2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

# A heuristic fuzzy algorithm bio-inspired by Evolution Strategies for energy forecasting problems

Vitor N. Coelho, Frederico G. Guimarães, Agnaldo J. R. Reis, Igor M. Coelho
Bruno N. Coelho and Marcone J. F. Souza

***Abstract*— Improving the use of energy resources has been a great challenge in the last years. A new complex scenario involving a decentralized bidirectional communication between energy suppliers, distribution system and consumption is nowadays becoming reality. Sometimes cited as the largest and most complex machine ever built, Electric Grids (EG) are been transformed into Smart Grids (SG). Hence, the load forecasting problem has become more difficulty and more autonomous load predictors are needed in this new conjecture. In this paper a novel method, so-called MSES, bio-inspired by Evolution Strategies (ES) combined with Multi-Start (MS) procedure is described. This procedure is mainly based on a self-adaptive algorithm to calibrate the parameters of the fuzzy rules. MSES was implemented in C++ via OptFrame framework. Our main goal is to evaluate the performance of this algorithm in a grid environment. Real data from an electric utility have been used in order to test the proposed methodology. The obtained results are fully described and analyzed.**

## I. Introduction

ELECTRIC grids are changing from a centralized single supply model towards a decentralized bidirectional grid of suppliers and consumers. In this new environment, so-called Smart Grid (SG), a dynamic scenario filled with uncertainty is reality.

Rogers et al. [1] highlight that the demand side, consumers, will have to adapt to the available resources, in contrast to the current model in which the supply should always match the demand. In most countries, the migration to this new business model and the implementation of the SG has as its starting point the installation of smart meters [2] and sensors in residences and commercial buildings. On the other hand, if one considers the incentives and global efforts to reduce emissions of greenhouse gases, the supply side also becomes increasingly complex and difficult instead of being managed, since the final amount of energy in the power system is going to be formed by conventional central-station power units and a growing number of systems for generating renewable energy (primarily wind and photovoltaic units [3]).

Considering measurement systems with fast acquisitions rates (short-term), typically 15 to 30 minutes, one can expect a large amount of detailed data about the condition of the electrical network to be converted into valuable and useful information. These data are not only useful in single class problems, these huge datasets are now available in different ways to allow researchers from distinct areas to develop solutions for multifunctional and high complex problems. This current work is mainly focused on load forecasting in a SG scenario, whereas traditional forecasting methods have many limitations to tackle big data [4].

Predictive energy models can be developed using various Artificial Intelligence techniques, such as Artificial Neural Networks (ANN), fuzzy logic, data mining, mathematical programming and heuristic methods [5], [6], [7], [8], [9], [10], [11]. The need to develop high accurate models for energy consumption forecasting is imminent, starting from simple data mining and noise suppression methods to more complete and efficient machine learning algorithms.

Grosman & Lewin [12] use an algorithm based on the concept of Genetic Programming – GP [13] to generate a prediction model for dynamic control with nonlinear assumptions. Kashid & Maity [14] proposes a model based on GP for summer monsoon rains forecasting across India territory. Vladislavleva et al. [15] perform a forecasting model for predicting power output of wind farms based on meteorological data, using a hybrid method, integrating symbolic regression with GP. Recently, Çelekli et al. [16] propose a hybrid model, combining ANN with Gene Expression Programming (GEP) [17], to a manufacturing metallurgy problem, involving the forecasting of sorption of an azo-metal.

In terms of heuristic fuzzy algorithms, different approaches can be found in the literature. Huarng [18] proposed heuristic models by integrating problem-specific heuristic knowledge with Chens model to improve forecasting, the heuristic knowledge was used to guide the search for suitable fuzzy sets for index forecasting. Sousa & Asada [19] proposed a heuristic algorithm that employs fuzzy logic to the power system transmission expansion planning problem. It was based on the divide to conquer strategy, which is controlled by a fuzzy system. The algorithm provides high quality solutions with the use of fuzzy decision making. Recently, Zacharia & Nearchou [20] proposed a metaheuristic algorithm for the

Vitor N. Coelho is with the Graduate Program in Electrical Engineering, Universidade Federal de Minas Gerais, Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, MG, Brazil (email: vncoelho@ufmg.br).

Frederico G. Guimarães is with the Departament of Electrical Engineering, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brazil (email: fredericoguimaraes@ufmg.br).

Agnaldo J. R. Reis and Bruno N. Coelho are with the Department of Control Engineering and Automation, Universidade Federal de Ouro Preto, Ouro Preto, MG, Brazil (email: {agnreis, brunonazario}@gmail.com).

Igor M. Coelho is with the Institute of Computing, Universidade Federal Fluminense, Niteri, RJ, Brazil (email: imcoelho@ic.uff.br).

Marcone J. F. Souza is with the Department of Computer Science, Universidade Federal de Ouro Preto, Ouro Preto, MG, Brazil (email: marcone@iceb.ufop.br).

This work has been supported by the Brazilian agency CAPES. The authors also would like to thank Brazilian agency CNPq (grants 306694/2013-1, 552289/2011-6 and 306458/2010-1) and FAPEMIG (grants PPM CEX 497-13, APQ-04611-10), for supporting the development of this work.

fuzzy assembly line balancing type-E problem, a generalized version of the problem with fuzzy task processing time and line efficiency maximization was considered. Cheok et al. [21] combine bigram Markov language models with heuristic fuzzy rules to improve statistical language models recognition accuracy. Heuristic procedures were used to determine the confidence of the bigram model score and optical character recognition results, in way to improve the overall recognition accuracy.

In order to deal with energy forecasting problems, the use of a class of bio-inspired metaheuristics combined with the fuzzy logic is proposed. Consequent values of each fuzzy rule are summed and the mean of this sum is considered as the forecast value. The proposed approach in this paper incorporates the power of the Evolutionary Algorithms to adapt these fuzzy rules, calibrating it and improving the forecasting performance of the proposed model.

