A Seed-based Plant Propagation Algorithm: The Feeding Station Model

Muhammad Sulaiman*†, and Abdellah Salhi

Department of Mathematical Sciences, University of Essex, Colchester CO4 3SQ, UK

Keywords: Unconstrained global optimization, constrained optimization, engineering problems, seed-based plant propagation

Abstract

The seasonal production of fruit and seeds resembles opening a feeding station, such as a restaurant agents/ customers will arrive at a certain rate and pick fruit (get served) at a certain rate following some appropriate processes. Therefore, dispersion follows the resource process. Modelling this process results in a search/ optimisation algorithm that used dispersion as an exploration tool that, if well captured, will find the optimum of a function over a given search space. This paper presents such an algorithm and tests it on non-trivial problems.

I. Introduction

A variety of plants have evolved in generous ways to propagate. Propagation through seeds is perhaps the most common of them all and one which takes advantage of all sorts of agents ranging from wind to water, to birds and animals. Beside propagation using runners, the strawberry plant uses seeds as well. These seeds are judiciously placed on the surface of a very tasty and brightly coloured fruit, the strawberries, which attract a variety of agents such as birds and animals including humans, which help the propagation.

Plants rely heavily on the dispersion of their seeds to colonise new territories and to improve their survival [22, 21]. There are a lot of studies and models of seed dispersion particularly for trees [1, 2, 8, 21, 22]. Dispersion by wind and ballistic means are probably the most studied of all approaches [18, 52, 53]. However, in the case of the strawberry plant, given the way the seeds stick to the surface of the fruit, Figure(1), [14], dispersion by wind or mechanical means is very limited. Animals, however, and birds in particular are the ideal agents of dispersion [30, 47, 22, 21], in this case.

There are many biologically inspired optimization algorithms in the literature [7, 50]. Flower pollination algorithm (FPA) is inspired by the pollination of flowers through different agents [52], the Swarm data clustering algorithm is inspired by pollination by bees [28], Particle Swarm Optimization (PSO) is inspired by the foraging behavior of a school of fish or a flock of birds, [15, 10], Artificial Bee Colony (ABC) simulates the foraging behavior of honey bees [25, 26], Firefly algorithm is inspired by the flashing fireflies when trying to attract a mate [49, 16], Social Spider Optimization (SSO-C) is inspired by the cooperative behavior of social-spiders [12], to name a few of them.

The Plant Propagation Algorithm (PPA) also known as the strawberry algorithm was inspired by the way plants and specifically the strawberry plants propagate using runners, [40, 43]. The

*sulaiman513@yahoo.co.uk,

†Department of Mathematics, Abdul Wali Khan University Mardan KPK Pakistan.

attraction of PPA is that it can be implemented easily for all sorts of optimization problems. Moreover, it has few algorithm specific arbitrary parameters. PPA follows the principle that plants in good spots with plenty of nutrients will send many short runners. They send few long runners when in nutrient poor spots. Long runners PPA tries to explore the search space while short runners enable the algorithm to exploit the solution space well. It is necessary to make the performance of PPA better, in terms of convergence and efficiency.

In this paper we present a variant of PPA called the Seed-based Plant Propagation Algorithm the feeding station model (SbPPA). The main idea is inspired by the way frugivorous birds disperse the seeds of strawberry. The strawberry plants attract the frugivores and spread its seed for conservation in many habitats through long distances [44]. However, the spatial distribution of seeds depends on the availability of the strawberries on the plants and the number of visits by different agents to eat fruit.

SbPPA is tested on both unconstrained and constrained benchmark problems also used in [29, 12]. Experimental results are presented in Tables 3-4 in terms of best, mean, worst and standard deviation for all algorithms. The paper is organised as follows: In Section II we briefly introduce the feeding station model representing strawberry plants having fruits on them and the main characteristics of paths followed by different agents that disperse the seeds. Section III presents the SbPPA in pseudo code form. The experimental settings, results and convergence graphs for different problems are given in Section IV. In Section V the conclusion and possible future work are given.

II. Aspects of the Feeding Station Model of the Strawberry Plant

Some animals and plants depend on each other to conserve their species [41]. Thus, many plants require, for effective seed dispersal, the visits of frugivorous birds or animals according to a certain distribution, [21, 22, 24, 13].

Seed dispersal by different agents is also called "seed shadow"; this shows the abundance of seeds spread locally or globally around parent plants. In this context, the strawberry feeding station model is divided in two parts: (1) The quantity of fruit or seeds available to agents, or the rate at which the agents will visit the plants, and (2) a probability density, that tells us about the service rate with which the agents are served by the parent plants. This model tells us the quantity of seeds that is spread locally compared to that dispersed globally [23, 32, 17, 33, 5]. There are two aspects that need to be balanced. First exploitation, which is represented by the dispersal of seeds around the parent plants. Secondly, exploration which ensures that the search space is well covered.

As a queuing system [11], there are two basic components to this model: (1) the rate at which agents arrive at the strawberry plants, (2) the rate at which the agents eat fruit and leave the plants to disperse the seeds. The agents arrive at plants in a random process. Assume that during any unit of time, whenever the fruits are available, at most one agent will arrive at a time to the plants, satisfying the orderliness condition. It is further supposed that the probability of arrivals of agents to the plants remain the same for a particular period of time. This means that the arrival rate of agents is higher when there are ripe fruit on the plants and remains the same for a further period when there is no fruit on plants; this is called stationarity condition. The arrival of one agent does not affect the rest of arrivals; this is called independence. Based on these assumptions, we conclude that the probability of arrival of k agents during a cycle t of fruit production by strawberry plants can be denoted by random variable X', [31]. This can be expressed mathematically as

$$P(X' = k) = \frac{(\lambda t)^k e^{-\lambda t}}{k!}, \tag{1}$$

where λ denotes the mean arrival rate of agents per unit time, t is the length of the time interval. On the other hand, the time taken by agents in successfully eating fruit and leaving to disperse its seeds, in other words the service time for agents are expressed by a random variable, which follows the exponential probability distribution [3]. This can be expressed as follows,

$$S(t) = \mu e^{-\mu t}, \tag{2}$$

where μ is the average number of agents that can eat fruit at time t. As some fruit goes to the ground around the plants after becoming fully ripe, this shows that the number of arrivals are less than the fruits available on plants. Mathematically, this can be expressed as the arrival rate of agents is less than the fruits available on all plants, where $\lambda < \mu$.

We assume that the system is in steady state. Let A denote the average number of agents on the plants, and A_q the average number of agents in the queue. If we denote the average number of agents eating fruits by $\frac{\lambda}{\mu}$, then by Little's formulas [36], we have

$$A = A_q + \frac{\lambda}{\mu}, \tag{3}$$

based on Equation (3), we need to maximize the following problem

$$Maximize\ A_q = A - \frac{\lambda}{\mu}, \tag{4}$$

$$\text{subject to} \quad \begin{aligned} g_1(\lambda, \mu) &= \lambda, \mu > 0, \\ g_2(\lambda, \mu) &= \lambda < \mu + 1, \end{aligned} \tag{5}$$

where $A = 10$, which represents the population size in the implementation. The simple limits on the variables are $0 < \lambda, \mu \leq 100$, After solving the problem we get $\lambda = 1.1$, $\mu = 0.1$ and $A_q = 1$.

