M. J. Gacto, M. Galende, R. Alcalá and F. Herrera, "Obtaining accurate TSK Fuzzy Rule-Based Systems by Multi-Objective Evolutionary Learning in high-dimensional regression problems," 2013 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Hyderabad, India, 2013, pp. 1-7, doi: 10.1109/FUZZ-IEEE.2013.6622381

Obtaining Accurate TSK Fuzzy Rule-Based Systems by Multi-Objective Evolutionary Learning in High-Dimensional Regression Problems

María José Gacto
Dept. Computer Science,
University of Jaén,
23071 Jaén, Spain
Email: mgacto@ujaen.es

Marta Galende
CARTIF Centro Tecnológico,
47151 Boecillo (Valladolid), Spain
Email: margal@cartif.es

Rafael Alcalá and Francisco Herrera
Dept. Computer Science and Artificial
Intelligence, University of Granada,
18071 Granada, Spain
Email: {alcala,herrera}@decsai.ugr.es

***Abstract*—This paper addresses the challenging problem of fuzzy modeling in high-dimensional and large scale regression datasets. To this end, we propose a scalable two-stage method for obtaining accurate fuzzy models in high-dimensional regression problems using approximate Takagi-Sugeno-Kang Fuzzy Rule-Based Systems. In the first stage, we propose an effective Multi-Objective Evolutionary Algorithm, based on an embedded genetic Data Base learning (involved variables, granularities and a slight lateral displacement of fuzzy partitions) together with an inductive rule base learning within the same process. The second stage is a post-processing process based on a second MOEA to perform a rule selection and a fine scatter-based tuning of the Membership Functions. Moreover, it incorporates an efficient Kalman filter to estimate the coefficients of the consequent polynomial functions in the Takagi-Sugeno-Kang rules. In both stages, we include mechanisms in order to significantly improve the accuracy of the model and to ensure a fast convergence in high-dimensional regression problems.**

The proposed method is compared to the classical ANFIS method and to a well-known evolutionary learning algorithm for obtaining accurate TSK systems in 8 datasets with different sizes and dimensions, obtaining better results.

***Index Terms*—Accurate Fuzzy Modeling, Multi-Objective Genetic Algorithms, Regression, High-Dimensional and Large-Scale Problems.**

I. Introduction

A good alternative for precise fuzzy modeling is the use of the Takagi-Sugeno-Kang (TSK) Fuzzy Rule-Based Systems (FRBSs) [1], [2]. The rule structure in TSK model is formed by linguistic variables in the antecedent and a polynomial function of the input variables in the consequent. This rule structure involves the loss of interpretability to some degree, although it allows the model to be more accurate which is a good property for precise fuzzy modeling. However, due to the large number of involved parameters and the necessity to discover an appropiate structure for the obtained rule-based model, this becomes a challenging problem when addressing high-dimensional or large scale regression datasets [3], [4], [5].

Moreover, the learning of premises and consequents is usually done in different stages, even alternatively, due to the high complexity of the involved search space. But ideally, both parts (antecedents and consequents) should be obtained together within the same process, since they are dependent on each other.

Evolutionary Algorithms (EAs) are able to learn together the antecedents and consequents of the TSK rules, but they still have scalability problems in terms of computational time and convergence in datasets with high number of variables (high dimensional datasets) and/or with large amount of data (large-scale datasets). In fact, this is one of the open topics on the application of EAs to learn FRBSs, which are known as Genetic Fuzzy Systems [6] in general and Multi-Objective Evolutionary Fuzzy Systems [7] in particular, when considering Multi-Objective Evolutionary Algorithms (MOEAs) [8], [9].

In this contribution, we present a scalable two-stage evolutionary method for precise fuzzy modeling by TSK FRBSs. In the first stage, we propose an effective MOEA, based on an embedded genetic Data Base (DB) learning [6] (involved variables, granularities and a slight lateral displacement [10] of fuzzy partitions). The Rule Base (RB) is obtained within the same process using an efficient *ad-hoc* algorithm to estimate the coefficients of the TSK consequents. The proposed MOEA includes some specific mechanisms to ensure a fast learning of the candidate TSK FRBS structure. The second stage is a post-processing process based on a second MOEA to perform a rule selection and a fine scatter-based tuning of the Membership Functions (MFs). Moreover, it incorporates a new efficient hybridization of a Kalman filter [11] to estimate the coefficients of the consequent polynomial functions, which helps to significantly improve the performance of the model. In both stages, we propose the use of MOEAs as a tool to control the dimensionality of the models and the system overfitting, but with the main global objective of obtaining accurate models.

