

# Enabling Field Force Operational Sustainability: A Big Bang-Big Crunch Type-2 Fuzzy Logic System for Goal-Driven Simulation

Emmanuel Ferreyra, Hani Hagrais

The Computational Intelligence Centre,  
School of Computer Science and Electronic Engineering,  
University of Essex, Colchester, UK.  
{eferre, hani}@essex.ac.uk

Mathias Kern, Gilbert Owusu

Business Modelling and Operational Transformation  
Practice, Research and Innovation, BT,  
Austral Park, Martlesham Heath, Ipswich, UK.  
{mathias.kern, gilbert.owusu}@bt.com

**Abstract**—Business operational sustainability must allow creating economic value, building healthy ecosystems and developing strong communities. Hence, there is a need to develop solutions which can safeguard companies' business sustainability. Various solutions could have different costs and deliver different benefits. Therefore, there is a need to evaluate these solutions before being implemented. In reality, companies require achieving certain targets according to their plans and strategies. Goal-Driven Simulation (GDS) is an approach that allows evaluating solutions before implementing them in real-life while focusing on achieving desired targets. This paper presents a GDS based on interval type-2 Fuzzy Logic System (IT2FLS) optimized by the big bang-big crunch (BB-BC) algorithm with application to field force allocation within the telecommunications sector. The obtained results show the suitability of the proposed approach to model unexpected factors to protect the business sustainability in the telecommunications industry field force allocation domain.

**Keywords**— *interval type-2 fuzzy logic systems; goal-driven simulation; big bang-big crunch; operational sustainability;*

## I. INTRODUCTION

There is a strong need in various industries to develop intelligent tools to support decision-making with direct impact in many areas of their business activity. Designing effective intelligent solutions in the current digital economy is not a trivial task as this is influenced by complex operational conditions. For example, unexpected behaviour of various external factors such as surges in demand or difficulties in provisioning among others. Hence, companies are facing new problems that could affect their market presence and profitability. Therefore, intelligent solutions aim to safeguard companies' business sustainability by providing optimal solutions according to the operating conditions. Commonly, these solutions are supported by mathematical and/or theoretical models in the form of computer algorithms. Consequently, goal driven simulation (GDS) emerged to mimic real-world processes or systems over a specific period of time [1]. GDS can be employed for multiple aims, such as forecasting, understanding systems' behaviour and for heuristic purposes [2] among others.

Essentially, GDS attempts to find the value of a specific set of inputs to achieve a desired output by employing a simulation

model [3]. Formerly, many problems were modelled with closed-form solutions. However, most of the current real-world problems present various variables and multiple components. In addition, many problems incorporate uncertainty and non-linearity is observed in such situations. Traditional simulation allows to create and evaluate "What-if? Scenarios" [4]. On the other hand, GDS allows to perform "How-to? Scenarios". Consequently, GDS attempts to determinate which inputs required to be modified to accomplishment the desired goal.

Solutions recommended by GDS approaches incorporate various levels of complexity. These aim to protect business sustainability. In other words, these solutions take into consideration companies' financial situation, social and environmental risk and regulatory compliance. Therefore, these solutions are not purely theoretical, but also highly practical. Due this importance, GDS is an active line of research. Current real-world processes that drive economic activities demand better tools to achieve companies' desired aims. The ability to handle the uncertainty associated with unstructured environments is a key element for succeeding on this. Handling uncertainty in scheduling related problems has been solved based on mathematical models, by employing probability theory [5], [6], numerical analysis [7], and interval mathematics [8], [9]. However, most real-world problems are too complex to be modelled mathematically. Notably, type-2 FLSs does not require mathematical formulations for handling linguistic and numerical uncertainties [10]. This paper proposes a GDS based on an interval type-2 Fuzzy Logic System (IT2FLS) optimized by the big bang-big crunch (BB-BC) algorithm which is characterised by low computational cost, ease of implementation, and fast convergence. The presented GDS aims to enhance the operational sustainability in the service provider industry in the field scheduling domain specifically within telecommunication service providers (TSPs).

The remainder of this paper is organised as follows. Section II presents an overview of the goal-driven simulation problem. Section III provides a brief overview of type-2 FLSs. Section IV outlines the big bang-big crunch optimisation algorithm. Section V depicts the proposed system. Section VI reports the experiments and their results. Finally, section VII draws the conclusions and future work.

## II. OVERVIEW OF THE GOAL-DRIVEN SIMULATION PROBLEM

In supplier companies, field scheduling refers to the allocation of a set of tasks to complete on a set of available workforce with different skills [11]. Successful operation relies on an efficient resource (field force) allocation [12]. Therefore, the available workforce fulfils planned positions [13]. This relation is commonly studied under the umbrella of operational workforce planning [14].

The workforce in Telecommunication Service Providers (TSPs) might be composed of qualified engineers. Let us assume that the tasks to be completed take place in the customer's premises. Consequently, the management of demand and supply factors can be achieved by building geographical hierarchies. Without loss of generality, let us adopt that tasks are logged into a small unit known as service distribution point (SDP). The grouping of SDPs is referred as working area (WA). Sub-regions are the result of grouped WAs. Finally, a region encompasses various sub-areas [15]. The goal is to ensure the achievement of operational business plans by the effective allocation of the available workforce [14].

Field scheduling incorporates numerous factors, such as travel distances, tasks' priorities, preferred skills, work-types, etc. These elements make scheduling processes expensive. Notably, computer simulation has helped on this matter. Firstly, by proving its importance for understanding the dynamics in intelligent transportation systems (ITS) [16]. Next, by modelling logistic systems and scheduling algorithms in multi-agent simulation systems within the transportation engineering domain [17]. Recently, the significance of simulation in data assimilation was studied [18] and the relevance of simulation performance was evaluated in real-word conditions and compared to design modalities [19]. Most recently, GDS approaches have succeeded in achieving desired performance targets [20] and in finding preferred service composition on demand [21] within pervasive computing. However, these approaches reduced the problem of study to a linear model and to a heuristic distributed backward-chaining model, respectively. As result, these solutions are adjacent to the "What-if?" form. This paper studies achieving certain targets by finding the appropriate input configuration when employing simulation in field scheduling processes. Therefore, we perform "How-to? Scenarios" instead of traditional "What-if? Scenarios".