In this context, *Evolution Strategies* – ES [22], stands out as a robust and flexible framework which has been effectively applied to many combinatorial optimization problems [23], [24], however, up to the moment, with only sparse/none applications in forecasting and prediction area. Thus, we propose a heuristic algorithm based on Multi-Start – MS [25] and ES metaheuristics. MS was used to generate good initial solutions and ES procedure to refine these solutions.

The remainder of this paper is organized as follows. Section II describes the generic problems where the new strategy is aiming to tackle. Section III describes the proposed algorithm. Section IV presents the computational experiments, and, finally, Section V draws the final considerations and future work.

## II. Forecasting Problems

The class of forecasting problems tackled in this work can be defined as a set of discrete or real values $M$, where $M = \{m_1, m_2, ..., m_n\}$, such that $m_i \in \mathbb{R}$, represent the $n$ available measured values to perform the forecasting of finite sequences $P$, where $P = (p_{t+1}, p_{t+2}, ..., p_{t+k})$ ,or simply $P_n$ [26], such that $p_i \in \mathbb{R}$, $k$ indicates the number of steps to be predicted.

In this context, the complexity of the problem is directly related to several factors such as:

- amount of missing data and outliers belonging to the training set $M$;
- measurement errors in $M$;
- distribution and behavior of the data set $M$ (non-stationarity, degree of non-linearity, seasonality, tendency, etc.);
- size of the training set $M$;
- number of steps $k$ to be forecast;
- among others.

In view of these basic attributes, the proposed method can be adapted to tackle different areas and forecasting problems, such as load forecasting in SG environment [27], and also other areas, such as earthquake prediction [28], wind energy forecasting [29], risk analysis for credit granting, among others.

## III. Methodology

### A. Fuzzy model

The fuzzy model proposed in this paper is presented in Eq. 1.

$$y(o) = \frac{\sum_{i=1}^{z} v_i \omega(o_i - s.A_i) + \sum_{j=1}^{z} w_j \omega(-o_j + s.B_j)}{\sum_{i=1}^{z} \omega(o_i - s.A_i) + \sum_{j=1}^{z} \omega(-o_j + s.B_j)} \quad (1)$$

Each term of this equation is presented below:

- $A$, where $A = \{a_1, a_2, ..., a_z\}$ are the upper bound limits of the fuzzy rules;
- $V$, where $V = \{v_1, v_2, ..., v_z\}$ are the consequent value of each rule in vector $A$;
- $B$, where $B = \{b_1, b_2, ..., b_z\}$ are the lower bound limits of the fuzzy rules;
- $W$, where $W = \{w_1, w_2, ..., w_z\}$ are the consequent value of each rule in vector $B$;
- $\omega$ can be any desired function (e.g trapezoid , sigmoid, Heaviside step, among others).

Such that $y_{i \in \{A,V,B,W\}, j \in O} \in \mathbb{R}$. The notation $s.X$ indicates that the vector $X$ is related to a specific solution $s$.

Finally, these fuzzy rules can be seen as:

1) the difference from a current value $o_i$ and a given rule in vector $s_{A_i}$ is multiplied by a weight $s_{V_i}$.
2) the difference from a current value $o_j$ and a given rule in vector $s_{B_j}$ is multiplied by a weight $s_{B_i}$.

If the $\omega$ function is even the idea of the fuzzy rules $s_{A_i}$ and $s_{B_j}$ would be essentially the same, otherwise, it could be seen as the boundaries demarcating positive and negative differences.

### B. Solution representation

One solution is represented by a matrix $R = [Y]$, being $Y$ a matrix $4 \times |O|$ of forecasting generations, where $O = \{o_1, o_2, ..., o_z\}$ is the options vector with $|O| = z$ options. Possible choices for this options set $O$ are measurements from the set $M$, derivatives and integrals and the mean of these values. The $4$ lines of matrix $Y$ represent the four vectors that generate the forecasting, described in Section III-A.

Figure 1 illustrates a possible solution using three options, $z(K-1)$, $z(K-2)$ and $\frac{z(K-1)+z(K-2)}{2}$ or $o_1, o_2$ and $o_3$, respectively. Values used in this example were chosen arbitrarily. The first option is the measurement value $z(K-1)$, the second option is the measurement $z(K-2)$ and the third one is the average of these values (i.e. $\frac{z(K-1)+z(K-2)}{2}$). $z(K)$ is denoted by the forecast value, if $K \in P$, or measured value, if $K \in M$, for a given time instant $t$. According to the same idea, $z(K-1)$ is predicted or measured value for the previous instant $t-1$. If there is a need to use the values $K$ from the predicted set $P$, the accuracy and precision of the algorithm is impaired. This aspect will be verified in Section IV, since the data used in this work requires the forecasting of the next 24 hours ahead. One last parameter, so-called precision $p$, regulates the number of columns in each option.

For instance, the chosen precision in Figure 1 was $p = 1$, however, if a precision $p = 10$ had been chosen, this example would have had 30 columns (i.e. $p * z$).

$M = \{100, 105, 94, 85, 100, 101, 90, 120, 125, 115, 111\}$

$k = 2$ (i.e. 2 steps ahead)

$$s = \left[\begin{array}{c|ccc} & z(K-1) & z(K-2) & \frac{z(K-1)+z(K-2)}{2} \\ A & 87 & 95 & 103 \\ V & 70 & 80 & 95 \\ B & 100 & 90 & 110 \\ W & 110 & 50 & 80 \end{array}\right]$$

Fig. 1. Solution example

### C. Applying a solution with $\omega$ functions

Like the ANN, the method proposed here has several ways to calibrate and train it, similar to that ones used on neural networks. Algorithm 1 exemplifies how to apply a solution using functions of a defined class $\omega$.