To show the advantages of the proposed method, it is tested on 8 different problems with a number of variables ranging from 2 to 40 and a number of samples ranging from 1056 to 22784. In addition, we compared our proposal against two methods, the classical ANFIS method [12] and an evolu-

tionary learning algorithm [13], in order to have some good performance references. The results obtained demonstrate the effectiveness of the proposed method in terms of accuracy, and particularly in terms of scalability.

The next section describes the general TSK fuzzy model structure considered in this work. Section III presents the proposed method describing its main characteristics and the genetic operators considered. Sections IV and V shows the experimental study and draws some conclusions.

II. Takagi-Sugeno-Kang Fuzzy Rule-Based Systems

In [1], [2], Takagi and Sugeno proposed a fuzzy model based on rules in which the antecedents are comprised of linguistic variables and in which the consequent is not a linguistic variable but a function of the input variables. These kinds of rules present the following structure:

$$\begin{array}{c} If\ X_1\ is\ A_1\ and\ \ldots\ and\ X_n\ is\ A_n\ then \\ Y = p_1 \cdot X_1 + \ldots + p_n \cdot X_n + p_0, \end{array} ,$$

where X_i are the system input variables, Y is the system output variable, p_i are real-values coefficients and A_i are fuzzy sets. Such rules are called *TSK fuzzy rules*, in allusion to its creators [2].

The output of a TSK FRBS considering a Knowledge Base (KB) composed of m TSK rules is computed as the weighted average of the individual rule output Y_i, $i = 1 \ldots m$:

$$\frac{\sum_{i=1}^{m} h_i \cdot Y_i}{\sum_{i=1}^{m} h_i},$$

with $h_i = T(A_1(x_1), \ldots, A_n(x_n))$ being the matching degree between the antecedent part of the ith rule and the current system inputs $x = (x_1, \ldots, x_n)$, and with T being a t-norm.

TSK FRBSs have been applied successfully to a great quantity of problems. The main advantage of these kinds of systems is the fact of presenting a compact system equation for estimating the parameters p_i using classical methods, and obtaining an accurate system, which can be very useful for accurate fuzzy modeling.

On the other hand, instead of considering linguistic partitions, we could consider scatter partitions. The scatter approach is based on rules presenting the following structure:

$$R_i : If\ X_1\ is\ A_{i1}\ and\ \ldots\ and\ X_n\ is\ A_{in}\ then\ \ Y\ is\ B_i\ ,$$

where A_i and B_i are fuzzy sets specific to each fuzzy rule. Approaches based on scatter partitions present interesting advantages that make them very suitable for precise modeling purposes:

- The expressive power of the rules that present their own specificity in terms of the fuzzy sets involved in them, thus introducing additional degrees of freedom in the system.
- The number of rules is adapted to the complexity of the problem, needing fewer rules in simple problems, and being able to use more rules if it is necessary. This is likely to be of benefit in tackling the curse of dimensionality when scaling to multidimensional systems.

In this contribution we focus on developing accurate TSK fuzzy models based on scatter partitions, which can provide more accurate solutions to different problems, especially real-world high-dimensional and large-scale regression problems with accuracy as the main requirement.

III. Proposed method

This section presents the proposed two stage method for regression problems with high number of variables and/or examples. In the first stage, an effective MOEA is applied to learn an initial DB, based on a fuzzy grid in order to obtain zero-order TSK candidate rules and the second stage applies an advanced post-processing MOEA for fine scatter-based evolutionary tuning of MFs combined with a rule selection. These algorithms incorporate some of the ideas of the fast and scalable multi-objective genetic fuzzy system, FSMOGFS^e [3], for linguistic fuzzy modeling in complex regression problems.

In the following, we include a preliminary section describing a mechanism for error estimation in large-scale problems [3] and an adaptation of Wang and Mendel [14] method (WM) for obtaining zero-order TSK rules. Then, sections III-B and III-C present both stages of the proposed method.

