery and on board recharging that allows two hours of continuous operation.

A (real) rat's eyes have poor visual acuity, high sensitivity that gives excellent performance in low light conditions, and a wide field of view. A custom solution is currently under development, designed to allow the robot to see well in low light conditions and over a wide field of view. For this study the prototype design uses a single low-cost USB webcam for the robot rat's vision sensor.

A rat has whiskers that can discriminate texture and sense proximity for close obstacle avoidance. This prototype design uses four Sharp IR range sensors arrayed at the front to give proximity information for obstacle avoidance.

A rat can integrate its self motion given by leg movement and vestibular information. The robot has encoders on the wheels which provide an estimate of the distance travelled.

A rat does all its thinking on-rat. On-robot computational capacity is given by a custom embedded controller coupled with a RoBoard mainboard with a 1GHz Vortex86DX CPU, 256MB RAM, and 4GB microSDHC card currently running Windows XP. The RoBoard has a wireless LAN connection so that it can communicate with other computers to gain access to additional computational capacity. A separate sensor and actuator interface controller handles the robot motion and reading sensors. This interface controller also has an LCD and navigation pad (similar to small portable devices) to allow user interaction.

The robot has a *distributed cognitive control architecture* (*DCCA*) that will support the testing of a range of neural models. In this context 'distributed' refers to modular, layered systems which can be implemented across physically separate computational platforms; 'cognitive' refers to neutrally-inspired or high-fidelity neural networks; and 'control' indicates that the robots operate in closed feedback systems. The DCCA is implemented using a robot software framework.

A robot server-client interface, *Player* (Gerkey, Vaughan, & Howard, 2003; Vaughan, 2008) is used as the basis for the framework. This framework allows studies in a real environment or in a virtual reality world simulation, allows pluggable modules for a variety of tasks, and connects to appropriate visualization tools. *Player* is free software that provides a client-server network interface that abstracts the robot hardware, sensors and actuators. This network interface allows for modularity and distribution of computation. *Player* has bindings for several different compiled and interpreted programming languages including: C, C++, Python, and MATLAB. The interpreted programming languages enable rapid prototyping and are commonly used by neuroscientists.

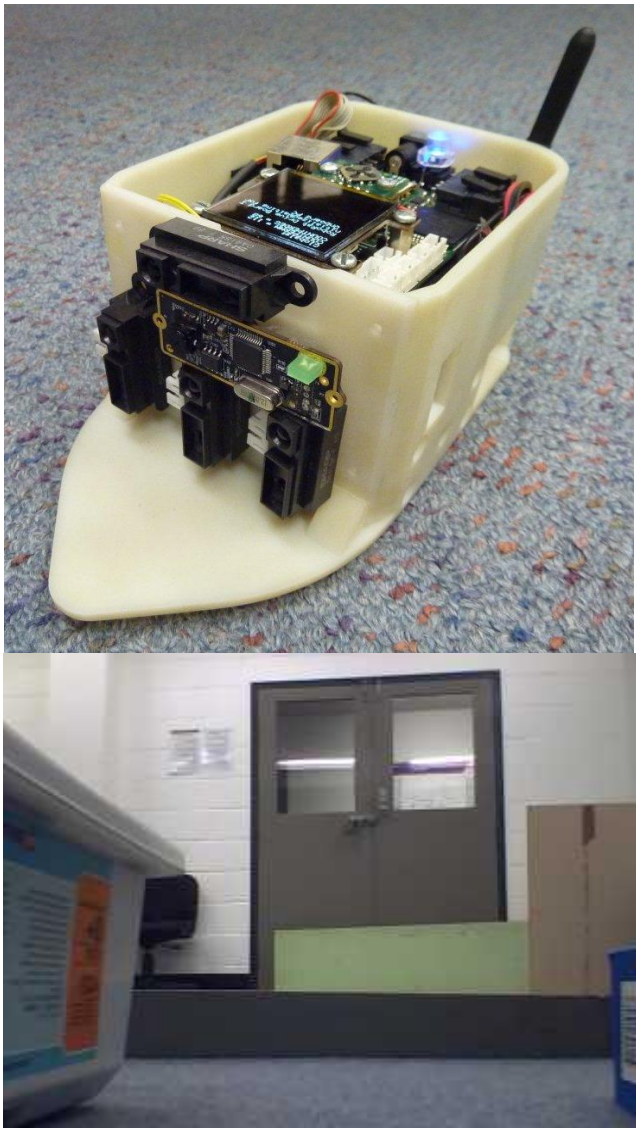


Fig 1. (top) The current state of the robot rat, showing the web camera, and four IR proximity sensors at the front, the Wi-Fi antenna 'tail' at the back and the LCD and navigation button user interface on the top. For this paper the left and right IR sensors were angled out at 45 degrees. (bottom) An image from the robot's camera sent over the wireless LAN as a 320 pixel by 240 pixel JPG image. Note the narrow field of view.

RatSLAM navigation

RatSLAM is a biologically inspired SLAM system based on models of mapping and navigation processes in the rodent hippocampus. RatSLAM contains three major modules; a vision system for appearance-based scene recognition, a neural network that represents the location and orientation state of the robot, and a graphical mapping algorithm that creates semi-metric topological maps. This section provides a brief overview of RatSLAM; a more technical system description can be found in (Milford & Wyeth, 2008, 2009).

Attractor Dynamics and Path Integration

RatSLAM represents the location and orientation state of the robot using a three-dimensional continuous attractor network (CAN). Continuous attractor networks are a popular method of modeling the spatially responsive cells found in the rodent brain (Arleo & Gerstner, 2000; Samsonovich & McNaughton, 1997; Stringer, Rolls, Trappenberg, & de Araujo, 2002; Stringer, Trappenberg, Rolls, & de Araujo, 2002). RatSLAM uses a rate-coded continuous attractor network. The network is arranged in a three-dimensional structure, where each of the three dimensions corresponds to one of the three spatial dimensions x', y', and θ' (Fig 2). Each cell is connected to nearby cells by both excitatory and inhibitory connections, which "wrap" across the opposing faces of the network structure. The connectivity is designed such that during robot navigation, the pose cell network will usually have a single cluster of highly active units, often referred to as an "activity packet" or "activity bump". The centre of this activity packet encodes the robot's location and orientation. Path integration is performed by shifting the activity in the pose cells based on self-motion information, such as wheel encoder counts. In a similar manner to the attractor dynamics, path integration can shift activity off one face of the pose cell structure, wrapping the activity around to the opposing face. Copying and shifting activity offers stable path integration performance over a wider range of movement speeds and under irregular system iteration rates, when compared with methods that shift activity through weighted connections (Arleo & Gerstner, 2000).

Local View Cells and Visual Pose Recalibration

The RatSLAM vision system learns a collection of visual templates representing what the robot sees at different locations in the environment. Each visual template is represented by a *local view cell*, which becomes active when the robot sees a visual scene similar to the template. To enable recalibration of the robot pose representation, connections are formed between co-active local view and pose cells. If the robot sees a familiar visual scene, the corresponding local view cell will activate, in turn activating the pose cells it is connected to. The activity packet will move towards the location associated with that visual scene, providing a means for correcting odometric drift and closing a loop.

Experience Mapping

The experience map is a semi-metric topological map driven by output from both the pose cells and local view cells. As a graphical map it contains representations of places, called experiences, and links between these experiences describing properties of the transition between them. Each experience is associated with a certain pose cell network state and local view cell network state, but exists in a separate co-ordinate space to the pose cell network, called experience map space. New experiences are generated when no current experiences sufficiently match the activity states in the pose and local view cell networks. A graph relaxation method distributes odometry errors throughout the map.

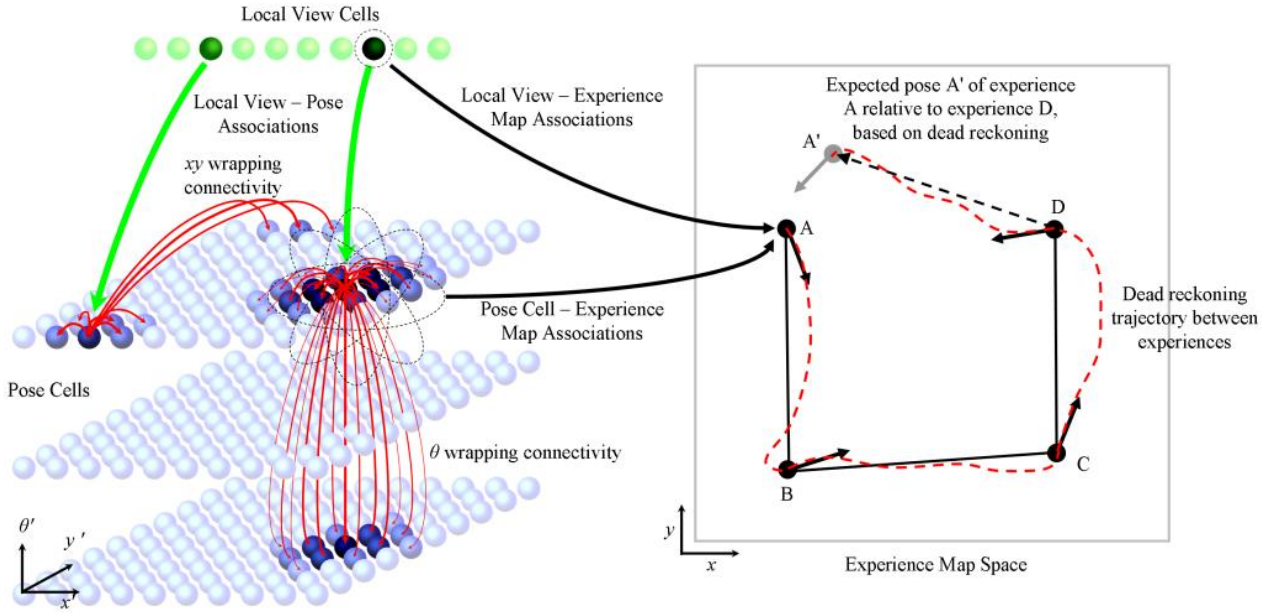


Fig 2. The RatSLAM system consists of the pose cells, which encode the robot's location and orientation state, the local view cells, which encode the robot's visual experience in the environment, and the experience map, which provides a semi-metric topological map that is used for navigation (Milford & Wyeth, 2009).

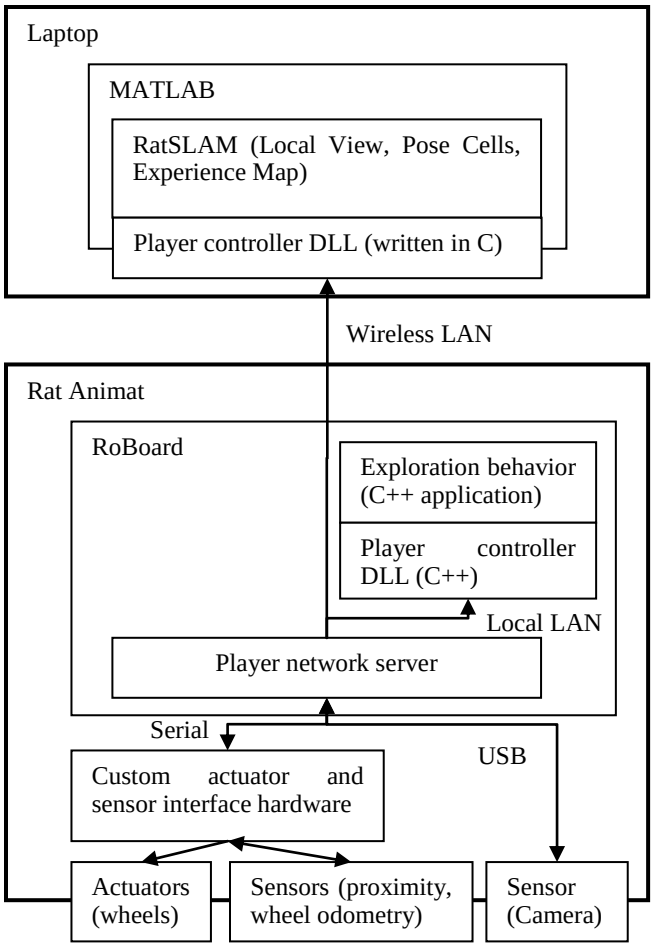


Fig. 3. This diagram shows the computational architecture demonstrating the possibilities using this rat animat and the *Player* framework. Arrows show the direction of main messages.

Experimental setup

The demonstration environment for the study was an approximately 1.5 x 1.5 meter figure of eight environment with walls of the same height as the robot, so the animat can see the rest of the lab for distal cues. The figure of eight has three loops (a large loop follows the outside wall of the arena, and two smaller loops follow the inner walls of the top and bottom halves of the figure of eight).

For this implementation of RatSLAM the view templates are histograms of column sums of the grayscale images given by the camera. New templates are compared to the stored templates using a correlation metric, with allowance for some rotation. The comparison determines whether the view is new or familiar: if new, a view template is created, and if familiar the best matching view template is determined. The bottom third of each image is typically the ground and has few distinct features appropriate for appearance based SLAM. Therefore, the robot only uses the top two thirds of the image for the view template histogram. Experiments were run for ten minutes with the robot navigating the three loops (one outer plus two inner) multiple times.

For this study the robot explored the environment using a center following behavior that attempted to maintain the same distance between the left and right wall based on readings from the IR proximity sensors. When the proximity to either wall becomes larger than a threshold then the robot would revert to either left or right wall following. These exploration behaviors were subsumed by obstacle avoidance based on the distance given by the IR sensors. For the majority of the experiment the robot travelled at 0.1 m/s. The exploration behavior ran on the robot connecting to *Player* via a local LAN connection receiving proximity distance and sending robot velocity commands at 4Hz

This study ran a MATLAB implementation of the RatSLAM navigation system on a laptop. The MATLAB version received odometry information (translational and rotational velocities) and camera images from the robot rat over wireless LAN. Fig 3 shows the experimental computational architecture. RatSLAM initially runs at 4Hz in real time but after the initial fast response, performance decreases due to the unbounded nature of the view templates and experience map in this lightweight MATLAB implementation. Because of the unbounded nature of the MATLAB version of RatSLAM, and to combine with overhead tracked images, the result figures were generated by logging the robot's camera images over Wireless LAN and then processing them offline.

Results

Fig 4 shows a comparison between the path given by the overhead tracking system, the integrated odometry path (given by the wheel velocities) and the final topological experience map given by RatSLAM. The experience map shows that the robot rat has approximately mapped the figure of eight environment. The paths show coherence within each loop, but the three loops don't completely overlap for three reasons. The first is that the centre, left and right wall following behaviors follow parallel but offset paths down the corridor resulting in different visual sequences. The second is that the centre following behavior has oscillations, particularly immediately after turning corners, which has an impact on the visual sequence. The third, and most important, is that traveling in both directions down a corridor results in different experience paths due to the forward facing camera not matching view templates. One of the primary causes is the camera's narrow field of view (approximately 50 degrees).

The experiment demonstrates the general nature of the RatSLAM system. Only minor adjustment of the visual processing algorithm was required from other applications of the RatSLAM system.

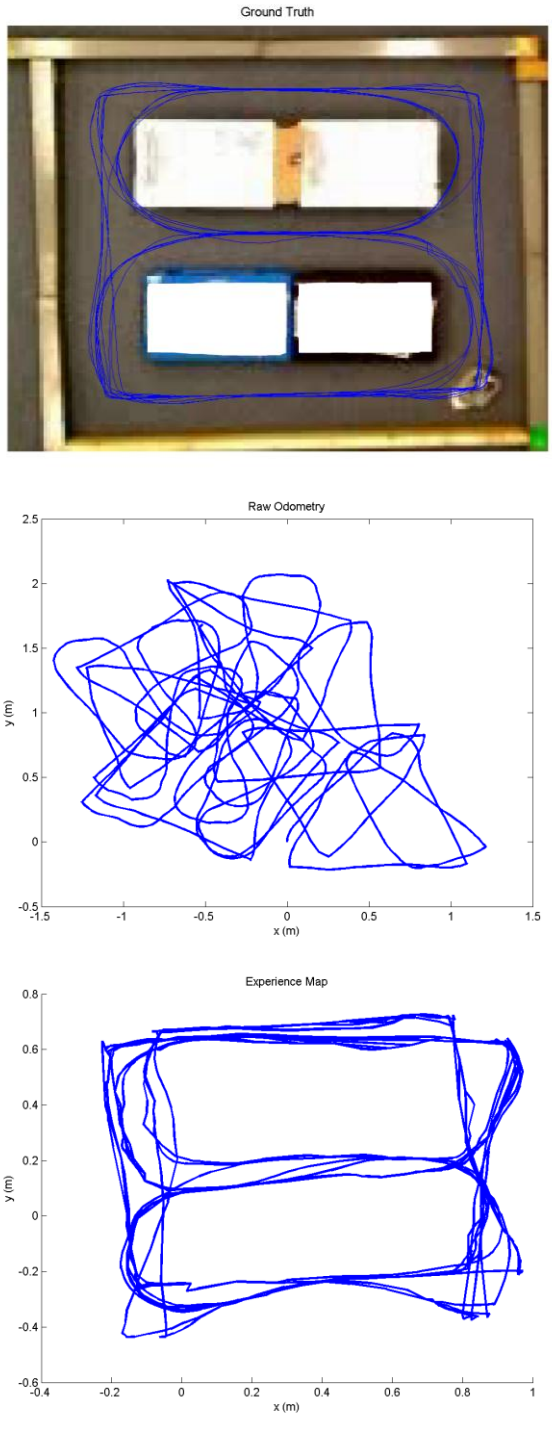


Fig. 4. (top) Path given by the overhead tracking system. The rat animat is in the bottom right corner. (middle) Raw odometry path given by integrating wheel velocities. (bottom) Semi-metric topological RatSLAM experience map that approximates the overhead tracked path.

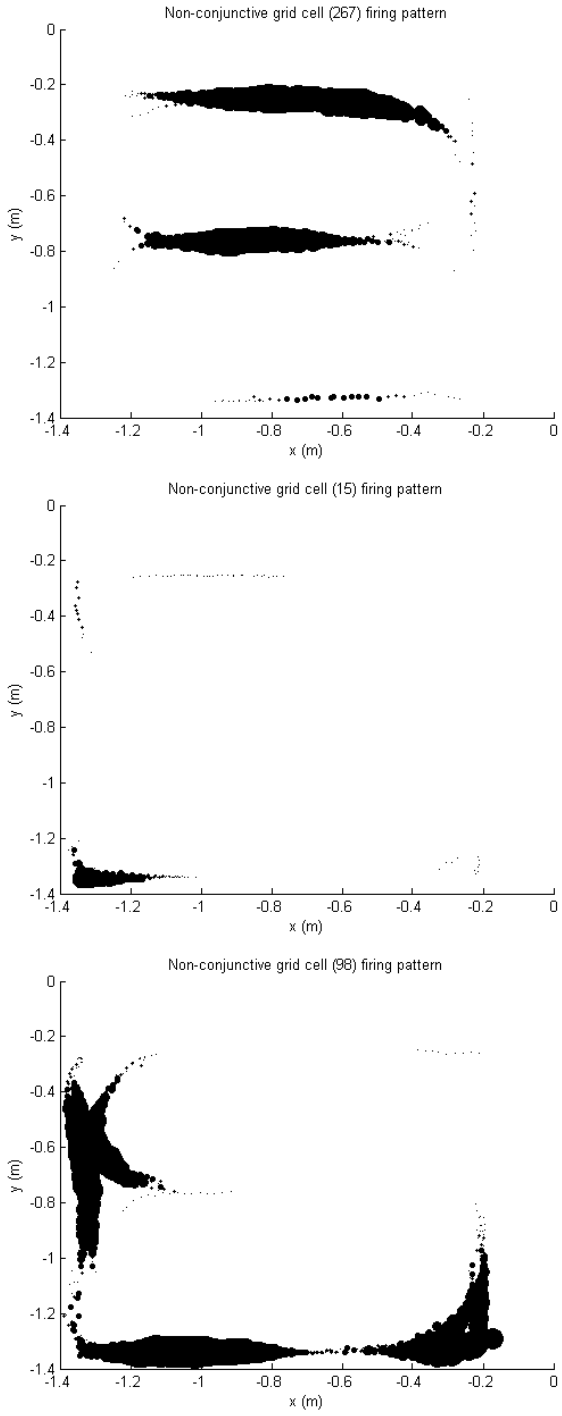


Fig. 5. Three 'non-conjunctive grid cells' as given by summing along the theta direction in the RatSLAM Pose Cell system. The size of the circle represents the level of activity. The figures show that the cells have different firing patterns. (top) The cell fires predominately in two corridors. (middle) The cell fires only in one corner of the environment. (bottom) The cell fires strongly in multiple locations in the environments.

Grid Cells

One of the original inspirations for the RatSLAM design was the rodent hippocampus. By plotting activity in an internal network of the distributed cognitive control architecture versus the position of the rat animat, it is possible to gain a firing field similar to 'non-conjunctive grid cells' prevalent in the rodent research field. These cells give a regular non-directional firing pattern. The equivalent of the 'non-conjunctive grid cells' is created by summing the activity of the RatSLAM pose cells along the θ' dimension, and plotting their average activity levels against the robot's overhead tracked location. Fig 5 shows the firing fields for three 'non-conjuctive grid cells'. The fields show that the cells fire in different locations and with different spatial properties. Some cells fire only in one part of the environment, whereas others fire across multiple sections. Note that the more typical regular firing pattern is not demonstrated in these plots because of the relatively small size of the environment compared to the pose cell network.

Discussion and Conclusions

This paper has described a new rodent animat platform similar in size to a large rat, which is capable of exploring and mapping an environment with multiple loops in real time. On board capabilities include visual, proximity, and odometry sensors, wheeled actuation and on-robot PC equivalent computation. The rat animat's distributed cognitive control architecture is not limited by on-robot computational resources as the *Player* framework allows for transparent communication over wireless LAN. The results demonstrate the rat animat's and *Player's* possibilities with using C/C++ and MATLAB in real time behaviors and SLAM distributed across the robot and other computers. This is significant as it will open up the platform to a broader range of researchers.

The paper began by highlighting the importance of embodiment with regard to the size of the real animal and the corresponding constraints on capabilities. This study has demonstrated that computational resources equivalent to a PC are now possible on a rat sized robot as well as real time connection to off-robot computation. The RatSLAM algorithm has shown itself to be remarkably generic, as it was ported from the pioneer robot to the robot rat with minimal adjustments. The order of magnitude change in camera height from the Pioneer robot to the rat animat does give a different perspective on the environment although this did not require any changes to the visual template matching technique. Changing from an omni-directional visual sensor to the forward facing small field of view sensor has had the most dramatic effect on the system performance as shown by the experience map connectivity. The experience map would benefit from using a visual sensor with a field of view similar to a real rat.

There are many avenues for future work. To allow longer experiments and users to interact with the robot via the web over the long term, the platform will need to be able to autonomously recharge with a docking station. Whiskers are important sensors for rodents that allow them to wall follow, detect obstacles and discriminate textures. Work has begun on developing a whisker system for this platform with these capabilities. On the neural controller side, the SLAM system needs to be integrated with a behavior system at a minimum capable of goal directed navigation and exploration. RatSLAM will also benefit from an improved visual perception system (hardware and neural controller) to improve performance. Other work will extend the behaviors for survival, social interactions and language games.

Acknowledgements

This work was supported in part by the Australian Research Council Special Research Initiative TS0669699, "Thinking Systems: Navigating through Real and Conceptual Spaces."

References

Arleo, A., & Gerstner, W. (2000). Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity. *Biological Cybernetics, 83*, 287–299.

Barrera, A., & Weitzenfeld, A. (2008). Computational modeling of spatial cognition in rats and robotic experimentation: Goal-oriented navigation and place recognition in multiple directions. *Biomedical Robotics and Biomechatronics (BioRob2008)*, 789-794.

Beer, R. D. (2008). The Dynamics of Brain–Body–Environment Systems: A Status Report. In P. Calvo & T. Gomila (Eds.), *Handbook of cognitive science: an embodied approach*.

Beer, R. D., & Williams, P. L. (2009). Animals and Animats: Why Not Both Iguanas? *Adaptive Behavior, 17*(4).

Brooks, R. (1991). Intelligence without representation. *Artificial Intelligence*.

Burgess, N., Donnett, J. G., Jeffery, K. J., & O'Keefe, J. (1997). Robotic and neuronal simulation of the hippocampus and rat navigation. *Philosophical Transactions of the Royal Society of London B Biological Sciences, 352*, 1535-1543.

Doya, K., & Uchibe, E. (2005). The Cyber Rodent Project: Exploration of Adaptive Mechanisms for Self-Preservation and Self-Reproduction. *Adaptive Behaviour, 13*.

Fend, M. (2004). *The Artificial Mouse - A robot with whiskers and vision.* Paper presented at the 35th International Symposium on Robotics, Paris.

Fend, M., Bovet, S., & Pfeifer, R. (2006). On the influence of morphology of tactile sensors for behaviour and control. *Robotics and Autonomous System, 54*(8), 686-695.

Fox, C. W., Mitchinson, B., Pearson, M. J., Pipe, A. G., & Prescott, T. J. (2009). Contact type dependency of texture classification in a whiskered mobile robot. *Autonomous Robots, 26*, 223-239.

Gerkey, B. P., Vaughan, R. T., & Howard, A. (2003). *The Player/Stage Project: Tools for Multi-Robot and*

Distributed Sensor Systems. Paper presented at the International Conference on Advanced Robotics (ICAR 2003), Coimbra, Portugal.

Hafting, T., Fyhn, M., Molden, S., Moser, M., & Moser, E. I. (2005). Microstructure of a spatial map in the entorhinal cortex. *Nature, 436*, 801-806.

Ishii, H., Ogura, M., Kurisu, S., & Komura, A. (2006). *Experimental Study on Task Teaching to Real Rats Through Interaction with a Robotic Rat* Paper presented at the 9th International Conference on Simulation of Adaptive Behavior, SAB 2006.

Lacheze, L., Benosman, R., & Meyer, J.-A. (2008). *Integration of an Omnidirectional Visual System with the Control Architecture of Psikharpax* Paper presented at the 10th International Conference on Simulation of Adaptive Behavior, Osaka, Japan.

Mataric, M. J. (1991). Navigating with a rat brain: A neurobiologically-inspired model for robot spatial representation. *From Animals to Animats, MIT Press, Cambridge, MA*.

Meyera, J.-A., Guillota, A., Girarda, B., Khamassia, M., Pirimb, P., & Berthozc, A. (2005). The Psikharpax project: towards building an artificial rat. *Robotics and Autonomous System, 50*(4), 211-223.

Milford, M., & Wyeth, G. (2008). Mapping a Suburb with a Single Camera using a Biologically Inspired SLAM System. *IEEE Transactions of Robotics (Special Issue on Visual SLAM), 24*(5).

Milford, M., & Wyeth, G. (2009). Persistent Navigation and Mapping using a Biologically Inspired SLAM System. *International Journal of Robotics Research*.

O'Keefe, J., & Conway, D. H. (1978). Hippocampal place units in the freely moving rat: why they fire where they fire. *Experimental Brain Research, 31*(4), 573-590.

Patanè, F., Mattoli, V., Laschi, C., Mazzolai, B., Dario, P., Ishii, H., et al. (2007). *Biomechatronic Design and Development of a Legged Rat Robot*. Paper presented at the 2007 IEEE International Conference on Robotics and Biomimetics.

Pearson, M. J., Pipe, A. G., Melhuish, C., Mitchinson, B., & Prescott, T. J. (2007). Whiskerbot: A Robotic Active Touch System Modeled on the Rat Whisker Sensory System. *Adaptive Behavior, 15*(223).

Ranck Jr, J. B. (1984). Head-direction cells in the deep cell layers of dorsal presubiculum in freely moving rats. *Society Neuroscience Abstracts, 10*.

Samsonovich, A., & McNaughton, B. L. (1997). Path Integration and Cognitive Mapping in a Continuous Attractor Neural Network Model. *The Journal of Neuroscience, 17*(15), 5900-5920.

Stringer, S. M., Rolls, E. T., Trappenberg, T. P., & de Araujo, I. E. T. (2002). Self-organizing continuous attractor networks and path integration: two-dimensional models of place cells. *Network: Computation in Neural Systems, 13*(4), 429-446.

Stringer, S. M., Trappenberg, T. P., Rolls, E. T., & de Araujo, I. E. T. (2002). Self-organizing continuous attractor networks and path integration: one-dimensional models of head direction cells. *Network: Computation in Neural Systems, 13*(2), 217-242.

Takanishi, A., Aoki, T., Ito, M., Ohkawa, Y., & Yamaguchi, J. (1998). *Interaction between creature and robot: development of an experiment system for rat and rat robot interaction*. Paper presented at the IEEE Intelligent Robots and Systems.

Vaughan, R. (2008). Massively multi-robot simulation in stage. *Swarm Intelligence, 2*.

Webb, B. (2000). What does robotics offer animal behaviour? *Animal Behaviour, 60*, 545-558.

Webb, B. (2001). Can robots make good models of biological behaviour? *Behavioral and Brain Sciences, 24*(6), 1033-1050.

Emotion in Decisions of Life and Death - Its Role in Brain-Body-Environment Interactions for Predator and Prey

Claire O'Bryne and Lola Cañamero
Adaptive Systems Research Group, University of Hertfordshire
College Lane, Hatfield, Herts AL10 9AB, UK
C.S.O-Bryne@herts.ac.uk, L.Canamero@herts.ac.uk

Abstract

Taking inspiration from the biological world, in our work we are attempting to create and examine artificial predator-prey relationships using two LEGO robots. We do so to explore the possible adaptive value of emotion-like states for action selection in this context. However, we also aim to study and consider these concepts together at different levels of abstraction. For example, in terms of individual agents' brain-body-environment interactions, as well as the (emergent) predator-prey relationships resulting from these. Here, we discuss some of the background concepts and motivations driving the design of our implementation and experiments. First, we explain why we think the predator-prey relationship is so interesting. Narrowing our focus to emotion-based architectures, this is followed by a review of existing literature, comparing different types and highlighting the novel aspects of our own. We conclude with our proposed contributions to the literature and thus, ultimately, the design and creation of artificial life.

Introduction

In our work we are, broadly speaking, interested in seeing what existing ideas about emotion in biological agents can do for the creation of more adaptive artificial agents. Concentrating on ideas about the role of emotion in rational decision-making, we are especially concerned with how such ideas might help us address the problem of action selection using emotion-based architectures. Action selection referring to the problem all agents (biological and artificial) must necessarily face of "what to do next" [Bryson (2007)], we are further interested in (and advocate) studying it within the context of (biological and artificial) predator-prey relationships.

By focusing on this type of relationship, besides enabling us to better explore and develop our ideas about the role of emotion for adaptive behaviour in dynamic environments, we suggest it allows us to obtain more detailed insights due to and regarding specific aspects or characteristics of this type of environment. This includes those requiring some kind of appropriate risk assessment (such as perception of danger) and, in turn, risk-taking. Consequently, one of our main aims is to consider in greater depth how action selection mechanisms might be developed so as to be adaptive in such situations. That is, where an agent's decisions are literally those of "life and death".

Considering relatively recent ideas about the importance of the body for intelligent and adaptive behaviour [Pfeifer and Scheier (1999); Pfeifer and Bongard (2006)], we explore the link between action selection and emotion in terms of brain-body-environment interactions. Asking whether we should stop focusing so much on abstracting away features of body, in favour of developing emotion-based architectures oriented more towards ideas such as those inherent to the notions of internal robotics [Parisi (2004)] and morphological computation i.e. those explicitly giving agent body a more proactive role in the generation of behaviour.

To do this, and because we are interested in identifying factors (particularly those relating to the concepts of embodiment and embeddedness) that might affect such interactions, we have developed robots that both model and provide a means for studying the (different types of) relationship between a single predator and prey agent. Specifically, we use an implementation of a predator-prey type scenario previously developed to study action selection: the Hazardous Three Resource Problem (H3RP) [Avila-García and Cañamero (2005)].

Here though, we set aside the more technical details of our experiments and implementation. Firstly, for a more general consideration and outline of our ideas as to why the predator-prey relationship is so interesting and relevant to the problem of action selection (also detailing our main research interests and questions). Secondly, to review the literature so as to compare more general features of our work, robots and implemented emotion-based architecture with those of other researchers. And finally, to detail the ways in which we hope our work will make its own contribution to the existing literature, for both the problem of action selection and role of emotion in adaptive systems.

The Predator-Prey Relationship and Problem of Action Selection in the Literature

The relationship between predator and prey is one that should be of particular interest to those studying action se-

lection. Indeed, it is of interest across and within many disciplines. While there are many aspects of this scenario to interest researchers, what often stands out is the fact it is a relationship between two agents. Moreover, it is a relationship characterised by a dependency of one agent (the predator) on another (the prey) for its continued survival. This results in interactions between agents that will determine the success of each agent, with a push-pull effect. Where one wins, the other will likely suffer some corresponding cost or loss.

Looking at the literature, research has explored this scenario from various perspectives. From the level of the individual over a lifetime [Kelly et al. (1999)] to populations across generations [Nolfi and Floreano (1998); Buason et al. (2005)]. Yet the way this relationship has most often been studied is through the development of action selection mechanisms for the prey that will result in it fleeing whenever it sees a predator. In effect, making this the more or less automatically optimal or decided choice of action, regardless of the task currently being performed.

Strangely, researchers have also commonly continued to focus on one type of agent only (predator or prey) with the action selection problem of the other agent being of secondary to no interest. We regard this as possibly leading to a more superficial look at, or treatment of, the action selection problem for artificial predators and prey. A perspective which may lead to less rich, or realistic, solutions than might be the case or useful in real life and real time.

For example, this emphasis does not take into account or allow for the possibility that in fact there may be times in which the more adaptive behaviour would be for the prey to "take the risk" of being attacked by its predator. Or, indeed, the case that there are some, if not many, environments in which life must constantly be risked in order to achieve long-term survival. Perhaps in favour of satisfying some other survival need or task. Looking towards ethological studies for evidence and inspiration, researchers illustrate this could also be true of biological organisms.

For instance, Cooper Jr (2000) found a species of lizard will tolerate predators to come closer before they decide to "flee" under certain conditions, including when they were eating food. Though it could be argued this might also reflect the possibility that the lizard's attention is more directed on feeding than awareness of or perception of the predator. More interestingly, it could be that some kind of economic model allows for "risk-taking" or a kind of "cost-benefit" analysis in terms of risk assessment that is adaptive for agents. Then too, this could lead to a role for emotion-like states as quick, real-time assessors of risk in relation to certain stimuli.

Our Interest in the Predator-Prey Relationship

The predator-prey relationship may be of interest for action selection researchers for many other reasons. However, for us, among the most interesting are:

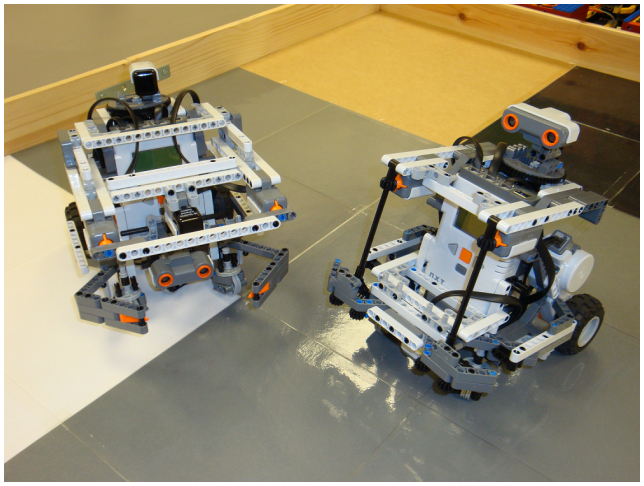


Figure 1: Our Implementation: Predator (left) and Prey (right) robots developed for early experiments [O'Bryne et al. (2009)]. These agents have been built using two LEGO NXTs. Our initial experiments have focused on developing different "brains" for our agents (emotion-based architectures); looking at the results in terms of adaptive value (production of adaptive behaviour) in different "bodies" and "environments" (by connecting architectures to the environment in different ways, such as using different physical sensors; and varying properties of the partner robot i.e. predator or prey agent)

- Adding a predator (or prey) to a given agent's environment is a way of making that environment dynamic. It leads to to changes over time that the agent must respond to adaptively and often increases environmental complexity. Thus, in terms of action selection, it can act as a good test for how well an individual agent (or the action selection mechanism implemented within it) can cope with increases in the dynamics of their environment. Importantly, the typical nature of these are usually such that each agent has to make quick decisions in order to make adaptive ones. This leads to a trade-off, where if the agent hesitates or ponders too long, all could be lost anyway (game over, especially for the prey).

- It allows us to study action selection at a higher or more general level, within the context of two agents in a very unique relationship. Typically, one in which, where one agent wins, the other will invariably lose. This may affect the demands for (and guide the design of) the agents and action selection mechanisms themselves, especially as the relationship is characterised by a dependency of one on the other i.e. predator is dependent on prey. Admittedly, prey might also be said to be dependent on predator. For instance, at the population level, to avoid over-population. Yet such dependency is likely to be much more indirect. This thereby makes the balance of opportunity cost and stakes for each agent in any interaction unequal. Where predator loses a meal, prey loses its life.

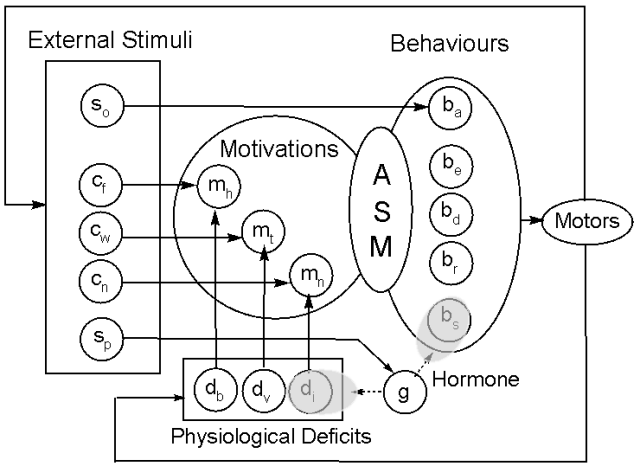


Figure 2: Overview of our developed architecture ("brain") for a prey agent: internal "body" is represented through physiological variables, deficits of which act as drives which, combined with the presence/absence of external stimuli, are used to calculate motivational and behavioural intensity. For example, calculations of motivational intensity for a motivation representing hunger will take into account both physiological deficits such as blood sugar and the presence/absence of food in the environment. In our experiments we vary external "body" using different physical sensors. Emotion-like states are modelled by the addition of a gland (g); releasing a "hormone" in the presence of a specific stimulus (in this case the predator) which affects both perception of internal physiological deficits, increasing calculations of motivational intensity, and the behaviour selected in terms of physical response (speed or tempo of behaviour is increased if hormone is present)

- It provides us with (if nothing else a wealth of biological) inspiration for building action selection mechanisms both a) capable of dealing with situations of high and immediate risk (used by prey) and b) capable of adapting to another agent's behaviour (environmental dynamics) for the agent's own advantage (used by prey and predator). It is also a problem that may call for compromises, increasingly specialised or more adaptive behaviours and, more specifically for us, interesting trade-offs. Namely, between the basic choices for the prey of whether it should flee or not, and for the predator of whether it should attack/hunt or not. Somehow, these agents must be able to effectively weigh up and make these decisions in the limited time available.

- It allows us to focus on the interactions that result between (the action selection mechanisms of) two agents with different sensory abilities, brains, bodies, motivations, possibly emotions (especially at the time of interaction) and behavioural repertoires. Starting our own "arms race" between such agents, we can develop and fine-tune features of these agents to enable one to gain an advantage over the other. This could not only produce and drive the production of increasingly more adaptive agents, but also lead to a better understanding of the (different types of) predator-prey relationship(s), as well as the circumstances when certain components of action selection mechanisms might be most adaptive.

- It allows us to look in more detail at the requirements for adaptive behaviour in this context. For example, it allows us to ask whether a predator needs more "brain power" than its prey in order to be able to catch it, or simply different types of behaviours and abilities. Similarly, it allows us to explore those ways in which we might increase or examine the adaptive value of predator and prey action selection mechanisms. This could include the use of methods across disciplines. For instance, we might analyse developed prey agents' behaviour in a similar way to Cooper's lizards: in terms of the assessments of risk or cost-benefit analyses that he suggests can be used to explain their behaviour.

Our Research

Driven by these interests, we have been using our robotic predator and prey to develop and explore the adaptive value of emotion for emotion-based architectures (see Figures 1 and 2). Both to gain insights as well as explore (test) links between concepts of emotion, action selection, adaptive value, dynamic environments, the brain-body-environment and predator-prey relationship. Adopting a bottom-up approach, we introduce emotion-like states using a mechanism that simulates the effects of neuromodulation (albeit at a more abstract level than that of the neuron). What is particularly attractive about this mechanism is it can be used as secondary controller to an existing architecture.

Broadly, we look to see under what conditions our

emotion-based architectures (especially those implementing our chosen mechanism) prove adaptive for agents. We believe a systematic study, in the context of the H3RP, will increase our understanding of the adaptive value and potential of this mechanism. Not only in terms of action selection, but in terms of predator-prey scenarios. Our mechanism was chosen primarily because neuromodulation has previously been noted as a possible "substrate of emotion". And it is within this general framework that we formulate our more concrete experimental research question(s).

Experimentally, this has led to an attempt to identify factors affecting the adaptive value of the mechanism simulating neuromodulation. Both as a proposed substrate of emotion and biasor of action selection, in the predator-prey scenario. However, we are interested not only in what this will tell us about the possible adaptive value of emotion, but also its likely link to and dependence on properties of a given body and environment (implementation or task[s]) [O'Bryne et al. (2009)].

More specifically, we ask how changes in the physical (e.g. sensory-perceptual and motor-behavioural) abilities of predator and prey agents interact to affect the balance or dynamics of their relationship. The abilities we aim to study have primarily included the distance into the agent's environment information about stimuli can be obtained. We are not only interested in such relationships in terms of the advantage of one over the other in given encounters i.e. who "wins", but more importantly the behavioural interactions and adaptive value of the mechanism simulating neuromodulation.

In the context of brain-body-environment interactions [Chiel and Beer (1997)] we think such questions are interesting. Not only are we explicitly exploring the importance of certain specific aspects of body in producing adaptive behaviour. But we are also considering their importance for the successful integration of emotion and emergence of specific, adaptive behaviours within a predator-prey situation. Looking not only at what kind of role emotion might play with regards to brain-body-environment interactions, but also how the presence of another agent (prey or predator) might concurrently affect and direct this relationship or interactions.

To put this another way, we ask what will happen to the dynamics of a predator-prey relationship when sensory capabilities, including perceptual distances, are varied. We want to know what will happen in terms of physical and behavioural advantage, as well as the consequent adaptive value, of a mechanism simulating neuromodulation (as a biasor of action selection).

A Comparison with other Emotion-Based Architectures

To give an idea of where we place our work and architectures in relation to that of others, as well as to give an overview of related literature, it might be useful to conduct a quick comparison of different types of emotion-based architectures. Specifically, those which have also been implemented in robots. Here we look to do so in order to effectively, albeit briefly, contrast our work with that of three other researchers: Breazeal, Arkin and Avila-García.

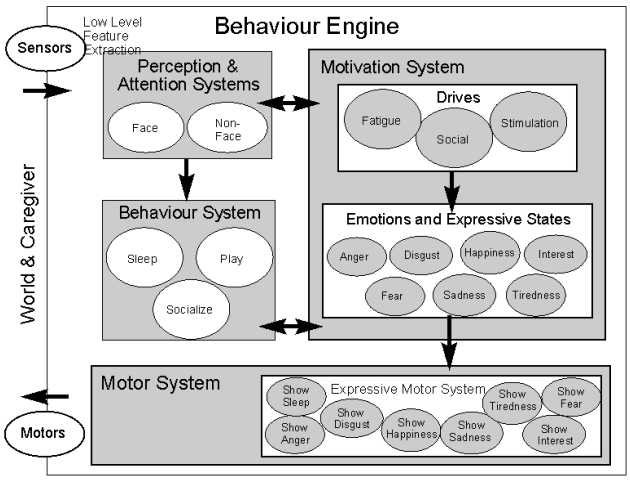


Figure 3: Illustration and overview of Breazeal's architecture for Kismet: Incorporating ideas about different types of emotions and connecting them to different motor responses (emotional expressions) [Breazeal and Scassellati (2000)]

We chose each of these researchers and their architectures for different reasons: Breazeal [Breazeal and Scassellati (2000)] provides us with a "classic" architecture for comparison, Arkin [Moshkina et al. (2009)] with a relatively recent addition (TAME being the "state of the art" in the history of his work) and Avila-García's work [Avila-García (2004)] is in many ways closest to our own. Such similarity makes it important for us to identify the ways in which our approach and architectures differ.

So as to get more of an overview of the differences between them, we will look at these researchers' research in reasonably broad terms, using some simple criteria. We do so here in the context of how each of these researchers treat or incorporate ideas about emotion in their architectures; what their primary motivations are, including the problem or domain of interest studied; and what they consider adaptive action selection to be (i.e. their measures of adaptive value).

Function and Integration of Emotion

Illustrations of the types of architecture produced by each researcher, including our own, are produced in Figures 2-5. First, we should look at how each one sees "emotion" in this context i.e. their ideas as to the function and integration of emotion for action selection mechanisms. As can be seen from Figure 3, Breazeal's architecture explicitly introduces emotions as a subset of motivations. Ideas about the function of emotion as being communicative are incorpo-

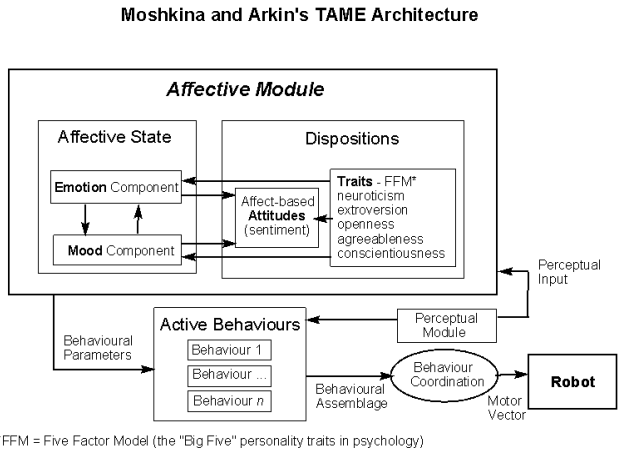


Figure 4: Illustration and overview of Moshkina and Arkin's TAME Architecture: Incorporating ideas about and explicitly modelling personality and emotion using concepts connecting Traits, Attitudes, Moods and Emotions - each of these varying in their temporal effects and influence on each other [Moshkina et al. (2009)]

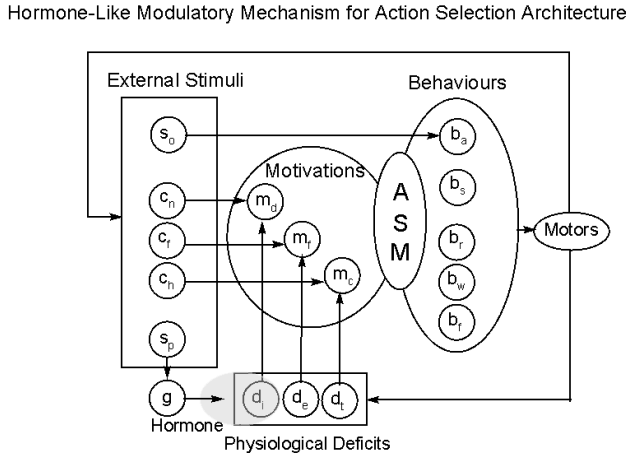


Figure 5: Illustration and overview of Avila-García's hormone-like modulation of an action selection architecture: Emotion-like states are modelled by the addition of a gland (g); releasing a "hormone" in the presence of a specific stimulus (in the case of his predator-prey scenario, the H3RP, the predator) which affects both perception of internal physiological deficits, increasing calculations of motivational intensity: concentration decays over time [Avila-García (2004)]

rated through the modelling of emotional expressions (the "actions" selected by her implemented robot Kismet) and internal "emotions" are used to activate a robot's physical expression at any given time.