Moreover, frugivores may travel for a long distance to disperse seeds far away from parent SP; in doing so, they obey a Lévy distribution [45, 46, 38].

II.1 Lévy distribution

Randomization in metaheuristics is generally achieved by utilizing pseudorandom numbers, in light of some regular stochastic methodologies. Lévy distributions is one of the probability density distributions for random variables. Here the random variables represent the directions of arbitrary flights by frugivores. This function of random variables ranges over real numbers with a domain called "search space".

The flight lengths of the agents served by SP, is assumed to be a heavy tailed power law distribution represented by,

$$L(s) \sim |s|^{-1-\beta}, \tag{6}$$

where $L(s)$ denotes the Lévy distribution with index $\beta \in (0, 2)$.

Lévy flights are a unique arbitrary excursions whose step lengths are drawn from (6). Another form of Lévy distribution can be written as,

$$L(s, \gamma, \mu) = \begin{cases} \sqrt{\frac{\gamma}{2\pi}} exp\left[-\frac{\gamma}{2(s-\mu)}\right]\left(\frac{1}{(s-\mu)}\right)^{\frac{3}{2}}, & 0 < \mu < s < \infty \\ 0 & Otherwise, \end{cases} \tag{7}$$

this implies that

$$\lim_{s\to\infty} L(s,\gamma,\mu) = \sqrt{\frac{\gamma}{2\pi}}\left(\frac{1}{s}\right)^{\frac{3}{2}}, \tag{8}$$

In terms of Fourier transform [50] the limiting value of $L(s)$ can be written as under,

$$\lim_{s\to\infty} L(s) = \frac{\alpha\beta\Gamma(\beta)\sin(\frac{\pi\beta}{2})}{\pi|s|^{1+\beta}}, \tag{9}$$

where $\Gamma(\beta)$ is the Gamma function defined by

$$\Gamma(\beta) = \int_0^\infty x^{\beta-1}e^{-x}dx. \tag{10}$$

The steps are generated by using Mantegna's algorithm. This algorithm ensures the behaviour of Lévy flights to be symmetric and stable as shown in Figure (3b).

III. Strawberry Plant Propagation Algorithm: The Feeding Station Model

The Plant Propagation Algorithm (PPA), recently developed in [40, 43], emulates the way strawberry plants (SP) propagate by runners. Here we considered the propagation through seeds. The main objective of SbPPA is the optimal reproduction of new plants through seeds dispersion, by using different dispersal means.

We assume that the arrival of different agents to the plants for eating fruits, is according to Poisson distribution. The mean arrival rate $\lambda = 1.1$, and $NP = 10$ is the total number of agents in our population. Let $k = 1, 2, \ldots, A$ be the number of agents visiting the plants per unit time. By using these assumptions we get Figure (2) according to Equation (1).

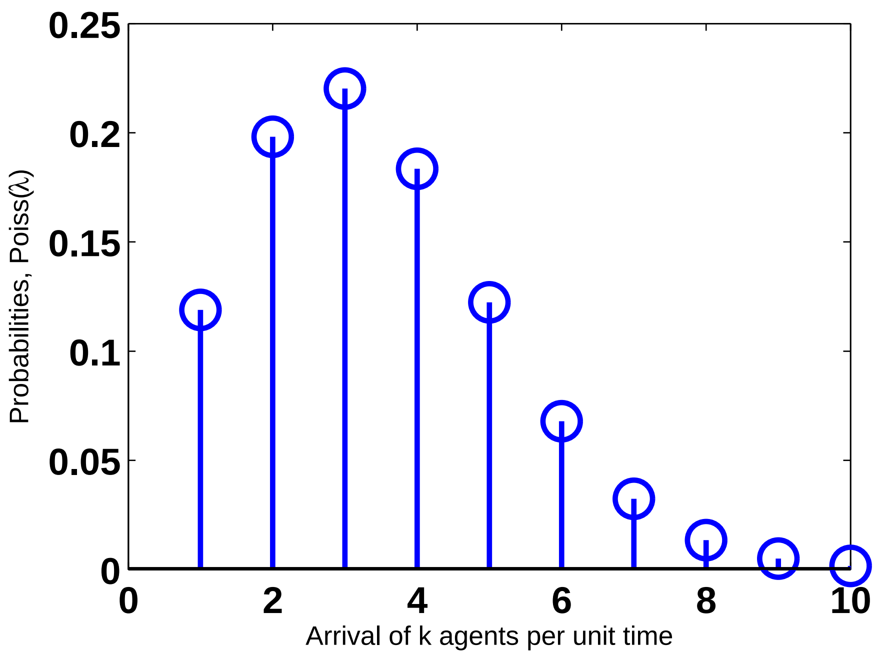

Figure 2: Agents arrival at strawberry plants to eat fruit and disperse seed

(a) Strawberry fruit with seeds

(b) Strawberry garden flower

(c) A fruit eaten by bird(s)

(d) A bird eating strawberries

(e) Strawberry plants spreading seed and sending runners around them

Figure 1: Strawberry plant propagation: through seed dispersion [48, 39, 37, 35]

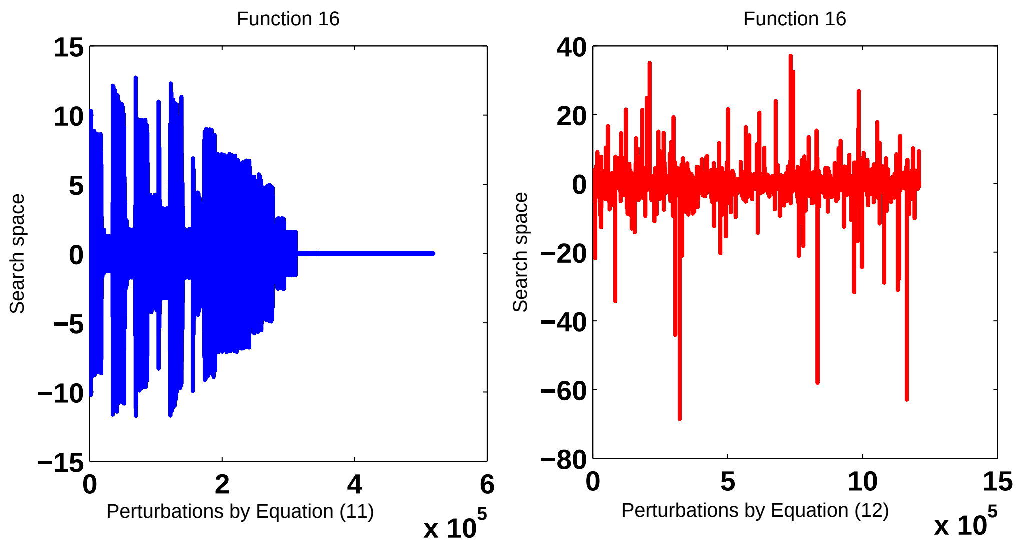

Figure 3: Overall performance of SbPPA on problem 16

The probability $Poiss(\lambda) < 0.05$ means that, the chances for seeds to be taken far away from SP, are lower and the propagation is supported either by runners or seeds fallen down from plants. In this case, Equation (11) below is used, which is helping the algorithm to exploit the search space,