A. Some mechanisms to address scalability and dimensionality

In this section, we present two mechanisms used in the proposed algorithm. The first one is an error estimation mechanism used in both stages of the algorithm. This mechanism avoids to use a big percentage of the examples for error computation, estimating it from a reduced subset of the examples. The second one is used only in the first stage to derive a set of TSK zero-order rules as the RB generation process.

1) Mechanism for Error Estimation: In order to handle the scalability problem in datasets with a large amount of data, we propose to use a new mechanism presented in [3] for fast error computation on large-scale datasets. This procedure is based on taking a small percentage of the training examples to estimate the error of new generated solutions. Once these errors are estimated, only those solutions in the elite set (non dominated solutions) are evaluated considering the whole set of examples.

The subset of examples E^e for error estimation is obtained by random selection of $\lfloor r^e * m \rfloor$ examples in each generation. Where r^e is the percentage of samples used to estimate the error and m being the dataset size. If$\lfloor r^e * m \rfloor \geq 1000$ *then* $r^e = 1000/m$, i.e., no more than 1000 examples will be considered. E^e is fixed for a generation. After each generation the examples are replaced by random selection from those examples that were not used in the previous generation. In this way, we promote a rotation of the selected examples.

2) An effective generation of TSK candidate rules: We apply an adaptation of the WM method [14] in order to obtain a whole KB from a given DB (a given set of linguistic terms and their associated MFs definitions). In contrast to WM, the consequents of the TSK rules are obtained with all

the coefficients with value 0 and the independent terms are computed as the average of the examples covered by the rule weighted by their matching.

This method in problems with a high number of variables and/or examples can take a long time to derive thousands of rules. To avoid this undesired situation, once it is integrated within the MOEA of our first stage, a cropping criterion has been added to the method. In this way, the method stops the process if the RB reaches a limit of 100 rules and mark the RB as incomplete. We propose a maximum number of 100 rules for the rule cropping mechanism based on some empirical trials, which showed no significant differences in models obtained with more rules. Higher values or even those that do not use cropping do not obtain significantly more accurate solutions. To penalize incomplete solutions (which should disappear during the evolution of the first stage MOEA), we estimate the number of rules as the product of the number of labels of the input variables and in the case of the Mean Squared Error it is penalized with a fixed large error.

B. First stage: Initial KB learning

The proposed MOEA is based on the embedded genetic DB learning [6] (used variables, granularities and lateral displacements of fuzzy partitions [10]) which allows a fast learning of the structure of the initial TSK FRBSs, reducing its dimensionality and making use of some effective mechanisms in order to ensure a fast convergence in high-dimensional and large-scale regression datasets.

The following subsections describe the main features of the proposed algorithm: coding scheme, objectives, initial population, crossover and mutation operators, incest prevention mechanism and stopping condition.

1) DB Codification: A double coding scheme ($C = C_G + C_L$) to represent both parts, *granularity* and *translation parameters*, is considered:

- Number of labels (C_G): This part is a vector of integer numbers with size N (with N representing the number of input variables) in which the granularities of the different variables are coded,

$$C_G = (L^1, \ldots, L^N) \ .$$

  Each gene L^i represents the number of labels used by the i-th variable and takes values in the set $\{2, \ldots, 7\}$. Additionally, it can take a value equal to 1 to determine that the corresponding variable is not used.
- Lateral displacements (C_L): In order to decrease the complexity of the classic learning of parameters (3 parameters per MF), we will follow an effective application [3] of the 2-tuples representation scheme [10] that uses only one displacement parameter per variable. See figure 1 for an example of this kind of representation.
  In this way, this part is a vector of real numbers with size N in which the lateral displacements of the different variables are coded [10]. In this way, the C_L 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 $[-0.1, 0.1]$),

$$C_L = (\alpha^1, \ldots, \alpha^N) \ .$$

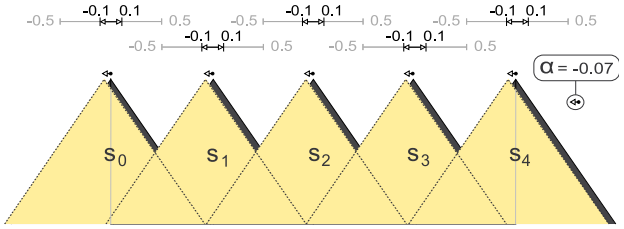

Fig. 1. Slight lateral displacement in [-0.1, 0.1].