Contrastingly, from Figure 4, we see that lately Arkin has been contributing towards the development of a different kind of architecture. The TAME architecture introduces and incorporates emotions in a more "sophisticated" model, where emotion is treated as one of a number of affective phenomena to be explicitly modelled (traits, attitudes, moods and emotions). Similarly to Kismet, the robots (AIBO and Nao) in which TAME has been implemented have used emotion in a communicative context. This is in contrast to some of his earlier architectures, looking "up the food chain", which were generally based on the ideas of his earliest architecture (AuRA) and also looked at other possible functions of emotion (non-communicative) for individual, autonomous agents.

With more relevance for our own work, Figure 5 presents one of Avila-García's architectures. This is where we most closely align ourselves with regards to the function and integration of emotion. This is because, in his architecture, Avila-García does not explicitly label any one component as "emotion" (something we also advocate). Instead, we both prefer a more bottom-up approach: trying to model one of the suggested neural "substrates of emotion". Namely, neuromodulation [Fellous (1999)]. We do this in order to examine the emergent properties of a system, which may consequently resemble the "emotion-like" behaviours of real-life adaptive agents.

Thus, we have both attempted to simulate the effects of neuromodulation for the benefit (adaptively) of action selection mechanisms. In addition, at a level of abstraction which has resulted in the development of hormone-like mechanisms ("hormone-release" occurring in the presence of relevant external stimuli) which affect action selection over time. In particular, Avila-García examined different ways in which such a mechanism can act as a biasor of action selection, modulator of perception (both interoception and exteroception) and "second-order controller" for existing architectures (in this case a motivation-based one).

However, one way in which our currently developed architecture differs, is that we try to integrate this kind of mechanism more pervasively or intricately with the rest of our architecture. As Figure 2 shows, we have linked our hormone-like mechanism not only to calculations of motivational intensity, but also the intensity of behavioural response. To give an example, in recent experiments, this has translated into an implementation of a prey agent that, when its "hormone level" increases, so too does its physical speed. Thus, we use this "substrate" not only to modulate perception, but to influence behaviour more dynamically and physically, in terms of factors such as time.

We think this has the advantage of effectively making

"short-cuts" or more direct links between a perceived external stimulus and physical response or readiness of action, which may especially help in the problem of allocation of limited "energy" resources. Moreover, we go further to consider the interactions between two agents (and their architectures) rather than looking at one individually (though this is not explicitly illustrated in Figure 2).

Problem or Domain of Interest

Next, we would like to turn to and compare the particular areas or "problems" that these architectures, or their implementations, have been designed to study or solve. We attempt to do so here with regards to each researcher's particular contribution to the study of action selection, reflected in the implementations each researcher has developed, as well as the particular context (environment/scenario/task) it has looked at the role of emotion or emotion-like states in. In this way, we can also examine some of the features of action selection that each focuses on.

Whilst each architecture can itself be considered a contribution to the action selection literature, and all have been implemented in robots which is especially appealing, they have each been implemented for quite different purposes and in quite different environments: Kismet to model social interactions between infant and caregiver (thus human-robot interactions); Arkin's TAME to model affect more sophisticatedly for human-robot interaction; Avila-García's to test the properties of architectures across different types of environment/scenarios (only one of which includes a predator-prey type scenario); and ours to study action selection within a very particular context and relationship (predator-prey) in order to examine brain-body-environment interactions.

First, in more general terms, we can say that the primary implementations of both Breazeal's and Arkin's architectures have been in the area and interests of human-robot interaction. The robot head Kismet is Breazeal's result and TAME has been implemented in both Sony's AIBO dog and the humanoid Nao. While this is of course an extremely relevant and interesting area for the study of the role of emotion (particularly with regards to communicative functions and interactions) what sets such architectures apart for us is that they are designed to say as much, if not more, about our own emotions and interpretation of other agents' (robots) behaviours. That is to say, they may reveal more about us and less about the adaptive value of emotion for the robot.

We regard this as bringing a dimension to their work that we currently prefer to leave out of our own, in favour of focusing our study more exclusively on artificial agents. One of the advantages of a synthetic approach is that we can study the interactions resulting between two agents we already know the exact internal workings of. Introducing a human participant negates this as we do not know the exact workings of such an agent. Thus, we are less concerned with their impact on our own (human) behaviours and perceptions of them as agents (though of course we may always inadvertently introduce our own bias as researchers if we are not careful in how we study them).

Avila-García similarly goes a different way to Breazeal and Arkin. He implements his architectures across different scenarios, also using LEGO robots (Taurus and Sador being examples of these). However, he focuses instead on developing ways to quantitatively and qualitatively measure these implementations as individual adaptive systems, to identify their specific properties in different contexts. He considers other agents solely with regards to how they may add to the environmental dynamics, and possibly environmental complexity (rather than as an agent in a partnership or some kind of artificial ecology, which can affect and be affected by other agents).

By not focusing on one particular problem, Avila-García was able to look at the properties of architectures, in particular arbitration mechanisms, across different scenarios. He developed several types of scenario for the study of action selection, including a robotic two-resource problem; competitive two-resource problem; and hazardous three-resource problem (H3RP). Yet, even in his predator-prey type scenario (the H3RP) action selection did not involve situations of such high risk as might be expected of such a relationship. This was due to his development of a more "parasitic" type of predator-prey relationship (allowing his agents some leeway in choosing to change activity).

This does not mean that we do not want to, or do not aim to contribute towards developing ideas that may also be of use to these other domains of interest. More, we think by focusing on our particular scenario now (that of predator-prey) we will be able to bring something particularly special or unique to the problems of these other architectures later. Currently, for instance, all three of these other architectures, when you consider the implementations, do not seem capable of producing adaptive behaviour in situations where both the two-way relationship between two agents is accounted for, and the right decision or action selection is vital for agent survival i.e. studying both agents in high risk situations.

What is primarily different about our own motivation then, is with regards to the kinds of decision and environmental demands we want our architecture to deal with. This includes situations where there may not be enough time or flexibility to allow for mistakes or trial-and-error learning; instead requiring split-second judgements. More to the point, we want to study the predator-prey scenario for a much more in-depth look at this kind of relationship, where a predator is not just an environmental dynamic.

For example, if a robot were to identify another agent as a predator, we would like to see our robot's emotion-based architecture capable of using its "fear" to better make those split-second decisions that will direct action selection towards the agent's own survival. This could involve some

means of "fleeing" the scene, but might even involve our prey robot staying to "brave" it out or "defend" its position or resources. More, we also want the robot predator to be able to adapt to such behaviour, somehow weighing up the situation in the limited time available to better direct action selection.

Finally, another difference can be seen in the type of intelligence or adaptive behaviour studied. For example, Breazeal and Arkin can be said to study action selection and emotion more focused on ideas of human-level intelligence and emotions (though Arkin has in fact previously developed ones he suggests demonstrate a lower, more insect-like intelligence). Once more in common with Avila-García, in contrast we attempt to create simpler creatures for study. For example, considering these concepts more in terms of animal-like mechanisms of adaptive behaviour and intelligence.

While Arkin has previously studied architectures aiming towards insect-level intelligence (incorporating and developing ideas about motivation and emotion), in "moving up the food chain" [Arkin (2005)] it does appear he left a somewhat expansive gap between the level of insect and that of animals. Using our bottom-up approach, this is where we would like our work to fit. Between the reactive architecture he attributes to an insect; and the more deliberative architectures he chooses for those interacting with humans.

Measures (of Adaptive Value)

Finally, we can also compare researchers in terms of the level of analysis and criteria each expects will be used to measure the adaptive value of their architectures in a given implementation. Without going into unnecessary detail, perhaps due to their interest in human-robot interaction, in this respect both Breazeal and Arkin can be said to have focused on the use of both internally and externally-derived measures i.e. measuring, for different purposes, both external effects of their robots' action selection on human response; and the internal parameters of the system or architecture over time. When involving observations, this is often a lengthy process with regards to analysis, but has the benefit of allowing us to directly study interactions between humans and robots.

Conversely, Avila-García's architectures were studied placing focus mainly on the use of more internally-derived and summarative measures. He developed measures of analysis that consider the viability of agents over an individual life span (presumably choosing this as the correct level of analysis to study adaptive value). But, just as interestingly, Avila-García also considered and suggested action selection be studied in terms of activity cycles rather than separate decisions. Similarly, we would like to consider how analysis of behaviour over time might bring us more insights into our architecture's behaviour in different predator-prey scenarios.

In our work though, perhaps more in common with Breazeal and Arkin, we try to combine the use of both externally *and* internally-derived measures for studying the performance of our agents. We also attempt to go further, for a more comparative look. One of our primary concerns is thus to ask at what level of study we will find out most or understand our systems best. Especially with regards to what one might consider adaptive value to be (and in terms of brain-body-environment interactions). In this way we again seek to bridge the gap between these architectures, this time in respect of the level their researchers have proposed we analyse them at.

One source of inspiration for us in this endeavour again comes from another discipline: ethology. Though dynamic systems theory has developed tools to study the interactions of dynamic systems, we use the analogy of animal-like behaviour to suggest that the ethologists have already developed many tools to be used in the analysis of our animat agents. In particular, many of these methods allow us to combine both considerations of internal and external data (as derived or collected from experiments).

Contributions

Having considered our own research using such criteria, the contributions we therefore hope our work will make, especially towards the literature on action selection and emotion (or affect) include:

For "Affective" Action Selection:

- Further development of our architectures and implementation. In initial experiments, we divided perception into proximal and distal types (combinations of which making further sub-problems or versions of the H3RP). This enables and hopefully justifies direct comparison, especially in terms of the interactions of different physical properties of predator and prey, with previous findings using the same framework (such as Avila-García's). At the same time, it introduces a new dimension for study (an aspect of embodiment, in this case perceptual field or "sensory ability"). Such a comparison will, for example, enable us to identify aspects of the original scenario that may be crucial for the success of our proposed emotion-like mechanism.

- A more systematic study of the predator-prey type relationship than has been conducted yet in the action selection literature with regards to affect. For instance, looking to see the *minimal* conditions under which our chosen mechanism (or emotion in general) might be adaptive. Both with regards to the capabilities of our agents' "brains" and "bodies", as well as features of the environment: varying both abilities of predator and prey. For, while others have looked at the role of emotion in the predator-prey scenario, they do not necessarily know or have not necessarily taken into consideration how their

mechanisms or emotion-based architectures might work, or be developed to work, in increasingly more dynamic environments. Or with different types of embodiment.

- An analysis of costs and benefits of both emotions and decisions in the predator-prey relationship. Looking at neuromodulatory effects as the basis for emotion, when used in different ways for agents (such as aggression for predator, fear for prey). But, in addition, also looking at action selection mechanisms more in terms of trade-offs. So, examining mechanisms as assessors of risk or opportunity cost: quick or rough-and-ready filters for behaviour and representations of the importance and limited nature of time. Looking at action selection in terms of a trade-off, between the time taken to decide and time taken for environmental circumstances to change adversely, temporally-adaptive responses may follow.

For Analysis of Adaptive Systems:

- A comparison and evaluation of measures of adaptive value (both quantitative and qualitative) that might be adopted. From internal measures of viability from examination of an individual agent, to Markov Models constructed from external observational data (by adopting the idea of activity cycles, thereby looking to analyse temporal behaviour of agents rather than simple life span etc).

- An analysis of the action selection problem in terms of the brain-body-environment relationship. Taking a broader look at action selection, so as to be asking whether we should actually be looking at the architecture alone in isolation, or whether we find out more by considering elements together. For example, considering both architecture and body, predator and prey, together, rather than individually. Moreover, looking at how (more realistic) two-way interactions may affect the performance of architectures and where emotion might fit in the relationship.

For System Design:

- A demonstration of how we might manipulate or adjust parameters so as to better "fine-tune" our mechanism and increase its value for adaptive action selection in this context (of the H3RP and predator-prey relationship). In particular, looking at how we might benefit from further distributing control and neuromodulatory influence across both agent architecture and agent body (as generators of brain-body-environment interactions).

We suggest that together these contributions will enable us to make an altogether much more comprehensive, perhaps even synergistic, contribution to the literature regarding action selection. Not only linking concepts such as action selection and emotion to the predator-prey relationship and brain-body-environment interactions; but, in turn, highlighting their more general contributions to the more intelligent design or creation of artificial life.

Acknowledgements

Funded by UH Studentship

References

Arkin, R. (2005). Moving up the food chain: Motivation and Emotion in behavior-based robots. *Who Needs Emotions: The Brain Meets the Robot*, pages 245–270.

Avila-García, O. (2004). *Towards emotional modulation of action selection in motivated autonomous robots*. PhD thesis, University of Hertfordshire, Hertfordshire.

Avila-García, O. and Cañamero, L. (2005). Hormonal modulation of perception in motivation-based action selection architectures. *SSAISB*.

Breazeal, C. and Scassellati, B. (2000). Infant-like social interactions between a robot and a human caregiver. *Adaptive Behavior*, 8(1):49.

Bryson, J. J. (2007). Mechanisms of action selection: Introduction to the special issue. *Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems*, 15(1):5–8.

Buason, G., Bergfeldt, N., and Ziemke, T. (2005). Brains, bodies, and beyond: Competitive co-evolution of robot controllers, morphologies and environments. *Genetic Programming and Evolvable Machines*, 6(1):25–51.

Chiel, H. J. and Beer, R. D. (1997). The brain has a body: adaptive behavior emerges from interactions of nervous system, body and environment. *Trends in Neuroscience*, 20:553–557.

Cooper Jr, W. (2000). Tradeoffs between predation risk and feeding in a lizard, the broad-headed skink (Eumeces laticeps). *Behaviour*, pages 1175–1189.

Fellous, J. (1999). Neuromodulatory basis of emotion. *The Neuroscientist*, pages 283–294.

Kelly, I., Holland, O., Scull, M., and McFarland, D. (1999). Artificial autonomy in the natural world: building a robot predator. pages 289–93.

Moshkina, L., Arkin, R., Lee, J., and Jung, H. (2009). Time-varying affective response for humanoid robots. *Progress in Robotics*, 44:1–9.

Nolfi, S. and Floreano, D. (1998). Coevolving Predator and Prey Robots: Do Arms Races Arise in Artificial Evolution? *Artificial Life*, 4(4):311–335.

O'Bryne, C., Cañamero, L., and Murray, J. (2009). The importance of the body in affect-modulated action selection: A case study comparing proximal versus distal perception in a prey-predator scenario. Affective Computing and Intelligent Interaction (ACII 2009). Proceedings 3rd International Conference. IEEE Computer Society Press.

Parisi, D. (2004). Internal robotics. *Connection Science*, 16(4):325–338.

Pfeifer, R. and Bongard, J. C. (2006). *How the body shapes the way we think: a new view of intelligence*. MIT Press, Cambridge, MA, USA.

Pfeifer, R. and Scheier, C. (1999). *Understanding intelligence*. MIT Press, Cambridge, MA, USA.

Socio-Technical Systems

Memes in Artificial Life Simulations of Life History Evolution

John A. Bullinaria

School of Computer Science, University of Birmingham, Birmingham, B15 2TT, UK
j.a.bullinaria@cs.bham.ac.uk

Abstract

The effect that learning has on Life History Evolution has recently been studied using a series of Artificial Life simulations in which populations of competing individuals evolve to learn to perform well on simple abstract tasks. Those simulations assumed that learning was achieved by identifying patterns in sets of training data, i.e. through direct experience. In practice, learning is not only by direct experience, but also by imitation of others. Such imitative information transfer is now often formulated in terms of memes being passed between individuals, and it is clear that this is a substantial part of real learning processes. This paper extends the previous study by incorporating imitation and memes to provide a more complete account of learning as a factor in Life History Evolution.

Introduction

Computational models based on neural networks that learn from a stream of experience (i.e. representative input-output samples) have provided good accounts of numerous aspects of human behaviour. Extending those models to Artificial Life simulations of evolving populations of competing neural network based individuals can then lead to improved understanding of more general aspects of human development and "life history", such as the periods of protection that parents offer their young and ages at first reproduction (Bullinaria, 2009). Those simulations elucidated the trade-off between learning quickly and learning well, and showed how evolution can balance the trade-off to result in the emergence of extended periods of parental protection during which learning could be completed slowly and effectively without the impact of fitness based natural selection pressures.

The Bullinaria (2009) Life History Evolution study began by using a simple artificial neural network based system that allowed each individual to learn from a set of training patterns, and then moved on to study non-neural network abstractions of that kind of learning process, that were more computationally efficient for large scale evolutionary simulations. What all those simulations assumed was that the learning was achieved by identifying patterns in relevant training data, i.e. through direct experience. In practice, learning is not purely by direct experience, but also by imitation of learned performance of others. Such information transfer can be formulated in terms of *memes* being passed between individuals (e.g., Brodie, 1996; Blackmore, 1999), and it is clear that this, in its most general form, is a large part of the human learning process, and maybe also of other animal species. It is therefore important to incorporate imitation and memes into any complete account of learning as a factor in Life History Evolution. As always, there will be trade-offs between the various costs involved (Stearns, 1989, 1992). In many ways, the relevant trade-offs are clear from a theoretical point of view, but the interactions are complex and highly dependent on the associated parameters. It is only by running comprehensive series of simulations that the effect of the various parameter values becomes apparent.

Already Higgs (2000) has simulated the evolution of learning by imitation, but that study didn't consider how that learning might interact with more traditional neural learning by direct experience, and it is not immediately obvious how best to bring those different forms of learning together. One of the key results of Bullinaria (2009) was that it is possible to abstract out almost all the details of the neural learning, and still be left with a system that resulted in the evolution of the same life history properties. Although it was not the intention at the time, that abstraction process also provides a relatively straightforward way of incorporating imitative learning into the same system. Therefore, the aim of this paper is to introduce a parameterized account of memes and imitation into the approach of Bullinaria (2009), and begin to explore the effect that imitation has on the various life history and human development factors.

In the remainder of this paper, the underlying Artificial Life framework is first described, and then the details are provided about how the direct learning and imitation processes can be modelled efficiently. This is followed by a presentation of the results from a representative series of simulations designed to test and explore many of the key relevant issues. The paper ends with some discussion and conclusions.

The Artificial Life Framework

The simulation approach involves evolving populations of individuals, each specified by a set of innate parameters, that must learn to perform well on some abstract task. The fitness of each individual at each stage will simply be how well it has so far learned the given task. Forcing the individuals to compete to survive and procreate, according to their relative fitness, results in the emergence of populations of increasing ability. Moreover, to compete effectively in a population consisting of individuals of all ages, each individual must not only learn how to perform well, but must also be able to learn *quickly* how to achieve that good performance, or at least quickly enough that it can survive after its parents have withdrawn their protection. This leads to the evolution of

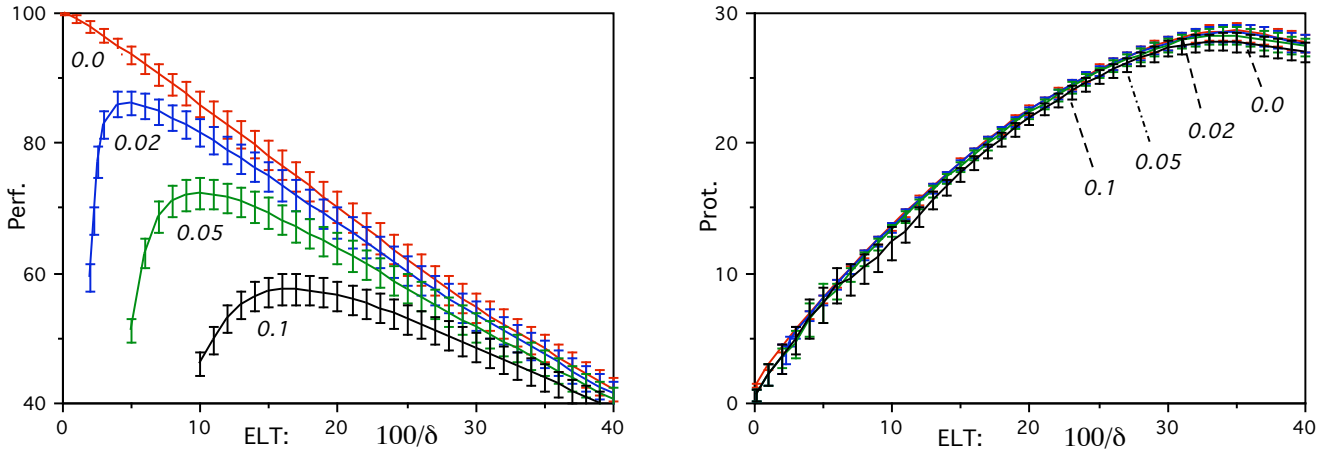


Figure 1: The mean evolved performance levels and protection periods as a function of the ELT $100/\delta$ when the linear individual performance improvement with age stops with probability $\rho\delta$ at a random cost in the range [0,100], for $\rho \in \{0, 0.02, 0.05, 0.1\}$.

riskier learning strategies than over-simplified "generational" approaches that involve weaker selection pressures and do not match real environments so well (Bullinaria, 2007a).

In all the simulations, a fixed population size is maintained (that is consistent with fixed total food resources available to support the population) by replacing the individuals that have died by children of the most fit individuals. Deaths occur by losing a fitness comparison "fight" against other individuals, or randomly due to old age beyond a natural life-span (set here to be around twice the time typically taken to learn the simulated task, namely 30 simulated years). The children are generated by cross-over and mutation from two parents chosen each simulated year by pair-wise fitness comparisons of the eligible individuals. This is implemented by having each child inherit innate parameters chosen randomly from the corresponding ranges spanned by its two parents, plus a random mutation (from a Gaussian distribution) that gives it a significant chance of falling outside that range. Although these details are clearly over-simplifications of real animal populations, they constitute a manageable approximation of all the key processes, and have proved effective in numerous previous studies (e.g., Bullinaria, 2007a,b, 2009).

The Bullinaria (2009) study began with a learning process based on standard fully connected Multi-Layer Perceptron neural networks with one hidden layer, sigmoidal processing units, and training by gradient descent using the cross-entropy error function on simple classification/categorization tasks. The main life history factor explored in that study was the protection of children by their parents until they had reached a certain age, so they could not be killed by competitors before then. That added an implicit cost to the parents in that the more they protected their children, the more likely they were to die themselves through competition. Simulations that evolved the protection period, as well as all the neural learning parameters, established that clear learning advantages and better adult performances were possible if children received longer periods of parental protection, but only if the children were not allowed to reproduce during their period of protection. If procreation was not prevented in that way, the competition to reproduce led to learning strategies that result in worse adult performance. When procreation is prevented while protected, a compromise protection period evolves that balances the improved learning performance against the reduced period for procreation. It was also shown that the evolved protection period increases with life-span, rather than remaining at a fixed duration determined by the learning task complexity, illustrating the trade-off involved and confirming the importance of learning well.

Abstracting the Neural Learning Process

An important result of Bullinaria (2009) was that it is possible to approximate the full neural network learning process by a single performance level that varies as a simple parameterized function of age, and still end up with qualitatively the same Life History Evolution results. The simplest stochastic approximation would be to have each individual's learning performance (i.e. fitness) rise approximately linearly with age from 0 up to 100% in steps drawn randomly each year from the range $[0, 2\delta]$. Simulations using different learning rates δ then show that the population mean performance falls almost linearly with the Expected Learning Time (ELT), i.e. $100/\delta$, and the evolved protection period rises approximately linearly with $100/\delta$, but peaks near the point at which individuals start dying of old age. Predictably, the best mean performance is achieved with very high learning rates δ, for which all individuals reach perfect performance before their first round of competition to survive or procreate at the end of their first year. Consequently, if the learning rate δ is evolved along with the protection period, it quickly achieves very high levels, and the protection period goes to zero. Of course, with real neural networks one cannot just keep on increasing the learning rate and expect the learning time to decrease with it. Eventually, at some task dependent point, the approximation to true gradient descent breaks down, and the learning performance deteriorates. In that case, the evolutionary process will find the best values for the learning parameters, and having slower learning with longer protection periods does consistently emerge to provide a clear advantage.

A better approximation to the full neural learning process, that has faster learning leading to riskier learning strategies

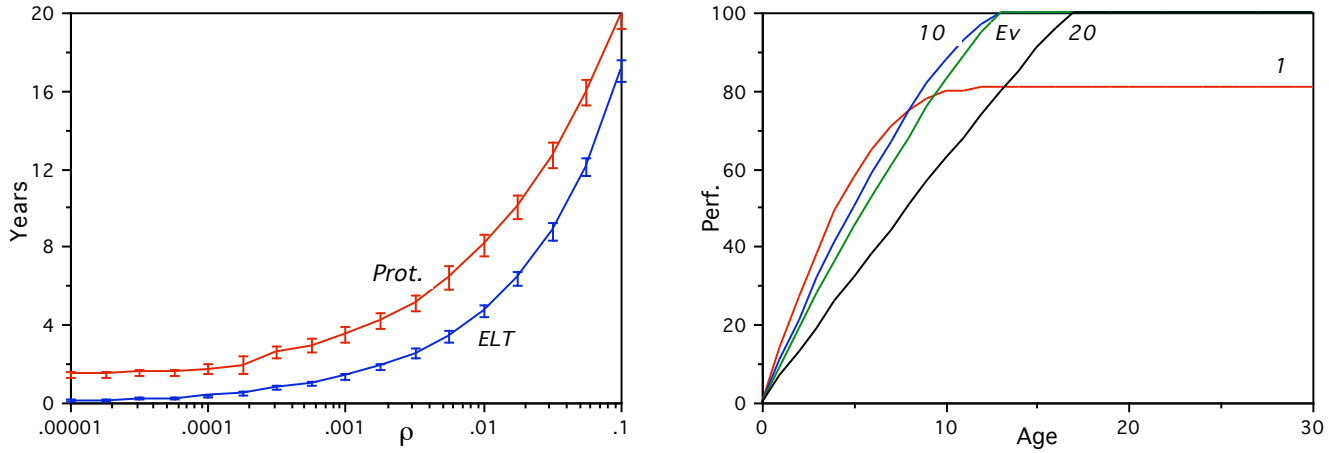


Figure 2: The mean evolved ELT $100/\delta$ and protection period as a function of learning task difficulty parameter ρ (left), and the median learning performance as a function of age for the evolved and three other protection periods with $\rho = 0.04$ (right).

which increasingly lead to persistent poor performance, is achieved by simply having the learning process stop at some random point in the performance range [0, 100] with a probability $\rho\delta$ that increases linearly with both the learning rate δ and an associated "task difficulty" parameter ρ. The left graph of Figure 1 shows how the mean performance then depends on the ELT $100/\delta$ for four representative values of ρ. The higher ρ is, the lower the value of δ at which significant deviations from the earlier $\rho = 0$ case arise. The right graph shows that the relation between the evolved protection period and $100/\delta$ is not much affected by the size of ρ.

The performance plot shows a clear maximum for each value of ρ, and successful evolutionary processes will result in the emergence of the corresponding optimal learning rates δ with their associated non-zero protection periods. The left graph of Figure 2 shows the mean Expected Learning Times $100/\delta$ and protection periods that actually emerge through evolution as a function of the parameter ρ. As ρ increases, the best possible learning time $100/\delta$ also increases, and the best protection period follows suit. The evolved protection period is always slightly longer than the ELT $100/\delta$. This is because of the stochastic nature of the learning process and the fact that the mutations lead to distributions of learning rates and protection periods, and the obvious advantage of protection periods being long enough to accommodate a reasonable number of individuals that are slower than average.

The parameter ρ is seen to act as an abstract measure of learning difficulty, and can be regarded as an approximate representation of the difficulty the neural network learning algorithm has with its given task. Although this is a rough approximation to reality, it does have the required properties. Relatively easy tasks correspond to low ρ, are learned quickly, and have short associated protection periods. Harder, or more complex, tasks correspond to higher values of ρ, take longer to learn, and benefit from longer protection periods. The individual performance levels that emerge in the abstracted learning models were compared directly by Bullinaria (2009) with those arising from the full evolutionary neural network simulations, and a good qualitative correspondence was found for $\rho = 0.04$. The right graph of Figure 2 shows the median performance levels as a function of age for this case. The mean evolved ELT $100/\delta$ is around 10 years and the mean evolved protection period is around 14 years. As for the full neural simulations, the results arising with evolved protection period (*Ev*) were compared with three fixed protection periods (*1, 10, 20*). The linear learning approximation and uniform distribution of residual errors are rough approximations of the real neural learning processes, but the broad pattern of results is found to be the same: Longer protection periods allow slower learning and result in better adult performance, but not allowing procreation while being protected prevents the evolved protection periods from becoming excessively long. The effects of changing the age at onset of "old age", and of allowing procreation while protected, are also found to be in line with those of the full evolving neural networks.

There certainly remains much scope for more accurate parameterizations for specific real learning processes, as discussed by Bullinaria (2009), but the current set-up will suffice for the preliminary investigation of memes here.

Incorporating Imitative Learning

The main aim of the abstracted neural learning process was to improve the computational efficiency, and hence allow more detailed Life History factors to be simulated, but it also renders it feasible and fairly straightforward to incorporate learning by imitation into the same performance function.

The basic idea is that it will often be more efficient to imitate the successful behaviour of another individual than it is to learn it from direct experience. One can think of the transmission of behavioral practices or cultural ideas between individuals, and those *memes* will replicate and respond to natural selection pressures in a manner analogous to genes (Dawkins, 1976; Brodie, 1996; Blackmore, 1999). It seems likely that humans have evolved to learn by imitation as well as direct experience across a wide variety of tasks (e.g., Richerson and Boyd, 1992; Offerman and Sonnemans, 1998), though other species appear to imitate to a much lesser extent (e.g., Byrne and Russon, 1998; Blackmore, 1999; Zentall, 2001). There has been considerable recent interest in this idea across a range of disciplines (e.g., Hurley and Chater, 2005;

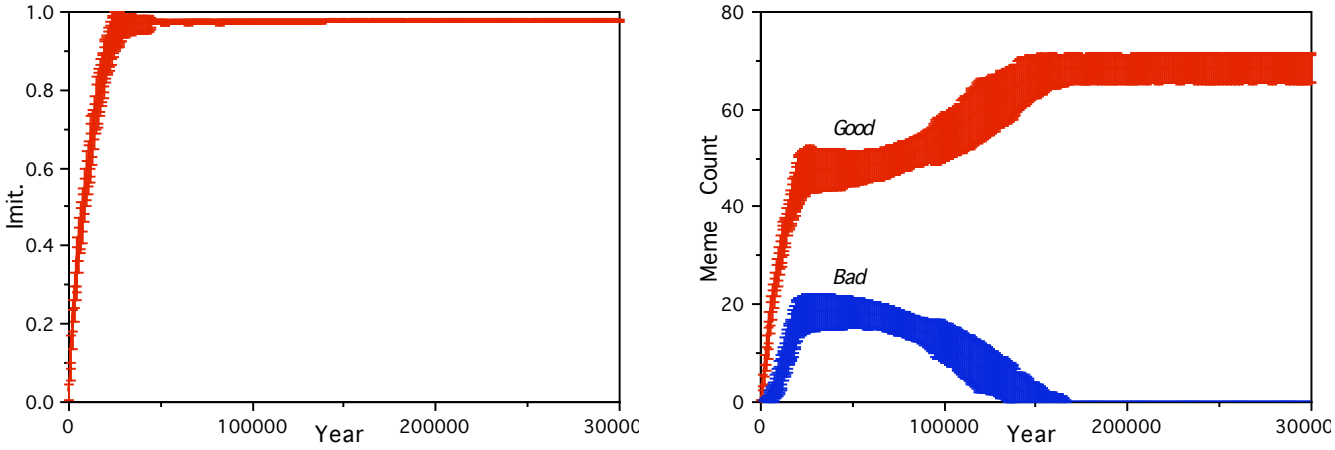


Figure 3: The evolution of imitability (left), and the change in average numbers of good and bad memes known by individuals throughout evolution (right), for 16 runs of the basic imitation-only simulation with limited brain sizes.

Nehaniv and Dautenhahn, 2009). The thinking here is that Artificial Life simulations will be best placed to explore this issue in the context of other Life History traits.

Some interesting preliminary work has already been carried out. Belew (1990) and Best (1999) have introduced imitation based cultural factors into the Hinton and Nowlan (1987) model of learning guiding evolution, but that work is far removed from the neural inspired learning relevant to the life history factors of relevance here. Borenstein and Ruppin (2003) address many of the limitations of those earlier studies, and do incorporate neural learning mechanisms, but they actually prevent cultural evolution by not allowing meme transmission between generations and only allowing innate behaviours to be imitated.

The study of Higgs (2000) comes closest to exploring the life history issues of interest here. That paper considered the evolution of populations of individuals that may invent and imitate memes, and investigated a range of factors that affect how the imitation rates, fitness levels, and number of memes evolve. The key finding was that imitative ability does consistently emerge under a range of conditions, even when some memes have a negative effect on fitness, and/or there is an inherent cost in the ability to imitate. In many ways it is obvious that if there exist memes with a range of positive and negative effects on fitness, then not imitating will leave the fitness at some baseline, whilst imitation will result in a range of fitness levels above and below that baseline. Selection on the basis of fitness will then favour those individuals that have imitated the good memes, and hence favour imitative ability. Moreover, since it favours individuals that have acquired and can pass on those good memes, the good memes will tend to propagate at the expense of the bad memes. Memes acting together (i.e. *memeplexes*), the interplay of genetic and cultural fitness, and the interaction of genetic and mimetic replicators, all complicate this simple picture (e.g., Brodie, 1996; Blackmore, 1999; Best 1999), but these are all things that can be incorporated into future simulations.

The main question this paper aims to address is: how can the Life History Evolution approach of Bullinaria (2009) be extended in a way that enables these issues to be studied in conjunction with direct lifetime learning processes?

Simulating Memes and Imitation

For the extraction of reliable conclusions from Artificial Life simulations it is important to avoid confounding factors, so to explore general ideas it is usually wise to keep the models much simpler than when the aim is to model particular real life scenarios. Moreover, it is important to parameterize the models (e.g., like introducing the parameter ρ above) so that they remain relevant to a range of species, tasks, etc. and allow comparisons between them. The aim here is to develop such a parameterized framework that is general enough to cover learning from others in the most general sense, that includes (but is not limited to) simple imitation.

Unfortunately, the details of the Higgs (2000) study do not match with the current aims. In particular, it did not consider the details of any of the processes taking place during the individuals' lifetimes, and it used non-overlapping generations which means a total absence of the competition between individuals of different ages that underlies so many of the issues of interest here. Other factors simply complicate the analysis unnecessarily, such as using Gaussian distributions for the meme fitnesses and mutations, the non-linear relation between learning ability and probability of imitation, and the unbounded number of memes that can be invented. So, instead of following the approach of Higgs (2000), the approach of Bullinaria (2009) will be extended in a minimal computationally efficient manner to include the key concepts of memes and their imitation.

The starting point is to assume that there exist a set of M memes $\{m_j : j = 1,\ldots,M\}$ and that each individual i at each stage of its life will have acquired some subset of them to be stored in their brain of size B_i. There is no need to specify exactly what the memes represent, nor worry about the details of the imitation process. It will also be assumed that all the memes are of equal complexity and imitability, though they may contribute unequally to fitness of the individuals that possess then. To begin with, the individuals' baseline fitness will be 0, and half the memes will be deemed good memes that increase this by 1, and the other half will be bad memes that decrease it by 1. So each individual i can potentially increase its fitness during its lifetime from 0 up to B_i.

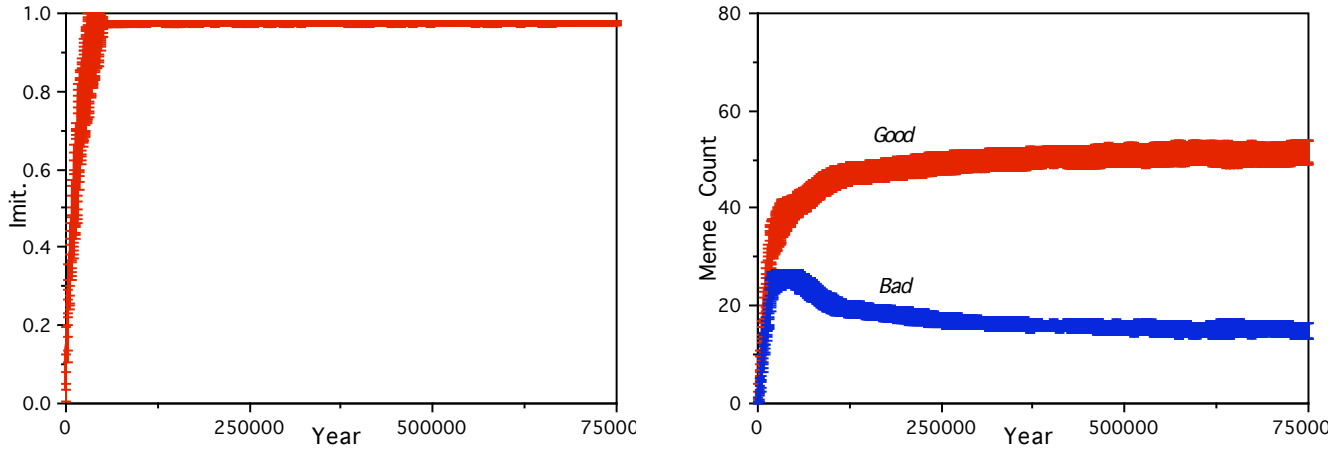


Figure 4: The evolution of imitability (left), and average numbers of good and bad memes known by individuals (right), for 16 runs of imitation-only simulations with limited brain sizes and cultural fitness based imitation selection.

The imitative ability α_i of all individuals i in the initial population will be zero, but the mutations and crossovers as described above enable it to evolve from zero up to a maximum of 1 if that proves beneficial. Then during each simulated year, each individual can acquire up to $\alpha_i \phi B_i$ memes from other individuals, where ϕ is a parameter that specifies the maximum rate at which memes can be copied. To inject memes into the populations with minimal disruption to the imitative process, each year one randomly chosen individual acquires one randomly chosen meme with probability r if its brain is not already full. Figure 3 shows what happens if $M = 400$, $B_i = 100$, $\phi = 0.1$ and $r = 0.01$, with just the imitabilities α_i allowed to evolve. The tournament based selection of parents, deaths and copied individuals give the good memes an advantage over bad memes, so the number of bad memes rises more slowly than the good memes, and when the number of known memes reaches the level that brains regularly reach full capacity (~20,000 years), the number of bad memes begins to fall and eventually becomes negligible (~150,000 years). There is a clear advantage to acquiring memes throughout, and so the imitability quickly rises to near 1. The behaviour during the lifetime of a typical evolved individual is a simple linear acquisition of memes over the first $1/\phi = 10$ years, at which point the brain reaches full capacity and maximum performance is achieved. Children are then produced until death due to old age. Most deaths due to competition occur during the meme acquisition period.

There are interesting dependencies on who exactly is imitated to acquire memes. If memes are copied from random individuals, there is still enough selection pressure to eradicate the bad memes, but it takes about twice as long (~300,000 years). If each individual first acquires memes from their own parents, before imitating random others, the number of bad memes disappears more quickly (~130,000 years). If parents are imitated before fitness selected others, the bad memes go even more quickly (~120,000 years). Since parents have already gone through fitness selection to become parents, and are also older and more experienced, they are a better source of memes than other fitness selected individuals. In fact, if individuals *only* copy from their parents, significant numbers of bad memes never build up at any stages of evolution.

Another factor that affects the results is the basing of who to imitate on cultural fitness (Higgs, 2000). In this case, each meme has a cultural fitness that is not correlated with its standard (biological) fitness, and individuals are chosen for imitation according to the total cultural fitness they have acquired. As Figure 4 shows, this allows memes of high cultural fitness to persist in the population, even if they are actually bad memes. This is independent of what contributes to the cultural fitness of those bad memes. Obviously, there are numerous related factors, such as cognitive dissonance (Cooper, 2007) and memes associating into memeplexes (Blackmore, 1999), that will increase or decrease this effect to varying degrees, and these are more issues that may be worth attempting to incorporate into future simulations.

The effect of copying fidelity also needs consideration. This can easily be approximated by having a fraction $1–f$ of good memes incorrectly copied and thereby transformed into bad memes. As the fidelity f is reduced from 1, the pattern changes from that like Figure 3 but with increasing times needed to eradicate the bad memes, to something like Figure 4 with persistent levels of bad memes.

Finally, it is important to understand how the results depend on the relation between the total number of memes and the brain capacity. For $M = 200$, $B_i = 200$ and everything else the same, the simulation results of Figure 3 take on the rather different pattern seen in Figure 5. Now all individuals can acquire all memes, and it proves much more difficult to separate the good from the bad so that selection pressures can act. In this case evolution ends up with only slightly more good memes than bad, and there is little pressure towards high levels of imitability. Interestingly though, the strategy of only imitating ones own parents does manage to prevent the build-up of bad memes in this case too.

A central recurring feature of the Higgs (2000) study was a "mimetic transition" at which there is a dramatic rise in imitative ability and number of memes, and it was shown how numerous factors affected the timing of that transition. In the current framework, that transition virtually always happens right at the start of the evolutionary process.

There is certainly much more to memes and imitation than has been introduced here (e.g., Brodie, 1996; Blackmore,

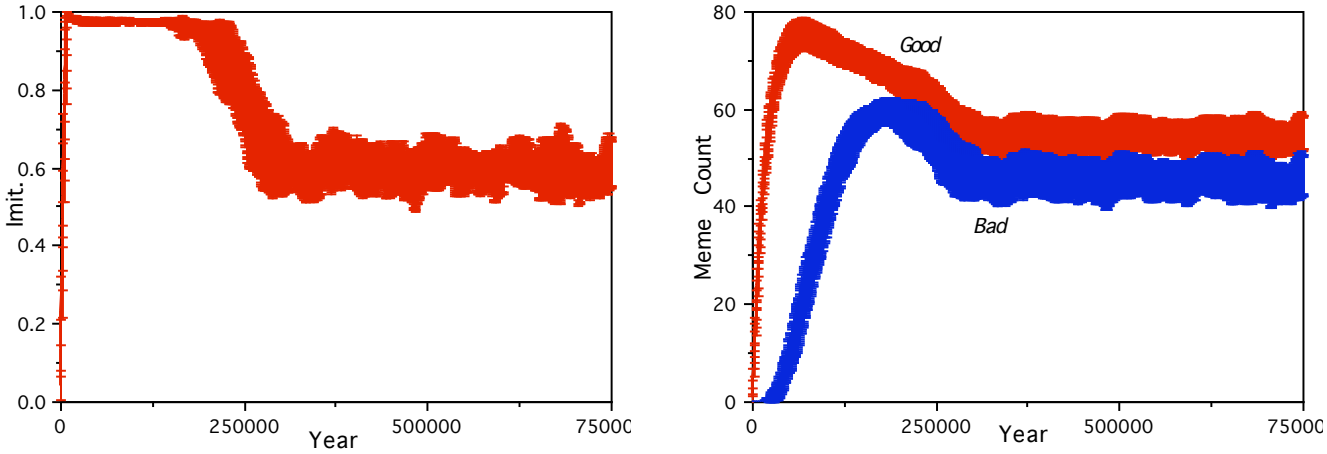


Figure 5: The evolution of imitability (left), and average numbers of good and bad memes known by individuals (right), for 16 runs of the basic imitation-only simulation with brains large enough to accommodate all known memes.

1999), but the framework as described above already includes all the key ideas necessary to make progress.

Simulating Direct Learning

Having formulated the key mimetic factors, the direct lifetime learning factors of Bullinaria (2009) can now be reinstated. The natural way to do this in terms of memes is to have $\delta_i \psi B_i$ random memes learned each year, where δ_i is an evolvable learning rate, and ψ is an intrinsic measure of learning difficulty. The time to learn to brain capacity is then $1/\delta_i\psi$, and for $\psi = 0.01$ the expected learning time matches that of the Bullinaria (2009) simulations. The learning difficulty parameter ρ that prevents the evolution of unrealistically high learning rates can be implemented easily here by learning a bad meme rather than a good meme with probability $\rho\delta$. Then the evolved learning rates balance the trade-off between learning quickly and having too many fitness reducing bad memes, with results equivalent to the full neural network simulations of Bullinaria (2009).

Life History Simulation Results

The simulations become even more interesting when the imitation and direct learning occur together and interact with life history traits such as protection periods. But, before doing that, there are a few more important details that need to be added to render the simulations reasonably realistic.

First, it is possible for an individual to acquire both good and bad "versions" of the same meme via different routes. The resolution of meme inconsistencies in reality is known to be a complex issue (Cooper, 2007), but a convenient approach to start with here is to have the good and bad memes come in pairs that simply cancel each other out if they occur together. In this way, a bad meme arising from direct learning can be removed if the corresponding good meme is copied from another individual. Similarly, a bad meme arising from poor copying fidelity can be removed by later acquiring the corresponding good meme by direct learning or by copying from a different individual.

Second, in reality, the rate of meme acquisition is unlikely to be as constant as in the processes described above. Instead, more realistic results are produced by a stochastic version, where each usage of the parameters α_i and δ_I are replaced by random numbers from the respective ranges $[0, 2\alpha_i]$ and $[0, 2\delta_i]$, like in the Bullinaria (2009) study.

Figure 6 shows the evolution of the key parameters and resultant meme counts when $M = 400$, $B_i = 100$, $\phi = 0.1$, $\psi = 0.01$, $r = 0.01$, $f = 0.9$ and $\rho = 0.001$. In this case, both copying and direct learning contribute to the learning process, and bad memes are kept to very low levels. The protection period settles to slightly above the typical learning time as in the full neural simulations of Figure 2.

The implementational details obviously affect exactly what emerges from the simulations, and it is those differences that reflect the wide range of life history patterns for the different species that have emerged from biological evolution. Varying the details and parameters allows a systematic exploration of the trade-offs and interactions that lead to specific traits. A few simple examples will now illustrate the kind of factors that can be investigated within this framework.

