2009 Ninth International Conference on Intelligent Systems Design and Applications

Handling High-Dimensional Regression Problems by Means of an Efficient Multi-Objective Evolutionary Algorithm

María José Gacto
Dept. Computer Science
University of Jaén
E-23071 Jaén, Spain
Email: mjgacto@ugr.es

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

***Abstract*—Linguistic fuzzy modeling in high dimensional regression problems is a challenging topic since conventional linguistic fuzzy rule-based systems suffer from exponential rule explosion when the number of variables and/or data examples becomes high. A good way to face this problem 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 final linguistic model.**

In this contribution, we propose an effective multi-objective evolutionary algorithm that based on the data base learning *a priori* (involved variables, granularities and slight uniform displacements of the fuzzy partitions) allows a fast derivation of simple and quite accurate linguistic models, making use of some effective mechanisms in order to ensure a fast convergence. The good results obtained in several large-scale regression problems demonstrate the effectiveness of the proposed approach.

***Index Terms*—High-dimensional regression problems; linguistic fuzzy modeling; complexity reduction; multi-objective genetic fuzzy systems;**

I. Introduction

Linguistic fuzzy modeling in high dimensional regression problems 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]. A good way to face this problem 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 (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). It is, by learning the KB main components, a Data Base (DB) containing the definitions of the linguistic fuzzy partitions and a Rule Base (RB) containing the associated set of rules, together.

An efficient way to obtain the whole KB of FRBSs consists of obtaining the DB and the RB within the same process but separately, based on the evolutionary learning of the DB *a priori* [2], [3], [4]. This allows learning the most adequate context [2] for each fuzzy partition, which strongly affects the final model complexity. However, this approach can not face itself the following contradictory requirements:

- The obtained linguistic models should be *simple* and *transparent* enough, but also *competitive* in terms of the generalization error (the derivation of more specific or less specific models [5] easily affects the final performance).
- The evolutionary learning algorithm should be *effective*, but also *scalable* 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 work, we propose a convenient reduction of the search space for learning the DB *a priori* (variable selection, granularities and MF parameters) and an effective and efficient Multi-Objective Evolutionary Algorithm (MOEA) that makes use of some effective mechanisms in order to ensure a fast convergence. In order to reduce the search space [6], we propose to perform 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 pre-screening on promising granularities, which avoids the derivation of very specific systems, presenting overfitting, and preserves equidistributed strong fuzzy partitions. On the other hand, the proposed MOEA includes such concepts as incest prevention and restarting in order to improve the algorithm convergence [7], together with some mechanisms to step up the learning process (such as a rule cropping criterion in the RB generation process). The results obtained in several large-scale regression problems demonstrates the effectiveness of this method in terms of simplicity, generalization ability and scalability.

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 in high dimensional problems. Section IV shows an experimental study of the proposed method. Finally, Section V points out some conclusions.

978-0-7695-3872-3/09 $26.00 © 2009 IEEE
DOI 10.1109/ISDA.2009.214

II. Lateral Displacement of Fuzzy Partitions

In [6], a new model for the tuning of FRBSs (post-processing) was proposed considering the linguistic 2-tuples representation scheme, which allows the lateral displacement of the support of a MF considering only one parameter per MF. This displacement parameter is a number in [-0.5, 0.5), expressing this interval the domain of a MF when it is moving between its two adjacent lateral MFs. In this way, the lateral tuning of MFs allows a fine adaptation of each MF comprising the DB.

Our main aim in this work 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 highly dimensional problems are considered. Moreover, to perform a fine adaptation of the parameters 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 a similar or near accuracy) that could represent more simple and interesting systems.

To solve this problem, we propose to perform a single lateral displacement of fuzzy partitions by applying a common displacement parameter to all the MFs at each linguistic variable, i.e., all the MFs are uniformly displaced depending on the displacement parameter associated to each fuzzy partition. In order to avoid very specific parameters and to preserve as much as possible the original meanings of the MFs we propose a short variation interval for these displacements. In this way, we can represent the translation of a linguistic partition S by the 2-tuple notation as,

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

with s_i being the i-th MF in S. Figure 1 shows the lateral displacement of a linguistic partition S for a concrete α value. Some interesting characteristics following this approach are:

- The search space is reduced providing a fast convergence. This makes easier to explore different granularities that can represent promising linguistic partitions.
- The constrained variation interval avoids a fine adaptation of the MFs just allowing a simple pre-screening on promising granularities, which avoids the derivation of very specific systems presenting overfitting.

All that eases a fast derivation of promising models based on equidistributed strong fuzzy partitions. Once these models are obtained, a fine tuning [8], [9] (post processing) could be easily applied depending on the user preferences. We do not consider this possibility in this contribution by focusing only on the learning stage.