2) Objectives: In order to evaluate a given individual, the adaptation of the WM method (see subsection III-A2) is applied to the associated DB, in order to obtain the corresponding RB. Once a complete KB is obtained the following objectives are calculated:

- Minimize the Mean Squared Error (MSE):

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

  with $|E|$ being the dataset size, $F(x^l)$ being the output obtained from the FRBS decoded from a given chromosome when the l-th example is considered and y^l being the known desired output.
- Minimize the Number of Rules (NR), to control the dimensionality.
- Maximize the medium coverage degree of the examples, to control overfitting.

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

- 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_G part, these solutions have been generated by considering all the possible combinations in the input variables, i.e., from 2 labels to 7 labels. Additionally, for each of the these combinations two copies are included with different values in the C_L part. The first one with random values in $[-0.1, 0.0]$ and the second one with random values in $[0.0, 0.1]$. If there is no space for these solutions, they are included from the smallest granularities (the most interesting combinations in principle) to the highest possible ones.
- In the second subset, we generate random solutions in order to completely fill the population (values in $\{2, \ldots, 7\}$ for C_G and values in $[-0.1, 0.1]$ for C_L).

Finally, except in the cases of problems with less than three input variables, an input variable v is removed at random, $L^v = 1$. This action is repeated until no more than 5 variables remain in all the individuals. 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).

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_G

part. The Parent Centric BLX (PCBLX) operator [15], which is based on BLX-α, is applied to the C_L part.

In this way, four new individuals are obtained by combining the two offspring generated from C_G with the two offspring generated from C_L. For each of them, the mutation operator is applied with probability P_m. The mutation operator decreases by 1 the granularity in a gene g selected at random ($L^g = L^g - 1$) 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_L. Finally, after considering mutation, only the two most accurate individuals are taken as descendants.

5) Incest Prevention and Stopping Condition: An incest prevention mechanism has been included in the C_L parts by following the concepts of CHC [16], to maintain the population diversity and avoid premature convergence. Only parents whose hamming distance divided by 4 is greater than a threshold is crossed. Because it uses a real encoding scheme in C_L, each gene is transformed into gray code with a fixed number of bits per gene ($BGenes$). This threshold value is initialized as follows: $L = (\#GenesC_L * BGene)/4$, where $\#GenesC_L$ is the number of genes in the C_L part. The algorithm ends when a maximum number of evaluations are reached or when L is below zero.

C. Second stage: post-processing (rule selection, fine tuning of MFs and one-order consequent coefficients learning)

Once a complete zero-order TSK KB is obtained in the first stage, a post-processing MOEA is applied to perform a tuning of MFs and a rule selection, which will help to significantly improve the accuracy. To this end, we present a new MOEA for accurate TSK FRBSs tuning and rule selection based on a previous MOEA, namely SPEA2$_{E/E}$ [17], [18]. The new proposed MOEA includes the error estimation procedure, described in III-A1. Further, a least-squares-based iterative mechanism has been integrated to allow consequent parameters adaptation accordingly to the system evolution.

The following subsections describe the main components of the post-processing MOEA.

1) Coding Scheme and Objectives: A triple coding scheme for *classical tuning* (C_T), *rule selection* (C_S) and *coefficients of the consequents* (C_C) is used: $C = C_T + C_S + C_C$

- Tuning of MFs (C_T): in this part a real coding is used where we consider the parameters of all the MFs per rule individually,
  $C_i = (\ldots, a_1^i, b_1^i, c_1^i, \ldots, a_{N'}^i, b_{N'}^i, c_{N'}^i, \ldots), i = 1, \ldots, m,$
  with a_j^i, b_j^i *and* c_j^i being the definition points of the j-th MF of the i-th rule, with N' being the number of input variables determined in the first stage and with m being the number of initial rules.
- Rule selection (C_S): consists of binary-coded strings with size m. Depending on whether a rule is selected or not, values '1' or '0' are respectively assigned to the corresponding gene.
- Coefficients of the consequents (C_C): This is a vector of real numbers with size $(N' + 1) * m$ in which the coefficients of the consequent polynomial function for each TSK rule are encoded,
  $C_C = (\ldots, p_1^i, \ldots, p_{N'}^i, p_0, \ldots), i = 1, \ldots, m.$

This stage of the algorithm considers the same three objectives presented in section III-B2.