**Algorithm 1**: Apply a solution using $\omega$ functions

**Input**: Solution $s$, options values $O = \{o_1, o_2, o_3\}$
**Output**: Predicted value $y$

1 $y \leftarrow 0$; $m \leftarrow 0$
2 **for** $i \leftarrow 1$ ***To*** $z$ **do**
3 $\quad diffA = o_i - s.A_i$
4 $\quad diffB = -o_i + s.B_i$
5 $\quad applyA = \omega(diffA)$
6 $\quad applyB = \omega(diffB)$
7 $\quad y \leftarrow y + applyA * s.V_i + applyB * s.W_i$
8 $\quad m \leftarrow m + applyA + applyB$
9 **end**
10 $y \leftarrow y/m$
11 **return** $x$

In Lines 3 and 4, differences between the options values and the ones on the solution are calculated. Lines 5 and 6 compute these differences and return abscissas of function $\omega$. In Line 7, estimation value $y$ receives the weighted sum of $s.V_i$ and $s.W_i$. It should be noticed that divisor $m$ is, in this case, the sum of the weights $applyA$ and $applyB$.

In this current version of the proposed algorithm, a sigmoid function, with its range belonging to $[0, 1]$, was also proposed in way to reduce the discontinuity present in the sum of Heaviside functions.

### D. Applying a solution with Heaviside step function

Algorithm 2 exemplifies how to have a predicted value $y$ from a given solution $s$ based on the Heaviside step function.

Line 3 of Algorithm 2 verifies if the option value from the vector $MOF$ (measured or forecast) has its value bigger than option value in vector $s.A_i$. If bigger, the forecasting value $y$ consequently receives the application value of stored in the vector $s.V_i$. The process is analogous for the other two vectors $s.B_i$ and $s.W_i$. Finally, in Line 12, variable $y$ becomes the average of all accepted values.

**Algorithm 2**: Apply a solution based on Heaviside step function

**Input**: Solution $s$, options values $O = \{o_1, o_2, o_3\}$
**Output**: Predicted value $y$

1 $y \leftarrow 0$; $m \leftarrow 0$
2 **for** $i \leftarrow 1$ ***To*** $z$ **do**
3 $\quad$ **if** $o_i > s.A_i$ **then**
4 $\quad\quad y \leftarrow y + s.V_i$
5 $\quad\quad m \leftarrow m + 1$
6 $\quad$ **end**
7 $\quad$ **if** $o_i < s.B_i$ **then**
8 $\quad\quad y \leftarrow y + s.W_i$
9 $\quad\quad m \leftarrow m + 1$
10 $\quad$ **end**
11 **end**
12 $y \leftarrow y/m$
13 **return** $x$

Let, $u(o_i - t) = \begin{cases} 1 & \text{for} \quad o_i \geq t \\ 0 & \text{for} \quad o_i < t \end{cases}$, be the Heaviside step function. Thus,

Thus, in this case, the fuzzy model described in Eq. 1 of Subsection III-A can be rewritten as presented in Eq. 2.

$$y(o) = \frac{\sum_{i=1}^{z} v_i u(o_i - s.A_i) + \sum_{j=1}^{z} w_j u(-o_j + s.B_j)}{\sum_{i=1}^{z} u(o_i - s.A_i) + \sum_{j=1}^{z} u(-o_j + s.B_j)} \tag{2}$$

The problem shown in Figure 1 prompts the forecast of the next two steps ($k = 2$) (two hours ahead) subsequent to the time instant $t$, in which the power consumption is equal to 111. Therefore, in this case, to obtain the step $t + 2$, we must feed solution $s$ with values predicted in previous steps, known as recursive prediction. Figure 2 presents the practical application of this solution.

**Step 1** - Time point $t + 1$

$P = \{\}$

$$s(t+1) = \left[\begin{array}{c|ccc} & 111 & 115 & 113 \\ A & 87 & 95 & 103 \\ V & 110 & 95 & 100 \\ B & 107 & 90 & 114 \\ W & 110 & 50 & 120 \end{array}\right]$$

$y = (110 + 95 + 100 + 120)/4 = 106.25$
$P = \{106.25\}$

**Step 2** - Time point $t + 2$

$P = \{106.25\}$

$$s(t+2) = \left[\begin{array}{c|ccc} & 106,25 & 111 & 108,625 \\ A & 87 & 95 & 103 \\ V & 110 & 95 & 100 \\ B & 107 & 90 & 114 \\ W & 110 & 50 & 120 \end{array}\right]$$

$y = (110 + 110 + 95 + 100 + 120)/5 = 107$
$P = \{106.25, 107\}$

Fig. 2. Applying a solution $s$

### E. Objective function and solution evaluation

Given a set of discrete training points $T \in M | T = \{t_1, t_2, ..., t_\rho\}$, where $\rho \leq n$ is the cardinality of the training set, a solution $s$ is evaluated from the forecasting of some of these known points belonging to the set $T$.

In our approach, evaluating a solution $s$ is basically generate all possible forecasting of the set $T$ and compare it to the measured values. According to the designed options (Section III-B), there is a part of the set that is necessary to perform initial forecasting, and will not be used to evaluate the solution $s$. For instance, the solution example depicted in Figure 1 needs at least two measured points ($z(K-1)$ and $z(K-2)$) to start the solution evaluation. From a given training set $T = [100, 105, 94, 85, 100, 101]$ with $\rho = 6$ is possible to make up to four forecasting (with $k = 1$), based on these forecasting, different evaluation metrics can be used. However, as the number of steps ahead in this example is $k = 2$, only two forecasting could actually be done. Figure 3 exemplifies the evaluation of the solution given in Figure 1 with the training described above.

In the evaluating example presented in Figure 3, $f$ forecasting were made, where $f = (\rho - 2)/k$ and $X = \{x_1, x_2, ..., x_{\rho-2}\}$ is the set of forecasting values. With $k = 2$ and $\rho = 6$, the evaluation was made over the whole of $f = 2$ forecasting of two steps ahead. The forecasting set obtained was $X = \{108.75, 106.25, 115, 95\}$ against the training measured set $TM = \{94, 85, 100, 101\}$. It is emphasized that the values $T - TM = \{100, 105\}$ were only used to generate the initials forecasting.