$$x^*_{i,j} = \begin{cases} x_{i,j} + \xi_j(x_{i,j} - x_{l,j}) & \text{if } PR \leq 0.8 \\ x_{i,j} & Otherwise, \end{cases} \tag{11}$$

where PR denotes the perturbation rate and it tunes the intensity of displacements by which the seeds will be dispersed locally around the SP, $x^*_{i,j}, x_{i,j} \in [a_j \; b_j]$ are the j^{th} coordinates of the seeds X_i and X^*_i respectively, a_j and b_j are the j^{th} lower and upper bounds defining the search space of the problem and $\xi_j \in [-1 \; 1]$. The indices l and i are mutually exclusive.

On the other hand, if $Poiss(\lambda) \geq 0.05$ (we choose 0.05 to give more weight to global dispersion), here the complete role of global dispersion is played by seeds, this is implemented by using the following equation,

$$x^*_{i,j} = \begin{cases} x_{i,j} + \mathrm{L_i}(x_{i,j} - \theta_j) & \text{if } PR \leq 0.8, \; \theta_j \in [a_j \; b_j] \\ x_{i,j} & Otherwise. \end{cases} \tag{12}$$

Here $\mathrm{L_i}$ is a step drawn from the Lévy distribution [50], θ_j is a random coordinate within the search space. The effects on the current solutions due to perturbations applied by Equation (11) and Equation (12) are shown in Figure (3).

As mentioned in the pseudo-code of SbPPA, we first collect best solutions from the first NP trial runs to form a population of potentially good solutions denoted by pop_{best}. The convergence rate of SbPPA, is shown in Figures (4-5), for different test problems used in our experiments. The statistical results best, worst, mean and standard deviation are calculated based on pop_{best}.

The seed based propagation process of SP can be represented in the following steps:

1. The dispersal of seeds or the propagation by runners in the neighbourhood of the SP, as shown in Figure 1_e, is carried out either by fruit fallen from strawberry plants after they become ripe or by runners. The step lengths for this phase are calculated using Equation (11).

2. Seeds are spread globally through frugivores, as shown in Figure $1_{c,d}$. The step lengths for those travelling agents are drawn from the Lévy distribution.

3. The probabilities, $Poiss(\lambda)$, that a certain amount k of agents will arrive to SP to eat fruits and disperse it, is used as a switch between global and local search.

For implementation, we assume that each SP produces one fruit, and each fruit is assumed to have one seed, we mean by a solution X_i the position of the i^{th} seed to be dispersed. The number of seeds in the population is denoted by NP. Initially we generate a random population of NP seeds using Equation (13),

$$x_{i,j} = a_j + (b_j - a_j)\eta_j, j = 1, ..., n \tag{13}$$

where $x_{i,j} \in [a_j, b_j]$ is the j^{th} entry of solution X_i, a_j and b_j are the j^{th} coordinates of the bounds describing the search space of the problem and $\eta_j \in (0, 1)$. This means $X_i = [x_{i,j}]$, for $j = 1, ..., n$ represents the position of the i^{th} seed in population pop.

IV. Experimental Setting And Discussion

In our experiments we test SbPPA against other state-of-the-art algorithms. Our set of test problems include benchmark constrained and unconstrained optimization problems [42, 34, 12]. The results are compared in terms of best, worst, mean and standard deviations obtained by SbPPA, ABC [25, 27], PSO [20], FF [16], HPA [29] and SSO-C [12]. The detailed descriptions of these problems are given in Appendix I. The significance of results are shown according to the following notations:

- (+) when SbPPA is better
- (-) when SbPPA is worse
- ($\approx$) when the results are approximately same as SbPPA.

IV.1 Parameter Settings

The parameter settings are give in Table 1-2:

Algorithm 1 **Seed-based Plant Propagation Algorithm (SbPPA): The Feeding Station Model**

Initialize: $g_{max} \leftarrow$ maximum number of generations, $max_{eval} \leftarrow$ maximum function evaluations, $r \leftarrow$ counter for trial runs
Set $r = 1$
if $r \leq NP$ **then**
  Create a random population of seeds $pop = \{X_i \mid i = 1, 2, ..., NP\}$, using Equation (13) and add the best solutions from each trial run, in pop_{best}.
  Evaluate the population.
end if
while $r > NP$ **do**
  Use population pop_{best}.
end while
Set $ngen = 1$,
while $(ngen < g_{max})$ **or** $(n_{eval} < max_{eval})$ **do**
  for $i = 1$ to NP **do**
    if $Poiss(\lambda)_i \geq 0.05$ **then**, ▷ (Global or local seed dispersion)
      for $j = 1$ to n **do** ▷ (n is number of dimensions)
        if rand$\leq PR$ **then**, ▷ (PR=Perturbation rate)
          Update the current entry according to Equation (12)
        end if
      end for
    else
      for $j = 1$ to n **do**
        if rand$\leq PR$ **then**,
          Update the current entry according to Equation (11)
        end if
      end for
    end if
  end for
end while
Return: Update current population.

Table 1: Parameters used for each algorithm for solving unconstrained global optimization problems $f_1 - f_{10}$, All experiments are repeated 30 times.

PSO [15, 29]	ABC [25, 29]	HPA [29]	SbPPA
M=100	SN=100	Agents=100	NP=10
$G_{max} = \frac{(Dimension \times 20{,}000)}{NP}$	MCN=$\frac{(Dimension \times 20{,}000)}{NP}$	Iteration number=$\frac{(Dimension \times 20{,}000)}{NP}$	Iteration number=$\frac{(Dimension \times 20{,}000)}{NP}$
$c_1 = 2$	MR=0.8	$c_1 = 2$	PR=0.8, $Poiss(\lambda) = 0.05$
$c_2 = 2$	limit=$\frac{(SN \times dimension)}{2}$	$c_2 = 2$	$k = 1, 2, \ldots, A$
W= $\frac{(G_{max} - iteration_{index})}{G_{max}}$	-	limit=$\frac{(SN \times dimension)}{2}$	$\lambda = 1.1$
-	-	W= $\frac{(G_{max} - iteration_{index})}{G_{max}}$	-

Table 2: Parameters used for each algorithm for solving constrained optimization problems $f_{11} - f_{18}$, All experiments are repeated 30 times.