2) Initial Gene Pool: The initial population is obtained with all individuals having all genes with value '1' in C_S. In the C_T part, the initial DB is included as an initial solution and the remaining individuals are randomly generated maintaining their values within their respective variation intervals.

Finally, the C_C part of the first individual includes the consequents obtained in the first stage. Then, we apply only one iteration of the standard Kalman filter to the initial individual on the reduced subset of examples E^e in order to obtain the estimated coefficients. The remaining individuals are initialized with these same coefficients. We do not use the Kalman filter to obtain the coefficients for all individuals, because it would significantly increase the computational time of the algorithm. In the next subsection, we present a different way to integrate the Kalman filter to apply it through evolution.

3) Crossover and Mutation Operators: The BLX-0.5 [19] crossover is applied to obtain the C_T part of the offspring. The binary part C_S is obtained based on the C_T parts (MF parameters) of the corresponding parents and offspring [20], [18]. The parent with the closest distance to the offspring in the C_T values of the corresponding rule is the one that determines whether this rule is selected or not for the offspring by directly copying its value in C_S for the corresponding gene (see [18] for more information on this operator).

The C_C part is obtained by directly copying its values from the parent with the closest distance in C_S to the offspring. In this way, the coefficients are only inherited from the closest parent since, actually, they will be mainly learned through the integrated efficient Kalman filter proposed in the following section.

The mutation operator is only applied in the C_S part and this favors rule extraction since mutation is only engaged to remove rules. A rule is removed at random whether this operator is applied.

4) Efficient application of the Kalman filter: Kalman filter [11] is a classic technique to estimate the coefficients of the consequent polynomial function in the TSK rules. This technique obtains good results in training, but usually presents overfitting, and therefore bad results in test. To avoid this undesired situation, only a small percentage of samples (the same percentage used to estimate errors, see section III-A1) is used to estimate the coefficients of the TSK rules. In this sense, once a new solution is generated by crossover and mutation, and evaluated on the small percentage of examples, if the estimated error is the best error until this moment, which means it would be non dominated and therefore it will be evaluated in the whole set of examples, the Kalman filter is applied in the same subset of examples to obtain

the corresponding consequent parameters before this whole evaluation. This way working provides a validation mechanism for the obtained coefficients since they should also work on the examples not used by the Kalman filter. We do not use the Kalman filter to obtain the coefficients for all individuals, because it would significantly increase the computational time of the algorithm.

Further, in order to save much more time and in order to make it converge together with MFs and rules (by selection), only one iteration of the Kalman filter is run each time. Thus, the Kalman filter is only initialized at the beginning of the algorithm and at restarting, so that the coefficients are progressively improved for those combinations of MFs and rules that continuously promote more accurate solutions. This is possible due to the kind of process (post-processing) which does not change the system structure and by considering that not selected rules are not activated by examples (matching 0) to apply the filter.

5) Restarting: This mechanism is applied when the threshold value L is below zero (L is set to its initial value). The restarting operator is applied by including the best individual for each objective. The remaining individuals take the values of the most accurate individual in the C_S part and values generated at random in the C_T part one iteration of the standard Kalman filter are applied on the most accurate individual to obtain the C_C part of the remaining individuals, which are initialized with these coefficients in their corresponding C_C parts. Additionally, it regenerates the default rule, taking into account examples whose coverage is under 0.2, for the best individual in the MSE objective.

In each stage of the algorithm (between restarting points), the number of solutions in the external population considered to form the mating pool is progressively reduced, by focusing only on those with the best accuracy. To do that, the solutions are sorted from the best to the worst (considering accuracy as criterion) and the number of solutions considered for selection is reduced progressively from 100% at the beginning to 50% at the end of each stage. It is done by taking into account the value of L. In the last evaluations when restart is disabled, this mechanism for focusing on the most accurate solutions (the most difficult objective), is also disabled in order to obtain a wide, well-formed Pareto front, from the most accurate solutions to the most interpretable ones.