The goal is to minimize the error $\sum e_i = tm_i - x_i | \forall i \in X$, where $tm_i$ is the measured value at the same instant of time prediction $x_i$. Furthermore, the objective function can be any desired quality indicator or even more than one, in case of multi-objective approaches. In this paper, the main indicator used was the MAPE (Eq. 3).

$$\sum e_i = \frac{abs(tm_i - x_i)}{tm_i} | \forall i \in X \tag{3}$$

$T = \{100, 105, 94, 85, 100, 101\}$
$X = \{\}$

**Forecasting** $f = 1$
$P = \{\}$
**Step 1**

$$s(t+1) = \left[\begin{array}{c|ccc} & 105 & 100 & 102.5 \\ A & 87 & 95 & 103 \\ V & 110 & 95 & 100 \\ B & 107 & 90 & 114 \\ W & 110 & 50 & 120 \end{array}\right]$$

$y = (110 + 110 + 95 + 120)/4 = 108.75$
$P = \{108.75\}$
**Step 2**

$$s(t+2) = \left[\begin{array}{c|ccc} & 108.75 & 105 & 106.97 \\ A & 87 & 95 & 103 \\ V & 110 & 95 & 100 \\ B & 107 & 90 & 114 \\ W & 110 & 50 & 120 \end{array}\right]$$

$y = (110 + 95 + 100 + 120)/4 = 106.25$
$P = \{108.75, 106.25\}$
$X = X \cup P = \{108.75, 106.25\}$

**Forecasting** $f = 2$
$P = \{\}$
**Step 1**

$$s(t+1) = \left[\begin{array}{c|ccc} & 85 & 94 & 89.5 \\ A & 87 & 95 & 103 \\ V & 110 & 95 & 100 \\ B & 107 & 90 & 114 \\ W & 110 & 50 & 120 \end{array}\right]$$

$y = (110 + 120)/2 = 115$
$P = \{115\}$
**Step 2**

$$s(t+2) = \left[\begin{array}{c|ccc} & 115 & 85 & 106.97 \\ A & 87 & 95 & 103 \\ V & 110 & 95 & 100 \\ B & 107 & 90 & 114 \\ W & 110 & 50 & 120 \end{array}\right]$$

$y = (110 + 50 + 100 + 120)/4 = 95$
$P = \{115, 95\}$
$X = X \cup P = \{108.75, 106.25, 115, 95\}$

Fig. 3. Evaluating a solution $s$

### F. The MSES Algorithm

The proposed algorithm, called *MSES*, consists on the combination of the metaheuristic procedures *Multi-Start* – MS [25] and Evolution Strategy – ES [22]. From the MS procedure, the construction phase was used to generate viable and good quality initial solutions, as can be verified in the BuildInitialRandomSolution procedure. The pseudocode is outlined in Algorithm 3.

The initial population of the algorithm (lines 1 to 6 of Algorithm 3) consists of $\mu$ individuals, and it is created according to the BuildInitialRandomSolution procedure (line 2). This procedure consists in calculating the mean and standard deviation of the available training set $T$, from those two values, a normal distribution is used to generate each value of the vectors $A$, $B$, $V$ and $W$. In this work, the options used were ($z(K-1)$, $z(K-2)$, $z(K-24)$, $z(K-168)$, $z(K-336)$, $z(K-504)$, $z(K-672)$, average values, derivative and integral of the last two previous values, $z(K-1)$ - $z(K-2)$). Values $z(K-1)$ and $z(K-2)$ correspond to the last two measured values, $z(K-24)$ is the demand at the same time on the last day, $z(K-168)$ (same time on the last week), $z(K-336)$ (2 weeks), $z(K-504)$ (3 weeks) and $z(K-672)$ (4 weeks) correspond to the demand at the weekday and time on the last four weeks ago, respectively. Average is the mean of those values and integral is the sum of them. Finally, option derivative was based on the instantaneous change rate of the last two measured demand ($z(K-1)$ and $z(K-2)$). For integral option, the initial values of vectors $A$ and $B$ were generated using basic operations involving trapezoids areas. As for the vector of derivatives, the initial values were 0.

Finally, line 4 merges the MS solution $s$ and the matrix of standard deviations $M$ (the use of this matrix can be verified in algorithm 6), generated according to Algorithm 4. In this sense, the following nomenclature is used: let $ind^S$ be the solution $s$ of the individual $ind$; and let $ind^M$ be the matrix with the standard deviation values, with the same size of the

**Algorithm 3**: *MSES*

**Input**: $\gamma$, Function $f(.)$
**Input**: $r$ neighborhoods: $N^{addX_1}$, $N^{addX_2}$, $N^{addX_{...}}$, $N^{addX_r}$
**Output**: Population $Pop$

1 **for** $i \leftarrow 1$ ***to*** $\mu$ **do**
2 $\quad s \leftarrow$ BuildInitialRandomSolution()
3 $\quad M \leftarrow$ BuildStdVectors($startStdDesv$)
4 $\quad ind \leftarrow s + M$
5 $\quad Pop_i \leftarrow ind$
6 **end**
7 **while** *stop criterion not satisfied* **do**
8 $\quad$ **for** $i \leftarrow 1$ ***to*** $\lambda$ **do**
9 $\quad\quad$ Generate a random number $x \in [1, \mu]$
10 $\quad\quad ind \leftarrow Pop_x$
11 $\quad\quad ind \leftarrow$ UpdateParameters($ind$, $\sigma_{update}$)
12 $\quad\quad ind \leftarrow$ ApplyMutation($ind$)
13 $\quad\quad Pop_i^{offspring} \leftarrow ind$
14 $\quad$ **end**
15 $\quad$ **for** $i \leftarrow 1$ ***to*** $\kappa$ **do**
16 $\quad\quad$ Generate a random number $x \in [1, \lambda]$
17 $\quad$ **end**
18 $\quad Pop$ = Selection ($f$, $Pop$, $Pop^{offspring}$)
19 **end**
20 **return** *Pop*

solution $s$ (i.e., same number of rows and columns).

