Development of a Fuzzy Rule-Based System using Genetic Programming for Forecasting Problems

Adriano S. Koshiyama, Marley M.B.R. Vellasco and Ricardo Tanscheit
Department of Electrical Engineering
Pontifical Catholic University of Rio de Janeiro
Rua Marquês de São Vicente, 225, Gávea
Rio de Janeiro, RJ – Brazil, PO Box: 38097, Tel.: +55 21 3527-1001
Email: [adriano,marley,ricardo]@ele.puc-rio.br

***Abstract*—This work presents a novel genetic fuzzy system for forecasting, called Genetic Programming Fuzzy Inference System for Forecasting problems (GPFIS-Forecast), which generates an interpretable fuzzy rule base by using Multi-Gene Genetic Programming to define the premises terms of fuzzy rules. The main differences between GPFIS-Forecast and other genetic fuzzy systems lie in its fuzzy inference process, because it: (*i*) enables premises to be include negation, t-conorm and linguistic hedge operators; (*ii*) applies methods to define a consequent term more compatible with a given premise; and (*iii*) makes use of aggregation operators to weigh fuzzy rules in accordance with their influence on the problem. GPFIS-Forecast has been tested in the NN3 Competition, in order to evaluate its performance in a benchmark problem. In this case, it has produced competitive results when compared to other forecasting approaches.**

I. Introduction

Classical forecasting approaches employ econometric tools, which are strongly based on statistical models: ARIMA, GARCH, (S)TAR and their generalized versions [1]–[4]. These models normally impose some restrictions on time series characteristics (e.g., stationary, normally distributed noise, etc.). Computational intelligence models, such as Artificial Neural Networks, have also been successfully used [1], [5]–[7], albeit at the expense of interpretability. Genetic Fuzzy Systems (GFSs) [8]–[10] are another alternative, given their capability of combining reasonable accuracy and interpretability.

What mainly distinguishes GFSs from statistical and other computational intelligence models is their capacity of drawing knowledge from databases to generate a system that is linguistically comprehensible and that produces results with reasonable accuracy. This is due to the association of a Fuzzy Inference System (FIS) with a Genetic-Based Meta-Heuristic (GBMH) – based on Darwin's concept of natural selection and genetic recombination. Most of the works in GFSs deal with Classification, Control and Regression problems [8], [10]–[13], while the forecasting area is less explored. In fact GFSs based on Genetic Programming [14], [15] have never been applied in forecasting, despite being suitable for problems with variable-length code [9] such as a fuzzy rule base.

This work proposes the Genetic Programming Fuzzy Inference System for Forecasting problems (GPFIS-Forecast) model. The main characteristics of GPFIS-Forecast are: (*i*) it employs Multi-Gene Genetic Programming [16], [17] to build the premises of fuzzy rules with t-norm, linguistic hedge, negation operators, etc.; (*ii*) it connects those premises to suitable consequent terms; (*iii*) it makes use of aggregation operators to weigh fuzzy rules according to their influence. GPFIS-Forecast has applied on the last 11 time series of NN3 Competition [18], in order to be evaluated in a benchmark problem.

The following section presents a background section: this embeds a related works on GFSs applied to forecasting problems and a description of the Multi-Gene Genetic Programming main features. Section 3 exhibits GPFIS-Forecast model, while Section 5 outlines some case studies. Finally, section 6 concludes the work.

II. Background

A. Related Works

This section presents a brief review on applications of Genetic Fuzzy Systems (GFSs) to forecasting problems.

Bergmeir et al. [19] introduce a GBMH called Memetic Algorithms, in order to fit transition parameters of a Neuro-Coefficient Smooth Transition Autoregressive Model (NC-STAR) [20]. This model corresponds to a Takagi-Sugeno-Kang FIS with logistic membership functions. The Memetic Algorithm is used for adjusting transition between models, logistic membership functions slopes, lags selection and weights given to exogenous variables. An empirical evaluation is performed by applying it to various real-world time series originated from three forecasting competitions. Aznarte et al. [21] developed a GFS based in NCSTAR model in two steps: (*i*) initially elaborate a fuzzy rule base using Evolutionary Strategy (1+1)-ES [22] for financial time series forecasting, and (*ii*) perform a fuzzy rule subset selection, using a binary coding GA [23]. It was applied to 23 financial time series and compared to three statistical models.

Hadavandi et al. [24] proposed the integration of a GFS and data clustering to build an expert system for sales forecasting. At first, all records of data are grouped into k clusters by using k-means. Then, for each cluster, an independent GFS is used for parameter tuning and fuzzy rule base learning. The model has been applied to a printed circuit board (PCB) sales forecasting problem and compared to other sales forecasting methods. Finally, Chen et al. [25] present a GFS with a hierarchical Takagi-Sugeno-Kang FIS. Database parameters are fine-tuned by Probabilistic Incremental Program Evolution, while the fuzzy rule base is generated by Evolutionary Programming.

Its performance is evaluated through a Mackey-Glass time series. It should be mentioned that there are several works in forecasting that employ Fuzzy Logic ([26]), [27], [28]), but these are not related to the GFSs area.

In general, most of these works show little regard to interpretability issues. Besides, they do not explore linguistic hedges and negation operators. Procedures for the selection of consequent terms have not been reported and few works weigh fuzzy rules. In addition, GFSs based on Genetic Programming have never been applied to forecasting problems. The GPFIS-Forecast proposed here addresses all those issues, aiming at allying accuracy and interpretability.