IV. Experiments and Analysis of Results

To evaluate the effectiveness of the proposed approach in high-dimensional and large-scale regression datasets, we have used 8 real-world problems with different numbers of variables and cases, covering a range from 2 to 40 input variables and from 1056 to 22784. Table I sums up the main characteristics of the different problems considered in this study and shows the link to the KEEL project webpage [21] from which they can be downloaded.

TABLE I
Datasets considered for the experimental study

Problem	Abbr.	Variables	Cases
Plastic Strength	PLA	2	1650
Electrical Maintenance	ELE	4	1056
Abalone	ABA	8	4177
Weather Izmir	WIZ	9	1461
Weather Ankara	WAN	9	1609
House-16H	HOU	16	22784
Pole Telecommunications	POLE	26	14998
Ailerons	AIL	40	13750

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

The methods considered for the experimental study are:

- **ANFIS** [12] is a neural FRBS to obtain global semantics-based TSK FRBSs. This classical method obtains very accurate FRBSs, thanks to gradient descent and least squares estimation mechanisms.
- **LEL-TSK** [13] obtains highly accurate local semantics-based TSK rules. This well-known two-stage evolutionary algorithm based on MOGUL (a methodology to obtain Genetic FRBSs under the Iterative Rule Learning approach) has been developed to consider the interaction between input and output variables as a way to increase the accuracy of the obtained models.
- **MOEA-TSK** proposed here for precise fuzzy modeling.

LEL-TSK method is available in *http://www.keel.es/* and considered the error as the sole objective. The values of the parameters considered by LEL-TSK [13] are those proposed by the authors of the method and the number of evaluations is over 100000. In the case of ANFIS method [12] the parameters considered are: 5 labels per variable, 100 iterations, 0.1 step size, 0.9 and 1.1 as degrees of decrease and increase the step size, respectively.

In the case of the MOEA proposed method (MOEA-TSK) based on the well-known $SPEA2$ [22], we have considered an external population size of 61 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.

In all the experiments, we adopted a *5-fold cross-validation model*, i.e., we randomly split the dataset into 5 folds, each containing 20% of the patterns of the dataset, and used four folds for training and one for testing [1]. For each of the five partitions, we executed six trials of the algorithms (6 different seeds). For each dataset, we therefore consider the average results of 30 runs. In the case of the proposed method, the average values are calculated considering the most accurate solution from each obtained Pareto front.

The results obtained by the studied methods are shown in Table II. 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

[1]The corresponding data partitions (5-fold) for these datasets are available at the KEEL project webpage [21]: `http://sci2s.ugr.es/keel/datasets.php`

TABLE II
AVERAGE RESULTS OF THE ANALIZED METHODS.

	ANFIS			LEL-TSK			MOEA-TSK		
Problem	**NR**	**MSE**$_{tra}$	**MSE**$_{tst}$	**NR**	**MSE**$_{tra}$	**MSE**$_{tst}$	**R/V**	**MSE**$_{tra}$	**MSE**$_{tst}$
PLA	10	1.053	1.151	66	**1.032**	1.188	19.2/2	1.057	**1.136**
ELE	20	**1653**	2103471	45	2928	3752	36.9/4	2270	**3192**
ABA	-	-	-	107	**2.040**	2.412	23.1/4.2	2.205	**2.392**
WAN	-	-	-	123	0.709	1.632	48/4.7	**0.701**	**1.189**
WIZ	-	-	-	116	**0.700**	2.227	29.1/4	0.729	**0.944**
HOU	-	-	-	-	-	-	30.5/5	**8.29E+08**	**8.64E+08**
POLE	-	-	-	-	-	-	46.3/6.3	**57.96**	**61.02**
AIL	-	-	-	-	-	-	48.4/6	**1.394E-08**	**1.506E-08**

shows the average number of rules (NR), the second and third columns show the average MSE in training and test data (Tra./Tst.). Further, since the proposed algorithm is able to reduce the number of used variables, we also show the average number of variables together with the rules (R/V).

No values are shown for ANFIS and LEL-TSK in several datasets since the large number of variables and cases provoked memory overflow errors after several hours running (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 8 variables, but it was impossible to run them in more complex problems).

