A Fast and Scalable Multiobjective Genetic Fuzzy System for Linguistic Fuzzy Modeling in High-Dimensional Regression Problems

Rafael Alcalá, María José Gacto, and Francisco Herrera, *Member, IEEE*

***Abstract*—Linguistic fuzzy modeling in high-dimensional regression problems poses the challenge of exponential-rule explosion when the number of variables and/or instances becomes high. One way to address this problem is by determining the used variables, the linguistic partitioning and the rule set together, in order to only evolve very simple, but still accurate models. However, evolving these components together is a difficult task, which involves a complex search space. In this study, we propose an effective multiobjective evolutionary algorithm that, based on embedded genetic database (DB) learning (involved variables, granularities, and slight fuzzy-partition displacements), allows the fast learning of simple and quite-accurate linguistic models. Some efficient mechanisms have been designed to ensure a very fast, but not premature, convergence in problems with a high number of variables. Further, since additional problems could arise for datasets with a large number of instances, we also propose a general mechanism for the estimation of the model error when using evolutionary algorithms, by only considering a reduced subset of the examples. By doing so, we can also apply a fast postprocessing stage for further refining the learned solutions. We tested our approach on 17 real-world datasets with different numbers of variables and instances. Three well-known methods based on embedded genetic DB learning have been executed as references. We compared the different approaches by applying nonparametric statistical tests for multiple comparisons. The results confirm the effectiveness of the proposed method not only in terms of scalability but in terms of the simplicity and generalizability of the obtained models as well.**

***Index Terms*—Embedded genetic database learning, high-dimensional regression problems, linguistic fuzzy modeling, multiobjective genetic fuzzy systems, scalability.**

I. Introduction

LINGUISTIC fuzzy modeling in high-dimensional and large-scale regression datasets is a challenging topic since conventional linguistic fuzzy-rule-based systems (FRBSs) suffer from exponential-rule explosion when the number of variables and/or data examples becomes high [1], [2]. Another problem when we deal with high-dimensional datasets is the analysis of algorithm scalability on big databases (DBs), emphasizing the training time and the convergence toward compact and interpretable models [3]. This way, we can distinguish two kinds of problems: high dimensionality when a large number of variables have to be considered, and scalability in datasets with a large amount of data.

A good way to address both problems is by searching for a good and simple global structure within the same process, in order to consider the relationships among the different components defining the knowledge base (KB) of the obtained linguistic models, i.e., by learning the main components of the KB, a DB containing the definitions of the linguistic fuzzy partitions and a rule base (RB) containing the associated set of rules, together. Since this method involves using different coding schemes to represent each solution, evolutionary algorithms, particularly genetic algorithms (GAs), are useful for this task. These kinds of global-search techniques have been successfully applied to learn fuzzy systems in recent years, thus giving rise to the so-called genetic fuzzy systems (GFSs) [3]–[5]. Furthermore, the application of multiobjective evolutionary algorithms (MOEAs) to the derivation of compact linguistic FRBSs is a prolific framework in which we can find several interesting and recent works. Some MOEAs were proposed as postprocessing techniques [6]–[13], while others were proposed as learning techniques [11], [14]–[18].

However, this method involves a lot of components/parameters that should be determined together: selection of important variables, determination of a good number of linguistic terms or granularities per variable, parametric definition of the membership functions (MFs) and associated set of rules. Since it involves using different coding schemes to represent a complete solution and, therefore, a very complex search space, this is a difficult task. In fact, the balance among problem size, algorithm scalability, and solution quality is an important topic for GFSs that is worth studying in depth [3], which has not been directly taken into account in the mentioned evolutionary approaches devoted to linguistic fuzzy modeling.

An efficient way to obtain the entire KB of an FRBS is to obtain the DB and the RB within the same process but separately, as based on embedded genetic DB learning [19]–[24]. This is an evolutionary process that learns the DB and wraps a simple method to derive a set of rules for each DB definition. This enables the most-adequate context [20] for each fuzzy partition to be learned, which strongly affects the final model complexity.

Manuscript received February 16, 2010; revised July 23, 2010, November 12, 2010, and February 14, 2011; accepted February 15, 2011. Date of publication March 24, 2011; date of current version August 8, 2011. This work was supported by the Spanish Ministry of Education and Science under Grant TIN2008-06681-C06-01.

R. Alcalá and F. Herrera are with the Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain (e-mail: alcala@decsai.ugr.es; herrera@decsai.ugr.es).

M. J. Gacto is with the Department of Computer Science, University of Jaén, 23071 Jaén, Spain (e-mail: mjgacto@ugr.es).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TFUZZ.2011.2131657

However, this approach cannot solve the following contradictory requirements.

1) The obtained linguistic models should be not only *simple* and *transparent*, but *competitive* as well, in terms of the generalization error.
2) The evolutionary learning algorithm should be not only *effective*, but *scalable* as well, in terms of the time and memory consumed, in order to be useful for a wide range of high-dimensional or large-scale problems.

In this study, we propose a convenient reduction of the search space for the embedded genetic DB learning (i.e., variable selection, granularities, and MF parameters) and an effective and efficient MOEA as a tool that makes use of some specific mechanisms in order to ensure a fast convergence. To reduce the search space [25], we propose the performance of a slight *lateral displacement of fuzzy partitions* by applying a common displacement parameter to all the MFs at each linguistic variable. This allows a simple prescreening of promising granularities, which avoids the derivation of very specific systems presenting overfitting, and preserves equidistributed strong fuzzy partitions. In addition, the proposed MOEA includes such concepts as incest prevention and restarting in order to improve the algorithm convergence [26], together with some mechanisms to step up the learning process, such as a rule-cropping criterion in the RB-generation process.

The proposed method is able to handle dimensionality; however, it does not directly solve scalability with respect to the number of available data in the dataset. To deal with this, we also propose a mechanism to avoid using a big percentage of the examples for error computation, thereby estimating it from a reduced subset of the examples. By doing so, we can also apply a postprocessing stage to further refine the learned solutions. We have applied a previous MOEA [27], [28], namely, SPEA2$_{E/E}$, including this new error-estimation procedure for fine tuning of the MFs and rule selection, which will help to significantly improve the performance of the simple global structure (which is initially based on strong fuzzy partitions), while the complexity is decreased.

We tested our approach on 17 real-world problems with a number of variables ranging from four to 85, and a number of samples ranging from 337 to 40 768. When it was possible, depending on the dimensionality, we executed three well-known accuracy-driven single-objective methods based on embedded genetic DB learning in order to have some good performance references. To assess the results obtained by the different algorithms, we have applied nonparametric statistical tests [29]–[32] for multiple comparisons, taking into consideration for the MOEA the average of the most-accurate solution from each Pareto front. The results obtained demonstrate the effectiveness of the proposed method, particularly not only in terms of scalability, but also in terms of simplicity and the generalizability of the obtained models.

This contribution is arranged as follows. Section II proposes the lateral displacement of fuzzy partitions. In Section III, we present an effective MOEA to learn FRBSs for high-dimensional problems. Section IV proposes the new method for fast error computation and its application to the proposed algorithm and to a known algorithm for postprocessing, i.e., SPEA2$_{E/E}$. Section V shows the experimental study on the proposed method and describes a web page associated with the paper (i.e., `http://sci2s.ugr.es/FS-MOGFS/`) that contains complementary material to this study. Finally, Section VI draws some conclusions.

II. Proposal for the Lateral Displacement of Linguistic Fuzzy Partitions

In [25], a new model for tuning of FRBSs was proposed, considering the linguistic 2-tuple representation scheme introduced in [33], which allows the lateral displacement of the support of a label. The main achievement is that, since the three parameters usually considered per label [4], [34]–[40] are reduced to only one symbolic translation parameter, this proposal decreases the learning problem complexity, thus facilitating the derivation of optimal models [14], [25], [41]. In any event, an FRBS based on linguistic two tuples could be represented as a classical Mamdani FRBS [42], [43]. For a more detailed description of this tuning approach, see [25] or the web page associated with the paper (i.e., `http://sci2s.ugr.es/FS-MOGFS/`).

The lateral tuning of MFs allows a good adaptation for each MF comprising the DB. However, our main aim in this study is to learn a good, simple, and general KB in a fast way. Learning all the components of the KB together represents a huge search space when high-dimensional problems are considered. To perform a good adaptation for each individual MF while learning the system structure could lead to very complex systems, since it is difficult to obtain the best parameters for each concrete system structure. Once relatively good parameters are obtained for a system structure, convergence starts in this zone, and it is difficult to explore other good configurations (with similar accuracy) that could represent more simple and interesting systems.

To solve this problem, we propose the application of a single lateral displacement of linguistic fuzzy partitions by applying a common α displacement parameter to all the MFs at each linguistic variable, i.e., all the MFs are uniformly displaced depending on the displacement parameter associated with each linguistic fuzzy partition. In order to avoid very specific parameters and to preserve the original meanings of the MFs as much as possible, we propose the use of a short displacement interval, $[-0.1, 0.1)$ in our case, as the range to express the relative shifts associated with the labels. This way, we can represent the translation of a linguistic partition S by the 2-tuple notation as

$$(S, \alpha),\ \alpha \in [-0.1, 0.1) \Rightarrow (s_i, \alpha) \qquad \forall s_i \in S.$$

Fig. 1 shows the lateral displacement of a linguistic partition S for a concrete α value. Some interesting characteristics of this approach are as follows.

1) The search space is reduced providing a fast convergence. This makes it easier to explore different granularities that can represent promising linguistic partitions.
2) The constrained variation interval avoids a fine adaptation of the MFs, thereby allowing only a simple prescreening

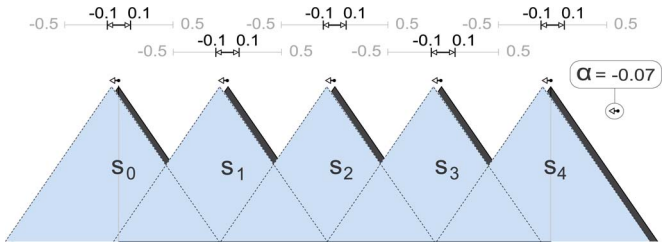

Fig. 1. Lateral displacement in [–0.1, 0.1] of the whole linguistic partition $S = \{s_0, s_1, s_2, s_3, s_4\}$.

of promising granularities, which avoids the derivation of very specific systems presenting overfitting.

All these properties facilitate a fast derivation of promising models based on uniformly distributed strong fuzzy partitions. Once they are obtained, a fine tuning [6], [7], [25], [37]–[39] (i.e., postprocessing) could be applied easily depending on user preference. We do not consider this possibility in this contribution as we are focused only on the learning stage.

III. Fast and Scalable Multiobjective Genetic Fuzzy System for Embedded Genetic Database Learning

An alternative to learn an entire KB is iterative rule learning (IRL). However, IRL approaches usually obtain models with too many rules involving all the system variables. For this reason, they are usually devoted to obtaining approximate models, which mainly focused on accuracy, in problems with a reasonable number of variables. An example can be found in [44], where a three-stage learning process is used to obtain a large set of accurate approximate Takagi–Sugeno–Kang (TSK) fuzzy rules. A methodology with a similar philosophy [45] consists of the use of a clustering method for first generating a set of initial TSK local rules in order to subsequently reduce complexity without affecting accuracy too much. Both approaches allow TSK models (i.e., approximate FRBSs) to be obtained, considering all the variables in each rule and presenting highly accurate results in problems with a reasonable number of variables.

In the case of linguistic FRBSs (which need the definition of a permanent appropriate grid), an efficient way to learn the entire KB consists of obtaining the DB and RB within the same process but separately, which is based on embedded genetic DB learning [19]–[24], [46]. This method allows us to learn the most-adequate context [20], [46] for each fuzzy partition, which is necessary in different contextual situations (different applications).

Even though different optimization techniques could be considered for the embedded learning of the DB parameters, in this study, we consider an MOEA for this task, which allows different coding schemes to be handled within the same process and to improve both system accuracy and simplicity (which are essential to handle high-dimensional problems). To this end, we can learn different parts of the DB together (i.e., number of labels and parameters), thus considering the relationships among them.