**Algorithm 4**: BuildStdVectors

**Input**: Standard deviation $\sigma_{startStdDesv}$
**Output**: Standard deviation matrix $M$

1 **for** $i \leftarrow 1$ ***to*** $4$ **do**
2 $\quad$ **for** $o \leftarrow 1$ ***to*** $z$ **do**
3 $\quad\quad M_{i,o} \leftarrow \sigma_{startStdDesv}$
4 $\quad$ **end**
5 **end**
6 **return** *M*

In line 11 of Algorithm 3 the mutation procedure is called for the selected individual, the pseudocode of this procedure is described in Algorithm 5.

**Algorithm 5**: UpdateParameters

**Input**: Individual $ind$ and standard deviation $\sigma_{update}$
**Output**: Individual $ind$ with its parameter matrix updated

1 **for** $i \leftarrow 1$ ***to*** $4$ **do**
2 $\quad$ **for** $o \leftarrow 1$ ***to*** $z$ **do**
3 $\quad\quad M_{i,o} \leftarrow M_{i,o} + N(0, \sigma_{update})$
4 $\quad$ **end**
5 **end**
6 **return** *ind*

For each position $(i, o)$ in the matrix $M$ of the Algorithm 5, a normal distribution, centered at mean zero and standard deviation $\sigma_{update}$, is applied to update each value of this matrix, as can be verified at line 3.

The procedure ApplyMutation (line 12 of the Algorithm 3) is illustrated in Algorithm 6.

**Algorithm 6**: ApplyMutation

**Input**: Individual $ind$
**Output**: Individual $ind$ with its solution updated

1 **for** $i \leftarrow 1$ ***to*** $4$ **do**
2 $\quad$ **for** $o \leftarrow 1$ ***to*** $z$ **do**
3 $\quad\quad ind^{S_{i,o}} \leftarrow ind^{S_{i,o}} + N(0, M_{i,o})$
4 $\quad$ **end**
5 **end**
6 **return** $s$

In line 3 of Algorithm 6 each position $(i, o)$ of a solution $s$ related to a individual $ind$ is updated according to a normal distribution with mean equal to zero and std. deviation $M_{i,o}$.

The selection procedure (line 18 of the Algorithm 3) can be any desired selection strategy, as long as the strategy returns a population of size $\mu$. We used two basic forms of competition, both with the same notation of [22]. In the first, denoted by $(\mu + \lambda)$, there is competition between parents and offspring. In this strategy the $\mu$ best individuals are selected among parents and offspring. In the second selection strategy used, denoted by $(\mu, \lambda)$, individuals who survive to the next generation are the $\mu$ best ones in the offspring. It is clear that using the strategy $(\mu, \lambda)$ as a way of selection, the population that survives to the next generation suffers a considerable selective pressure, however, this pressure becomes even greater when the strategy $(\mu+\lambda)$ is used.

## IV. Computational Experiments

The MSES algorithm was implemented in C++ in the framework OptFrame 2.0 [1] [30], [31]. This framework has been successfully applied to other problems in the literature (see [32], [33] and [34]).

It is important to point out that all code used in this research is, from this moment, available as an example on OptFrame core, as an open-source tool under GNU LGPL 3.

The tests were carried out on a OPTIPLEX 9010 Intel Core i7-3770, 3.40 x 8 GHZ with 32GB of RAM, with operating system Ubuntu 12.04.3 precise, and compiled by g++ 4.6.3, using the Eclipse Kepler Release.

The dataset (Figure 4) used to validate the proposal was obtained from the EirGrid Free Data [35]. It consists in intraday electricity demand measurements for the 50 week period from Monday, 7 January 2013 to Sunday, 22 December 2013 (8400 samples). All experiments used hourly data. Two different groups of training and validation set were created. The first one, so-called EG1-1, consists in the first 20 weeks (3360 samples) to train the algorithm, the remaining ten weeks (1680 samples) to evaluate post-sample accuracy of 24 hours ahead forecast, this experiment is similar those

[1] Available at http://sourceforge.net/projects/optframe/

presented in Taylor & McSharry [36], but in that case a private dataset, related to 10 Europeans countries, was used. The second dataset, so-called EG1-2, presented in this paper consists in 40 weeks (6720 samples) to train the algorithm and a training set with the same characteristics as the last one.

After some empirical analysis, MSES deviation parameters were set as $\sigma_{startStdDesv} = 2$ and $\sigma_{update} = 1$.

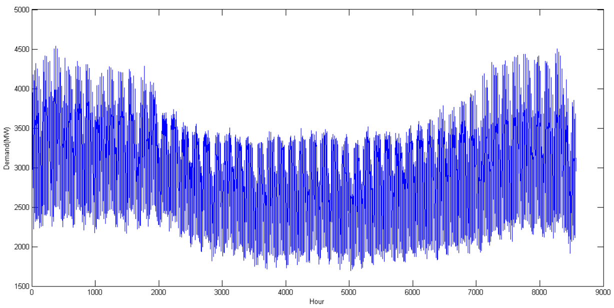


Fig. 4. EirGrid dataset – 07/01/2013 to 22/12/2013

### A. Time-to-Target plot results

In this first experiment, a time-to-target plots (TTTplots) was done to check the efficiency of the proposed algorithm in reaching the solution currently used by the company. Run time distributions or time-to-target plots display, on the ordinate axis, the probability that an algorithm will find a solution at least as good as a given target value within a given running time, shown on the abscissa axis. Time-to-target plots were first used in [37]. Run time distributions have been advocated also in [38] as a way to characterize the running times of stochastic algorithms for combinatorial optimization.

Aiex et. al [39] described a Perl program to create time-to-target plots for measured times that are assumed to fit a shifted exponential distribution, closely following [40]. Such plots are very useful to compare different algorithms or strategies for solving a given problem and have been widely used as a tool for algorithm design and comparison.