B. Multi-Gene Genetic Programming

Genetic Programming (GP) [14], [15] belongs to the Evolutionary Computation field. Typically, it employs a population of individuals, each of them denoted by a tree structure that codifies a mathematical equation, which describes the relationship between the output Y and a set of input variables X_j $(j = 1, ..., J)$. Based on these ideas, Multi-Gene Genetic Programming (MGGP) [16], [17], [29], [30] generalizes GP as it denotes an individual as a structure of trees, also called genes, that similarly receives X_j and tries to predict Y.

Each individual is composed of D trees or functions ($d = 1, ..., D$) that relate X_j to Y through user-defined mathematical operations. It is easy to verify that MGGP generates solutions similar to those of GP when $D = 1$. In GP terminology, the X_j input variables are included in the Terminal Set, while the mathematical operations (plus, minus, etc.) are part of the Function Set (or Mathematical Operations Set).

With respect to genetic operators, mutation in MGGP is similar to that in GP. As for crossover, the level at which the operation is performed must be specified: it is possible to apply crossover at high and low levels. Figure 1a presents a multi-gene individual with five equations ($D = 5$) accomplishing a mutation, while Figure 1b shows the low level crossover operation.

The low level is the space where it is possible to manipulate structures (Terminals and Mathematical Operations) of equations present in an individual. In this case, both operations are similar to those performed in GP. The high level, on the other hand, is the space where expressions can be manipulated in a macro way. An example of high level crossover is shown in Figure 1c. By observing the dashed lines it can be seen that the equations were switched from an individual to the other. The cutting point can be symmetric – the same number of equations is exchanged between individuals –, or asymmetric. Intuitively, high level crossover has a deeper effect on the output than low level crossover and mutation have.

In general, the evolutionary process in MGGP differs from that in GP due to the addition of two parameters: maximum number of trees per individual and high level crossover rate. A high value is normally used for the first parameter to assure a smooth evolutionary process. On the other hand, the high level crossover rate, similarly to other genetic operators rates, needs to be adjusted.

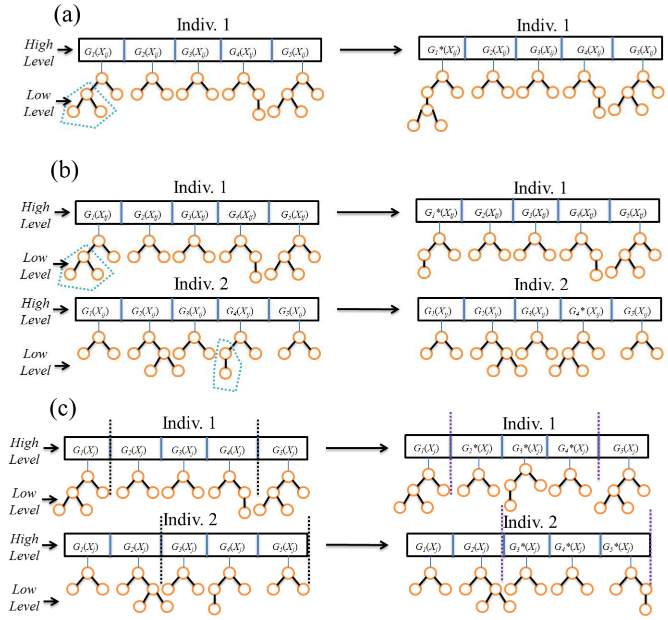

Fig. 1. Application example of MGGP operators: (a) mutation; (b) low level crossover; and (c) high level crossover.

III. GPFIS-Forecasting Model

GPFIS-Forecast is a typical Pittsburgh-type GFS [8]. Its development begins with the mapping of crisp values into membership degrees to fuzzy sets (Fuzzification). Then, the fuzzy inference process is divided into three subsections: (*i*) generation of fuzzy rule premises (Formulation); (*ii*) assignment of a consequent term to each premise (Splitting) and (*iii*) aggregation of each activated fuzzy rule (Aggregation). Finally, Defuzzification and Evaluation are performed.

A. Fuzzification

In univariate time series analysis, the main information for forecasting the behaviour of a time series $y_t \in Y$ at instant t $(t = 1, ..., T)$ consists of its respective P lags $y_{t-1} \in Y_1, ..., y_{t-p} \in Y_p, ..., y_{t-P} \in Y_P$ $(p = 1, ..., P)$. L fuzzy sets are associated to each p-th lag and are given by $A_{lp} = \{(y_{t-p}, \mu_{A_{lp}}(y_{t-p}))|y_{t-p} \in Y_p\}$, where $\mu_{A_{lp}} : Y_p \to [0, 1]$ is a membership function that assigns to each observation y_{t-p} a membership degree $\mu_{A_{lp}}(y_{t-p})$ to a fuzzy set A_{lp}. Similarly, for Y (output variable), K fuzzy sets B_k $(k = 1, ..., K)$ are associated.

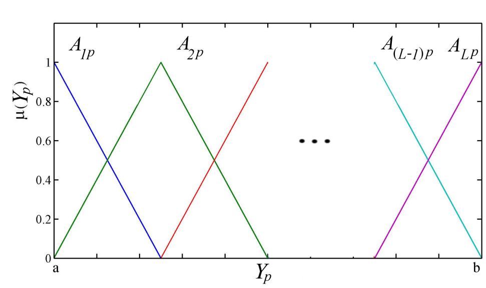

Fig. 2. Membership functions to $y_{t-p} \in Y_p$ variables. For Y variable, substitute occasionally each A_{lp} for B_k.