The learning scheme considered to obtain complete KBs comprises two main components: DB evolutionary learning and an

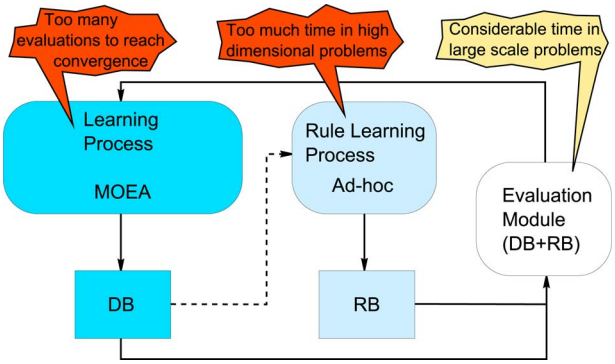

Fig. 2. Learning scheme of the KB.

RB *ad hoc* rule-learning process. In the following, an effective design of the learning process is first discussed and proposed later to present the specific fast MOEA as the most important part of the proposed technique.

A. Convergence and Scalability Discussion for the Embedded Algorithm Design

Some problems arise when high-dimensional datasets are considered (see Fig. 2). The two main problems are as follows:

1) *The large number of evaluations needed to reach convergence*: We solve this problem in two ways. By learning together the number of labels and single-partition displacement parameters for each linguistic variable instead of the three definition points for each MF (reduced search space); and by developing an advanced MOEA based on the well-known SPEA2, we ensure an effective tradeoff between exploration and exploitation. This specific MOEA is able to stop the process when convergence is reached. This way, we can ensure a fast but effective convergence in order to avoid unneeded evaluations.
2) *Too much time is required to generate the RB*: This problem is related to the previous one. Each evaluation requires generating an RB based on the coded DB. Even though a fast ad hoc rule-generation method will be used, this method can take a significant amount of time in high-dimensional problems. Due to the required number of evaluations, it poses a problem. We solve this problem by including a cropping criterion in the RB-generation method, thus avoiding the generation of excessively large RBs that expend too much time and make no sense in linguistic fuzzy modeling. Additionally, we enable the removal of unnecessary variables while evolving, thus leading to DBs that do not provoke an excessive number of rules when the RB-generation process is applied.

The two previous problems are directly related to the learning process. However, a third problem arises in large-scale problems (i.e., datasets with a large number of data) since each evaluation can take a considerable amount of time (see Fig. 2). This problem is greatly reduced by solving the two previous ones and is also related to data preprocessing [47] or parallel computation [48], [49]. We address this problem in Section IV, thus proposing a general scheme for fast error estimation.

1) Embedded Genetic Database Learning: Taking into account the previous discussion, the proposed algorithm comprises of the following two main components.

1) An effective MOEA based on SPEA2 [50] with two minimization objectives (i.e., system error and number of rules) in order to learn promising DBs. In order to improve its search ability and good convergence, this MOEA implements such concepts as incest prevention and restarting [26] as well. It allows us to define the following:
   a) the number of labels per variable, which determine the corresponding uniformly distributed strong linguistic partitions. We will enable the possibility of removing unnecessary variables by allowing granularity 1, which means that the corresponding variable is not considered in the final model;
   b) the lateral displacements for each linguistic partition.
2) A quick *ad hoc* data-driven method to learn an RB from each DB definition within the evolutionary process. The cooperative action of both components allows the whole definition of the KB (i.e., DB and RB) to be obtained. The simple Wang and Mendel algorithm [51] (WM) will be considered for this task by adding an RB cropping mechanism.

Due to the importance of the cropping mechanism for high-dimensional datasets, we first explain this approach devoted to shortening the time spent on evaluating nonsensical KBs. Then, the MOEA for evolving DBs that integrates this version of WM is explained in depth.

2) Cropping Mechanism for the Ad Hoc Wang and Mendel Algorithm: The WM process is based on the existence of a predefined DB and a set of input–output training data $E = \{e_1, \ldots, e_l, \ldots, e_m\}$ with $e_l = (x_1^l, \ldots, x_{N-1}^l, y^l)$, $l \in \{1, \ldots, m\}$, m being the dataset size, and $N-1$ being the number of input variables. Since, in high-dimensional problems, WM can take a long time to derive thousands of rules, a cropping criterion has been added to this method. This way, the RB is generated by means of the following steps, integrating the WM cropping mechanism as the last step of the process.

1) Initially, the RB is empty, and the data are randomly ordered (the data are reordered at each generation of the evolutionary algorithm).
2) For each example e_l in E, we do the following.
   a) Generate the rule with the labels best covering the example $(x_1^l, \ldots, x_{N-1}^l, y^l)$.
   b) Compute the covering degree of the complete rule (i.e., antecedent and consequent).
   c) If there is no rule with the same antecedent in the RB, add the obtained rule to the RB together with its covering degree.
      Otherwise, maintain the consequent and covering degree of the rule with the highest coverage.
   d) Stop the process if the RB reaches a limit of 50 rules, and mark the RB as incomplete.

In this contribution, we propose a maximum number of rules, i.e., 50, for the rule-cropping mechanism, which are based on some empirical trials. The smaller this number is, the faster the method is, and the simpler the solutions are. However, the precision of the models finally obtained is significantly affected with too small values. We detected this phenomenon in values clearly under 50, thereby making this number a good limit for a large variety of problems. Higher values or even those that do not use cropping do not obtain significantly more accurate solutions in terms of generalizability (see the web page associated with the paper for some examples demonstrating this at `http://sci2s.ugr.es/FS-MOGFS/`).

B. Proposed Multiobjective Evolutionary Algorithm

This section presents the proposed MOEA for the embedded genetic DB learning, namely, fast and scalable multiobjective GFS (FSMOGFS). In the following, the components needed to implement this algorithm are explained in depth, which include DB codification, objectives and incomplete RBs penalization, initial gene pool, crossover and mutation, incest prevention, restarting, and stopping condition.

1) Database Codification: A double-coding scheme (i.e., $C = C_1 + C_2$) to represent both parts, i.e., *granularity* and *translation parameters*, is considered.

1) *Number of labels (C_1):* This part is a vector of integer numbers with size N (with N representing the number of linguistic variables) in which the granularities of the different variables are coded
$$C_1 = (L^1, \ldots, L^N).$$
Each gene L^i represents the number of labels used by the ith variable and takes values in the set $\{2, \ldots, 7\}$. Additionally, in the case of input variables, it can take a value equal to 1 to determine that the corresponding variable is not used.
2) *Lateral displacements (C_2):* This part is a vector of real numbers with size N in which the displacements of the different variables are coded. This way, the C_2 part has the following structure (where each gene is the displacement value of the fuzzy partition of the corresponding linguistic variable and takes values from $[-0.1, 0.1]$)
$$C_2 = (\alpha^1, \ldots, \alpha^N).$$

2) Objectives and Incomplete Rule Bases Penalization: Once a complete KB is obtained, the following two objectives are minimized for this problem: the number of rules (i.e., simplicity) and the mean-squared error (i.e., accuracy):

$$\text{MSE} = \frac{1}{2 \cdot |E|} \sum_{l=1}^{|E|} (F(x^l) - y^l)^2$$

with $|E|$ being the dataset size, $F(x^l)$ being the output obtained from the FRBS decoded from a given chromosome when the lth example is considered, and y^l being the known desired output. The fuzzy-inference system considered to obtain $F(x^l)$ is the *center of gravity weighted by the matching* strategy as a defuzzification operator and the *minimum t-norm* as implication and conjunctive operators.

In order to obtain a complete KB from a given chromosome, we apply WM to the DB coded by this chromosome, considering a cropping mechanism. First, in order to decode this DB, equidistant strong fuzzy partitions are defined considering the granularity values in C_1. Second, the MFs of each variable are uniformly displaced to their new position considering the displacement values in C_2.

WM is applied to the obtained DB, but it stops if the RB reaches a maximum of 50 rules and marks the RB as incomplete in order to penalize its objective values.

1) In the case of the number of rules, we estimate the worst-possible value as the product of the number of labels of the input variables in the decoded DB (which is a pessimistic proportional estimation of the number of rules).
2) In the case of the MSE, it is multiplied by 2.0 (if an example is not covered by the incomplete RB, the middle of the output domain is given as the estimated output).

This way, these solutions are not a problem for the computational time, and they compete with each other at secondary Pareto fronts, which is useful for the detection of promising combinations of selected variables and granularities at the first stages of the algorithm (i.e., until appropriate combinations of variables and granularities that allow the derivation of RBs with a good number of rules arise, which will dominate those previous incomplete solutions).

3) Initial Gene Pool: The initial population will be comprising following two different subsets of individuals.

1) In the first subset, each chromosome has the same number of labels for all the system input variables. In order to provide diversity in the C_1 part, these solutions have been generated by considering all the possible combinations in the antecedent part, i.e., from two labels to seven labels in all the input variables (i.e., six combinations). For each of these combinations, all the possible combinations are generated in the consequent part (i.e., six combinations per each input combination). Additionally, for each of the previous combinations, two copies are included with different values in the C_2 part. The first one with random values in $[-0.1, 0.0]$ and the second one with random values in $[0.0, 0.1]$. Thus, a total of 72 (i.e., $6 * 6 * 2$) different individuals are generated. If there is no space for these solutions, they are included from the smallest granularities (which is the most-interesting combinations, in principle) to the highest possible ones.
2) In the second subset, we generate random solutions in order to completely fill the population (values in $\{2, \ldots, 7\}$ for C_1 and values in $[-0.1, 0.1]$ for C_2).

Finally, except in the cases of problems with less than three input variables, an input variable v is removed at random, i.e., $L^v = 1$, in the first individual. This action is repeated until no more than ten variables remain in this individual. If the problem has no more than ten variables, this action is not repeated, and thus, only one variable is removed at random. This process is applied to all the individuals in the population in order to avoid the generation of solutions that make no sense (because of their exorbitant number of rules).

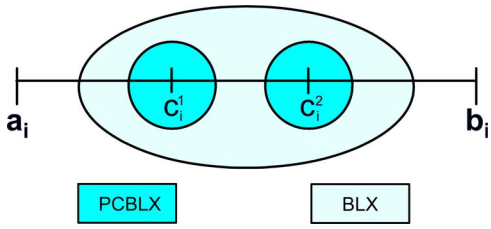

Fig. 3. Behavior of the PCBLX and BLX operators.

4) Crossover and Mutation Operators: The crossover operator depends on the part of the chromosome to which it is applied. A crossover point is randomly generated and the classical crossover operator is applied to this point for the C_1 part. The parent-centric BLX (PCBLX) operator [52], which is based on BLX-α, is applied to the C_2 part (Fig. 3 depicts the behavior of these kinds of operators). Specifically, PCBLX is described as follows. Let us assume that $X = (x_1 \ldots x_n)$, and $Y = (y_1 \ldots y_n)$, with $x_i, y_i \in [a_i, b_i] \subset \Re$ and $i = 1, \ldots, n$, are two real-coded chromosomes that are going to be crossed. The PCBLX operator generates the following two offspring.

1) $O_1 = (o_{11} \ldots o_{1n})$, where o_{1i} is randomly (uniformly) generated in the interval $[l_i^1, u_i^1]$, with $l_i^1 = \max\{a_i, x_i - I_i\}$, $u_i^1 = \min\{b_i, x_i + I_i\}$, and $I_i = \mid x_i - y_i \mid \cdot \alpha$. In our case, α has been fixed to 0.3.
2) $O_2 = (o_{21} \cdots o_{2n})$, where o_{2i} is randomly (uniformly) generated in the interval $[l_i^2, u_i^2]$, with $l_i^2 = \max\{a_i, y_i - I_i\}$, and $u_i^2 = \min\{b_i, y_i + I_i\}$.

This way, four new individuals are obtained by combining the two offspring generated from C_1 with the two offspring generated from C_2. For each of them, the mutation operator is applied with probability P_m. The mutation operator decreases the granularity by 1 in a gene g selected at random (i.e., $L^g = L^g - 1$) or randomly determines a higher granularity in $\{L^g + 1, \ldots, 7\}$ with the same probability. No decreasing is performed when it provokes DBs with only one input variable. The same gene is also changed at random in C_2. Finally, after considering mutation, only the two most-accurate individuals are taken as descendants.