PSO [20]	ABC [27]	FF [16]	SSO-C [12]	SbPPA
M=250	sn=40	Fireflies=25	N=50	NP=10
$G_{max} = 300$	MCN=6000	Iteration number= 2000	Iteration number=500	Iteration number=2400
$c_1 = 2$	MR=0.8	q=1.5	PF=0.7	PR=0.8, $Poiss(\lambda) = 0.05$
$c_2 = 2$	-	$\alpha = 0.001$	-	$k = 1, 2, \ldots, A$
Weight factors= 0.9 to 0.4	-	-	-	$\lambda = 1.1$

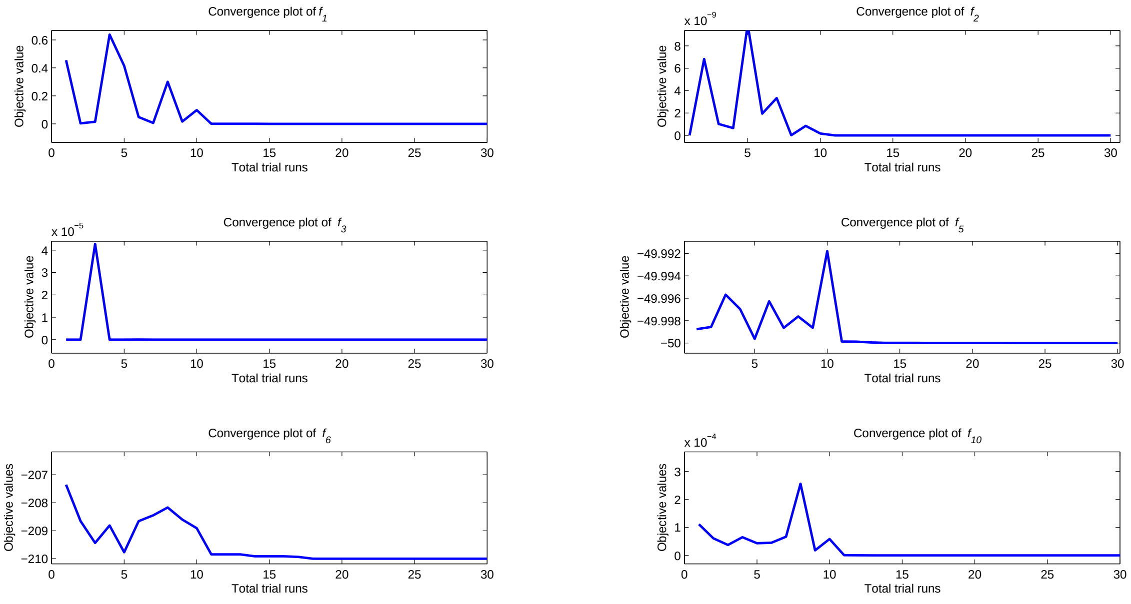

Figure 4: Performance of SbPPA on unconstrained global optimization problems

Table 3: Results obtained by SbPPA, HPA, PSO and ABC. All problems in this table are unconstrained.

Fun	Dim	Algorithm	Best	Worst	Mean	SD
1	4	ABC	(+) 0.0129	(+) 0.6106	(+) 0.1157	(+) 0.111
		PSO	(-) 6.8991E-08	(+) 0.0045	(+) 0.001	(+) 0.0013
		HPA	(+) 2.0323E-06	(+) 0.0456	(+) 0.009	(+) 0.0122
		SbPPA	1.08E-07	7.05E-06	3.05E-06	3.14E-06
2	2	ABC	(+) 1.2452E-08	(+) 8.4415E-06	(+) 1.8978E-06	(+) 1.8537E-06
		PSO	(≈) 0	(≈) 0	(≈) 0	(≈) 0
		HPA	(≈) 0	(≈) 0	(≈) 0	(≈) 0
		SbPPA	0	0	0	0
3	2	ABC	(≈) 0	(+) 4.8555E-06	(+) 4.1307E-07	(+) 1.2260E-06
		PSO	(≈) 0	(+) 3.5733E-07	(+) 1.1911E-08	(+) 6.4142E-08
		HPA	(≈) 0	(≈) 0	(≈) 0	(≈) 0
		SbPPA	0	0	0	0
4	2	ABC	(≈) -1.03163	(≈) -1.03163	(≈) -1.03163	(≈) 0
		PSO	(≈) -1.03163	(≈) -1.03163	(≈) -1.03163	(≈) 0
		HPA	(≈) -1.03163	(≈) -1.03163	(≈) -1.03163	(≈) 0
		SbPPA	-1.031628	-1.031628	-1.031628	0
5	6	ABC	(≈) -50.0000	(≈) -50.0000	(≈) -50.0000	(-) 0
		PSO	(≈) -50.0000	(≈) -50.0000	(≈) -50.0000	(-) 0
		HPA	(≈) -50.0000	(≈) -50.0000	(≈) -50.0000	(-) 0
		SbPPA	-50.0000	-50.0000	-50.0000	5.88E-09
6	10	ABC	(+) -209.9929	(+) -209.8437	(+) -209.9471	(+) 0.044
		PSO	(≈) -210.0000	(≈) -210.0000	(≈) -210.0000	(-) 0
		HPA	(≈) -210.0000	(≈) -210.0000	(≈) -210.0000	(+) 1
		SbPPA	-210.0000	-210.0000	-210.0000	4.86E-06
7	30	ABC	(+) 2.6055E-16	(+) 5.5392E-16	(+) 4.7403E-16	(+) 9.2969E-17
		PSO	(≈) 0	(≈) 0	(≈) 0	(≈) 0
		HPA	(≈) 0	(≈) 0	(≈) 0	(≈) 0
		SbPPA	0	0	0	0
8	30	ABC	(+) 2.9407E-16	(+) 5.5463E-16	(+) 4.8909E-16	(+) 9.0442E-17
		PSO	(≈) 0	(≈) 0	(≈) 0	(≈) 0
		HPA	(≈) 0	(≈) 0	(≈) 0	(≈) 0
		SbPPA	0	0	0	0
9	30	ABC	(≈) 0	(+) 1.1102E-16	(+) 9.2519E-17	(+) 4.1376E-17
		PSO	(≈) 0	(+) 1.1765E-01	(+) 2.0633E-02	(+) 2.3206E-02
		HPA	(≈) 0	(≈) 0	(≈) 0	(≈) 0
		SbPPA	0	0	0	0
10	30	ABC	(+) 2.9310E-14	(+) 3.9968E-14	(+) 3.2744E-14	(+) 2.5094E-15
		PSO	(≈) 7.9936E-15	(+) 1.5099E-14	(-) 8.5857E-15	(+) 1.8536E-15
		HPA	(≈) 7.9936E-15	(+) 1.5099E-14	(+) 1.1309E-14	(+) 3.54E-15
		SbPPA	7.994E-15	7.99361E-15	7.994E-15	7.99361E-15

Table 4: Results obtained by SbPPA, PSO, ABC, FF and SSO-C. All problems in this table are standard constrained optimization problems