Three aspects are taken into account when defining the membership functions: (*i*) form (triangular, trapezoidal, etc.); (*ii*) support set of $\mu_{A_{lp}}(y_{t-p})$; (*iii*) an appropriate linguistic term, qualifying the subspace constituted by $\mu_{A_{lp}}(y_{t-p})$ with a context-driven adjective. Ideally, these tasks should be carried out by an expert, whose knowledge would improve comprehensibility. In practice, it is not always easy to find a suitable expert. Therefore it is very common [8], [31], [32] to define membership functions as shown in Figure 2.

Five triangular membership functions ($l = 1, ..., 5$) , strongly partitioned (as in Figure 2), have been used throughout the experiments in this work.

B. Fuzzy Inference

1) Formulation: A fuzzy rule premise is commonly defined by:

"If Y_1 is A_{l1} and ... and Y_p is A_{lp} and ... and Y_P is A_{lP}"

or, in mathematical terms:

$$\mu_{A_d}(\mathbf{y}_{t,P}) = \mu_{A_{l1}}(y_{t-1}) * ... * \mu_{A_{lP}}(y_{t-P}) \tag{1}$$

where $\mu_{A_d}(y_{t-1}, ..., y_{t-P}) = \mu_{A_d}(\mathbf{y}_{t,P})$ is the joint membership degree of P lags of y_t with respect to the d-th premise ($d = 1, ..., D$), computed by using a t-norm $*$. Note that $\mathbf{y}_{t,P} = [y_{t-1}, ..., y_{t-p}, ..., y_{t-P}]$. A premise can be elaborated by using t-norms, t-conorms, linguistic hedges and negation operators to combine the $\mu_{A_{lp}}(y_{t-p})$. As a consequence, the number of possible combinations grows as the number of variables, operators and fuzzy sets increase. Therefore, GPFIS-Forecast employs MGGP to search for the most promising combinations, i.e., fuzzy rule premises. Figure 3 exemplifies a typical solution provided by MGGP.

For example, premise 1 represents: $\mu_{A_1}(\mathbf{y}_{t,P}) = \mu_{A_{21}}(y_{t-1}) * \mu_{A_{32}}(y_{t-2})$ and, in linguistic terms, "*If Y_1 is A_{21} and Y_2 is A_{32}*". Let $\mu_{A_d}(\mathbf{y}_{t,P})$ be the d-th premise codified in the d-th tree of an MGGP individual. Table I presents the components used for reaching the solutions shown in Figure 3.

TABLE I. INPUT FUZZY SETS AND FUZZY OPERATORS USED TO GENERATE THE SOLUTION ILLUSTRATED IN FIGURE 3.

Input Fuzzy Sets (Terminals Set)	Fuzzy Operators Set (Functions Set)
$\mu_{A_{11}}(y_{t-1})$, ..., $\mu_{A_{L1}}(y_{t-1})$, ..., $\mu_{A_{lp}}(y_{t-p})$, ..., $\mu_{A_{LP}}(y_{t-P})$	t-norm ($*$), linguistic hedge (dilatation operator – $\sqrt{}$) and classical negation operator

In GPFIS-Forecast, the set of $\mu_{A_{lp}}(y_{t-p})$ represents the Input Fuzzy Sets or, in GP terminology, the Terminal Set, while the Functions Set is replaced by the Fuzzy Operators Set. Thus MGGP is used for obtaining a set of fuzzy rules premises $\mu_{A_d}(\mathbf{y}_{t,P})$. In order to fully develop a fuzzy rule base, it is necessary to define the consequent term best suited to each $\mu_{A_d}(\mathbf{y}_{t,P})$.

2) Splitting: There are two ways to define which consequent term is best suited to a fuzzy rule premise: (*i*) allow a GBMH to perform this search (a common procedure in several works); or (*ii*) employ methods that directly draw information from the dataset so as to connect a premise to a consequent term. In GPFIS-Forecast the second option has been adopted in order to prevent a premise with a large coverage in the dataset, or able to predict a certain region of the output, to be associated to an unsuitable consequent term. Instead of searching for all elements of a fuzzy rule, as a GBMH does, GPFIS-Forecast measures the compatibility between $\mu_{A_d}(\mathbf{y}_{t,P})$ and the consequent terms. This also promotes reduction of the search space.

The Fuzzy Confidence Degree (FCD_k) is given by:

$$FCD_k = \frac{\sum_{t=1}^{T} \mu_{A_d}(\mathbf{y}_{t,P})\ \mu_{B_k}(y_t)}{\sqrt{\sum_{t=1}^{T} \mu_{A_d}(\mathbf{y}_{t,P})^2}\ \sqrt{\sum_{t=1}^{T} \mu_{B_k}(y_t)^2}} \in [0,1] \tag{2}$$

where $\sum_{t=1}^{T} \mu_{A_d}(\mathbf{y}_{t,P})\ \ \mu_{B_k}(y_t)$ is the compatibility between the d-th premise and the k-th consequent term, while $\sqrt{\sum_{t=1}^{T} \mu_{A_d}(\mathbf{y}_{t,P})^2}$ and $\sqrt{\sum_{t=1}^{T} \mu_{B_k}(y_t)^2}$ are normalizing factors. When $\mu_{A_d}(\mathbf{y}_{t,P}) = \mu_{B_k}(y_t)$ for all t, then $FCD_k = 1$, i.e., premise and consequent term are totally compatible. A consequent term for $\mu_{A_d}(\mathbf{y}_{t,P})$ is selected as the k-th consequent which maximize FCD_k. A premise with $FCD_k = 0$, for all k, is not associated to any consequent term (and not considered as a fuzzy rule).

