

A Novel Genetic Fuzzy System for Regression Problems

Adriano S. Koshiyama, Marley M.B.R. Velasco
and Ricardo Tanscheit

Abstract Solving a regression problem is equivalent to finding a model that relates the behavior of an output or response variable to a given set of input or explanatory variables. An example of such a problem would be that of a company that wishes to evaluate how the demand for its product varies in accordance to its and other competitors' prices. Another example could be the assessment of an increase in electricity consumption due to weather changes. In such problems, it is important to obtain not only accurate predictions but also interpretable models that can tell which features, and their relationship, are the most relevant. In order to meet both requirements—linguistic interpretability and reasonable accuracy—this work presents a novel Genetic Fuzzy System (GFS), called Genetic Programming Fuzzy Inference System for Regression problems (GPFIS-Regress). This GFS makes use of Multi-Gene Genetic Programming to build the premises of fuzzy rules, including in it t-norms, negation and linguistic hedge operators. In a subsequent stage, GPFIS-Regress defines a consequent term that is more compatible with a given premise and makes use of aggregation operators to weigh fuzzy rules in accordance with their influence on the problem. The system has been evaluated on a set of benchmarks and has also been compared to other GFSs, showing competitive results in terms of accuracy and interpretability issues.

1 Introduction

Regression problems are widely reported in the literature [1, 4, 17, 24, 30]. Generalized Linear Models [27], Neural Networks [18] and Genetic Programming [23] tend to provide solutions with high accuracy. However, high precision is not

A.S. Koshiyama (✉) · M.M.B.R. Velasco · R. Tanscheit
Department of Electrical Engineering, Pontifical Catholic University
of Rio de Janeiro, Rio de Janeiro, RJ, Brazil
e-mail: adriano@ele.puc-rio.br

M.M.B.R. Velasco
e-mail: marley@ele.puc-rio.br

R. Tanscheit
e-mail: ricardo@ele.puc-rio.br

© Springer International Publishing Switzerland 2016
M. Collan et al. (eds.), *Fuzzy Technology*, Studies in Fuzziness
and Soft Computing 335, DOI 10.1007/978-3-319-26986-3_5

always associated to a reasonable interpretability, that is, it may be difficult to identify, in linguistic terms, the relation between the response variable (output) and the explanatory variables (inputs).

A GFS integrates a Fuzzy Inference System (FIS) and a Genetic Based Meta-Heuristic (GBMH), which is based on Darwinian concepts of natural selection and genetic recombination. Therefore, a GFS provides fair accuracy and linguistic interpretability (FIS component) through the automatic learning of its parameters and rules (GBMH component) by using information extracted from a dataset or a plant. The number of works related to GFSs applied to regression problems has increased over the years and are mostly based on improving the Genetic Based Meta-Heuristic counterpart of GFSs by using Multi-Objective Evolutionary Algorithms [1, 5, 31]. In general most of these works 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 regression problems.

This work presents a novel GFS called Genetic Programming Fuzzy Inference System for Regression problems (GPFIS-Regress). The main characteristics of this model are: (i) it makes use of Multi-Gene Genetic Programming [21, 34], a Genetic Programming generalization that works on a single-objective framework, which can be more reliable computationally in some situations than multi-objective approaches; (ii) it employs aggregation, negation and linguistic hedge operators in a simplified manner; (iii) it applies some heuristics to define the consequent term best suited to a given antecedent term.

This work is organized as follows: Sect. 2.1 presents some related works on GFSs applied to regression problems and Sect. 2.2. covers the main concepts of the GBMH used in GPFIS-Regress: Multi-Gene Genetic Programming. Section 3 presents the GPFIS-Regress model; case studies are dealt with in Sect. 4. Section 5 concludes the work.

2 Background

2.1 Related Works

In general, GFSs designed for solving regression problems are similar to those devised for classification. This is due to the similarity between those problems, except for the output variable: in regression the consequent term is a fuzzy set, while in classification it is a classical set. Nevertheless, in both cases interpretability is a relevant requirement. Therefore, most works in this subject employ Multi-Objective Evolutionary Algorithms (MOEAs) as the GBMH for rule base synthesis. One of the few that does not follow this concept is that of Alcalá et al. [2], which presents one of the first applications of 2-tuple fuzzy linguistic representation [20]. In this work a GFS, based on a Genetic Algorithm (GA), learns both the granularity and

the displacement of the membership functions for each variable. Wang and Mendel's algorithm [35] is used for rule generation. This model is applied to two real cases.

The work of Antonelli et al. [5] proposes a multi-objective GFS to generate a Mamdani-type FIS, with reasonable accuracy and rule base compactness. This system learns the granularity and the fuzzy rule set (a typical knowledge-base discovery approach). It introduces the concept of virtual and concrete rule base: the virtual one is based on the highest number of membership functions for each variable, while the concrete one is based on the values observed in the individual of the MOEA population. This algorithm was applied to two benchmarks for regression.

Pulkkinen and Koivisto [31] present a GFS that learns most of the FIS parameters. A MOEA is employed for fine-tuning membership functions and for defining the granularity and the fuzzy rule base. A feature selection procedure is performed before initialization and an adaptable solution from Wang & Mendel's algorithm is included in an individual as an initial seed. This model performs equally or better than other recent multi-objective and single-objective GFSs for six benchmark problems.