The issue of whether to allow procreation while protected produced interesting results in the direct learning study of Bullinaria (2009). In that case, if procreation was allowed while protected, the protection periods rose so that there were only deaths due to old age and no deaths by competition, and the selection pressure to learn fast to procreate early resulted in higher learning rates that led to poorer adult performance. This no longer happens in the current meme based framework. Since the errors arising from faster learning can now be corrected by copying (or being taught), such fast learning will emerge without a deterioration of the final adult performance. Increased protection periods again remove the worry of early death due to competition, so if some unlucky individuals are slow in correcting their direct learning errors, that is compensated overall by the faster early learning in others. The balance between the two forms of learning, parameterized here by ϕ, ψ, f and ρ, will determine exactly what emerges, and the way forward would be to attempt to understand species specific differences in terms of variations in such parameter values.

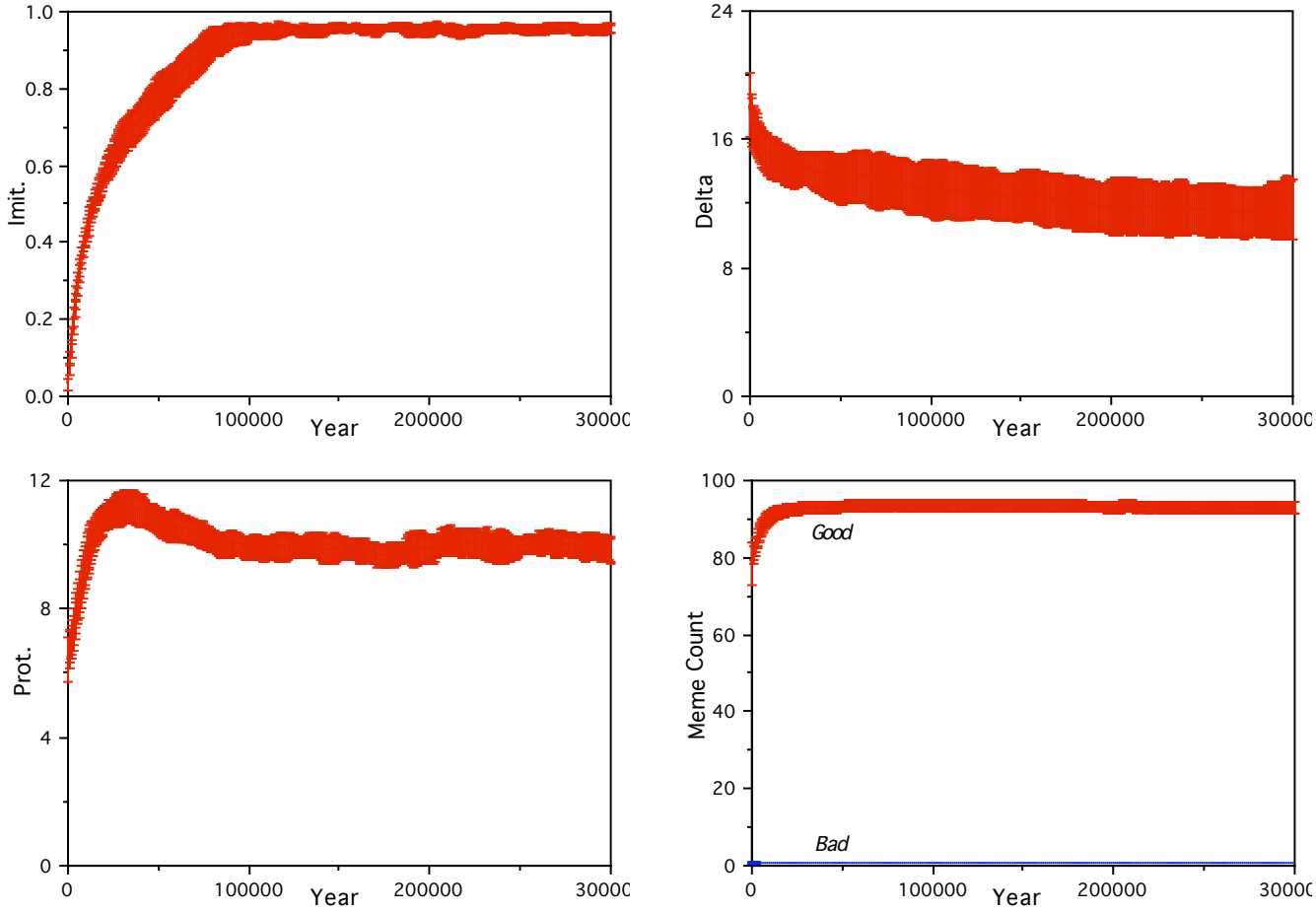


Figure 6: Evolution of the full imitation and direct learning system with copying fidelity f = 0.9 and ρ = 0.001: the average imitability α (top left), learning rate δ (top right), protection period (bottom left) and resultant meme counts (bottom right).

The copying fidelity, parameterized by f, has a particularly large effect on what emerges. If it is raised from the 0.9 of Figure 6 up to 1.0, so that all the copying is exact, evolution results in perfect performance being achieved more quickly and more reliably. One might predict that the evolved direct learning rates δ will then decrease to enable more reliable memes for copying, but they actually increase from 12 to 19, because copying can now more effectively correct any direct learning errors. Overall, the evolved protection period can be reduced from 10.0 to 7.6 years to enable a longer procreation period. The trade-offs are such that fidelity differences affect what emerges in different ways depending on the values of the other parameters. This again illustrates the need for a flexible modeling framework to explore such interactions.

If the copying fidelity is very low, a high imitative ability α never evolves because it introduces too many bad memes into the population, and one ends up with direct learning only, as appears to be the case for most animal species apart from humans. Also, if mechanisms are not available to remove bad memes, interesting changes in imitative ability can arise throughout evolution. For example, Figure 7 shows one such case in which the number of bad memes repeatedly rises to such high levels that the best strategy is to stop copying until all the carriers have died, and then start again.

The brain size is another crucial factor that can be evolved, and in the simulations described above it invariably grows to the maximum allowed. Obviously, for real animals there are significant costs associated with having larger brains, and trading those costs against the improved performance that results from a bigger brain leads to particular brain sizes emerging (e.g., Blackmore, 1999; Striedter, 2005). It actually proves easy to add such costs into the simulations to limit the brain sizes that emerge, but the cost implementations are not yet sophisticated enough that the models can provide reliable testable predictions about particular species.

Discussion and Conclusions

This paper has made the first steps in introducing imitative learning and memes into Artificial Life simulations of Life History Evolution. The main contribution has been to present a flexible framework which allows a computationally efficient way of parameterizing and exploring any hypotheses in this field. There are certainly numerous simplifications and approximations involved, which have been highlighted throughout, but the basic structures and ideas are in place, and they have already been shown to replicate the key results of earlier approaches and improve upon them.

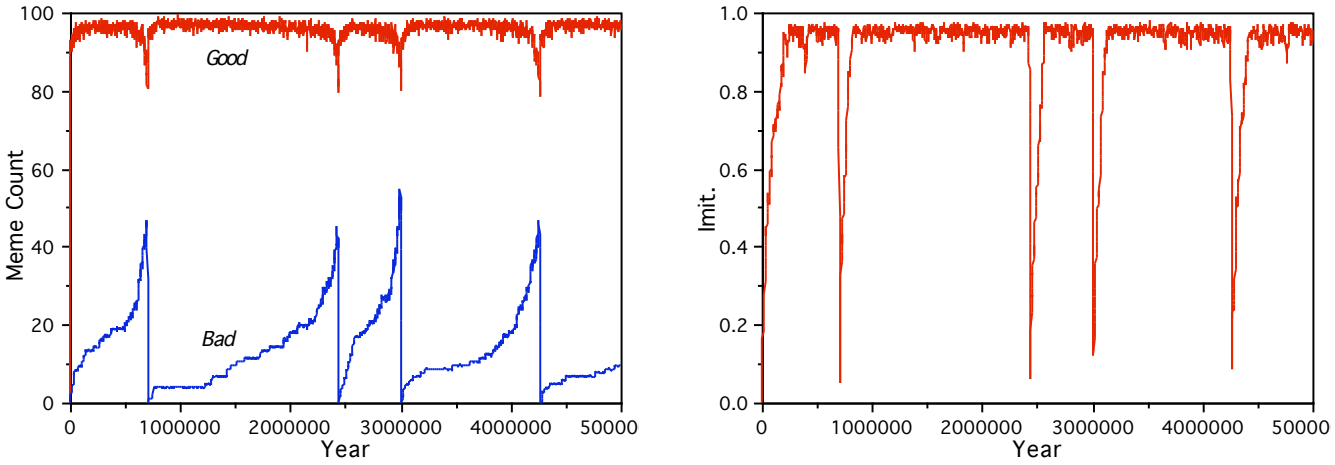


Figure 7: When bad memes (left) are allowed to build up, the evolvable imitative ability (right) can fall quickly to very low values so that the bad memes die out, and then return to the earlier high level until the problem arises again.

Even this simplified framework can be used to investigate an enormous number of interactions and trade-offs. This paper has only presented results from a small selection of simulations to illustrate the kinds of issues that can be explored. Experiments studying further issues will be reported in a longer paper elsewhere. The simulation results so far are in line with existing intuitions, which instills confidence that they can now be taken further with some reliability to explore issues for which our intuitions are not so clear and controversy remains.

There are numerous aspects of the current set-up that could be improved further without too much effort. One would be the refinement of the parameterization of direct learning, and the relation of that to different types of animal learning. Some preliminary attempts involving more parameters and different distributions of good and bad memes have shown that they do indeed re-balance the trade-offs slightly, but no fundamentally different behaviours have yet emerged. Specific details of the mechanisms for removing bad memes tend to have a more dramatic effect on the results, as Figure 7 shows. Building in associations between good and bad memes and simulating the creation of memeplexes (Blackmore, 1999), and introducing related mechanisms for the resolution of cognitive dissonance (Shultz and Lepper, 1996; Cooper, 2007), are obvious avenues for future enhancement of the framework in that direction, but it is not clear what fundamentally new results might emerge from that. More challenging future work will involve the incorporation into the existing framework of more realistic additional indirect performance costs related to biological factors (such as the cost of running a larger brain, or of providing parental protection, or of allowing copying, or of teaching), and better distinction between types of learned behaviour and related factors such as ease of copying.

References

Belew, R.K. (1990). Evolution, learning, and culture: Computational metaphors for adaptive algorithms. *Complex Systems*, 4:11-49.

Best, M.L. (1999). How culture can guide evolution: An inquiry into gene/meme enhancement and opposition. *Adaptive Behavior*, 7: 289-306.

Blackmore, S. (1999). *The Meme Machine*. Oxford, UK: OUP.

Borenstein, E. and Ruppin, E. (2003). Enhancing autonomous agents evolution with learning by imitation. *AISB Journal,* 1:335-347.

Brodie, R. (1996). *Virus of the Mind: The New Science of the Meme*. Seattle, WA: Integral Press.

Bullinaria, J.A. (2007a). Using evolution to improve neural network learning: Pitfalls and solutions. *Neural Computing and Applications*, 16:209-226.

Bullinaria, J.A. (2007b). Understanding the emergence of modularity in neural systems. *Cognitive Science*, 31:673-695.

Bullinaria, J.A. (2009). Lifetime learning as a factor in Life History Evolution. *Artificial Life*, 15:389-409.

Byrne, R.W. and Russon, A.E. (1998). Learning by imitation: A hierarchical approach. *Behavioral and Brain Sciences*, 21:667-684.

Cooper, J. (2007). *Cognitive Dissonance: 50 Years of a Classic Theory*. London, UK: Sage.

Dawkins, R. (1976). *The Selfish Gene*. Oxford, UK: OUP.

Higgs, P.G. (2000). The mimetic transition: A simulation study of the evolution of learning by imitation. *Proceedings of the Royal Society B: Biological Sciences,* 267:1355–1361.

Hinton, G.E. and Nowlan, S.J. (1987). How learning can guide evolution. *Complex Systems*, 1:495-502.

Hurley, S. and Chater, N., editors (2005). *Perspectives on Imitation: From Neuroscience to Social Science.* Cambridge, MA: MIT Press.

Nehaniv, C.L. and Dautenhahn, K., editors (2009). *Imitation and Social Learning in Robots, Humans and Animals.* Cambridge, UK: CUP.

Offerman, T. and Sonnemans, J. (1998). Learning by experience and learning by imitating successful others. *Journal of Economic Behavior and Organization*, 34:559-575.

Richerson, P.J. and Boyd, R. (1992). Cultural inheritance and evolutionary ecology. In: E.A. Smith and B. Winterhalder, editors, *Evolutionary Ecology and Human Behaviour*, 61-91. Chicago, IL: Aldine de Gruyter.

Shultz, T.R. and Lepper, M.R. (1996). Cognitive dissonance reduction as constraint satisfaction. *Psychological Review*, 103: 219-240.

Stearns, S.C. (1989). Trade-offs in life history evolution. *Functional Ecology*, 3:259-268.

Stearns, S.C. (1992). *The Evolution of Life Histories.* Oxford, UK: OUP.

Striedter, G.F. (2005). *Principles of Brain Evolution*. Sunderland, MA: Sinauer.

Zentall, T. (2001). Imitation in animals: Evidence, function, and mechanisms. *Cybernetics and Systems*, 32:53–96.

Darwinian evolution of culture as reflected in patent records

Andrew Buchanan,[1] Norman Packard,[2] and Mark Bedau[1,2,3*]
[1]Center for Advanced Computation, Reed College, Portland
[2]ProtoLife Inc, San Francisco
[3]Initiative for Science, Society, and Policy, University of Southern Denmark, Odense
[*]Corresponding author: mab@reed.edu

Abstract

We argue that culture undergoes an evolutionary process, analogous to biological evolution. As evidence, we analyze the bibliographic information of all utility patents issued in the United States from 1976 through 2007, which comprise over three million patents. The set of issued patents is regarded as an evolving population. A patent is considered to "reproduce" when it is cited by a new patent, and variability is introduced into the population by the innovations in new patents. We analyze patent records with statistics that quantify the degree to which the population of patents is shaped by natural selection, and we find convincing evidence of Darwinian evolution. Further, weighting our statistics by the classification distance between parent and child shows that the most fecund patents are "door-opening" technologies that enable an especially broad range of further innovations.

Introduction

We study the evolution of technology as reflected in US patent records. Everyone agrees that technology evolves, but there is controversy about what this means, and especially whether the evolution of technology is "Darwinian" in some interesting sense (Jablonka (2002); Benzon (1996)). By Darwinian evolution, here, we mean that the process of natural selection in a population is a significant factor in explaining how the traits in the population change over time. Natural selection, in turn, is defined as the process by which heritable traits that make members of a population more likely to survive and reproduce tend to be increasingly represented in the population over time.It should be noted that our conception of Darwinian evolution is consistent with cultural evolution being simultaneously significantly shaped by many non-Darwinian mechanisms, like random genetic drift, pleiotropy, and epigenesis (Jablonka and Lamb (2005); Sperber (1996)).

In this paper, we develop methods to address the following two questions:

1. Does natural selection shape the evolution of technology?
2. If so, what kinds of technological innovations especially drive its evolution?

Our aim is both to show the value of the methods, even when applied in new settings and adapted to new contexts, and also to investigate and learn from the first fruits of applying the methods to patent data. In the end, our conclusions will be two: (1) Natural selection significantly shapes the evolution of patented technology, and (2) the statistical evidence corroborates the hypothesis that so-called "door-opening" technologies have been especially important drivers of the evolution of technology.

Our project applies earlier work on evolutionary activity statistics (Bedau and Packard (1992); Bedau et al. (1997, 1998); Bedau and Brown (1999); Rechtsteiner and Bedau (1999); Raven and Bedau (2003)) and significantly expands and develops an earlier similar pilot project (Skusa and Bedau (2002); Bedau (2003)).

Patent data

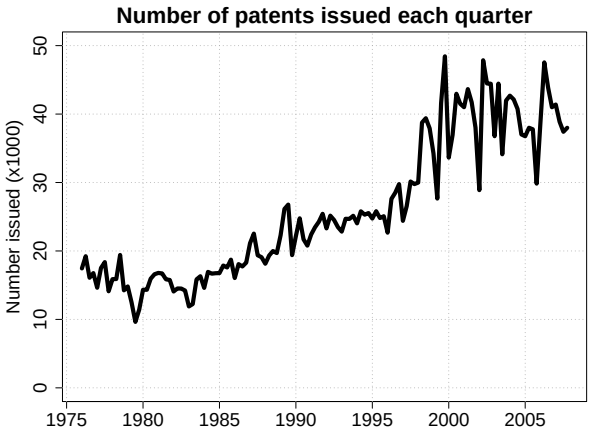


Figure 1: Number of patents issued each quarter, over the thirty years in our database.

The patent data we mine in this experiment consists of records of US patents issued over thirty years from 1976 through 2007. Figure 1 shows that the rate at which patents have been issued has doubled over the past thirty years.

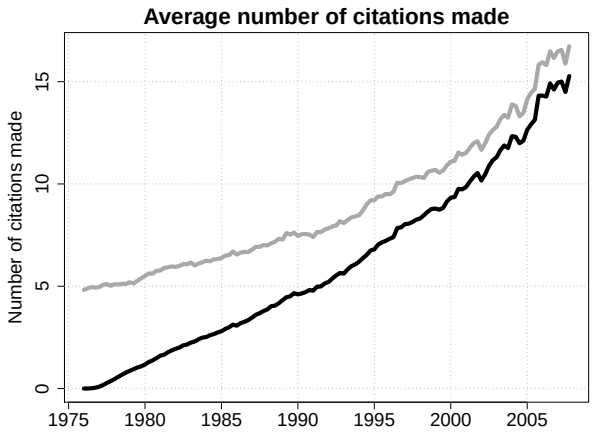


Figure 2: Average number of citations made per quarter; upper curve includes all citations made, lower curve includes only citations made to patents within our dataset.

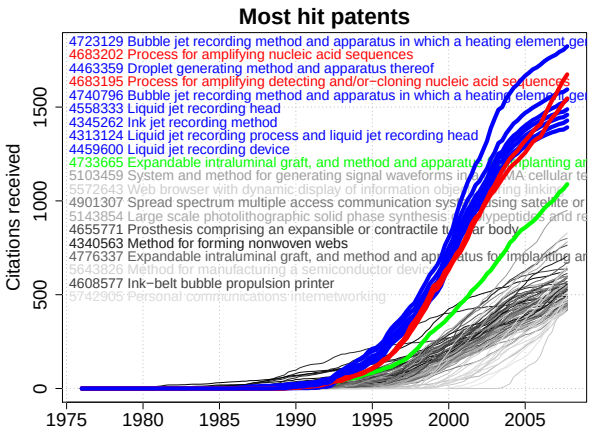


Figure 3: The cumulative number of citations as a function of time. Each curve represents citations accumulated by a particular patent. Only the top 100 patents are shown. Patent numbers and titles are printed in the same color as the corresponding citation curve.

In this study we focus only on a few key pieces of information in the patent record: patent number, title, issue date, IPC code, and references. The patent number serves by design as a unique identifier for each patent and we use it as such.

Each US patent is assigned a handful of IPC codes by the inventor and patent examiners at the USPTO, designed to classify the invention. In this paper we use IPC codes to measure the degree of similarity and dissimilarity between two inventions. The IPC codes are also used to control for differences in citation practices in diverse technical fields.

Each patent record is required by the USPTO to cite all of the previous inventions on which it depends. These citations establish an intention's "prior art" and are compiled by both patent examiners at the USPTO in and the inventor. Figure 2 shows a three-fold rise in the average number of citations each patent makes over the past thirty years. Citations play a pivotal role in our evolutionary analysis of the patent data. We develop a precise formalism for key statistics about citations, and visualize the evolution of technology by highlighting the most heavily cited inventions.

Evolutionary activity

We regard the evolutionary activity of a patent as the cumulative number of times other patents cite it. For patent p, $c^t(p)$ is defined as the set of patents issued at time t that cite p, and C_p^t as the cumulative citations to patent p up to t:

$$C_p^t = \sum_{t'=0}^{t'=t} \sum_{p' \in c^{t'}(p)} f^t(p, p'), \qquad (1)$$

where $f^t(p, p')$ is a counting function, constructed to count contributions of citations to the cumulative sum. The simplest version of a counting function is $f^t(p, p') \equiv 1$, in which case each citation in $c^t(p)$ is counted with equal weight. For this case, C_p^t is illustrated in Figure 3. The counting function $f^t(p, p')$ may be crafted to emphasize or de-emphasize different aspects of the population, as discussed below.

In Figure 3, we overlay the patent number and title for the twenty most heavily cited patents in our dataset. In this and all subsequent plots, we color the citation waves as follows: Top inkjet printing patents are blue, top polymerase chain reaction (PCR) patents are red, and the top stents patent is green. All other patents are colored various shades of gray. We focus on inkjet printing, PCR, and stents because all of the ten most heavily cited patents in Figure 3, by a significant margin, are innovations in one of those three areas of technology. Later in this paper we consider what makes those three technologies so fecund.

The average behavior of C_p^t, obtained by averaging over all patents issued at each new time t is illustrated in Figure 4 (the time resolution is quarterly). Notice that the curves are roughly straight lines, indicating that patents continue to receive citations at roughly the same rate over their life in the database. Notice also that the slopes of the lines increase through the first two decades of in our data and then level off.

Shadow models

In order to determine which aspects of the patent data might be shaped by natural selection, we construct a "shadow patent" system. Shadow patents and real patents exhibit many of the same statistics, by construction. If a real patent is issued, then so is a shadow patent, and if a real patent makes a citation, then so does a shadow patent. Thus, by construction, Figures 1 and 2 are identical for real and

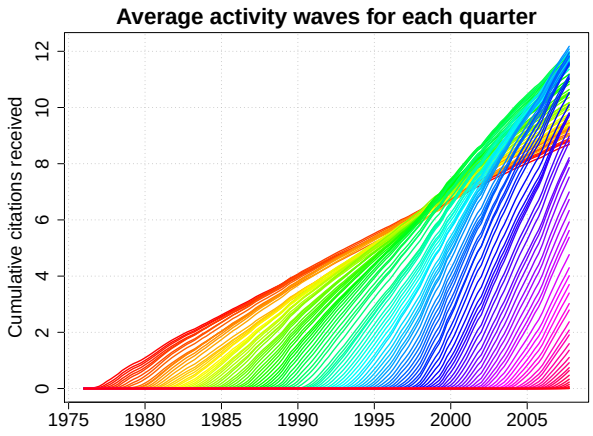


Figure 4: Average number of citations per quarter. Each curve represents the cumulative sum of the citations received of all patents issued in a given quarter.

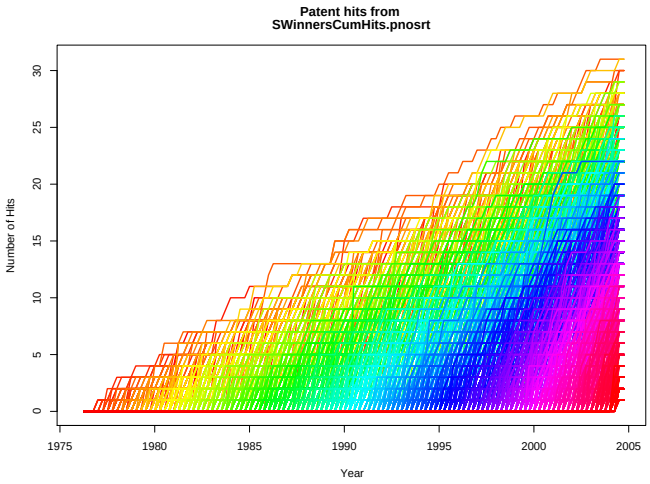


Figure 5: The cumulative number of citations of the most heavily cited patents issued each quarter in a shadow patent model (see text).

shadow patents.

However, the same does not necessarily hold for Figure 3. When shadow patents choose *which* patents to cite, they do so *randomly* and with equal probability from the pool of earlier patents. To test the hypothesis that heavily cited real patents are heavily cited just by chance (given the number of patents being issued and the number of citations being made), we simulate shadow patents and observe typical maximal citation levels. If the most cited real patents have significantly more citations that the most cited shadow patent, then the real citation levels are not statistical fluctuations.

Figure 5 shows the cumulative citations of the most heavily cited shadow patents issued each quarter. Comparison of the y-axis in Figures 3 and 5 shows that heavily cited real patents get orders of magnitude more citations than any shadow patent. We conclude that the striking fecundity of heavily cited patents is no accident. It is not mere noise. Rather, there must be something special about the meaning or content of heavily cited patents that makes them so fecund.

Super star patents

The significant rise of evolutionary activity, measured by raw cumulative citation counts C_p^t, over shadow model activity is itself evidence of the process of Darwinian evolution, driven by selection of the fittest.

Further insight may be gained by examining particular high-fitness patents, to create narratives that may contribute to our intuition about the evolutionary process. Studying the patents in Figure 3 reveals that the most heavily cited patents typically involve one of the following three innovations: inkjet printing, PCR, and stents.

Inkjet printing: The Japanese company, Canon, holds a spate of patents on inkjet printing that have been very heavily cited. Although originally developed for putting ink on paper, the fundamental innovation behind inkjet printing actually involves the ability to extremely precisely position extremely small bits of matter ("ink"). Beside traditional inks, for the original printing applications, the printed materials now also include skin cells (so skin grafts can be printed), DNA or RNA primers (on microarray chips), and metals. Depositing successive layers of materials means that we can print certain arbitrary three dimensional structures. One now reads about inkjet printing technology being used to print batteries, clocks and flexible video screens, among other things.

PCR: Polymerase chain reaction is one of the cornerstones of contemporary biotechnology. Patented (number 4683202) in 1987 by Kary Mullis of Cetus Corporation (one of the first biotech firms), PCR makes it possible to rapidly make millions of copies of an arbitrary DNA sequence. This method has been extensively modified to achieve many different kinds of genetic manipulations. It is now a fundamental tool in a wide range of biotech applications. In 1993 Mullis received the Nobel Prize in Chemistry for his work on PCR.

Stents: Stents are man-made tubes that are used to hold open conduits in the body, such as coronary arteries partially occluded with plaque. In 1986 Julio Palmaz patented a stent that could be expanded within a blood vessel by an inserted angioplasty balloon. This procedure allows some blocked coronary arteries to be repaired without open-heart surgery, allowing much simpler and safer treatment. Citations to this patent indicate that it opened the door to a wide range of minimally invasive blood vessel therapies. Stents have been in the news recently because of patent litigation between

Boston Scientific and Johnson and Johnson, and because of controversy about the merits of drug-coated stents.

Eliminating data biases and artifacts

The definition of evolutionary activity in terms of the raw cumulative citation counts C_p^t as described above may suffer from artifacts in the data that are not related to evolutionary selection of the fittest, which effect evolutionary activity aims to capture. This leads to variations in the definition of activity, obtained by modifying C_p^t to counter these effects through a process of normalization. The canonical way in which C_p^t will be modified is through the definition of the counting function $f^t(p, p')$. We will see how modified counting functions will enable biases and artifacts to be compensated for explicitly. Generally, these modifications may contain a parameter that must be chosen for a certain level of compensation; for this reason these modified counting functions may be regarded as heuristic, rather than fundamental.

A simple example of such an artifact is evident from Figure 2, in which the number of citations grows with time. This leads us to expect that patents issued later would accumulate citations more rapidly than patents issued earlier. Patents are more likely to cite (relatively) recent patents, and over time the number of citations made increases, thus favoring later patents.

A normalization to adjust for this effect uses the counting function

$$f^t_{\text{rate}} = \frac{R^{t'}/N^{t'}}{R^t/N^t}, \tag{2}$$

where N^t is the total number of patents issued at time t, and R^t is the total number of citations made by patents issued at t, and t' is the (arbitrary) baseline time point in the dataset. The total number of citations made must be equal to the total received so $\sum_t \sum_p R_p^t = \sum_t \sum_p C_p^t$. The effect of this normalization is to value all citations in terms of the baseline citation rate, similar to adjusting historical prices for inflation. Because patents at the beginning of the dataset make one third as many citations as those at the end, their citations are given three times as much weight. Then, the adjusted cumulative citation sum, $C^t_{\text{rate }p}$, is computed from equation (1) using $f^t(p, p') \equiv f^t_{\text{rate}}$.

The dynamics of $C^t_{\text{rate }p}$ is illustrated in Figure 6. Notice that this normalization significantly boosts the citation counts for earlier patents, as expected. Notice also that the same ten patents involving inkjet printing, PCR, and stents still occupy the top ten positions in the graph. Thus, although normalizing by prior expected probability of being cited does significantly change which patents are judged to be technology super stars, the narrative of technology evolution being most strongly driven by innovation in inkjet printing, PCR, and stents.

Different IPC classifications are known to have average citation rates that vary by orders of magnitude. These

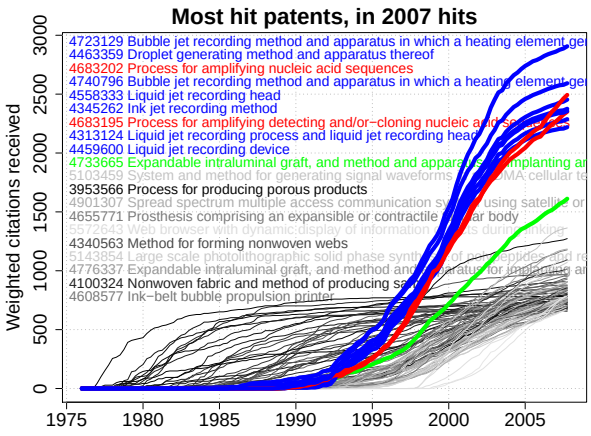


Figure 6: Normalization by relative rate of citation due to changes in the number of citations that are being given over time. Activity is valued in terms of most recent citation rates.

skewed IPC citation distributions might be thought to create further artifacts in our cumulative citation statistics. We can test that hypothesis by introducing a new counting function, f_{IPC}, to normalize by the mean number of citations made by patents in a given category.

The IPC classification of a patent has five levels, $I(p) = (c_1, ..., c_5)$, where each c_i may be thought of as an integer labeling different categories. So, to define the new counting function, we first define the categories of interest to be all possible values of the first two category coordinates, $\mathbf{c} = (c_1, c_2)$. The total number of citations made by patents in the category at time t is

$$R^t_{\mathbf{c}} = \sum_{p' \in p} r(p')\delta(c_1 - I(p')_1)\delta(c_2 - I(p')_2),$$

where $\delta(x) = 1$ if $x = 0$ and 0 otherwise and $r(p')$ is the number of citations made by p'. So we can define f_{IPC} to be a function that depends only on the citing patent:

$$f^t_{\text{IPC}}(p') = \sum_{\mathbf{c}} \frac{R^{t'}_{\mathbf{c}}/N^{t'}_{\mathbf{c}}}{R^t_{\mathbf{c}}/N^t_{\mathbf{c}}}\delta(c_1 - I(p')_1)\delta(c_2 - I(p')_2). \tag{3}$$

E.g., a patent in category A01 issued in 1976 has its outgoing citations doubled in weight because A01 patents issued in 1976 made half as many citations on average as B02 patents from 2007 (chosen as the arbitrary baseline rate). In this way the contributions to evolutionary activity of different categories and different times are equalized.

Figure 7 shows a plot of $C^t_{\text{IPC }p}$, defined by equation (1), with $f^t(p, p') \equiv f^t_{\text{IPC}}(p')$. This figure shows that the skewed IPC citation distribution strongly affects the cumulative citation values. Comparison with Figure 6 shows that the cumulative citations for PCR (red) patents have been significantly raised, while those for inkjet printing (blue) have

been significantly lowered, as have stent patents (green). Nevertheless, those same three narratives still play a dominant role in driving technological innovations.

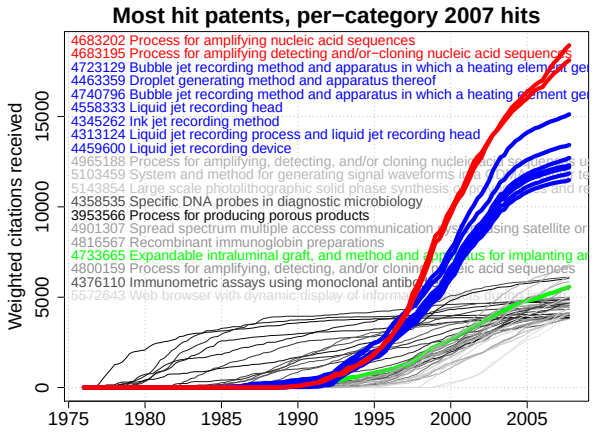


Figure 7: Normalization by mean outgoing citation rate for individual IPC categories (first two levels). This rate varies over time. Contribution to activity is weighted based on the mean number of citations made by patents in that (level 2) category at that time.

Another important effect present in the data is that some patents are cited by subsequent patents that are closely related, and that often have the same assignee. We refer to this as "self-citation" because of the effective redundancy. It is not surprising that citation counts can become inflated due to self-citations; if a company makes an innovation, it is motivated to build on that innovation and to patent further developments. However, this might create an artificially large citation count for some patents that all derive from the same source. A simple normalization to adjust for this effect uses a counting function that discounts self-citations, as follows:

$$f_{\text{self}}(p,p') = \begin{cases} \alpha & \text{if } p \text{ and } p' \text{ have the same assignee} \\ 1 & \text{otherwise} \end{cases}$$

with $\alpha < 1$. Then, the adjusted cumulative citation sum, $C^t_{\text{self}(p,p')}$, is computed from equation (1) using $f^t(p,p') \equiv f_{\text{rate}}(p,p') f_{\text{self}}(p,p')$, where we include normalization with respect to changing mean citation rates, as described above for f^t_{rate}.

Figure 8 shows a plot of $C^t_{\text{self}}(p,p')$ for $\alpha = 0.33$ (other values of α produce similar results). This normalization reshuffles the relative impact of the top patents. One effect is the dramatic drop in inkjet printing patents (blue). Those patents cover inventions developed at Canon, and numerous subsequent Canon patents cite their earlier inventions as prior art. However, relatively few other groups cite Canon's inkjet printing patents. By contrast, the PCR and stent patents virtually unaffected in both relative and absolute terms.

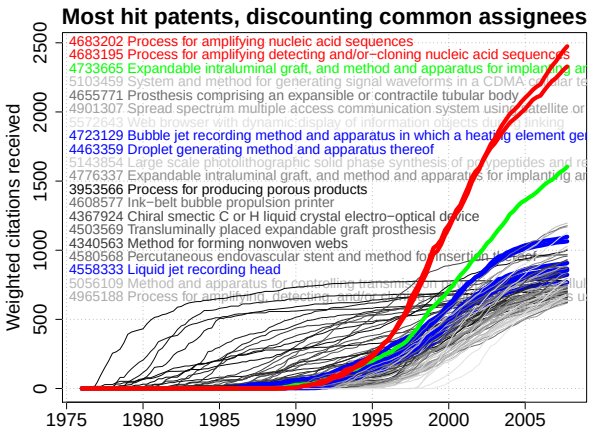


Figure 8: Discounting for self-citations. Notice that the ranking of superstar patents significantly changes, but PCR (red), inkjet printing (blue), and stents (green) remain superstars.

We may combine any or all these normalizations, aiming to obtain the cleanest possible picture of which technologies most strongly drive innovation in the evolution of technology. When we do so, we see that the three top stories (PCR, inkjet printing, and stents) remain dominant among the most fecund technologies. It is striking that, while our efforts to reduce artifacts in cumulative citation counts does significantly change the relative ranking of our stories, the same stories consistently remain significant.

Door-opening innovations

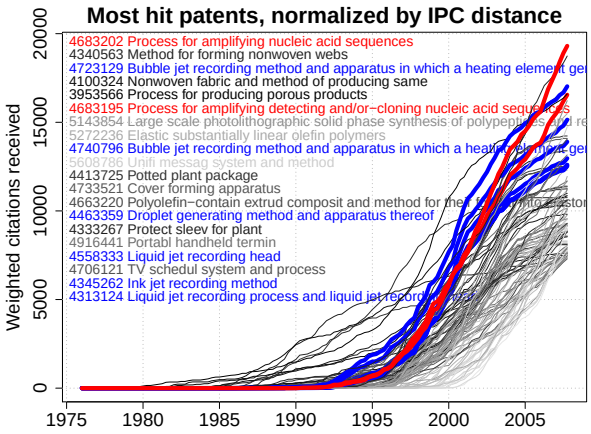


Figure 9: Weighting citation counts by the exponential of IPC distance, so that citations by patents in distant IPC categories count much more. This rewards door-opening innovations and penalizes innovations that merely spur further innovations of the same type.

A crucial aspect of biological evolution seems to be the

ability of biological innovations to "open doors" to entire new universes of innovation, e.g., through the creation of new modes of interaction and new ecological niches, on all scales from molecular to macro-population. Door-opening innovations contrast with inventions that represent "incremental progress," in which new innovations have similar IPC classifications to their ancestors. We may ask if door-opening innovations are important players in the evolution of the patent population.

Our cumulative citation statistics may be modified to address the question of how and whether door-opening patents are present in the dataset, and in particular, whether they are present in the stars that emerge. The modification to address this question takes substantially the same form as the modifications discussed above for eliminating biases and artifacts in the data: define a new counting function that emphasizes, or accentuates the property being investigated. Such use of a counting function is heuristic, in the sense that there is typically not a fundamental formulation, but rather a range of possibilities, corresponding to the testing of a range of different hypotheses.

To formulate the question quantitatively, we use IPC categories to quantify the evolutionary impact of a patent in terms of the breadth of different kinds of patents that cite it. The intuition is that if a patent is cited by patents from very similar IPC categories, then it has relatively narrow impact. By contrast, if a patent is cited by patents in radically different IPC categories, then is has a much broader impact and is opening doors to more kinds of innovations. This intuition may be quantified by weighting the citation count more heavily for more distant IPC categories.

Specifically, if $I(p)$ is the IPC vector $(c_1, ..., c_5)$, with c_1 being the coarsest grain IPC resolution, and c_5 being the finest grain resolution, we define the IPC distance between two patents as

$$d_{\mathrm{IPC}} = 5 - n_{\mathrm{IPC}},$$

where n_{IPC} is the maximum integer such that $I(p)_i = I(p')_i$ for all $i \leq n_{\mathrm{IPC}}$. Then we may create a counting function that weights by this distance, exponentiating it to emphasize the effect:

$$f^{t'}_{\mathrm{IPCd}}(p, p') = 2^{d_{\mathrm{IPC}}}. \quad (4)$$

Now, we can compute $C^t_{\mathrm{IPCd}\ p}$ from equation (1), using $f^t(p, p') \equiv f^t_{\mathrm{IPCd}}(p, p')$.

A plot of $C^t_{\mathrm{IPCd}\ p}$ is shown in Figure 9. Note that PCR and inkjet printing remain significant innovations, indicating that they are all likely to be door-opening innovations. The argument is this: If those inventions were not door-opening but instead represented incremental progress, then weighting by IPC distance would drastically lower their relative citation levels. But instead those patents remain superstars. So, they must be door-opening.

Stents do not appear among the top hundred patents with this weighting. This suggests that while significant, stents are not door-opening to the extent that inkjet printing and PCR are. Intuitively this makes sense, stents are a more specialized type of invention. The difference between stents and the other superstars is also apparent in other normalizations where it trails the other superstars.

Conclusion

Our results show that technology undergoes a Darwinian evolutionary process, analogous to biological evolution. The set of issued patents can be viewed as an evolving population of "organisms" that reproduce when they are cited by later inventions. In the end, we can treat an invention's fecundity (evolutionary activity) as its fitness, for its fecundity directly measures the patent's impact on the composition of future populations.

We interpret cumulative citation count as evolutionary activity, that is, as direct evidence of the dynamics being produced by a Darwinian evolutionary process driven by differential selection. The dramatically high citation counts for the most cited patents show that high fecundity cannot be explained merely as a statistical fluctuation. This comparison with a no-selection null hypothesis embodied in the shadow patents is convincing evidence for Darwinian evolution of technology.

In addition to the population-level conclusion based on cumulative citation rates across the entire population of patents, the conclusion is reinforced by examining individual patents that are "stars," in the sense that they have exceptionally high numbers of citations. The narratives for the star patents are intuitively consistent with the interpretation of the patent population as undergoing Darwinian evolution.

The cumulative citation count on which this conclusion is based can be adjusted, to account for biases inherent in the data. We have discussed various such adjustments, and we find that the evidence for Darwinian evolution is consistently and strongly present over all versions of adjustments we have examined. The decisions for making the adjustments are delicate, and can have a substantial effect on the particular patents that emerge as stars, and on the narratives that accompany them. Some of the difficulties are inherent in the data, e.g., its finiteness, and consequently the absence of citations to the latest patents in the dataset.

Further, heuristic adjustments to our cumulative citation count statistics may be made to emphasize or uncover certain structure in the data. We have used one such adjustment, exponential boost of citations that cross IPC boundaries, to discover which patents appear to be issued for "door-opening" technologies, i.e., those that enable a broad range of further kinds innovations in areas different from the original area the patent was issued in. Applying these statistics largely corroborates the hypothesis that the patent superstars are door-opening technologies.

Acknowledgements

Thanks to Devin Chalmers, Cooper Francis, and Noah Pepper for stimulating discussions about how to quantify the evolution of technology.

References

Bedau, M. A. (2003). Objectifying values in science: A case study. In Machamer, P. and Wolters, G., editors, *Science, Values, and Objectivity*, pages 190–219. University of Pittsburgh Press, Pittsburgh, PA.

Bedau, M. A. and Brown, C. T. (1999). Visualizing evolutionary activity of genotypes. *Artificial Life*, 5:17–35.

Bedau, M. A. and Packard, N. H. (1992). Measurement of evolutionary activity, teleology, and life. In Rasmussen, C. L. C. T. D. F. S., editor, *Artificial Life II*, pages 431–461. Addison-Wesley, Redwood City, CA.

Bedau, M. A., Snyder, E., Brown, C. T., and Packard, N. H. (1997). A comparison of evolutionary activity in artificial evolving systems and in the biosphere. In Husbands, P. and Harvey, I., editors, *Proceedings of the Fourth European Conference on Artificial Life*, pages 125–134. MIT Press, Cambridge, MA.

Bedau, M. A., Snyder, E., and Packard, N. H. (1998). A classification of long-term evolutionary dynamics. In Adami, C., Belew, R., Kitano, H., and Taylor, C., editors, *Artificial Life VI*, pages 228–237. MIT Press, Cambridge, MA.

Benzon, W. (1996). Culture as an evolutionary arena. *Journal of Social & Evolutionary Systems*, 19(4):321–365.

Jablonka, E. (2002). Between development and evolution: How to model cultural change. In Wheeler, M., Ziman, J., and Boden, M. A., editors, *The evolution of cultural entities*, pages 27–41. Oxford University Press, New York.

Jablonka, E. and Lamb, M. J. (2005). *Evolution in four dimensions: genetic, epigenetic, behavioral, and symbolic variation in the history of life*. MIT Press, Cambridge, Mass.

Raven, M. J. and Bedau, M. A. (2003). General framework for evolutionary activity. In Banzhaf, W., Christaller, T., Dittrich, P., Kim, J., and Ziegler, J., editors, *Advances in Artificial Life, 7th European Conference ECAL 2003*, volume 2801 of *Lecture Notes in Artificial Intelligence*, pages 676–685, Berlin. Springer.

Rechtsteiner, A. and Bedau, M. A. (1999). A generic model for quantitative comparison of genotypic evolutionary activity. In D. Floreano, J.-D. Nicoud, F. M., editor, *Advances in Artificial Life*, pages 109–118. Springer.

Skusa, A. and Bedau, M. A. (2002). Towards a comparison of evolutionary creativity in biological and cultural evolution. In Bedau;, R. S. M. A. and Abbass, H. A., editors, *Artificial Life VIII*, pages 233–242. MIT Press, Cambridge, MA.

Sperber, D. (1996). *Explaining culture: a naturalistic approach*. Blackwell, Cambridge, Mass.

High-content words in patent records reflect key innovations in the evolution of technology

Devin Chalmers[1], C. Cooper Francis[1], Noah Pepper[1], and Mark A. Bedau[1,2,3,*]

[1]Center for Advanced Computation, Reed College, Portland, Oregon, USA
[2]FOLSATEC, European School of Molecular Medicine, Milan, Italy
[3]Initiative for Science, Society, and Policy, University of Southern Denmark, Denmark
[*]Contact author: mark.bedau@reed.edu

Abstract

We study the evolution of technology as reflected in the US utility patents granted in the period 1976-2009. Previous work by Skusa and Bedau (2002) and Buchanan et al. (2010) used cumulative citation statistics to identify the inventions that most affect the course of evolution (those with the highest innovative impact). Here we examine the text of patent records (specifically, titles and abstracts) to identify which features are responsible for the high impact on later innovations. We use the TFIDF metric (term frequency times inverse document frequency) to identify which words best convey a patents explicit content. Because a new patent is required to cite all important earlier patents ("prior art") that introduced innovations on which the new patent depends, we use the TFIDF scores of words in citing patents to identify a patent's *emergent content*. A patent's emergent content explains its impact on subsequent inventions; it reflects what traits in an invention actually led to a significant number of subsequent innovations. We illustrate two ways to visualize the explicit and emergent content of patents: word arrays and clouds. Examining the emergent content of populations of patents issued during different epochs reveals when important new ideas appear in the evolution of technology and how they affect its subsequent evolution.

Introduction

This paper presents a method to quantify and visualize certain aspects of the evolution of technology as reflected in patent records. Previous work by Skusa and Bedau (2002) (summarized by Bedau (2003)) used citation statistics to visualize and quantify one specific subset of cultural evolution: the evolution of technology as reflected in patent records. Buchanan et al. (2010) developed and extended this use of patent citations to identify which new inventions over the past three decades have seeded the greatest number of further innovations, termed patent "superstars." They concluded that three of the most important inventions in the past three decades were ink-jet printing, PCR, and stents, and they further showed that many superstar patents are "door-opening" inventions that spawn an especially wide range of further types of innovations.

This previous work highlights the importance of answering the following questions:

1. How can we identify which features characterize the core content of an invention?
2. In particular, which features make superstar patents so successful at spawning future inventions?
3. How have the key features driving technological innovation changed over the past few decades?

This paper aims to answer these three questions.

First, following the approach of Skusa and Bedau (2002) and Buchanan et al. (2010), we use citation statistics to identify how the key inventions driving technological evolution (patent superstars) have changed over the past few decades. To determine the content of these patents, a human can simply examine and interpret its title and abstract, but this process is labor intensive and introduces an element of subjectivity. We want to automate the process and make it objective, but this requires a method for identifying which terms in a document from a corpus especially indicate the distinctive content of that document. The TFIDF metric (term frequency times inverse document frequency, described below) is commonly used for precisely this purpose, so we identify the high-content terms in a patent record as those terms with high TFIDF scores. This method can naturally be generalized to identify high-content sequences of terms, or n-grams.

There is a complication that must be discussed. The high-content terms in a patent tend to reflect what the *inventor* believes are the important features of the invention; below, we term this the invention's *explicit* content. However, the features of an invention that actually play the biggest role in spawning further innovation might not be anticipated by the inventor, so they might not be well reflected in the patent's explicit content. Instead, they might be only *implicitly* reflected in the terms in the patent's title and abstract. Accordingly, to determine what features *actually* are important for an invention's fecundity, we look to the high-content words in the patents that cite the invention; below, we term this the invention's *emergent* content.