3) Aggregation: A premise associated to the k-th consequent term (i.e., a fuzzy rule) is denoted by $\mu_{A_{d(k)}}(\mathbf{y}_{t,P})$, which, in linguistic terms, means: "If Y_1 is A_{l1}, and ..., and Y_P is A_{lP}, then Y is B_k". Therefore, the whole fuzzy rule base is given by $\mu_{A_{1(k)}}(\mathbf{y}_{t,P})$, ..., $\mu_{A_{D(k)}}(\mathbf{y}_{t,P})$, $\forall k = 1, ..., K$. A new observation y_t^* of the time series and its respective P lags $\mathbf{y}_{t,P}^*$ may have a non zero membership degree to several premises, associated either to the same or to different consequent terms. In order to generate a consensual value, the aggregation step tries to combine the activation degrees of all fuzzy rules associated to the same consequent term.

Consider $D^{(k)}$ as the number of fuzzy rules associated to k-th consequent term ($d^{(k)} = 1^{(k)}, 2^{(k)} ..., D^{(k)}$). Given an aggregation operator $g : [0,1]^{D^{(k)}} \rightarrow [0,1]$ (see [33], [34]), the predicted membership degree of $\mathbf{y}_{t,P}^*$ to each k-th consequent term ($\hat{\mu}_{B_k}(y_t^*)$) is computed by:

$$\hat{\mu}_{B_1}(y_t^*) = g[\mu_{A_{1(1)}}(\mathbf{y}_{t,P}^*), ..., \mu_{A_{D(1)}}(\mathbf{y}_{t,P})^*] \tag{3}$$

$$\hat{\mu}_{B_2}(y_t^*) = g[\mu_{A_{1(2)}}(\mathbf{y}_{t,P}^*), ..., \mu_{A_{D(2)}}(\mathbf{y}_{t,P}^*)] \tag{4}$$

$$...$$

$$\hat{\mu}_{B_K}(y_t^*) = g[\mu_{A_{1(K)}}(\mathbf{y}_{t,P}^*), ..., \mu_{A_{D(K)}}(\mathbf{y}_{t,P}^*)] \tag{5}$$

There are many aggregation operators available (e.g., see [34]), the Maximum being the most widely used [35]. Nevertheless other operators such as arithmetic and weighted averages may also be used. As for weighted arithmetic mean, it is necessary to solve a Restricted Least Squares problem (RLS) in order to establish the weights:

$$min : \sum_{t=1}^{T} (\hat{\mu}_{B_k}(y_t) - \sum_{d^{(k)}=1}^{D^{(k)}} w_{d^{(k)}} \mu_{A_{d^{(k)}}}(\mathbf{y}_{t,P}))^2 \tag{6}$$

$$s.t. : \sum_{d^{(k)}=1}^{D^{(k)}} w_{d^{(k)}} = 1\ and\ w_{d^{(k)}} \geq 0$$

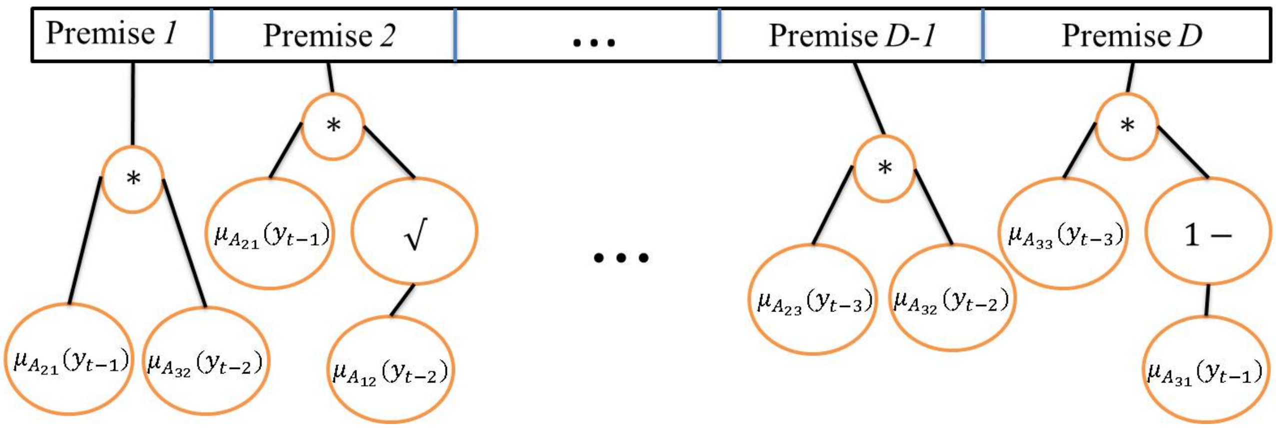

Fig. 3. Example of fuzzy rule premises codified in an MGGP individual.

where $w_{d^{(k)}}$ is the weight or the influence degree of $\mu_{A_{d(k)}}(\mathbf{y}_{t,P})$ in the prediction of elements related to the k-th consequent term. This is a typical Quadratic Programming problem, the solution of which is easily computed by using algorithms discussed in [36].