A batch of 120 executions was made, algorithm *MSES* was applied to solve the first proposed instance, EG1-1, with a target MAPE equal to 3.6. The performance ended only when the algorithm had found the target value. These times were then sorted in ascending order, so that, for each execution $i = 1, 2, ..., 120$ there is a time $t_i$ and a probability $p_i = (i - 0.5)/120$ associated. A time limit of 1200 seconds was imposed to each execution. If the algorithm could not reach the target solution, its execution was discarded. It should be noticed that the target MAPE is related to the objective function described in Section III-E. This target value does not measure the quality indicator MAPE in the blind training set, that is unknown and not evaluated in this stage.

Table I shows eight variants of the algorithm *MSES*, created from different values for its parameters and fuzzy

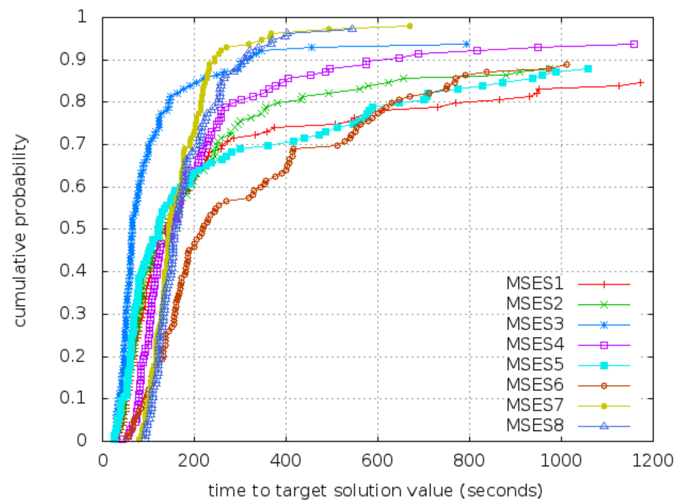


Fig. 5. Superimposed empirical distribution

rules (objective function). Variants 1 to 4 have its population with 30 individuals and 160 offspring, on the other hand variants 5 to 8 have 100 parents and 600 offspring at each iterations. Two different selection strategies were proposed, as described in Section III-F. Two different accuracies, namely precision (as presented in Section III-B), were proposed. The objective function 1 is the one using Heaviside function (Section III-D), and 2 is the sigmoid function (Section III-C).

TABLE I
VARIANTS

| Acronym | $\mu$ | $\lambda$ | Selection | Precision | Objective function |
|---|---|---|---|---|---|
| MSES1 | 30 | 160 | $(\mu + \lambda)$ | 20 | 1 |
| MSES2 | 30 | 160 | $(\mu + \lambda)$ | 5 | 1 |
| MSES3 | 30 | 160 | $(\mu, \lambda)$ | 5 | 1 |
| MSES4 | 30 | 160 | $(\mu, \lambda)$ | 5 | 2 |
| MSES5 | 100 | 600 | $(\mu + \lambda)$ | 5 | 1 |
| MSES6 | 100 | 600 | $(\mu + \lambda)$ | 5 | 2 |
| MSES7 | 100 | 600 | $(\mu, \lambda)$ | 5 | 1 |
| MSES8 | 100 | 600 | $(\mu, \lambda)$ | 20 | 1 |

The results of the graph $t_i \times p_i$ are shown in Figure 5, a superimposed empirical probability curve. It can be seen that variant MSES1 presented the premature convergence, since it did not reach the target 17 times. Variant MSES7 showed the best performance in reaching the given target MAPE, presenting the lowest computational time and the higher probabilities in reaching the target. Heaviside objective function (1) showed the best performance in terms of computational time, on the other hand, sigmoid function (2) reached the target in most time of its executions. This result is entirely consistent, since the sigmoid function has more neighborhoods and probably more ways to scape from an attraction basin. Comparing variants with selection $(\mu, \lambda)$ in relation to $(\mu + \lambda)$, one can conclude that low selective pressure prevents premature convergence by high accurately individuals in the parents set.

*B. Benchmark results*

The MSES7 variant was selected to perform this last computational experiment. It was run 10 times for both proposed instances (EG1-1 and EG1-2) with 20 minutes per training. After the training, the algorithm was applied to a blind forecasting on the remaining ten weeks, as explained in the beginning of this section. As the validation has its computational time less than 0.001ms, it is not considered.

Table II presents the obtained results. Column "Best" present the best forecast, in terms of MAPE indicator, in 10 executions. Columns "Average" and "Std. Desv" indicates the average MAPE of the blind training set and the standard deviation, respectively.

TABLE II
COMPUTATIONAL RESULTS

| | MAPE | | |
|---|---|---|---|
| Instance | Best (%) | Average (%) | Std. Desv |
| Heaviside step function (1) | | | |
| EG1-1 | 3.46 | 3.81 | 0.16 |
| EG1-2 | 2.37 | 2.46 | 0.07 |
| Sigmoide function (2) | | | |
| EG1-1 | 3.34 | 3.76 | 0.31 |
| EG1-2 | 2.29 | 2.36 | 0.09 |

Analyzing Table II, it can be seen that *MSES7* obtained a good quality solution, regarding to MAPE indicator, with standard deviation up to 0.31. Using sigmoid function (Section III-C) the algorithm was able to perform slightly better forecasts than using Heaviside step function (Section III-D), but, this fact should be carefully analyzed in future experiments due to statistical indicators indicating crossed confidence intervals for both algorithms.

Finally, Figure 6 displays a forecasting example with the blind validation set of EG1-2 instance, using the variant *MSES7*, using objective function 2, on its best execution with MAPE equal to 2.29%.