The explicit and implicit content of sets of patents can be visualized by two complementary methods: word arrays

and word clouds. By applying these methods to patents from successive epochs, we visualize how the explicit and emergent content of key inventions have changed over time. Our results described below indicate that innovation in the later half of the 1970s was especially active concerning automobile emissions and personal electronics. In the 1980s, the dominant technology drivers shifted to zeolites and semiconductors. The 1990s and 2000s were both dominated by a range of further technologies, especially inkjet printing, PCR, stents, e-commerce, wireless communication, and solid-state storage.

Our work here illustrates how citations and key terms in patent records provide a rich empirical foundation for the study of the evolution of technology. Since technology is one aspect of culture, this work helps illuminate the similarities and differences between cultural and biological evolution. As the papers in Wheeler et al. (2002) indicate, a variety of approaches are being applied to the study of the evolution of culture. The application of the concept of *memes* from Dawkins (1989) is especially hotly disputed, as illustrated by comparison of Sperber (1996), Fracchia and Lewontin (1999), Dennett (2006), and the papers in Aunger (2000). Rather than adding to these polemics, we provide an empirically grounded account of the actual evolution of one important aspect of culture–patented technological innovations–and we develop a method for identifying the key features in inventions that make their impact on new innovations especially big. This line of research might eventually help resolve some of the controversies about cultural evolution, including those about memetics.

The patent record

Patents are granted to inventions only if the patent's examiners are satisfied that the invention is novel, non-obvious, and useful. A patent's novelty is documented by citing the previous patents (and sometimes published papers) on which it depends and builds; these are known as the patent's "prior art." Perko and Narin (1997) and Hall et al. (2005) explain that the patent examiner is the ultimate referee of what patents must be cited, and can add citations that were neglected or omitted on the application.

Our data set consists of records of all the utility patents granted between 1976 and 2009 in the US. (That time window was chosen because of the ready availability of patent data for that period.) In this study, a patent's title and abstract are concatenated to constitute its "record." (A natural generalization of our methods would add further text to a patent's record, such as its claims. Our analysis also uses certain other information about a patent, such as its unique identifying number and, most importantly, the previous patents which it cites–its "prior art.")

Our corpus of 3,630,466 patent records contains 459,232,327 individual word tokens, employing a dictionary of 993,544 word types. Our analysis relies crucially on citations among patents. The patents in our data set bestowed a total of 38,893,014 citations, of which 30,198,227 (about 80%) hit patents in our dataset. Our patents on average cite 10.97 earlier patents and are cited 8.25 times, but 87,695 (2.4%) cite no previous patents.

Our investigation of the evolution of technology is motivated by an analogy with biological evolution. A patented invention is viewed as an organism, and different inventions compete for adoption by users in various niches. The spread of inventions in niches is analogous to the Darwinian process of natural selection (we make no assumptions here about how close that analogy is). When a new patent cites prior art (i.e., earlier patented inventions on which it depends and builds), we consider the earlier patent to have spawned an incipient daughter species.[1] Those inventions that spawn especially many incipient daughter species and so are most heavily cited, are the inventions that drive the course of the evolution of technology.

From patent citations, it is possible to reconstruct the entire phylogeny of the evolving network of patented inventions. The entire set of patent records is analogous to the entire fossil record, except that the patent record is virtually complete and mostly accurate and unambiguous.[2] Accordingly the phylogenies that can be reconstructed are stunningly complete, covering every patent (organism in the population). It would be a biologist's dream to work with empirical phylogenies that are this dense and accurate.

Shadow patents

In order to test whether the citation patterns that we observe in the patent data could have been created by a random process that ignores the content of the patents involved, we construct a system of "shadow" patents. By construction, shadow patents mirror (or "shadow") many aspects of real patents.

The precise mechanism for generating shadow patents is as follows: If p real patents were granted in year y, then p shadow patents are also granted that year. If a particular patent, i, is granted in year y and cites c earlier patents, then the shadow patent, i_s, is also granted in year y and cites c earlier shadow patents. However, whereas a real patent cites its prior art, a shadow patent cites earlier patent chosen at random (with replacement) from the patents cited by real

[1] For simplicity of exposition and when no confusion should result, we will sometimes speak of a patent when we mean to refer to the invention that is patented.

[2] It is worth noting that the patent record is somewhat "dirty." Cleaning the data involves various ad hoc and approximate procedures, and raw data is sometimes corrupted or lost. It should be noted in addition that simple citation metrics can draw an incomplete picture of what is happening in the patent data. We know from Cohen et al. (2000) that patent value, citation rate, patent frequency and citation methodology vary greatly in different industries. This should prompt a salutary dose of skepticism about simplistic sweeping interpretations of citation patterns.

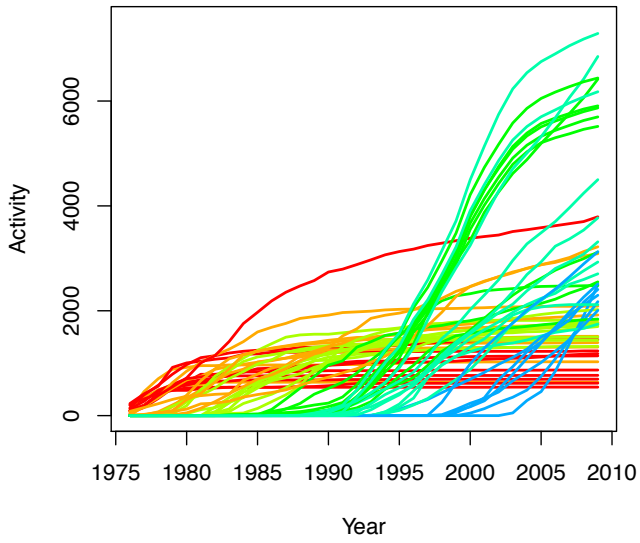


Figure 1: Cumulative citations (or "activity") of the twenty most heavily cited patents from each decade (see Table 2), divided by the prior expected probability of being cited.

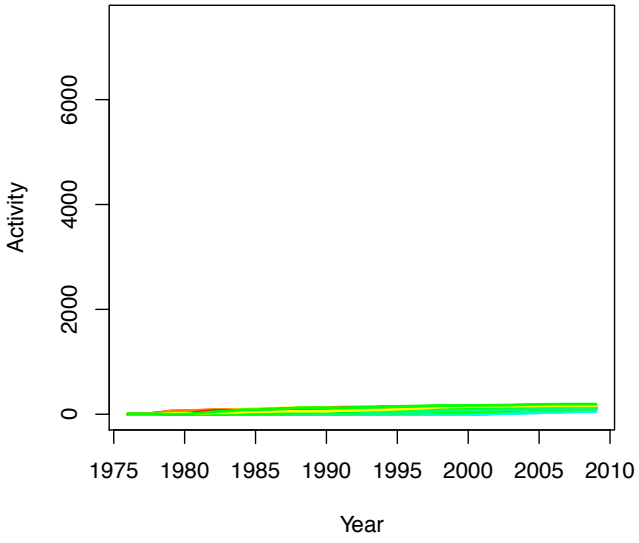


Figure 2: Cumulative citations (or "activity") of the twenty most heavily cited "shadow" patents from each decade. Compare with Figure 1.

patents granted in year y.

The system of shadow patents is a null hypothesis against which we measure whether the citation patterns we observe in real patents could have been created by a random process that ignores the content of the patents.

Highly cited inventions

Following Skusa and Bedau (2002) and Buchanan et al. (2010), we begin by examining the most highly-cited patents, for their high citation counts show that they have an especially great influence on the subsequent evolution of technology. Because of variation in the citation rate and size of the patent corpus each year, we normalize citation counts to make them comparable across epochs, as follows: In a given year, each incoming citation count is divided by the a priori expected probability of a patent being cited at a given time. Assuming that all patents have an equal probability of being cited, this prior probability of being cited at t is calculated as the number of citations given by all the patents issued at t (the number of citations given out) divided by the number of patents issued up to t (the number of patents that could be cited). Exploration of different normalizations is available in Buchanan et al. (2010).

First we examine the twenty patents that received the most citations from all of the patents issued in each of the last few decades. Table 2 describes most of the main innovations covered by those patents. While some heavily cited patents fall outside of the kinds of innovations we list, most do fit in our list. Since our data starts in 1976, relatively few of the citations from the first decade contribute to our analysis.

Figure 1 shows the cumulative citations received by the twenty most heavily cited patents in each decade, colored by the year in which the patent was granted. These cumulative citation counts dramatically illustrate which patents are most influencing the evolution of technology at any given time. Analysis of the patent titles and abstracts reveals that the most "fecund" innovations of the past three decades fall into the following technology sectors: automobile emissions, personal electronics, zeolites, semiconductors, inkjet printing, PCR and stents. This decade-by-decade analysis corroborates and extends the results reported by Buchanan et al. (2010).

Figure 1 can be directly compared with Figure 2, which shows the cumulative citation counts of the most heavily cited *shadow* patents. (Real and shadow patents are normalized identically.) Note that the most heavily cited real patents receive two orders of magnitude more citations than their shadow counterparts. This indicates that heavy citation counts observed in the real patents are not merely an artifact of the numbers of patents giving and receiving citations. Randomly distributed citations would never produce the high citation counts observed for the most fecund inventions.

Many details about the evolution of technology can be read off from Figure 1. For example, the most highly-cited patents in the 1970s (concerning automobile emission and personal electronics) are never cited after the 70s and become dormant (indicated by flat lines). In addition,

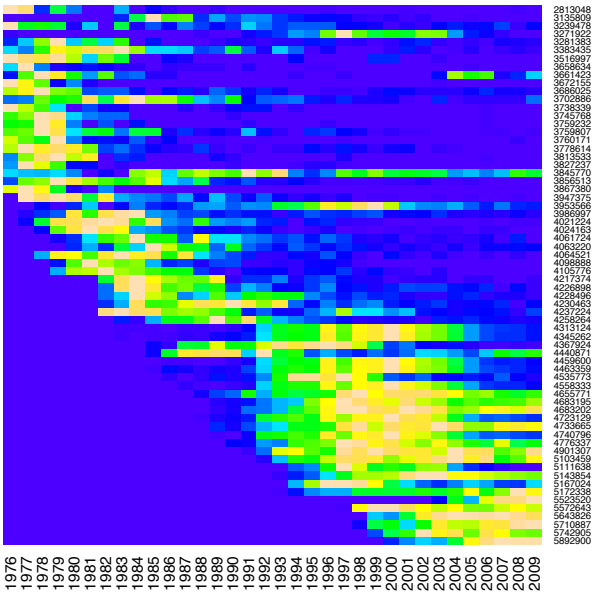


Figure 3: The citation rate for the twenty most heavily cited patents from each decade (see Table 2). Citations are normalized as in Figure 1, and scaled to the interval [0, 1].

one patent (concerning zeolites) is especially heavily cited through most of the 1980s, but its influence subsequently is dominated by a new group of patents (about inkjet printing, PCR, and stents) from the late 1980s, which eventually achieve the highest citation counts overall.

Figure 3 plots the citation rate time series for each of the patents depicted in Figure 1, scaled to the range $[0, 1]$. (Mathematically, this corresponds to the *slope* of the patents shown in Figure 1.) This heatmap shows each patent at each moment, with hotter colors indicating patents that are spawning more new inventions. The heatmap shows that citation rates for most of the most heavily cited patents have cooled off by 2005, and a new crop of patents (about, e.g., genetically modified organisms, e-commerce, and solid-state storage) are heating up today.

The TFIDF measure of high-content words

In this paper, we identify the words that best capture the content of an invention by applying the TFIDF metric to the words in the invention's patent record. TFIDF scores are a standard way to measure the significance of a word in a given document within a corpus, as Spärch Jones (1972) and Salton and McGill (1983) explain. The intuitive idea behind the TFIDF metric is that the most significant words in a document are used frequently within that document, but are not widely used in other documents from the corpus. Accordingly, the measure has two components: term frequency (TF), and inverse document frequency (IDF). Term frequency is just the frequency of a word w in a document d:

$$\text{TF}(w, d) = \frac{|\{w' \in d : w' = w\}|}{|\{w \in d\}|}.$$

The inverse document frequency of a word w in a corpus D is simply the logarithm of the inverse of the fraction of documents in D which contain w:

$$\text{IDF}_D(w) = \log \frac{|D|}{|\{d \in D : w \in d\}|}.$$

Then the TFIDF score for a word w in a document d in a corpus D is just the product of these two measures:

$$\text{TFIDF}_D(w, d) = \text{TF}(w, d) \times \text{IDF}_D(w).$$

To illustrate the TFIDF metric in the patent record, consider the title and abstract of US patent number 4683202 (granted 28 July 1987), which happens to be the most cited patent in the last decade:

> **Process for amplifying nucleic acid sequences**
> The present invention is directed to a process for amplifying any desired specific nucleic acid sequence contained in a nucleic acid or mixture thereof. The process comprises treating separate complementary strands of the nucleic acid with a molar excess of two oligonucleotide primers, and extending the primers to form complementary primer extension products which act as templates for synthesizing the desired nucleic acid sequence. The steps of the reaction may be carried out stepwise or simultaneously and can be repeated as often as desired.

The title and abstract contain 90 word tokens and 56 word types. The most frequent word is 'the', appearing seven times, for a term frequency of TF $= 0.0778$. However, the ubiquitousness of 'the' gives it a very high document frequency within the patent corpus, and so a low inverse document frequency, IDF $= 0.009$, which shrinks its resulting TFIDF score.

The words in the title and abstract of Patent 4683202 with the highest and lowest TFIDF scores appear in Table 1. Note that words with the highest TFIDF scores convey a lot of information about the topic of this patent; for example, 'nucleic', 'acid', 'primers', and 'amplifying' all have high TFIDF scores. By contrast, words with the lowest TFIDF scores ('the', 'and', 'a', ...) convey virtually no information about the patent. Instead, they are so-called "stop words" that reflect grammar and logic rather than content.

The emergent content of patents

The evolution of technology that we study consists of the rise and fall of superstar patents that dominate different epochs. This raises a question: What is the *content* of the

Rank	Term	Count	TF	IDF	TFIDF
1	nucleic	5	0.0556	2.3167	0.1287
2	acid	5	0.0556	1.4203	0.0789
3	primers	2	0.0222	3.2907	0.0731
4	amplifying	2	0.0222	2.6341	0.0585
5	complementary	2	0.0222	2.2645	0.0503
⋮	⋮	⋮	⋮	⋮	⋮
51	in	1	0.0111	0.1187	0.0013
52	is	1	0.0111	0.1151	0.0013
53	of	3	0.0333	0.023	0.0008
54	the	7	0.0778	0.009	0.0007
55	and	2	0.0222	0.0217	0.0005
56	a	3	0.0333	0.0135	0.0004

Table 1: TFIDF values for words in the title or abstract of patent no. 4683202, *Process for amplifying nucleic acid sequences*.

innovations in the superstar patents? Which of their features make them superstars? People can often glean such information by reading superstar patents' titles and abstracts. For example, personal inspection of Table 2 reveals a lot about the content of the most highly cited patents during recent decades. Here we develop methods for determining a patent's content without human intervention. Specifically, we use TFIDF profiles of the words in a patent to measure the patent's content.

We start with some definitions. We write $C(p_1, p_2)$ if patent p_1 cites patent p_2, and we let $\overleftarrow{C}(p)$ be the set of patents that cite p, i.e., p's "incoming" citations:

$$\overleftarrow{C}(p) = \{p' : C(p', p)\}.$$

Then, the number of patents that cite p, or $|\overleftarrow{C}(p)|$, can be used to identify the superstars of a set of patents, or $\text{superstars}_N(P)$, as the N most heavily cited patents in P, ranked by $|\overleftarrow{C}(p)|$.

Let the representative (or high-content) words of a patent p in the patent record $\mathbb{P}$ be the set of words w in the patent with TFIDF above a given threshold, θ:

$$\text{TFIDF}_\theta(p) = \{w \in p : \text{TFIDF}_{\mathbb{P}}(p, w) \geq \theta\}$$

(For this paper, we typically use a threshold of $\theta = 0.05$, which eliminates most stop words and typically picks out just a few words from each patent.)

These concepts easily extend to a set of patents, P. We can identify their citers,

$$\overleftarrow{C}(P) = \bigcup_{p \in P} \overleftarrow{C}(p).$$

and their high-content words,

$$\text{TFIDF}_\theta(P) = \bigcup_{p \in P} \text{TFIDF}_\theta(p).$$

A central hypothesis in our paper is that the high-TFIDF words in a patent, or set of patents, are key to revealing their content. We consider $\text{TFIDF}_\theta(P)$ to be the explicit content of a set of patents, and we consider the *emergent content* of a set of patents, P, to be the high-content words in the set of patents that *cite* patents in P, or

$$\text{TFIDF}_\theta(\overleftarrow{C}(P)).$$

This content is "emergent" because it is implicit; it depends on what subsequent inventions "see" in the inventions in P, and how the inventions function as prior art. Analogously, $\text{TFIDF}_\theta(\overleftarrow{C}(\text{superstars}(P)))$ is the emergent content of the superstars of a set of patents, P. We give examples of both kinds of emergent content below.

Visualizing emergent content with word arrays

The evolution of the emergent content of the patents consists of a list of words with various associated numerical values. A word's value can include such things as the word's TFIDF score, its frequency in the corpus, or the number of patents that contain the word. The evolution of the emergent content in a set of patents can be visualized in various ways, once two things have been determined: (1) Which words contribute to the content? (2) How is the word's numerical value calculated? The visualization methods described here work for any evolving list of words with associated numerical values.

Word arrays are simply lists of words in some fixed, meaningful order, each associated with its numerical value in a given time period. Word arrays are analogous to gene chips, which visualize the expression profile of protein-producing genes. Since word arrays can be represented in one dimension, aligning word arrays from successive snapshots of a population of patent records yields a two-dimensional "movie" of the evolving meaning of a given period of the evolution of technology.

Figure 4 shows the raw time behavior of the emergent content of the superstar patents in Table 2. The words were selected from the citers of the patents in the table. Word frequencies were computed over all of the abstracts of patents issued in each year. For each word, a time vector of values is computed, with each entry $c_{w,t}$ the word w's raw frequency in year t:

$$c_{w,t} = \frac{\sum_{p \in P_t} |\{w' \in p : w' = w\}|}{\sum_{p \in P_t} |\{w \in p\}|}$$

In Figure 4, each word's vector has been scaled to fit the range $[0, 1]$, in order to show each word's rise and fall relative to itself. Figure 4 provides one perspective on the evolving content that is driving innovation in the evolution of technology.

Successive columns in a word array indicate successive moments of time. Figure 4 is like a "film strip" of the evolu-

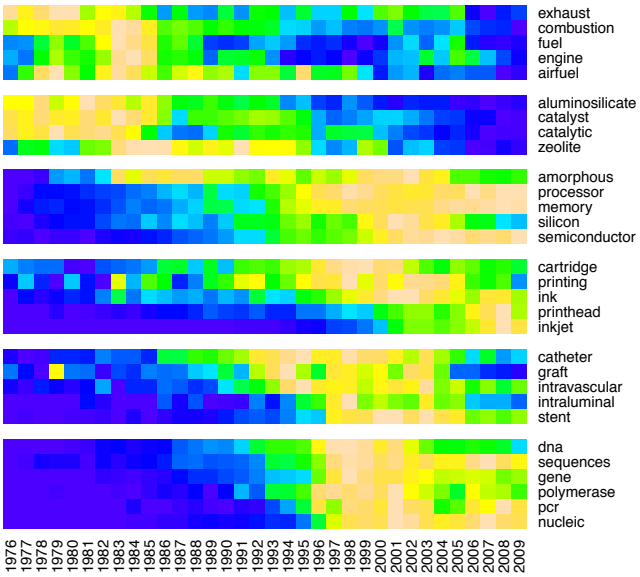


Figure 4: The relative frequency over time of a subset of the emergent content of the top technology patents identified in Table 2. The x-axis is years, and the y-axis is individual high-content words.

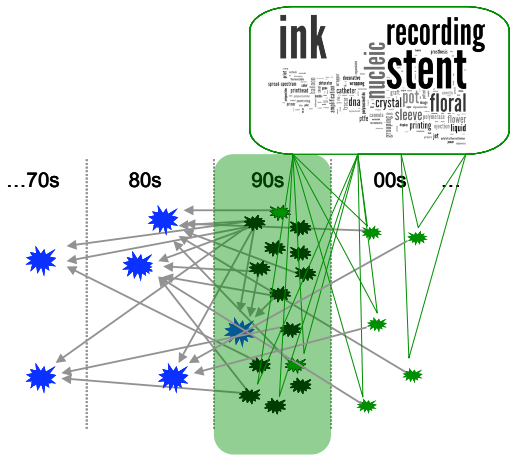


Figure 5: Cartoon sketching the three stages by which word clouds emerge out of a set of patents (e.g, those issued in the 1990s). First, the superstars (blue stars) of the patents issued in the 1990s are identified, then their citers (green stars) are identified, and finally the emergent content of the superstars is identified: $\text{TFIDF}_\theta(\overleftarrow{C}(\text{superstars}(\text{patents}_{1990s}))$. Gray lines are citations between patents.

tion of certain high-impact players in the evolution of technology; each single column is a single frame in the film. It is evident that the main innovation drivers of the 1970s (automobile exhaust and personal computing) are almost completely dormant today. Similarly, the main technology drivers of the 1990s and 2000s (inkjet printing, PCR, stents, and semiconductors) were almost completely dormant for all of the 1970s and 1980s. Furthermore, inspection shows that stents have been cooling off recently, while key components of the PCR and semiconductor genealogies remain very hot.

Visualizing emergent content with word clouds

The word clouds described in this section are another way to visualize how the content of inventions changes over the decades. A word cloud is a two-dimensional agglomeration of the high-content words in some patents, with the words sized according to their numerical value. Since the most important words are the largest, people can easily read the key content in word clouds.

The algorithm for calculating word clouds from a set, P, of patents in a decade has three steps, illustrated in Figure 5:

1. Determine the decade's superstar patents (colored blue in the diagram), superstars(P); these are the patents most heavily cited by the patents issued in the decade.

2. Determine all the patents (green stars) that cite any of the decade's superstars, including patents granted after the decade in question: $\overleftarrow{C}(\text{superstars}(P))$.

3. Identify the emergent content of the superstar patents, $\text{TFIDF}_\theta(\overleftarrow{C}(\text{superstars}(P))$, arrange the words in a cloud,[3] and size each word w by the number of patents in the decade that contain the word: $|\{p \in P : \text{TFIDF}_P(p, w) \geq \theta\}|$.

We illustrate word clouds by focusing on the superstar patents in each decade, and extracting the emergent content of superstars in the familiar way. In this case, we choose to size the words in a word cloud by the number of patents in the corpus that contain the word.

Figure 6 shows the word clouds that emerge from the patents in each decade in our data set: the 1970s (starting with 1976), 1980s, 1990s, and 2000s. Collecting and smoothly connecting these snapshots yields a movie of how the key innovations in patented technology evolve over time.

Conclusions

There are many differences between biological evolution and the evolution of technology, but there are also important similarities. The most important similarity here is the non-randomness or adaptive quality of the key features of the entities that have the greatest impact on new innovations. Comparison with shadow patents confirms that citation rates of the most heavily cited patents would virtually never occur if patents were cited at random and irrespective of their

[3] Word cloud layout algorithm by Jonathan Feinberg, Wordle.net and IBM Research, http://www.wordle.net/credits.

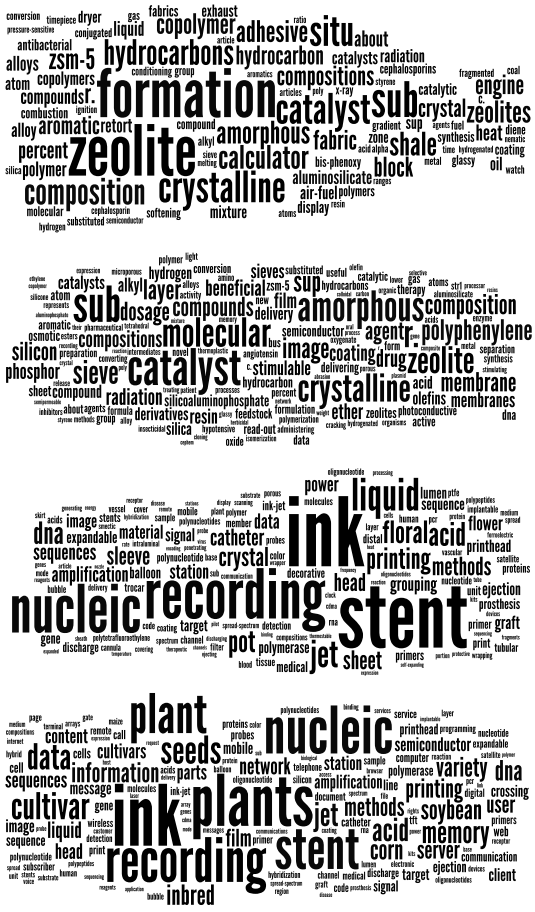


Figure 6: The emergent word clouds for top cited patents in the 1970s, 1980s, 1990s, and 2000s (from top to bottom). The word clouds are still shots from a movie of the evolving meaning of the main technologies driving the evolution of technology.

specific features.

We identify the "emergent" content of sets of patents as the "explicit" content of the patents that cite the patents in the set, measured by high TFIDF scores. We use word arrays and word clouds to visualize the evolution of the key features of patents that have an especially high impact on new innovations. This brings us closer to understanding what makes superstar patents so heavily cited.

Here, the environment that drives adaptation is the technological and economic context of an epoch. If patents and inventions are significantly analogous to biological organisms, then we have created a new way to identify and visualize the emergent semantics of technological evolution through time. Whereas the citation record of patents provides a phylogeny of patented inventions, word arrays and clouds represent the changing emergent content of the drivers of technological innovation through time.

Acknowledgments.

For helpful comments on the topics discussed herein, thanks to Andrew Buchanan, Bobby Gadda, Norman Packard, Emily Parke, and Federico Vaggi, and to the anonymous referees for Artificial Life XII.

References

Aunger, R. (2000). Oxford University Press, Oxford; New York.

Bedau, M. A. (2003). Objectifying values in science: A case study. In P. Machamer, G. W., editor, *Science, Values, and Objectivity*, pages 190–219. University of Pittsburgh Press, Pittsburgh, PA.

Buchanan, A., Packard, N., and Bedau, M. (2010). Adaptive innovative impact on the evolution of technology in the patent record. In *Artificial Life XII (this volume)*. MIT Press, Cambridge, MA.

Cohen, W. M., Nelson, R. R., and Walsh, J. P. (2000). Protecting their intellectual assets: Appropriability conditions and why u.s. manufacturing firms patent (or not). *NBER Working Paper Series*, w7552.

Dawkins, R. (1989). *The selfish gene*. Oxford University Press, Oxford; New York, 2nd edition.

Dennett, D. C. (2006). *Breaking the spell: Religion as a natural phenomenon*. Viking.

Fracchia, J. and Lewontin, R. C. (1999). Does culture evolve? *History and Theory*, 38(4):52–78.

Hall, B. H., Jaffe, A., and Trajtenberg, M. (2005). Market value and patent citations: A first look. *Rand Journal of Economics*.

Perko, J. S. and Narin, F. (1997). The transfer of public science to patented technology: A case study in agricultural science. *Journal of Technology Transfer*, 22: 65–72.

Salton, G. and McGill, M. J. (1983). *Introduction to modern information retrieval*. McGraw-Hill.

Skusa, A. and Bedau, M. A. (2002). Towards a comparison of evolutionary creativity in biological and cultural evolution. In Standish, R., Bedau, M. A., and Abbass, H. A., editors, *Artificial Life VIII*, pages 233–242. MIT Press, Cambridge, MA.

Spärch Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21.

Sperber, D. (1996). *Explaining culture: A naturalistic approach*. Blackwell, Cambridge, Mass.

Wheeler, M., Ziman, J., and Boden, M. A., editors (2002). *The evolution of cultural entities*. Oxford University Press.

Table 2: Major innovations (or technology "superstars") as reflected in citation patterns from each decade.

Selections from the twenty patents that received the most citations from patents issued in 1976-1979	**Citations**
Automobile emissions	
3827237: Method and apparatus for removal of noxious components from the exhaust of internal combustion engines	69
3759232: Method and apparatus to remove polluting components from the exhaust gases of internal combustion engines	44
3745768: Apparatus to control the proportion of air and fuel in the air-fuel mixture of internal combustion engines	44
Personal electronics	
3760171: Programmable calculators having display means and multiple memories	69
3672155: Solid state watch	40
3947375: Liquid crystal materials and devices	39
3813533: Clock calculator	39
Selections from the twenty patents that received the most citations from patents issued in 1980-1989	**Citations**
Zeolites	
3702886: Crystaline zeolite ZSM-5 and method of preparing the same	196
4061724: Crystalline silica	120
4440871: Crystalline silicoaluminophosphates	93
Semiconductors	
3856513: Novel amorphous metals and amorphous metal articles	119
4226898: Amorphous semiconductors equivalent to crystalline semiconductors produced by a glow discharge process	115
4217374: Amorphous semiconductors equivalent to crystalline semiconductors	109
4064521: Semiconductor device having a body of amorphous silicon	108
Selections from the twenty patents that received the most citations from patents issued in 1990-1999	**Citations**
Ink-jet printing	
4723129: Bubble jet recording method and apparatus in which a heating element generates bubbles in a liquid flow path to project droplets	753
4463359: Droplet generating method and apparatus	677
4740796: Bubble jet recording method and apparatus in which a heating element generates bubbles in multiple liquid flow paths to project droplets	663
4558333: Liquid jet recording head	637
4345262: Ink jet recording method	630
4313124: Liquid jet recording process and liquid jet recording head	612
4459600: Liquid jet recording device	599
PCR	
4683195: Process for amplifying, detecting, and/or-cloning nucleic acid sequences	620
4683202: Process for amplifying nucleic acid sequences	597
Stents	
4733665: Expandable intraluminal graft, and method and apparatus for	349
4655771: Prosthesis comprising an expansible or contractile tubular body	277
4776337: Expandable intraluminal graft, and method and apparatus for	268
Selections from the twenty patents that received the most citations from patents issued in 2000-2009	**Citations**
Ink-jet printing	
4723129, 4740796, 4463359, 4558333, 4345262, 4313124, 4459600 (see above)	6518
PCR	
4683202, 4683195 (see above)	2526
E-commerce	
5572643: Web browser with dynamic display of information objects during linking	839
5892900: Systems and methods for secure transaction management and electronic rights protection	770
5710887: Computer system and method for electronic commerce	655
Wireless communication	
5103459: System and method for generating signal waveforms in a CDMA cellular telephone system	802
5742905: Personal communications internetworking	762
4901307: Spread spectrum multiple access communication system using satellite or terrestrial repeaters	665
Solid-state storage	
5643826: Method for manufacturing a semiconductor device	831
5172338: Multi-state EEprom read and write circuits and techniques	629
Stents	
4733665: (see above)	940

Expression of Fashion in Female Preferences for a Mate by Conformity and Differentiation Genes

Atsuko Mutoh, Shohei Kato, Nobuhiro Inuzuka and Hidenori Itoh

Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya 466-8555, Japan.
{atsuko, shohey, inuzuka, itoh}@ics.nitech.ac.jp

Abstract

It is generally thought that living things have desires for conformity as well as desires for differentiation, which makes their preferences show fashion. Recently, it was shown that there were fashion in preferences of how female birds chose their mates. We think fashion in female preferences for a mate is related to their desires and that the strengths of desires among living species are genetically different from one to another. We describe the strength of desires among living species as being artificial agents of genes. In this paper, we simulate the phenomenon of fashion in female preferences for a mate by using an agent model that consists of imported conformity and differentiation as genes. In this experiment, we found that there were two kinds of periodic phenomena of fashion and reported the influence of conformity and differentiation on the transition of female preferences.

Introduction

Fashion expresses the process of the penetration and spread of particular ideas into society. Factors for the generation of different fashions in each era have been attributed to the antithetical desires for conformity and differentiation (Simmel, 1957). In the animal world, many behaviors have been observed that suggest the existence of desires for conformity and differentiation, such as imitation, herd, staking territory, and individual actions.

Generally, fashion is considered to be present in preferences. Until recent years, it was believed that in the animal selection of mates, factors for the evolution of male ornamentation are usually uniform even as time passes; in other words, there was no fashion in the preferences of females. However, the research of Chaine et al. has shown the existence of a species of bird called the Lark Bunting (Calamospiza melanocorys) whose preference of male ornamentation by females change every year (Chaine and Lyon, 2008). However, the reason for this is not understood. We believe that the phenomenon of fashion in preferences, seen in some female birds, contribute to the existence of desires for conformity and differentiation in mate selection. We study this using computational simulation.

Conformity behaviors are behaviors that are similar in one's environment. Conformity behaviors make an organization uniform and establish the majority (Asch, 1951). However, the entire population is not just made up of the majority as a result of conformity behaviors. According to Simmel, fashion is created not just by conformity to others (conforming behaviors), but also by antagonism to exclusive desires, that is, by the desire to differentiate oneself from others (non-conforming behaviors) (Simmel, 1971). It is believed that non-conformity behaviors can preserve diversity, and that conformity behaviors create fashion. Fujii et al. carried out simulation experiments on the effects of conformity and non-conformity behaviors by individuals in a population. The results showed that many non-conforming individuals were needed to create fashion (Fujii et al., 2002).

Until now, we have expressed inborn bodily characteristics as genes, and acquired preferences as memes. We proposed an evolutionary model of artificial life (agents) that combine genes and memes, and observed their influence on changes in preferences concerning mate selection (Mizuno et al., 2005)(Tokuhara et al., 2005). In this paper, we propose a model that adds genes that express strength of desires for conformity and differentiation in order to represent different value systems for agents created in our previous model. By doing so, we can observe computationally mate selection behaviors by agents. We discuss the evolution and expression of fashion by the agents' responses to the environment as generations proceed.

Agent Model

We have described an enhanced Lerena model (Lerena, 2000) in the form of an agent model consisting of both hereditary traits (genes) and acquired traits (memes) (Mizuno et al., 2005)(Tokuhara et al., 2005). This agent model introduces memes into the existing Lerena model. The concept of memes was proposed by R. Dawkins (Dawkins, 1989). He described a meme as both a base factor and a unit of cultural infomation. Our agent model was able to represent constant (i.e., hereditary) and variable(i.e., acquired) information as genes and memes, respectively. In this paper, we describe a new model that reflects the concept of conformity and differentiation.

Agents

An agent a_i consists of the sex sex_i, age age_i, energy $energy_i$, dyad genes $gene_i$ and dyad meme pools $meme\ pools_i$ as follows.

$$a_i(sex_i, age_i, energy_i, gene_i, meme\ pools_i). \quad (1)$$

Genes are hereditary: the first one is for gene traits, and the second relates to preferencs for gene traits. Meme pools are acquired: the first one is for meme traits, and the second relates to preferences for meme traits.

$$gene_i = (g_i^{trait}, g_i^{pref}), \quad (2)$$
$$meme\ pools_i = (m_i^{trait}, m_i^{pref}), \quad (3)$$

where g^{trait} is a gene trait, m^{trait} is a meme trait, and g^{pref} and m^{pref} are preferences for the g^{trait} and m^{trait}, respectively. Preference works to evaluate corresponding traits; for example, g^{pref} means the preference of the g^{trait} in mate choice. The expression of both preferences is limited to females ($sex_i = \mathrm{female}$), and the expression of both traits is limited to males ($sex_i = \mathrm{male}$).

Conformity-desire genes

We add conformity-desire genes to above-mentioned agents as follows.

$$g_i^{trait} = (\mathcal{G}_i^t, \mathcal{G}_i^{tclv}), \quad (4)$$
$$g_i^{pref} = (\mathcal{G}_i^p, \mathcal{G}_i^{pclv}). \quad (5)$$

The $\mathcal{G}_i^t$ is a gene trait, and the $\mathcal{G}_i^p$ is a gene preference. The $\mathcal{G}_i^{tclv}$ and the $\mathcal{G}_i^{pclv}$ are conformity-desire genes of an agent a_i. They have a real-value between 0 and 1. In our model, the nearer a conformity-desire gene value is to 0, the stronger the differentiation desire the agent has. Conversely, the nearer a conformity-desire gene value is to 1, the stronger the conformity desire the agent has. Using Equations (1)-(5), we represent individuals having desires for conformity and differentiation as genes.

Plainness and ornateness

Male and female agents have gene traits and preferences. They consist of bit string data. Since we present the plainness or ornateness of them, we use a $cf(\mathcal{G}_j^t)$ function that counts the number of 1s in the bit string data of a gene trait $\mathcal{G}_j^t$. If $cf()$ of the agent is over half of a bit length, we call the trait and preference of the agent 'trait (preference) a' also known as a ornateness trait and preference. If not, we call the trait and preference of the agent 'trait (preference) b', also known as a plainness trait and preference. This model uses a $cf()$ function to calculate the consumption energy of the agents. The more ornate the agent, the more energy is required to act.

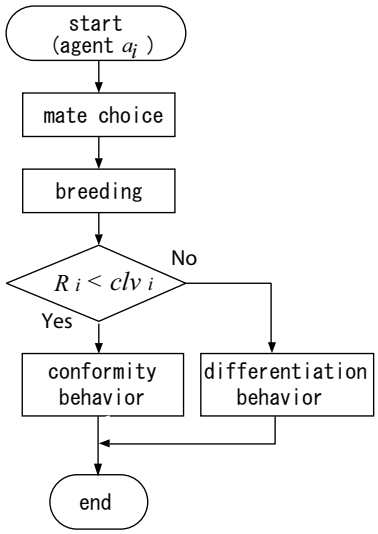


Figure 1: A flowchart of agent actions during each step.

Action

A single run is the repetition of three procedures:

1. mate choice
2. breeding
3. decision between conformity and differentiation behavior
4. conformity or differentiation behavior

A flowchart of agent actions is shown in Figure 1. First, each female selects a male as a mate on the basis of preference. After breeding, each female and male agent selects and performs a conformity or differentiation behavior. These behaviors are operations that rewrite memes. Agents age $1[age]$ during each step. $L_m[age]$ females and $L_f[age]$ males are removed from the population. Agents burn energy by each action and recover after one step. Next, we explain each action.

Mate choice A female a_i selects the best-matched male a_j as a mate from L reference males. The reference population consists of randomly selected L males. The female evaluates a male by calculating the hamming distance between her own preferences and the traits displayed by the male. The evaluation value $P_{i,j}$ for mate choice is determined using agent a_i's gene preference $\mathcal{G}_i^p$ and meme preference m_i^{pref}, and agent a_j's gene trait $\mathcal{G}_j^t$ and meme trait m_j^{trait} as follows.

$$P_{i,j} = w_1\ H(\mathcal{G}_i^p, \mathcal{G}_j^t) + w_2\ H(m_i^{pref}, m_j^{trait}), \quad (6)$$

where $H(A, B)$ is the hamming distance between A and B, and w_1 and w_2 are weight parameters. Agent a_i prefers a_j

to a_k when $P_{i,j} < P_{i,k}$. After choicing a mate, a femele a_i is added to the queue $waiting_j$ for selected male a_j.

Breeding Suppose that female a_i selects male a_j. A new agent a_l is produced as the child of a_i and a_j. This new agent a_l has the following composition.

$$a_l(sex_l, 0, energy_{DV}, ((\mathcal{G}_l^t, \mathcal{G}_l^{tclv}), (\mathcal{G}_l^p, \mathcal{G}_l^{pclv})), (m_{DV}^{trait}, m_{DV}^{pref})), \quad (7)$$

$$(\mathcal{G}_l^t, \mathcal{G}_l^{tclv}) = (mutb(crb(\mathcal{G}_i^t, \mathcal{G}_j^t)), mutr(crr(\mathcal{G}_i^{tclv}, \mathcal{G}_j^{tclv}))), \quad (8)$$

$$(\mathcal{G}_l^p, \mathcal{G}_l^{pclv})) = (mutb(crb(\mathcal{G}_i^p, \mathcal{G}_j^p)), mutr(crr(\mathcal{G}_i^{pclv}, \mathcal{G}_j^{pclv}))), \quad (9)$$

where sex_l is either male or female with an even probability; age_l is zero; $energy_l$ is default $energy_{DV}$; genes $(g_l^{trait} g_l^{pref})$ are determined by genetic operations of Equation (8) and (9); $mutb(A)$ is a mutate-function that reverses each bit of A with probability γ; $mutr(A)$ is a boundary mutate-function for real-value A with a probability γ; $crb(A, B)$ is a cross-function that returns either A and B with an even probability; $crr(A, B)$ is Blend cross-function (Eshelman, 1991) with A and B. In this model, we think that all agents should mature before they are included in the population. Thus, we abbreviate the process by supposing the growth of agents. Memes $(m_l^{trait}, m_l^{pref})$ are not inherited from parents. Thus, their defaults are $(m_{DV}^{trait}, m_{DV}^{pref})$.

By breeding, male a_j uses energy C_j^{crs} as follows.

$$C_j^{crs} = \alpha^{crs}(cf(\mathcal{G}_j^t) + cf(m_j^{trait})) + 1. \quad (10)$$

The more ornate the agent, the more energy is needed to breed. Thus, ornate traits are a disadvantage for childbirth.

A femalea_i is limited to only one round of breeding for each step. On the other hand, a male a_j is not limited. He can breed repeatedly with femeles in the queue $waiting_j$ while their energy is greater than zero.

Decision between conformity and differentiation behaviors An agent decides between conformity and differentiation behaviors after breeding. First, an agent a_i selects M agents of the same sex randomly from a population. Then, an agent a_i perceives the local proliferation rate R_i as follows.

$$R_i = \max(num(a), num(b))/M, \quad (11)$$

where $num(a)$ is the number of agents having a trait (preference) a in M agents. Agents in this model have a trait (preference) a or trait (preference) b as mentioned above. Thus, the range of the local proliferation rate R_i is 0.5 to 1.

As mentioned above, we assume that living species have desires for comformity and differentiation. The proposed model has the following mechanism. If an agent feels that a local proliferation rate is high, he desires differentiation. If not, he desires comformity.

We define the local proliferation rate that an agent considers high as a bifurcation value. The bifurcation value clv_i of an agent a_i is calculated using conformity-desire genes as follows.

$$clv_i = \frac{\mathcal{G}_i^{tclv} + 1}{2}. \quad (0.5 \leq clv_i \leq 1) \quad (12)$$

In addition, Equation (12) is a calculus equation for either male or female agents. An agent a_i decides between conformity and differentiation behavior by using its own bifurcation value clv_i and the perceived local proliferation rate. In particular, if the magnitude relation of their values is $R_i < clv_i$ (i.e., the agent does not feel the local proliferation rate is high), the agent excutes a conformity behavior. On the other hand, if $R_i \geq clv_i$ (i.e., the agent feels the local proliferation rate is high), the agent excutes a differentiation behavior.

Conformity behavior The conformity behavior means that an agent a_i imitates the meme m_k^{trait} of a male a_k who is the most popular as indicated by mate choice. Specifically, the imitation target is the male agent who breeds the most times out of N males selected randomly from a population.

In imitation, an agent a_i can change its own meme m_i^{trait} (m_i^{pref}) by reversing one bit in its bit string data to come close to the meme m_k^{trait} (m_k^{pref}) of target male a_k. By its behavior, the male a_i uses energy C_i^{imt} as follows.

$$C_i^{imt} = \alpha^{imt}(cf(\mathcal{G}_i^t) + cf(m_i^{trait})) + 1. \quad (13)$$

Equation (13) is a calculus equation for either male or female agents. Conformity behaviors are repeated while their energy is over zero, i.e., multiple bits are imitated. On the basis of C_i^{imt}, the more ornate an agent a_i, the larger the energy cost it requires. Thus, the more ornate it is, the smaller the number of bits that can be changed.

Differentiation behavior The differentiation behavior means that an agenta_i imitates reversely the meme m_k^{trait} of a male a_k who is the most popular as indicated by mate choice. Specifically, the reverse-imitation target is the male agent who breeds the most times out of N males selected randomly from a population.

In reverse-imitation, an agent a_i can change its own meme m_i^{trait} (m_i^{pref}) by reversing one bit in its bit string data to back away to the meme m_k^{trait} (m_k^{pref}) of target male a_k. By its behavior, the male a_i uses energy C_i^{crt} as follows.

$$C_i^{crt} = \alpha^{crt}(cf(\mathcal{G}_i^t) + cf(m_i^{trait})) + 1. \quad (14)$$

Equation (14) is a calculus equation for either male or female agents. Differentiation behaviors are repeated while

their energy is over zero, i.e., multiple bits are imitated. On the basis of C_i^{crt}, the more ornate an agent a_i, the larger the energy cost it requires. Thus, the more ornate it is, the smaller the number of bits that can be changed.

Experiments

Next, we explain an experiment with the proposed model, where many male and female agents exist and are evolvable.

Experimental settings

All agents are dead after $5[step]$ (life time). A population of 1000 agents, consisting of 500 females and 500 males, is evolved from an initial state where: (1) the genes $\mathcal{G}^t$ and $\mathcal{G}^p$, the memes m^{trait} and m^{pref} are encoded by bit-strings; the length of these strings is 10 bits each; (2) the initial values of the genes g^{trait} and g^{pref} are given randomly to all agents; (3) the initial values of the memes m^{trait} and m^{pref} are given median ($cf(m_{DV}^{trait}) = cf(m_{DV}^{pref}) = 5$). The parameterization used in these sets of simulation runs is as follows: (1) reference population size for mate choice and conformity and differentiation behavior ($L = N = M = 40$); (2) weight parameters in mate choice ($w_1 = w_2 = 0.5$); (3) initial values of energy ($energy_{DV} = 100$); (4) parameters in costs ($\alpha^{crs} = 3.5$, $\alpha^{imt} = 2.0$, $\alpha^{crt} = 4.0$, $\gamma = 0.005$).

We defined cases with $cf(\mathcal{G}^p) + cf(m^{pref}) > 10$ and $cf(\mathcal{G}^p) + cf(m^{pref}) \leq 10$ as ornate and plain cases, respectively. In this experiment, we examine survival ratio of female agents with ornate and plain preferences.

Results

In our experiment, we set the preference of more than half of the female agents as preference for the majority, and the rest as preference for the minority. We then focused on the turnover between majority and minority. The results of the 10,000-step simulation, run 20 times, showed that turnover between preference for the majority and the minority occurred frequently in all the trials. Figure 2 provides an example of the change in the preference of females that is often seen in the experiment. We could confirm repetition of turnover between the two different preferences of the majority and minority.

Also, Figure 3 shows by generation the average values of the conformity-desire gene of male $\mathcal{G}^{tclv}$ and female $\mathcal{G}^{pclv}$ agents for 20 trials. Whereas the female conformity-desire gene $\mathcal{G}^{pclv}$ did not change in the vicinity of strength 0.5 through 10,000 steps, the male conformity-desire gene $\mathcal{G}^{tclv}$ increased immediately after the start of the experiment, and after 2000 steps, it stabilized between 0.62 and 0.67.