C. Defuzzification

Proposition 1: Consider $y_t \in Y$, with $a \leq y_t \leq b$ where $a, b \in \mathbb{R}$, and, associated to Y, K triangular membership functions, normal, 2-overlapped[1] and strongly partitioned (identical to Figure 2). Then y_t can be rewritten as [37]:

$$y_t = c_1\mu_{B_1}(y_t) + c_2\mu_{B_2}(y_t) + ... + c_J\mu_{B_K}(y_t) \tag{7}$$

where c_1, ..., c_K is the "center" ($\mu_{B_k}(c_k) = 1$) of each k-th membership function ($k = 1, ..., K$).

This linear combination, which is a defuzzification procedure, is usually known as the Height Method. From this proposition, the following conclusions can be drawn:

1) If $\mu_{B_k}(y_t)$ is known, then y_t is also known.
2) If only a prediction $\hat{\mu}_{B_k}(y_t)$ of $\mu_{B_k}(y_t)$ is known, such that $\sup_{y_t} |\mu_{B_k}(y_t) - \hat{\mu}_{B_k}(y_t)| \leq \epsilon$, when $\epsilon \rightarrow 0$ the defuzzification output $\hat{y}_t$ that approximates y_t is given by:

$$\hat{y}_t = c_1\hat{\mu}_{B_1}(y_t) + c_2\hat{\mu}_{B_2}(y_t) + ... + c_K\hat{\mu}_{B_K}(y_t) \tag{8}$$

When $\hat{\mu}_{B_k}(y_t) \approx \mu_{B_k}(y_t)$ is not verified, the Mean of Maximum or the Center of Gravity defuzzification methods may provide a better performance. However, due to the widespread use of strongly partitioned fuzzy sets in the experiments with GPFIS-Forecast, a normalized version of the Height Method (8) has been employed:

$$\hat{y}_t = \frac{c_1\hat{\mu}_{B_1}(y_t) + ... + c_K\hat{\mu}_{B_K}(y_t)}{\hat{\mu}_{B_1}(y_t) + ... + \hat{\mu}_{B_K}(y_t)} \tag{9}$$

It is possible to evaluate an individual of GPFIS-Forecast by using $\hat{y}_t$.

[1] a fuzzy set is normal if it has some element with maximum membership equal to 1. Also, fuzzy sets are 2-overlapped if $\min(\mu_{B_u}(y_t), \mu_{B_z}(y_t), \mu_{B_v}(y_t)) = 0, \forall u, v, z \in k = 1, ..., K$

D. Evaluation

The Evaluation procedure in GPFIS-Forecast is defined by a primary objective – error minimization – and a secondary objective – complexity reduction. The primary objective is responsible for ranking individuals in the population, while the secondary one is used as a tiebreaker criteria.

A simple fitness function for forecasting problems is the Symmetric Mean Absolute Percentage Error ($SMAPE$):

$$SMAPE = \frac{\sum_{t=1}^{T} \frac{|y_t - \hat{y}_t|}{(y_t + \hat{y}_t)/2}}{T} \tag{10}$$

The best individual in the population is the solution which minimizes equation (10). GPFIS-Forecast tries to reduce the complexity of the rule base by employing a simple heuristic: Lexicographic Parsimony Pressure [38]. This technique is only used in the selection phase: given two individuals with the same fitness, the best one is that with fewer nodes. Fewer nodes indicate rules with fewer antecedent elements, linguistic hedges and negation operators, as well as few premises ($\mu_{A_d}(\mathbf{y}_{t,P})$), and, therefore, a small fuzzy rule set. After evaluation, a set of individuals is selected (through a tournament procedure) and recombined. Mutation, low-level crossover or high-level crossover are applied to some subset. Finally, the new population is generated.

This process is repeated until a stopping criteria is met. When this occurs, the final population is returned and the best individual provides its forecasts.

IV. Case Studies

A. Experiments Description

The NN3 Competition (Neural Network Competition) occurred in 2007 as a special session of the International Joint Conference on Neural Networks (IJCNN[2]). The competition's aim was to perform an empirical evaluation of several forecasting models, mostly Artificial Neural Networks, to verify which of them produced the best out-of-sample accuracy. It should be mentioned that, as no applications of genetic fuzzy systems

[2] http:\\www.ijcnn2007.org\competition.htm

could be found in this competition[3], comparisons between GPFIS-Forecast and other GFSs could not be performed.

Table II summarizes the main features of the NN3 Competition.

TABLE II. MAIN FEATURES OF NN3 COMPETITION.

Number of Competitors	33
Number of Time Series	111
Periodicity	Monthly
Time Series Minimum\Maximum Size	69\114
Evaluation Measure	SMAPE (10)
Forecasting Horizon	18-months ahead

Results could be reported considering the whole set of 111 time series or the last 11 ones. Some competitors submitted results for the last 11 time series only. The competition considered a submission (i.e., a competitor) only if the method was considered as a new approach. If a forecasting method had been previously published or largely employed in the industry, it was considered as a forecasting benchmark model. In this present work is presented some preliminary results only over the last 11 time series of this competition.

Table III presents the main configurations of GPFIS-Forecast. The SMAPE in-sample was used as the fitness function. The experimental procedure consisted of 2 stages: (*i*) GPFIS-Forecast was executed ten times, performing 10,000 evaluations for each time series; (*ii*) for each model, multi-step forecasts 18-steps ahead were performed and the SMAPE and Root Mean Squared-Error (RMSE) were reported for the out-of-sample set. The parameters' values have been defined as a result of preliminary tests.