Implementing "How-to? Scenarios" is complex, mainly because the need of determining automatically the value of inputs by executing the simulation model backwards from the desired goal. If successful, these solutions reduce manual adjustment of inputs by trial and error until the goal is achieved. However, there are some risks such as large computation periods due to a potential large set of testing and the possibility of not finding the adequate configuration. Evidently, these situations attempt against operational sustainability. The proposed approach detailed in section V attempts to solve this issue.

## III. OVERVIEW OF TYPE-2 FUZZY LOGIC SYSTEMS

Zadeh [22] formalised fuzzy set theory, which, in turn, resulted in the foundation of fuzzy logic systems (FLSs). FLSs aim to provide human descriptive understanding to problem

solving. Without loss of generality, a fuzzy set is a collection of elements that allow its members different grades of belonging among the range  $[0, 1]$ . Thus, the measure of the degree of similarity of an element in such fuzzy set is given by a membership function [22].

A type-1 fuzzy set can be designed by defining precise and crisp memberships values in  $[0, 1]$  where a type-2 fuzzy set can be designed by defining fuzzy membership functions for each element in the  $x$ -domain. Besides differences in their construction, type-2 FLSs can handle higher levels of uncertainty compared to type-1 FLSs. This is achieved by incorporating different degrees of freedom via the foot-print of uncertainty (FOU) combined with the 3-dimensional nature of type-2 fuzzy sets. Therefore, in type-2 FLS the third dimension allows a secondary membership function that is associated with a degree of membership. When this secondary membership function takes a maximum uncertainty of 1 in a specific interval  $[a, b]$  then an interval type-2 fuzzy set is created as shown in Fig. 1. As can be observed, the key elements of the interval type-2 fuzzy sets are: the foot print of uncertainty (FOU), the upper membership function UMF and the lower membership function (LMF). Observe that the maximum uncertainty expressed in the secondary membership function is equal to 1, therefore an interval-type 2 fuzzy set can be simplified as depicted in Fig. 2.

Fuzzy sets are associated with linguistic terms which form parts of fuzzy rules which are conditional statements in the form: IF *some antecedent* THEN *some consequent*. [22].

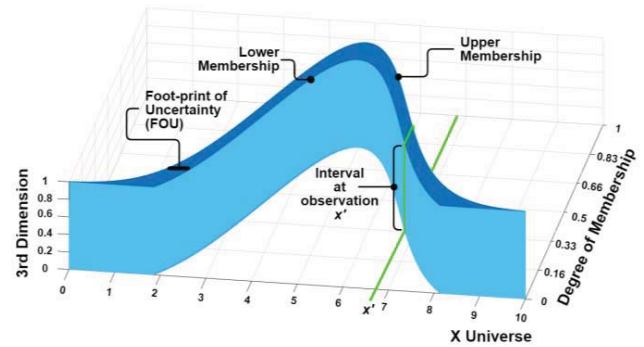


Fig. 1. 3-dimensional view of an interval type-2 fuzzy set for a given input.

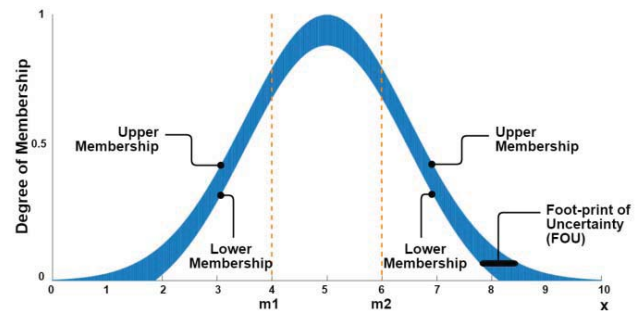


Fig. 2. Simplified view of a Gaussian interval type-2 fuzzy set with uncertain mean.

In interval type-2 FLSs, first, crisp inputs are fuzzified into type-2 input fuzzy sets. Next, type-2 input fuzzy sets actuate the inference engine in conjunction with the rule base. Note that rules will remain as type-1 FLS, however, the membership functions representing the antecedents and/or consequents will be represented as interval type-2 fuzzy sets. Subsequently, the inference engine will provide the corresponding mapping to type-2 output fuzzy sets. The type reducer will process the output type-2 fuzzy set and type-reduce it to a type-1 fuzzy set for defuzzification (see Defuzzifier) providing the result in crisp outputs [23], [24].

#### IV. OVERVIEW OF THE BIG BANG-BIG CRUNCH ALGORITHM

In [25], a method called Big Bang-Big Crunch (BB-BC) based on the evolution of the universe was introduced. Without loss of generality, BB-BC is conformed of two phases: the Big Bang phase and the Big Crunch phase. The former generates random points, the latter compacts those points to a single representative point. The steps for this optimisation algorithm can be enumerated as follows:

*Step 1:* Similarly to other search algorithms based on evolution, the initial Big Bang population is randomly generated by considering the whole research space.

*Step 2:* The successive Big Bang phases are randomly distributed by the centre of mass that corresponds to the best fit individual in a similar approach. Here, the cost function values of all candidates are computed.