Analyzing the results shown in Table II, we can draw the following conclusions:

- ANFIS [12] and LEL-TSK [13] methods obtain very accurate results on training, which usually leads them to present overfitting and very bad test errors.
- The proposed method presents quite simple solutions (less number of variables and rules) without significant overfitting. This method obtains the best results on test error in all the studied datasets.

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 2.83GHz, 8 GB RAM by using only one of the four cores). Table III shows the running times of the different algorithms. ANFIS [12] and LEL-TSK [13] algorithms can take a significant amount of time in problems when the number of variables and/or instances becomes high. These algorithms can not run in high-dimensional datasets, because the large number of variables and cases provoked memory overflow errors.

TABLE III
AVERAGE TIME OF A RUN OF THE DIFFERENT METHODS

Problem	ANFIS	LEL-TSK	MOEA-TSK
PLA	0:01:00	0:18:12	0:03:09
ELE	0:00:16	0:02:01	0:01:01
ABA		2:41:04	0:28:55
WAN		1:41:19	0:47:12
WIZ		1:29:14	0:19:33
HOU			5:07:58
POLE			4:40:22
AIL			5:26:30

In order to depict the dependencies with respect to the number of variables and cases in the different datasets, Figure 2 includes three graphics representing the running times when the datasets are sorted by number of variables, by number of cases/examples or by the result of multiplying them (variables $\times$ examples). The times of these algorithms (ANFIS and LEL-TSK) in the datasets where they are not applicable have been estimated from the time in ABA (4000 examples approximately) by assuming a lineal progression of the time when more examples are considered. In the case of the proposed method, the times needed are quite good by taking into account that the X axis in these graphics considers a discrete representation with practically the double of variables for each of the last three datasets but the times increase in a very good proportion.

To sum up, the results show that the proposed algorithm makes it able to obtain the best results within a reasonable time (which is not highly affected in complex problems).

V. CONCLUSION

This work presents a scalable two-stage multi-objective genetic algorithm for precise fuzzy modeling of scatter-based TSK FRBSs in high-dimensional and large-scale regression problems.

The proposed method has been compared to two well recognized methods, ANFIS [12] and a related evolutionary learning algorithm (LEL-TSK [13]), in 8 problems with different numbers of samples and variables, showing better results in accuracy within a reasonable computation computing time. MOEA-TSK has shown that it is able to obtain very accurate models avoiding overfitting on test error. Moreover, the scalability of the proposed method is also a key characteristic, which is able to solve problems with 40 variables or more than 2200 cases in a fast way (still reasonable for an evolutionary-based approach).

ACKNOWLEDGMENT

Supported by the Spanish Ministry of Education and Science under grant no. TIN2011-28488, Spanish Ministry of Science and Innovation under grant no. DPI2009-14410-C02-02 and the Andalusian Government under grant no. P10-TIC-6858.

REFERENCES

[1] M. Sugeno and G. Kang, "Structure identification of fuzzy model," *Fuzzy Sets and Systems*, vol. 28, pp. 15–33, 1988.

[2] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its application to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. 15, no. 1, pp. 116–132, 1985.