The recent work of Alcalá et al. [1] uses a MOEA for accuracy and comprehension maximization. It defines membership functions granularities for each variable and uses Wang & Mendel's algorithm for rule generation. During the evolutionary process, membership functions are displaced following a 2-tuple fuzzy linguistic representation, as stated earlier. In a post-processing stage fine-tuning of the membership functions is performed. The proposed approach compares favorably to four other GFSs for 17 benchmark datasets.

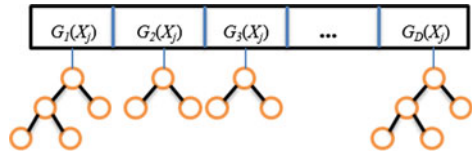
Benítez and Casillas [8] present a novel multi-objective GFS to deal with high-dimensional problems through a hierarchical structure. This model explores the concept of Fuzzy Inference Subsystems, which compose the hierarchical structure of a unique FIS. The MOEA has a 2-tuple fuzzy linguistic representation that indicates the displacement degree of triangular membership functions and which variables will belong to a subsystem. The fuzzy rule base is learned through Wang & Mendel's algorithm. This approach is compared to other GFSs for five benchmark problems.

Finally, Márquez et al. [26] employ a MOEA to adapt the conjunction operator (a parametric t-norm) that combines the premise terms of each fuzzy rule in order to maximize total accuracy and reduce the number of fuzzy rules. An initial rule base is generated through Wang & Mendel's approach [35], followed by a screening mechanism for rule set reduction. The codification also includes a binary segment that indicates which rules are considered in the system, as well as an integer value that represents the parametric t-norm to be used in a specific rule. An experimental study carried out with 17 datasets of different complexities attests the effectiveness of the mechanism, despite the large number of fuzzy rules.

2.2 Multi-Gene Genetic Programming

Genetic Programming (GP) [23, 30] belongs to the Evolutionary Computation field. Typically, it employs a population of individuals, each of them denoted by

Fig. 1 Example of multi-gene individual



a tree structure that codifies a mathematical equation that describes the relationship between the output Y and a set of input variables X_j ($j = 1, \dots, J$). Based on these ideas, Multi-Gene Genetic Programming (MGGP) [15, 17, 21, 34] 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 (Fig. 1).

Each individual is composed of D trees or functions ($d = 1, \dots, D$) 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).

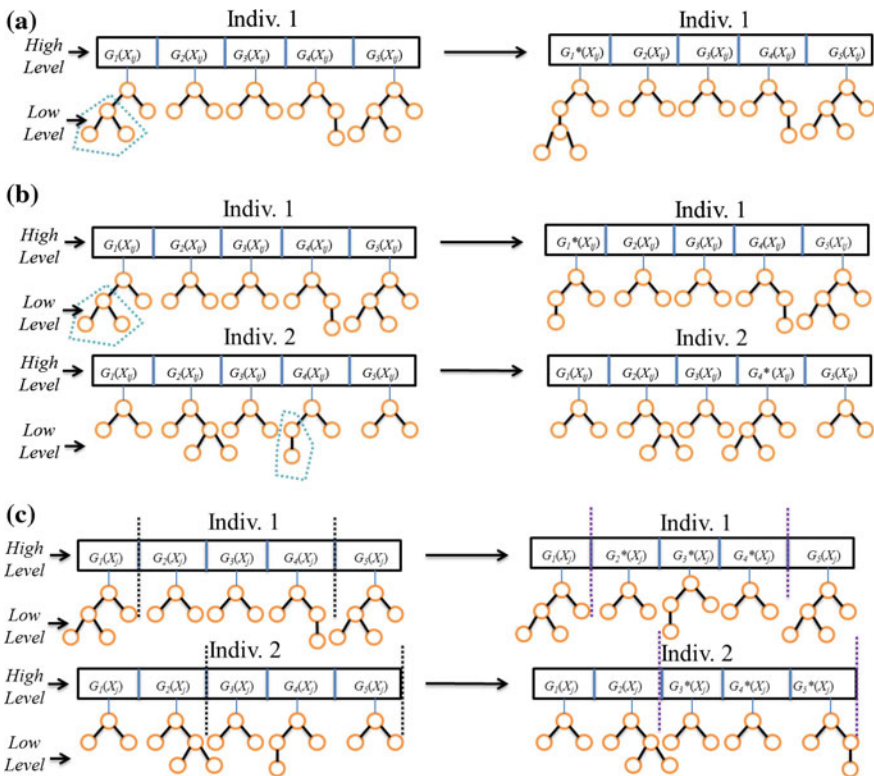


Fig. 2 Application example of MGGP operators: **a** mutation; **b** low level crossover; and **c** high level crossover

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 2a presents a multi-gene individual with five equations ($D = 5$) accomplishing a mutation, while Fig. 2b 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 Fig. 2c. 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. The high level crossover rate, similarly to other genetic operators rates, needs to be adjusted.

3 GPFIS-Regression Model

GPFIS-Regression is a typical Pittsburgh-type GFS [19]. 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 (Premises Splitting) and (iii) aggregation of each activated fuzzy rule (Aggregation). Finally, Defuzzification and Evaluation are performed.