5) Incest Prevention: An incest-prevention mechanism has been included by following the concepts of CHC [26] and by only taking into account the C_2 parts. Following the original CHC scheme (for binary coding), two parents are crossed if their hamming distance divided by 2 is more than a predetermined threshold L. Since C_2 makes use of a real coding scheme, we have to transform each gene considering a Gray code (i.e., binary code) with a fixed number of bits per gene (BITSGENE) that is determined by the system expert. This way, the threshold value is initialized as

$$L = (\#\text{Genes}C_2 \cdot \text{BITSGENE})/4.0.$$

Typically, L is decremented by 1 when there are no new individuals in the next generation. In order to step up the convergence, in our case, L will be decremented by 2 at each generation for problems with less than 50 variables. In order to increase the convergence speed for hard high-dimensional datasets, this

quantity is increased by 2 for every 50 additional variables (i.e., $L = L - 2 - 2 * \lfloor N/50 \rfloor$). Incest prevention represents a way to provide a good tradeoff between exploration and exploitation, thus avoiding unnecessary crosses of very similar solutions at the earlier stages of the algorithm.

6) Restarting and Stopping Condition: In order to get away from local optima, a restarting mechanism [26] (external population is forced to be empty) is applied by including the most-accurate individual as a part of the new population and by generating the remaining individuals at random (taking values between 1 and the granularity coded in the most-accurate individual for each gene of the C_1 part). This mechanism is applied when the threshold value L is below zero (L is set to its initial value).

The algorithm ends when a maximum number of evaluations are reached or when L is below zero for a second time, i.e., only two exploration/exploitation stages are needed to reach convergence.

IV. Partial-Error Computation on Large-Scale Databases for Elitist-Based Evolutionary Algorithms: Evolving With Estimated Errors

As stated in Section I, considering a postprocessing stage on the KBs obtained by the proposed learning algorithm represents a way to enhance the solutions. To this end, we will apply a previous MOEA, namely, SPEA2$_{E/E}$ [27], [28], to fine tune the MFs and rule selection that will help to significantly improve the performance of the simple global structure (which is initially based on strong fuzzy partitions), while the complexity is decreased.

While the problem of high dimensionality (i.e., high number of variables) is being solved at a first stage by the proposed learning algorithm, there are still two interdependant problems representing a difficult challenge in order to apply this second postprocessing stage: the convergence, which imposes a minimum number of evaluations needed (which is not a problem itself if the algorithm is well designed); and the large time consumed by error computation in large-scale datasets (i.e., datasets with a large amount of data). Since this last problem cannot be solved by their own learning or tuning strategy, and since, particularly in the case of tuning, it could present high computational times in some datasets, we propose, in this section, a new general mechanism for fast error computation on large-scale datasets. This procedure is based on taking a small percentage of the training examples to estimate the quantity of errors of bad solutions, thus only using all the examples to evaluate good candidate solutions. In what follows, we present the main steps to apply this fast fitness-evaluation procedure, which is defined, in general, as *evolving with estimated errors* with any kind of Elitist-based evolutionary algorithm, either single-objective or multiobjective elitist-based ones.

The evaluation process is based on the existence of a set of input–output training data $E = \{e_1, \ldots, e_l, \ldots, e_m\}$, with m being the dataset size. For a new solution C, whose performance is going to be computed, two sets of solutions have to be taken into account: the set of elite solutions (one solution in the case of single-objective algorithms) and the solutions previously evaluated at the current generation. Let r^e be the rate of examples used to estimate the error. In any case, *if* $\lfloor r^e * m \rfloor \geq 1000$, *then* $r^e = 1000/m$, i.e., no more than 1000 examples have to be considered. The subset of examples E^e for error estimation is obtained by randomly selecting $\lfloor r^e * m \rfloor$ new examples at each generation. Thus, E^e is kept fixed for a complete generation. After each generation, the examples are replaced by random selection from those examples that were not used in the previous generation. This way, we promote a rotation of the selected examples.

Let S_{elit} and S_{current} be the set of elite solutions and the set of evaluated solutions at the current generation, respectively. The fast error-computation process is as follows.

1) Compute the error of solution C in E^e (i.e., error estimation), and assign this error to C.
2) If by taking into account the estimated error and the solutions in S_{elit} and S_{current}, C is the candidate to become a member of S_{elit} (i.e., it presents the best error in single-objective algorithms, or it is a non dominated solution in multiobjective ones) continue to step 3; otherwise, go to step 4.
3) Perform a complete evaluation by considering the estimated error and the examples in $E - E^e$. This way, S_{elit} (the final output of the algorithm) will always contain solutions evaluated considering 100% of the examples.
4) Evaluations = evaluations + 1 (since this mechanism is proposed for saving time and not for saving evaluations).

The new mechanism for fast performance/error computation has been included in the postprocessing algorithm proposed in [27] in order to speed up the fine classic tuning process, thereby giving way to an algorithm with a low computational expense. It has also been included within the proposed learning algorithm FSMOGFS in order to improve the computational time with respect to the number of training data. With regard to this algorithm, we have to clarify that the reduced set of examples is only used to estimate the errors, i.e., we do not recommend using this set for the induction of rules. This way, we always use all the dataset to obtain the rules and the reduced one only to compute/estimate the errors.

V. Experiments and Analysis of Results

In order to evaluate the usefulness of the proposed approach, namely, FSMOGFS^e + TUN^e, in high-dimensional problems, we have used 17 real-world problems with different numbers of variables and cases. Table I sums up the main characteristics of the different problems considered in this study and shows the link to the knowledge extraction based on evolutionary learning (KEEL) project web page [53] from which they can be downloaded. These problems have been selected from minor to major complexity, covering a range from four to 85 input variables, and from 337 to 40 768 examples (even though each of them is complicated in itself in terms of the modeling task). The more complex problems are ELV, AIL, MV, and TIC because of the large number of variables and data. To the best of our knowledge, the problems have never been solved using GFSs or

TABLE I
DATASETS CONSIDERED FOR THE EXPERIMENTAL STUDY

Problem	Abbr.	Variables	Cases
Electrical Maintenance	ELE	4	1056
Auto MPG6	MPG6	5	398
Auto MPG8	MPG8	7	398
Analcat	ANA	7	4052
Abalone	ABA	8	4177
Stock	STP	9	950
Weather Izmir	WIZ	9	1461
Weather-Ankara	WAN	9	1609
MV Artificial Domain	MV	10	40768
Forest Fires	FOR	12	517
Mortgage	MOR	15	1049
Treasury	TRE	15	1049
Baseball	BAS	16	337
Elevators	ELV	18	16559
Computer-Activity	CA	21	8192
Ailerons	AIL	40	13750
The Insurance Company	TIC	85	9822

Available at `http://www.keel.es/`

linguistic fuzzy models. This is due to the long time needed to evaluate an individual and to the minimum number of evaluations needed to reach convergence of GFSs. Moreover, a large number of rules would be easily obtained for these kinds of problems, which make no sense in linguistic fuzzy modeling. Then, these problems represent an important challenge for this algorithm.

This section is organized as follows.

1) First, we describe the experimental setup and introduce the information shown at the web page associated with the paper in Section V-A.
2) Second, we compare the most-accurate solutions of our proposal with respect to three well-known accuracy-oriented single-objective related algorithms in Section V-B.
3) Third, we compare both stages, including or not the fast error-computation procedure in terms of the performance, in Section V-C.
4) Fourth, we show the computational costs of the different algorithms and discuss the scalability of the proposed approach in Section V-D.
5) Finally, in Section V-E, for each dataset, we plot the average Pareto fronts and the average results of the single-objective-based approaches. These plots provide reliable information on the form and characteristics of the Pareto fronts obtained, thus allowing us to check the trend and the kind of correlation of the training and the test errors.

A. Experimental Framework

This section describes the experimental setup, including a brief description of the methods and the nonparametric statistical tests considered for comparisons. It then introduces the contents of the web page with additional material associated with the paper.

1) Experimental Setup: In order to evaluate the effectiveness of the proposed method designed for fast learning and its applicability to large-scale problems (with and without a tuning stage; with and without fast error estimation), three well-known single-objective-based methods for the learning of accurate KBs have been considered for comparisons, i.e., GR-MF [19], GA-WM [20], and a more recent and effective approach, namely, GLD-WM [54]. These methods are also based on embedded genetic DB learning. Further, WM [51] is also considered as a reference since all of these approaches are based on it. A brief description of the studied methods is presented in the next three paragraphs, while Table II summarizes their main characteristics.

1) WM [51] algorithm is considered as a simple rule-generation method to quickly obtain RBs from a predefined DB. This method is considered as a reference since the studied algorithms are devoted to obtain good fuzzy partitions for the application of WM. The initial linguistic partitions for this method are comprised by L *linguistic terms* with uniformly distributed triangular MFs giving meaning to them. This way, we will refer this method as WM(L), with L taking values in $\{3, 5, 7\}$.
2) On the other hand, three different GFSs are devoted to obtain a complete KB (by embedded genetic DB learning) are considered for comparisons. Both are accuracy-oriented single-objective-based algorithms whose main objective is to obtain FRBSs as accurately as possible. The first one, i.e., GR-MF [19], learns the granularity for each fuzzy partition and the MFs parameters (their three definition points). The second one, I.E., GA-WM [20], learns the granularity, scaling factors, and the domains (i.e., the variable domain or working range to perform fuzzy partitioning) for each system variable. Both methods obtain the RB by means of the WM algorithm. The third one, i.e., GLD-WM [54], has been proposed more recently for an effective learning of the KB by obtaining the granularity and the individual lateral displacements of the MFs (i.e., fine parameter adaptation), which will have the benefit of obtaining higher accuracy in the results.
3) The proposed method, namely, FSMOGFS (see Section III) or FSMOGFS^e when it includes the fast error-estimation mechanism (see Section IV), has fewer freedom degrees than the other three genetic approaches selected for comparisons, which should obtain the most-accurate results from a theoretical viewpoint. Therefore, they represent a good accuracy goal for the proposed algorithm at the first stage in those problems in which they are still applicable (particularly in the case of GLD-WM, which was more recently designed to obtain as accurate as possible linguistic models). The addition of a postprocessing technique, namely, FSMOGFS + TUN or FSMOGFS^e + TUN^e when it includes the fast error-estimation mechanism, gives way to a new procedure/method working at two stages, which presents the same freedom degrees as the approaches selected for comparisons and should obtain the best results.

In all the experiments, we adopted a *fivefold cross-validation model*, i.e., we randomly split the dataset into five folds, each containing 20% of the patterns of the dataset, and used four

TABLE II
METHODS CONSIDERED FOR THE EXPERIMENTAL STUDY

Ref.	Method	Type of learning
[51]	WM(L)	Ad-hoc data-driven rule generation method with L labels
[19]	GR-MF	Gr. & MF parameters & RB by WM
[20]	GA-WM	Gr. & Scaling fact. & Domains & RB by WM
[54]	GLD-WM	Gr. & Individual Lateral MF parameters & RB by WM
-	FSMOGFS	Gr. & Lateral partition params. & RB by WM
-	FSMOGFS+TUN	FSMOGFS + (Tuning of MF parameters and rule selection by SPEA2$_{E/E}$ [27])
-	FSMOGFS^e	FSMOGFS including fast error estimation
-	FSMOGFS^e+TUN^e	FSMOGFS+TUN including fast error estimation

folds for training and one for testing.[1] For each of the five partitions, we executed six trials of the algorithms (i.e., six different seeds). For each dataset, we, therefore, consider the average results of 30 runs. In the case of FSMOGFS, the average values are calculated considering the most-accurate solution from each obtained Pareto front. Our main aim following this approach is to have the possibility of statistically comparing the single-objective approaches (only accuracy) with the most-accurate solution found by the proposed MOEA.