Discussion

Figure 2 shown is similar to periodic phenomena of fashion. In the proposed model, the process by which periodic phenomena of fashion of preference a and preference b is expressed is repeated in the following way:

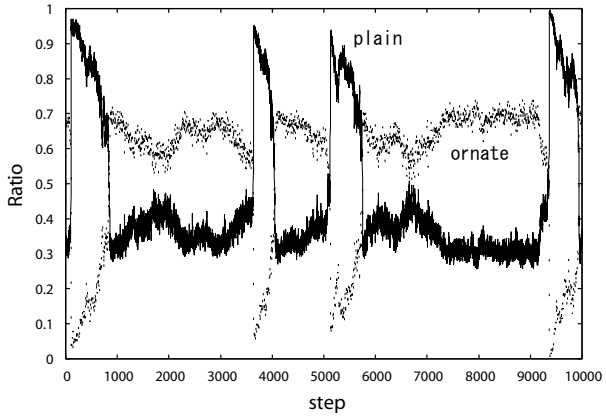


Figure 2: Survival ratio of female agents at each step. A solid line shows a plain preference. A broken line shows a ornate one.

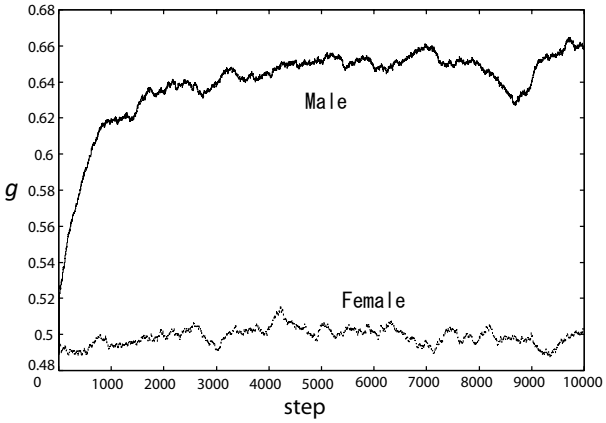


Figure 3: Averag values of the conformity-desire gene of male $\mathcal{G}^{tclv}$ and female $\mathcal{G}^{pclv}$ at each step.

(i) Preference a increases due to conformity behaviors, and it becomes easy for the local proliferation rate of preference a to increase.

(ii) An agent with preference b is created by an agent that takes a differentiation behavior when the local proliferation rate of preference a exceeds the agent's bifurcation value.

(iii) Female with preference b selects male mates with trait b, so preference b increases as a result of females' conformity behaviors in the environment.

In Figure 3, the reason that the male conformity-desire gene is higher in strength compared to females is that in this model, the power to select mates belongs to the females. Because males that copy traits that are popular to females are more easily selected, males with strong differentiation

desires-that is, males with weak conformity-desire gene-are easily selected out. On the other hand, the conformity-desire gene of females do not attain a high level compared to males because 1) male are popular as a result of female conformity behaviors, so there are females born that cannot find a mate, and 2) females have the power to select mates, so they are successful in mating even if they act in a differentiating manner.

From our results, we found that there is a high probability that agents that carry out differentiation behaviors, which trigger the conversion of fashion in process(ii), are females with weak conformity-desire gene.

Furthermore, change was also observed in female agents' preference for plainness or ornateness. As can be seen from the results shown in Figure 2, when preference for plainness becomes the majority over preference for ornateness, plainness' ratio of survival becomes greater, but its duration becomes shorter. When ornate preference becomes the majority over plain preference, its survival does not become great, but its duration is long. This set of phenomena was confirmed in all 20 trials.

The reason for the difference seen in the change of fashion as a result of such change in preferences is believed to be attributable to the difference in the cost of assuming behaviors by agents. For agents with plain preference, the cost of behaviors compared to agents with ornate preference is low, so it is easier to beget progeny and for the number of individuals to increase. Because the local proliferation rate perceived by each agent becomes high, it becomes easier for each agent to assume a differentiation behavior. As a result, it becomes easier for the switching of fashion by differentiation behavior to occur.

On the other hand, the behaviors for ornate preference incur greater cost compared to plain preference, so it is harder to beget progeny and for the number of individuals to increase. The local proliferation rate perceived by each agent does not become high, so it becomes hard for agents to take differentiation behaviors. As a result, the traits and preferences homogenize and stabilize.

The appearance of the sudden increase and decrease of female agents with plain preference confirmed in our experiment approximates a "craze" phenomenon. Also, the appearance of a stable fashion among female agents with ornate preference approximates a "boom" phenomenon.

Compare with the Lark Bunting

According to the report presented by Chaine et al., for the Lark Bunting, whose females have preferences that show traits of fashion, many males with small bodies are successful in mating compared to males with large bodies when the small-body phenotype is in fashion. Furthermore, the duration of the fashion is short. If having a big body is hypothesized to be disadvantageous for survival compared to having a small body (incurring a high cost for behaviors), in our model we can consider a big body as ornate phenotype and a small body as plain phenotype. The phenomena observed in our experiment, namely that survival ratio is high when plain preference is in fashion and this duration is short, and that the survival ratio is low when ornate preference is in fashion and this duration is long, match a part of the fashion phenomena of preference observed in female Lark Bunting.

Effects of the reference population size

Next, we carry out experiments to determine the effects that reference population size M, a parameter inherent in our proposed model for deciding learned behaviors, have on changes in fashion.

Here, we define the change in the survival ratio of agents with target traits (preferences) in the stabilized period (which is the period after 2000 steps that stabilize the conformity-desire gene according to the diagram) of each experiment as either "craze" or "boom".

> "Craze":
> The survival ratio increases from less than 50 percent to more than 90 percent and again drops to less than 50 percent within 1000 steps.
>
> "Boom":
> The survival ratio increases from less than 50 percent to more than 50 percent, and maintains the state of greater than 50 percent for more than 1,500 steps before dropping to less than 50 percent again.

We changed the size of the reference population, M, in the range of $5 \leq M \leq 100$, and studied the number of occurrences of "craze" and "boom" as defined above. Figure 4 shows the average frequency of occurrences of "craze" in plain preference and "boom" in ornate preference over 20 trials.

The results of the experiments showed that when $M = 5$, "craze" and "boom" were almost never observed. However, as the size of the reference population increases, their frequency increases. "Craze" occurred most frequently when $M = 40$, and "boom" occurred most frequently when $M = 20$.

For "craze" to occur, there must be rapid increase of the majority by conformity behaviors and switching between majority and minority due to differentiation behaviors. It is expected that as the reference population size becomes smaller, the average value of the local proliferation rate becomes higher, so differentiation behaviors occur more easily and conformity behaviors occur with more difficulty. Inversely, as the reference population size becomes bigger, the average value of the local proliferation rate becomes lower, so differentiation behaviors occur with more difficulty and conformity behaviors occur more easily. The results of Figure 4 also suggest that the size of the reference population size when deciding on learning actions contribute to the frequency of "craze" and "boom".

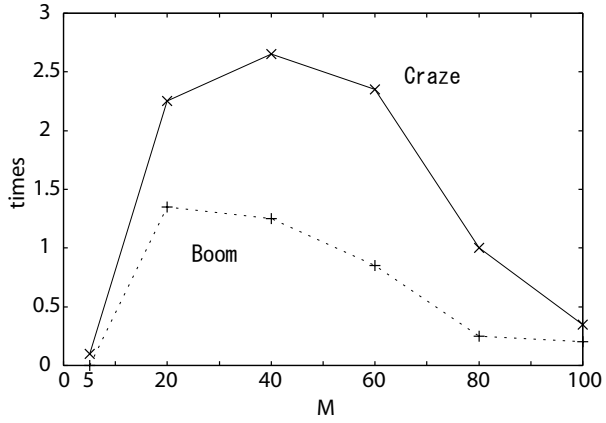


Figure 4: Average frequency of occurrences of "craze" and "boom" at each number of reference populations M. A solid line shows "craze". A broken line shows "boom".

Conclusion

In this paper, we proposed a new model for mate choice involving genes and memes that introduces conformity-desire genes that correspond to the value systems of individual agents. We expressed agents that combine desire for conformity, which is believed to belong to some animals via conformity-desire genes, and desire for differentiation. Furthermore, we created a model that sought to carry out conformity behaviors and differentiation behaviors through conformity-desire genes possessed by agents themselves and the local proliferation rate perceived from the environment. From the results of experiments using our proposed model, we confirmed two types of periodic phenomena of fashion expression. For preferences that incur a high cost for behaviors, a stable "boom" was often observed. For preferences that incur a low cost, a "craze"-like fashion phenomenon, with sudden penetration and then decay, was often observed. We also discovered that the existence of female agents that carry out differentiation behaviors is important for the expression of periodic phenomena of fashion.

From here, it is necessary to match the results of this experiment in detail against real-world animals whose females have preferences that can be seen as fashion, and are the targets of this model. However, we expect that verification of the model will face great difficulty because of the very few case studies of animals whose female have preferences that can be seen as fashion when it comes to mate choice. Therefore, it is desirable to collect data on a variety of fashion phenomena in the real world, including mate choice. Also, mate choice in the real world is not simple like the model. There are a variety of factors involved in propagation, such as the asymmetry in roles between males and females. It is necessary to improve the model based on the findings of this paper so that it better conforms to the real world.

The characteristics of the two types of periodic phenomena of fashion as a result of the difference in cost expressed in our model can be applied to fashion phenomena in general society. For example, because an expensive product cannot be possessed by many people, a moderate degree of differentiation desire is maintained, and a hypothesis can be made that a "craze" will not occur easily. Form the results of last experiments, it is also possible to discuss the relationship between the differences in information space between individuals and the ease with which a "craze" occurs. From here on, we want to extend our proposed model to be a model of general society.

References

Asch, S. (1951). Effects of group pressure upon the modification and distortion of judgements. *Leadership and Men*, pages 177–190.

Chaine, A. and Lyon, B. (2008). Adaptive plasticity in female mate choice dampens sexual selection on male ornaments in the lark bunting. *The Weekly Journal of the American Association for the Advancement of Science*, 319(5862):459–462.

Dawkins, R. (1989). *The Selfish Gene*. Oxford University Press.

Eshelman, L. J. (1991). The chc adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. *Foundations of Genetic Algoritms*, pages 265–283.

Fujii, S., Wang, Z., and Nakamori, Y. (2002). Analysis for fashion emergence by agent-based simulation. *The Second International Workshop on Agent-based Approaches in Economic and Social Complex System*.

Lerena, P. (2000). Sexual preferences: Dimension and complexity. *Proceedings of the Sixth International Conference of The Society for Adaptive Behavior*, pages 395–404.

Mizuno, Y., Mutoh, A., Kato, S., and Itoh, H. (2005). A behavioral model based on meme and qualia for multi-agent social behavior. *19th International Conference on Advanced Information Networking and Applications*, 2:181–184.

Simmel, G. (1957). *Fashion*, volume 62. American Journal of Sociology.

Simmel, G. (1971). *On Individuality and Social Forms*. The University of Chicago Press.

Tokuhara, S., Mutoh, A., Kato, S., and Itoh, H. (2005). A sexual selection model with genes and memes. *Proc. of the 7th International Conference on Artificial Evolution(EA 2005), on CD-ROM*.

Partner Selection: Finding the Right Combination of Players

Pedro Mariano[1] and Luís Correia[2]

[1] Transverse Activity on Intelligent Robotics – IEETA – DETI, Universidade de Aveiro, Portugal
[2]LabMAg – Dep. de Informática, Faculdade de Ciências, Universidade de Lisboa, Portugal
plsm@ua.pt Luis.Correia@di.fc.ul.pt

Abstract

In games that model cooperative dilemmas, if players are able to choose with whom they will play, they will seek out cooperative partners while escaping free riders. In this paper we recast the problem of selecting with whom to play as a problem of finding the right combination of players. With this approach, we present a model suitable to any n-player game. The model is adaptive and we present three update policies. If a player has enough cooperative partners, then with our model a player is able to only select them. We show informal proofs of our claim and illustrate our model under different scenarios.

Introduction

Cooperative dilemmas have been modelled by several games, for instance Iterated Prisoner's Dilemma (IPD), Ultimatum, Investment, Centipede, and Public Good Provision (PGP) (Gintis, 2000b; Fudenberg and Tirole, 1991; Axelrod, 1997). Theoretical analysis of these games predicts the prevalence of free-riders, exploiters, and other types of non-prosocial behaviour (Gintis, 2000b). Despite this, experiments involving people show significant pro-social behaviour. Several theories, trust management, reputation, norms, punishments, have been put forward to explain these results under different forms. However these theories are usually attached to particular games.

In this paper we focus on partner selection. It has been reported in human experiments (Coricelli et al., 2004; Ehrhart and Keser, 1999) that if players are able to select their partners they will seek cooperative partners while escaping free riders. We present a model of partner selection tailored for any n-player game that allows a player to select the most favourable combination of partners. In contrast with previous results, our model relies solely on private information.

The model we present should be used by a player during its life cycle when it has to play a game. The player uses private information gathered from previous games to select partners to play a game. Although with our model a player can in some conditions only select cooperative partners, we do not prevent it from being selected by uncooperative players.

The goal of our model is to allow cooperative players to tentatively select cooperative partners. We assume that a selected player cannot refuse to play and therefore it can be selected by uncooperative players. This situation is not unlike neighbourhood choice, for instance. Someone chooses a neighbourhood for its general reputation but she may not refuse to have any new neighbour no matter how the newcomer is uncooperative.

Related Work

Volunteering is a form of partner selection where a player can choose to participate in a game or not, Aktipis (2004); Hauert et al. (2002); Orbell and Dawes (1993). For each interaction, it introduces the possibility of not playing. However the payoff for not playing lays between the maximum and minimum payoffs obtainable in the game. This relation alters the equilibria in the original game and thus creates new ones. This is the case in Orbell and Dawes (1993) where the payoff for not playing is zero (in their game there are positive and negative payoffs). They justify their choice of this value because people can evaluate and compare game actions that lead to positive or to negative payoffs. The same happens in Hauert et al. (2002). They focused on the PGP game. Players that do not play get a payoff that is higher than the payoff obtained by a defector in a group of defectors but lower than the payoff obtained by a cooperator in a group of cooperators. They found out that their system exhibits a rock-scissors-paper dynamics where players with the option of participating cyclally appear and disappear from the population. In both works players do not have memory of past encounters nor can identify other players. In Price (2006) the author refers that in experiments involving human subjects, people usually cooperate when they can choose their interaction partners, and they cooperate when they perceive altruistic behaviour.

Model Description

In a n-player game, a player has to select $n-1$ partners to play a game from a population of m candidates. Its problem is to find those combinations that yield the highest utilities.

We assume that the player has access to those m candidates, but our model can easily be adapted to a scenario where candidates may enter or leave the population. We assume that the population may contain candidates that behave stochastically, namely, they sometimes cooperate but they also free ride.

For large m and n it may not be feasible for a player to process all the possible combinations. Therefore, a player maintains a pool $\mathbf{c}$ of l combinations that is updated as it plays games. Each combination has a probability to be selected. These probabilities are stored in a vector $\mathbf{w}$. Finally, the player has a utility threshold $u_{_T}$. Representing a strategy by s, a player is then characterised by a 4-tuple:

$$\alpha = (s, \mathbf{c}, \mathbf{w}, u_{_T}) \quad . \tag{1}$$

When a player has to play a game, it selects a combination from vector $\mathbf{c}$ using the probability vector $\mathbf{w}$. It compares the utility obtained with $u_{_T}$ and decides if it should update the two vectors. If it is lower, then other combination should be favoured.

In the following discussion, we will assume that k is the slot index of the selected combination. We will now discuss some vector update policies.

Drastic Update – Policy A

If the selected combination yields a utility lower than $u_{_T}$, its probability is multiplied by a factor, δ, lower than 1.

$$w_k^{t+1} = \begin{cases} \delta w_k^t & \text{if} \quad u < u_{_T} \\ w_k^t & \text{if} \quad u \geq u_{_T} \end{cases} \quad . \tag{2}$$

The probabilities of other combinations are updated as follows:

$$w_i^{t+1} = \begin{cases} w_i^t + \dfrac{(1-\delta) w_k^t}{l-1} & \text{if} \quad u < u_{_T} \\ w_i^t & \text{if} \quad u \geq u_{_T} \end{cases} \quad , \tag{3}$$

where $i \neq k$, in order to maintain sum to unit.

In slot k of vector $\mathbf{c}$ a randomly drawn combination replaces the selected combination in case it yielded a lower utility:

$$c_k^{t+1} = \begin{cases} \mathrm{rnd}(\mathcal{C} \setminus \{c_i^t : 1 \leq i \leq l\}) & \text{if} \quad u < u_{_T} \\ c_k^t & \text{if} \quad u \geq u_{_T} \end{cases} \quad , \tag{4}$$

where $\mathcal{C}$ is the set of all combinations of $n-1$ elements out of m candidates, and rnd is a function that given a set returns a random element.

The initial probability vector, $\mathbf{w}^0$, may have random values or constant value l^{-1}. It has been shown that the choice of $\mathbf{w}^0$ does not change game dynamics (Mariano et al., 2009a). In order to give a fair chance to all initial combinations, we prefer the uniform distribution.

The rationale for the drastic update is that combinations that contain free riders, exploiters, etc., are removed from the pool. It explores new combinations because it is always replacing lower ones. Although the replacing combination has initially a lower probability to be selected, it may absorb the probabilities of other lower combinations. An important aspect is that combinations with only cooperators never leave the pool and absorb the probabilities of lower combinations. This means that in the long run, the probability mass of combinations with cooperators approaches 1.

If there are no good combinations, then the pool will never stabilise, with combinations constantly entering. Their time in the pool will be proportional to their cooperation level.

Smooth Update – Policy B

This update policy has a parameter $\epsilon < 1$ that determines when the combination vector is updated. Whenever a combination yields a utility lower than $u_{_T}$, its probability decreases as it is multiplied by a factor δ lower than 1. If the probability reaches value ϵ we consider that the corresponding combination should leave the pool. It will be replaced by a new randomly generated combination. In order be fair, the new combination is assigned probability l^{-1}. This means that we have to decrease the other combinations' probabilities. We opt for a decrease proportional to their value. Formalising, the probability to select combination c_k is updated as:

$$w_k^{t+1} = \begin{cases} l^{-1} & \text{if} \quad u < u_{_T} \wedge w_k^t \leq \epsilon \\ \delta w_k^t & \text{if} \quad u < u_{_T} \wedge w_k^t > \epsilon \\ w_k^t & \text{if} \quad u \geq u_{_T} \end{cases} \quad , \tag{5}$$

and the probability to select the other combinations is:

$$w_i^{t+1} = \begin{cases} w_i^t \dfrac{1 - l^{-1}}{\sum\limits_{j \neq k} w_j^t} & \text{if } u < u_{_T} \wedge w_k^t \leq \epsilon \\ w_i^t + \dfrac{(1-\delta) w_k^t}{l-1} & \text{if } u < u_{_T} \wedge w_k^t > \epsilon \\ w_i^t & \text{if } u \geq u_{_T} \end{cases} \quad . \tag{6}$$

The combination vector is updated as follows:

$$c_k^{t+1} = \begin{cases} \mathrm{rnd}(\mathcal{C} \setminus \{c_i^t : 1 \leq i \leq l\}) & \text{if } u < u_{_T} \wedge w_k^t \leq \epsilon \\ c_k^t & \text{otherwise} \end{cases} \tag{7}$$

The first probability vector, $\mathbf{w}^0$ is initialised with constant value l^{-1}, in order to give a fair chance to all initial combinations.

As long as the pool size is smaller than the number of good combinations, in the long run, the pool will only contain those combinations. Again, a good combination is never replaced. If the pool size is higher, then bad combinations will always have in the long run a probability of being selected ranging from ϵ to l^{-1}.

Drastic Proportional Update – Policy C

The probability of a good combination is only indirectly increased by the update policies we have described. A better solution is a probability proportional to the utility obtained with the corresponding combination. Even among good combinations there can be differences due to different types of cooperators in the population. For instance, some candidates may behave stochastically in terms of their cooperativeness.

In this policy, vector $\mathbf{w}$ is best described as a weight vector. Whenever a combination is selected, if the utility obtained, u, is higher than threshold $u_{_T}$ its weight is updated in order to approach the true combination utility. If the utility obtained is lower than $u_{_T}$, a random combination is selected and the weight reset to some value.

Like in previous policies, we opt for having an initial weight vector with identical values, $w_k^0 = u_{_T} - \underline{u}$. The decision threshold $u_{_T}$ is used when a new combination enters the pool. Parameter $\delta < 1$ is used to gradually approximate the true utility of a combination. Formalising, the update policy is:

$$w_k^{t+1} = \begin{cases} \delta w_k^t + (1-\delta)(u - \underline{u}) & \text{if} \quad u \geq u_{_T} \\ u_{_T} - \underline{u} & \text{if} \quad u < u_{_T} \end{cases}, \quad (8)$$

where $\underline{u}$ is the lowest utility obtained by a player. The combination vector is updated using the policy described by Equation (4).

This policy is general enough to encompass games with negative utilities. To guarantee this, weights assigned to new combinations are shifted by $\underline{u}$.

As in the previous vector update policies, if the pool size is smaller than the number of good combinations, in the long run the pool will only contain those combinations. Again, a good combination is never replaced. If the pool size is higher, then bad combinations will always have, in the long run, a non-zero probability of being selected, which is less than l^{-1} and higher than:

$$\frac{u_{_T} - \underline{u}}{u_{_T} - \underline{u} + (l-1)(\overline{u} - \underline{u})} \quad (9)$$

which corresponds to the limit probabilities of a pool with $l-1$ perfect combinations. Although this value is inversely proportional to l, if we increase l but other parameters remain constant (in particular number of good combinations), the probability mass of good combinations decreases.

Adaptive Utility Threshold

As the goal of this model is for cooperative players to only select their kin, the ideal value for threshold $u_{_T}$ is the utility obtained by a strategy profile composed of only cooperative strategies. We will use parameter $u_{_P}$ to represent this value. It may happen that a player does not have enough pure cooperative partners. Therefore, no single partner combination will remain forever in vector $\mathbf{c}$. In this case, the player could lower threshold $u_{_T}$ in order to reach a stable regime.

The player should raise the threshold if vector $\mathbf{c}$ is stable. But we must take care in order to guarantee that the threshold does not oscillate too much. We opt for a regime similar to the thermal one used in a Simulated Annealing algorithm (Kirkpatrick et al., 1983).

The rule to update the utility threshold is based on the number of changes that occurred in the combination vector in the last h games. The rationale being that a high number of changes, larger than $h_{_T}$, means that there are not enough cooperative candidates and the threshold should decrease. On the other hand, no changes means that the threshold can increase in order to select better cooperators. The model has additional parameters that control the change in utility threshold, β and γ. The utility threshold update policy is:

$$u_{_T}^{t+1} = \begin{cases} (1-\beta e^{-\gamma t})u_{_T}^t + \beta e^{-\gamma t} u_{_P} & \text{if} \quad \#c = 0 \\ (1-\beta e^{-\gamma t})u_{_T}^t + \beta e^{-\gamma t} \underline{u} & \text{if} \quad \#c > h_{_T} \\ u_{_T}^t & \text{otherwise} \end{cases} \quad (10)$$

where $\#c$ represents the number of changes in the combination vector in the last h games. Parameter $\beta \in [0,1]$ controls the magnitude of change in $u_{_T}$. For $\beta = 0$ there is no change. The value of $\gamma \in [0,1]$ controls the decay of $u_{_T}$ with the number of games. For $\gamma = 0$ there is no decay and for other values we may consider that the threshold stabilises after $10/\gamma$ games.

The initial utility threshold is set to the Pareto utility, $u_{_T}^0 = u_{_P}$. The threshold can never go bellow the lowest utility obtained by a player, $\underline{u}$.

Discussion

We have presented three policies of partner selection suitable for any n-player game with stochastic players. We stress the fact that in the three models a player selects partners based only on private information. This information consists on the utilities obtained in each game. The utility is not necessarily equal to the payoff a game ascribes to a player. It may depend on the payoff of all players, as in the utility of *homo equalitarium* (Gintis, 2000a).

Update policy A is identical to the policy presented in Mariano et al. (2009a). However here we extend that model to select partner combinations instead of a single partner. Moreover we can handle stochastic strategies. Update policy A only requires one combination of good partners while update policies B and C require l combinations of good partners. If there are fewer, then with update policies B and C there will be bad combinations in the pool with non-zero probability. While this is a drawback, update policy B does not promptly remove bad combinations from the pool, but only removes them when their probabilities are lower than threshold ϵ. This allows combinations with stochastic players to remain longer in the pool. As for policy C,

the probability of a partner combination is proportional to their utility, thus the best combinations are favoured over bad ones.

All models aim at keeping the combinations that yield the highest utility in the long run. Despite the computational effort needed by the policies, it is rational for a player to follow one of them instead of randomly selecting partners.

The vector update policy is performed by the player that selects partners, but it can also be performed by players that are selected. In particular, if the combination, from the viewpoint of the partner exists in his pool, then he can apply one of the three update policies. This is an improvement over previous work (Mariano et al., 2009b) as the partner selection model was only used by the player that selected partners. In 2-player games, if the player has enough computational resources its pool can cover the entire population of candidates.

This paper also introduces an adaptive process to modify the utility threshold used in all the policies. The goal of this adaptation is to stabilise the contents of the combination vector while maintaining a higher probability to select the best possible combinations. For instance, if the number of pure cooperators is scarce, a player should accept, as good, combinations with stochastic cooperators, which provide sub-optimal utilities (less than $u_{_P}$). Also, the adaptive process may recover from a situation where the threshold is low and new good candidates appear.

Experimental Analysis

We have performed simulations using the PGP game (Boyd et al., 2003; Hauert et al., 2002). This game is commonly studied to analyse cooperative dilemmas. Moreover, it is a n-player game. We analysed the games played by a particular player paying special attention to the evolution of vectors **w** and **c** and the number of games played with every candidate.

Simulation Description

In the PGP game, a player that contributes to the good, incurs in a cost c. The good is worth g for each player. Let x be the proportion of players that provide the good. The payoff of a player that provides the good is $gx - c$ while players that defect get gx. The game has a single iteration. The strategy used by players is probabilistic and is defined by parameter p which is the probability to provide the good. We assume that the utility of a player is equal to its payoff. In the simulations we set $g = 10$ and $c = 4$. The number of players in a game varied between three and five.

Partner candidate population composition was chosen in order to illustrate interesting behaviour of update policies: with update policy A the population has fewer than $n - 1$ cooperative partners; with update policies B and C the number of combinations with only cooperative partners is less than l. Table 1 lists the number of available partner combinations per population size and players.

		Players		
		3	4	5
Candidates	10	45	120	210
	30	435	4060	27405
	50	1225	19600	230300
	100	4950	161700	3921225

Table 1: Number of available partner combinations for different number of candidates and players in the PGP game.

id	strategies			
P_1	2	$(p = 1)$	8	$(p = 0.5)$
P_2	3	$(p = 1)$	7	$(p = 0.5)$
P_3	4	$(p = 1)$	6	$(p = 0.5)$
P_4	2	$(p = 1)$	18	$(p = 0.5)$
P_5	3	$(p = 1)$	17	$(p = 0.5)$
P_6	4	$(p = 1)$	16	$(p = 0.5)$

Table 2: Candidate populations used in the simulations.

Different hand-tailored partner candidate populations were used. They varied in the number of cooperative strategies and population size. Table 2 presents the candidate populations used. The number of cooperative partners varied between two and four. The rest of the population was filled with mixed strategies that cooperated with probability 0.5. Population size was either ten or twenty. The size of the population of candidates was chosen to reflect the size of small communities (Price, 2006).

Pool size, represented by parameter l, was selected from set $\{10, 20, 30\}$. A higher value means more combinations may be analysed, but there will be more bad combinations in the pool.

The player that was used to analyse the partner selection algorithm used a pure cooperative strategy $(p = 1)$. The player ran the algorithm during $R = 1000$ games. After each game, we measured vectors **w** and **c**, the selected combination, utility threshold $u_{_T}$ and the player payoff.

All probability vector update policies used $\delta = 0.5$. Regarding update policy B extra parameter, ϵ, instead of using an absolute value, in the simulations we used $\epsilon = l^{-1}\epsilon'$, with $\epsilon' \in \{0.2, 1\}$.

Regarding the adaptive utility threshold policy, for update policies A and C a history size of 20 was used. Since update policy B only updates the probability vector when the probability is lower than parameter ϵ different history sizes and values for parameter ϵ were tested in order to observe any relevant behaviour. History size was taken from set $\{20, 40, 60, 80, 100\}$. As for the remaining parameters, we set $\beta = 0.1$, $\gamma = 0.002$ and $h_{_T} = 8$.

To obtain statistically significant results, 30 simulations

were performed for each parameter combination. The appendix describes the implementation of the vector update policies and other relevant details.

Result Analysis

Figure 1 shows the average and standard deviation per each parameter combination of the following values: average payoff, number of changes in combination vector and last utility threshold u_T^R. The key is shown separately in Figures 1a and 1b.

Average payoff is higher with policy A mainly due to bad combinations having a low probability value. Recall that in this policy the probability of good combinations never decrease. This causes bad combinations to have a probability approaching zero. In contrast, policies B and C decrease the probability of combinations (good ones included) when a new combination enters the pool. Therefore, in these two policies, bad combinations will always have a non-zero probability of being selected. Average payoff increases with the number of cooperators in the candidate population while in most parameter combinations it decreases with pool size. The bigger is the number of cooperators the higher is the number of available partner combinations. The bigger is the pool size the higher is the probability to select bad combinations. Average payoff is inversely proportional to candidate population size. The reason being the higher number of uncooperative partners.

As for changes in the probability vector, update policy A has lower values compared with the other update policies. A higher number means that a player takes longer to find a suitable combination of partners. There is not a clear trend on the number of changes versus other parameters: in some settings the number of changes is proportional to pool size. In update policy A in particular, when the number of cooperators is equal to or higher than n, the number of players in a game, there are few changes. There are simulations with candidate population size equal to 20 (results not shown) where the number of changes in **c**, the combination vector, is higher then the corresponding parameter combination but with size equal to 10. The reason being the higher number of uncooperative partners.

The plots of u_T^R, the last utility threshold, show that update policy A has slightly larger values than policy C. In simulations where the number of cooperative partners is equal to $n-1$, the best payoff a cooperative player can get is $g(n-1)/n - c$. This is a reasonable value for u_T as it guarantees a combination of partners where all but one are cooperative. For other values of the number of cooperative partners and number of players, Table 3 presents the best payoff a cooperator can obtain.

The simulations where the number of cooperators in candidate population is equal or higher than $n-1$ are a special case for update policy A. This policy is able to find a combination of only cooperative partners, thus the threshold is nearer $g - c = 6$.

		Players		
		3	4	5
Cooperators	2	$\frac{2g}{3} - c$	$\frac{2g}{4} - c$	$\frac{2g}{5} - c$
	3	$g - c$	$\frac{3g}{4} - c$	$\frac{3g}{5} - c$
	4	$g - c$	$g - c$	$\frac{4g}{5} - c$

Table 3: Best payoff obtained by a cooperative player per number of players and number of cooperators in candidate population.

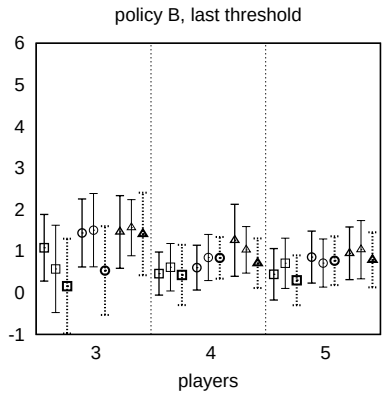


Figure 2: Plot of average and standard deviation of u_T^R from simulations where negative values where observed. Results from simulations with update policy B, $\epsilon' = 1$, population size is 20 and history size is 60.

We comment the results of update policy B separately because of its rule to update the combination vector. Since an update is only triggered when the probability is lower than ϵ, if the probability is very low, then the corresponding combination is selected infrequently. Thus changes in the probability vector are rare. In particular, when history size is 20 and $\epsilon' = 0.2$, no changes occur. Despite this, average payoffs are similar to those obtained by a player that uses update policy C. When we increase history size and use $\epsilon' = 1$ then there are simulations were changes occur, but in a lower quantity when compared to the other policies. As for utility threshold, we observed simulations with negative values (see Figure 2). This is due to a large history size. Let h_s be history size. If there are h_T consecutive rounds with changes in **c**, then in the following $h_s - h_T$ rounds u_T will be decreased towards the minimum utility obtained in a game. Recall that the minimum utility in PGP is $g/n - c \approx -1$ (all players do not cooperate except one). When changes are scarce, the utility threshold remained at u_P.

The plots in Figure 1 only show an inversely relation be-

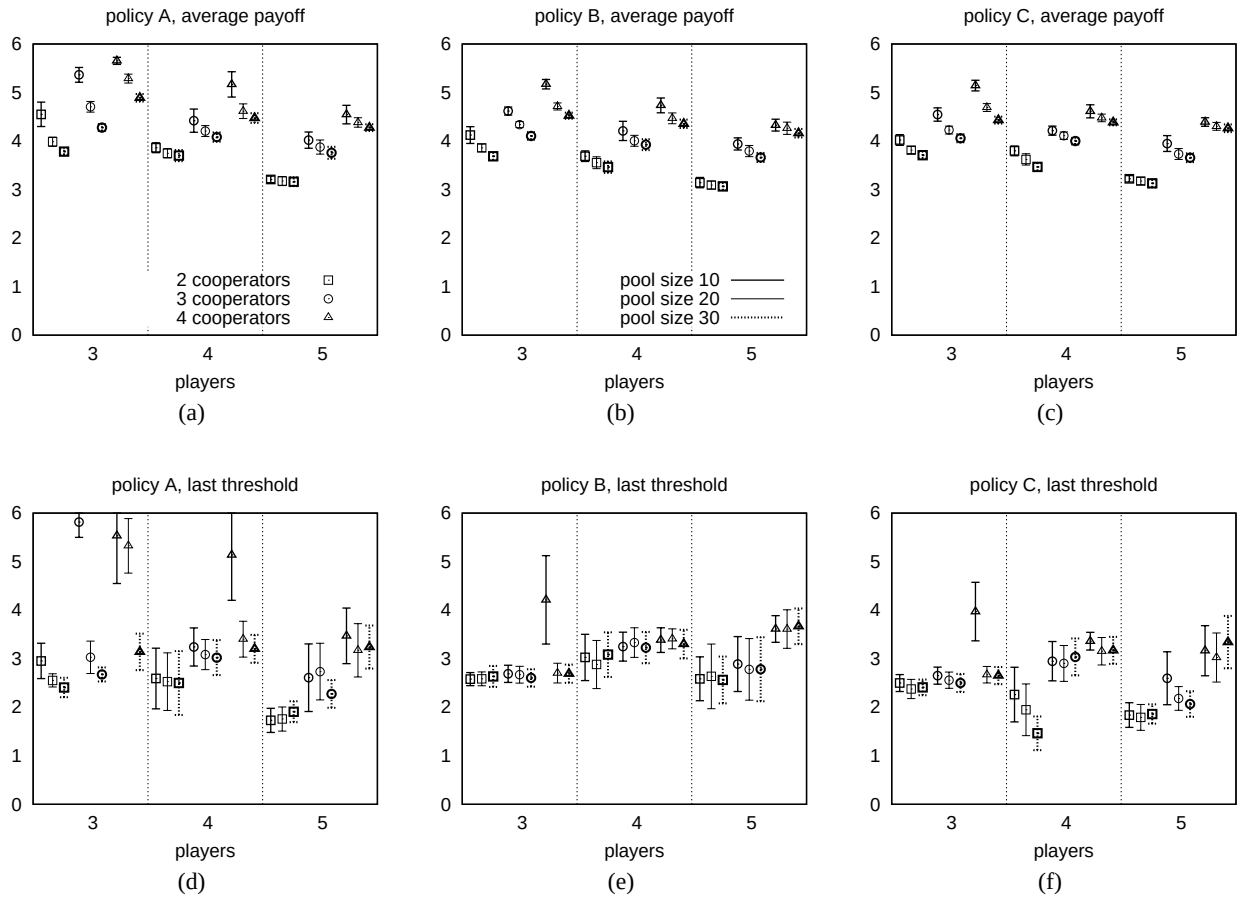


Figure 1: Results from the simulations with population size equal to 10 and history size equal to 20. Plots on the left column are from update policy A, the middle column has plots with update policy B with $\epsilon' = 1$ while the rightmost refers to update policy C. Error lines show the average and standard deviation of, from top to bottom, average utility, number of changes in combination vector, $\mathbf{c}$, and last utility threshold, u_T^{1000}. Due to layout reasons, the key is displayed in Figures 1a and 1b. Line style represents pool size, from left to right: bold solid $l = 10$, mild solid $l = 20$, dotted $l = 30$. Point style represents number of cooperators in candidate population, from left to right: square $\#(p = 1) = 2$, circle 3, triangle 4.

tween average payoff and pool size. In order to search for other relations between parameters, we performed significance tests for the product-moment correlation coefficient at 0.5% level between parameters and measured values. We have found that average payoff is directly proportional to the number of cooperators (in the partner candidate population) and inversely proportional to the number of players in a game (Tables 4a and 4b). As for the number of changes of the combination vector and the last value of the utility threshold, u_T^R, we did not find a clear correlation. However, analysing in more detail, we could see that, for policies A and C, there is an inversely proportional correlation between the average payoff and the pool size. Also, for policies A and C, u_T^R is correlated with the number of cooperators and the number of players. It is directly proportional to the number of cooperators and inversely proportional to the number of players. For most of policy B cases there is no correlation. This can be explained by its use of parameter ϵ. For instance, when ϵ' is 0.2 the chance of a combination being replaced is so low that the utility threshold mostly remains unchanged.

In Table 4d we see the results obtained while maintaining all parameters and varying only the update policy. There is a clear correlation between the policy and changes, average payoff and u_T^R. It indicates that policy B has the worst results and that policy A is the best. Nevertheless we made a deeper comparison between policies A and C (in Tables 4e and 4f). The result observed in Table 4d while still favouring policy A is not so clear. Policy C in a few cases obtains better results and in some more is comparable to A.

Conclusions

We have recast partner selection in n-player games, with stochastic strategies, as a problem of selecting the right combination of players. To support this approach, each player maintains a pool of partner combinations and a probability it associates to each combination. We have presented three policies to update probabilities and to replace player combinations. We have given informal proofs of how a player will only select combinations with cooperative players. One of these policies, A, is able to increasingly select a single good combination, if there is only one. We have also presented a model that updates a threshold for policy replacement used by the three policies. This update aims at adapting a player to situations were there are not enough cooperative partners.

The experimental part focused on the interesting behaviour of a player, which is the situation of not having sufficient cooperative partners. Results show that with the threshold update policy a player was able to select combinations mostly with good cooperators. Results also showed that the threshold converged to a reasonable value.

A drastic update policy, A, is able to obtain better results in most cases. This confirms that the capacity of policy A to increase the probability of selecting a good combination, even if it is the only one in the pool, is a significant advantage for partner selection in n-player games.

As for future work, we aim at improving the selection of partner combination. Instead of randomly picking partners to the new combination, a proportional selection should be done. We plan to assign to each partner a probability of entering a combination.

We are currently investigating the conditions that favour the evolution of partner selection.

As we have said, our model does not prevent a player from begin selected by uncooperative. We also plan to investigate the possibility of refusal. However, this raises the question of the refusal payoff. As we have mentioned some authors chose a payoff higher than the minimum payoff in the original game, thus altering the equilibria in the game.

References

Aktipis, C. A. (2004). Know when to walk away: contingent movement and the evolution of cooperation. *Journal of Theoretical Biology*, 231:249–260.

Axelrod, R., editor (1997). *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration*. Princeton Studies in Complexity. Princeton University Press.

Boyd, R., Gintis, H., Bowles, S., and Richerson, P. J. (2003). The evolution of altruistic punishment. *Proceedings of the National Academy of Sciences*, 100(6):3531–3535.

Coricelli, G., Fehr, D., and Fellner, G. (2004). Partner selection in public goods experiments. *Journal of Conflict Resolution*, 48(3):356–378.

Ehrhart, K.-M. and Keser, C. (1999). Mobility and cooperation: On the run. CIRANO Working Papers 99s-24, CIRANO.

Fudenberg, D. and Tirole, J. (1991). *Game Theory*. MIT Press.

Gintis, H. (2000a). *Game Theory Evolving - A problem-centered introduction to modeling strategic interaction*. Princeton University Press.

Gintis, H. (2000b). Strong reciprocity and human sociality. *Journal of Theoretical Biology*, 206:169–179.

Hauert, C., Monte, S. D., Hofbauer, J., and Sigmund, K. (2002). Volunteering as red queen mechanism for cooperation in public goods games. *Science*, 296:1129–1132.

Kirkpatrick, S., Jr., C. D. G., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Nature*, 220(4598):671–680.

Mariano, P., Correia, L., and Grilo, C. (2009a). How to build the network of contacts. In Lopes, L. S., Lau, N., Mariano, P., and Rocha, L. M., editors, *New Trends in Artificial Intelligence*, pages 65–76. Universidade de Aveiro. 14th Portuguese Conference on Artificial Intelligence, EPIA 2009.

Mariano, P., Correia, L., and Grilo, C. (2009b). Selection of cooperative partners in n-player games. In Kampis, G. and Szathmáry, E., editors, *Advances in Artificial Life*.

Matsumoto, M. and Nishimura, T. (1998). Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*, 8(1):3–30.

	changes	avg payoff	u_τ^R
$+$	12	72	46
$-$	24	0	0
$\times$	36	0	26

(a) Correlation with number of cooperators in partner candidate population

	changes	avg payoff	u_τ^R
$+$	10	0	4
$-$	9	72	40
$\times$	53	0	28

(b) Correlation with number of players in the game.

	changes	avg payoff	u_τ^R
$+$	17	0	1
$-$	10	53	19
$\times$	45	19	52

(c) Correlation with pool size.

	changes	avg payoff	u_τ^R
$+$	54	54	0
$-$	0	0	49
$\times$	0	0	5

(d) Correlation with update policies ordered as B with $\epsilon' = 0.2$, B with $\epsilon' = 1$, C and A.

	changes	avg payoff	u_τ^R
$+$	4	34	24
$-$	18	0	0
$\times$	32	20	30

(e) Correlation with update policies A and C ordered with C first then A.

	changes	avg payoff	u_τ^R
$+$	11	42	31
$-$	23	0	0
$\times$	20	12	23

(f) Correlation with update policies A and C, with significance level 5%.

Table 4: Correlation significance tests at 0.5% level except in 4f. The values represent the number of parameter combinations with $+$ positive, $-$ negative and $\times$ no correlation. All possible parameter combinations were used except for history size fixed at 20.

Orbell, J. M. and Dawes, R. M. (1993). Social welfare, cooperators' advantage, and the option of not playing the game. *American Sociological Review*, 58(6):787–800.

Price, M. (2006). Monitoring, reputation, and "greenbeard" reciprocity in a Shuar work team. *Journal of Organizational Behavior*, 27:201–219.

Appendix

Implementation Details of the Probability Vector Update Policy

The probabilities in vector $\mathbf{w}$ where represented as partial sums of 31 bit integers. The motivation to use integers is due to the fact that floating point division can yield approximate values and thus the sum of the probabilities may not add up to 1. As we used integers, whenever a probability was decreased, the others were incremented by the quotient of the division presented in the policy equations (see for instance Equation (3)). As for the remainder, a random probability was chosen.

The use of partial sums allows a faster algorithm, with time complexity $O(\log l)$, to select a combination to play with. A random integer in the range $[0, 2^{31}]$ was chosen and then a binary search was performed. Although updating the probability vector has time complexity $O(l/2)$, because on average half partial sums must be updated, when the vector converges only selections take place.

As for the pseudo-random number generator, we used an implementation of the Mersenne Twister, a uniform generator with a large period (Matsumoto and Nishimura, 1998).

Language as Autopoiesis : Experimental Approach to Agency in Linguistic Communication

Keisuke Suzuki[1], Ryoko Uno[2], and Takashi Ikegami[3]

[1]Laboratory for Adaptive Intelligence, RIKEN, Brain Science Institute
[2]Institute of Symbiotic Science and Technology, Tokyo University of Agriculture and Technology
[3]The Graduate School of Arts and Sciences, The University of Tokyo
ksk@brain.riken.jp, ryokouno@cc.tuat.ac.jp, ikeg@sacral.c.u-tokyo.ac.jp

Extended Abstract

One of the challenges of artificial life is to implement agency in the creature. This paper is going to argue for the concept of agency existing in linguistic communication. It is usual and normal to see that agency exists outside of language: it is the user of the language who is equipped with agency, and language is not ostensibly related to it. On the other hand, since the invention of the Turing test, it has been an unsolvable question whether agency is a physical property, or something that is attributed from the outside. Here, it is argued that agency emerges in linguistic communication itself. For developing this idea, we have designed a new communication game between two human subjects in order to see how ”agency” is organized in each communication pattern (which is intended to be a proto-language).

Some researchers, most notably Galantucci (2005), already reported evolution of artificial language in human communication necessary to tell some information to others. Here, our focus is not on language as informative tools, but on language itself as goal of communication, in which it has own agency.

We asked 20 subjects (10 pairs) to communicate using an artificial language, where the expressions are the spatial pattern of the triplet in a 3-by-3 bit square. The subjects are allowed to rewrite the pattern alternatively. Here are some examples from our data. (2) is in response to (1):

```
@  @
  @          (1)
@  @
```

```
###
#@         (2)
#  @
```

Each trial consisted of 16 exchanges of pattern messages between two subjects. Then, the subjects were asked to report their intentions behind the sent messages, and their interpretations of the received messages. The pattern of symbol arrays was analyzed mathematically, and the reports linguistically. We especially focused on how topics shifted during the communication.

Our analysis shows that when the Hamming distance between the patterns of symbol arrays was small, the agents tended to report the messages using metaphorical expressions and not in a literal descriptive manner. The report in (3) explains the intention of (1) to use metaphorical expressions, while that in (4) describes the pattern in (2) literally.

(3) Cherry blossoms are beautiful.

(4) Break the circle by movement from top left to bottom right.

It should be noted that in this experiment, the subjects are forced to exchange messages, so the language pattern should be sufficiently attractive to keep the communication going. Once an attractive pattern emerges, the pattern may inherit the characteristic of being attractive, irrespective of the subjects' intentions. The pattern dynamics are, therefore, operationally

closed in the same sense that Luhmann (1986) defined a social system as being autopoietic. This perspective is also found in a simulation model for demonstrating the Luhmann's concept by Dittrich et al. (2003).

We found that when the Hamming distance between successive patterns gets smaller, human subjects tend to use metaphorical expressions in order to overcome the monotonous development of the pattern exchanges. Thus, the emerging pattern dynamics inversely subdued the subjects, which proves that the communication is indeed structurally coupling system.

References

Luhmann, N., (1986). The autopoiesis of social systems. In Felix Geyer and Johannes van der Zouwen editors, *Sociocybernetic Paradoxes. Sage*, pages 172-192 Sage, London.