Fun	Fun Name	Optimal	Algorithm	Best	Mean	Worst	SD
11	CP1	-15	PSO	(≈) -15	(≈) -15	(≈) -15	(-) 0
			ABC	(≈) -15	(≈) -15	(≈) -15	(-) 0
			FF	(+) 14.999	(+) 14.988	(+) 14.798	(+) 6.40E-07
			SSO-C	(≈) -15	(≈) -15	(≈) -15	(-) 0
			SbPPA	-15	-15	-15	1.95E-15
12	CP2	-30665.539	PSO	(≈) -30665.5	(+) -30662.8	(+) -30650.4	(+) 5.20E-02
			ABC	(≈) -30665.5	(+) -30664.9	(+) -30659.1	(+) 8.20E-02
			FF	(≈) -3.07E+04	(+) -30662	(+) -30649	(+) 5.20E-02
			SSO-C	(≈) -3.07E+04	(≈) -30665.5	(+) -30665.1	(+) 1.10E-04
			SbPPA	-30665.5	-30665.5	-30665.5	2.21E-06
13	CP3	-6961.814	PSO	(+) -6.96E+03	(+) -6958.37	(+) -6942.09	(+) 6.70E-02
			ABC	(-) -6961.81	(+) -6958.02	(+) -6955.34	(-) 2.10E-02
			FF	(+) -6959.99	(+) -6.95E+03	(+) -6947.63	(-) 3.80E-02
			SSO-C	(-) -6961.81	(+) -6961.01	(+) -6960.92	(-) 1.10E-03
			SbPPA	-6961.5	-6961.38	-6961.45	0.043637
14	CP4	24.306	PSO	(-) 24.327	(+) 2.45E+01	(+) 24.843	(+) 1.32E-01
			ABC	(+) 24.48	(+) 2.66E+01	(+) 28.4	(+) 1.14
			FF	(-) 23.97	(+) 28.54	(+) 30.14	(+) 2.25
			SSO-C	(-) 24.306	(-) 24.306	(-) 24.306	(-) 4.95E-05
			SbPPA	24.34442	24.37536	24.37021	0.012632
15	CP5	-0.7499	PSO	(≈) -0.7499	(+) -0.749	(+) -0.7486	(+) 1.20E-03
			ABC	(≈) -0.7499	(+) -0.7495	(+) -0.749	(+) 1.67E-03
			FF	(+) -0.7497	(+) -0.7491	(+) -0.7479	(+) 1.50E-03
			SSO-C	(≈) -0.7499	(≈) -0.7499	(≈) -0.7499	(-) 4.10E-09
			SbPPA	0.7499	0.749901	0.7499	1.66E-07
16	Spring Design Problem	Not Known	PSO	(+) 0.012858	(+) 0.014863	(+) 0.019145	(+) 0.001262
			ABC	(≈) 0.012665	(+) 0.012851	(+) 0.01321	(+) 0.000118
			FF	(≈) 0.012665	(+) 0.012931	(+) 0.01342	(+) 0.001454
			SSO-C	(≈) 0.012665	(+) 0.012765	(+) 0.012868	(+) 9.29E-05
			SbPPA	0.012665	0.012666	0.012666	3.39E-10
17	Welded Beam Design Problem	Not Known	PSO	(+) 1.846408	(+) 2.011146	(+) 2.237389	(+) 0.108513
			ABC	(+) 1.798173	(+) 2.167358	(+) 2.887044	(+) 0.254266
			FF	(+) 1.724854	(+) 2.197401	(+) 2.931001	(+) 0.195264
			SSO-C	(≈) 1.724852	(+) 1.746462	(+) 1.799332	(+) 0.02573
			SbPPA	1.724852	1.724852	1.724852	4.06E-08
18	Speed Reducer Design Optimization	Not Known	PSO	(+) 3044.453	(+) 3079.262	(+) 3177.515	(+) 26.21731
			ABC	(+) 2996.116	(+) 2998.063	(+) 3002.756	(+) 6.354562
			FF	(+) 2996.947	(+) 3000.005	(+) 3005.836	(+) 8.356535
			SSO-C	(≈) 2996.113	(≈) 2996.113	(≈) 2996.113	(+) 1.34E-12
			SbPPA	2996.114	2996.114	2996.114	0

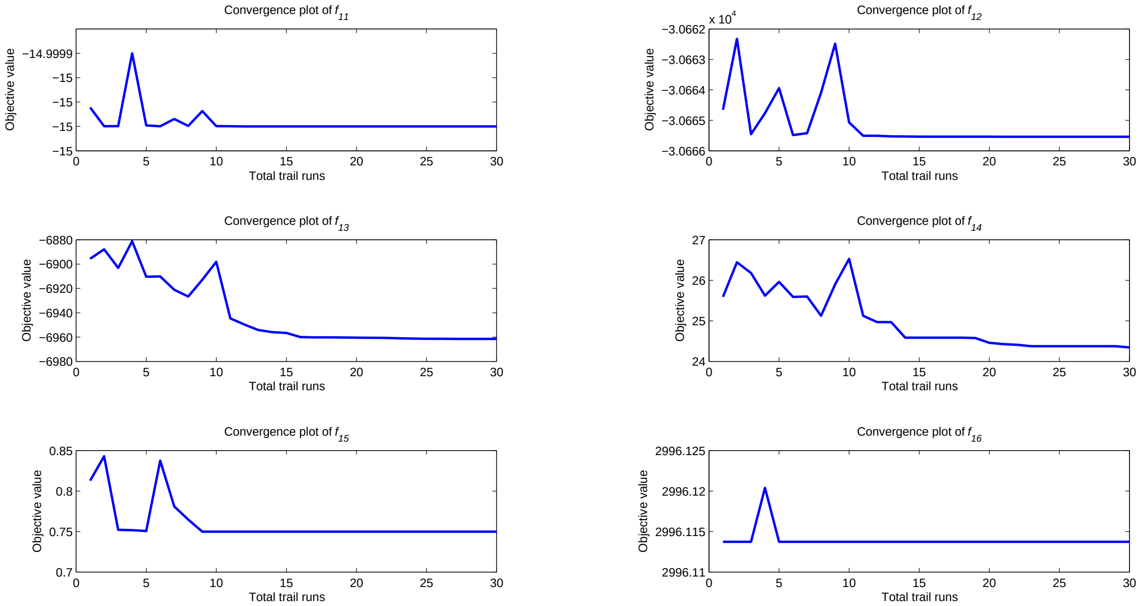

Figure 5: Performance of SbPPA on constrained global optimization problems. The problems solved in this table are standard constrained optimization problems

V. Conclusion

A new algorithm mimicking the seed-based plant propagation (SbPPA) is designed and implemented for both unconstrained and constrained optimization problems. The performance of SbPPA is compared with a number of well established algorithms. The results are compiled in terms of best, mean, worst and standard deviation. SbPPA is very easy to implement as it needs less arbitrary parameter settings. An alternative strategy is adopted to update our current population. The effects on convergence are shown through convergence plots, Figures (4-5), of some of the solved problems. Note that the success rate of SbPPA depends on the quality of the initial population. SbPPA is being tested on discrete real world problems.