In order to assess whether significant differences exist among the results, we adopt statistical analysis [29]–[32] and, in particular, nonparametric tests, according to the recommendations made in [29] and [30], where a set of simple, safe, and robust nonparametric tests for statistical comparisons of classifiers has been analyzed. We will employ different approaches for multiple comparison, including Friedman's test [55], Iman and Davenport's test [56], and Holm's method [57]. For a detail description of these tests and for detail explanation of the use of nonparametric tests for data mining and computational intelligence, see `http://sci2s.ugr.es/sicidm/`. To perform the tests, we use a level of confidence $\alpha = 0.1$.

The values of the input parameters considered by GR-MF, GA-WM, and GLD-WM are population size of 61, 100 000 evaluations, 0.6 as crossover probability, and 0.2 as mutation probability per chromosome. In the case of the SPEA2-based methods (i.e., FSMOGFS, FSMOGFS^e, FSMOGFS + TUN, and FSMOGFS^e + TUN^e), we have considered an external population size of 61 (the same size used by the named single-objective algorithms) and a proportion of 1/3 rounded to 200 as standard population size. The remaining parameters for them are a maximum of 100 000 evaluations, 0.2 as mutation probability (crossover is always applied in SPEA2), 30 bits per gene for the Gray codification, $r^e = 0.2$ for the fast error-computation technique, and the set $\{2, \ldots, 7\}$ as possible numbers of labels in all the system variables for the learning approaches. The same set $\{2, \ldots, 7\}$ has been considered for GR-MF, GA-WM, and GLD-WM after comparing this configuration with the original configuration indicated by the authors in the corresponding papers [19], [20], [54] (i.e., $\{3, \ldots, 9\}$). See this study as complementary material in the associated web page at `http://sci2s.ugr.ugr.es/FS-MOGFS/`.

[1]The corresponding data partitions (i.e., fivefold) for these datasets are available at the KEEL project web page [53]: `http://sci2s.ugr.es/keel/datasets.php`.

2) Web Page Associated With the Paper: In order to provide additional material to the paper's content, we have developed a web page at `http://sci2s.ugr.es/FS-MOGFS/` in which we have included the following information.

1) the datasets' partitions employed in this paper. These partitions can be found in a table together with the main characteristics of the used datasets;
2) an Excel file with the complete tables of results. We have included an Excel file with the training and test results for all the algorithms so that any interested researcher can use them to include their own results and extend the present comparison; a figure with the average Pareto fronts for all the studied datasets;
3) some examples of the influence of the lateral displacements and the rule-cropping strategy in the proposed method;
4) some representative models in some of the datasets considered are also depicted to graphically show the kinds of models obtained by the proposed algorithm. Additionally, we have depicted the KB of the most accurate solution from the first data partition and seed in all the datasets and included them in a zip file;
5) the results of GR-MF, GA-WM, and GLD-WM with the sets $\{2, \ldots, 7\}$ and the original configuration indicated by the authors in the corresponding papers [19], [20], [54], $\{3, \ldots, 9\}$, as possible numbers of labels in all the system variables. We have included these results and the assessment of Wilcoxon's signed-rank test [58], [59] for pairwise comparison in favor of versions with $\{2, \ldots, 7\}$ in both, i.e., test error and number of rules;
6) a description of Wilcoxon's signed-rank test [58]–[60] and an explanation on how to apply it in the regression framework;
7) an introduction to the lateral tuning of MFs [25] as preliminary information to the paper.

B. Results and Analysis of the Most-Accurate Solution

The results obtained by the studied methods are shown in Table III. This table is grouped in columns by algorithms, and it shows the average of the results obtained by each algorithm in all the studied datasets. For each one, the first column shows the average number of rules and used variables (*R*/*V*). The second and third columns show the average MSE in training and test data (Tra./Tst.) together with their respective standard deviations (SDs). No values are shown with GR-MF, GA-WM, and GLD-WM in MV, ELV, CA, AIL, and TIC because the large number

TABLE III
AVERAGE RESULTS OF THE DIFFERENT ALGORITHMS IN COMPLEXITY AND ACCURACY (TRAINING/TEST)

DATASET	Measure	WM(3)			WM(5)			WM(7)			GR-MF			GA-WM			GLD-WM			FSMOGFS^e			FSMOGFS^e+TUN^e		
($V/Size$)		R/V	Tra.	Tst.	R/V	Tra.	Tst.	R/V	Tra.	Tst.	R/V	Tra.	Tst.	R/V	Tra.	Tst.	R/V	Tra.	Tst.	R/V	Tra.	Tst.	R/V	Tra.	Tst.
ELE	Mean	27/4	192241	192647	65/4	56135	56359	103/4	53092	55495	97/4	16645	18637	47/4	17230	18977	33/4	11483	13384	9/2	16153	16338	**8/2**	**9665**	**10548**
(4/1056)	SD		9658	14436		1498	4685		1955	9452		2319	3386		2501	3195		1085	1978		450	1162		823	1150
MPG6	Mean	43/5	13.552	14.649	115/5	4.136	6.096	194/5	2.642	6.382	243/5	**1.423**	28.93	186/5	1.879	8.824	82/5	2.294	**4.387**	39/3	3.894	4.866	**20/3**	2.860	4.562
(5/398)	SD		1.239	3.204		0.317	2.416		0.11	2.126		0.073	8.633		0.235	6.079		0.249	0.899		0.212	0.644		0.218	0.714
MPG8	Mean	71/7	12.709	13.739	161/7	4.143	7.195	223/7	2.372	9.811	262/7	**1.356**	49.36	214/7	1.563	15.22	135/7	1.709	4.782	43/3	3.885	**4.695**	**23/3**	2.757	4.747
(7/398)	SD		1.885	2.528		0.317	2.731		0.152	1.646		0.104	16.2		0.183	9.13		0.170	1.445		0.312	1.168		0.342	1.235
ANA	Mean	72/7	0.187	0.189	124/7	0.027	0.03	171/7	0.012	0.017	148/7	0.005	0.017	150/7	**0.003**	0.008	92/7	0.006	0.008	23/3	0.006	0.006	**10/3**	**0.003**	**0.003**
(7/4052)	SD		0.001	0.005		0	0.002		0	0.003		0.001	0.008		0.001	0.005		0.001	0.004		0.000	0.001		0.000	0.001
ABA	Mean	68/8	8.407	8.422	199/8	3.341	3.474	368/8	3.057	3.268	498/8	**2.358**	2.885	143/8	2.433	2.549	31/8	2.487	2.545	15/3	2.670	2.708	**8/3**	2.445	**2.509**
(8/4177)	SD		0.443	0.545		0.13	0.247		0.084	0.185		0.052	0.263		0.052	0.163		0.078	0.170		0.139	0.216		0.114	0.184
STP	Mean	123/9	8.852	8.951	265/9	1.576	1.624	378/9	0.611	1.488	343/9	0.4	1.543	344/9	0.389	2.192	217/9	**0.299**	**0.435**	43/3	1.393	1.456	**23/3**	0.764	0.912
(9/950)	SD		0.508	1.193		0.09	0.09		0.029	1.634		0.019	2.484		0.017	3.168		0.025	0.067		0.077	0.159		0.139	0.181
WIZ	Mean	105/9	6.944	7.368	399/9	3.107	5.961	652/9	2.036	10.56	331/9	1.176	9.602	218/9	1.233	3.529	107/9	**0.926**	1.150	17/2	1.519	1.571	**10/2**	0.929	**1.011**
(9/1461)	SD		0.72	0.909		0.27	2.498		0.048	2.197		0.077	8.879		0.065	4.023		0.041	0.123		0.094	0.168		0.057	0.164
WAN	Mean	156/9	16.063	16.393	457/9	4.878	6.305	853/9	2.692	6.538	397/9	1.406	7.381	279/9	1.522	2.82	133/9	**1.111**	2.075	11/2	1.897	2.151	**8/2**	1.441	**1.635**
(9/1609)	SD		0.961	1.7		0.405	1.052		0.11	2.101		0.067	5.404		0.065	2.825		0.077	1.407		0.208	0.916		0.178	0.582
FOR	Mean	246/12	2030	3793	375/12	1435	34235	401/12	340	1E+05	396/12	113	3300	395/12	**47**	3693	377/12	49	3847	35/3	1892	**2449**	**10/3**	1418	2628
(12/517)	SD		531	2340		505	4356		147	4708		17	2207		24	2787		18	2714		505	2146		539	2108
MOR	Mean	78/15	0.985	0.973	199/15	0.128	0.134	257/15	0.095	0.137	209/15	0.03	0.176	160/15	0.02	0.093	78/15	**0.016**	0.022	11/2	0.033	0.034	**7/2**	**0.016**	**0.019**
(15/1049)	SD		0.129	0.09		0.005	0.012		0.006	0.056		0.002	0.28		0.003	0.147		0.002	0.005		0.004	0.007		0.003	0.006
TRE	Mean	75/15	1.636	1.631	196/15	0.401	0.405	261/15	0.17	0.176	189/15	0.066	0.144	136/15	0.045	0.064	70/15	**0.033**	0.045	15/3	0.046	0.052	**9/3**	0.034	**0.044**
(15/1049)	SD		0.121	0.181		0.014	0.055		0.009	0.017		0.011	0.191		0.007	0.046		0.005	0.015		0.005	0.010		0.003	0.015
BAS	Mean	181/16	1.921	3.695	253/16	0.782	6.198	264/16	0.316	10.6	262/16	0.255	12.44	262/16	0.202	11.71	244/16	**0.138**	3.610	28/6	1.706	**2.483**	**17/6**	1.413	2.613
(16/337)	SD		0.109	0.739		0.047	0.686		0.006	1.339		0.02	2.177		0.031	2.562		0.014	0.621		0.124	0.372		0.197	0.585
MV	Mean	3812/10	12.404	12.62	24472/10	4.031	5.019	30616/10	1.963	24.83										16/3	0.159	0.160	**14/3**	**0.158**	**0.158**
(10/40768)	SD		0.245	0.228		0.027	0.076		0.002	1.352											0.031	0.032		0.038	0.037
ELV	Mean	530/18	1.723	1.73	4132/18	1.141	1.215	7769/18	0.995	1.461										15/3	1.000	1.000	**8/3**	**0.900**	**0.900**
(18/16559)	SD		0.109	0.064		0.018	0.022		0.008	0.038											0.100	0.100		0.100	0.100
CA	Mean	425/21	40.384	40.956	1539/21	8.449	12.44	2774/21	5.327	19.14										29/5	6.201	6.320	**14/5**	**5.021**	**5.216**
(21/8192)	SD		3.115	4.637		0.351	1.148		0.06	2.807											0.455	0.489		0.422	0.483
AIL	Mean	1074/40	3.539	3.581	6581/40	2.508	2.995	8593/40	**1.381**	4.678										32/4	2.343	2.367	**15/4**	**1.955**	**2.000**
(40/13750)	SD		0.225	0.259		0.025	0.084		0.011	0.198											0.222	0.236		0.268	0.274
TIC	Mean	5802/85	0.015	0.05	6598/85	0.007	0.09	6732/85	**0.007**	0.098										43/7	0.027	0.028	**20/7**	**0.027**	**0.028**
(85/9822)	SD		0	0.001		0	0.001		0	0											0.000	0.002		0.000	0.002

Results in this table (Tra./Tst. and SD) should be multiplied by 10^5, 10^{-5}, or 10^{-8} in the case of BAS, ELV, or AIL, respectively. GR-MF, GA-WM, and GLD-WM were not applicable to MV, ELV, CA, AIL, and TIC because of the large number of variables and cases provoked memory overflow errors.

of variables and cases provoked memory overflow errors after several hours running without finishing the evaluation of the initial population (some memory issues were improved in these methods to solve this problem, which helped to show results in at least some of the datasets with more than seven variables, but it was impossible to run them in these problems).

As stated above, we have included WM as a reference on different fixed granularities. By contrast, we want to compare all the studied GFSs in order to determine whether or not the proposed approach with and without tuning is working properly in terms of the test error and the number of rules. We do not include the methods without fast error estimation here since we are proposing this mechanism based on the fact that the results are almost the same and in order to speed up the process with these kinds of complex problems. In any case, since their counterparts without fast error computation are a good alternative as well, they will also be analyzed in the following sections.