Dittrich, P., Kron, T., Banzhaf, W. (2003). On the Scalability of Social Order: Modeling the Problem of Double and Multi Contingency Following Luhmann. *Journal of Artificial Societies and Social Simulation* 6(1):

Galantucci, B. (2005). An experimental study of the emergence of human communication systems. *Cognitive Science* 29:737–767,

On the Emergence of Indexical and Symbolic Interpretation in Artificial Creatures, or What is this I Hear?

Angelo Loula[1,2], Ricardo Gudwin[2] and João Queiroz[3*]

[1] Informatics Area, Department of Exact Sciences, State University of Feira de Santana (UEFS), Brazil
[2] Department of Computer Engineering and Industrial Automation, School of Electrical and Computer Engineering, State University of Campinas (UNICAMP), Brazil
[3] Institute of Arts and Design, Federal University of Juiz de Fora (UFJF), Brazil
[*]queirozj@pq.cnpq.br

Abstract

Communication processes rely on the production and interpretation of representations, thus an important issue is to understand what types of representations are involved during the emergence of interpretations. Here we present an experiment to evaluate conditions for the emergence of interpretations of different representation types. To design our experiment, we follow biological inspirations and a theoretical framework of representation processes. Our results show that different interpretations process can emerge depending on the adaptation cost of cognitive traits and on the availability of cognitive shortcuts.

Introduction

The emergence of semiotic competences (morphosintax, grammaticality, semantics, pragmatics) has been studied through various computational perspectives, including embodied robotics, animats, synthetic ethology, and others. Particularly, virtual simulations have been used extensively to model and simulate the emergence of different types of representations (for a review of works, see Nolfi and Mirolli, 2010, Christiansen and Kirby, 2003, Wagner et al. 2003). Here we propose a synthetic experimental protocol to examine the conditions underlying the emergence of two types of representations (symbols and indexes) in a community of artificial creatures able to interact through communication processes. Empirical constraints come from evidences in studies of animal communication as e.g. the minimum brain model for animal communication, proposed by Queiroz & Ribeiro (2002), which provided us biological inspirations to develop our algorithms.

Despite the many works on the emergence of communication in a community of artificial creatures, there are still important open questions that need further exploration. Particularly, based on the fact that representations can be of different types and that communication processes rely on the production and interpretation of representations, an important issue is to understand what types of representations are involved during the emergence of interpretations in a community of artificial creatures.

In the next section, we will briefly review related work on the emergence of communication and representations processes. Then we present the theoretical and empirical constraints that guided our computational model and simulation. Next, we present our ALife experiment and its results, and, finally, we outline our conclusions and point to future perspectives on the study of the emergence of different representation types.

Related work

To illustrate the open issue of understanding the semiotic process of interpretation in communication events, we bring forward two representative works that simulate the emergence of communication in a community of artificial agents.

Floreano and coleagues (2007) studied the evolutionary conditions that might allow the emergence of a reliable communication system in a community of simulated robots, relying on biological motivations on animal communication. The robots could use a visual signal, turning on or off a light ring, to communicate with other robots about the position of food source. They found that if selection acts on group level instead of individual level, or if members of a community are genetically similar, a reliable communication system could emerge. The robots simulated in this experiment were controlled by artificial neural networks, with a direct connection between the input layer and the output layer. Floreano and coleagues did not discuss how was the light signal interpreted by the robots, or what it represented to them, but, from the neural controller architecture, we can infer that any light signal received was directly mapped to a displacement speeed, so the robot blindly reacted to it without relating to what it could represent, until it finally reached the food source itself.

Cangelosi (2001) is one of the few works to actually propose the emergence of different modalities of representations in a experiment on the evolution of communication. In an experiment with artificial creatures in a grid word, Cangelosi (2001) simulated the emergence of communication systems to name edible and poisonous mushrooms. He had also relied on biological motivations to define a food forage goal for the creatures. In typifying communication systems, Cangelosi (2001) distinguished between signals, which have direct relation with world

entities, and symbols, which in addition are related to other symbols, and built two experiments to study the evolution of each type. The simulated creatures were controlled by a 3-layer neural network with an input layer that receive the visual and auditory sensory data, an intermediate layer that joint together sensory data and an output layer that defined movement and the emission of a signal. In his experiments, the neural networks were both evolved and trained in various tasks, and, in the end, a shared communication system emerged, involving signals and symbols, according to Cangelosi. But he did not described how were these signals and symbols interpreted by the creatures, i.e. if a signal heard was first mapped to a mushroom as its referent, and then to an action, or if it was mapped to an action, with a referent being associated with it. Since the intermediate neural layer might develop either solution, it is not possible to infer what could have happened.

Besides Floreano et al (2007) and Cangelosi (2001), other works have studied the emergence of communication traits and the acquisition of vocabulary or language among artificial agents (see Nolfi and Mirolli, 2010, Christiansen and Kirby, 2003, Wagner et al. 2003). But we have not found works that have studied the emergence of different types of interpretations processes and differentiated the interpretation processes that emerged.

Theoretical and Empirical Constraints

Computational models and simulations are based on different tools that are heavily influenced by meta-principles (theoretical constraints) and biological motivations (empirical constraints) in the design of the environment and the morphological definitions of sensors, effectors, cognitive architecture and processes of the conceived systems and scenarios. This theoretical basis influences modeling on different degrees depending on how it constrains the model being built and what decisions it leaves to the experimenter. Depending on the theoretical framework, this allows us to test the various factors influencing semiotic onto-phylogenetic processes, such as the differences between innate and learned communication systems, the adaptive role of compositional languages, the adaptive advantage of symbolic processes, the hypothetic substrate of these processes, the mutual influences between different semiotic competences and low level cognitive tasks (attention, perceptual categorization, motor skill), and the hierarchical presupposition of fundamental kinds of semiotic competences operating on symbol-grounding processes.

Sign-mediated processes, such as the interpretation of representations in communicative contexts, show a remarkable variety. A basic typology (and the most fundamental one) differentiates between iconic, indexical, and symbolic processes. Icons, indexes, and symbols are differentiated on how the sign relates to what it refers to, its object (Peirce 1958; see Ribeiro et al. 2007). They match, respectively, relations of similarity, contiguity, and law between sign and object. Icons are signs that stand for their objects by a similarity or resemblance, no matter if they show any spatio-temporal physical correlation with an existent object. In this case, a sign refers to an object in virtue of a certain quality which is shared between them. Indexes are signs which refer to their objects due to a direct physical connection between them. Since (in this case) the sign should be determined by the object (e.g. by means of a causal relationship) both must exist as actual events. This is an important feature distinguishing iconic from indexical sign-mediated processes. In the other hand, spatio-temporal co-variation is the most characteristic property of indexical processes. Symbols are signs that are related to their object through a determinative relation of law, rule or convention[1]. A symbol becomes a sign of some object merely or mainly by the fact that it is used and understood as such by the interpreter, who establishes this connection.

Communication is a process that occurs among natural systems and as such we can employ empirical evidences on building our synthetic experiment. Animals communicate in various situations, from courtship and dominance to predator warning and food calls (see Hauser, 1997). To further explore the mechanisms behind communication, a minimum brain model can be useful to understand what cognitive resources might be available and process underlining certain behaviors. Queiroz and Ribeiro (2002) described a minimum vertebrate brain for vervet monkeys predator warning vocalization behavior (Seyfarth et al 1980). It was modeled as being composed by three major representational relays or domains: the sensory, the associative and the motor. According to such minimalist design, different first-order sensory representational domains (RD1s) receive unimodal stimuli, which are then associated in a second-order multi-modal representation domain (RD2) so as to elicit symbolic responses to alarm-calls by means of a first-order motor representation domain (RD1m).

The theoretical descriptions and biological evidences described above guided the design of our computer experiment. We were interested in studying the emergence of indexical and symbolic interpretation competences, so, to start of, we needed to specify the requirements for each and also how to recognize each of them in our experiment. Indexical interpretation is a reactive interpretation of signs, such that the interpreter is directed by the sign to recognize its object as something spatio-temporally connected to it, so for our creatures to have this competence, they must be able to reactively respond to sensory stimulus with prompt motor answer. In the minimum brain model, this corresponds to an individual capable of connecting RD1s to RD1m without the need for RD2. But a symbolic interpretation undergoes the mediation of the interpreter to connect the sign to its object, in such a way that a habit (either inborn or acquired) must be present to establish this association. Thus, in symbolic interpretation, RD2 must be present once it is the only domain able to establish connections between different representation modes. Thus, our artificial creatures must be able to receive sensory data, both visual and auditory, in its respective RD1s, that can be connected directly to RD1m, defining motor actions (Type 1 architecture), or connected to RD1m indirectly, through the mediation of RD2, that associates auditory stimulus to visual stimulus acting as a associative

[1] Differently from Cangelosi's (2001) definition of symbol, based on Deacon's approach (1997), Peirce (1958) did not require symbols to be related to each other to be called symbols.

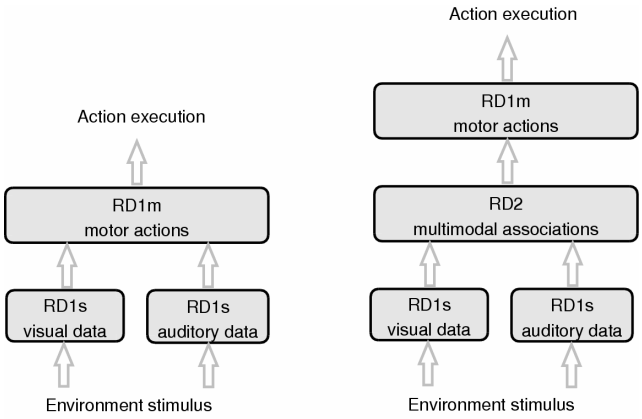


Figure 1: Possible cognitive architectures for representations interpretations. Left: Type 1 architecture, RD1s are connected directly to RD1m. Right: Type 2 architecture, data from visual RD1s and auditory RD1s can be associated in RD2 before connecting to RD1m.

memory module (Type 2 architecture) (figure 1). To evaluate what conditions might elicit each response type – indexical or symbolic –, we implemented these two possible cognitive processing paths as mutually exclusive paths: either the creature responds to auditory events indexically and reactively responds with motor actions, or the creatures responds to auditory events symbolically and associates them with a visual stimulus and responds as if that was really seen. For an external observer, which only watches the information available to the creature and its motor responses, these means changes in the interpretation process.

Building the Experiment

After specifying the brain model requirements and defining the phenomena of interest, we need to set up the scenario where we can test the conditions for both semiotic processes to emerge. To do so, we rely on the empirical evidences of animals vocalizing for food quality, recruiting other group members to feed, and so we designed an experiment where creatures are selected by artificial evolution for their foraging success. Lower quality resources are scattered throughout the environment and a single location receives highest quality resources. One creature (vocalizer) is placed fixed in this high quality resource position, vocalizing a sign continuously. At start, the other creatures (interpreters) do not know how to respond appropriately to sensory inputs and neither recognize the sign vocalized as a sign. But an evolutionary process of variation and selection is applied, with the hope to evolve individuals to better accomplish the task of food foraging. During the evolutionary process, for each start-up conditions, we observe the emergence of indexical or symbolic interpretation for the vocalizations.

The environment is a 50 by 50 grid world (figure 2) and there are 20 positions with only one resource unit each. There is also one position with 500 resource units, where an immovable vocalizer creature is also placed. The vocalizer's sole behavior is to produce a fixed vocal sign, reproduced at every instant. Fifty interpreter creatures are randomly placed in this grid and are capable to visually sense food up to a distance of 4 cells and auditory sense vocalizations up to a distance of 25 cells. This sensory range difference models an environment where vision is limited by the presence of other elements such as vegetation, restraining far vision such as in a open field. The creatures can either see a resource and its position (ahead, left, right, back) or hear a vocalization and its position, if any is within range. Interpreter creatures have a limited repertoire of action: move forward, turn left, turn right, collect resource, or do nothing; and are controlled by (genetically based) Mealy finite state machines (FSM), with up to 4 states (see figure 3). An FSM was chosen as the control architecture because it is quite simple and direct to analyze how it is functioning, permitting direct identification of the processes underlying the creatures' cognition. The creatures always respond to visual inputs with one of the motor actions, and can also respond to auditory input with a direct motor action (a reactive, indexical process) (Type 1 architecture). Alternatively, they can also choose to establish an internal association between the heard stimulus and the visual representation domain (Type 2 architecture). This internal association links what is heard with the view of a collectible resource, i.e. the creature can interpret the sign heard as a resource and act as if the resource was seen. Additionally, the creature may also ignore the sign heard, interpreting it as nothing and acting as if no sensory data was received.

At start, creatures are controlled by randomly constructed FSMs, and are allowed to live for 100 iterations for a trial, trying to collect resources. Artificial evolution selects individuals for their foraging success (number of resources collected in all trials). The 10 best individuals, i.e. the 10 individuals that collected the most resources, are allowed to

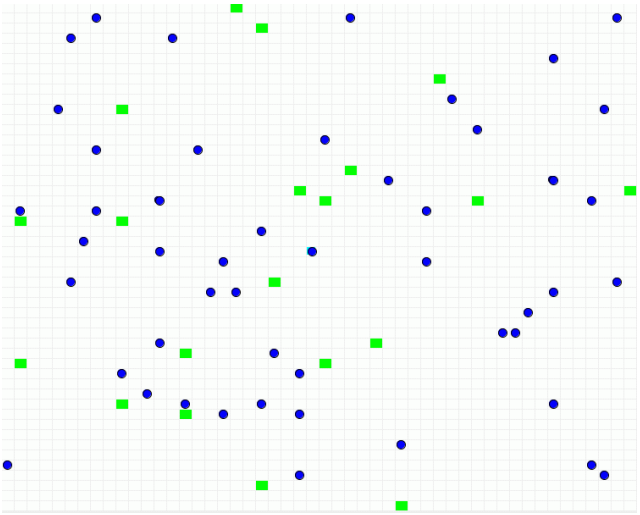


Figure 2: The grid environment. Creatures are blue circles, low quality resource positions are in green cells, and high quality one in the cyan cell in the center.

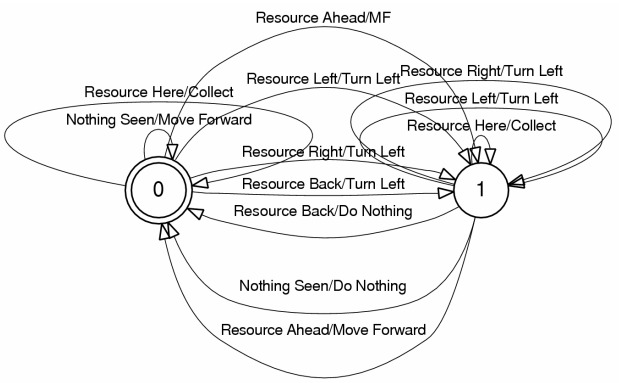


Figure 3: An example of a FSM that controls the creatures. The circles are states and a double circle marks the start state. Arches represent transitions and are labeled according to the sensory event and the action to take over when that event occurs.

breed and make up the next generation. These 10 individuals are copied to the next population and the 40 remaining individuals are a product of mutations (including a cognitive architecture type mutation) and crossovers of the FSMs of the best individuals.

The mutations can be of changing an action for a sensory event in a state, changing the next state after a transition, changing the start state and add or remove a state. The number of mutations is selected from a Poison probability distribution with an expected value of 2. The crossover exchanges states and transitions originating from the selected states between two FSM in a uniform way. All FSM undergo a correction process to fix error that might occur during these operations, such as a transition pointing to a non-existing state.

Every generation undergoes the 10 trials for 500 generations, but, in the first 200 generations (cycle 1), the vocalizer creature is not present and interpreters do not have an auditory sensor, but in the 300 subsequent generations the vocalizer creature is present and interpreters are able to hear (cycle 2). At the start of cycle 2, all creatures are set to ignore the vocalizations, as if it was not relevant, but a small mutation probability is set for changing the kind of response to vocalizations which can be of reacting to them by moving to or to link it with the view of a resource. This corresponds to a change to a Type 1 cognitive architecture (indexical) or to a Type 2 cognitive architecture (symbolic). Besides the probability of going from Type 1 architecture to Type 2 architecture is lower than the other way around, to simulate the fact that such a significant cognitive change is not that easy to happen.

We are interested in observing the overall adaptation process to the foraging task, and are specially focused on the interpretation process, related to the cognitive architecture type, that might result.

Results

To evaluate conditions that might conduct to either an indexical interpretation or to a symbolic interpretation of vocalizations (or even no interpretation at all), we first ran the experiment as described above and observed the evolutionary process and its final result, to see what kind of vocalization response and what type of cognitive architecture would prevail and consequently what type of interpretation process would be chosen. In figure 4, we present the fitness of the best individual, the mean fitness of the 10 best individuals and the mean fitness for the population. In just a few generations, best individuals where able to collect more than 200 resource items and then their foraging success oscillates around 300 items until the end of cycle 1.

Checking the FSM controlling the creatures, by generation 50, they can almost correctly respond to the view of a resource: if it is ahead, move forward, if in the left side, turn left, if at resource, collect resource, but still with bad responses when resource is at right side or at back. And when nothing is seen, they move forward. The oscillations in amount of items collected are due to the random start position of individuals.

At the end of cycle 1, at generation 200, the best individual responds properly to the view of resource, but maybe not optimally. This individual responds to the view of resource in the right with a turn to left, but since it also responds to the view of resource with a turn to the left, the final behavior allows the creature to go in the direction of the resource. If a resource shows up at right it turns left, and then the resource is at its back, so it turns left again, and the resource ends up at the left side now and it turns left once more and then moves forward to collect the resource.

After generation 200, cycle 2 starts, and a vocalizer is placed in the high quality resource position, emitting continually a vocal call. At first all creatures are set to ignore anything heard, so they interpret this as nothing at all. We can observe from figure 3 that the population evaluation rapidly increases and, in generation 210, the best individual reached an amount of resources collected around 800. The individuals adapted fast to the presence of new information in the environment, that enabled them to more easily locate the high quality resource position. The evaluation of the best individual also oscillates much less compared to cycle 1. This is because the start position does not affect as much the individual ability to find the high quality resource position, once the hearing sensor has a much greater range then the visual sensor. But we are interested particularly in the type of response the individual has to vocalizations, whether it was an indexical interpretation, a symbolic interpretation or interpreted as nothing. Figure 5 exhibits the type of response the individuals had along the generations.

In cycle 1, the vocalizer is not present and individuals are not able to hear. But in cycle 2, their hearing sensor is functional and hearing stimulus are received, but all individuals start with a default behavior of ignoring data coming from the hearing sensor and act as if no sensory data is available. In a short period, alternative responses to hearing a vocalization appear in the population, and by generation 205, the population is equally split with all three kinds of response: *indexical* response, *symbolical* response and *ignore*

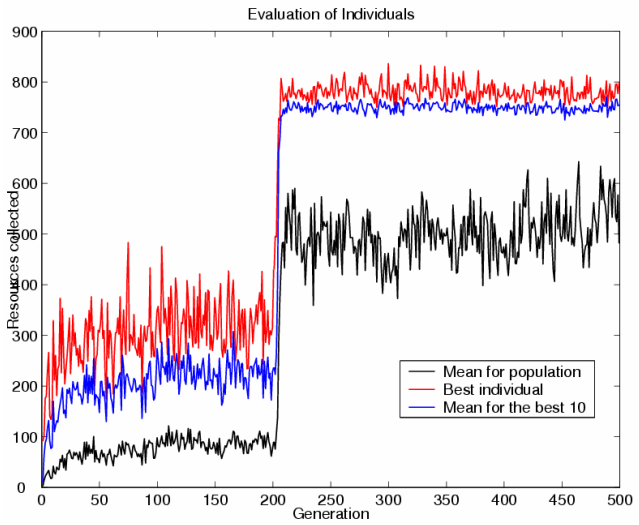


Figure 4: Evaluation of individuals along the generations for the first experiment.

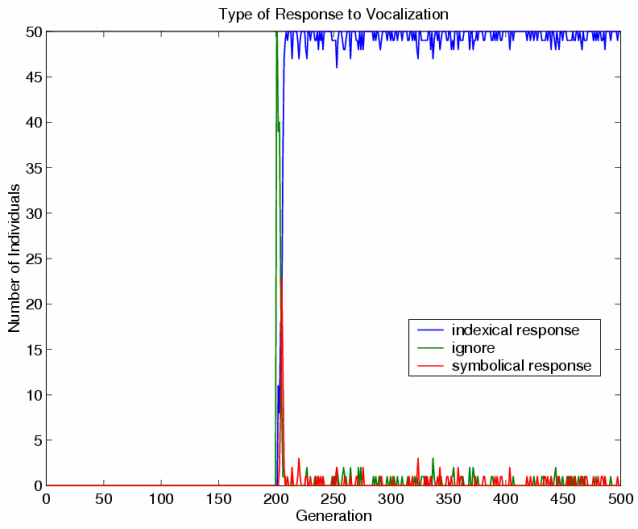


Figure 5: Response type of individuals along the generations for the first experiment.

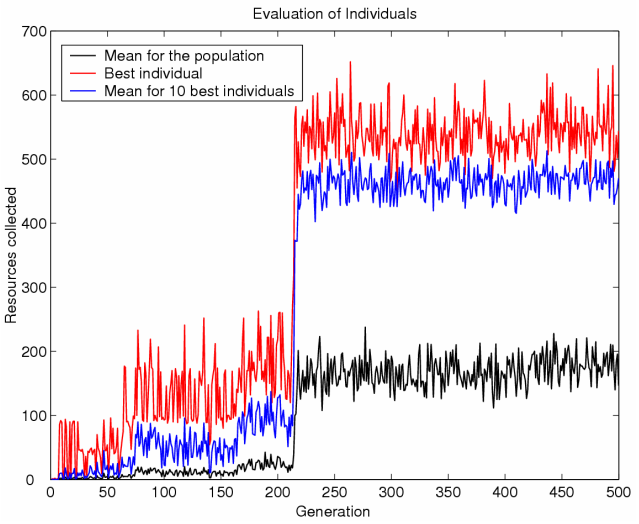


Figure 6: Evaluation of individuals along the generations for the second experiment.

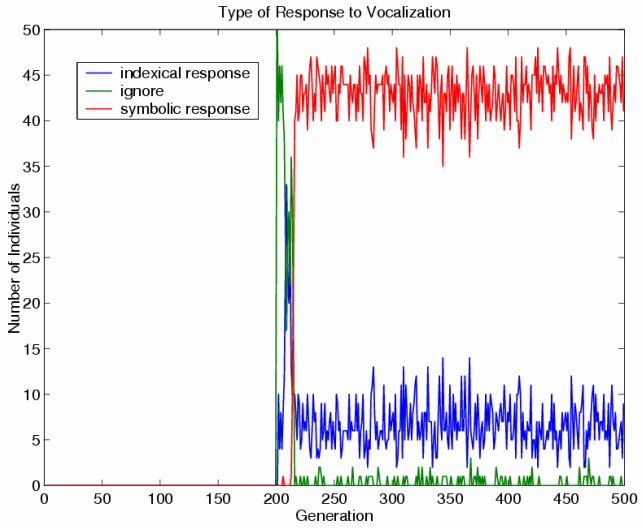


Figure 7: Response type of individuals along the generations for the second experiment.

response. This means, first, that the *ignore* response has severely declined, and, second, that the other two are rising but tied. In a closer look at generation 205, we can see that the best individual is one that responds indexically and collected 728 resource units, and the best individual with symbolic response collected 691 items. However, the mutation operator that changes a Type 1 cognitive architecture (indexical) to a Type 2 cognitive architecture (symbolic) has a quite low probability of happening, and once learning to coordinate sensory data with correct moves is an easy process in this context, as we can see from the fast adaptation in cycle 1, and, moreover, moving from Type 2 architecture to Type 1 is more probable than the other way around, adaptations involving *indexical* response stabilize faster and take over all individuals, exactly what happened after generation 210.

To further test our computational model, we started a new set up for our experiment, where actions coordination in RD1m would be harder to acquire. For that, we impose a restriction that before any movement (moving forward and turning), the creature had to 'prepare' itself by having a null action (do nothing response). To appropriately coordinate its actions then, the creature must use its internal states (finite states machines are capable of dealing with internal states), to 'remember' whether a preparatory action was taken to then take. This makes the task of coordinating sense data and appropriate actions harder.

After simulating these conditions, it can be noticed that it took longer, in cycle 1, for the creatures to evolve an adequate behavior to collect food. By generation 50, for example, the best individual was still not able to move itself around when

no resource was seen, it was only able to collect a resource when it was placed in front, to the left, or exactly at a resource position. Only after generation 160, the creatures started to move forward when no resource was seen, instead of staying still when nothing is seen. Comparing to the previous experiment, this new challenge considerably required more effort for adaptation. The amount of resources collected by the creatures is also lower then in the previous experiments, due to fact that they spend a lot of the iterations 'preparing' movements (figure 6).

After cycle 2 starts in this second experiment, we can notice that the amount of resources collected by the creatures grows almost as fast as in the same transition in the first experiment. By generation 217, around 550 resources were collected by the best individual. But the vocalization interpretation evolution was not as smooth as in the first experiment (figure 7).

In the start of cycle 2, only indexical responses appear as an alternative to ignore heard signs, and by generation 212 the population is split between ignoring the vocalization and indexically responding to it with a direct action. But even though the vocalization helps finding the high quality resource, an indexical response to it is quite faulty, providing bad actions as responses. By generation 213, the first creatures start responding symbolically to the vocalization, interpreting it as if a resource was seen, and reusing the already acquired behavior in cycle 1. The symbolic response take over the population after 20 generations and is adopted by the majority of the population. Nevertheless, we can see that this response preference is not as stable as the indexical response in experiment 1, because it is more probable to go from a symbolic response to a indexical response then the other way around. But all 10 best individuals in each generation, after this convergence, are interpreting the vocalization symbolically.

Discussion

These two experiments allow us to see conditions that might guide the emergence of indexical or symbolic interpretation. In the first experiment, the acquisition of *indexical* competence, for associating arbitrary signs directly to expected motor responses is a cheap process and prevails in the population, even though the creatures already acquired the ability to coordinate visual sensory data with actions during cycle 1, and reusing this ability for auditory data would seem faster. This is due to the relative ease of learning a new ability, in face of the low probability to acquire the ability of *symbolic* response.

In the second experiment, the cost of coordinating sensory data and actions is higher, and the adaptation of symbolic responding to vocalizations does act as a viable *cognitive shortcut*, that will use the already costly acquired trait of coordinating RD1s visual and RD1m, so there is no need to learn a new coordination again. We propose that a symbolic interpretation process can happen if a cognitive trait is hard to be acquired and the symbolic interpretation of a sign will connect it with another sign for which the creature already has an appropriate response.

One further test we ran (to be described in a future work) was of removing cycle 1 from the second experiment and let the simulation start at cycle 2, with the vocalizer placed in the high quality resource and all creatures able to hear, but starting with random FSMs. It would be expected that since there was no acquired trait a symbolic response would no prevail, but surprisingly the creatures spend quite a few generations ignoring any sign heard. Only after they are able to almost adequately coordinate visual data with actions, they start interpreting the vocalizations, and they do it symbolically.

Conclusion

The emergence of interpretation processes in computational models is an open issue in Artificial Life experiments. Even though there has been already many experiments on the emergence of different traits of communication systems, the research area still lacks studies on the modalities of processes underling the interpretation of the signs been communicated, and on the conditions that might conduct to the emergence of different modalities of interpretation.

Here we proposed a synthetic experiment to examine the conditions for the emergence of symbolic and indexical interpretation processes. Simulated creatures could interpret available vocalizations in three ways: not interpreting it, interpreting it indexically or interpret it symbolically. From the results obtained, we can conclude that indexical interpretation can emerge when the acquisition of a direct coupling of sensory and motor domains is a cheap process, and symbolic interpretation of signs can emerge as a cognitive shortcut across different sensory modalities, when coordinating representations and actions directly is a costly trait to acquire.

These are initial experiments on the study of conditions for the emergence of different modalities of interpretation processes. Other possible set ups for our experiment will make certain connections faulty (like the connection between RD1s visual and RD1m) and test the robustness of this competence and of it being used as a cognitive shortcut. Furthermore, another experiment will also be built in a scenario where all creatures can hear each other and also vocalize, with no immovable creature, and test not only sign interpretation processes but also sign production processes.

Acknowledgments. R.G. acknowledges FAPESP for funding support. J.Q. thanks the Brazilian National Research Council (CNPq) and the State of Minas Gerais Foundation for Research Support (FAPEMIG).

References

Braitenberg, V. (1984). Vehicles: Experiments in synthetic psychology. Cambridge, MA: MIT Press.

Cangelosi A. (2001). Evolution of communication and language using signals, symbols and words. IEEE Transactions on Evolutionary Computation. 5(2), 93-101.

Christiansen, M.H., & Kirby, S. (2003). Language evolution: consensus and controversies. Trends in Cognitive Sciences, 7 (7), 300-307.

Deacon, T. (1998). The Symbolic Species: The Co-Evolution of Language and the Brain. New York: W. W. Norton & Company.

Floreano, D., Mitri, S., Magnenat, S. and Keller, L. (2007) Evolutionary Conditions for the Emergence of Communication in Robots. Current Biology, 17 pp. 514-519.

Hauser, M. D. (1997), The Evolution of Communication, Cambridge, MA: MIT Press

Nolfi Stefano, Mirolli Marco (2010) Evolution of Communication and Language in Embodied and Situated Agents, Berlin: Springer Verlag.

Peirce CS (1958) Collected Papers of Charles Sanders Peirce. Harvard University Press, Cam-bridge, Mass..

Queiroz, J. & Ribeiro, S. (2002). The biological substrate of icons, indexes, and symbols in animal communication: A neurosemiotic analysis of vervet monkey alarm calls. In Shapiro, M. (Ed.) The Peirce Seminar Papers 5 (pp.69–78). Berghahn Books, New York.

Ribeiro S, Loula A, Araújo I, Gudwin R, Queiroz J (2007) Symbols are not uniquely human. Biosystems 90:263-272..

Seyfarth, R., Cheney, D., & Marler, P. (1980). Monkey responses to three different alarm calls: Evidence of predator classification and semantic communication. Science, 210, 801–803.

Wagner, K., Reggia, J. A., Uriagereka, J., & Wilkinson, G. S. (2003) Progress in the simulation of emergent communication and language. Adaptive Behavior, 11(1):37-69.

A Self-Supervised Classifier Ensemble for Source Recognition in Acoustic Sensor Arrays

Edgar E. Vallejo[1] and Charles E. Taylor[2]

[1]ITESM Campus Estado de México, Atizapán de Zaragoza, 52926, México
[2]University of California, Los Angeles, Los Angeles, CA, 90095, USA
taylor@biology.ucla.edu

Abstract

In this paper, we propose a collective self-supervised learning method to be deployed in acoustic sensor arrays. We describe a series of experiments on the automated classification of tropical bird species and bird individuals from their songs by a classifier ensemble. Simulation results showed that accurate classification can be achieved using the proposed model.

Introduction

Adaptive sensor arrays provide excellent platforms for testing hypotheses about critical properties of living systems, including collective and social behavior, communication and language, emergent structures and behaviors, among others. Further, understanding the capabilities and limitations of sensor arrays are useful for understanding self-organization in its own right, and may also prove helpful in guiding the construction of artificial agents that possess problem-solving abilities.

Over the past few years, we have been concerned with developing acoustic sensor arrays for use in observing and analyzing bird diversity and behavior (Vallejo and Taylor, 2009). We would like each sensor to see and "understand" part of the situation – depending on its own location – then to fuse their experiences with other such sensors to form a single, coherent understanding by the ensemble (Taylor, 2002). The ideal is that the array will act something like a living membrane, sensitive to what is going on within it, around it and passing through it.

So far, we have developed and tested sensor arrays that can identify their own location and sense bird vocalizations in real-world settings. We have developed filters to identify species (in some instances individual birds) and software tools to localize those individuals in natural environments. In the same vein, we have determined, to some extent, the conditions under which different classification approaches, both supervised and unsupervised, would be particularly effective (Vilches, et al 2006; Escobar, et al 2007; Vallejo et al 2007; Trifa, et al, 2008; Kirschel, et al 2009).

A problem with unsupervised learning methods has been that a particular bird species might be attached to one category in one part of the array, but to another category in other parts of the array. Therefore, achieving coherence and consistency in classification at the ensemble level have remained elusive. The main goal of the learning process should not only be to allow individual nodes to classify environmental sources accurately, but also to achieve coherent and consistent classification capabilities along the entire sensor array.

Toward that goal we have devised a self-supervised classifier ensemble model in which individual nodes of the array collectively act as both learners and teachers during the learning process. At each training step, each node of the array uses the classification outcomes of its neighbor nodes as output targets and learns accordingly. Therefore, the provision of labeled data from an external teacher is not necessary as the ensemble uses self-supervision for achieving collective classification capabilities.

Here we report simulation results on birds species recognition from their songs using the proposed model. Preliminary results indicate that consistent and coherent classification capabilities could be deployed in sensor arrays using self-supervised classification. Moreover, the time required for achieving convergence in learning have been improved for unsupervised classification.

Related work

In this section, we summarize the work of our laboratories aimed at developing filters to identify species, and individual birds in natural environments. These employ a variety of supervised and unsupervised approaches, as described below.

The simplest is to calculate the power spectrum, whereby the amount of energy at each wavelength is calculated and used to form a vector, typical to that individual or species.

We obtain better results by generating a sonogram of the vocalization, then look at particular features of those sonograms that might be particular to the species or individuals. We have found it most helpful is to adapt methods from human voice recognition to create a Markov Transition Matrix appropriate to the vocalizations of each individual or species. We are also looking at other methods that appear promising, especially data mining and Self-Organizing

Maps.

A collection of software tools have proven helpful for feature extraction, by providing efficient representations of bird songs while at the same time preserving the essential information contained in the songs. The emphasis has been on feature selection and on the conversion of analog waveforms into efficient digital representations. These tools, some of which are described in Kirschel et al, (2009), are mostly built on the signal processing toolbox of MatLab. Such transformations of signals are intended to minimize the communication capacity required for transmission of bird songs over a sensor network, to minimize the storage capacity required for saving such information in databases, and to provide the simplest possible accurate descriptions of a signal so as to minimize the subsequent complexity of identification and localization of individual birds.

Following feature extraction, we explored the use of different data mining techniques for the classification of bird species. The main goal has been to understand the importance of particular features of the acoustic signal that are distinctive for the accurate discrimination of bird species. A secondary goal has been to reduce the dimensionality of the acoustic signal in order to minimize the computational resources required for its manipulation and analysis.

Our approach has been to obtain large collections of temporal and spectral attributes using signal processing software tools to characterize bird songs and to use data mining to extract implicit and potentially useful information from these data. In this way, we have obtained a collection of association rules that describe correlations among features that appear to be inherent to a group of individuals and their conspecifics (Vilches et al, 2007).

Particularly, we used decision tree-based ID3 and J.48 algorithms for the identification of the most informative attributes and then use the selected attributes for species discrimination using a Naive Bayes classifier. Experimental results showed considerable dimensionality reduction can be achieved without significant loss in species classification accuracy with respect to alternative methods (Vilches et al, 2006).

In addition, we have explored the use of Self-Organizing Maps (SOMs) for the acoustic classification of bird species and individuals. The overall goal has been to examine the scope in which unsupervised learning is capable of conferring meaningful categorization abilities and increasing autonomy to sensor arrays.

Despite its preliminary character, our experiments with SOMs indicate that accurate unsupervised categorization of bird species can be achieved using two-dimensional SOMs (Escobar et al, 2007). However, unsupervised classification of bird individuals have proven to be extremely difficult for SOMs so we are beginning to explore complementary approaches such as semi-supervised and supervised classification.

Bird song is thought to possess a hierarchical organization similar to that used for describing human language. As a result bird song is typically described as consisting of phrases, syllables and elements (Catchpole and Slater, 1995). We have drawn inspirations from the structure of bird song to formulate a hierarchical approach for species and individual unsupervised classification.

The overall approach has been to transform the acoustic signal of bird songs into strings of symbols. This transformation is achieved by the unsupervised classification of syllables of the original acoustic signal using a competitive learning network. Unsupervised species classification is achieved using a second competitive learning network that classifies strings of symbols from their syllable structure (i. e. syntactical) features (Vallejo et al, 2007).

Our experiments suggested that using different abstraction levels for the description of bird song provides a convenient approach for analyzing different aspects of the acoustic signals. On the one hand, temporal and spectral features have proven to be useful for the categorization of song segments. On the other hand, compositional features of syllables have proven to be sufficiently informative for species classification.

Despite of their obvious advantages, unsupervised learning methods have shown important limitations in practice. For example, even though individual nodes have been competent at discriminating bird species, and in some cases individual birds, achieving consistency and coherence in classification along the entire sensor array has been less satisfactory. In this paper, we further elaborate on this particular aspect of source recognition.

Methods and tools

Biological context

The principal field site for our work has been the rainforest environment at the Estacion Chajul in the Reserva de la Biosfera Montes Azules, in Chiapas Mexico (approximately $16°6'44''$ N and $90°56'27''$ W). The species of birds in our analysis have been antbirds from the suboscine families *Thamnophilidae* and *Formicariidae*. The songs of suboscines are less complicated than those of some others, and are thought to be largely determined genetically, rather than learned, making them more stable and appropriate for testing methods of classification. The species toward which we have directed most of our attention are Barred Antshrikes (BAS) (*Thamnophilus doliatus*), Dusky Antbirds (DAB) (*Cercomacra tyrannina*), Great Antshrikes (GAS) (*Taraba major*), and the Mexican Antthrushs (MAT) (*Formicarius analis*). The spectrograms describing the songs of each species are shown in Figure 1. It is apparent that the songs from different species posses a similar structure. In effect, they consist of repetitive segments of sounds that span similar frequency spectra. These similarities pose challenges for

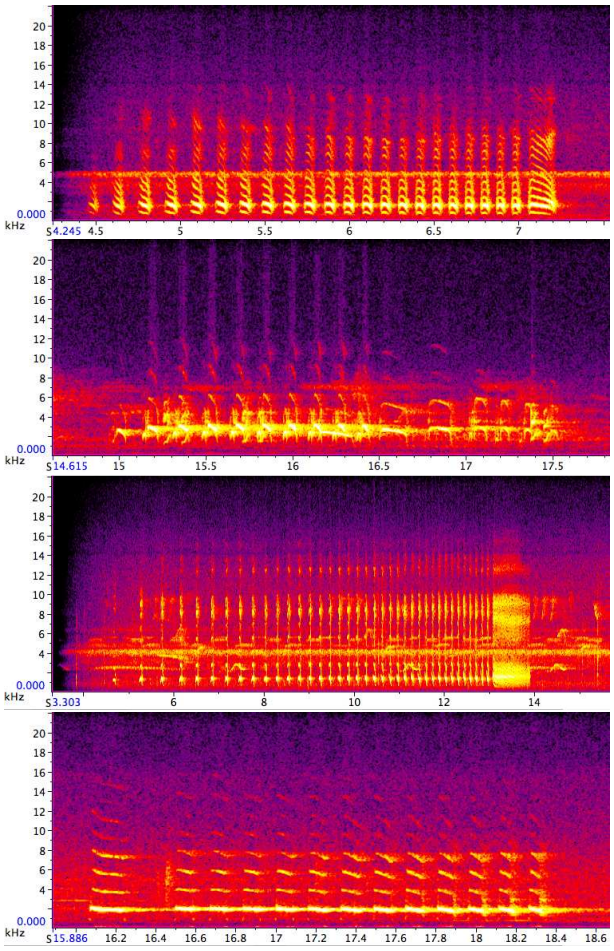


Figure 1: Spectrograms for antbirds in this study. From top, BAS, DAB, GAS, and MAT. The spectrograms were obtained from the Raven sound analysis software tool (Charif et al., 2004).

automated species recognition; especially for those methods that rely on unsupervised classification.

Sensor arrays

Th sensor arrays we are using consist of Acoustic ENSBox subarrays (Girod et al, 2006), pictured in Figure 2. These are ARM-based embedded platform designed for rapid development and deployment of distributed acoustic sensing applications. Each subarray node is self contained, with an embedded processor and a four channel microphone array that can process data locally as well as archive it and forward to other nodes wirelessly.

Typically, 5 - 8 nodes are deployed concurrently to form a distributed system of sensor sub-arrays. They are typically placed 10 - 30m apart encompassing the area to be monitored. They are automatically calibrated, to determine their node locations and orientation, then activated to perform

Figure 2: The Acoustic ENSBox Version 2, shown deployed near Chajul Station at left. A detailed description of both the hardware and software of this platform may be found in Collier (2010).

streaming event recognition and acquire data when triggered by animal vocalizations.

This approach provides greater sensor coverage, and creates a multi-hop wireless network for forwarding data and results back to a base station where data can be archived and displayed. Since each sub-array is small and has a fixed geometry, data from a single sub-array can be processed using algorithms that rely on coherence. Data from several subarrays can be fused to perform source localization (Ali, et al 2008). Mre detailed descriptions of the hardware and software of this platform may be found in Collier (2010) and Collier et al (2010a).

Self-supervised classifier ensemble

For this study, we devised a self-supervised classifier ensemble model (El Gayar, 2004). Different versions of self-supervised learning have been increasingly used for modeling different aspects of life-like behavior such as pattern classification, sensory motor coordination and motion planning, among others (Cohen, 2007; Lieb, 2005).

The proposed classifier ensemble consists of a collection of competitive neural networks in which classification is achieved by self-supervised learning as described below. Each competitive learning network, in turn, consists of a single layer of output units C_i, each fully connected to a set of inputs o_j via excitatory connections w_{ij}. Figure 3 shows an example of such a network.

The presence of an external source initiates the operation of those nodes of the ensemble that perceived the external stimulus. Particularly, if a node of the ensemble detects an input stimulus, it proceeds to determine the output unit that most resembles the input signal. Formally, given an input vector $\mathbf{o}$, the winner is the unit C_{i^*} with the weight vector $\mathbf{w}_{i^*}$ as follows:

$$|\mathbf{w}_{i^*} - \mathbf{o}| \leq |\mathbf{w}_i - \mathbf{o}| \text{ (for all } i)$$

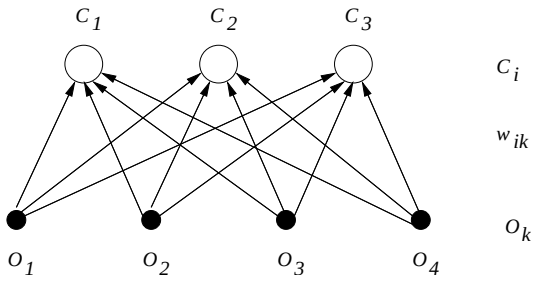


Figure 3: Simple competitive learning network. Each unit C_i can be seen as possessing a prototype that is used to represent a collection of inputs belonging to the same category.

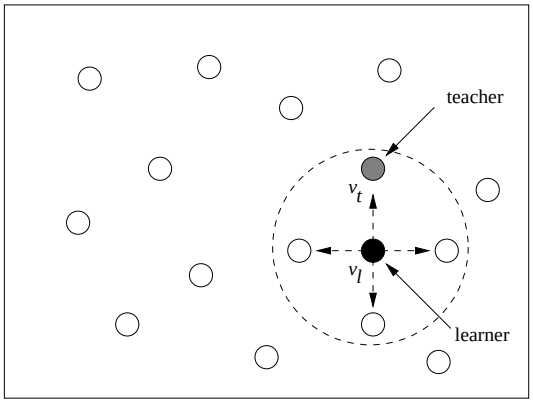


Figure 4: The learning procedure. Learner node v_l interacts with teacher node v_t and then iterates over all of the neighbor nodes.

Once the output nit for a given input has been determined, the node of the ensemble becomes a learner and its neighbor nodes become teachers, as shown Figure 4. For example, a learner node v_l of the ensemble detects an input $\mathbf{o}$ from the environment and determines the winner unit C_{i^*l}. The learner node v_l then communicates with the teacher node v_t to use the teacher's winner unit C_{i^*t} as label for the input $\mathbf{o}$.

The learner node v_l then updates the weights w_{i^*j} for the winning unit C_{i^*} only, as follows:

$$\Delta w_{i^*j} = \begin{cases} +\eta(o_j - w_{i^*j}) & \text{if } C_{i^*l} = C_{i^*t} \\ -\eta(o_j - w_{i^*j}) & \text{if } C_{i^*l} \neq C_{i^*t} \end{cases}$$

where $\eta \in [0, 1]$ is the learning constant.

A prediction derived from the formulation of the learning algorithm is that learning at the node level would be accelerated by the interaction of the learner node with a group of teacher nodes instead of using a target output provided by an external teacher. Furthermore, coherence and consistency of classification at the ensemble level would be incidental to the collective learning process.

The operation of the collective self-supervised learning procedure is described using the pseudocode in Table 1.

1. Create a set N of neural networks with initial random weights (one for each node)
2. `Do until` number of simulation steps k is met
   (a) `For each` node $v_l \in N$ that detects an input signal `do`
      i. Determine the winner unit C_{i^*l} of v_l
      ii. Select a set $T \subseteq N$ of networks in the neighborhood of v_l
      iii. `For each` node $v_t \in T$ `do`
      Modify the weights of v_l using the learning rule:

$$\Delta w_{i^*j} = \begin{cases} +\eta(o_j - w_{i^*j}) & \text{if } C_{i^*l} = C_{i^*t} \\ -\eta(o_j - w_{i^*j}) & \text{if } C_{i^*l} \neq C_{i^*t} \end{cases}$$

   `End for`
  `End for`
`End do`

Table 1: Training algorithm.

Parameter	Value
Nodes	16-32
Neighbors	2-8
Categories	4-8
Learning constant	0.01-0.1
Simulation steps	100-2000

Table 2: Parameters for the simulations. The values of the learning constant and simulation steps were determined empirically.

Experiments and results

Bird species recognition

We conducted simulations in order to explore the capabilities of the proposed classifier ensemble on the discrimination of bird species from their songs. We use recordings obtained by Martin L. Cody at our field site. From these recordings, we generated a collection of unlabeled training and validation sets using the procedure described in (Vallejo, et al 2007). Twelve training and twelve validation samples for each species of antbirds (BAS, DAB, GAS and MAT) were used in our experiments.