TABLE III. MAIN CONFIGURATIONS USED IN GPFIS-FORECAST MODEL.

Parameter	Value
Population Size	100
Number of Generations	100
Tree Maximum Depth	5
Tournament Size	2
High Level Crossover Rate	50%
Low Level Crossover Rate	85%
Mutation Rate	15%
Elitism Rate	1%
Input Fuzzy Sets	5 fuzzy sets, displayed like Figure 2
Fuzzy Operators	Product, Classical Negation and Square-Root
Splitting Method	FCD
Aggregation Operator	RLS
Defuzzification	Height Method

B. Results and Discussions

Table IV exhibits the SMAPE, RMSE, average number of rules and premise elements per rule obtained by GPFIS-Forecast for every time series in the out-of-sample period. The lowest SMAPE was 2,5619 % for the NN3-101 time series, while the highest was 67,8101% for the NN3-110 time series.

Table V shows the position of GPFIS-Forecast with respect to other competitors[4]. It can be seen that GPFIS-Forecast remains at second position with an average SMAPE difference of 0.53% to that of Yan's model, a Generalized Regression Neural Network. The average number of fuzzy rules for this subset was 15.48, where each fuzzy rule is composed of 2.46 premise elements in average. Data in Table IV enable any models to be fully compared with GPFIS-Forecast.

TABLE IV. SMAPE AND RMSE RESULTS OF GPFIS-FORECAST MODEL OBTAINED IN THE OUT-OF-SAMPLE PERIOD. OBSERVATION: # R – AVERAGE NUMBER OF RULES AND # A – AVERAGE PREMISE ELEMENTS PER RULE.

Time Series	RMSE	SMAPE (%)	# R	# A
NN3-101	167.09	2.5619	17.61	2.85
NN3-102	1540.97	25.0294	11.35	3.14
NN3-103	4880.03	18.1570	14.74	2.14
NN3-104	690.62	8.7113	18.28	2.56
NN3-105	230.12	3.8457	20.37	1.89
NN3-106	273.19	4.7937	20.44	2.23
NN3-107	181.48	4.1587	11.62	1.88
NN3-108	1343.28	30.7407	12.83	2.54
NN3-109	414.41	10.2083	11.05	2.43
NN3-110	832.04	30.8678	13.50	2.65
NN3-111	704.85	17.2718	18.45	2.75

TABLE V. AVERAGE RESULTS ON LAST 11 TIME SERIES FOR NN3 COMPETITION.

Rank	Method-Team	SMAPE*
-	*CI Benchmark* - Theta AI (Nikolopoulos)	13,07%
-	*Stat. Benchmark* - Autobox (Reily)	13,49%
-	*Stat. Benchmark* - ForecastPro (Stellwagen)	13,52%
1	Yan	13,68%
-	*Stat. Benchmark* - Theta (Nikolopoulos)	13,70%
2	**GPFIS-Forecast**	**14,21%**
3	llies, Jager, Kosuchinas, Rincon, Sakenas and Vaskevcius	14,26%
4	Chen and Yao	14,46%
5	Yousefi, Miromeni and Lucas	14,49%
6	Ahmed, Atiya, Gayar and El-Shishiny	14,52%
7	Flores, Anaya, Ramirez e Morales	15,00%
8	Adeodato, Vasconcelos, Arnaud, Chunha e Monteiro	15,10%
	Stat. Contender - Wildi	15,32%
9	Luna, Soares e Ballini	15,35%
10	Theodosiou e Swamy	16,19%

* - Out-of-sample period.

An example of a rule base generated by GPFIS-Forecast for the time series NN3-103 is shown in Table VI. This rule base has been extracted from the best individual in the population, whose SMAPE out-of-sample is similar to the average value presented in Table IV. The five linguistic terms are: Very Low (VL), Low (L), Medium (M), High (H), Very High (VH). Rule 1, for example, should be read as: "If Y_{t-12} is Very (Very Low), then Y_t is Very Low, with weight equal to 1". As this is the only rule for the consequent term VL, the weight is equal to one due to the RLS operator. In addition, this rule exemplifies the use of the square-root modifier: the linguistic term Very Low has been changed to Very (Very Low). The use of negation allows GPFIS-Forecast to model more complex time series, but it can bee noticed that this results in some lengthy rules, which does not favor interpretability.

With respect to computation time, GPFIS-Forecast has taken, in average, one minute to run for each of the time series on an Intel i5 processor with 4GB of RAM.

V. CONCLUSION

This work has presented a novel Genetic Fuzzy System (GFS) called Genetic Programming Fuzzy Inference System

[3]http:\\www.neural-forecasting-competition.com\NN3\results.htm

[4]Submissions in Italic are statistical and computational intelligence methods that entered the competition as "benchmarks".

TABLE VI. FUZZY RULE BASE FOR NN3-103 TIME SERIES. OBSERVATION: V(.) REPRESENT THE CONCENTRATION OPERATOR (SQUARE-ROOT) APPLIED ON A LINGUISTIC TERM.