*Step 3:* Once completed the Big Bang phase, the Big Crunch takes place. Here, a contraction procedure is carried out to conform a centre of mass, in the form of:

$$\bar{x}^c = \frac{\sum_{i=1}^N \frac{1}{f^i} \bar{x}^i}{\sum_{i=1}^N \frac{1}{f^i}} \quad (1)$$

Where, the position of the centre of mass is denoted by  $\bar{x}^c$ , the position of the candidate is given by  $\bar{x}^i$ , the cost function value of the  $i$ th candidate is expressed by  $f^i$  and  $N$  is the population size.

*Step 4:* Subsequently, new candidates are calculated around the centre of mass. For this, as the iterations elapse the value of a normal distributed random number is added or subtracted, in the form of:

$$\bar{x}^{new} = \bar{x}^c + \frac{Y\rho(x_{max} - x_{min})}{k} \quad (2)$$

Where,  $\bar{x}^c$  is the centre of mass,  $Y$  represents the normal random number,  $\rho$  denotes the parameter limiting the search space,  $x_{max}$  and  $x_{min}$  are upper and lower limits respectively, and  $k$  is the iteration step.

*Step 5:* If the stopping criteria have not been met, then iterate from step 2. Common stopping criteria methods include: reaching the maximum number of iterations, an error values that results lower than a given threshold or elapsed optimisation time [26].

BB-BC has been implemented successfully in data clustering [27], in type-2 fuzzy PID cascade controller strategies [28], in type-2 fuzzy logic systems for machine vision [29] and in type-2 fuzzy logic based human behaviour recognition systems [30] among others. We used the BB-BC algorithm to optimise the rule base of the proposed IT2FLS.

#### V. THE PROPOSED INTERVAL TYPE-2 FLS FOR GOAL-DRIVEN SIMULATION

An overview on the proposed interval type-2 FLS is depicted in Fig. 3. The proposal is comprised of five phases. First, raw data is generated. Next, baseline scenarios are simulated. Then, knowledge is acquired and implemented as the core for the goal-driven simulation approach. Subsequently, a testing phase is carried employing unseen data. Finally, a benchmark evaluation is carried out by implementing a shallow neural network. Details on the elements implemented on these phases are provided in the following subsections.

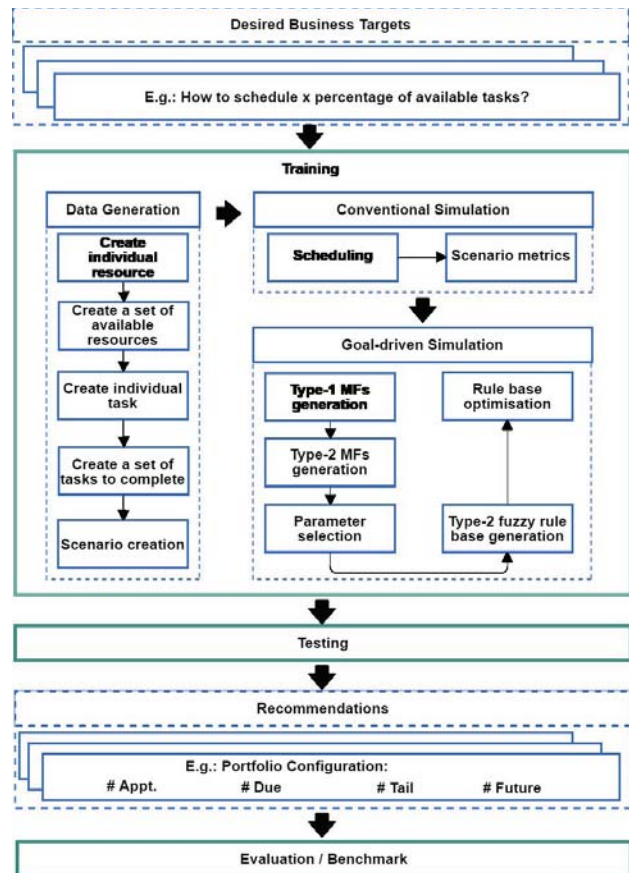


Fig. 3. A high level overview of the proposed IT2FLS.

##### A. Data Generation

Initial data generation is performed to obtain a series of simulation scenarios as baseline for future goal-driven computation. In order to achieve this, a certain domain is selected within the UK territory according to the hierarchy briefly described in section II. The minimum required data includes detailed information about availability of skilled



engineers, the expected tasks to be completed and a baseline metric that encompasses the relationship between supply and demand. Prior to this data generation, statistical distribution of the characteristics corresponding to the selected domain were analysed. Subsequently, routines with the logic for generation were created. In order to reduce complexity, the data manipulation for generating an individual resource is reported in Table I. Similarly, Table II describes the steps for creating a set of available resources for a given scenario.

TABLE I. CREATE AN INDIVIDUAL RESOURCE

Step	Description
1	Create random resource identifier
2	Randomly select resource location and preferred working area
3	Randomly select resource worktypes
4	Randomly select resource availability
5	Create entry in resource master table
6	Create entry in resource preferred working area table
7	Create entry in resource area table
8	Create entries in resource skills table
9	Create entry in resource availability table

TABLE II. CREATE A SET OF AVAILABLE RESOURCES

Step	Description
1	Determine random number of desired resources
2	Log information about number of resources
3	Create individual resources according to number generated in step 1

For generating the expected tasks to be completed, Tables III and IV were implemented. The former describes individual task creation and the latter highlights the creation of a set of tasks. As a result, Fig. 4 illustrates an example of task distribution for the selected domain.

TABLE III. CREATE AN INDIVIDUAL TASK

Step	Description
1	Randomly select task location
2	Randomly select skill and task category
3	Randomly select importance and duration
4	Create entry in master task and task skills tables

TABLE IV. CREATE A SET OF TASKS TO COMPLETE

Step	Description
1	Determine random number of desired tasks
2	Log information about number of tasks
3	Create individual tasks according to number generated in step 1

The relationship between supply and demand denotes an initial scenario. As result, this data generation process delivered a portfolio of 40,795 tasks. 56% (22,772) corresponding to repair work and 44% (18,023) to provision work. The total estimated duration for this sample is 323,023 hours. Fig. 5 shows a distribution of the work types for the generated tasks.

addition, the number of available engineers range between 4 and 139 distributed randomly among 131 scenarios.