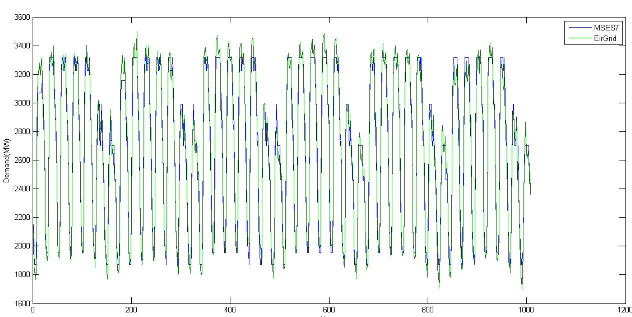


Fig. 6. Forecasting against measured values – EG1-2

## V. CONCLUSIONS AND EXTENSIONS

In this paper, a class of energy load forecasting problem with realistic assumptions was discussed. Despite its practical relevance, this variant of load forecasting has received little attention of heuristic based methods. Because of its difficulty and large number of expected variables in a future Smart Grid (SG) environment, a new framework for forecasting is proposed. This new approach consists on a fuzzy algorithm bio-inspired by Evolution Strategies, so-called *MSES*, combining power of the constructive heuristic Multi-Start with the natural bio-inspired procedure Evolution Strategy, going through a large search space of feasible solutions.

From real based literature datasets, provided by EirGrid [35], the algorithm efficiency was verified. It was capable to find good quality solutions with low variability in reduced computational time. Eight different variants of the *MSES* were proposed, based on time-to-target empirical probability experiment one variant was selected. Selected variant was tested with two different dataset. The one that presented the best results got an average MAPE of 2.37 %, competitive with the current literature, as can be verified in [36]. Overall, the proposed method proved to be a powerful tool that can support short-term forecasting in a 24 hour-ahead model .

The success of the proposed method is promising, particularly in view of the method's flexibility, as it is mainly based on metaheuristics. Therefore, it can be used in various everyday situations with minor adjustments. Ongoing work is aiming to gain insight into the method and apply it to other load forecasting configurations, using different load forecasting horizons, extending the study for short and long term load forecasting.

As future extensions for this work, it is proposed to develop a multi-objective version of the problem addressed, using different quality indicators to evaluate the solutions.

Furthermore, the inclusion of new objectives in vogue on the SG context is proposed. Improve the algorithm to re-train after new data acquisitions is also a requested requirement in the SG. It is also propose to adapt this algorithm to other forecasting problems, such as earthquake prediction, risk analysis for credit granting, among others. Finally, a parallel version of *MSES* would be an import future extension, in order to take advantage of *multi-core* technology already present in current machines and with easy abstraction for heuristic algorithms. Entire code used in this research is, from this moment, available as example on the OptFrame website. Thus, it is expected that future researchers continue contributing to enhancing the proposed method, increasing its efficiency, optimizing neighborhood structures, improving the evaluating function and other mechanisms presented in this paper.

## REFERENCES

[1] A. Rogers, S. D. Ramchurn, and N. R. Jennings, "Challenges for autonomous agents and multi-agent systems research," in *Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI-12)*, Toronto, CA, 2012, pp. 2166–2172.

[2] M. P. McHenry, "Technical and governance considerations for advanced metering infrastructure/smart meters: Technology, security, uncertainty, costs, benefits, and risks," *Energy Policy*, vol. 59, pp. 834 – 842, 2013.

[3] N. Batista, R. Melcio, J. Matias, and J. Catalo, "Photovoltaic and wind energy systems monitoring and building/home energy management using zigbee devices within a smart grid," *Energy*, vol. 49, no. 0, pp. 306 – 315, 2013.

[4] K. Krishnan, "Chapter 1 - introduction to big data," in *Data Warehousing in the Age of Big Data*. Boston: Morgan Kaufmann, 2013, pp. 3 – 14.

[5] A. Reis and A. Alves da Silva, "Feature extraction via multiresolution analysis for short-term load forecasting," *Power Systems, IEEE Transactions on*, vol. 20, no. 1, pp. 189–198, 2005.

[6] E. Delarue and W. D'haeseleer, "Adaptive mixed-integer programming unit commitment strategy for determining the value of forecasting," *Applied Energy*, vol. 85, no. 4, pp. 171 – 181, 2008.

[7] C. A. Georgopoulou and K. C. Giannakoglou, "Two-level, two-objective evolutionary algorithms for solving unit commitment problems," *Applied Energy*, vol. 86, no. 7-8, pp. 1229 – 1239, 2009.

[8] A. Chakrabarty, S. Banerjee, S. Maity, and A. Chatterjee, "Fuzzy model predictive control of non-linear processes using convolution models and foraging algorithms," *Measurement*, vol. 46, no. 4, pp. 1616 – 1629, 2013.

[9] S. Das, S. Das, and I. Pan, "Multi-objective optimization framework for networked predictive controller design," {*ISA*} *Transactions*, vol. 52, no. 1, pp. 56 – 77, 2013.

[10] E. Fallah-Mehdipour, O. B. Haddad, and M. Mario, "Prediction and simulation of monthly groundwater levels by genetic programming," *Journal of Hydro-environment Research*, no. 0, pp. –, 2013.

[11] K. Salahshoor, E. Safari, and I. Ahangari, "A novel adaptive fuzzy predictive control for hybrid systems with mixed inputs," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 5 - 6, pp. 1512 – 1531, 2013.

[12] B. Grosman and D. R. Lewin, "Automated nonlinear model predictive control using genetic programming," *Computers & Chemical Engineering*, vol. 26, no. 4 - 5, pp. 631 – 640, 2002.

[13] J. R. Koza, *Genetic programming: on the programming of computers by means of natural selection*. Cambridge, MA, USA: MIT Press, 1992.

[14] S. S. Kashid and R. Maity, "Prediction of monthly rainfall on homogeneous monsoon regions of india based on large scale circulation patterns using genetic programming," *Journal of Hydrology*, vol. 454 - 455, no. 0, pp. 26 – 41, 2012.

[15] E. Vladislavleva, T. Friedrich, F. Neumann, and M. Wagner, "Predicting the energy output of wind farms based on weather data: Important variables and their correlation," *Renewable Energy*, vol. 50, no. 0, pp. 236 – 243, 2013.