Focusing on the number of rules, it is clear that the proposed algorithms have the advantage since they consider feature selection. What is remarkable about the effect of the postprocessing mechanism is that it is able to significantly reduce the number of rules, while the system error is decreased.

TABLE IV
RANKINGS THROUGH FRIEDMAN'S TEST ON TST. APPLYING ONE OR BOTH STAGES OF THE PROPOSED ALGORITHM

Only first stage: FSMOGFS^e		Complete algorithm: FSMOGFS^e+TUN^e	
Algorithm	Ranking on Tst.	Algorithm	Ranking on Tst.
GLD-WM	1.58	FSMOGFS^e+TUN^e	1.17
FSMOGFS^e	1.75	GLD-WM	2.08
GA-WM	3.0	GA-WM	3.08
GR-MF	3.67	GR-MF	3.67

In case of the test error, we adopt a statistical analysis. Since we will compare more than two algorithms together, we use nonparametric tests for multiple comparison. In order to perform a multiple comparison, it is necessary to check whether any of the results obtained by the algorithms present any inequality. In the case of finding this, we can find out, by using a *posthoc* test, which algorithms' partners' average results are dissimilar. We will use the results obtained in Tst., thus defining the control algorithm as the best-performing algorithm (which obtains the lowest value of ranking, computed through a Friedman test [55]). In order to test whether significant differences exist among all the mean values, we use Iman and Davenport's test [56]. Finally, we use Holm's [57] *posthoc* test to compare the control algorithm with the remainder.

TABLE V
HOLM'S *POSTHOC* TEST FOR THE STUDIED METHODS WITH $\alpha = 0.1$ ON TST

Control Algorithm: GLD-WM						Control Algorithm: FSMOGFS^e+TUN^e					
i	Algorithm	z	p	α/i	Hypothesis	i	Algorithm	z	p	α/i	Hypothesis
3	GR-MF	4.74	7.72E-5	0.03	Rejected	3	GR-MF	4.74	2.10E-6	0.03	Rejected
2	GA-WM	3.64	0.007	0.05	Rejected	2	GA-WM	3.64	2.76E-4	0.05	Rejected
1	FSMOGFS^e	1.74	0.75	0.1	Accepted	1	GLD-WM	1.74	0.082	0.1	Rejected
	Comparison						Comparison				
5	FSMOGFS^e vs. GR-MF	3.637	2.762E-4	0.02	Rejected	4	GLD-WM vs. GR-MF	3.004	0.003	0.025	Rejected
3	FSMOGFS^e vs. GA-WM	2.372	0.018	0.033	Rejected	3	GLD-WM vs. GA-WM	1.897	0.058	0.033	Accepted

Applying one or both stages of the proposed algorithm.

Table IV shows the rankings of the different methods considered in this study when we only apply the first stage of our algorithm (i.e., first study on FSMOGFS^e; see the left part of Table IV) and when we apply both stages (i.e., second study on FSMOGFS^e + TUN^e; see right part of Table IV). ImanDavenport's test tells us that significant differences exist among the observed results in all datasets, with p-values (i.e., 8.974E-7 and 1.622E-9) on Tst. for both studies, respectively. The best rankings are obtained by GLD-WM when only the first stage is considered and by FSMOGFS^e + TUN^e when the complete algorithm is considered.

We nowapply Holm's test to compare the best ranking method with the remaining methods for each study, and to obtain the results on FSMOGFS^e versus GR-MF and FSMOGFS^e versus GA-WM (for the first study) and on GLD-WM versus GR-MF and GLD-WM versus GA-WM (for the second study). Table V presents these results. In this table, the algorithms are ordered with respect to the z-value obtained for each study.

In the case of only using the first stage of the algorithm (first study in the left part of the table), Holm's test rejects the hypothesis of equality with the rest of the methods ($p < \alpha/i$), but FSMOGFS^e in Tst., thus indicating that GLD-WM outperforms the previous approaches, but not FSMOGFS^e. Further, when we check the statistical results with the same test on FSMOGFS^e versus GR-MF and FSMOGFS^e versus GA-WM in the bottom part of Table V, it is clear that FSMOGFS^e outperforms GR-MF and GA-WM as well. From this analysis, we can state that FSMOGFS^e outperforms the previous methods but GLD-WM in accuracy, while, of course, it outperforms all of them in complexity and scalability.

In the case of using the complete algorithm (second study in the right part of the table), Holm's test rejects the hypothesis of equality with the rest of the methods in Tst. ($p < \alpha/i$). From this analysis, we can state that FSMOGFS^e + TUN^e outperforms the previous methods in accuracy and, of course, in complexity and scalability. On the other hand, we can check in the bottom part of Table V that GLD-WM outperforms GR-MF and is very close to outperforming GA-WM (the hypothesis is accepted because of the low quantity of datasets available in the regression framework, which makes it more difficult to assess the differences in this case).

Analyzing the results shown in Table III and the statistical evidence obtained, we can highlight the following.

1) FSMOGFS^e + TUN^e obtained the best results in the test error, with FSMOGFS^e being the key point in these results and TUN^e a good complementary stage. Even though FSMOGFS^e has been designed to obtain simpler models, it is still preferable with respect to the previous approaches (thereby obtaining not so great results in accuracy with respect to GLD-WM but simpler solutions based on strong equally distributed fuzzy partitions).
2) The larger the granularities are, and therefore, the more rules obtained, the more the overfitting increases. It is particularly clear in the most-complex datasets when taking into account the results from WM (granularities from 3 to 7).
3) Both single-objective-based GFSs (i.e., GR-MF and GA-WM) overfit in most of the datasets, even though we selected the versions with the best test values to give them the possibility of competing in the best conditions. This is probably due to the low proportion of data with respect to the number of variables in these kinds of large-scale datasets. However, this is not the case with GLD-WM (one of the state-of-the-art algorithms in terms of accuracy for linguistic fuzzy modeling), which presents very competitive results in both, training and test sets.

To sum up, the proposed method obtained very simple solutions, in general, without significant overfitting, i.e., highly correlated values in training and test in all the datasets before and after fine tuning of the MFs. Another interesting aspect of the algorithm is the number of variables that it considers in the different datasets (a value of around 3–4 in most of them). In this sense, and taking into account that MPG6 is the same dataset as MPG8 without the two variables removed by experts, it seems that the method is good to remove these variables, thus obtaining practically the same results in number of rules and accuracy. Then, the proposed approach seems good, even in the case that variables without interesting additional information are initially included in the datasets. This property makes the method scalable for high-dimensional problems, for which it is still able to obtain good solutions from the point of view of the accuracy-interpretability tradeoff.

C. Analysis of the Use of Partial Performance Computation

In this section, we present the results of the different versions of the proposed technique in order to check the effects of the fast error-estimation mechanism. These results in the first stage (only fast learning) and in the complete process (including post-processing) are shown in Table VI separately. The best results are shown in boldface for each of the parts.

Taking into account the results in this table, we can observe that very similar results were obtained in the number of rules and in both kinds of errors for both parts. In any case, we can

TABLE VI
AVERAGE RESULTS OF THE PROPOSED ALGORITHMS WITHOUT AND WITH FAST FITNESS COMPUTATION IN COMPLEXITY AND ACCURACY (TRAINING/TEST)

Data set	Measure	FSMOGFS			FSMOGFS^e			FSMOGFS+TUN			FSMOGFS^e+TUN^e		
NAME$_{(V/Size)}$		R/V	Tra.	Tst.	R/V	Tra.	Tst.	R/V	Tra.	Tst.	R/V	Tra.	Tst.
ELE$_{4/1056}$	Mean	10/2	16018	**16083**	**9/2**	16153	16338	9/2	8803	9842	**8/2**	9665	10548
	SD		314	1108		450	1162		739	1391		823	1150
MPG6$_{5/398}$	Mean	**38/3**	3.85	**4.82**	39/3	3.894	4.866	22/3	2.778	**4.548**	**20/3**	2.860	4.562
	SD		0.198	0.772		0.212	0.644		0.220	1.047		0.218	0.714
MPG8$_{7/398}$	Mean	**40/3**	3.827	**4.453**	43/3	3.885	4.695	24/3	2.725	**4.381**	**23/3**	2.757	4.747
	SD		0.274	1.049		0.312	1.168		0.294	0.909		0.342	1.235
ANA$_{7/4052}$	Mean	25/3	0.006	**0.006**	**23/3**	0.006	**0.006**	17/3	0.003	**0.003**	**10/3**	0.003	**0.003**
	SD		0	0.001		0.000	0.001		0.000	0.001		0.000	0.001
ABA$_{8/4177}$	Mean	17/3	2.682	**2.697**	**15/3**	2.670	2.708	10/3	2.393	**2.454**	**8/3**	2.445	2.509
	SD		0.149	0.204		0.139	0.216		0.092	0.163		0.114	0.184
STP$_{9/950}$	Mean	44/3	1.361	1.46	**43/3**	1.393	**1.456**	25/3	0.724	**0.892**	**23/3**	0.764	0.912
	SD		0.095	0.156		0.077	0.159		0.112	0.154		0.139	0.181
WIZ$_{9/1461}$	Mean	23/3	1.469	**1.567**	**17/2**	1.519	1.571	15/3	0.867	**1.011**	**10/2**	0.929	**1.011**
	SD		0.08	0.223		0.094	0.168		0.040	0.177		0.057	0.164
WAN$_{9/1609}$	Mean	12/2	1.81	**1.823**	**11/2**	1.897	2.151	11/2	1.313	**1.581**	**8/2**	1.441	1.635
	SD		0.06	0.143		0.208	0.916		0.174	0.580		0.178	0.582
FOR$_{12/517}$	Mean	**34/4**	1873	**2254**	35/3	1892	2449	33/3	1593	**2406**	**10/3**	1418	2628
	SD		497	2265		505	2146		570	2161		539	2108
MOR$_{15/1049}$	Mean	12/2	0.032	**0.033**	**11/2**	0.033	0.034	9/3	0.015	**0.018**	**7/2**	0.016	0.019
	SD		0.005	0.008		0.004	0.007		0.004	0.005		0.003	0.006
TRE$_{15/1049}$	Mean	17/3	0.046	**0.049**	**15/3**	0.046	0.052	11/3	0.030	**0.040**	**9/3**	0.034	0.044
	SD		0.004	0.01		0.005	0.010		0.004	0.012		0.003	0.015
BAS$_{16/337}$	Mean	33/6	1.673	2.575	**28/6**	1.706	**2.483**	21/6	1.305	2.699	**17/6**	1.413	**2.613**
	SD		0.103	0.521		0.124	0.372		0.172	0.620		0.197	0.585
MV$_{10/40768}$	Mean	20/3	0.531	0.531	**16/3**	0.159	**0.160**	16/3	0.159	0.160	**14/3**	0.158	**0.158**
	SD		0.06	0.062		0.031	0.032		0.031	0.032		0.038	0.037
ELV$_{18/16559}$	Mean	21/3	1.024	1.025	**15/3**	1.000	**1.000**	8/3	0.900	**0.900**	**8/3**	0.900	**0.900**
	SD		0.065	0.063		0.100	0.100		0.200	0.200		0.100	0.100
CA$_{21/8192}$	Mean	**28/5**	6.046	**6.135**	29/5	6.201	6.320	15/5	4.763	**5.063**	**14/5**	5.021	5.216
	SD		0.456	0.474		0.455	0.489		0.404	0.760		0.422	0.483
AIL$_{40/13750}$	Mean	33/4	2.305	**2.308**	**32/4**	2.343	2.367	20/4	1.864	**1.905**	**15/4**	1.955	2.000
	SD		0.21	0.206		0.222	0.236		0.221	0.233		0.268	0.274
TIC$_{85/9822}$	Mean	44/8	0.027	**0.027**	**43/7**	0.027	0.028	25/7	0.026	**0.027**	**20/7**	0.027	0.028
	SD		0	0.001		0.000	0.002		0.000	0.002		0.000	0.002

Results in this table (Tra./Tst. and SD) should be multiplied by 10^5, 10^{-5}, or 10^{-8} in the case of BAS, ELV, or AIL, respectively.

find some little differences. FSMOGFS^e and FSMOGFS^e + TUN^e show a small increment in the test error with respect to their counterparts. Besides, they also show a small decrement in the number of rules, thereby showing that the proposed mechanism helps to obtain models with a little less complexity (in all the 17 datasets for the last stage) and a little bit less accuracy (in 12 of the 17). In any event, by checking one by one the results in each part of the table and taking into account the differences shown in Table III for the different methods, we can consider that the mechanism is very useful in both stages, i.e., FSMOGFS^e and TUN^e, since it allows almost equivalent results with respect to the original counterparts to be obtained, even though FSMOGFS^e + TUN^e also includes the slight differences derived from FSMOGFS^e.