3.1 Fuzzification

In regression problems, the main information for predicting the behavior of an output $y_i \in Y$ ($i = 1, \dots, n$) consists of its J attributes or features $x_{ij} \in X_j$ ($j = 1, \dots, J$). A total of L fuzzy sets are associated to each j th feature and are given by $A_{lj} = \{(x_{ij}, \mu_{A_{lj}}(x_{ij})) | x_{ij} \in X_j\}$, where $\mu_{A_{lj}} : X_j \rightarrow [0, 1]$ is a membership function that assigns to each observation x_{ij} a membership degree $\mu_{A_{lj}}(x_{ij})$ to a fuzzy set A_{lj} . Similarly, for Y (output variable), K fuzzy sets B_k ($k = 1, \dots, K$) are associated.

Three aspects are taken into account when defining membership functions: (i) form (triangular, trapezoidal, etc.); (ii) support set of $\mu_{A_{lj}}(x_{ij})$; (iii) an appropriate linguistic term, qualifying the subspace constituted by $\mu_{A_{lj}}(x_{ij})$ with a context-driven

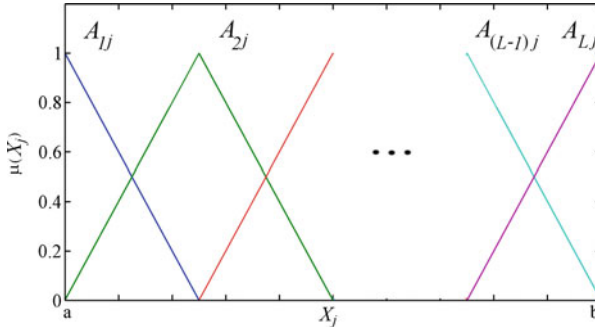


Fig. 3 Membership functions for $x_{ij} \in X_j$ variables. For Y read A_{ij} as B_k

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 [9, 19, 22] to define membership functions as shown in Fig. 3.

3.2 Fuzzy Inference

3.2.1 Formulation

A fuzzy rule premise is commonly defined by:

“If X_1 is A_{11} and ... and X_j is A_{1j} and ... and X_j is X_{1j} ”

or, in mathematical terms:

$$\mu_{A_d}(x_{i1}, \dots, x_{ij}) = \mu_{A_d}(\mathbf{x}_i) = \mu_{A_{i1}}(x_{i1}) * \dots * \mu_{A_{ij}}(x_{ij}) \tag{1}$$

where $\mu_{A_d}(x_{i1}, \dots, x_{ij}) = \mu_{A_d}(\mathbf{x}_i)$ is the joint membership degree of the i th pattern $\mathbf{x}_i = [x_{i1}, \dots, x_{ij}]$ with respect to the d th premise ($d = 1, \dots, D$), computed by using a t-norm $*$. A premise can be elaborated by using t-norms, t-conorms, linguistic hedges and negation operators to combine the $\mu_{A_{ij}}(x_{ij})$. As a consequence, the number of possible combinations grows as the number of variables, operators and fuzzy sets increase. Therefore, GPFIS-Regress employs MGGP to search for the most promising combinations, i.e., fuzzy rule premises. Figure 4 exemplifies a typical solution provided by MGGP.

For example, premise 1 represents: $\mu_{A_1}(\mathbf{x}_i) = \mu_{A_{21}}(x_{i1}) * \mu_{A_{32}}(x_{i2})$ and, in linguistic terms, “If X_1 is A_{21} and X_2 is A_{32} ”. Let $\mu_{A_d}(\mathbf{x}_i)$ be the d th premise codified in the d th tree of an MGGP individual. Table 1 presents the components used for reaching the solutions shown in Fig. 4.

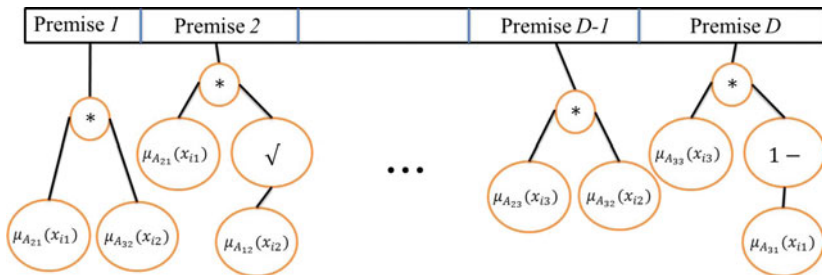


Fig. 4 Example of fuzzy rule premises codified in an MGGP individual

Table 1 Input fuzzy sets and operators to generate solutions in Fig. 4

Input fuzzy sets (Terminals set)	Fuzzy operators set (Functions set)
$\mu_{A_{11}}(y_{t-1}), \dots, \mu_{A_{L1}}(y_{t-1}), \dots, \mu_{A_{lp}}(y_{t-p}), \dots, \mu_{A_{lp}}(y_{t-p})$	t-norm (*), linguistic hedge (dilatation operator— $\sqrt{\quad}$) and classical negation operator

In GPFIS-Regress, the set of $\mu_{A_{ij}}(x_{ij})$ 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 $\mu_{A_d}(\mathbf{x}_i)$. In order to fully develop a fuzzy rule base, it is necessary to define the consequent term best suited to each $\mu_{A_d}(\mathbf{x}_i)$.