Multiple simulations were conducted using different combinations of parameter values as shown in Table 2. The following were the major results:

1. The classifier ensemble produced a meaningful classification of the unlabeled training sets. Table 3 shows the accuracy in classification in a typical simulation.
2. The classifier ensemble produced acceptable generalization performance when confronted to labeled validation sets, as shown in Figure 5.
3. Reasonable numbers of training steps (˜500) are required

procedure	accuracy	classified	misclassified
training	93.75%	45	3
testing	91.66%	44	4

Table 3: Classification results

procedure	accuracy	classified	misclassified
training	77.50%	33	7
testing	72.50%	31	9

Table 4: Classification results

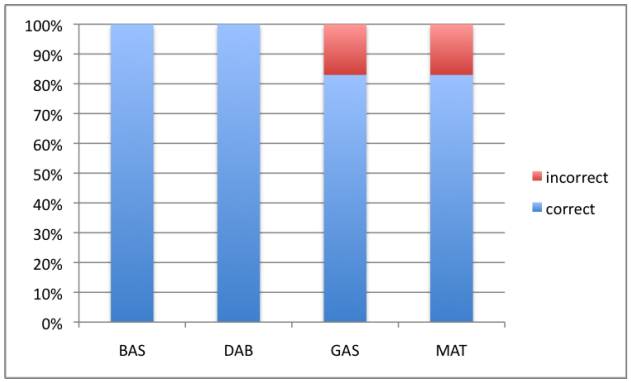


Figure 5: Classification results during validation. Misclassified samples are false negatives

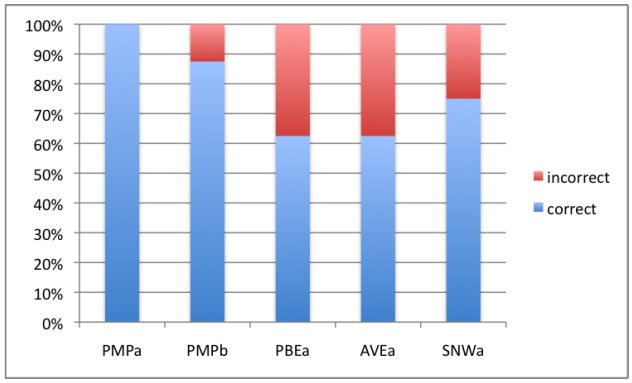


Figure 6: Classification results during validation. Misclassified samples are false negatives

for achieving coherent and consistent classification along the entire classifier ensemble.

4. Low communication bandwidth would be required for data transmission between nodes of a sensor arrays during self-supervised learning.

5. Coherence and consistency in classification along the entire classifier ensemble is achieved without compromising the accuracy of classification of individual nodes.

Bird individuals classification

It is sometimes possible to distinguish individual singers. Songs were recorded from each of 5 Mexican Antthrushs (MAT) (*Formicarius analis*) bird individual during December 2006, by Martin Cody. The identification of each singer was inferred from timing and location. The individuals were identified by labels PMPa, PMPb, PBEa, AVEa, and SNWa, Samples of 16 songs from each of the 4 territories they occupied (labeled PMP, PBE, AVE, SNW) were included. The sonogram of each song was measured for 7 traits, including length and maximum or minimum frequency at various parts of the song, so that each song was represented by a vector. From this dataset, it is apparent that some individuals are clearly distinguished while others are much less so, at least by inspection.

Multiple simulations were conducted using different combinations of parameter values as the previous experiment. The classification results obtained in a typical simulation are shown in Table 4. Specific results during validation are shown in Figure 6.

Conclusions and future work

Our long term goal is to provide sensor arrays with the adaptation capabilities required to identify the meaning of bird vocalizations in the social context of the vocalizing animals. This requires event recognition, symbol grounding and adaptive communication in order for the array to arrive at a collective understanding (Lee et al, 2003). Previous studies have established plausible scenarios for the emergence of these capabilities in sensor arrays (Collier and Taylor, 2005).

Several methods for event recognition have been suggested, e.g. (Nolfi, 2005). We are currently examining methods based on information theory, among others (Kobele et al, 2004). Symbol grounding, identifying and binding semantically meaningful events to symbols, then communicating that information among parts of the arrays is of great importance.

Once events have been recognized then we can use the unsupervised classification to categorize the songs . A problem has been that new events might be attached to one symbol in one part of the array, but to another symbol in other parts of the array. Our future efforts will be directed at testing the prediction that coherence and consistency in communication could be achieved in sensor arrays using the method proposed here.

Finally, we are developing the linguistic structure that is necessary to describe these songs and events in an expressive, learnable manner, based on the ideas developed by Stabler et al (2003).

Overall, adaptive sensor arrays seem promising platforms for monitoring applications. In the near future, our efforts

will be directed towards enabling sensor arrays with increasing adaptability and cognitive abilities. To accomplish this we will build largely on the results reported here.

Acknowledgements

This work was supported by the US National Science Foundation under Award Number 0410438 and by Consejo Nacional de Ciencia y Tecnología under Award Number I0110/127/2008. The authors would like to thank Lewis Girod, Travis C. Collier, Alex Kirschel, Martin L. Cody and Kung Yao for their invaluable contribution to this paper.

References

Ali, A. M. S., Asgari, T. C. Collier, M. Allen, L. Girod, R. E. Hudson, K. Yao, C. E. Taylor,Blumstein, D. T. (2008) An empirical study of collaborative acoustic source localization. *J. Sign. Process Syst.* **57**:415-436.

Catchpole, C. K., Slater, P. L. B. (1995) *Bird song biological themes and variations.* Cambridge University Press.

Charif, R. A., Clark, C. W., Fistrup, K. M.: *Raven 1.2 user's manual.* Cornell Laboratory of Ornithology, Ithaca, NY, 2004.

Chen, C-E. , A. Ali, W. Asgari, H. Park, R. E. Hudson, K. Yao, Taylor, C. E. (2006) Design and testing of robust acoustic arrays for localization and enhancement of several bird sources. In *Fifth International Conference on Information Processing in Sensor Networks*

Coen, M. H. (2007) Learning to sing like a bird: Self-supervised acquisition of birdsong. In *Proceedings of the Twenty Second National Conference on Artificial Intelligence (AAAI'07)*

Collier, T. C., Taylor C.E. (2004) Self-Organization in Sensor Networks. *Journal of Parallel and Distributed Computing* **64**:7 pp.866–873.

Collier, T. C. (2010) Wireless sensor network-based acoustic localization for studying animal communication in terrestrial environments. PhD Thesis, Department of Ecology and Evolutionary Biology, University of California, Los Angeles.

Collier, T. C., Kirschel, A. N. G., and C. E. Taylor (2010) Acoustic localization of antbirds in a Mexican rainforest using a wireless sensor network. *The Journal of Accoustical Society of America* In press.

El Gayar, N. (2004) An Experimental Study of a Self-Supervised Classifier Ensemble. *International Journal of Information Technology*.

Escobar, I. A., Vilches, E., Vallejo, E. E., Cody, M. L., Taylor, C. E. (2007) Self-organizing acoustic categories in sensor arrays. In F. Almeida e Costa, M. L. Rocha, E. Costa et al (eds.) *Advances in Artificial Life, 9th European Conference, ECAL 2007*. LNAI 4648, pp. 1161-1160, Springer-Verlag.

Girod, L., Lukac, M., Trifa, V., Estrin, D. (2006) The Design and Implementation of a self-calibrating distributed acoustic sensing platform. In *ACM SenSys*.

Hertz, J., Krogh A., Palmer, R. G. (1991) *Introduction to the theory of neural computation.* Addison Wesley, 1991.

Kirschel, A. N. G. , D. A. Earl, Y. Yao, I. A. Escobar, E. Vilches, E. E. Vallejo, and C. E. Taylor (2009) Using songs to identify individual Mexican antthrush *Formicarius moniliger*: Comparison of four classification methods *Bioacoustics* **19**:1-20

Kobele, G. M., J. Riggle, R. Brooks, D. Friedlander, C. Taylor, E. Stabler (2004) Induction of Prototypes in a Robotic Setting Using Local Search MDL. in M. Sugisaka and H. Tanaka, (eds.), *Proceedings of the Ninth International Symposium on Artificial Life and Robotics* Beppu, Oita Japan, pp 482-485.

Lee Y., Riggle, J., Collier, T.C. et al (2003) Adaptive communication among collaborative agents: Preliminary results with symbol grounding. In M. Sugisaka, H. Tanaka (eds), *Proceedings of the Eighth International Symposium on Artificial Life and Robotics* (AROB8th), Beppu, Oita Japan, Jan 24-26, pp.149-155.

Lieb, D., Lookingbill, A. and Thrun, S. (2005) Adaptive road following using self-supervised learning and reverse optical flow. *Proceedings of robotics: science and Systems*.

Nolfi, S. (2005) Categories Formation in Self-Organizing Embodied Agents. in H. Cohen and C. Lefebvre (eds), , *Handbook of Categorization in Cognitive Science* Elsevier, Amsterdam.

Stabler, E. P., Collier, T. C., Kobele, G. M., et al (2003) The learning and emergence of mildly context sensitive languages. In W. Banzhaf, T. Christaller, P. Dittrich, et al (Eds.) *Advances in Artificial Life, 7th European Conference, ECAL 2003.* Springer-Verlag.

Taylor, C. E. (2002) From cognition in animals to cognition in superorganisms. In M. Bekoff, C. Allen and G. Gurghardt, (eds.), *The Cognitive Animal. Empirical and Theoretical Perspectives on Animal Cognition* The MIT Press.

Trifa, V. M., Kirschel, A. N., Taylor, C. E., Vallejo, E. E (2008) Automated species recognition of antbirds in a Mexican rainforest using hidden Markov models. *The Journal of Accoustical Society of America* **123**:2424-2431.

Vallejo, E. E. and Taylor, C. E. (2004) A simple model for the evolution of a lexicon. In *Proceedings of the International Symposium on Artificial Life and Robotics AROB9th*.

Vallejo, E. E. and Taylor, C. E. (2009) Adaptive sensor arrays for acoustic monitoring of bird behavior and diversity: preliminary results on source identification using support vector machines. *Artificial Life and Robotics*, vol 14, num 4, pp. 485-488. Springer-Verlag.

Vallejo, E. E., Cody, M. L., Taylor, C. E. (2007) Unsupervised acoustic classification of bird species using hierarchical self-organizing maps. In M. Randall, H. A. Abbass, and J. Wiles (Eds.) *Progress in Artificial Life, Third Australian Conference, ACAL 2007.* LNAI 4828, pp. 212-221, Springer-Verlag, 2007.

Vilches, E., Escobar, I. A., Vallejo, E. E., Taylor, C. E. (2006) Data mining applied to acoustic bird species recognition. In *18th International Conference on Pattern Recognition, ICPR 2006* Volume 3, pp. 400-403., IEEE.

Philosophy of Artificial Life

Robust Explanations in Artificial Life

Eric Silverman[1] and Takashi Ikegami[1]

[1]Department of General Systems Studies,
The Graduate School of Arts and Sciences,
The University of Tokyo, 3-8-1 Komaba, Tokyo 153-8902
erics@sacral.c.u-tokyo.ac.jp
ikeg@sacral.c.u-tokyo.ac.jp

Extended Abstract

Finding robust explanations of behaviours in Alife and related fields is made difficult by the lack of any formalised definition of robustness. A concerted effort to develop a framework which allows for robust explanations of those behaviours to be developed is needed, as well as a discussion of what constitutes a potentially useful definition for behavioural robustness. To this end, we must differentiate between two senses of robustness: robustness in systems; and robustness in explanation.

When discussing systems, robustness is often described as a property which gives the system a certain resilience against perturbation. A robust system is thus able to retain functionality despite variation. In contrast, we define a robust explanation as a scientific explanation which can identify causal factors that underlie a phenomenon in a variety of circumstances.

The concept of robustness analysis, pioneered by Levins (1966), has illuminated the importance of developing a comprehensive research programme to develop such explanations. Levins argues that doing so requires the study of multiple models of that same phenomenon. Each model should be distinct, containing differing core assumptions or methodologies. If these different models still produce similar results, we can develop what Levins calls a robust theorem: an explanation of the behaviour of interest which is largely independent of the details of the models being studied.

The difficulty for Alife researchers lies in developing an appropriate set of models to produce robust explanations. Weisberg (2005) provides an intensive examination of robustness analysis, describing the concept of a robust property, or a property common to multiple models which contain different idealising assumptions. This leads to a discussion of the need to find common structures between models: those elements which give rise to the robust property. However, many models in Alife not only have different idealising assumptions, but may be based on vastly different methodologies entirely.

In order to escape this conundrum, we need a unified framework under which to search for common structures in order to perform robustness analysis. Models in Alife can frequently share a conceptual relationship - they examine similar behaviours within biological systems, but using fundamentally different methods. The way forward is to create experiments and simulations which share common grounding and related contexts, even when these experiments are quite different in implementation.

An examination of our own work in robotics (Hubert et al, 2009) and biochemical experiments (Ikegami 2009) will provide an example of how divergent methodologies can be used to develop a framework of idealising assumptions. This framework can then form the basis for the development of robust explanations. The commonalities found between the robust behavior of the robot (Hubert et al, 2009) and the biochemical experiments (Ikegami 2009) demonstrate recovery mechanisms which can keep a system from degrading into non-moving states. Here self-movement creates robustness and robustness enables ”intentional” behavior. Through an examination of these common structures, we can begin to develop a framework for robust explanations of these self-movement behaviours.

References

Hubert, J., Matsuda, E., Silverman, E., and Ikegami, T. (2009). A robotic approach to understanding robustness. In Proceedings of Mobiligence 2009, Awaji, Japan.

Ikegami, T. (2009). The search for a first cell under the maximalism design principle. Journal of Technoetic Arts, In Press.

Levins, R. (1966). The strategy of model-building in population biology. American Scientist, 54:421–431.

Weisberg, M. (2005). Robustness analysis. Philosophy of Science, 73:730–742.

Weak Emergence and Complexity

Henrik Thorén[1] and Philip Gerlee[2]

[1]Lund University, Department of Philosophy, Kungshuset, Lundagård, 222 22, Sweden, henrik.thoren@fil.lu.se
[2]Center for Models of Life, Niels Bohr Institute, Blegdamsvej 17, 2100, Copenhagen, Denmark , gerlee@nbi.dk

Abstract

In this paper we consider Mark Bedau's notion of *weak emergence* (WE) and relate it to various attempts to objectively construe complexity. We argue that the heavy reliance on a specific notion of complexity risks rendering the concept superfluous. Furthermore we discuss what sort of systems might reasonably be understood as exhibiting emergence at all and point out that the macro-level needs to be at least minimally structured. A worry may thus be formed that macro-level generalisations provide the sort of short-cut that is explicitly excluded from WE thus potentially making the concept apply only to chaotic systems of limited interest (in this context).

Introduction

Artificial life research can in many instances be characterised as a search for the surprising. A very general question posed by researchers in the field is: what type of behaviour can we expect from a system with the following dynamics? If the answer is obvious or expected the system is often neglected or simply not classified as Alife because it is not life-like enough. Biological life is full of surprises and therefore ALife should be as well.

Fortunately systems with interesting and often surprising behaviour are not difficult to find. Classical examples include cellular automata of class IV (Wolfram, 2002), evolving systems such as Tierra (Ray, 1992), Avida (Ofria and Wilke, 2004) and more recently systems investigating chemical interactions such as Urdar (Gerlee and Lundh, 2010) and the Organic Builder (Hutton, 2009).

This notion of surprise or appearance of higher-order structure such as universal computation in CA or the evolution of parasites in Tierra is often in the literature labelled with the term *emergence*. The notion of emergence is however originally a philosophical term, with many precise albeit disparate definitions. In order to bring the concept more formally into the ALife-community Bedau (1997) recently introduced the notion of *weak emergence*, which takes a simulation-based approach to the definition of emergence. Roughly put, the idea being that a property P of a system S is weakly emergent iff the only procedure for deciding if S will have P at some later time is to simulate the system.

His approach has however been met with critique from several philosophers, e.g. for being too broad (Stephan, 2006). A defense of the thesis has been presented on several occasions (Bedau, 2003, 2008), clarifying his intentions and arguing for the merits of WE.

In this paper we will argue that Bedau's definition of weak emergence relies so heavily on a notion of *complexity* it risks conflating into it. Further we note that complex systems often exhibit higher-order structures, which can be described by law-like generalisations on that level, but this contradicts the very notion of weak emergence, suggesting that it misses the point all together. Whatever the outcome of this debate is we also note that established measures of complexity can lead to a quantification of weak emergence applicable to both real and artificial systems.

Emergence

The concept of *emergence* is usually traced back to a handful of British thinkers active during the second half of the 19th century among them figuring names such as John Stuart Mill, Samuel Alexander and C.D. Broad. They considered themselves as inhabiting a moderate position in which both dualism in the form of vitalism and mechanism could be avoided (Kim, 1999, 4). At its intuitive base the idea is that a whole can be more than the sum of its parts. Complexes may have properties not analysable in terms of the properties of their constituent parts. At the time this thought was very much empirically justifiable. The special sciences—chemistry was a favourite example—seemed to be hopelessly irreducible to ontologically more fundamental sciences, such as for instance physics.

Despite its appeal the idea withered to the onslaught of the unity of science movement and fell out of vogue from the 30s and onwards, not to be considered seriously again until the ultimate demise of that tradition in the early 70s.[1] Since

[1]Quantum mechanical explanations of chemical bonds is often blamed, chemistry being a favourite example of emergence for these philosophers and scientists.

then emergentism has experienced a small renaissance, not least within the scientific community. The interest in complexity as of the past couple of decades seem to have ushered its return.[2] In philosophical quarters emergentism or similar positions found new defenders among non-reductive materialists.

A central tenet of British emergentism was that emergents were entirely unpredictable from knowledge of their emergent base Kim (2006). The early Emergentists considered the appearance of emergent properties as metaphysically contingent, brute facts of nature. No amount of knowledge about the underlying structure allows one to predict the emergent. But since supervenience was thought to hold, appearance of emergent properties were considered to be lawful. Given that one had observed some emergent property in connection with some specific microstructure an "emergence law" (transordinal law on Broad's terminology) could be formulated. Such a law would be a fundamental law of nature. 'Prediction' should hence be understood as *theoretical* prediction, or derivation, and not as what one may call *inductive* prediction. Broad e.g. writes "[i]f emergence be true they [the emergent properties] *could* not have been deduced from any amount of of reflexion on the properties of these constituents taken separately or in non-living wholes..." (Broad, 1925, 75) Mill seem to have held a view very similar to this.[3] Properties of wholes that could be deduced straight-forwardly from the properties of their constituent parts were referred to as *resultant* properties. Oft cited C. Lloyd Morgan (1923) writes concerning the distinction between resultant and emergent properties.[4]

> ...both distinguish those properties (a) which are additive or subtractive only, and predictable, from those (b) which are new and unpredictable; both insist on the claim that the latter no less than the former fall under the rubric of uniform causation. (Morgan, 1923)

As Kim (1999) has pointed out there is reason not to take the 'additivity and subtractivity' requirement literally. The idea was to pick out properties that could be predicted by means of *some* compositional principle, as e.g. additivity or subtractivity. Other principles however were clearly acceptable; the law of composition of forces being a favourite example.[5] So a resultant property is such that it can be calculated from knowledge of the basal properties by means of some compositional principle. Emergent properties of some whole were understood in contrast to this as properties that: 1), supervene on some basal property; and 2), is not predictable by means of such a compositional principle (and knowledge of properties of the parts).

But this is clearly not enough to make the distinction lucid. As the early Emergentists well understood given one is to combine a few quantities it is logically contingent what sort of principle one should use. Physics is riddled with straight-forward compositional principles and it seems that faced with a new case it is an entirely empirical matter which one is appropriate. Thus this would render cases like weight addition, composition of forces etc. cases of emergence which is clearly not right and definitely not what the early Emergentists had in mind. Broad and Mill solved this dilemma by putting restrictions on these principles disallowing principles working for properties of parts in other combinations. As McLaughin (2008, 92f) has pointed out the problem with such an approach is that almost nothing counts as emergent.[6]

An alternative strategy involves prohibiting what Van Gluick (2001) calls *specific value emergence*. Strictly speaking specific value emergence is not a form of emergence at all, but rather the most trivial form of resultance. Suppose we have a whole consisting of two proper parts a kilogram each in weight. The whole will weigh two kilograms despite none of the parts having that *specific* weight. We will return to this idea in the section below as this is part of Bedau's strategy.

Conclusively what is sometimes called *strong* emergence has been offered significant attention in the philosophical debate in the past twenty or so years and it has been found to suffer from serious problems. A lot of these problems stem from the difficulty to get the emergence/resultance distinction just right. Either too much or too little counts as emergent. Contemporary accounts typically strive for weaker formulations trying to salvage some part of the concept whilst giving others up. Mitchell (2009) does this by means of defending a form of downward causation deploying a multiple realisation argument. A different strategy is put to work by Bedau that defends a notion of emergence that tries to find objective criteria for a form of unpredictability that seems to fit the purposes.

[2] A search on Google Scholar combining the keywords complexity and emergence generates over a million hits. A quick browse through the philosophical literature will also reveal a connection between the terms 'emergence' and 'complex' that seems deeper than the connection warranted by taking 'complex' to denote an object that has parts.

[3] Mill never used the term 'emergence' but discussed what he called *heteropathic effects*, effects to which the causes do not abide by any principle of composition of causes. See McLaughin (1997) for a thorough discussion of Mill's views on this matter.

[4] The "both" here refer to the thinkers to which Morgan claims to owe this distinction; John Stuart Mill and George Henry Lewes.

[5] See e.g. (Mill, 1869, 210ff)

[6] Interestingly Kim (2006) has voiced critique seemingly pointing in the opposite direction claiming that emergence an accounts such as the above is under-characterised. The problem is that both supervenience and (in this case) non-derivability are negatively defined. Though not a decisive argument it raises the problem that the phenomena *emergence* might not be a genuine category.

Weak Emergence

Within the field of Artificial Life philosopher Mark Bedau has over a number of years developed and defended a variety of emergence he calls *weak emergence* (henceforth WE). WE may be characterised as a *strong* form of epistemological emergence since it does not rely on psychological or logical limitations of human cognition but rather an objective notion of complexity.

Bedau has written extensively on the subject but here we are going focus on two more recent works, Bedau (2003) and Bedau (2008) respectively. In these texts one find several characterisations, in the first article WE is defined in terms of a requirement of simulation, in the second an appeal to *explanatory incompressibility* is voiced. Bedau himself however views these two varieties as essentially one, "[t]hese two definitions are similarly indirect, and they are essentially equivalent" (2008, 444). We shall also treat them as such. Hence we believe that the following reflects Bedau's idea well. For a macro-property M of a system S to be WE the following two criteria should be met;

1. M is nominally emergent.

2. There is a derivation from P to M but that derivation can only be generated through simulation.

Nominal emergence is understood as the "...notion of a macro property that is the kind of property that cannot be a micro-property." (Bedau, 2003, 158) Notably this is equivalent to what Van Gluick (2001) calls *modest kind emergence*, at least taken in the stronger modal version. The necessity claim here is not further specified though the name suggest nominal necessity. In that case this qualification taken by itself includes a host of phenomena on both sides of the resultant/emergent divide. Bedau seems well aware of this (Bedau, 2003, 158).

This second criteria is a little more difficult. Importantly Bedau accepts (for the systems under scrutiny anyway) what he calls *causal fundamentalism*, the thesis that "...macro causal powers supervene on and are determined by micro causal powers" (Bedau, 2003, 159). So strictly speaking WE properties are only resultant, as there exists a derivation from micro to macro. Bedau's idea however is to pick out a certain kind of derivation. In Bedau (2003) this is to be thought of as "derivation by simulation," and this in turn should be interpreted in the strongest possible sense. Bedau writes:

> A derivation by simulation involves the temporal iteration of the spatial aggregation of local causal interactions among micro elements. (Bedau, 2003, 164)

What Bedau seems to be saying is that a simulation here is a process that produces or reproduces the *actual* mechanism in question.[7] Hence WE phenomena appear in accurate computer simulations and natural systems alike.[8] A central feature of such a derivation is that it must be done stepwise so that the further into the future one is interested in making predictions, the longer the derivation will be. In Bedau (2008) WE is thought of in terms of *incompressible generative explanations* connecting micro-state P with emergent M. Bedau writes:

> An explanation is generative just in case it exactly and correctly explains how macro-events unfold over time, how they are generated dynamically. (Bedau, 2008, 445)

This characterisation also requires the 'explanation' to follow the *actual* procedure (crawling the causal web) and 'short-cuts' are explicitly prohibited.

> If an explanation of some macro-property of some system is incompressible, then there is no short-cut generative explanation of that macro-property that is true, complete, accurate, and can avoid crawling the causal web. (Bedau, 2008, 446)

Let is try to construe this in a more formal fashion.[9] Suppose we have a micro-P (an initial condition) and a macro-M (at some later time) that stand in a WE relation to each other.[10] Then there is some sequence $P_1, P_2, ..., P_n$ connecting P and M, let us call this sequence D. There is no other sequence connecting P and M that is shorter than D and also satisfies the criteria of being true, complete, accurate and avoids crawling the causal web. We take it that if it is true and complete it must also be accurate and "crawling the causal web" entails that for every other derivation E that is exactly as long as D, then E is identical to D.

What about false derivations that are shorter but none-theless accurately predict M from P? It seems that this characterisation is much too strong. Truth, completeness, accuracy and causal web-crawling trivially homes in on just these micro-sequences, regardless of the system at hand. If it is the dynamics one is interested in, then broad and approximative statistical models that essentially leap-frogs the bowels of whatever process one is studying, just won't do. But that is

[7] On a weaker understanding one would only require from the simulation that it be sufficiently similar with respect to some characteristics of the original process. The mechanism driving the simulation however would not have to be qualitatively identical to process which it mimics.

[8] Of course inaccurate simulations could also exhibit WE, put perhaps with other emergents than the ones belonging to the system they are mimicking.

[9] In the below section we use 'derivation' instead of 'explanation,' we do not however think it matters. The explanation Bedau seem to have in mind are derivational. Besides 'derivation' is the preferred term in Bedau (2003).

[10] Bedau interchangeably talks about objects, properties, states and facts so let us give this a neutral account.

so regardless of whether it is possible to do so or not. So it seem Bedau would have to opt for some more inclusive idea of what exactly amounts to a short-cut. Perhaps the idea that derivations concerning states further away requires more computational power is more important and promising. Our worry however is that in order to avoid making a characterisation that is non-trivial Bedau would have to accept that there can be no regularities at all elsewhere in the system, and this in turn warrants the question whether the system at hand has any macro-level at all. We will however return to this topic in our discussion.

What sort of systems might this be true of then? Bedau relates this to systems that are *complex*. Emergents, on Bedau's take, is not epistemological in the sense that emergents are dependent on "human frailty." To the contrary not even infinite knowers could avoid using this type derivation in making successful predictions regarding these systems.

> Incompressibility of explanations is a consequence of the objective complexity of the local micro-causal interactions that are ultimately generating the emergent behavior being explained (Bedau, 2008, 453).

Thus Bedau means to move the 'ontological burden' away from the notion of emergence, where it has shown to be problematic, to the notion of complexity. We will now move on to discuss the notion of complexity introducing a few of formal complexity measures, and propose a link between WE and the complexity of a system.

Complexity

An intuitive understanding of the predicate 'complex' with regards to some object (process or pattern) entails that the object is structured in such a way that it is very difficult (or perhaps impossible) to describe.[11] In recent years the study of complex systems have enjoyed some popularity, especially within biology and ecology but also within e.g. statistical mechanics where the aim often have been to provide formal definitions or objective criteria. A quantitative measure has however turned out to be difficult to find. This is at least partially due to disparate use of the term in various disciplines; complexity is often thought to be salient in structures such as the human brain, weather and climate systems, but also in single-celled organisms. In the scientific community it has been in use since the rise of systems theory and cybernetics in the 40s and 50s, and has the last 20 years experienced a revival. On some construals the notion seems to approximate the concept of emergence. Consider for example the definition by Simon (1962):

> Roughly, by a complex system I mean one made up of a large number of parts that interact in a nonsimple way. In such systems, the whole is more than the sum of the parts, not in an ultimate, metaphysical sense, but in the important pragmatic sense that, given the properties of the parts and the laws of their interaction, it is not a trivial matter to infer the properties of the whole.

This definition falls close to the weak sense of emergence, but of course depends on how we interpret 'not a trivial matter'. A more recent remark by physicist Nigel Goldenfeld (Editorial, 2009) states that:

> Complexity starts where causality breaks down.

This claim is even stronger, and might put complexity on par with stronger notions emergence. However, independent of the exact interpretation of these statements our point is that the notions of emergence and complexity are intertwined, and that Bedau's notion in fact lies close to well-developed quantitative measures of complexity. Before we proceed with this thesis, let us look more closely into what we mean by complexity and how to measure it.

The concept of complexity has a relatively short history in the natural sciences. Before the 20th century the physical sciences were confined to the study of *simplicity*, while biology and the medical sciences, unable to explain the omnipresence of complex form and function, were concerned with collection and classification of living systems. It is here important to distinguish between systems which are complex and those which are merely complicated, or as put by Weaver (1948): complex in a organised vs. disorganised way. By complicated systems we refer to those which consist of large number of interacting parts with many degrees of freedom, such as an ideal gas, which yield to a statistical description, while complex systems are those which tend to organise themselves and exhibit structure despite being governed by local microscopic rules of interaction.

Intuitively we would like to class objects as being complex if they lie somewhere in between complete order and randomness. The human eye and the organisation of a colony of termites are things typically considered complex, while a crystal structure with its endless repetition, or an unstructured gas both fall outside our notion of complexity. To capture this intuition into a quantitative measure has however turned out to be immensely difficult. Many attempts have been made at defining complexity, either from a *structural* or *functional* point of view (McShea, 1996; Wimsatt, 1972), although none fully satisfactory, and the most successful route has instead been to consider the complexity of strings, called *sequence* complexity.

The first attempt along these lines was made by Kolmogorov (1968) (and later Chaitin (1975)) and quantifies the complexity of a sequence as the shortest possible description of that sequence. This is done by considering the shortest computer program or algorithm which when executed will reproduce the sequence in question, and from this

[11] One may thus note that already on this early stage there is some tension between ontological and epistemological aspects of the concept.

complexity measure has gained its name Algorithmic Complexity (AC). It is also related to the amount of information contained in the sequence as defined by Shannon entropy (Shannon, 1948). The problem with this measure is that it assigns maximal complexity to sequences that are completely random, and also assigns low complexity to intricate objects that can be generated with simple rules. A prime example of this is the Mandelbrot set, which because it can be generated with a very short algorithm has a low AC, although its structure suggests otherwise. AC therefore deviates from our intuitive notion of complexity, at least in some instances.

By measuring the running time of the shortest computer program generating the sequence, instead of its length, Bennett (1988) was able to overcome the problem of assigning low complexity to seemingly complex mathematical objects. This approach was motivated by the fact that complex objects often have a long causal history, and by equating the history with running time a quantitative measure can be defined. These attempts are nevertheless intractable because the length of the shortest program is provably non-computable, and we have no way of *a priori* telling which program is the most plausible.

This shortcoming was addressed by Grassberger (1986) who suggested an Effective Measure Complexity, which measures the complexity of a sequence as the value of having observed all previous symbols in the sequence when guessing the next. A similar measure termed Statistical Complexity was developed by Crutchfield and Young (1989), and measures the minimum amount of information required to make optimal guesses of the symbols in the sequences at an error rate h, where h is the Shannon entropy of the sequence. One drawback with these two measures is that they cannot measure the complexity the of a single sequence, but only of the ensemble from which sequences are drawn, although one can argue that complexity in fact is a property of an ensemble and not of a single object.

Applying these measures to dynamical processes can be accomplished by mapping the trajectory of the system, by a partition of the state space, into a symbol sequence which can then be analysed. For example the trajectory of the logistic map can be mapped to a binary alphabet and the corresponding binary sequence then reflects the complexity of the underlying dynamical systems, which turns out to be maximal at the period-doubling accumulation (Crutchfield and Young, 1989; Crutchfield, 1994). However, the structure of objects such as living organisms are currently impossible to capture by the dynamics of their underlying processes, which means that the above measures still fall short of a satisfactory account of complexity.

Systems which exhibit a high degree of complexity (in the sense of EMC and SC) have the interesting property that they exhibit structure (i.e. they are not maximally random) but at the same time the future state of the system is difficult to predict. This property has been termed "computational irreducibility" (Wolfram, 2002) and more precisely means that there is no way of predicting how the system will behave except by explicit simulation. Please note that this also holds for chaotic systems[12], but is of less interest as it is the combination of structure and unpredictability which we usually find interesting.

Precisely which systems qualify as computationally irreducible is currently unclear, but one sufficient condition is computational universality (i.e. Turing completeness). This condition is met by a few surprisingly simple systems such as Wolframs one-dimensional CA rule 110 (Cook, 2004), and the Game of Life (Berlekamp et al., 1982), which for some specific initial conditions instantiate a Universal Turing Machine. At least for a subset of these initial conditions the system is computationally irreducible, otherwise it would violate the halting problem. This suggests a link between universality and complexity which led Wolfram (2002) to formulate the Principle of Computational Equivalence, which states that all processes in nature (that are not obviously simple) can be considered as computations, and are of such complexity that they attain computational irreducibility. The human brain, an ant colony and a weather system, are according to the principle of the same computational sophistication, and instantiate computations which are irreducible. This is an intriguing and very bold statement, which if it is true, clearly has bearing on the ontological status of these objects.

Returning to WE several connections should become clear. Obviously unpredictability plays an integral part. Moreover incompressibility as Bedau thinks of it is very similar to computational irreducibility. Systems which are computationally irreducible and thus in principle impossible to forecast (and do not exhibit chaos) are precisely those of high complexity. This was already noted by Bedau (2003), but he did not follow through on the connection, which in the end leads to an interesting conclusion. In avoiding the metaphysical pitfall of the otherwise attractive idea of ontological emergence by appealing to complexity one find similar questions can be stated yet again, is *complexity* to be understood in ontological or epistemological terms? Wolfram's claim is that computational irreducibility and thus ontological complexity is ubiquitous in nature, and possibly the only one worth considering, although both concepts could clearly coexist.

Although the question of ontological complexity might be impossible to answer the link established between weak emergence and complexity might allow for quantification of the emergence a system exhibits. Systems with low complexity are easy to forecast, while those with high complexity might be impossible to predict the future of without ac-

[12]The relation between WE and deterministic chaos will be discussed below.

tually iterating the dynamics. This might provide a different route to quantifying weak emergence than the one suggested by Hovda (2008), which measures the degree of emergence as the length of a formal derivation of property P from the initial conditions, and instead focuses on the amount of information needed to make optimal predictions about the future of the system with respect to some property P.

It is also worth mentioning that complexity has previously been suggested as a route to defining emergence, by considering the predictive efficiency of a set of causal variables describing a system (Shalizi and Moore, 2003). The predictive efficiency can be quantified as the ratio between EMC and SC, and a set of variables are considered emergent from another set if 1) one is a coarse-graining of the other and 2) the coarse-grained variables can be predicted with higher efficiency. The prototypical example for this type emergence is the relation between statistical mechanics and thermodynamics.

Discussion

Complexity is usually thought to relate to emergence by causing it, or giving rise to it. Once a system reaches a certain degree of complexity emergent properties will start to appear. The relationship is more curious however. The reason is that complexity itself is an obvious systemic property that, at least in the systems under scrutiny here, spring from micro-structures that do not exhibit it. Quite to the contrary, at their ontological bottom they are notoriously simple. One the other hand the opposite might be true. A system may have a microstructure that is beyond description whilst being highly predictable on the macro-level. In that case we would perhaps talk of the emergence of simplicity. Given of course we deploy a weaker version of the concept. In the previous section we established a link between WE and complexity as measured by statistical complexity or effective measure complexity. We will now elaborate on this and the implications it has.

Interestingly it is often in complex systems that we find higher-level structure that behaves lawfully with respect to some higher-order dynamics. This is precisely the domain of the special sciences. Let us consider two examples of this lawfulness: In the Game of Life (GOL) (Berlekamp et al., 1982) there is a configuration known as a 'glider'. It consists of five active cells and has the peculiar property of moving across the lattice in a diagonal fashion. Now if we know that a glider is moving in a particular direction and at a given time is located at position x, then if it does not collide with any other cells predicting its position for all future times is easy, and does not require that we simulate the entire system. Next consider the dynamics of an ant colony. Without knowing the exact details of the anatomy of a particular ant, we can by coarse-graining it into what type of ant it is (queen, soldier etc.) get a good picture of what duties it will have in the colony. The system clearly exhibits regularities which allows us to formulate higher-order laws (or at least law-like generalisations), which in turn allow for prediction of the dynamics.

Although these systems, might be computationally irreducible on the micro-level they are still amenable to a coarse-grained description which can make reasonable predictions about the future state of the system. There is thus a clear tension in the link between WE and complexity that was presented above. Complex systems are possibly computationally irreducible and thus WE, but at the same time a WE system does not allow any short-cut derivations, which is precisely what higher-order structure allow. But again picking out systems with no higher-level structure at all seems to exclude precisely the kind of systems about which talk of emergence is the most appropriate.

Higher-order descriptions are typically coarse-grained in more than one respect; firstly by individuating the system differently (e.g. by using functional definitions), and secondly that they may imply some loss of accuracy in the predictions. This can happen in two ways, either as a consequence of noise, or as consequence of abstraction to more general terms.

The loss of accuracy is dependent on the level of coarse-graining one applies to the system. At the level of no coarse-graining we have to, assuming that the system is computationally irreducible, iterate the dynamics explicitly to make predictions about the future state of the system, e.g. if it will have a certain property P at time t. Now if we move one level up in the coarse-graining, e.g. in GOL we start talking about gliders and blinkers, we might be able to formulate laws at this level which faithfully describe the system, such as the fact the gliders move diagonally at the speed of light. These laws allows us to circumvent the actual simulation, but on the other hand introduces inaccuracy in the description. It also denies us any knowledge about the micro-state of the system at future times, as coarse-graining procedures by definition are non-invertible.

In the above example of the ant colony, knowing the type of ant only gives us a better than null prediction as to its behaviour, obviously not a perfect prediction of the future actions of the ant in question. For every coarse-grained description of the system we thus have an error rate of prediction. What we save in terms of not having to simulate the system at the 'basal' level is lost in the power of prediction. The rate at which this error increases varies between different systems depending on their regularity. Now, one way to read Bedau is to say that a WE occurs when the error rate of prediction on all coarse-grained levels is sufficiently high. To reliably forecast the dynamics it is necessary to revert to an explicit simulation of the system.

This discussion can in fact be couched in terms of Cruthfields ϵ-machine reconstruction (Crutchfield, 1994), where automata with different 'causal' states are able to predict

the future state of a system with varying accuracy. Viewing different levels of description as different ϵ-machines, we can make a formal comparison of both their complexity[13] and their accuracy. A similar approach to different levels of description has been pursued by Dennett (1991) in his discussion on the reality of patterns and ultimately beliefs in nature. He also notices the inherent trade-off between an accurate and complicated description versus a simple one with a higher error rate, and that this leads to a multitude of possible 'patterns' in the same data.

The above discussion covered systems which exhibit structure on some higher level, but there is also an interesting link between WE and deterministic chaos (DC). Chaotic systems are generally governed by local micro-level rules, or non-linear equations of evolution, and their hallmark is their sensitive dependence of initial conditions. This means that trajectories at machine precision distance from each other diverge exponentially, and implies that predictions about the future state of the system are difficult or impossible to make.[14]

These systems do not show regular structure[15], except possibly for some isolated regions of parameter space, and are also highly sensitive to initial conditions. The future state of a chaotic system is difficult to predict without simulation, and for reasonable choices of a property P it thus fulfills the criterion for WE, i.e. there are no short-cuts for predicting if the system will have P, it can only be decided by explicit simulation.

Depending on our rigour when accepting short-cut derivations, based on their accuracy, we naturally get different degrees of overlap between weakly emergent and chaotic systems. If we only accept predictions which are perfectly accurate then the class of WE-systems might incorporate both chaotic and complex systems, while if our criterion for accuracy is lower, and we accept statistical laws, then WE coincides more with systems considered chaotic.

Suppose we consider a form of system of which a concept of emergence does some actual work. As we have noted before the most obvious category consists of systems that have higher levels that are at least minimally structured, i.e. systems that succumb to macro-level generalisation of some form and degree of accuracy.[16] However, as discussed above, these systems seems to be excluded by definition from WE. The reason would be that macro-level regularities plausibly could be understood as exactly the kind of 'short-cut' Bedau dismisses. If this is true it seems WE can only be applicable to systems that are macroscopically unstructured. But it seems systems that lack structured macroscopic levels are usually uninteresting.

In a way this worry seems entirely misguided. The reason is that since these macro-level generalisation are located on the macro-level they themselves constitutes the emergents in this contexts and it is the derivation *of* them rather than *between* them that is under scrutiny. In other words, the rules which govern the higher-order structures (e.g. the collision of two gliders in GOL) are not derivable except by simulation from the micro-level dynamics.

To determine if this objection is genuine it seems one would have to specify what is micro and macro properties for the system under investigation. Though this might seem conceptually trivial it is decidedly less than straight forward in this particular context. We have already hinted at an example; a lot of kinds are functionally defined in GOL, take e.g. spaceships; anything that moves whilst retaining its shape over a relatively short period of time is a spaceship. Thus it makes out a *kind* on some non-basal level of description. But since any number of different micro-level configurations might exhibit this behaviour it seems there won't be a micro structural definition of spaceships. Some specific kinds of spaceships do have micro structural definitions, gliders are an example of that.

Other interesting candidates are more abstract systemic features like *chaos* or *complexity* that both seem to intuitively fit well on at least some conceptions of emergence. These predicates are usually ascribed (in this context at least) to entire systems where microscopical structures typically are very simple. They are thus systemic properties that are genuinely novel—systems with simple microstructures are not always complex—and they apparently aren't trivial in the sense that one can easily find configurations in e.g. GOL that do not exhibit complexity on any technical understanding of the term. Yet another category that might coextend with the one just mentioned concerns questions regarding specific initial states. Suppose one has a certain initial state for GOL and wants to know if it will produce a bounded dynamic or not. For some configurations these questions are computationally irreducible and thus also weakly emergent on Bedau's understanding, but what sort of macro-properties do these future states represent?

These are the types of questions that need to be addressed if we are to get a proper account of the relation between weak emergence, complexity and deterministic chaos.

In this paper we have elaborated on the connection between weak emergence and complexity. We found that WE lies very close to certain measures of complexity, and this might allow for a quantitative measure of WE. Further we noticed that complex systems often exhibit higher-order structure

[13] If the machine is minimal, then its statistical complexity is the amount of memory (in bits) required for the agent to predict the environment at the given level 'ϵ' of accuracy

[14] See Kellert (1993) for an extensive argument of the latter.

[15] Here we disregard from coarse-grained structure such as invariant measures, which can be defined for chaotic systems exhibiting ergodicity.

[16] This needs to further specified but following Fodor (1974) we think that minimally the higher level consists of functional kinds, usually however these kinds will allow for something more, macro-level laws or at least law-like generalisations.

which allows for coarse-grained prediction of the dynamics. This is in possible contradiction to the definition of WE, which implies that the scope of WE is narrow and possibly only covering systems exhibiting deterministic chaos. Instead we propose a different interpretation of the concept which focuses on the derivability of the rules acting on the higher levels in the system.

Acknowledgment

This work was in part supported by Adlerbertska Forskningsstiftelsen.

References

Bedau, M. (1997). *Philosophical Perspectives: Mind, Causation, and World*, chapter Weak Emergence, pages 375–399. Blackwell.

Bedau, M. A. (2003). Downward causation and the autonomy of weak emergence. *Principia Revista Internacional de Empistemologica*, 6:5–50. Page references are made to Bedau and Humphreys (2008).

Bedau, M. A. (2008). Is weak emergence just in the mind? *Minds & Machines*, 18:443–459.

Bedau, M. A. and Humphreys, P., editors (2008). *Emergence: Contemporary Readings in the Philosophy of Science*. MIT press.

Bennett, C. (1988). *The Universal Turing Machine*, chapter Logical depth and physical complexity, pages 227–257. Oxford University Press.

Berlekamp, E., Conway, J., and Guy, R. (1982). *Winning ways for your mathematical plays*. New York: Academic Press.

Broad, C. D. (1925). *The Mind and Its Place in Nature*. Routledge.

Chaitin, G. (1975). Theory of program size formally identical to information-theory. *J Acm*, 22(3):329–340.

Cook, M. (2004). Universality in elementary cellular automata. *Complex Systems*, 15:1–40.

Crutchfield, J. (1994). The calculi of emergence - computation, dynamics and induction. *Physica D*, 75(1-3):11–54.

Crutchfield, J. and Young, K. (1989). Inferring statistical complexity. *Physical Review Letters*, 63(2):105–108.

Dennett, D. (1991). Real patterns. *The Journal of Philosophy*, 88:27–51.

Editorial (2009). No man is an island. *Nature Physics*, 5:1.

Fodor, J. (1974). Special sciences. *Synthese*, 28:97–115.

Gerlee, P. and Lundh, T. (2010). Productivity and diversity in a cross-feeding population of artificial organisms. *Evolution, In Press.*

Grantham, T. (2006). Is macroevolution more than successive rounds of microevolution. *Paleontology*, 50:75–85.

Grassberger, P. (1986). Toward a quantative theory of self-generated complexity. *Int J Theor Phys*, 25(9):907–938.

Hovda, P. (2008). Quantifying weak emergence. *Minds and Machines*, 18:461–473.

Hutton, T. (2009). The organic builder: A public experiment in artificial chemistries and self-replication. *Artificial Life*, 15:21–28.

Kellert, S. H. (1993). *In the Wake of Chaos*. The University of Chicago Press.

Kim, J. (1999). Making sense of emergence. *Philosophical Studies*, 95:3–36.

Kim, J. (2006). Emergence: core ideas and issues. *Synthese*, 151:547–559.

Kolmogorov, A. N. (1968). Three approaches to the quantitative definition of information. *International Journal of Computer Mathematics*, 2:157–168.

McLaughin, B. (1997). Emergence and supervenience. *Intellectica*, 25:25–43.

McLaughin, B. (2008). The rise and fall of british emergentism. In Bedau, M. A. and Humphreys, P., editors, *Emergence: Contemporary Readings in Philosophy and Science*. MIT press.

McShea, D. (1996). Perspective: Metazoan complexity and evolution: Is there a trend? *Evolution*, pages 477–492.

Mill, J. S. (1869). *System of logic*. Harper & Brothers, New York.

Mitchell, S. D. (2009). *Unsimple Truths: Science, Compexity, Policy*. The University of Chicago Press.

Morgan, C. (1923). *Emergent evolution*. Williams and Norgate, London.

Ofria, C. and Wilke, C. (2004). Avida: A software platform for research in computational evolutionary biology. *Artificial Life*, 10:191–229.

Ray, T. (1992). An approach to the synthesis of life. In Langton, C., Taylor, C., Farmer, J., and Rasmussen, S., editors, *Artificial Life II*, pages 371–408, Redwood City, CA. Addison-Wesley.

Shalizi, C. and Moore, R. (2003). What is a macrostate? subjective observations and objective dynamics. *http://arxiv.org/abs/cond-mat/0303625*.

Shannon, C. E. (1948). A mathematical theory of information. *Bell System Technical Journal*, 27:379–423.

Simon (1962). Architecture of complexity. *Proceedings of the American Philosophical Society*, 106:1–17.

Stephan, A. (2006). The dual role of 'emergence' in the philosophy of mind. *Synthese*, 151:485–498.

Van Gluick, R. (2001). Reduction, emergence and other recent options on the mind-body problem: a philosophical overview. *Journal of Consciousness Studies*, 8:1–34.

Weaver, W. (1948). Science and complexity. *American Scientist*, 36(4):536–544.

Wimsatt, W. (1972). Complexity and organization. *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association*, 1972:67–86.