Rule	Premise	Consequent	Weight
R1	If Y_{t-12} is V(VL)	VL	1.00
R2	If Y_{t-6} is not L and Y_{t-4} is not L and Y_{t-3} is not VL	L	0.12
R3	If Y_{t-11} is not VH and Y_{t-10} is not L and Y_{t-9} is not M and Y_{t-1} is V(L)	L	0.17
R4	If Y_{t-1} is not L and Y_{t-12} is L and Y_{t-9} is not L and Y_{t-7} is H	L	0.71
R5	If Y_{t-5} is not VH and Y_{t-3} is not VH and Y_{t-10} is not L and Y_{t-7} is M and Y_{t-6} is M and Y_{t-1} is V(M)	M	1.00
R6	If Y_{t-6} is not L and Y_{t-3} is L	H	0.19
R7	If Y_{t-11} is not VH and Y_{t-10} is not L Y_{t-9} is not M and Y_{t-12} is not L	H	0.52
R8	If Y_{t-1} is V(H) and Y_{t-6} is not VH and Y_{t-4} is not H and Y_{t-9} is not VH	H	0.19
R9	If Y_{t-2} is VH and Y_{t-8} is M and Y_{t-9} is M and Y_{t-9} is not M and Y_{t-10} is not L	VH	0.82
R10	If Y_{t-4} is V(VL) and Y_{t-10} is not L and Y_{t-11} is not VH	VH	0.18

for Forecasting problems (GPFIS-Forecast). By using Multi-Gene Genetic Programming to elaborate premise terms of fuzzy rules, it generates an interpretable fuzzy rule base. Differences between GPFIS-Forecast and other GFSs have been reported, mainly with regard to the fuzzy inference process, since GPFIS-Forecast: (*i*) allows the use of premises composed of negation, t-conorms and linguistic hedge operators; (*ii*) applies procedures to associate a fuzzy rule premise to a suitable consequent term; (*iii*) employs aggregation operators to weigh fuzzy rules according to their influence on the problem. GPFIS-Forecast presented competitive results in the NN3 benchmark, either from prediction or interpretability perspectives.

Future works should employ GPFIS-Forecast to the whole benchmark set of NNe (111 time series) and to forecast economic and financial variables, such as inflation and unemployment rate, or asset prices. A multivariate extension of GPFIS-Forecast to forecast several time series simultaneously shall also be considered. In this case, it will be necessary to develop new procedures to associate a premise to a consequent term (or to multiple consequent terms). Furthermore aggregation and defuzzification strategies will have to take into account the interaction between the time series and combine them to find a more interpretable and accurate multivariate system.

REFERENCES

[1] M. C. Medeiros, A. Veiga, and C. E. Pedreira, "Modeling exchange rates: smooth transitions, neural networks, and linear models," *IEEE Transactions on Neural Networks*, vol. 12, no. 4, pp. 755–764, 2001.

[2] K. M. Dominguez, "When do central bank interventions influence intra-daily and longer-term exchange rate movements?" *Journal of International Money and Finance*, vol. 25, no. 7, pp. 1051 – 1071, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0261560606000738

[3] A. L. Villavicencio, "Nonlinearities or outliers in real exchange rates?" *Economic Modelling*, vol. 25, no. 4, pp. 714 – 730, 2008.

[4] R. Kruse, M. Frömmel, L. Menkhoff, and P. Sibbertsen, "What do we know about real exchange rate nonlinearities?" *Empirical Economics*, vol. 43, no. 2, pp. 457–474, 2012.

[5] M. Sewell and J. Shawe-Taylor, "Forecasting foreign exchange rates using kernel methods," *Expert Systems with Applications*, vol. 39, no. 9, pp. 7652 – 7662, 2012.

[6] B. Majhi, M. Rout, R. Majhi, G. Panda, and P. J. Fleming, "New robust forecasting models for exchange rates prediction," *Expert Systems with Applications*, vol. 39, no. 16, pp. 12 658 – 12 670, 2012.

[7] F. Bahramy and S. F. Crone, "Forecasting foreign exchange rates using support vector regression," in *2013 IEEE Conference on Computational Intelligence for Financial Engineering Economics (CIFEr)*, 2013, pp. 34–41.

[8] F. Herrera, "Genetic fuzzy systems: taxonomy, current research trends and prospects," *Evolutionary Intelligence*, vol. 1, no. 1, pp. 27–46, 2008.

[9] O. Cordón, F. Gomide, F. Herrera, F. Hoffmann, and L. Magdalena, "Ten years of genetic fuzzy systems: Current framework and new trends," *Fuzzy Sets and Systems*, vol. 41, pp. 5–31, 2004.

[10] M. Fazzolari, R. Alcala, Y. Nojima, H. Ishibuchi, and F. Herrera, "A review of the application of multiobjective evolutionary fuzzy systems: Current status and further directions," *Fuzzy Systems, IEEE Transactions on*, vol. 21, no. 1, pp. 45–65, 2013.

[11] R. Alcalá, M. J. Gacto, and F. Herrera, "A fast and scalable multiobjective genetic fuzzy system for linguistic fuzzy modeling in high-dimensional regression problems," *Fuzzy Systems, IEEE Transactions on*, vol. 19, no. 4, pp. 666–681, 2011.

[12] J. A. Sanz, M. Galar, A. Jurio, A. Brugos, M. Pagola, and H. Bustince, "Medical diagnosis of cardiovascular diseases using an interval-valued fuzzy rule-based classification system," *Applied Soft Computing*, no. 0, p. A publicar, 2013.

[13] A. Tsakonas, "Local and global optimization for takagi–sugeno fuzzy system by memetic genetic programming," *Expert Systems with Applications*, vol. 40, no. 8, pp. 3282–3298, 2013.

[14] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: MIT Press, 1992.

[15] R. Poli, W. B. Langdon, and N. F. McPhee, *A Field Guide to Genetic Programming*. Rayleigh: Lulu.com, 2008.