3.2.2 Premises 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-Regress 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-Regress measures the compatibility between $\mu_{A_d}(\mathbf{x}_i)$ and the consequent terms. This also promotes reduction of the search space.

In this sense, the Similarity Degree (SD_k) between the $\mu_{A_d}(\mathbf{x}_i)$ and the consequent terms is employed:

$$SD_k = \min\left(1 - \frac{\sum_{i=1}^n |\mu_{A_d}(\mathbf{x}_i) - \mu_{B_k}(y_i)|}{n}, I_{\{0,1\}}\right) \in [0, 1] \quad (2)$$

where $\sum_{i=1}^n |\mu_{A_d}(\mathbf{x}_i) - \mu_{B_k}(y_i)|$ is the Manhattan distance between the d th premise and the k th consequent term, while $I_{\{0,1\}}$ is an indicator variable, which takes value 0

when $\mu_{A_d}(\mathbf{x}_i) = 0, \forall i$, and 1 otherwise. When $\mu_{A_d}(\mathbf{x}_i) = \mu_{B_k}(y_i)$ for all i , then $FCD_k = 1$, i.e., premise and consequent term are totally similar. A consequent term for $\mu_{A_d}(\mathbf{x}_i)$ is selected as the k th consequent which maximize SD_k . A premise with $SD_k = 0$, for all k , is not associated to any consequent term (and not considered as a fuzzy rule).

3.2.3 Aggregation

A premise associated to the k th consequent term (i.e. a fuzzy rule) is denoted by $\mu_{A_{d^{(k)}}}(\mathbf{x}_i)$, which, in linguistic terms, means: “If X_1 is A_{1l} , and ..., and X_j is A_{jl} , then Y is B_k ”. Therefore, the whole fuzzy rule base is given by $\mu_{A_{1^{(k)}}}(\mathbf{x}_i), \dots, \mu_{A_{D^{(k)}}}(\mathbf{x}_i), \forall k = 1, \dots, K$. A new pattern \mathbf{x}_i^* 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 ($d^{(k)} = 1^{(k)}, 2^{(k)} \dots, D^{(k)}$). Given an aggregation operator $g : [0, 1]^{D^{(k)}} \rightarrow [0, 1]$ (see [7, 10]), the predicted membership degree of \mathbf{x}_i^* to each k th consequent term— $\hat{\mu}_{B_k}(y_i^*)$ —is computed by:

$$\hat{\mu}_{B_1}(y_i^*) = g[\mu_{A_{1(1)}}(\mathbf{x}_i^*), \dots, \mu_{A_{D(1)}}(\mathbf{x}_i^*)] \quad (3)$$

$$\hat{\mu}_{B_2}(y_i^*) = g[\mu_{A_{1(2)}}(\mathbf{x}_i^*), \dots, \mu_{A_{D(2)}}(\mathbf{x}_i^*)] \quad (4)$$

...

$$\hat{\mu}_{B_K}(y_i^*) = g[\mu_{A_{1(K)}}(\mathbf{x}_i^*), \dots, \mu_{A_{D(K)}}(\mathbf{x}_i^*)] \quad (5)$$

There are many aggregation operators available (e.g., see [6, 10, 36]), the Maximum being the most widely used [29]. 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:

$$\begin{aligned} \min : & \sum_{i=1}^n (\hat{\mu}_{B_k}(y_i) - \sum_{d^{(k)}=1}^{D^{(k)}} w_{d^{(k)}} \mu_{A_{d^{(k)}}}(\mathbf{x}_i))^2 \\ \text{s.t.} : & \sum_{d^{(k)}=1}^{D^{(k)}} w_{d^{(k)}} = 1 \text{ and } w_{d^{(k)}} \geq 0 \end{aligned} \quad (6)$$

where $w_{d^{(k)}}$ is the weight or the influence degree of $\mu_{A_{d^{(k)}}}(\mathbf{x}_i)$ 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 [11, 33]. This aggregation procedure is called Weighted Average by Restricted Least Squares (WARLS).

3.3 Defuzzification

Proposition 1 Consider $y_i \in Y$, with $a \leq y_i \leq b$ where $a, b \in \mathbb{R}$, and, associated to Y , K triangular membership functions, normal, 2-overlapped¹ and strongly partitioned (identical to Fig. 3). Then y_i can be rewritten as:

$$y_i = c_1\mu_{B_1}(y_i) + c_2\mu_{B_2}(y_i) + \dots + c_J\mu_{B_K}(y_i) \quad (7)$$

where c_1, \dots, c_K is the “center”— $\mu_{B_k}(c_k) = 1$ —of each k th membership function.