VI. Acknowledgments

This work is supported by Abdul Wali Khan University, Mardan, Pakistan, Grant No. F.16-5/ P& D/ AWKUM /238.

I. Appendix

I. Set of Unconstrained Global Optimization Problems

Table 5: Unconstrained Global Optimization Problems Used In Our Experiments.

Fun	Ftn. Name	D	C	Range	Min	Formulation
f_1	Colville	4	UN	[-10 10]	0	$f(x) = 100(x_1^2 - x_2) + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1)$
f_2	Matyas	2	UN	[-10 10]	0	$f(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$
f_3	Schaffer	2	MN	[-100 100]	0	$f(x) = 0.5 + \frac{\sin^2(\sqrt{\sum_{i=1}^n x_i^2}) - 0.5}{(1+0.001(\sum_{i=1}^n x_i^2))^2}$
f_4	Six Hump Camel Back	2	MN	[-5 5]	-1.03163	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$
f_5	Trid6	6	UN	[-36 36]	-50	$f(x) = \sum_{i=1}^{6}(x_i - 1)^2 - \sum_{i=2}^{6} x_i x_{i-1}$
f_6	Trid10	10	UN	[-100 100]	-210	$f(x) = \sum_{i=1}^{10}(x_i - 1)^2 - \sum_{i=2}^{10} x_i x_{i-1}$
f_7	Sphere	30	US	[-100 100]	0	$f(x) = \sum_{i=1}^{n} x_i^2$
f_8	SumSquares	30	US	[-10 10]	0	$f(x) = \sum_{i=1}^{n} i x_i^2$
f_9	Griewank	30	MN	[-600 600]	0	$f(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$
f_{10}	Ackley	30	MN	[-32 32]	0	$f(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e$

II. Set of Constrained Global Optimization Problems Used in Our Experiments

II.1 CP1

$$
\begin{aligned}
\text{Min} \quad & f(x) = 5\textstyle\sum_{d=1}^{4} x_d - 5\sum_{d=1}^{4} x_d^2 - \sum_{d=5}^{13} x_d \\
\text{subject to} \quad & g_1(x) = 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \le 0 \\
& g_2(x) = 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \le 0 \\
& g_3(x) = 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \le 0 \\
& g_4(x) = -8x_1 + x_{10} \le 0 \\
& g_5(x) = -8x_2 + x_{11} \le 0 \\
& g_6(x) = -8x_3 + x_{12} \le 0 \\
& g_7(x) = -2x_4 - x_5 + x_{10} \le 0 \\
& g_8(x) = -2x_6 - x_7 + x_{11} \le 0 \\
& g_9(x) = -2x_8 - x_9 + x_{12} \le 0,
\end{aligned}
$$

where bounds are $0 \le x_i \le 1$ $(i = 1, ..., 9, 13)$, $0 \le x_i \le 100$ $(i = 10, 11, 12)$. The global optimum is at $x^* = (1, 1, 1, 1, 1, 1, 1, 1, 1, , 3, 3, 3, 1), f(x^*) = -15$.

II.2 CP2

$$
\begin{aligned}
\text{Min} \quad & f(x) = 5.3578547x_2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141 \\
\text{subject to} \quad & g_1(x) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 - 92 \le 0 \\
& g_2(x) = -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5 \le 0 \\
& g_3(x) = 80.51249 + 0.0071317x2x5 + 0.0029955x1x2 - 0.0021813x_2 - 110 \le 0 \\
& g_4(x) = -80.51249 - 0.0071317x_2x_5 + 0.0029955x_1x_2 - 0.0021813x_2 + 90 \le 0 \\
& g_5(x) = 9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 - 25 \le 0 \\
& g_6(x) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \le 0,
\end{aligned}
$$

where $78 \le x_1 \le 102$, $33 \le x_2 \le 45$, $27 \le x_i \le 45$ $(i = 3,4,5)$. The optimum solution is $x^* = (78, 33, 29.995256025682, 45, 36.775812905788)$, where $f(x^*) = -30665.539$. Constraints g_1 and g_6 are active.

II.3 CP3

$$
\begin{aligned}
\text{Min} \quad & f(x) = (x_1 - 10)^3 + (x_2 - 20)^3 \\
\text{subject to} \quad & g_1(x) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \le 0 \\
& g_2(x) = (x_1 - 6)^2 + (x^2 - 5)^2 - 82.81 \le 0,
\end{aligned}
$$

where $13 \le x_1 \le 100$ and $0 \le x_2 \le 100$. The optimum solution is $x^* = (14.095, 0.84296)$ where $f(x^*) = -6961.81388$. Both constraints are active.

II.4 CP4

$$
\begin{aligned}
\text{Min} \quad & f(x) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 \\
& \quad + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45 \\
\text{subject to} \quad & g_1(x) = -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \le 0 \\
& g_2(x) = 10x_1 - 8x_2 - 17x_7 + 2x_8 \le 0 \\
& g_3(x) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \le 0 \\
& g_4(x) = 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \le 0 \\
& g_5(x) = 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x^4 - 40 \le 0 \\
& g_6(x) = x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \le 0 \\
& g_7(x) = 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \le 0 \\
& g_8(x) = -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \le 0,
\end{aligned}
$$

where $-10 \le x_i \le 10$ $(i = 1, ..., 10)$. The global optimum is
$x^* = (2.171996, 2.363683, 8.773926, 5.095984, 0.9906548, 1.430574, 1.321644, 9.828726, 8.280092, 8.375927)$,
where $f(x^*) = 24.3062091$. Constraints g_1, g_2, g_3, g_4, g_5 and g_6 are active.

II.5 CP5

$$\begin{aligned} \text{Min} \quad & f(x) = x_1^2 + (x_2 - 1)^2 \\ \text{subject to} \quad & g_1(x) = x_2 - x_1^2 = 0, \end{aligned}$$

where $1 \leq x_1 \leq 1, 1 \leq x_2 \leq 1$. The optimum solution is $x^* = (\pm 1/\sqrt{(2)}, 1/2)$, where $f(x^*) = 0.7499$.