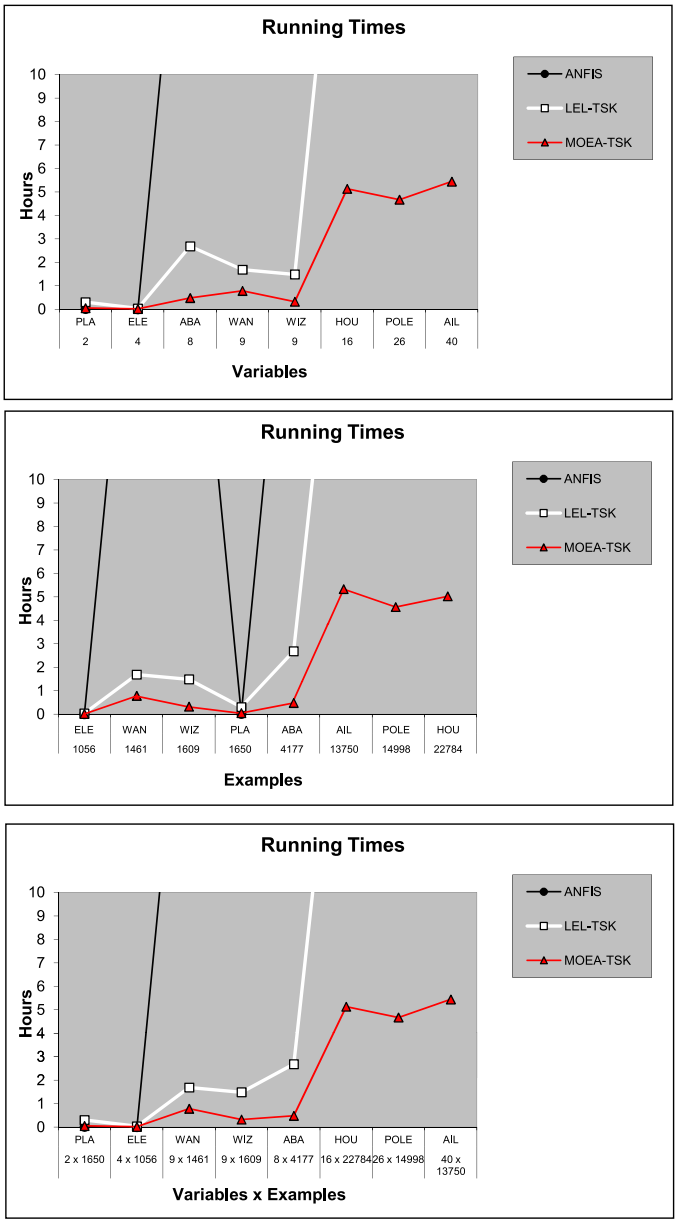

Fig. 2. Analysis of the running times in the different problems. The X axis considers discrete representation on the different data sets showing the associated values.

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

[4] M. Antonelli, P. Ducange, and F. Marcelloni, "Genetic training instance selection in multiobjective evolutionary fuzzy systems: A coevolutionary approach," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 2, pp. 276–290, 2012.

[5] M. Cococcioni, B. Lazzerini, and F. Marcelloni, "On reducing computational overhead in multi-objective genetic takagi-sugeno fuzzy systems," *Applied Soft Computing*, vol. 11, no. 1, pp. 675–688, 2011.

[6] F. Herrera, "Genetic fuzzy systems: taxonomy, current research trends and prospects," *Evolutionary Intelligence*, vol. 1, pp. 27–46, 2008.

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

[8] K. Deb, *Multi-objective optimization using evolutionary algorithms*. NY, USA: John Wiley & Sons, 2001.

[9] C. A. Coello, D. A. V. Veldhuizen, and G. B. Lamont, Eds., *Evolutionary algorithms for solving multi-objective problems*. Kluwer Academic Publishers, 2002.

[10] 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, 2007.

[11] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

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

[13] 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," *International Journal of Intelligent Systems*, vol. 22, no. 9, pp. 909–941, 2007.

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

[15] F. Herrera, M. Lozano, and A. M. Sánchez, "A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study," *International Journal of Intelligent Systems*, vol. 18, pp. 309–338, 2003.

[16] L. J. 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. Morgan Kaufman, 1991, vol. 1, pp. 265–283.

[17] M. J. Gacto, R. Alcalá, and F. Herrera, "An improved multi-objective genetic algorithm for tuning linguistic fuzzy systems," in *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2008)*.

[18] ——, "A multi-objective evolutionary algorithm for an effective tuning of fuzzy logic controllers in heating, ventilating and air conditioning systems," *Applied Intelligence*, vol. 36, no. 2, pp. 330–347, 2012.

[19] L. J. Eshelman and J. D. Schaffer, "Real-coded genetic algorithms and interval-schemata," *Foundations of Genetic Algorithms*, vol. 2, pp. 187–202, 1993.

[20] 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 Computing*, vol. 13, no. 5, pp. 419–436, 2009.

[21] 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 Computing*, vol. 13, no. 3, pp. 307–318, 2009.

[22] E. Zitzler, M. Laumanns, and L. Thiele, "Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization," in *Proc. Evolutionary Methods for Design, Optimization and Control with App. to Industrial Problems*, Barcelona, Spain, 2001, pp. 95–100.