[16] A. elekli, H. Bozkurt, and F. Geyik, "Use of artificial neural networks and genetic algorithms for prediction of sorption of an azo-metal complex dye onto lentil straw," *Bioresource Technology*, vol. 129, no. 0, pp. 396 – 401, 2013.

[17] Y. Yang, X. Li, L. Gao, and X. Shao, "A new approach for predicting and collaborative evaluating the cutting force in face milling based on gene expression programming," *Journal of Network and Computer Applications*, no. 0, pp. –, 2013.

[18] K. Huarng, "Heuristic models of fuzzy time series for forecasting," *Fuzzy Sets and Systems*, vol. 123, no. 3, pp. 369 – 386, 2001.

[19] A. S. Sousa and E. N. Asada, "Combined heuristic with fuzzy system to transmission system expansion planning," *Electric Power Systems Research*, vol. 81, no. 1, pp. 123 – 128, 2011.

[20] P. Zacharia and A. C. Nearchou, "A meta-heuristic algorithm for the fuzzy assembly line balancing type-e problem," *Computers & Operations Research*, vol. 40, no. 12, pp. 3033 – 3044, 2013.

[21] A. D. Cheok, Z. Jian, and E. S. Chng, "Efficient mobile phone chinese optical character recognition systems by use of heuristic fuzzy rules and bigram markov language models," *Applied Soft Computing*, vol. 8, no. 2, pp. 1005 – 1017, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494607000889

[22] H. G. Beyer and H. P. Schwefel, "Evolution strategies - a comprehensive introduction," *Natural Computing*, vol. 1, pp. 3–52, 2002.

[23] T. Chen and H. Chen, "Mixed-discrete structural optimization using a rank-niche evolution strategy," *Engineering Optimization*, vol. 41, no. 1, pp. 39–58, 2009.

[24] L. Costa and P. Oliveira, "Evolutionary algorithms approach to the solution of mixed integer non-linear programming problems," *Computers & Chemical Engineering*, vol. 25, no. 10, pp. 257–266, 2001.

[25] R. Mart, "Multi-start methods," in *Handbook of Metaheuristics*, ser. International Series in Operations Research & amp; Management Science, F. Glover and G. Kochenberger, Eds. Springer New York, 2003, vol. 57, pp. 355–368.

[26] E. L. Lima, *Analysis course*. Rio de Janeiro: Association National Institute of Pure and Applied Mathematics: Euclides project, 2009, vol. 1.

[27] Y.-S. Lee and L.-I. Tong, "Forecasting nonlinear time series of energy consumption using a hybrid dynamic model," *Applied Energy*, vol. 94, no. 0, pp. 251 – 256, 2012.

[28] Y. Jafarian, E. Kermani, and M. H. Baziar, "Empirical predictive model for the vmax/amax ratio of strong ground motions using genetic programming," *Computers & Geosciences*, vol. 36, no. 12, pp. 1523 – 1531, 2010.

[29] A. M. Foley, P. G. Leahy, A. Marvuglia, and E. J. McKeogh, "Current methods and advances in forecasting of wind power generation," *Renewable Energy*, vol. 37, no. 1, pp. 1 – 8, 2012.

[30] I. M. Coelho, P. L. A. Munhoz, M. N. Haddad, V. N. Coelho, M. M. Silva, M. J. F. Souza, and L. S. Ochi, "A computational framework for combinatorial optimization problems," in *VII ALIO/EURO Workshop on Applied Combinatorial Optimization*, Porto, 2011, pp. 51–54.

[31] I. M. Coelho, S. Ribas, M. H. P. Perche, P. L. A. Munhoz, M. F. Souza, and L. S. Ochi, "Optframe: a computational framework for combinatorial optimization problems," in *XLII Simpsio Brasileiro de Pesquisa Operacional*, Bento Gonalves, RS, 2010, pp. 1887 – 1898.

[32] M. J. F. Souza, I. M. Coelho, S. Ribas, H. G. Santos, and L. H. C. Merschmann, "A hybrid heuristic algorithm for the open-pit-mining operational planning problem," *European Journal of Operational Research, EJOR*, vol. 207, pp. 1041–1051, 2010.

[33] I. M. Coelho, P. L. A. Munhoz, M. N. Haddad, M. J. F. Souza, and L. S. Ochi, "A hybrid heuristic based on general variable neighborhood search for the single vehicle routing problem with deliveries and selective pickups," *Electronic Notes in Discrete Mathematics*, vol. 39, no. 0, pp. 99 – 106, 2012.

[34] V. N. Coelho, M. J. F. Souza, I. M. Coelho, F. G. Guimaraes, T. Lust, and R. C. Cruz, "Multi-objective approaches for the open-pit mining operational planning problem," *Electronic Notes in Discrete Mathematics*, vol. 39, no. 0, pp. 233 – 240, 2012.

[35] EirGrid National Control Center. [Online]. Available: http://www.eirgrid.com/operations/systemperformancedata/systemdemand/

[36] J. Taylor and P. McSharry, "Short-term load forecasting methods: An evaluation based on european data," *Power Systems, IEEE Transactions on*, vol. 22, no. 4, pp. 2213–2219, 2007.

[37] T. A. Feo, M. G. C. Resende, and S. H. Smith, "A greedy randomized adaptive search procedure for maximum independent set," *Operations Research*, vol. 42, pp. 860–878, 1994.

[38] C. C. Ribeiro and M. G. C. Resende, "Path-relinking intensification methods for stochastic local search algorithms," *Journal of Heuristics*, vol. 18, pp. 193–214, 2012.

[39] R. Aiex, M. Resende, and C. Ribeiro, "Tttplots: a perl program to create time-to-target plots," *Optimization Letters*, vol. 1, pp. 355–366, 2007.

[40] R. M. Aiex, M. G. C. Resende, and C. C. Ribeiro, "Probability distribution of solution time in GRASP: An experimental investigation," *Journal of Heuristics*, vol. 8, pp. 343–373, 2002.