D. Computational Times and Scalability of the Proposed Algorithm

With respect to scalability, it is very important to analyze the running times of the different methods (these times were obtained in an Intel Core 2 Quad Q9550 2.83-GHz, 8-GB RAM by using only one of the four cores). Table VII shows the running times of the fast WM algorithm (i.e., *ad hoc* method) and its cropped version. Of course, WM is practically instantaneous in many of the datasets. However, it is very interesting to see the times this simple method can take in the case of MV, ELV, CA, AIL, and TIC (which is more than 1 min in most of the cases). Since each individual evaluation in the genetic approaches is based on running WM, it represents a time-computing problem for the embedded algorithms. This is one of the main reasons why we propose the rule cropping strategy included in FSMOGFS, which is needed to ensure a maximum computing time for the WM module independently of the number of cases in any dataset (only lineal with respect to the number of variables). In fact, it can be seen from the table that while the original version is mainly dependent on the number of examples, the cropped

TABLE VII
AVERAGE TIME OF A RUN OF WM AND ITS CROPPED VERSION—MINUTES AND SECONDS (M:S) – (S)

Method	MV	ELV	CA	AIL	TIC
WM(3)	00:47 - 0.0003	00:05 - 0.0002	00:18 - 0.0009	00:16 - 0.0007	01:55 - 0.0049
WM(5)	05:10 - 0.0003	00:39 - 0.0006	00:19 - 0.0009	01:46 - 0.0018	02:09 - 0.0062
WM(7)	06:12 - 0.0005	01:10 - 0.0006	00:35 - 0.0010	02:13 - 0.0019	02:10 - 0.0068

TABLE VIII
AVERAGE TIME OF A RUN OF THE DIFFERENT GFSS—HOURS, MINUTES, AND SECONDS (H:M:S)

Method	ELE	MPG6	MPG8	ANA	ABA	STP	WIZ	WAN	FOR	MOR	TRE	BAS	MV	ELV	CA	AIL	TIC
GR-MF	09:30	08:01	23:43	2:58:27	3:26:09	32:50	58:22	1:19:12	17:40	41:06	40:47	13:29	-	-	-	-	-
GA-WM	07:27	07:49	10:45	1:46:41	2:02:47	34:43	43:34	1:16:59	21:03	40:29	39:52	16:46	-	-	-	-	-
GLD-WM	09:48	04:22	13:04	1:32:41	1:14:35	39:18	44:16	53:18	37:06	40:25	42:58	32:09	-	-	-	-	-
FSMOGFS	00:08	00:07	00:10	00:50	01:38	00:24	00:35	00:33	00:26	00:38	00:38	00:22	17:31	12:18	08:28	24:14	29:28
FSMOGFS+TUN	02:05	01:46	01:49	09:45	12:10	03:32	04:20	03:51	01:29	02:09	02:24	01:55	2:01:51	54:54	42:18	43:19	59:19
FSMOGFS^e	00:04	00:04	00:23	00:23	00:37	00:12	00:14	00:14	00:17	00:16	00:16	00:11	04:59	03:38	02:45	05:17	09:23
FSMOGFS^e+TUN^e	00:42	01:00	01:31	05:17	03:54	01:31	01:08	00:57	01:07	00:38	00:46	00:58	12:16	09:39	11:55	10:02	20:45

TABLE IX
AVERAGE NUMBER OF COMPLETE EVALUATIONS PER RUN OF THE PROPOSED APPROACH WITHOUT AND WITH PARTIAL EVALUATIONS (EVOLVING WITH ESTIMATED ERRORS)

Method*	ELE	MPG6	MPG8	ANA	ABA	STP	WIZ	WAN	FOR	MOR	TRE	BAS	MV	ELV	CA	AIL	TIC
FSMOGFS	4094	4868	6466	6432	7026	7816	7805	7784	9970	12374	12367	12947	8706	14608	16651	30588	32191
FSMOGFS+TUN	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
FSMOGFS^e	4085	4877	6441	6429	7028	7795	7786	7787	9979	12367	12317	12960	8711	14592	16682	30641	32287
FSMOGFS^e+TUN^e	30349	47398	40319	46799	38121	41673	33883	32268	64323	34051	35994	50598	31960	42736	43200	55383	67857

* (a maximum of 100000 individuals have been evaluated in all the fourth algorithms)

version is more dependent on the number of variables. The strong time reductions show why the cropping strategy is able to apply these kinds of techniques within an evolutionary process.

On the other hand, the running times of the studied GFSs are shown in Table VIII. In this case, except for the very complex datasets, the proposed method is able to obtain solutions taking only seconds or around 1 min. The times for the remaining datasets are also very good, thus taking into account the kinds of problems they represent and the evolutionary nature of this algorithm. From these times, we can highlight the following facts.

1) It is no more than 30 min in the worst case of FSMOGFS and no more than 2 h and 2 min in the worst case of FSMOGFS + TUN. This last result is obtained in MV with 40 768 examples, thus showing a significant increase that is dependent on the large number of examples due to the additional time required by the second stage for fine tuning.
2) It is no more than 10 min in the worst case of FSMOGFS^e and no more than 21 min in the worst case of FSMOGFS^e + TUN^e. As fast error estimation is considered, the large quantity of data of the MV problem does not affect the overall two-stage approach, thus solving this problem in 12 min and 16 s, which, in this case, represents 89.83% with respect to not using the mechanism. In fact, FSMOGFS^e + TUN^e needs less time than the initial learning algorithm without tuning, i.e., FSMOGFS, in the most-complex datasets (i.e., MV, ELV, AIL, and TIC).

In order to show the real reason for the great time savings, Table IX shows the average number of evaluations per run in the different versions of the proposed approach. For the learning stage, we will compute the number of evaluations by adding 1 per solution evaluated in all cases FSMOGFS and FSMOGFS^e. This way, in the case of using postprocessing as a second stage, this will proceed from the number of evaluations that the learning process (i.e., first stage) consumed. In the case of FSMOGFS^e + TUN^e, in order to show how the fast error computation affects the times, we consider in the table the total number of examples used throughout the process (until reaching the stopping criterion) divided by the total number of training examples to count part of the second stage, i.e., TUN^e. This is only to show the method's behavior since for the stopping criterion, we always consider 100 000 trials, i.e., evaluated individuals independent of the kind of evaluation (i.e., partial or complete).

E. Analysis of the Pareto Fronts: Average Solutions

This section analyzes the performance of the proposed algorithm in the remaining solutions that it obtains in the Pareto fronts. For this, we plot the average Pareto fronts composed by the average values of the obtained solutions in each of the 30 Pareto fronts. The first average solution that we want to plot is the one shown in the previous sections, i.e., the average of the most-accurate solutions obtained in each of the 30 Pareto fronts. The second average solution is obtained in the same way, but considering the second most-accurate solutions in each of the 30 Pareto fronts. This process is repeated until no more solutions remain in any of the 30 Pareto fronts. We should remark that there will be a moment at which any of the Pareto fronts will have no more solutions to compute the average of 30 solutions, i.e., the ith most-accurate solution is not available in all the Pareto fronts. In this case, the ith average solution is calculated considering the ith solutions of those Pareto fronts in which these solutions are available.

We can then find two different parts in the average Pareto fronts, i.e., *the statistically trusted zone* (i.e., the one ensuring that there will be those solutions in all the Pareto fronts) and *the nonstatistically trusted zone* (i.e., the one showing that some other solutions are available in some of the 30 runs performed). Thus, we can analyze the correlation and differences between the different solutions obtained by the FSMOGFS algorithm in terms of the Pareto fronts obtained with the average values, which provide more reliable information than the solutions obtained in a simple run. This method represents an extension of the idea of analyzing the most-accurate solutions in the Pareto fronts (i.e.,

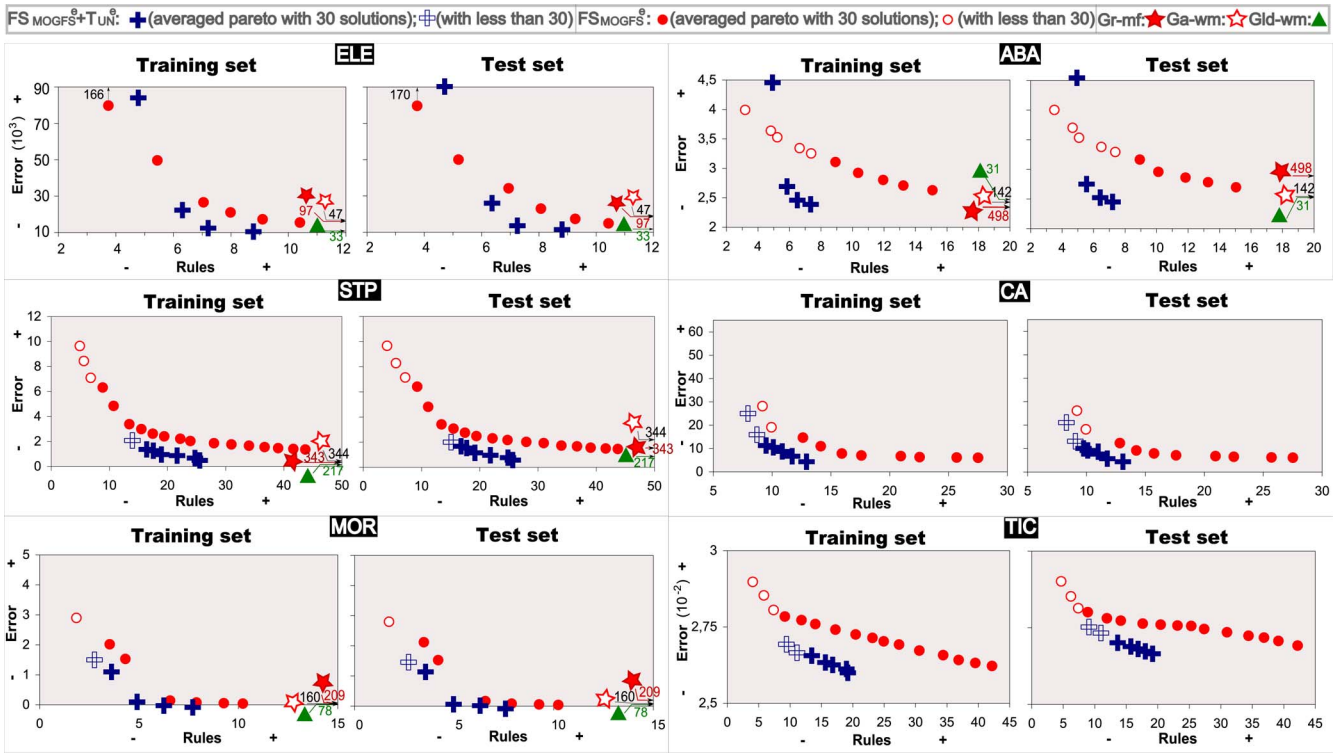

Fig. 4. Average Pareto fronts obtained by FSMOGFS^e and FSMOGFS^e + TUN^e and average solutions obtained by GR-MF, GA-WM, and GLD-WM on the different datasets.