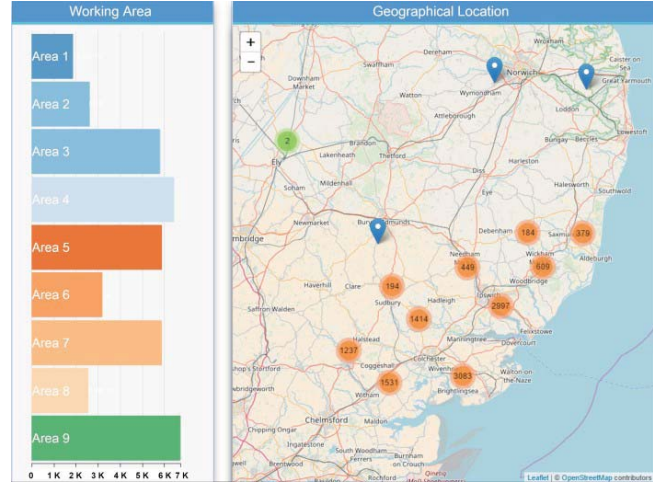


Fig. 4. Example of task distribution by working area (left) and geographical location (right).

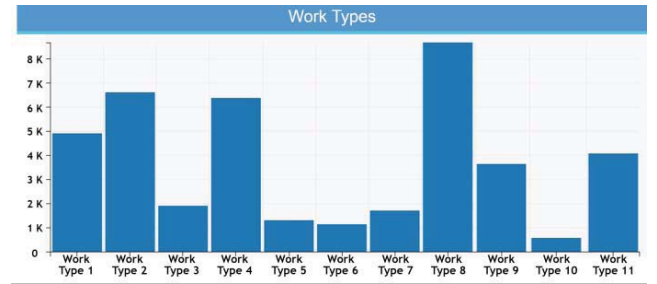


Fig. 5. Work types distribution for the generated sample, where the y-axis measures number of tasks and x-axis the corresponding work types.

### B. Conventional Simulation

The data generated in the previous section is fed into the scheduling algorithm. Regardless of the scheduling algorithm employed, as most algorithms of this nature optimise their results by prioritising tasks and resources according to cost function settings and many other internal parameters. This proposal manages the conventional scheduling algorithm as a black-box element. In other words, no exhaustive internal detail is required to obtain initial simulation results.

This simulation step computes the allocation of the available tasks among the available resources in the form of simulated scenarios. A simulated scenario has certain metrics that reflect the characteristics of the solution. Table V introduces some of the measures for a given simulation result. Essentially, some performance indicators can be concluded from these metrics. For example, computing the scheduling rate  $SR$  for a given scenario  $i$  can be obtained by the following relation:

$$SR_i = \frac{pA_i^s + pNA_i^s + rA_i^s + rNA_i^s}{aT_i} \times 100\% \quad (3)$$

Where,  $pA_i^s$  and  $pNA_i^s$  denote the number of successful scheduled *provision* appointments and non-appointments for a given scenario  $i$ , respectively. Similarly,  $rA_i^s$  and  $rNA_i^s$  represent the number of successful scheduled *repair* appointments and non-appointments for a given scenario  $i$ , respectively. Finally,  $aT_i$  represents the total of available task to be scheduled for a given scenario  $i$ .

TABLE V. SCENARIO METRICS (REDUCED) FOR A GIVEN SCENARIO ( $i$ )

<i>Relevant Measures for Scheduled Scenarios</i>			
Measure	Symbology	Measure	Symbology
Available resources	$R_i$	Available working time	$WT_i$
Available tasks	$J_i$	Provision appointment success	$pA_i^s$
Resources with job	$RJ_i$	Provision non-appointment success	$pNA_i^s$
Travel time spent	$T_i$	Repair appointment success	$rA_i^s$
Task time spent	$TJ_i$	Repair non-appointment success	$rNA_i^s$
Locations visited	$L_i$	Scheduled tasks	$S_i$

### C. Goal-driven simulation

The data generated in the previous sections is used for the implementation of the learning phase for the goal-driven simulation approach. First, we apply a clustering technique to each feature. Second, we transform the outcome of the previous step to the means for implementing an interval type-2 fuzzy logic sets with a FLS (as introduced in section III). Next, a parameter selector is implemented for associating the relevant features to the desired goal. Next, we construct the rule base. Subsequently, this is optimised by the big bang-big crunch algorithm. Finally, a training phase is carried out for knowledge acquisition. As a result, the proposed GDS approach is modelled. The following sections provide detailed description on these steps.

1) *Type-1 membership functions generation*: Based on clustering classification, the type-1 membership functions (T1MFs) are obtained. The aim is to associate each data observation to all possible valid clusters with a membership value. For this, we employed the fuzzy c-means algorithm (FCM). The FCM algorithm was first proposed by Dunn [31] and later improved by Bezdek [32]. The FCM is an extension of the K-means algorithm [33]. Without loss of generality, the FCM encompasses the following steps:

*Step 1*: Random initialisation of the cluster membership value.

*Step 2*: Compute the cluster centroid  $C_j$  of the  $j^{th}$  cluster as follows:

$$C_j = \frac{\sum_{i=1}^N \mu_{ij}^m x_i}{\sum_{i=1}^N \mu_{ij}^m} \quad (4)$$

Where,  $N$  represent the sample size,  $\mu_{ij}$  denotes the degree of membership of  $x_i$  in the  $j^{th}$  cluster. And the exponent that controls the degree of fuzzy overlap is given by  $m$ .