II.6 Welded Beam Design Optimisation

The welded beam design is a standard test problem for constrained design optimisation [9, 51]. There are four design variables: the width w and length L of the welded area, the depth d and thickness h of the main beam. The objective is to minimise the overall fabrication cost, under the appropriate constraints of shear stress τ, bending stress σ, buckling load P and maximum end deflection δ. The optimization model is summarized as follows, where $x^T = (w, L, d, h)$.

$$\textit{Minimise} \quad f(x) = 1.10471 w^2 L + 0.04811 dh(14.0 + L), \tag{14}$$

subject to

$$\begin{aligned} g_1(x) &= w - h \leq 0, \\ g_2(x) &= \delta(x) - 0.25 \leq 0, \\ g_3(x) &= \tau(x) - 13,600 \leq 0, \\ g_4(x) &= \sigma(x) - 30,000 \leq 0, \\ g_5(x) &= 1.10471 w^2 + 0.04811 dh(14.0 + L) - 5.0 \leq 0, \\ g_6(x) &= 0.125 - w \leq 0, \\ g_7(x) &= 6000 - P(x) \leq 0, \end{aligned} \tag{15}$$

where

$$\begin{aligned} \sigma(x) &= \frac{504,000}{hd^2}, & & \\ D &= \frac{1}{2}\sqrt{L^2 + (w+d)^2}, & Q &= 6000\left(14 + \frac{L}{2}\right), \\ \delta &= \frac{65,856}{30,000hd^3}, & J &= \sqrt{2}wL\left(\frac{L^2}{6} + \frac{(w+d)^2}{2}\right), \\ \alpha &= \frac{6000}{\sqrt{2}wL}, & \beta &= \frac{QD}{J}, \\ P &= 0.61423 \times 10^6 \frac{dh^3}{6}\left(1 - \frac{\sqrt[d]{\frac{30}{48}}}{28}\right). & \tau(x) &= \sqrt{\alpha^2 + \frac{\alpha\beta L}{D} + \beta^2}. \end{aligned} \tag{16}$$

II.7 Speed Reducer Design Optimization

The problem of designing a speed reducer [19] is a standard test problem. It consists of the design variables as: face width x_1, module of teeth x_2, number of teeth on pinion x_3, length of the first shaft between bearings x_4, length of the second shaft between bearings x_5, diameter of the first shaft x_6, and diameter of the first shaft x_7 (all variables continuous except x_3 that is integer). The weight of the speed reducer is to be minimized subject to constraints on bending stress of the gear teeth, surface stress, transverse deflections of the shafts and stresses in the shaft, [9]. The mathematical formulation of the problem, where $x^T = (x_1, x_2, x_3, x_4, x_5, x_6, x_7)$, is as follows.

$$\begin{aligned} \textit{Minimise} \quad f(x) =& 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 43.0934) \\ &- 1.508x_1(x_6^2 + x_7^3) + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2), \end{aligned} \tag{17}$$

subject to

$$\begin{aligned} g_1(x) &= \frac{27}{x_1x_2^2x_3} - 1 \le 0, \\ g_2(x) &= \frac{397.5}{x_1x_2^2x_3^2} - 1 \le 0, \\ g_3(x) &= \frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \le 0, \\ g_4(x) &= \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \le 0, \\ g_5(x) &= \frac{1.0}{110x_6^3}\sqrt{\left(\frac{745.0x_4}{x_2x_3}\right)^2 + 16.9 \times 10^6} - 1 \le 0, \\ g_6(x) &= \frac{1.0}{85x_7^3}\sqrt{\left(\frac{745.0x_5}{x_2x_3}\right)^2 + 157.5 \times 10^6} - 1 \le 0, \\ g_7(x) &= \frac{x_2x_3}{40} - 1 \le 0, \\ g_8(x) &= \frac{5x_2}{x_1} - 1 \le 0, \\ g_9(x) &= \frac{x_1}{12x_2} - 1 \le 0, \\ g_{10}(x) &= \frac{1.5x_6 + 1.9}{x_4} - 1 \le 0, \\ g_{11}(x) &= \frac{1.1x_7 + 1.9}{x_5} - 1 \le 0. \end{aligned} \tag{18}$$

The simple limits on the design variables are
$2.6 \le x_1 \le 3.6$, $0.7 \le x_2 \le 0.8$,
$17 \le x_3 \le 28$, $7.3 \le x_4 \le 8.3$, $7.8 \le x_5 \le 8.3$,
$2.9 \le x_6 \le 3.9$ and $5.0 \le x_7 \le 5.5$.

II.8 Spring Design Optimisation

The main objective of this problem [4, 6] is to minimize the weight of a tension/compression string, subject to constraints of minimum deflection, shear stress, surge frequency, and limits on outside diameter and on design variables. There are three design variables: the wire diameter x_1, the mean coil diameter x_2, and the number of active coils x_3, [9]. The mathematical formulation of this problem, where $x^T = (x_1, x_2, x_3)$, is as follows.

$$\textit{Minimize } f(x) = (x_3 + 2)x_2x_1^2, \tag{19}$$

subject to

$$
\begin{aligned}
g_1(x) &= 1 - \frac{x_2^3 x_3}{7,178 x_1^4} \leq 0, \\
g_2(x) &= \frac{4x_2^2 - x_1 x_2}{12,566(x_2 x_1^3) - x_1^4} + \frac{1}{5,108 x_1^2} - 1 \leq 0, \\
g_3(x) &= 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0, \\
g_4(x) &= \frac{x_2 + x_1}{1.5} - 1 \leq 0.
\end{aligned} \tag{20}
$$

The simple limits on the design variables are $0.05 \leq x_1 \leq 2.0$, $0.25 \leq x_2 \leq 1.3$ and $2.0 \leq x_3 \leq 15.0$.

References

[1] Warren G Abrahamson et al. *Plant-animal interactions.* McGraw Hill Inc., 1989.

[2] Alan N Andersen and Richard W Braithwaite. Plant-animal interactions. In *Landscape and vegetation ecology of the Kakadu Region, northern Australia*, pages 137–154. Springer, 1996.

[3] Alfredo HS Ang and Wilson H Tang. Probability concepts in engineering. *Planning*, 1(4):1–3, 2004.

[4] JS. Arora. *Introduction to optimum design*. Academic Press, 2004.

[5] Carol K Augspurger and Susan E Franson. Wind dispersal of artifical fruits varying in mass, area, and morphology. *Ecology*, pages 27–42, 1987.

[6] A. D. Belegundu and JS Arora. A study of mathematical programming methods for structural optimization. part i: Theory. *International Journal for Numerical Methods in Engineering*, 21(9):1583–1599, 1985.

[7] J. Brownlee. *Clever algorithms: nature-inspired programming recipes*. Jason, B., 2011.

[8] John P Bryant. Plant-animal interactions. *Environmental Entomology*, 19(4):1169–1170, 1990.

[9] L. C. Cagnina, S. C. Esquivel, and C. A. C. Coello. Solving engineering optimization problems with the simple constrained particle swarm optimizer. *Informatica (Slovenia)*, 32(3):319–326, 2008.

[10] Maurice Clerc. *Particle swarm optimization*, volume 93. John Wiley & Sons, 2010.

[11] Robert B Cooper. Introduction to queueing theory. 1972.

[12] Erik Cuevas and Miguel Cienfuegos. A new algorithm inspired in the behavior of the social-spider for constrained optimization. *Expert Systems with Applications: An International Journal*, 41(2):412–425, 2014.

[13] M Debussche and P Isenmann. Bird-dispersed seed rain and seedling establishment in patchy mediterranean vegetation. *Oikos*, 69(3):414–426, 1994.