The proof can be found in in [28]. 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 $\mu_{B_k}(y_i)$ is known, then y_i is also known.
2. If only a prediction $\hat{\mu}_{B_k}(y_i)$ of $\mu_{B_k}(y_i)$ is known, such that $\sup_{y_i} |\mu_{B_k}(y_i) - \hat{\mu}_{B_k}(y_i)| \leq \varepsilon$, when $\varepsilon \rightarrow 0$ the defuzzification output \hat{y}_i that approximates y_i is given by:

$$\hat{y}_i = c_1\hat{\mu}_{B_1}(y_i) + c_2\hat{\mu}_{B_2}(y_i) + \dots + c_K\hat{\mu}_{B_K}(y_i) \quad (8)$$

When $\hat{\mu}_{B_k}(y_i) \approx \mu_{B_k}(y_i)$ is not verified, the Mean of Maximum or the Center of Gravity [32] defuzzification methods may lead to a better performance. However, due to the widespread use of strongly partitioned fuzzy sets in the experiments with GPFIS-Regress, a normalized version of the Height Method (8) has been employed:

$$\hat{y}_i = \frac{c_1\hat{\mu}_{B_1}(y_i) + \dots + c_K\hat{\mu}_{B_K}(y_i)}{\hat{\mu}_{B_1}(y_i) + \dots + \hat{\mu}_{B_K}(y_i)} \quad (9)$$

It is now possible to evaluate an individual of GPFIS-Regress by using \hat{y}_i .

3.4 Evaluation

The Evaluation procedure in GPFIS-Regress 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 regression problems is the Mean Squared Error (*MSE*):

$$MSE = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{2 * n} \quad (10)$$

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

The best individual in the population is the solution which minimizes (10). GPFIS-Regress tries to reduce the complexity of the rule base by employing a simple heuristic: Lexicographic Parsimony Pressure [25]. 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 ($\mu_{A_d}(\mathbf{x}_i)$), and, therefore, a small fuzzy rule set. After evaluation, a set of individuals is selected (through a tournament procedure) and recombined. This process is repeated until a stopping criteria is met. When this occurs, the final population is returned.

4 Case Studies

4.1 Experiments Description

Among the SFGs designed for solving regression problems, the Fast and Scalable Multi-Objective Genetic Fuzzy System (FS-MOGFS) [1] has been used in the experiments. In contrast to other works [2, 5, 8, 26, 31], FS-MOGFS has been chosen because:

1. it makes use of 17 regression datasets, where five of them are highly scalable and high-dimensional;
2. it presents a comparison between three different GFSs;
3. it describes in detail the parameters used for each model and the number of evaluations performed. Furthermore, the results show accuracy (training and test sets) and rule base compactness (average number of rules and of antecedents elements per rule).

In its basic version, FS-MOGFS consists of:

- Each chromosome (C) has two parts ($C = C_1 \cup C_2$): C_1 represents the number of triangular and uniformly distributed membership functions and $C_2 = [\alpha_1, \alpha_2, \dots, \alpha_j]$, where each α_j is a degree of displacement of the j th variable [2]. To obtain the best possible values for C , the model incorporates a Multi-Objective Genetic Algorithm (MOGA) based on SPEA2 [1]. The two objectives are: minimize the Mean Squared Error and the number of rules.
- In order to build the complete knowledge base (rules and membership functions), rule extraction via Wang & Mendel's algorithm is performed for each chromosome [35]. The Mamdani-type SIF employs the minimum for t-norm and implication, and center of gravity for defuzzification.

Extensions of FS-MOGFS have resulted in two other models: (i) **FS-MOGFS^e**—identical to FS-MOGFS, but with fast error computation by leaving aside a portion of the database; (ii) **FS-MOGFS+TUN**: similar to the previous one, but with fine tuning of membership functions parameters [16]. This model provided the best results

Table 2 Databases considered in experiments

Database	Acronym	J	n	Database	Acronym	J	n
Electrical maintenance	ELE	4	1056	Mortgage	MOR	15	1049
Auto MPG6	MPG6	5	398	Treasury	TRE	15	1049
Auto MPG8	MPG8	7	398	Baseball	BAS	16	337
Analcat	ANA	7	4052	MV artificial domain	MV	10	40768
Abalone	ABA	8	4177	Elevators	ELV	18	16559
Stock	STP	9	950	Computer-activity	CA	21	8192
Weather izmir	WIZ	9	1461	Ailerons	AIL	40	13750
Weather ankara	WAN	9	1609	The insurance company	TIC	85	9822
Forest fires	FOR	12	51				

and was therefore used for comparison with GPFIS-Regress. Databases shown in Table 2 [1] have been considered in case studies.

Five of the 17 databases are of high dimensionality: ELV, AIL, MV, CA e TIC; they have been obtained from the KEEL repository [1]. Similarly to the procedure adopted in Alcalá et al. [1], 100,000 evaluations (population size = 100 and number of generations = 1000) have been carried out in each execution. The remaining parameters are shown in Table 3. With six repeats of 5-fold cross-validation, GPFIS-Regress was executed 30 times. The metrics shown for each database are the average for the 30 trained models. The Mean Squared Error has been used as the fitness function [1].

It should be noted that preliminary tests considered three, five and seven fuzzy sets. As the results did not show any relevant difference as far as accuracy was concerned, five strongly partitioned fuzzy sets (Fig. 3) have been used throughout the experiments, as stated in Table 3.

In addition to FS-MOGFS+TUN, three other SFGs were used for comparison:

- GR-MF [13]: employs an evolutionary algorithm to define granularity and membership functions parameters of a Mamdani-type SIF. The Wang & Mendel method [35] is used for rule generation.