*Step 3*: For each cluster  $k$  in the total defined clusters  $K$ , update the degree of membership  $\mu_{ij}$  as follows:

$$\mu_{ij} = \frac{1}{\sum_{h=1}^K \left( \frac{|x_i - c_j|}{|x_i - c_h|} \right)^{\frac{2}{m-1}}} \quad (5)$$

*Step 4*: Compute the objective function as follows:

$$O_m = \sum_{i=1}^N \sum_{j=1}^K \mu_{ij} |x_i - c_j|^2 \quad (6)$$

*Step 5*: If the objective function has reached certain threshold value or maximum iteration parameter, then it stops. More detail of fuzzy set based clustering can be found in [34] and [35].

The output of the proposed approach implements the case study of the following goal: *How to schedule certain percentage of available tasks?* Consequently, the system yields a high level portfolio composition including: i) the number of expected appointed work (appt.); ii) the estimated non-appointed work due on scheduling date (due); iii) the quantity of expected non-appointed work due before scheduling date (tail) and iv) the amount of non-appointed work due after scheduling date (future). Each of this consequents are characterised by fuzzy sets, these are  $\{low, medium, high\}$ . As demonstrated in [38], generally Gaussian membership functions have reduced complexity because only the mean  $m$  and the standard deviation  $\sigma$  are require for their construction. The FCM algorithm computes these parameters. Therefore, the Gaussian T1MFs can be obtained as follows:

$$\mu_{X(x)} = e^{-\frac{(x-m)^2}{2\sigma^2}} \quad (7)$$

2) *Type-2 membership functions generation*: First, an uncertainty factor ( $\rho$ ) is defined to generate the Footprint of Uncertainty (FOU). This is employed to generate a bounded region that encompasses the obtained T1MFs. As a result, upper  $\bar{\mu}_{\bar{A}}(x)$  and lower membership  $\underline{\mu}_{\bar{A}}(x)$  functions are identified. T1MFs are also referred as primary membership functions. Commonly, Gaussian interval type-2 membership functions are obtained by blurring the mean or standard deviation of the obtained primary membership [38]. We apply the uncertainty factor ( $\rho$ ) to the standard deviation. Consequently, the IT2MFs have certain mean  $m$  and uncertain standard deviation  $\sigma [\sigma_1, \sigma_2]$ . Where  $\sigma_1 = \sigma - \rho$  and  $\sigma_2 = \sigma + \rho$ , respectively. Coordinates of the upper  $\bar{\mu}_{\bar{A}}(x)$  and lower  $\underline{\mu}_{\bar{A}}(x)$  membership functions can be found at:

$$\bar{\mu}_{\bar{A}}(x) = e^{-\frac{(x-m)^2}{2\sigma_2^2}} ; \underline{\mu}_{\bar{A}}(x) = e^{-\frac{(x-m)^2}{2\sigma_1^2}} \quad (8)$$

Fig. 6 reports an example for the MFs obtained for the number of expected appointed work (appt. output). Therefore, the uncertainty factor  $\rho$  is adjusted in the range [1, 100] in the relation to the standard deviation  $\sigma$  obtained by the FCM algorithm for each TIMF  $k$  such that:

$$\rho_k = \frac{\sigma_k}{100} \quad (9)$$

3) *Parameter selection*: Identification of relevant parameters for the given goal-driven problem is carried out. This aims to communicate and interact with business experts and simultaneously, it helps to instruct the system which features are relevant given the case study to be implemented. Naturally, different goals could be achieved by different parameters.

4) *Type-2 fuzzy rule base generation*: This stage implements the rule extraction method for multiple input – multiple output according to the parameter selection carried out in the previous step. This will result in a set of  $R$  rules. These rules are combined and mapped from input type-2 fuzzy sets to output type-2 fuzzy sets and take the following form [36]:

$$R_l: \text{If } x_1 \text{ is } \tilde{A}_1^l \text{ and } x_n \text{ is } \tilde{A}_n^l \text{ Then } y_1 \text{ is } \tilde{B}_1^l \text{ and } y_n \text{ is } \tilde{B}_n^l \quad (9)$$

Where,  $l = \{1, 2, \dots, R\}$ .  $R$  is the total number of rules and  $l$  is the rule index. The input being evaluated is denoted by  $x_n$  and  $n = \{1, 2, \dots, N\}$  is the total of  $N$  inputs. The IT2 fuzzy antecedent is described by  $\tilde{A}_n^l$  and  $n = \{1, 2, \dots, M\}$  is the total of  $M$  antecedents. The IT2 fuzzy consequents are given by  $\tilde{B}_n^l$  and  $n = \{1, 2, \dots, K\}$  is the total of  $K$  consequents. Details of this rule extraction method, known as Wang-Mendel approach are included in [22] and [37].

5) *Rule base optimisation*: The BB-BC algorithm is employed to optimise the rule base. This optimisation consists of reduction of number of rules and minimisation in the number of antecedents for the resulting rule base. In order to apply the steps described in section IV, both, the rule base and the available type-2 MFs are required to be encoded into a population form. The population encoding for the rule base results as follows:

$$\boxed{\tilde{A}_1^1} \dots \boxed{\tilde{A}_n^1} \boxed{\tilde{B}_1^1} \dots \boxed{\tilde{B}_n^1} \dots \boxed{\tilde{A}_1^l} \dots \boxed{\tilde{A}_n^l} \boxed{\tilde{B}_1^l} \dots \boxed{\tilde{B}_n^l}$$

The population encoding for the type-2 MFs results as follows:

$$\boxed{\alpha_l^1} \boxed{\alpha_m^1} \boxed{\alpha_h^1} \dots \boxed{\alpha_l^j} \boxed{\alpha_m^j} \boxed{\alpha_h^j} \boxed{\varpi_l^1} \boxed{\varpi_m^1} \boxed{\varpi_h^1} \dots \boxed{\varpi_l^k} \boxed{\varpi_m^k} \boxed{\varpi_h^k}$$

Where, the fuzzy sets  $\{low, medium, high\}$  are represented by the sub-indexes  $l, m$  and  $h$ , respectively. The total number of inputs is given by  $j = \{1, 2, \dots, N\}$ . The total number of outputs is given by  $k = \{1, 2, \dots, M\}$ . For each input, the parameter  $\alpha$  represents the uncertainty factor as indicated in Equation (7). Similarly, the  $\varpi$  parameter represents the uncertainty factor for each output.