[16] D. Searson, M. Willis, and G. Montague, "Co-evolution of non-linear PLS model components," *Journal of Chemometrics*, vol. 21, no. 12, pp. 592–603, 2007.

[17] A. H. Gandomi and A. H. Alavi, "A new multi-gene genetic programming approach to nonlinear system modeling. part i: materials and structural engineering problems," *Neural Computing and Applications*, vol. 21, no. 1, pp. 171–187, 2012.

[18] S. F. Crone, M. Hibon, and K. Nikolopoulos, "Advances in forecasting with neural networks? empirical evidence from the nn3 competition on time series prediction," *International Journal of Forecasting*, vol. 27, no. 3, pp. 635–660, 2011.

[19] C. Bergmeir, I. Triguero, D. Molina, J. L. Aznarte, and J. M. Benitez, "Time series modeling and forecasting using memetic algorithms for regime-switching models," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 11, pp. 1841–1847, 2012.

[20] M. C. Medeiros and A. Veiga, "A flexible coefficient smooth transition time series model," *IEEE Transactions on Neural Networks*, vol. 16, no. 1, pp. 97–113, 2005.

[21] J. L. Aznarte, J. Alcalá-Fdez, A. Arauzo-Azofra, and J. M. Benítez, "Financial time series forecasting with a bio-inspired fuzzy model," *Expert Systems with Applications*, vol. 39, no. 16, pp. 12 302–12 309, 2012.

[22] O. Cordón and F. Herrera, "A two-stage evolutionary process for designing tsk fuzzy rule-based systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 29, no. 6, pp. 703–715, 1999.

[23] ——, "Hybridizing genetic algorithms with sharing scheme and evolution strategies for designing approximate fuzzy rule-based systems," *Fuzzy Sets and Systems*, vol. 118, no. 2, pp. 235–255, 2001.

[24] E. Hadavandi, H. Shavandi, and A. Ghanbari, "An improved sales forecasting approach by the integration of genetic fuzzy systems and data clustering: Case study of printed circuit board," *Expert Systems with Applications*, vol. 38, no. 8, pp. 9392–9399, 2011.

[25] Y. Chen, B. Yang, A. Abraham, and L. Peng, "Automatic design of hierarchical takagi-sugeno type fuzzy systems using evolutionary algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 3, pp. 385–397, 2007.

[26] I.-H. Kuo, S.-J. Horng, T.-W. Kao, T.-L. Lin, C.-L. Lee, and Y. Pan, "An improved method for forecasting enrollments based on fuzzy time series and particle swarm optimization," *Expert Systems with Applications*, vol. 36, no. 3, Part 2, pp. 6108–6117, 2009.

[27] V. Leite, "An classification analysis of lithology using svm, mlp and ensemble methods."

[28] W. Stach, L. A. Kurgan, and W. Pedrycz, "Numerical and linguistic prediction of time series with the use of fuzzy cognitive maps," *Fuzzy Systems, IEEE Transactions on*, vol. 16, no. 1, pp. 61–72, 2008.

[29] M. Hinchliffe, H. Hiden, B. McKay, M. Willis, M. Tham, and G. Barton, "Modelling chemical process systems using a multi-gene genetic programming algorithm," in *Late Breaking Papers at the Genetic Programming 1996 Conference Stanford University July 28-31, 1996*, J. R. Koza, Ed. Stanford University, CA, USA: Stanford University, 1996, pp. 56–65.

[30] K. A. Fattah, "A new approach calculate oil-gas ratio for gas condensate and volatile oil reservoirs using genetic programming," *Oil and Gas Business*, vol. 1, pp. 311–323, 2012.

[31] F. Berlanga, A. Rivera, M. del Jesus, and F. Herrera, "Gp-coach: Genetic programming-based learning of compact and accurate fuzzy rule-based classification systems for high-dimensional problems," *Information Sciences*, vol. 180, no. 8, pp. 1183–1200, 2010.

[32] H. Ishibuchi, M. Yamane, and Y. Nojima, "Rule weight update in parallel distributed fuzzy genetics-based machine learning with data rotation," in *In IEEE International Conference on Fuzzy Systems, 2013. FUZZ-IEEE 2013*. IEEE, 2013, pp. 1–8.

[33] G. Beliakov, A. Pradera, and T. Calvo, *Aggregation functions: A guide for practitioners*. Heidelberg: Springer Publishing Company, 2008.

[34] T. Calvo, A. Kolesárová, M. Komorníková, and R. Mesiar, "Aggregation operators: Properties, classes and construction methods," in *Aggregation Operators*, ser. Studies in Fuzziness and Soft Computing, T. Calvo, G. Mayor, and R. Mesiar, Eds. Physica-Verlag HD, 2002, vol. 97, pp. 3–104.

[35] W. Pedrycz and F. Gomide, *An introduction to fuzzy sets: analysis and design*. Massachussets: MIT Press, 1998.

[36] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Massachussets: MIT Press, 2001.

[37] W. Pedrycz, *Granular Computing: Analysis and Design of Intelligent Systems*. Boca Raton: CRC Press, 2013.

[38] S. Luke and L. Panait, "Lexicographic parsimony pressure," in *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference*, W. B. Langdon, E. Cantú-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M. A. Potter, A. C. Schultz, J. F. Miller, E. Burke, and N. Jonoska, Eds. New York: Morgan Kaufmann Publishers, 2002, pp. 829–836.