Table 3 GPFIS-Regress main configuration

Parameter	Value
Population size	100
Number of generations	1000
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 Fig. 3
Fuzzy operators	Product, classical negation and square-root
Partitioning method	SD
Aggregation operator	WARLS
Defuzzification	Height method

- GA-WM [12]: a GA is used for synthesizing granularity and the support of triangular membership functions, as well as for defining the universe of discourse. The rule base is also obtained through Wang & Mendel’s algorithm.
- GLD-WM [2]: similar to FS-MOGFS+TUN with respect to granularity and membership functions displacement. Wang & Mendel’s algorithm is used for rule generation. Final tuning of membership functions is not performed.

Statistical analysis have followed recommendations from [1, 14] and have been performed in the KEEL software [3], with a significance level of 0,1 ($\alpha = 0.10$).

4.2 Results and Discussion

Table 4 shows the results obtained with GPFIS-Regress and their counterpart GFSs for each database in terms of MSE, average number of rules and of antecedent elements per rule. Results for models other than GPFIS-Regress have been taken from [1]. In general GPFIS-Regress has provided better results in 58 % of cases, followed by FS-MOGFS+TUN with 23 %. GLD-WM has performed better for one single database; the remaining SFGs performed below those three. In high-dimensional problems, GPFIS-Regress as attained better results for three of the five databases.

Table 5 presents results for the Friedman test and Holm method for low-dimensional databases, given a significance level of 10 % [1]. As GPFIS-Regress presented the lowest rank (1.5417), it was chosen as the reference model. It can be observed that GPFIS-Regress achieved higher accuracy than GR-MF, GA-WM and

Table 4 Results for GPFFIS-Regress and other GFFSs in terms of EQM

Data base	GR-MF			GA-WM			GLD-WM			FS-MOGFS+TUN			GPFFIS-regress		
	R/A	Training	Test	R/A	Training	Test	R/A	Training	Test	R/A	Training	Test	R/A	Training	Test
ELLE	97/4	16645	18637	47/4	17230	18977	33/4	11483	13384	9/2	8803	9842	16/3	14695	16818
σ		2319	3386		2501	3195		1085	1978		739	1391		815	493
MPG6	243/5	1.423	28.933	186/5	1.879	8.824	82/5	2.294	4.387	22/3	2.778	4.548	18/3	2.890	4.003
σ		0.073	8.633		0.235	6.079		0.249	0.899		0.220	1.047		0.066	0.336
MPG8	262/7	1.356	49.36	214/7	1.563	15.216	135/7	1.709	4.782	24/3	2.725	4.381	18/3	2.780	4.087
σ		0.104	16.2		0.183	9.13		0.170	1.445		0.294	0.909		0.091	0.116
ANA	148/7	0.005	0.017	150/7	0.003	0.008	92/7	0.006	0.008	17/3	0.003	0.003	8/3	0.002	0.003
σ		0.001	0.008		0.001	0.005		0.001	0.004		0.000	0.001		0.000	0.000
ABA	498/8	2.358	2.885	143/8	2.433	2.549	31/8	2.487	2.545	10/3	2.393	2.454	16/3	2.370	2.425
σ		0.052	0.263		0.052	0.163		0.078	0.170		0.092	0.163		0.068	0.154
STP	343/9	0.4	1.543	344/9	0.389	2.192	217/9	0.299	0.435	25/3	0.724	0.892	17/3	1.143	1.283
σ		0.019	2.484		0.017	3.168		0.025	0.067		0.112	0.154		0.173	0.256
WIZ	331/9	1.176	9.602	218/9	1.233	3.529	107/9	0.926	1.150	15/3	0.867	1.011	17/3	0.811	0.878
σ		0.077	8.879		0.065	4.023		0.041	0.123		0.040	0.177		0.046	0.040
WAN	397/9	1.406	7.381	279/9	1.522	2.82	133/9	1.111	2.075	11/2	1.313	1.581	15/2	1.307	1.367
σ		0.067	5.404		0.065	2.825		0.077	1.407		0.174	0.580		0.079	0.092
FOR	396/12	113	3300	395/12	47	3693	377/12	49	3847	33/3	1593	2406	8/3	1476	2446
σ		17	2207		24	2787		18	2714		570	2161		868	2456
MOR	209/15	0.03	0.176	160/15	0.02	0.093	78/15	0.016	0.022	9/3	0.015	0.018	15/3	0.013	0.015
σ		0.002	0.28		0.003	0.147		0.002	0.005		0.004	0.012		0.001	0.002

(continued)

Table 4 (continued)

Data base	GR-MF		GA-WM		GLD-WM		FS-MOGFS+TUN		GPFIS-regress						
TRE	189/15	0.066	0.144	136/15	0.045	0.064	70/15	0.033	0.045	11/3	0.030	0.040	15/3	0.031	0.037
σ		0.011	0.191		0.007	0.046		0.005	0.015		0.004	0.012		0.002	0.002
BAS	262/16	0.255	12.439	262/16	0.202	11.706	244/16	0.138	3.610	21/6	1.305	2.699	16/3	1.469	3.037
		0.02	2.177		0.031	2.562		0.014	0.621		0.172	0.620		0.067	0.246
MV	-	-	-	-	-	-	-	-	-	16/3	0.159	0.160	10/3	2.607	2,599
σ	-	-	-	-	-	-	-	-	-		0.031	0.032		1,499	1,490
ELV	-	-	-	-	-	-	-	-	-	8/3	0.900	0.900	12/3	0.875	0.886
σ	-	-	-	-	-	-	-	-	-		0.200	0.200		0.106	0.108
CA	-	-	-	-	-	-	-	-	-	15/5	4.763	5.063	15/4	4.885	5.060
σ	-	-	-	-	-	-	-	-	-		0.404	0.760		0.729	0.739
AIL	-	-	-	-	-	-	-	-	-	20/4	1.864	1.905	17/3	1.829	1.858
σ	-	-	-	-	-	-	-	-	-		0.221	0.233		0.003	0.003
TIC	-	-	-	-	-	-	-	-	-	25/7	0.026	0.027	11/4	0.026	0.027
σ	-	-	-	-	-	-	-	-	-		0.000	0.002		0.000	0.001