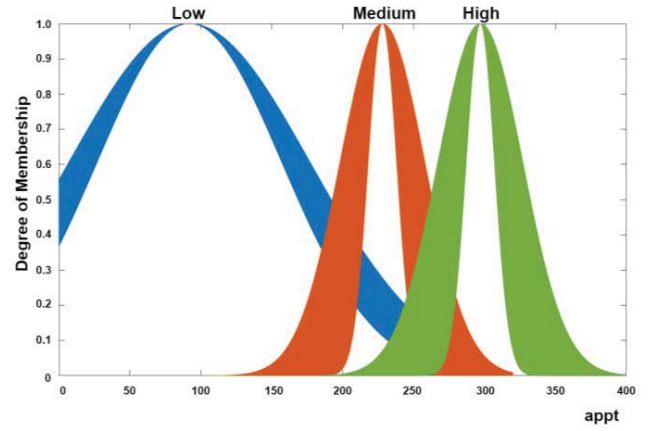


Fig. 6. Example of type-2 membership functions for the appt output by employing the FCM algorithm.

#### D. Testing Phase

After training the IT2FLS based on the domain and data described in previous sections, evaluation of unseen data was carried out. This focused on the desired goal: *How to schedule certain percentage of available tasks?* In order to reduce complexity, let us assume the following example to illustrate the inputs for the implementation of this phase: number of available resources; number of available tasks; number of resources with job; number of scheduled tasks. Similarly, let us hold the following assumptions to represent the corresponding outputs:

- Appointed work is integrated as follows: number of provision appt. success; number of provision appt. failure; number of repair appt. success; and number of repair appt. failure.
- Non-appointed work is integrated as follows: number of provision non-appt. success; number of provision non-appt. failure; number of repair non-appt. success; and number of repair non-appt. failure.

Based on these assumptions it can be illustrated that total number of parameters for the describe scenario is given by the total inputs, i.e. 4 multiplied by the proposed MFs, i.e. 3 for each parameter (namely,  $\{low, medium, high\}$ ), equal to 12. Similarly, parameters for the outputs yield 24 (i.e.  $8 \times 3$ ). Note that increasing the number of inputs and/or outputs can be easily scaled by the parameter selection procedure.

#### E. Benchmark

In order to situate the proposed system within a reference frame, a shallow neural network (feed-forward backpropagation) was implemented by employing the neural network MATLAB built-in tool. The idea is to monitor sensible outputs and analyse any significant deviation. Therefore, the corresponding network implemented is a 4-10-10-8 network with logsig neurons in all layers. Since the number of expected weights is considered low, the Levenberg-Marquardt algorithm was employed for training. The network is trained either, until the mean squared error is below a threshold value equal to 0.010 or the number of iterations is above a threshold vale equal to 2,000.

## VI. EXPERIMENTS AND RESULTS

For each obtained output an accuracy is measured according to the mean square error (MSE). The MSE is computed as follows:

$$MSE_{output} = \frac{1}{n} \sum_{i=0}^n (Y - Y')^2 \quad (10)$$

Where,  $Y'$  is the estimated value and the baseline value is given by  $Y$ . This for a given set of experiments  $i = \{1, 2, \dots, n\}$ . Table VI reports the average MSE obtained during the training and testing phases. The expected outputs can be grouped in two categories as follows:

$$Appt. = \sum_{i=0}^N pA_i^s + pA_i^f + rA_i^s + rA_i^f \quad (11)$$

Where, the total of appointed work ( $Appt.$ ) is determined by the total of provision appointment successful work ( $pA_i^s$ ), and the total of provision appointment failure work ( $pA_i^f$ ), and the total of repair appointment successful work ( $rA_i^s$ ), and the total of repair appointment failure work ( $rA_i^f$ ) for a given scenario  $i = \{1, 2, \dots, N\}$ . Similarly, the total of non-appointed work ( $Non-Appt.$ ) can be computed as follows:

$$Non - Appt. = \sum_{i=0}^N pNA_i^s + pNA_i^f + rNA_i^s + rNA_i^f \quad (12)$$

Where, total of provision non-appointment success is denoted by  $pNA_i^s$ . The total of provision non-appointment failure is represented by  $pNA_i^f$ . The total of repair non-appointment success is given by  $rNA_i^s$  and the total of repair non-appointment failure is denoted by  $rNA_i^f$  for a given scenario  $i = \{1, 2, \dots, N\}$ .

When focusing on the  $Appt.$  category, the proposed type-2 FLS outperformed its type-1 counterpart by 23% and 19% on training and testing, respectively. For the  $Non-Appt.$  output, the type-2 FLS kept this tendency with 8% and 4% on training and testing data, respectively. The shallow artificial neural network (ANN), which was employed as benchmark, outperformed the type-1 and type-2 FLSs in both categories (i.e.  $Appt.$  and  $Non-Appt.$ ) on testing and training. However, because of the black-box nature of ANNs the degree of interpretability is a known issue. Consequently, the effectiveness among the compared methods must achieve simultaneously two important factors, these are sensible performance and concise interpretability.