[14] Hennie J du Plessis, Reon J Brand, Christina Glyn-Woods, and Marlene A Goedhart. Efficient genetic transformation of strawberry (fragaria x ananassa duch.) cultivar selekta. In *III International Symposium on In Vitro Culture and Horticultural Breeding 447*, pages 289–294, 1996.

[15] R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on micro machine and human science*, pages 39–43. IEEE, 1995.

[16] Amir Hossein Gandomi, Xin-She Yang, and Amir Hossein Alavi. Mixed variable structural optimization using firefly algorithm. *Computers & Structures*, 89(23):2325–2336, 2011.

[17] SAH Geritz, TJ De Jong, and PGL Klinkhamer. The efficacy of dispersal in relation to safe site area and seed production. *Oecologia*, 62(2):219–221, 1984.

[18] Beverly J Glover. *Understanding flowers and flowering: an integrated approach*. Oxford University Press Oxford, 2007.

[19] J. Golinski. An adaptive optimization system applied to machine synthesis. *Mechanism and Machine Theory*, 8(4):419–436, 1974.

[20] Qie He and Ling Wang. A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. *Applied Mathematics and Computation*, 186(2):1407–1422, 2007.

[21] Carlos M Herrera. Seed dispersal by vertebrates. *Plant–animal interactions: an evolutionary approach*, pages 185–208, 2002.

[22] Carlos M Herrera and Olle Pellmyr. *Plant animal interactions: an evolutionary approach*. John Wiley & Sons, 2009.

[23] Daniel H Janzen. Herbivores and the number of tree species in tropical forests. *American naturalist*, pages 501–528, 1970.

[24] Pedro Jordano et al. Fruits and frugivory. *Seeds: the ecology of regeneration in plant communities*, 2:125–166, 2000.

[25] D. Karaboga. An idea based on honey bee swarm for numerical optimization. *Techn. Rep. TR06, Erciyes Univ. Press, Erciyes*, 2005.

[26] D. Karaboga and B. Basturk. On the performance of artificial bee colony (abc) algorithm. *Applied Soft Computing*, 8(1):687–697, 2008.

[27] Dervis Karaboga and Bahriye Akay. A modified artificial bee colony (abc) algorithm for constrained optimization problems. *Applied Soft Computing*, 11(3):3021–3031, 2011.

[28] Majid Kazemian, Yoosef Ramezani, Caro Lucas, and Behzad Moshiri. Swarm clustering based on flowers pollination by artificial bees. In *Swarm Intelligence in Data Mining*, pages 191–202. Springer, 2006.

[29] Mustafa Servet Kıran and Mesut Gündüz. A recombination-based hybridization of particle swarm optimization and artificial bee colony algorithm for continuous optimization problems. *Applied Soft Computing*, 13(4):2188–2203, 2013.

[30] Laurits W Krefting and Eugene I Roe. The role of some birds and mammals in seed germination. *Ecological Monographs*, pages 269–286, 1949.

[31] John A Lawrence and Barry A Pasternack. *Applied management science*. Wiley New York, 2002.

[32] Simon A Levin. Population dynamic models in heterogeneous environments. *Annual Review of Ecology and Systematics*, 7(1):287–310, 1976.

[33] Simon A Levin, Dan Cohen, and Alan Hastings. Dispersal strategies in patchy environments. *Theoretical population biology*, 26(2):165–191, 1984.

[34] JJ Liang, Thomas Philip Runarsson, Efren Mezura-Montes, Maurice Clerc, PN Suganthan, CA Coello Coello, and K Deb. Problem definitions and evaluation criteria for the cec 2006 special session on constrained real-parameter optimization. *Journal of Applied Mechanics*, 41, 2006.

[35] lifeisfull@ineedexchange.com. looking for nice strawberries to pick among the tiny ones, rotten ones and thistles! `http://lifeisfull.wordpress.com/page/5/`. Accessed April 21, 2014.

[36] John DC Little. A proof for the queuing formula: L= λ w. *Operations research*, 9(3):383–387, 1961.

[37] @monacoeye. Blue winged mountain-tanager eating tree strawberries at san isidro lodge. http://monacoeye.com/birds/index_files/anisognathus_somptuosus_blue_winged_mountain_tanager.php/. Accessed April 21, 2014.

[38] Andy M Reynolds and Mark A Frye. Free-flight odor tracking in drosophila is consistent with an optimal intermittent scale-free search. *PloS one*, 2(4):e354, 2007.

[39] Ruth. Strawberry eaten by birds. http://ghostmyst.blogspot.co.uk/2012/11/something-is-eating-my-strawberries.html/. Accessed April 21, 2014.

[40] A. Salhi and E.S. Fraga. Nature-inspired optimisation approaches and the new plant propagation algorithm. *Proceedings of the ICeMATH2011*, pages K2–1 to K2–8, 2011.

[41] Nigel E Stork and Christopher HC Lyal. Extinction or'co-extinction'rates? *Nature*, 366:307, 1993.

[42] P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, YP Chen, A. Auger, and S. Tiwari. Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization. *Nanyang Technological University, Singapore, Tech. Rep*, 2005005, 2005.

[43] Muhammad Sulaiman, Abdellah Salhi, Birsen Irem Selamoglu, and Omar Bahaaldin Kirikchi. A plant propagation algorithm for constrained engineering optimisation problems. *Mathematical Problems in Engineering*, Volume 2014, Article ID 627416, 10 pages, (2014).

[44] José Luis Tellería, Álvaro Ramírez, and Javier Pérez-Tris. Conservation of seed-dispersing migrant birds in mediterranean habitats: shedding light on patterns to preserve processes. *Biological Conservation*, 124(4):493–502, 2005.

[45] Darcy Wentworth Thompson et al. On growth and form. *On growth and form.*, 1942.

[46] Kyle S Van Houtan, Stuart L Pimm, John M Halley, Richard O Bierregaard, and Thomas E Lovejoy. Dispersal of amazonian birds in continuous and fragmented forest. *Ecology Letters*, 10(3):219–229, 2007.

[47] Daniel G Wenny and Douglas J Levey. Directed seed dispersal by bellbirds in a tropical cloud forest. *Proceedings of the National Academy of Sciences*, 95(11):6204–6207, 1998.

[48] Wikipedia. Strawberry. http://en.wikipedia.org/wiki/Strawberry/. Accessed April 21, 2014.

[49] X-S Yang. Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-Inspired Computation*, 2(2):78–84, 2010.

[50] X-S Yang. *Nature-inspired metaheuristic algorithms*. Luniver Press, 2011.

[51] X-S Yang and S Deb. Engineering optimisation by cuckoo search. *International Journal of Mathematical Modelling and Numerical Optimisation*, 1(4):330–343, 2010.

[52] Xin-She Yang. Flower pollination algorithm for global optimization. In *Unconventional Computation and Natural Computation*, pages 240–249. Springer, 2012.

[53] Xin-She Yang, Mehmet Karamanoglu, and Xingshi He. Multi-objective flower algorithm for optimization. *Procedia Computer Science*, 18:861–868, 2013.