σ = standard deviation of EQM. Results for Training/Test must be multiplied by 10^5 , 10^{-5} and 10^{-8} for BAS, ELV and AIL respectively. R/A—Averages of Rules and Antecedent Elements per Rule

Table 5 Results for Friedman test and Holm method

i	Model	Rank	
4	GR-MF	4.6667	
3	GA-WM	4.1250	
2	GLD-WM	2.8750	
1	FS-MOGFS+TUN	1.7917	
0	GPFIS	1.5417	
Test	p-value		
Friedman	<0.0001		
Method	$z = (R_0 - R_i)/SE$	p-value	Holm
GR-MF	4.8412	<0.0001	0.0250
GA-WM	4.0020	<0.0001	0.0333
GLD-WM	2.0655	0.0388	0.0500
FS-MOGFS+TUN	0.3872	0.6985	0.1000

GLD-WM have (p-value < 0.05). This has not been verified for GPFIS-Regress and FS-MOGFS+TUN (p-value > 0.10).

If GPFIS-Regress and FS-MOGFS+TUN are singled out for comparison, it can be observed that the former has achieved better results for 10 of the 17 databases, with two ties. The signal test has shown that the differences in results were not significant ($S = 10$, p-value = 0.3018). This may be due to the ties and to the small number of databases considered. As for rule base complexity, it can be noted that GPFIS-Regress obtained the most compact one in 53 % of cases.

As far as interpretability and implementation are concerned, GPFIS-Regress has an advantage over FS-MOGFS+TUN in aspects such as: (i) makes no change to membership functions parameters; (ii) employs a MHG with a single objective, while FS-MOGFS+TUN does a multi-objective search.

5 Conclusion

This work has presented a novel Genetic Fuzzy System for solving regression problems, called GPFIS-Regress, which makes use of Multi-Gene Genetic Programming and a novel way to formulate the Fuzzy Reasoning Method (Formulation-Splitting-Aggregation). GPFIS-Regress has been compared to four other Genetic Fuzzy Systems for 17 datasets of low and high dimensionality. Results have shown the potentialities of the proposed approach with respect to the state-of-art in the Genetic Fuzzy Systems area.

Further developments and experiments shall include: (i) evaluation of other t-norm, negation and linguistic hedges operators, as well as the use of t-conorms in rules premises; (ii) new premises splitting methods (through other similarity

measures) and application of the Restricted Least Squares procedure with some adaptation to associate a more suitable consequent term to a given premise; *(iii)* evaluation of other aggregation operators, such as nonlinear ones (weighted geometric mean, etc.); this may provide better results mostly in terms of accuracy. A fine-tuning of membership functions and Genetic Programming set-up parameters shall also be considered.

References

1. Alcalá, R., Gacto, M.J., Herrera, F.: A fast and scalable multiobjective genetic fuzzy system for linguistic fuzzy modeling in high-dimensional regression problems. *Fuzzy Syst. IEEE Trans.* **19**(4), 666–681 (2011)
2. Alcalá, R., Alcalá-Fdez, J., Herrera, F., Otero, J.: Genetic learning of accurate and compact fuzzy rule based systems based on the 2-tuples linguistic representation. *Int. J. Approx. Reason.* **44**(1), 45–64 (2007)
3. Alcalá-Fdez, J., Fernandez, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *J. Mult.-Valued Logic Soft Comput.* **17**(2–3), 255–287 (2011)
4. Angelov, P., Buswell, R.: Identification of evolving fuzzy rule-based models. *Fuzzy Syst. IEEE Trans.* **10**(5), 667–677 (2002)
5. Antonelli, M., Ducange, P., Lazzarini, B., Marcelloni, F.: Learning concurrently partition granularities and rule bases of mamdani fuzzy systems in a multi-objective evolutionary framework. *Int. J. Approx. Reason.* **50**(7), 1066–1080 (2009)
6. Beliakov, G., Warren, J.: Appropriate choice of aggregation operators in fuzzy decision support systems. *IEEE Trans. Fuzzy Syst.* **9**(6), 773–784 (2001)
7. Beliakov, G., Pradera, A., Calvo, T.: *Aggregation Functions: A Guide for Practitioners*. Springer Publishing Company, Heidelberg (2008)
8. Benítez, A.D., Casillas, J.: Multi-objective genetic learning of serial hierarchical fuzzy systems for large-scale problems. *Soft Comput.* **17**(1), 165–194 (2013)
9. Berlanga, F.J., Rivera, A.J., del Jesus, M.J., Herrera, F.: Gp-coach: genetic programming-based learning of compact and accurate fuzzy rule-based classification systems for high-dimensional problems. *Inf. Sci.* **180**(8), 1183–1200 (2010)
10. Calvo, T., Kolesárová, A., Komorníková, M., Mesiar, R.: Aggregation operators: properties, classes and construction methods. In: Calvo, T., Mayor, G., Mesiar, R. (eds.) *Aggregation Operators, Studies in Fuzziness and Soft Computing*, vol. 97, pp. 3–104. Physica-Verlag HD (2002)
11. Coleman, T.F., Li, Y.: A reflective newton method for minimizing a quadratic function subject to bounds on some of the variables. *SIAM J. Optim.* **6**(4), 1040–1058 (1996)
12. Cordon, O., Herrera, F., Magdalena, L., Villar, P.: A genetic learning process for the scaling factors, granularity and contexts of the fuzzy rule-based system data base. *Inf. Sci.* **136**(1–4), 85–107 (2001)
13. Cordon, O., Herrera, F., Villar, P.: Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base. *IEEE Trans. Fuzzy Syst.* **9**(4), 667–674 (2001)
14. Derrac, J., García, S., Molina, D., Herrera, F.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **1**(1), 3–18 (2011)
15. Fattah, K.A.: A new approach calculate oil-gas ratio for gas condensate and volatile oil reservoirs using genetic programming. *Oil Gas Bus.* **1**, 311–323 (2012)
16. Gacto, M.J., Alcalá, R., Herrera, F.: Adaptation and application of multi-objective evolutionary algorithms for rule reduction and parameter tuning of fuzzy rule-based systems. *Soft Comput.* **13**(5), 419–436 (2008)