TABLE VI. PERFORMANCE COMPARISON BETWEEN TYPE-1 FLS, TYPE-2 FLS AND A SHALLOW ARTIFICIAL NEURAL NETWORK

System Type	Training MSE		Testing MSE	
	Appt.	Non-Appt.	Appt.	Non-Appt.
Type-1 FLS	0.085	0.093	0.066	0.075
Type-2 FLS	0.065	0.085	0.053	0.072
Shallow ANN	0.010	0.010	0.015	0.014

Furthermore, the absolute average in the MSE difference between testing and training phases is further illustrated by the  $\Delta$ MSE description presented in Table VII. Where, the smallest

variation yields better performance. As can be observed, the type-2 FLS outperformed its type-1 counterpart. In addition, although the ANN presented the smallest variation, there is the issue that its learnt weights only work for stable datasets. In the other hand, the type-2 FLS is capable of adding uncertain factors as described in Section V. This reflects the type-2 FLS's ability to better handle the unseen uncertainties to be encountered in real-world simulation environments.

TABLE VII.  $\Delta$ MSE VARIATION BETWEEN TESTING AND TRAINING

System Type	$\Delta$ MSE		$\Delta$ MSE
	$\Delta$ Appt.	$\Delta$ Non-Appt.	Average $\Delta$
Type-1 FLS	0.019	0.018	0.0185
Type-2 FLS	0.012	0.013	0.0125
Shallow ANN	0.005	0.004	0.0045

## VII. CONCLUSIONS AND FUTURE WORK

This paper has proposed an interval type-2 fuzzy logic system optimised by the BB-BC optimisation algorithm for a goal-driven simulation approach in order to enable operational sustainability in the telecommunications service provider industry. The main research contributions of the described approach are twofold. Firstly, it demonstrates the modelling and scalability of real-life problems while obtaining reasonable results when comparing with traditional universal approximators, such as shallow artificial neural networks. Secondly, it provides concise interpretability while embedding uncertainty into the model. The proposed system allowed evaluation of the conditions for achieving certain business targets before being implemented in real-life conditions, which, in turn enables the field force operational sustainability nature of this research problem. Meticulously, this paper walked the reader through an incremental framework for implementing "How-to? Scenarios" by showcasing a particular real-life case of study. Results suggested sensible response to uncertainties when applying this approach. Additionally, it has been observed superior accuracy when employing a shallow artificial neural network. However, in order to provide transparency in real-life business operations two relevant issues are encountered with the shallow ANN approach. The former, refers to the impossibility to guarantee that the learnt weights can be applied to scenarios with complex relation between inputs and outputs and still yield an acceptable performance. The latter, discusses the limitations of ANNs to provide a concise level of interpretability. Therefore, the proposed approach demonstrated its suitability to better handle uncertainties and provide sensible results that safeguard businesses operational sustainability for goal-driven scenarios, by building a healthy simulation environment that builds interpretability while moving from conventional simulation to a target-oriented model.

Further lines of research include, the study of complex portfolio composition. These embrace, detailed distribution of work type and its required skill-set, inclusion of resource attending patters and the expectation of quality of service achieved by scenario. Finally, we will replicate the benchmark



exercise by employing deep learning techniques as part of our future work.