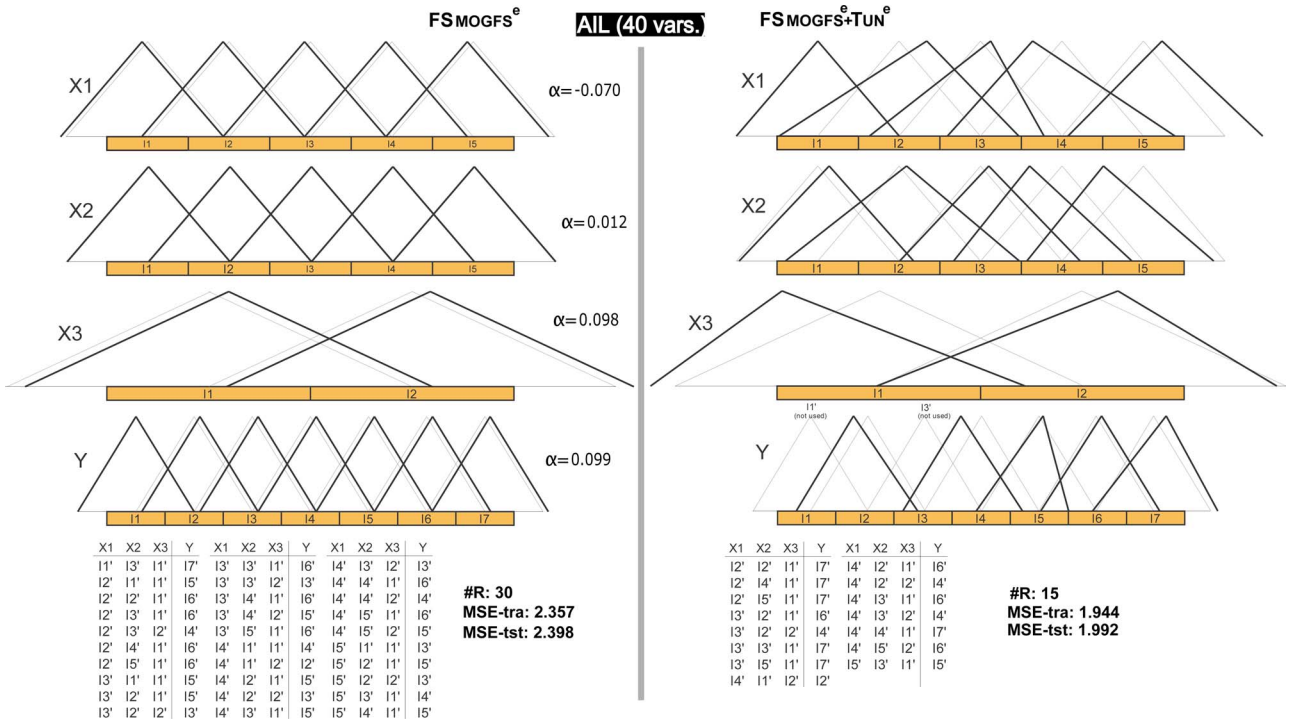

Fig. 5. Pareto fronts obtained by FSMOGFS^e and FSMOGFS^e + TUN^e on the dataset Ailerons.

first, second, ...) presented in [7], which is a postprocessing mechanism where the search is focused on the most-accurate solutions only. The average Pareto fronts represent what a user should statistically expect when he is choosing the ith most-accurate solution obtained from FSMOGFS.

The average Pareto fronts obtained in a representative set of the studied datasets are shown in Fig. 4. This figure also includes the average solutions obtained by the two single-objective-based GFSs considered for comparisons. We can see that in most of the datasets, many solutions in the trusted zone present better results

in test than those obtained by the single-objective (accuracy-oriented) based approaches.

A really important characteristic of the proposed method is the very high correlation among the values in training and the values in test. For each of the datasets considered, we can appreciate that the average solutions in test are a mimic of the solutions obtained in training. The main interest of this characteristic is that it makes the selection of any solution of the obtained Pareto fronts by a standard user very easy, i.e., by taking into account the final number of rules/variables and the training error (test errors are not available at the selection moment), any solution could be selected depending on the necessities of the final user expecting similar behavior in test, i.e., similar behavior when the model is applied in real life to solve a problem. This is not the typical situation since usually the solutions obtained in the Pareto front do not present totally correlated errors in training and test, which makes it very difficult to select a proper solution with the desired generalizability.

This means that there are very simple solutions that could be selected without losing a high degree of accuracy. In any case, as mentioned before, a user could select any solution in the Pareto front if the training error is acceptable to solve the problem. Moreover, we can see that there is no overfitting in any of the different parts of the obtained Pareto fronts.

The said behavior can also be checked for the remaining datasets in a figure with the plots of the average Pareto fronts together with the global result excel files on the web page associated with the paper at `http://sci2s.ugr.es/FS-MOGFS/`. In Fig. 5, we show two representative KBs (which are the results of a single trial) on the dataset Ailerons. Additionally, we have depicted the KB of the most-accurate solution from the first data partition and seed in all the datasets. These graphics have been included in a zip file on the web page associated with the paper, together with a brief analysis of some KB examples as complementary material to the paper.

VI. Conclusion

In this study, we have proposed an effective MOEA for the learning of linguistic KBs in high-dimensional regression problems, namely, FSMOGFS. This method, which is based on embedded DB learning, allows a slight uniform displacement of the linguistic fuzzy partitions and includes some effective mechanisms in order to enable the derivation of simple and accurate linguistic FRBSs in problems that are difficult to solve with standard evolutionary methods. A postprocessing stage performing a rule selection and a tuning of the MFs has been applied as well for further refinement of the simple learned solutions. This helps to significantly improve the performance of the simple global structure (which is initially based on strong fuzzy partitions), while the complexity is significantly decreased.

In order to also take into account datasets with a large amount of data, i.e., large-scale problems, we have also proposed a mechanism to avoid using a big percentage of the examples for error computation, estimating it from a reduced subset of the examples, but maintaining the performance and general behavior of the methods. This mechanism has been defined, in general, for *evolving with estimated errors* for use with any kind of Elitist-based evolutionary algorithm, either single-objective or multiobjective elitist-based ones. By doing so, we can also apply a postprocessing stage to further refine the learned solutions. We have included this new error-estimation procedure in both the learning and the postprocessing stages.

The results obtained in 17 datasets of different complexities confirm the effectiveness of the proposed method, particularly in terms of the simplicity and generalizability of the obtained models, but in terms of dimensionality and scalability as well (particularly when using the fast error-estimation mechanism). We have shown that the scalability of both FSMOGFS^e and FSMOGFS^e + TUN^e is a key characteristic of these approaches, which are able to solve problems with more than 40 000 cases or more than 80 variables in a very fast way. Additionally, FSMOGFS^e + TUN^e is able to obtain promising linguistic models, thus avoiding overfitting and keeping uniformly distributed strong fuzzy partitions in its first stage and refined ones in its second stage, with very competitive results in terms of accuracy.

The obtained KBs are the result of the application of the well-known WM algorithm, which, based on covering criteria, provides highly meaningful rules at each region of the modeled surface. Further, because of this and because FSMOGFS^e considers uniformly distributed fuzzy partitions, the models obtained by this method are able to be further postprocessed (approximate tuning, linear consequents learning, etc.), becoming a starting point for these kinds of techniques.

References

[1] W. E. Combs and J. E. Andrews, "Combinatorial rule explosion eliminated by a fuzzy rule configuration," *IEEE Trans. Fuzzy Syst.*, vol. 6, no. 1, pp. 1–11, Feb. 1998.

[2] Y. Jin, "Fuzzy modeling of high-dimensional systems: Complexity reduction and interpretability improvement," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 2, pp. 212–221, Apr. 2000.

[3] F. Herrera, "Genetic fuzzy systems: Taxonomy, current research trends and prospects," *Evol. Intell.*, vol. 1, pp. 27–46, 2008.

[4] O. Cordón, F. Herrera, F. Hoffmann, and L. Magdalena, *Genetic Fuzzy Systems. Evolutionary Tuning and Learning of Fuzzy Knowledge Base*, (Advances in Fuzzy Systems—Applications and Theory, vol. 19). Singapore: World Sci., 2001.

[5] O. Cordón, F. Gomide, F. Herrera, F. Hoffmann, and L. Magdalena, "Ten years of genetic fuzzy systems: Current framework and new trends," *Fuzzy Sets Syst.*, vol. 41, no. 1, pp. 5–31, 2004.

[6] R. Alcalá, M. J. Gacto, F. Herrera, and J. Alcalá-Fdez, "A multiobjective genetic algorithm for tuning and rule selection to obtain accurate and compact linguistic fuzzy rule-based systems," *Int. J. Uncertainty, Fuzziness Knowledge-Based Syst.*, vol. 15, no. 5, pp. 539–557, 2007.

[7] M. J. Gacto, R. Alcalá, and F. Herrera, "Adaptation and application of multi-objective evolutionary algorithms for rule reduction and parameter tuning of fuzzy rule-based systems," *Soft Comput.*, vol. 13, no. 5, pp. 419–436, 2009.

[8] A. Botta, B. Lazzerini, F. Marcelloni, and D. C. Stefanescu, "Context adaptation of fuzzy systems through a multi-objective evolutionary approach based on a novel interpretability index," *Soft Comput.*, vol. 13, no. 5, pp. 437–449, 2008.

[9] M. Gacto, R. Alcalá, and F. Herrera, "Integration of an index to preserve the semantic interpretability in the multi-objective evolutionary rule selection and tuning of linguistic fuzzy systems," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 3, pp. 515–531, Jun. 2010.

[10] H. Ishibuchi and T. Yamamoto, "Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining," *Fuzzy Sets Syst.*, vol. 141, no. 1, pp. 59–88, 2004.

[11] H. Ishibuchi and Y. Nojima, "Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning," *Int. J. Approx. Reason.*, vol. 44, no. 1, pp. 4–31, 2007.

[12] P. Pulkkinen and H. Koivisto, "Fuzzy classifier identification using decision tree and multiobjective evolutionary algorithms," *Int. J. Approx. Reason.*, vol. 48, pp. 526–543, 2008.

[13] P. Pulkkinen, J. Hytönen, and H. Koivisto, "Developing a bioaerosol detector using hybrid genetic fuzzy systems," *Eng. Appl. Artif. Intell.*, vol. 21, no. 8, pp. 1330–1346, 2008.

[14] R. Alcalá, P. Ducange, F. Herrera, B. Lazzerini, and F. Marcelloni, "A multi-objective evolutionary approach to concurrently learn rule and data bases of linguistic fuzzy rule-based systems," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 5, pp. 1106–1122, Oct. 2009.

[15] M. Antonelli, P. Ducange, B. Lazzerini, and F. Marcelloni, "Learning concurrently partition granularities and rule bases of Mamdani fuzzy systems in a multi-objective evolutionary framework," *Int. J. Approx. Reason.*, vol. 50, no. 7, pp. 1066–1080, 2009.

[16] M. Antonelli, P. Ducange, B. Lazzerini, and F. Marcelloni, "Multi-objective evolutionary learning of granularity, membership function parameters and rules of Mamdani fuzzy systems," *Evol. Intell.*, vol. 2, no. 1/2, pp. 21–37, 2009.

[17] M. Cococcioni, P. Ducange, B. Lazzerini, and F. Marcelloni, "A pareto-based multi-objective evolutionary approach to the identification of Mamdani fuzzy systems," *Soft Comput.*, vol. 11, pp. 1013–1031, 2007.

[18] A. A. Márquez, F. A. Márquez, and A. Peregrín, "Rule base and adaptive fuzzy operators cooperative learning of Mamdani fuzzy systems with multi-objective genetic algorithms," *Evol. Intell.*, vol. 2, no. 1-2, pp. 39–51, 2009.

[19] O. Cordón, F. Herrera, and P. Villar, "Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base," *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 4, pp. 667–674, Aug. 2001.

[20] O. Cordón, F. Herrera, L. Magdalena, and P. Villar, "A genetic learning process for the scaling factors, granularity and contexts of the fuzzy rule-based system data base," *Inf. Sci.*, vol. 136, pp. 85–107, 2001.

[21] B. Filipic and D. Juricic, "A genetic algorithm to support learning fuzzy control rules from examples," in *Genetic Algorithms and Soft Computing*, F. Herrera and J. L. Verdegay, Eds. Heidelberg, Germany: Physica-Verlag, 1996, pp. 403–418.