17. Gandomi, A.H., Alavi, A.H.: A new multi-gene genetic programming approach to nonlinear system modeling. *Neural Comput. Appl.* **21**(1), 171–187 (2012)
18. Haykin, S.: *Neural Netw. Learn. Mach.* Prentice-Hall, New York (2009)
19. Herrera, F.: Genetic fuzzy systems: taxonomy, current research trends and prospects. *Evol. Intell.* **1**(1), 27–46 (2008)
20. Herrera, F., Martínez, L.: A 2-tuple fuzzy linguistic representation model for computing with words. *IEEE Trans. Fuzzy Syst.* **8**(6), 746–752 (2000)
21. Hinchliffe, M., Hiden, H., McKay, B., Willis, M., Tham, M., Barton, G.: Modelling chemical process systems using a multi-gene. In: *Late Breaking Papers at the Genetic Programming*, pp. 56–65, Stanford University, Stanford, June 1996
22. Ishibuchi, H., Yamane, M., Nojima, Y.: Rule weight update in parallel distributed fuzzy genetics-based machine learning with data rotation. In: *IEEE International Conference on Fuzzy Systems, 2013. FUZZ-IEEE 2013*, pp. 1–8. IEEE (2013)
23. Koza, J.R.: *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Massachusetts (1992)
24. Kutner, M.H., Nachtsheim, C.J., Neter, J., Li, W.: *Applied Linear Statistical Models*, 8th edn. McGraw-Hill, New York (2005)
25. Luke, S., Panait, L.: Lexicographic parsimony pressure. In: Langdon, W.B., Cantú-Paz, E., Mathias, K., Roy, R., Davis, D., Poli, R., Balakrishnan, K., Honavar, V., Rudolph, G., Wegener, J., Bull, L., Potter, M.A., Schultz, A.C., Miller, J.F., Burke, E., Jonoska, N. (eds) *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 829–836. Morgan Kaufmann Publishers, New York (2002)
26. Márquez, A.A., Márquez, F.A., Roldán, A.M., Peregrín, A.: An efficient adaptive fuzzy inference system for complex and high dimensional regression problems in linguistic fuzzy modelling. *Knowl.-Based Syst.* **54**, 42–52 (2013)
27. McCullagh, P., Nelder, J.A.: *Generalized Linear Models*. Chapman Hall, London (1989)
28. Pedrycz, W.: *Granular Computing: Analysis and Design of Intelligent Systems*. CRC Press, Boca Raton (2013)
29. Pedrycz, W., Gomide, F.: *An Introduction to Fuzzy Sets: Analysis and Design*. MIT Press, Massachusetts (1998)
30. Poli, R., Langdon, W.B., McPhee, N.F.: *A Field Guide to Genetic Programming*. Lulu.com, Rayleigh (2008)
31. Pulkkinen, P., Koivisto, H.: A dynamically constrained multiobjective genetic fuzzy system for regression problems. *IEEE Trans. Fuzzy Syst.* **18**(1), 161–177 (2010)
32. Roychowdhury, S., Pedrycz, W.: A survey of defuzzification strategies. *Int. J. Intell. Syst.* **16**(6), 679–695 (2001)
33. Schölkopf, B., Smola, A.J.: *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Massachusetts (2001)
34. Searson, D., Willis, M., Montague, G.: Coevolution of nonlinear pls model components. *J. Chemom.* **21**(12), 592–603 (2007)
35. Wang, L.X., Mendel, J.M.: Generating fuzzy rules by learning from examples. *IEEE Trans. Syst. Man Cybern.* **22**(6), 1414–1427 (1992)
36. Yager, R.R., Kacprzyk, J.: *The Ordered Weighted Averaging Operators: Theory and Applications*. Kluwer, Norwell (1997)