## REFERENCES

- [1] B. L. Nelson, J. S. Carson and J. Banks, *Discrete event system simulation*, 1<sup>st</sup> ed. Prentice Hall, 1995, pp. 3.
- [2] J. A. Sokolowski and C. M. Banks, *Principles of modeling and simulation*, 1<sup>st</sup> ed. Wiley, 2009, pp. 3-24.
- [3] M. Page, J. Gensel and M. Boudis, "An algorithm for goal-driven simulation," in *Proceedings of the 31<sup>st</sup> Conference on Winter Simulation: Simulatin - a Bridge to the Future*, Vol. 1, WSC'99, Phoenix, Arizona, USA: ACM, 1999, pp. 578-585.
- [4] L. A. Molina, C. Gandarillas and M. A. Centeno, "Goal driven simulation intelligent back ends: a state of the art review," in *Proceedings Winter Simulation Conference*, 1996, pp. 734-739.
- [5] W. Duan, Z. Li, M. Ji, Y. Yang, S. Wang and B. Liu, "A hybrid estimation of distribution algorithm for distributed permutation flowshop scheduling with flowline eligibility," in *IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, BC, 2016, pp. 2581-2587.
- [6] C. Wu, L. Wang and X. I. Zheng, "An effective estimation of distribution algorithm for solving uniform parallel machine scheduling problem with precedence constraints," in *IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, BC, 2016, pp. 2626-2632.
- [7] A. Mavrommati, J. Schultz and T. D. Murphey, "Real-time dynamic-mode scheduling using single-integration hybrid optimization," in *IEEE Transactions on Automation Science and Engineering*, Volume 13, Issue 3, July 2016, pp. 1385-1398.
- [8] Y. Fan, G. H. Huang, and Y. P. Li, "Robust interval linear programming for environmental decision making under uncertainty," in *Journal of Engineering Optimization*, vol. 44, no. 11, November 2012, pp. 1321-1336.
- [9] A. H. Hossny; D. Creighton; S. Nahavandi, "Reducing the impact of bounded parametric uncertainty on Hodgson's scheduling algorithm using interval programming," in *IEEE Systems Journal*, Volume PP, Issue 99, September 2015, pp.1-11.
- [10] H. Hagnas, "Type-2 fuzzy logic controllers: A way forward for fuzzy systems in real world environments," in *IEEE World Congress on Computational Intelligence, WCCI 2008*, Hong Kong, China, June 1-6, 2008, pp. 181-200.
- [11] P. Raajan, P. Surekha and S. Sumathi, "A methodology to schedule and optimize job shop scheduling using computational intelligence paradigms," in *International Conference on Intelligent Control and Information Processing*, Dalian, 2010, pp. 809-814.
- [12] A. Mohamed, H. Hagnas, A. Liret, S. Shakya and G. Owusu, "A genetic interval type-2 fuzzy logic based approach for operational resource planning," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Hyderabad, 2013, pp. 1-8.
- [13] E. Ferreyra, H. Hagnas, A. Mohamed and G. Owusu, "A type-2 fuzzy logic system for engineers estimation in the workforce allocation domain," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Naples, 2017, pp. 1-6.
- [14] R. Morrison, *Data-driven organization design*, 1<sup>st</sup> ed. Croydon: Kogan Page Limited, 2015, pp. 296-308.
- [15] A. Starkey, H. Hagnas, S. Shakya, and G. Owusu, "A genetic type-2 fuzzy logic based approach for the optimal allocation of mobile field engineers to their working areas," in *IEEE International Conference on Fuzzy Systems*, Istanbul, November 2015, pp. 1-8.
- [16] J. Clark and G. Daigle, "The importance of simulation techniques in its research and analysis," in *Winter Simulation Conference Proceedings*, Dec. 1997, pp. 1236-1243.
- [17] M. H. Laarabi, "An overview of a multiagent-based simulation system for dynamic management of risk related to dangerous goods transport," in *IEEE International Systems Conference (SysCon)*. Apr. 2013, pp. 830-835.
- [18] H. Nagao, "What is required for data assimilation that is applicable to big data in the solid earth science? Importance of simulation/data-driven data assimilation," in *17<sup>th</sup> International Conference on Information Fusion (FUSION)*, July 2015, pp. 1-6.
- [19] B. Pfitzinger, "Applying simulations: on the importance of the simulation performance," in *Federated Conference on Computer Science and Information Systems (FedCSIS)*, Sep. 2016, pp. 1125-1128.
- [20] S. Bansal, R. Calandra, T. Xiao, S. Levine and C. J. Tomiini, "Goal-driven dynamics learning via Bayesian optimization," in *IEEE 56<sup>th</sup> Annual Conference on Decision and Control (CDC)*, Melbourne, VIC, 2017, pp. 5168-5173.
- [21] N. Chen, N. Cardozo and S. Clarke, "Goal-Driven Service Composition in Mobile and Pervasive Computing," in *IEEE Transactions on Services Computing*, vol. 11, no. 1, pp. 49-62, Jan.-Feb. 1 2018.
- [22] J. Mendel, *Uncertain rule-based fuzzy logic systems*, 2<sup>nd</sup> ed. Springer, 2017, pp. 24-25, 68, 87-93, 161.
- [23] H. Hagnas, "A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots," in *IEEE Transactions on Fuzzy Systems*, Volume 12, Issue 4, August 2004, pp. 524-539.
- [24] D. Ashock, *Evolutionary computation form modeling and optimization*, 1<sup>st</sup> ed. Springer, 2010, pp. 1-14.
- [25] O. K. Erol and I. Eksin, "A new optimizatin method: Big Bang-Big Crunch," in *Advances in Engineering Software*, vol. 37, no. 2, 2006, pp. 106-111.
- [26] M. Afshar and S. Talatahari, "Cosntrained big bang-big crunch algorithm for optimal solution of large scale reservoir operation problem," in *International Journal of Optimisation and Civil Engineering*, vol. 2, 2011, pp. 357-375.
- [27] K. Bijari, H. Zare, H. Veisi and H. Bobarshad, "Memory enriched big bang big crunch optimization algorithm for data clustering," in *Neural Comput & Applic*, 2016.
- [28] T. Kumbasar and H. Hagnas, "Big Bang-Big Crunch optimization based interval type-2 fuzzy PID cascade controller design strategy," in *Journal of Information Sciences*, October 2014, pp. 277-295.
- [29] B. Yao, H. Hagnas, D. Alghazzawi, M. Alhaddad, "A big bang-big crunch type-2 fuzzy logic system for machine vision-based event detection and summarization in real-world ambient assisted living," in *IEEE Transactions on Fuzzy Systems*, 2016.
- [30] B. Yao, H. Hagnas, M. Alhaddad, D. Alghazzawi, "A big bang-big crunch optimization for a type-2 fuzzy logic based human behaviour recognition system in intelligent environments," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Manchester, UK, Oct. 2013.
- [31] J. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact, well separated cluster," in *Cybernetics*, Volume 3, no. 3, 1973, pp. 32-57.
- [32] N. Pal and J. Bezdek, "On cluster validity for the fuzzy c-means model," in *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 3, Aug. 1995, pp. 370-379.
- [33] A. K. Jain, "Data clustering: 50 years beyond K-means", in *Pattern Recognition Letters*, Volume 31, Issue 8, 2010, pp. 651-666.
- [34] E. Backer, *Cluster Analysis by Optimal Decomposition of Induced Fuzzy Sets*. Delft University Press. 1978.
- [35] S. Eschrich, J. Ke, L.O. Hall, D.B. Goldgof, "Fast accurate fuzzy clustering through data reduction," in *IEEE Trans. Fuzzy Systems*, Volume 11, no. 2, pp. 262-270, 2003.
- [36] C. Lynch. H. Hagnas, V. Callaghan, "Embedded interval type-2 neuro-fuzzy speed controller for marine diesel engines", in *Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2006)*, Paris, France, July 2006, pp. 1340-1347.
- [37] H. Hagnas, F. Doctor, V. Callaghan, and A. Lopez, "An incremental adaptive lifelong learning approach for type-2 fuzzy embedded agents in ambient intelligent environments," in *IEEE Transactions on Fuzzy Systems*, Volume 15, Issue 1, 2007, pp. 41-55.
- [38] D. Wu, "Twelve considerations in choosing between Gaussian and trapezoidal membership functions in interval type-2 fuzzy logic controllers," in *IEEE International Conference on Fuzzy Systems*, Brisbane, QLD, 2012, pp. 1-8.