[22] D. Simon, "Sum normal optimization of fuzzy membership functions," *Int. J. Uncertainty, Fuzziness Knowledge-Based Syst.*, vol. 10, no. 4, pp. 363–384, 2002.

[23] W. Pedrycz, "Associations and rules in data mining: A link analysis," *Int. J. Intell. Syst.*, vol. 19, no. 7, pp. 653–670, 2004.

[24] Y. Teng and W. Wang, "Constructing a user-friendly ga-based fuzzy system directly from numerical data," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 5, pp. 2060–2070, Oct. 2004.

[25] R. Alcalá, J. Alcalá-Fdez, and F. Herrera, "A proposal for the genetic lateral tuning of linguistic fuzzy systems and its interaction with rule selection," *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 4, pp. 616–635, Aug. 2007.

[26] L. Eshelman, "The chc adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination," in *Foundations of Genetic Algorithms*, G. Rawlin, Ed. San Mateo, CA: Morgan Kaufmann, 1991, vol. 1, pp. 265–283.

[27] M. J. Gacto, R. Alcalá, and F. Herrera, "An improved multi-objective genetic algorithm for tuning linguistic fuzzy systems," in *Proc. Int. Conf. Inf. Process. Manage. Uncertainty Knowledge-Based Syst.*, Málaga, Spain, 2008, pp. 1121–1128.

[28] M. J. Gacto, R. Alcalá, and F. Herrera, "A multi-objective evolutionary algorithm for an effective tuning of fuzzy logic controllers in heating, ventilating and air conditioning systems," *Appl. Intell.,* doi:10.1007/s10489-010-0264-x.

[29] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learning Res.*, vol. 7, pp. 1–30, 2006.

[30] S. García and F. Herrera, "An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons," *J. Mach. Learning Res.*, vol. 9, pp. 2677–2694, 2008.

[31] S. García, A. Fernández, J. Luengo, and F. Herrera, "A study of statistical techniques and performance measures for genetics-based machine learning: Accuracy and interpretability," *Soft Comput.*, vol. 13, no. 10, pp. 959–977, 2009.

[32] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 special session on real parameter optimization," *J. Heuristics*, vol. 15, no. 6, pp. 617–644, 2009.

[33] F. Herrera and L. Martínez, "A 2-tuple fuzzy linguistic representation model for computing with words," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 6, pp. 746–752, Dec. 2000.

[34] O. Cordón and F. Herrera, "A three-stage evolutionary process for learning descriptive and approximative fuzzy logic controller knowledge bases from examples," *Int. J. Approx. Reason.*, vol. 17, no. 4, pp. 369–407, 1997.

[35] R. Babuška, J. Oosterhoff, A. Oudshoorn, and P. M. Bruijn, "Fuzzy self-tuning PI control of pH in fermentation," *Eng. Appl. Artif. Intell.*, vol. 15, no. 1, pp. 3–15, 2002.

[36] P. P. Bonissone, P. S. Khedar, and Y.-T. Chen, "Genetic algorithms for automated tuning of fuzzy controllers, a transportation application," in *Proc. 5th IEEE Int. Conf. Fuzzy Syst.*, New Orleans, LA, 1996, pp. 674–680.

[37] F. Herrera, M. Lozano, and J. L. Verdegay, "Tuning fuzzy logic controllers by genetic algorithms," *Int. J. Approx. Reason.*, vol. 12, pp. 299–315, 1995.

[38] J. S. R. Jang, "ANFIS: Adaptive network based fuzzy inference system," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 3, pp. 665–684, May/Jun. 1993.

[39] C. Karr, "Genetic algorithms for fuzzy controllers," *AI Expert*, vol. 6, no. 2, pp. 26–33, 1991.

[40] L. Zheng, "A practical guide to tune proportional and integral (pi) like fuzzy controllers," in *Proc. 1st IEEE Int. Conf. Fuzzy Syst.*, San Diego, CA, 1992, pp. 633–640.

[41] A. Fernandez, M. del Jesus, and F. Herrera, "On the 2-tuples based genetic tuning performance for fuzzy rule based classification systems in imbalanced data-sets," *Inf. Sci.*, vol. 180, no. 8, pp. 1268–1291, 2010.

[42] E. Mamdani, "Application of fuzzy algorithms for control of simple dynamic plant," *Proc. IEEE*, vol. 121, no. 12, pp. 1585–1588, 1974.

[43] E. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *Int. J. Man-Mach. Stud.*, vol. 7, pp. 1–13, 1975.

[44] R. Alcalá, J. Alcalá-Fdez, J. Casillas, O. Cordón, and F. Herrera, "Local identification of prototypes for genetic learning of accurate TSK fuzzy rule-based systems," *Int. J. Intell. Syst.*, vol. 22, no. 9, pp. 909–941, 2007.

[45] H. L. Wang, S. Kwong, Y. C. Jin, W. Wei, and K. F. Man, "Multi-objective hierarchical genetic algorithm for interpretable fuzzy rule-based knowledge extraction," *Fuzzy Sets Syst.*, vol. 149, no. 1, pp. 149–186, 2005.

[46] J. Casillas, O. Cordón, F. Herrera, and P. Villar, "A hybrid learning process for the knowledge base of a fuzzy rule-based system," in *Proc. Int. Conf. Inf. Process. Manage. Uncertainty Knowledge-Based Syst.*, vol. 3, Perugia, Italy, 2004, pp. 2189–2196.

[47] J. R. Cano, F. Herrera, and M. Lozano, "Evolutionary stratified training set selection for extracting classification rules with trade-off precision-interpretability," *Data Knowl. Eng.*, vol. 60, pp. 90–108, 2007.

[48] Y. Nojima, H. Ishibuchi, and I. Kuwajima, "Parallel distributed genetic fuzzy rule selection," *Soft Comput.*, vol. 13, no. 5, pp. 511–519, 2009.

[49] I. Robles, R. Alcalá, J. Benítez, and F. Herrera, "Evolutionary parallel and gradually distributed lateral tuning of fuzzy rule-based systems," *Evol. Intell.*, vol. 2, no. 1/2, pp. 5–19, 2009.

[50] E. Zitzler, M. Laumanns, and L. Thiele, "Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization," in *Proc. Evol. Methods Design, Optim. Control Appl. Ind. Prob.*, Barcelona, Spain, 2001, pp. 95–100.

[51] L. Wang and J. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, no. 6, pp. 1414–1427, Nov./Dec. 1992.

[52] F. Herrera, M. Lozano, and A. Sánchez, "A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study," *Int. J. Intell. Syst.*, vol. 18, pp. 309–338, 2003.

[53] J. Alcalá-Fdez, L. Sánchez, S. García, M. del Jesus, S. Ventura, J. Garrell, J. Otero, C. Romero, J. Bacardit, V. Rivas, J. Fernández, and F. Herrera, "KEEL: A software tool to assess evolutionary algorithms to data mining problems," *Soft Comput.*, vol. 13, no. 3, pp. 307–318, 2009.

[54] R. Alcalá, J. Alcalá-Fdez, F. Herrera, and J. Otero, "Genetic learning of accurate and compact fuzzy rule based systems based on the 2-tuples linguistic representation," *Int. J. Approx. Reason.*, vol. 44, no. 1, pp. 45–64, 2007.

[55] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *J. Amer. Stat. Assoc.*, vol. 32, pp. 675–701, 1937.
[56] R. L. Iman and J. H. Davenport, "Approximations of the critical region of the friedman statistic," *Commun. Statist. A Theory Methods*, vol. 9, pp. 571–595, 1980.
[57] S. Holm, "A simple sequentially rejective multiple test procedure," *Scand. J. Statist.*, vol. 6, pp. 65–70, 1979.
[58] D. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*. London, U.K./Boca Raton, FL: Chapman & Hall/CRC, 2003.
[59] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics*, vol. 1, pp. 80–83, 1945.
[60] J. Zar, *Biostatistical Analysis*. Upper Saddle River, NJ: Prentice-Hall, 1999.

Rafael Alcalá received the M.Sc. and Ph.D. degrees, both in computer science from the University of Granada, Granada, Spain, in 1998 and 2003, respectively.

He is currently an Associate Professor with the Department of Computer Science and Artificial Intelligence, University of Granada, where he is a member of the Soft Computing and Intelligent Information Systems Research Group. From 1998 to 2003, he was with the Department of Computer Science, University of Jaén, Jaén, Spain. He has authored or coauthored more than 75 papers in international journals, book chapters, and conferences. He has worked on several research projects supported by the Spanish Government and the European Union. He coedited the IEEE TRANSACTIONS ON FUZZY SYSTEMS Special Issue on "*Genetic Fuzzy Systems: What's next*," the *Evolutionary Intelligence* Special Issue on "*Genetic Fuzzy Systems: New Advances*," and the *Soft Computing* Special Issue on "*Evolutionary Fuzzy Systems.*" He is currently a member of the Editorial/Reviewer Board of the following journals: the *International Journal of Computational Intelligence Research*, the *Journal of Advanced Research in Fuzzy and Uncertain Systems*, and the *Journal of Universal Computer Science and Applied Intelligence*. His current research interests include multiobjective genetic algorithms and genetic fuzzy systems, particularly the learning/tuning of fuzzy systems for modeling and control with a good tradeoff between accuracy and interpretability, as well as fuzzy association rules.

Dr. Alcalá is a member of the Fuzzy Systems Technical Committee of the IEEE Computational Intelligence Society and the President of the "Genetic Fuzzy Systems" Task Force from January 2009. He was the Program Co-Chair at GEFS 2010, General Co-Chair at the Fifth IEEE International Workshop on Genetic and Evolutionary Fuzzy Systems (GEFS 2011) and the Area Co-Chair at FUZZ-IEEE 2011.

María José Gacto received the M.Sc. and Ph.D. degrees, both in computer science from the University of Granada, Granada, Spain, in 1999 and 2010, respectively.

She is currently a member of the Intelligent Systems and Data Mining Research Group, Department of Computer Science, University of Jaén, Jaén, Spain. She has more than 30 international publications. She has worked on several research projects supported by the Spanish Government and the European Union. Her current research interests include, multiobjective genetic algorithms and genetic fuzzy systems, particularly the learning/tuning of fuzzy systems for modeling and control with a good tradeoff between accuracy and interpretability.

Francisco Herrera (M'10) received the M.Sc. and Ph.D. degrees, both in mathematics from the University of Granada, Granada, Spain, in 1988 and 1991, respectively.

He is currently a Professor with the Department of Computer Science and Artificial Intelligence, University of Granada. He has authored or coauthored more than 200 papers in international journals. He is a coauthor of the book *Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases* (World Scientific, 2001). He has coedited five international books and 21 special issues in international journals on different soft-computing topics. He is currently the Editor-in-Chief of the international journal *Progress in Artificial Intelligence* (Springer) and is an Area Editor of the *Journal Soft Computing* (area of evolutionary and bioinspired algorithms). He is currently an Associate Editor of the following journals: the IEEE TRANSACTIONS ON FUZZY SYSTEMS, *Information Sciences*, *Advances in Fuzzy Systems*, and the *International Journal of Applied Metaheuristics Computing*. He is a member of the Editorial Boards of several journal, including *Fuzzy Sets and Systems*, *Applied Intelligence*, *Knowledge and Information Systems*, *Information Fusion*, *Evolutionary Intelligence*, the *International Journal of Hybrid Intelligent Systems*, *Memetic Computation*, and *Swarm and Evolutionary Computation*, among others. His current research interests include computing with words and decision making, bibliometrics, data mining, data preparation, instance selection, fuzzy-rule-based systems, genetic fuzzy systems, knowledge extraction based on evolutionary algorithms and memetic and genetic algorithms.

Dr. Herrera received the following honors and awards: the European Coordinating Committee for Artificial Intelligence Fellow 2009, the 2010 Spanish National Award on Computer Science ARITMEL to the "Spanish Engineer on Computer Science," and the International Cajastur "Mamdani" Prize for Soft Computing (Fourth ed. 2010).