Wolfram, S. (2002). *A new kind of science*. Wolfram Media.

Models of Artificial Life: Herbert Simon and Evolutionary Computation

Robert T. Pennock[1]

[1]Michigan State University's Lyman Briggs College, Departments of Philosophy and Computer Science & Engineering
pennock5@msu.edu

Extended Abstract

Herbert Simon is justly regarded as the father of artificial intelligence and even of the fields of computer science and cognitive science as we currently conceive them. His Nobel Prize was in economics, but he also made significant contributions to philosophy, political science, psychology, public policy, and beyond. Among his nearly a thousand publications, were many that dealt with issues of causality, complexity, problem solving, the discovery process, learning, scientific theory testing, simulation and modeling, and even consciousness. Many of his research interests revolved around questions about decision-making under conditions of uncertainty, which he took to be the usual case for both organizations and individuals. Human beings have "bounded rationality" and so are not in a position to optimize their choices, but rather must "satisfice". Notably absent from this amazing body of work, however, is much about biology. Though many of the ideas Simon investigated are directly or indirectly relevant to artificial life research, he never had the opportunity to consider what light his AI research might shed on ALife and vice versa. This is a significant loss, as ALife is an especially important case by which to consider Simon's theses about the "sciences of the artificial." (Simon 1984) What might he have said about what each field could learn from the other? This article reviews some of Simon's distinctive notions about models and model-based reasoning in AI and outlines the beginning of an answer. In particular, it considers how current work in digital evolution builds upon, extends and in some cases overturns Simon's ideas about complexity, discovery, learning, intelligence and more. It concludes by highlighting how the ALife "bottom up" approach of digital evolution provides a radically different perspective on artificial intelligence that complements Simon's "top-down" approach and opens up promising new avenues of investigation.

References

Simon, H. A. (1984). *The Sciences of the Artificial (2nd ed.)*. The MIT Press, Cambridge, MA.

What Simulations Can Do That Experiments Cannot, And Vice Versa

Paul Humphreys[1]
[1] Corcoran Department of Philosophy, University of Virginia, Charlottesville, VA 22904-4780, USA
pwh2a@virginia.edu

Extended Abstract

In this talk I shall explore what kind of knowledge can be obtained from computer simulations and the sense in which that knowledge is different from what can be gained from traditional experiments on the one hand and from traditional theoretical work on the other. Some recent literature has suggested that the more similar the experimental subjects are to the target systems, the greater the security of the inferences involved. Although that is true in an important sense, it is not the most relevant aspect when we are interested in the role that concrete implementations play in simulations. I shall illustrate my arguments with examples from artificial societies and artificial economics, two areas in which agent based models have significant similarities to artificial life models.

Algorithmic Feasibility of Observing Artificial Life Evolution

Janardan Misra

HTS Research, Bangalore, India 560076
janardan.misra@honeywell.com

Abstract

An observation process is a fundamental implicit component of the simulation based studies on artificial-evolutionary systems (AES) by which time-varying entities are identified and their behavior is observed to uncover higher-level "emergent" phenomena. In this paper, we analyze algorithmic feasibility of implementing an observation process and consequent automated discovery of the entities and the evolutionary processes in arbitrary AES models. We characterize the bounds for the worst case computational complexity for the process of discovery of possible presence of entity and population level reproduction with epigenetic development in the child entities involving mutations and heredity in presence of natural selection. In particular, we prove that if entities in an AES simulation are structurally distinguishable, the problem of observability of evolutionary processes is only polynomially harder w.r.t. the entity recognition. The complexity bounds are presented in parameterized form so that for any given AES model, if parameter estimates are known, corresponding bounds can be derived.

Background

Studies on *Artificial Evolutionary Systems* (AES) are recent attempts to complement real-life theories to study the principles underlying the complex phenomena of life without directly working with the real-life organisms. For example, AES studies can complement theoretical biology by uncovering potential evolutionary dynamics (Ostrowski et al., 2007; Lenski et al., 2003).

Observations play a fundamental role in AES research, in particular, for those AES studies, which focus on the problem of the "emergence" of life-like behavior. However, the mechanisms and analysis often employed in AES studies to discover the emergent entities and their life-like behavior remain useful only to the specific models and do not always have the generic perspective. Therefore an important aspect where AES studies demand increasing focus is to study observational processes and mechanisms used in AES studies in their own right resulting into a framework for *automated discovery* of life-forms and their dynamics in simulated environments. With AES studies involving mostly digitized universes and their simulations, it is actually desirable to explore by algorithmic means potentially varied possibilities which these simulations hold yet usually require such detailed observations that it may not always be feasible to carry out for human observers alone. Such an automated discovery of life-forms and the evolving dynamics may bring much promise in AES studies as compared to what could possibly be achieved only with manually controlled observations.

An example of such an automated discovery of life forms is discussed in (Sayama, 1998). In order to observe the living loops in his Cellular Automata (CA) model, another "Observer CA" system is designed and embedded within the simulator software. The observer CA is capable of performing the complex image processing operations on the CA configuration given to it as an input by the simulator CA to automatically identify the living loops of different types. Also recently (Stone et al., 2009) have discussed the integration of artificial life simulations with interactive games-based techniques to study simulation complexity for the behavioral representation of species in fragile or long-vanished landscapes and ecosystems.

However, because of its implicit nature and the multitude of AES models, a precise characterization of the observation process is generally a difficult problem. Importantly it needs to be defined independent of the low-level micro dynamics any specific AES model to permit the study of higher-level observationally "emergent" phenomena. Initial work on systematically studying the observational processes independent of the underlying AES models appeared in Henz and Misra (2007); Misra (2009). In (Henz and Misra, 2007) an observation process is characterized as an abstraction on the model universe for establishing the necessary elements and the level of evolutionary behavior in that model. Based upon this formal characterization, in Misra (2009), it was proved that the task of entity recognition in a simulation, is a NP-hard problem and therefore cannot be completed in polynomial number of steps. In this paper we extend this result further and present computational complexity theoretic analysis for the problem of algorithmic discovery of evolutionary phenomena in AES studies. The presented analysis on observing evolutionary behavior reveals important insights on how computation intensive an automated discovery

of life-like phenomena could be.

Related Work To the author's knowledge, there is not much work focusing on the algorithmic feasibility analysis of generic models for AES studies. However, interestingly, for few specific AES models, there exist parallel results. For example, Melkikh (2008) considered the computational analogue of the problem of the origin of species in a genome space under DNA Computing framework (Paun et al. (2006)) and has shown that in absence of a priori information about the possible species of organisms, the underlying computational problem is NP-hard. Similarly, Centler et al. (2008) prove that the problem of computing a reactive chemical organization is NP-hard.

Notations: Set notations: $\setminus$ (set difference), $\mathcal{P}$ (power set), $\rightsquigarrow$ (partial function). Logical operators: $\wedge$ (and), $\neg$ (not), $\Rightarrow$ (implication), $\Leftrightarrow$ (if and only if), $\exists$ (existential quantifier), and $\forall$ (universal quantifier). Programing pseudo code notation: $if \ldots then \ldots$. $\mathcal{N}^+$ is the set of positive integers. For a vector $x = (a_1, a_2, \ldots, a_r), i^{th}$ element (a_i) will be denoted as $x[i]$. Also basic notions from multiset theory (Singh et al., 2007) (e.g., $\biguplus$ (multiset join)) and the theory of computational complexity (Papadimitriou, 1994; Cormen et al., 2001) (e.g., 'big-Oh' notation - $\mathcal{O}$[1]) would be used in the formal exposition of the derived results.

The Formal Structure of the Framework

In this section we will briefly review the axiomatic framework presented in Henz and Misra (2007); Misra (2009). In the ensuing discussion, we will use "AES model" and "model", "Observation process" and "Observer" interchangeably to add convenience in presentation. *Axioms* are used to specify conditions which need to be satisfied in order to draw valid inferences e.g., recognition of entities and their causal relationships.

Observation Process and the Model Universe

Axiom 1 (The Axiom of Observable Life)**.** *Life-like phenomena in a AES model exists only if it can be observed using its simulations.*

In other words, existence of life-like behavior can only be proved with respect to an observation process and associated simulations.

Definition 1 (Observation Process)**.** An observation process is an algorithmic transformation from the underlying AES simulation model to observer abstractions $(Abs_{ind},\ Abs_{dep})$, where Abs_{ind} is the set of process independent abstractions and Abs_{dep} is the set of process dependent abstractions.

Definition 2 (States)**.** Σ: set of observed states of the model across simulations.

Definition 3 (Observed Run)**.** $\mathcal{T} : \Sigma \rightsquigarrow \mathcal{P}(\mathcal{N}^+)$: An observed sequence of states ordered with respect to the temporal progression of the model during its simulation.

$\mathcal{N}^+$ acts as a set of indexes for the states in the sequence. Since a state may appear multiple times in a simulation, subsets of $\mathcal{N}^+$ are used to denote that. Each such sequence represents one *observed run* of the model. We let $\Sigma_{\mathcal{T}}$ denote the set of unique states appearing in a specific run $\mathcal{T}$.

Entity Recognition

Definition 4 (Entity Set)**.** E_s: Multiset of entities observed and uniquely identified by the observer in a state s of the model for a given run $\mathcal{T}$. $E_{\mathcal{T}} = \biguplus_{s \in \Sigma_{\mathcal{T}}} E_s$ is the multiset of entities observed and uniquely identified by the observer across the states in a given run $\mathcal{T}$.

"Tagging" can be used as a mechanism for identifying individual entities whenever there exist multiple entities in the same state which are otherwise indistinguishable.

Axiom 2 (Axiom of Unique Identification of Entities)**.** *An entity must be uniquely identified in a given observed run $\mathcal{T}$.*

Axiom 3 (Axiom of Unique Identification in States)**.** *If two states are identical, i.e., consist of the identical multisets of atomic observable structures, then an observer must identify the same multisets of entities in these states irrespective of their temporal ordering in the observed run $\mathcal{T}$.*

Axiom 4 (Axiom of non-Ignorance)**.** *It must not be true that an observer omits identification of an entity in a state s but in a different state s' identifies it as consisting of the same atomic elements which were also available in s.*

Definition 5 (**Character Space**)**.** An observer should define a set of all possible mutually independent (or orthogonal) and measurable characteristics for possible entities in the model as a multi dimensional character space $\Upsilon = Char_1 \times Char_2 \times \ldots \times Char_d$, where each of $Char_i$ is the set of values for i^{th} characteristic.

Corresponding to each entity $e \in E_{\mathcal{T}}$ there is a point in Υ, say $(v_1, v_2, \ldots v_d)$, where $v_i \in Char_i$.

Observable characteristics need not to be limited to syntactic level or *structural properties* and may also include semantic properties, which are *observable patterns of behaviors* abstracted over a range of states.

Definition 6 (**Distance Measure**)**.** An observer defines a computable clustering distance measure $D : E_{\mathcal{T}} \times E_{\mathcal{T}} \rightarrow \mathbf{Diff}$, where $\mathbf{Diff}$ is the set of values to characterize the observable "differences" between entities in E.

[1] Asymptotic order notation, $\mathcal{O}$, is used to measure the bounds on computational complexity for algorithms and problems. If $f(n) = \mathcal{O}(g(n))$, then f is said to be upper bounded by g for all the values of the input of size n after certain point. Two useful asymptotic properties of $\mathcal{O}$ are: If $f_1(n) = \mathcal{O}(g_1(n))$ and $f_2(n) = \mathcal{O}(g_2(n))$, then $f_1(n) + f_2(n) = \mathcal{O}(\max\{g_1(n), g_2(n)\})$ and $f_1(n) * f_2(n) = \mathcal{O}(g_1(n) * g_2(n))$.

Definition 7 (**Mutation Bound**). Based upon the choice of D, an observer selects $\delta_{mut} \in \mathbf{Diff}$ as a vector such that each element in δ_{mut} specifies an observer-defined threshold on the recognizable mutational changes for corresponding characteristic.

It is important to note that the choice of δ_{mut} critically affects further inferences. For example, a choice of very large values would result in the lack of identification of variability in characteristics among entities. On the other hand, with relatively smaller values for δ_{mut}, it is difficult to recognize persistence of an entity across states under changes.

Next, a *Recognition relation* is defined to establish the persistence of entities across states in the presence of mutational changes:

Definition 8 (**Recognition Relation**). An observation process establishes recognition of entities across states of the model with (or without) mutations by defining a partial function $\mathbf{R}_{\delta_{\mathbf{mut}}}$: $E_{\mathcal{T}} \rightsquigarrow E_{\mathcal{T}}$, satisfying following axioms:

Axiom 5. *Entities to be recognized as the same should be observed in successive states.*

Axiom 6. *No two different entities in one state can be recognized as the same in the next state.*

Axiom 7. *If an entity e mutates and in the next state is identified as e', observer might be able to recognize e and e' as the same only if these changes (between e and e') are bounded by δ_{mut}.*

In order to infer meaningful relationship among entities, to be used as a basis for inferring macro level phenomena in the model, an observer needs to first identify "causal" relationships among entities independent of the underlying 'physical laws' or 'micro level dynamics' of the model.

Definition 9 (**Causality**). $C \subseteq \biguplus_{s \in \Sigma_{\mathcal{T}}} E_s \times E_{s+1}$. C establishes the observed causality among the entities appearing in the successive states of a run $\mathcal{T}$.

Since causality is largely an observer and model dependent, it is further refined by defining additional axioms for specific cases, for example, for the case of reproductive causality to infer reproductive relationships among entities (See Axiom 8).

Observing Evolution

In the following discussion we will define components in Abs_{dep} for observing the fundamental evolutionary components: reproduction with mutations and epigenetic developments, heredity, and natural selection.

Reproduction An observation process establishes reproduction by defining causal descendance relationships among the entities across states, whereby parent and the child entities are recognized by the observer as being sufficiently similar and "causally" connected across the states. Formally, we add a new Axiom for the causal relation C defined before:

Axiom 8 (**Reproductive Causality**). *If an entity e in state s is causally connected to entity e' in the next state $s+1$, then there must not be any other entity e'' in state s, which is recognized by the observer as (mutating to) e'.*

In essence, this formulation of causality is an abstract specification which demands observers to identify the entities which have been observed to be causal sources for the appearance of a new entity.

Similar to δ_{mut}, as discussed before, it is important to specify the limits under which an observer can identify whether an entity is a descendant of another entity even though they might not be identical. This limit on observable reproductive mutations is essential while working with models where epigenetic development in the entities can be observed (Mahner and Bunge, 1997). This is because in such models including examples from real life, "child" entity and the "parent" entities may not have identical characteristics the beginning and therefore an observation process needs to wait until whole epigenetic developmental process gets unfolded and only then compare the entities for similarities in their characteristics.

Definition 10 (**Reproductive Mutation Bound**). Based upon the choice of D, the observer selects $\delta_{rep\_mut} \in \mathbf{Diff}$, which will be used to bound reproductive mutational changes for proper recognition.

δ_{rep_mut} assists an observer to establish whether a particular entity could be treated as a "descendant" of another entity or not. It is important to note that the choice of δ_{rep_mut} also critically affects further inferences. For example, small values for δ_{rep_mut} might make it harder to establish reproductive relationships among entities and for such an observer every new entity would seem to be appearing *de novo* in the model. On the other hand choice of very large values would result in the lack of identification of variability in characteristics and thus make it difficult to infer natural selection.

An auxiliary relation Δ is used to determine that the differences due to reproductive mutations are bounded by δ_{rep_mut}.

Definition 11. $\Delta \subseteq E_{\mathcal{T}} \times E_{\mathcal{T}}$ s.t. $\forall e, e' \in E_{\mathcal{T}}$. if (e, e') is in Δ then their differences for each single characteristic $char_i$ must be bounded by $\delta_{rep\_mut}[i]$ and e should not be recognized as mutating to e'.

Based on the thus established notion of "causal" relationships between entities and Δ, we define $\mathbf{AncestorOf}$ relation, which connects entities for which an observer can establish descendance relationship across generations.

Definition 12. $\mathbf{AncestorOf} = (C \cup \mathbf{R}_{\delta_{\mathbf{mut}}})^+ \cap \Delta$

In this definition the transitive closure of $(C \cup \mathbf{R}_{\delta_{\mathbf{mut}}})$ captures the observed causality (C) across multiple states even in cases when "parent" entities might undergo mutational changes ($\mathbf{R}_{\delta_{\mathbf{mut}}}$) before "child" entities complete their "epigenetic" maturation with possible reproductive

mutations. Intersection with Δ ensures that causally related parent and child entities are not too different from each other, that is, reproductive mutational changes are under observable limit.

Using $\mathbf{AncestorOf}$ relation, we now can consider the cases of *entity level reproduction* and *Fecundity*:

Case 1: Entity Level Reproduction We consider the case where instances of individual entities can be observed as reproducing. For a given simulation $\mathcal{T}$ of the model, an observer defines the following $\mathbf{Parent}_{\mathbf{\Delta}}$ relation:

Definition 13. $\mathbf{Parent}_{\mathbf{\Delta}} = \{(p,c) \in \mathbf{AncestorOf} \mid \nexists e \in E_{\mathcal{T}} \,.\, [(p,e) \in \mathbf{AncestorOf} \wedge (e,c) \in \mathbf{AncestorOf}]\}$

The condition in defining $\mathbf{Parent}_{\mathbf{\Delta}}$ is used to ensure that p is the immediate parent of c and thus there is no intermediate ancestor e between p and c. Using $\mathbf{Parent}_{\mathbf{\Delta}}$ relation, in order for the observer to establish reproduction in the model, the following axiom should be satisfied:

Axiom 9 (**Reproduction**). *There should exist at least one instance of reproduction in a simulation $\mathcal{T}$ of the model i.e.,* $\mathbf{Parent}_{\mathbf{\Delta}} \neq \emptyset$.

Since for every $(p,c) \in \mathbf{Parent}_{\mathbf{\Delta}}$, some other $(p',c') \in \mathbf{AncestorOf}$ where p (and/or c) has been observed to change to p' (c') may also be present in the $\mathbf{Parent}_{\mathbf{\Delta}}$, therefore, let $\mathbf{Parent}_{\mathbf{\Delta}}^{\mathbf{min}}$ consist of temporally least parent-child pairs (p,c) from $\mathbf{Parent}_{\mathbf{\Delta}}$.

Case 2: Population Level Reproduction - Fecundity Owing to the *carrying capacity* of the environment, which limits the maximum possible size of a population, for natural selection it is the population level collective reproductive behavior (fecundity), which is significant. Therefore in order to ensure that there is no perpetual decline in the size of the population, following axiom should hold:

Axiom 10 (**Fecundity**). *There exist statistically significant number of different generations of reproducing entities in temporal ordering $G_1, G_2, \ldots, G_L$ such that for every generation of reproducing entities, there exists a generation of its descendant entities such that the size of descendant generation is equal or more than the current generation.*

Heredity yet another precondition for evolution, can in general be observed on two different levels: Syntactic level and Semantic level. On *syntactic level*, entity level inheritance is implied by the structural proximity between parents and their progenies ranging over several generations. For syntactic inheritance to persist, design of the model needs to ensure that environment, which controls the reaction semantics of entities, remains approximately constant over a course of time so that structural similarities also result into continued reproductive behavior. On the other hand, the semantic inheritance is implied in terms of *semantic relatedness* between entities, whereby progenies and their parental entities exhibit similarities in their behaviors (e.g., reproduction) under near identical set of environments. This in turn would require an observer to abstract the behavioral (e.g., reproductive) semantics from the observable reactions among entities in the model, which in turn might require non-trivial inferences in absence of the knowledge of the actual design of the model.

Heredity usually requires further mechanisms to reduce possible undoing of current mutations in future generations owing to new mutations. Therefore, in order to establish inheritance in AES models, sufficiently many generations of reproducing entities need to be observed to determine that the number of parent-child pairs where certain characteristics (both syntactic and semantic) were inherited by child entities without further mutations is significantly larger than those cases where mutations altered the characteristics in the child entities. We can express it as the following axiom:

Axiom 11 (**Heredity**). *Let Ω be a statistically large observed subsequence of a run T, then there exists a characteristic $Char_i$ such that the set of entities in Ω, where this characteristics were inherited without (further) mutation is statistically significant.*

Natural Selection Following the idea from (Bell, 2008, page 19) that on evolutionary scale rate of reproduction is the only attribute selected directly and characteristics affecting the rate of reproduction are selected only indirectly, we consider natural selection as a *statistical inference* on *average reproductive success* of a population of reproducing entities over an evolutionary time scale. Towards that we define following necessary and sufficient axioms as generally discussed in the literature (Stearns and Hoekstra, 2000):

Axiom 12 (**Observation on Evolutionary Time Scale**). *An Observer must observe statistically significant population of different reproducing entities, say $\mathbf{\Lambda_{min}}$, for statistically large number of states in a run $\mathcal{T}$.*

Axiom 13 (**Sorting**). *Entities in $\mathbf{\Lambda_{min}}$ should be different with respect to characteristics in Υ and there should exist differential rate of reproduction among these reproducing entities. Rate of reproduction $ror(e)$ for an entity e is the number of child entities it reproduces before undergoing any mutations beyond observable limit.*

Axiom 14 (**Heritable Variation**). *There must exist variation in the inherited mutations in the population of $\mathbf{\Lambda_{min}}$ implying that a significant fraction of the population of all reproducing entities should have at least one unique characteristics.*

Axiom 15 (**Correlation**). *There must be non zero correlation between heritable variation and differential rate of reproduction.*

Yet another important constraint from the evolutionary perspective is that reproduction in a model should not entirely cease because of the (harmful) mutations. Though this

constraint is implicitly captured in the axioms 12 and 13, we can still restate it below primarily since this weaker version may enables us to directly argue for the reasons of the absence of evolutionary behavior in a model:

Axiom 16 (Preservation of Reproduction under Mutations). *Some mutations do preserve reproduction. In other words, if there exist reproductive entities in a state s, either some mutants of these entities or their children should continue reproducing further.*

Software Architecture for an Observation Process

An implementation of the observation process discussed so far essentially demands deciding the level of abstraction on which observations need to be carried out with respect to the underlying AES model. Once it is decided by the designer of the model, either of the following two approaches can be considered for the software design:

Source Code Interleaving/Embedding The specified observational processes can be executed by interleaving the programs for the observations and corresponding interferences within the source code of the AES model simulation design itself. Advantage of such interleaving is that the implemented observation process can reuse some of the computational resources (e.g., memory) of the AES model.

Interactive Observations An observation process could also otherwise be programmed as a separate process itself together with the actual AES model simulation process. These two processes could communicate with each other asynchronously by exchanging the messages containing the required information on the state changes by the model simulation process, which then can be used by the observation process independently for drawing the inferences. This keeps the design of both the processes independent of each other, however unlike the earlier option, the observation process requires to have separate resources for itself. Nonetheless, by virtue of the independence between these two processes, simulation cum observation can be carried out in a distributed environment, which can be useful in case of certain AES studies requiring large amount of computational resources to uncover rare and complex phenomena or detailed dynamics not possible to execute on a single machine owing to main memory limitations or CPU speeds.

Computational Complexity

In the next few (sub)sections, we will estimate upper bounds on the worst case time complexity for the problem of establishing axioms dealing with evolutionary components in the framework for arbitrary AES models. For a discussion on the very choice of worst case computational complexity measure, we request reader to refer to the next Section.

Estimates for space complexity, though equally important, will not be addressed. Primary reason for that is that space (memory) requirement is often dependent upon the actual model at hand, the syntactic nature of the entities as determined by an observation process, and is often linear w.r.t. the total number and size of states observed.

An important problem to be considered while providing estimates on the computational complexity is that observed state progression during simulations might not correspond to the actual underlying reaction semantics for a specific entity. In other words, observed states during simulations progress according to the underlying updating rules for the model, which determine which subset of entities would react in any state. However, in the following analysis, we assume that all those entities, which are enabled to react in each state, are indeed allowed to react. In cases where it is not true, an observation process may store state subsequences of finite size where all (or most of) the enabled entities have been observed to react and then merge all the states in each of these subsequences into single meta states, which reflect the effect that most of those entities which can react have actually reacted.

Computational Complexity of Entity Recognition

Following basic result was proved in Misra (2009):

Theorem 1. *The problem of entity recognition using structural (syntactic) constraints is NP-hard.*

Assuming that all the states in a simulation are of comparable size (i.e., having roughly same number of atomic observable elements), let us use $\mathcal{O}(n)$ as the size of any state. Therefore, if the size of a run $\mathcal{T}$ is r, entity recognition using structural constraints in all the states $s_0, s_1, \ldots, s_r$ may require in the worst case $\mathcal{O}(r2^n)$ steps.

In case, where entities do not have overlapping structures, corresponding upper bound is $\mathcal{O}(rn2^n)$ steps.

Computational Complexity of Observing Evolutionary Components

We can now discuss some of the computational complexity theoretic aspects of observing various components of evolution. Also we will use the following notations:
t_c: expected number of time steps required to determine membership of an entity pair in the relation C.
t_Δ: expected number of time steps required to determine membership of an entity pair in the relation Δ.
$t_{\delta_{mut}}$: expected number of time steps required to determine membership of an entity pair in the relation $\mathbf{R}_{\delta_{\mathbf{mut}}}$.
$t_=$: expected number of time steps required to compare two entities for equality checking.
t_D: expected time steps required to compute function D to check the equality (or inequality) of the characteristics of two entities.

We further assume that checking the negation of a condition takes same number of time steps as checking

the condition itself. For example, t_Δ would also be the expected number of time steps required to determine that an entity pair is not in the relation Δ.

Computational Complexity of Observing Entity Level Reproduction Establishing the case for the entity level reproduction in the simplest case, where there are no epigenetic developments in the child entities, minimally demands identifying a single instance of a reproducing entity and its progeny in the next state during one simulation. Suppose an observer needs to determine that an entity p in a state s is an instance of a reproducing entity. For this, the observer needs to establish that under the specified definition of the causal relation C, there exists another entity c in the state $s+1$ such that $(p,c) \in C$ and that the reproductive mutations in c with respect to p are bound by δ_{rep_mut}, i.e., $(p,c) \in \Delta$, and that there does not exist any other entity in the state s, which is recognized as mutating to c. This process would at worst take $N_p^{(s)} = t_c + t_\Delta + |E_s| t_{\delta_{mut}}$ steps where $|E_s| t_{\delta_{mut}}$ factor comes owing to the fact that for each of the $|E_s|$ number of entities in the state s, we need to ascertain that it is not mutating to c. Since for a state s, such a reproducing instance may not be found quickly, in the worst case all the entities in the state s might need to be assessed under these steps. Therefore search for an reproducing instance in a state s may take at worst

$$\begin{aligned}\mathbf{T}_{rp} &= \sum_{p \in E_s} N_p^{(s)} = |E_s| N_p^{(s)} = |E_s|(t_c + t_\Delta + |E_s| t_{\delta_{mut}}) \\ &\leq 2^n (t_c + t_\Delta + 2^n t_{\delta_{mut}}) = \mathcal{O}(2^n \max\{t_c, t_\Delta, t_{\delta_{mut}} 2^n\})\end{aligned}$$

steps, where $|E_s| \leq 2^n$. Since finding such a state s, where a reproducing entity may be present itself may require search into a potentially large state subsequence of a run, it might take $\mathcal{O}(r) * \mathbf{T}_{rp} = \mathcal{O}(r 2^n \max\{t_c, t_\Delta, t_{\delta_{mut}} 2^n\})$ steps to establish the entity level reproduction, where r is the number of states in the state subsequence used in the search assuming that all the states are of comparable sizes. Therefore we have

Proposition 1. *Given the sets of entities in each state, additional time steps required for observing entity level reproduction, without epigenetic development in the child entities and mutational changes in the parent entities, in an AES is upper bounded by $\mathcal{O}(r 2^n \max\{t_c, t_\Delta, t_{\delta_{mut}} 2^n\})$, where r is the number of states observed before first instance of entity level reproduction is recognized.*

The case where entities do not have overlapping structures, total number of entities in a state are restricted by the number of atomic structures, that is, $|E_s| \leq n$. Therefore we have the following corresponding corollary:

Corollary 1.1. *Given the sets of entities in each state, additional time steps required for observing entity level reproduction in an AES where entities do not have overlapping structures is upper bounded by $\mathcal{O}(rn \max\{t_c, t_\Delta, t_{\delta_{mut}} n\})$.*

Next let us consider the general case of entity level reproduction with epigenetic developments in child entities and mutational changes in the parent entities. Towards that we have the following result:

Theorem 2. *Given the sets of entities in each state, additional time steps required for establishing an entity level reproduction is upper bounded by $\mathcal{O}\left(r 2^n \max\left\{t_{\delta_{mut}}, t_c 2^n, t_\Delta 2^n, t_{=} r^3 2^{3n}\right\}\right)$.*

The case where entities do not have overlapping structures, we have the following corresponding bound: $\mathcal{O}\left(rn \max\left\{t_{\delta_{mut}}, t_c n, t_\Delta n, t_{=} r^3 n^3\right\}\right)$

Computational Complexity of Observing Fecundity In order to establish fecundity having recognized an entity level reproduction, the first problem for an observation process is to determine the temporal granularities for the generations of the reproducing entities especially when there may exist different types of reproducing entities with different rates of reproduction. In that case, requirement is to determine how many entity types need to be considered. Towards this, the observation process could initially scan a constant number of states to collect all different kinds of reproducing entities together with their rates of reproductions. Based upon the initial estimates on these rates of reproductions, it may consider their least common multiple as the granularity for a generation and ignore other new types of entities while aiming to establish the fecundity axiom. However in case such initial estimates do not yield sufficient support for the fecundity and more reproducing entity types need to be considered, backtrack step is necessary. This process need to continue till statistically significant number of states have been observed to get support for the fecundity axiom or to assume it to be statistically unsatisfiable in that simulation.

Let us first consider the case of single state reproduction without any epigenetic developments. In this case, we have:

Proposition 2. *Given the set of entities in each state, the worst case computational complexity of observing fecundity without epigenetic development is upper bounded by $\mathcal{O}(L 2^{2n} \max\{t_c, t_\Delta, t_{\delta_{mut}}, L/2^{2n}\})$ where L is the number of generations of the reproducing entities.*

Next, we consider the more general case involving epigenetic developments in the child entities:

Theorem 3. *Given the set of entities in each state, the worst case computational complexity of observing fecundity is upper bounded by $\mathcal{O}(L \max\{t_{\delta_{mut}} 2^n,\ t_c 2^{2n},\ t_\Delta r_\pi 2^{2n},\ t_{=} r_\pi^4 2^{4n}, L\})$, where r_π is the maximum of the lengths of the reproduction cycles of the different types of observed reproducing entities across these generations.*

In a special case of *replication* (with epigenetic development) involving no reproductive mutations in the child

entities and no parental mutations would only demand identification using syntactic equivalence between entities and counting the entities belonging to various reproductive types only in last state of each generation. The worst case complexity for such process is upper bounded by $\sum_{1\leq i\leq L}(|E_{i\lambda}| * k * t_{=}) \leq L * 2^n * 2^n * t_{=} = \mathcal{O}(Lt_{=}2^{2n})$, where $E_{i\lambda}$ is the multiset of entities in the last state of the i^{th} generation and k is the number of different types of reproducing entities in each generation.

Also the case where entities do not have overlapping structures, we have the following corresponding bound:

$\mathcal{O}\left(Ln\max\{t_{\delta_{mut}}, t_c n, t_\Delta r_\pi n, t_{=} r_\pi^4 n^3, L\}\right)$.

Computational Complexity of Observing Heredity

Theorem 4. *Given the sets of recognized entities in each state, the worst case computational complexity of observing heredity in an AES is upper bounded by*

$\mathcal{O}\left(r2^n \max\left\{t_{\delta_{mut}}, t_c 2^n, t_\Delta 2^n, t_{=} r^3 2^{3n+1}, |\Upsilon|^2 t_d 2^n\right\}\right)$

The case where entities do not have overlapping structures, we have the following corresponding bound:

$\mathcal{O}\left(rn \max\left\{t_{\delta_{mut}}, t_c n, t_\Delta n, t_{=} r^3 n^3, |\Upsilon|^2 t_d n\right\}\right)$

Computational Complexity of Observing Natural Selection Given the sets of recognized entities in each state and the relations $\mathbf{R}^{+}_{\delta_{\mathbf{mut}}}$, $\mathbf{Parent}^{\mathbf{min}}_{\mathbf{\Delta}}$, $\mathbf{\Lambda_{min}}$, and ror from the earlier steps, additional time steps required for establishing axioms for natural selection are upper bounded as follows:

- The Axiom 12 of Observation on Evolutionary Time Scale: $\mathcal{O}\left(t_{=} r^3 2^{3n}\right)$.
- The Axiom 13 of Sorting: $\mathcal{O}(r2^n \max\{r2^n, |\Upsilon|\})$.
- The Axiom 14 of Heritable Variation: $\mathcal{O}(r2^{2n} \max\{r^3 2^{2n}, t_d|\Upsilon|\})$.
- The Axiom 15 of Correlation: $\mathcal{O}(r|\Upsilon|2^n)$.

Given the upper bounds for these axioms, the following result is immediate for natural selection:

Theorem 5. *Given the sets of recognized entities in each state and the relations* $\mathbf{R}^{+}_{\delta_{\mathbf{mut}}}$ *and* $\mathbf{Parent}^{\mathbf{min}}_{\mathbf{\Delta}}$*, additional time steps required for establishing natural selection in an AES are upper bounded by*

$$\mathcal{O}\left(r2^{2n} \max\left\{t_{=} r^2 2^n, t_d|\Upsilon|, r^3 2^{2n}\right\}\right)$$

Given the estimates for the upper bounds on the time steps required for constructing the entity sets E_Ω, $\mathbf{R}^{+}_{\delta_{\mathbf{mut}}}$, and $\mathbf{Parent}^{\mathbf{min}}_{\mathbf{\Delta}}$, the bound for the overall computational complexity of observing natural selection can be estimated:

Corollary 5.1. *Overall worst case computational complexity of establishing natural selection in an AES is upper bounded by*

$\mathcal{O}\left(r2^n \max\left\{t_{\delta_{mut}}, t_c 2^n, t_\Delta 2^n, t_{=} r^3 2^{3n+1}, t_d|\Upsilon|2^n\right\}\right)$

The case where entities do not have overlapping structures, we have the following corresponding bound:

$\mathcal{O}\left(rn \max\left\{t_{\delta_{mut}}, t_c n, t_\Delta n, t_{=} r^3 n^3, t_d|\Upsilon|n\right\}\right)$

Significance of Results

Before we conclude, it is necessary to discuss why to study these worst case computational complexity bounds? In practice, today, most of the AES studies are carried out with significant manual involvement throughout the simulation process and not all the AES studies are carried out to such an extent that their fullest potential is conclusively explored. However as the field would progress, automated exploration of myriad of possibilities which AES simulation studies could have would also become increasingly important. Such automation necessarily present us with fundamental questions on the hardness and limits of such exploration.

One of well studied questions in the domain of algorithm design and analysis is the computational complexity analysis, which gives an insight on the fundamental resource requirements for the problem at hand with respect to the increasing input size. The precise characterization of the inherent resource requirements resulting from such analysis helps an algorithm designer to devise appropriate strategies to optimally utilize the available resources (e.g., CPU cycles) and also to have an estimate of how much could be achieved with available resources.

Among many possible complexity analysis (e.g., average case analysis, amortized analysis etc.) the one which appears most natural and tractable for AES studies is the worst case analysis considered in this paper. The reason is that other than the worst case analysis, other analyses demand either a unifying AES model or a complete characterization of all the AES models. However, currently known and foreseeable AES models differ so fundamentally from each other in terms of their syntactic structures and semantic rules that it is extremely hard to solve either of the problems of defining a unifying AES model or complete characterization of all possible AES models upon which such analyses could be carried out. Also owing to these irreducible design differences, analysis for one AES model could not be generalized in a meaningful manner for other models and thus an inductive approach of building a theoretical framework starting from specific AES case studies may not yield expected answers. Therefore the only fruitful analysis, which appears feasible is the worst case analysis, which could be performed by rather defining a unifying framework for an observation process independent of the underlying AES models.

Further question, which may arise to the reader is how could these results be used in practice? To discuss this, let us informally interpret the presented theorems:

Entity Recognition Theorem 1 could be interpreted as stating that if one has a large and complex simulation for an AES model, it will be computationally expensive to automatically determine the kind of entities, which would

emerge over time without externally supplied meta information.

Evolutionary Components On the other hand the remaining theorems state that if entities are structurally distinguishable (i.e., the case of non overlapping structures), once they are identified in a simulation (automatically or otherwise), determining whether evolutionary processes are effective on these entities can be checked in computationally less-expensive manner.

Further, the parameterized form of the results could be used to determine resource bounds for specific AES models having estimates for the required parameters. For example, if in a given AES model entity recognition is feasible in polynomial number of time steps and observed entities do not have overlapping structures, in that case an automatic discovery of natural selection and other evolutionary components could also be carried out using only polynomial number of time steps. On a different note, the specified axioms and proof steps provide practical guidance on implementing the actual observation process, which, once designed could as well be used as reusable component for different AES models with minor changes.

Conclusion

The work on formal characterization of the observational processes can be seen as an attempt to fulfill the need for explicitly separating the design of the AES models from the abstractions used to describe their dynamic progression and the discovery of life-like behavior. We consider evolutionary behavior, as one such characteristic property of life-like phenomena and discuss basic components for observing evolutionary behavior in AES models.

Computational complexity theoretic analysis of the entity recognition as well as establishing evolutionary behavior reveals that an automated discovery of life-like phenomena could be computationally intensive in practice and techniques from the fields of pattern recognition and machine learning in general can be of significant use for such purposes.

The presented work can be further extended by considering other macro level emergent properties including metabolic processes (Bagley et al., 1992), structural and reactive complexity (Adami et al., 2000), self organization (Kauffman, 1993), autonomy and autopoisis (Zeleny, 1981). Associated computational complexity theoretic analysis can be further refined and strengthened by considering classes of models for which most of the parameters have precise bounds compared to the generic analysis presented in this paper.

References

Adami, C., Ofria, C., and Collier, T. C. (2000). Evolution of biological complexity. *Porc. of National Academy of Science*, 97:4463–4468.

Bagley, R. J., Farmer, J. D., and Fontana, W. (1992). Evolution of a metabolism. In *Artificial Life II*, pages 141–158, Redwood City, CA. Addison-Wesley.

Bell, G. (2008). *Selection: The Mechanism of Evolution*. Oxford University Press.

Centler, F., Kaleta, C., di Fenizio, P., and Dittrich, P. (2008). Computing chemical organizations in biological networks. *Bioinformatics*, 24(14):1611.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). *Introduction to Algorithms, Second Edition*. MIT Press, Cambridge, MA, USA.

Henz, M. and Misra, J. (2007). Towards a framework for observing artificial life forms. In *Proc. of the 2007 IEEE Symposium on Artificial Life (IEEE-ALife'07)*, pages 23–30. IEEE Computational Intelligence Society.

Kauffman, S. A. (1993). *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press.

Lenski, R. E., Ofria, C., Pennock, R. T., and Adami, C. (2003). The evolutionary origin of complex features. *Nature*, 423:139–144.

Mahner, M. and Bunge, M. (1997). *Foundations of Biophilosophy*. Springer.

Melkikh, A. V. (2008). Dna computing, computation complexity and problem of biological evolution rate. *Acta Biotheoretica*, 56(4):285 – 295.

Misra, J. (2009). Algorithmic feasibility of entity recognition in artificial life. In *Proc. of the 10th European Conference on Artificial Life (ECAL'09))*, volume 5778 of *LNAI*. Springer (In Press).

Ostrowski, E. A., Ofria, C., and Lenski, R. E. (2007). Ecological specialization and adaptive decay in digital organisms. *American Naturalist*, 169:E1–E20.

Papadimitriou, C. (1994). *Computational Complexity*. Addison-Wesley, Reading, MA.

Paun, G., Rozenberg, G., and Salomaa, A. (2006). *DNA Computing: New Computing Paradigms*. Springer-Verlag.

Sayama, H. (1998). *Constructing Evolutionary Systems on a Simple Deterministic Cellular Automata Space*. PhD thesis, Department of Information Science, Graduate School of Science, University of Tokyo.

Singh, D., Ibrahim, A. M., Yohanna, T., and Singh, J. N. (2007). An overview of the applications of multisets. *Novi Sad Journal of Mathematics*, 37(1):73–92.

Stearns, S. C. and Hoekstra, R. F. (2000). *Evolution–An Introduction*. Oxford University Press.

Stone, R., White, D., Guest, R., and Francis, B. (2009). The Virtual Scylla: an exploration of serious games, artificial life and simulation complexity. *Virtual Reality*, 13(1):13–25.

Zeleny, M., editor (1981). *Autopoiesis: A Theory of Living Organization*. North Holland, New York.

If There Is Something It Is Like to Be Alife, What Is It?

Owen Holland

Sackler Centre for Consciousness Science
University of Sussex
O.E.Holland@sussex.ac.uk

Extended Abstract

In the field of consciousness studies, the phrase 'Is there something it is like to be X?', derived from Nagel's 'What is it like to be a bat?' (Nagel 1974), has become an acceptable way of asking whether X is conscious. It is my contention that this is a question that should be asked in the context of artificial organisms of the type studied in Alife, and especially of physically embodied organisms; the fact that it has so rarely been asked within Alife is perhaps a legacy of the influence of behavior based ideas, which have emptied most such entities of internal representations and processes just as behaviorism banished them from psychology for the best part of a century. However, the question of whether and how some forms of consciousness can be produced in artefacts is the province of the new discipline of machine consciousness, which emerged from outside Alife, and is proceeding independently of it. I wish to bring the two together, and to do so I will ask and answer a slightly different question: if an Alife organism did have a form of consciousness, what would it be like? One of the advantages of asking this particular question is that we can answer objections that certain abilities are impossible (e.g. building and maintaining a world model) by pointing to current work in robotics and AI that demonstrates those abilities.

So what would such a consciousness be like? My claim is that, if it had developed through artificial evolution, it would be very like our own, and in particular it would have many of the same defects, deficiencies, and peculiarities. One problem with making this claim to an audience unfamiliar with the current state of consciousness research is that most people are blissfully unaware of the differences between objective reality and what our consciousness represents to us. I will briefly review the current state of knowledge in respect of this, and I will then show how distortions of time, memory, perception, and voluntary capacity may be the inevitable consequences of the evolution of progressively more capable entities, whether natural or artificial. This will entail a description of how and why world-models and self-models must arise, and of how and to what purpose they might interact.

An enduring problem in the study of consciousness is the explanatory gap – our continuing inability to account for the mental in terms of the physical (Levine 1983). I will not engage directly with this issue, but will instead avoid it by proposing what I call the representational principle of experience: in a system capable of conscious experience, what is experienced must be represented within the system, but not everything represented within the system will or can be experienced (Holland and Marques 2010). One attractive and much discussed possibility is that conscious experience is in some way centered around a model of the physical self. Using the principle, I will present evidence from both robotics and psychology that this, regrettably, is probably not the case.

References

Holland, O. (2007). A Strongly Embodied Approach to Machine Consciousness. *Journal of Consciousness Studies*, Vol. 14, No. 7.

Holland, O. E. and Marques, H. G. (in press). Functional Embodied Imagination and Episodic Memory. To appear in the *International Journal of Machine Consciousness*.

Levine, J. (1983). Materialism and qualia: the explanatory gap. *Pacific Philosophical Quarterly*, 64: 354-361.

Nagel, T. (1974). What Is It Like to Be a Bat?. *Philosophical Review*, 435-50.

Systems Definition of an Organism – A Formal Basis for Modeling Life

Margareta Segerståhl

Department of Biomedical Engineering and Computational Science (BECS),
School of Science and Technology, Aalto University
P. O. Box 12200, FI-00076 AALTO, FINLAND
mes@lce.hut.fi

Extended Abstract

In the study of life, main attention has been on the concrete physical and chemical properties of organisms. The behaviour of living systems has also been extensively studied both empirically and by computer simulations. Sometimes relatively simple rules can produce complex behaviour and patterns – an aspect of life that has been successfully applied to many artificial and engineering systems. But a general understanding is yet to be reached about the rules and conditions that could sufficiently explain the real complexity of life on earth and what distinguishes life from non-living. Currently, a number of competing theories and descriptions exist for the common purpose of defining life. One reason behind this unfortunate situation can be lack of formalism when it comes to defining real living organisms.

Cells are the basic constituent units of biological organisms. Unicellular organisms demonstrate, that a cell can also be an individual exhibiting all the typical descriptive properties of life. Hence, the problem of life is hereby reduced to the problem of understanding what cells are. This idea is far from new as the physical and chemical properties of cells have been extensively studied and used in many theoretical accounts of life and living. Here, however, I take a radically different approach and examine cells from a systems science point of view. This approach produces very different kind of data about more abstract system-level properties of cellular living.

A conceptual examination of real unicellular organisms showed that they typically combine active reproductive living with formation of dormant resistant survival forms. Examples include bacterial quorum sensing as well as differentiation of spores in bacteria and protista. This kind of biphasic life was hence considered to be prototypic and a transition model describing it was formulated. A critical point in the model is the entry into dormancy because it can regulate the trade-off between reproduction efficiency and survival probability. Further examination of the model structure revealed many interesting system properties. The structure provides clues about relevant selection pressures suggesting that complexity increase of living systems happens along two specific system axes. The model is general and formal enough to be applied to various aspect of biological life.

On the basis of this, a formal systems definition of an organism is given. It corresponds to a minimal description of a biphasic transition system. This description is conceptualized as an ideal organism. Ideal formalizations of more complex real organisms can also be derived. The ideal organism concept can be presented, examined and discussed using relatively simple expressions: open form vs. closed form, active state vs. passive state, directed transitions, discrete states, etc. This enables formalization to the point of detaching the conceptual organism from the chemical substance and physical environment of biological life. This may be of interest also to fields that study non-biological complex systems, which nevertheless are often thought to resemble living organisms: trading systems, corporations, as well as human language and societies are some examples.

Author Index

After 23 years of Artificial Life conferences, the hallmark for our community is still its scientific breadth and its inclusiveness. The Artificial Life conferences clearly continue to act as a Big Tent, where scientists from many different disciplines and domains meet to present new results and exchange ideas.

Artificial Life XII consists of more chemistry based contributions than earlier conferences, indicating how the wet (experimental) and the soft (computational) Artificial Life communities increasingly engage with each other. We also see a more general trend towards integration between wet, soft, hard (robotic), and mixed life-like systems, both within the Artificial Life community and across the broader scientific and technological landscapes.

As our community inches closer to an understanding of life as a physical process by *constructing* living processes in different media, we also increasingly assess the technological implications of the ability to engineer systems whose power is based on the core features of life. Such properties include robustness, adaptation, self-repair, self-assembly and self-replication, as well as centralized and distributed intelligence and evolution. Ideas change the world. Increasingly life-like technology is in the process of doing just that as we see the biology-technology boundary starting to blur.

Artificial Life XII is hosted by the Center for Fundamental Living Technology (FLinT), at the University of Southern Denmark (http://www.sdu.dk/flint/).

Cover design by Shïnïto

Artificial Life XII, MIT Press, Cambridge 2010