III. An Effective Multi-Objective Evolutionary Algorithm

As said, the proposed algorithm is based on the DB learning *a priori* [2], [3], [4]. Following this approach, the learning

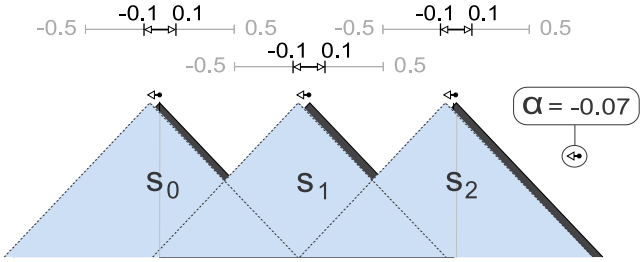

Fig. 1. Lateral displacement in [-0.1, 0.1] of the whole linguistic partition $S = \{s_0, s_1, s_2\}$. All the MFs are uniformly displaced.

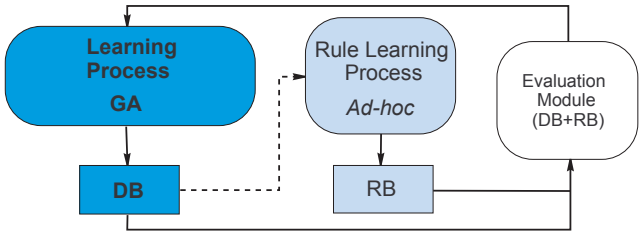

Fig. 2. Learning scheme considered for obtaining complete KBs.

scheme considered to obtain complete KBs is comprised of two main components: DB evolutionary learning and fast *ad-hoc* RB learning process (see Figure 2). However, some problems arise when high dimensional datasets are considered. The two main problems are:

1) *The large number of evaluations needed to reach convergence*. We solve this problem into two ways. By learning together the number of labels and the single partition displacement parameters (reduced search space); and, by developing an advanced MOEA that based on the well known SPEA2 ensures an effective trade-off between exploration and exploitation, thus avoiding not needed evaluations.
2) *Too much time is required to generate the RB*. Each evaluation requires the generation of a concrete RB based on the coded DB. The rule generation method can take a significant time in high dimensional problems. Due to the required number of evaluations, it represents a problem. We solve this problem by including a cropping criterion in the RB generation method. Additionally, we allow to remove the unnecessary variables while evolving, thus leading to DBs that do not provoke an excessive number of rules when the RB generation process is applied.

In this way, the proposed algorithm is comprised of these two main components:

- An efficient MOEA based on SPEA2 [10] with two minimization objectives (system error and number of rules) in order to learn promising DBs. This allows to define:
  - The number of labels for each variable, which determine the corresponding equidistributed strong linguistic partitions (including granularity 1 to express

that the corresponding variable is not used).
  - The single lateral displacements for each linguistic fuzzy partition.
- A quick *Ad-hoc* data-driven method to derive a RB from each DB definition generated by the evolutionary process. The Wang and Mendel algorithm [11] (WM) will be considered for this task by adding a cropping criterion for effectiveness.

As said, the proposed MOEA is based on the well-known $SPEA2$ [10] algorithm. However, in order to improve its search ability, this algorithm also implements such concepts as incest prevention and restarting [7]. In the following, the components needed to design this algorithm are explained. They are: DB codification, objectives, WM cropping criterion, initial gene pool, crossover and mutation, incest prevention, restarting and stopping condition.

A. DB Codification

A double coding scheme ($C = C_1 + C_2$) is considered for both parts, *granularity* and *translation parameters*:

- Number of labels (C_1): This part is a vector of integer numbers with size N (representing N 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 i-th 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 determining that the corresponding system variable is not used.
- Lateral displacements (C_2): This part is a vector with N real numbers in which the displacements of the different variables are coded. The C_2 part has the following structure (where each gene is the displacement value of the fuzzy partition of the corresponding linguistic variable an takes values in $[-0.1, 0.1]$),

  $$C_2 = (\alpha^1, \ldots, \alpha^N) \ .$$

B. Objectives and WM Cropping

Once a complete KB is obtained the following two objectives are minimized for this problem: the number of rules (simplicity) and the Mean Squared Error (accuracy),

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

with $|E|$ being the data set size, $F(x^l)$ being the output obtained from the FRBS decoded a given chromosome when the l-th 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 defuzzification operator and the *minimum t-norm* as implication and conjunctive operators.

As said, in order to obtain a complete KB from a given chromosome, we will apply WM on the DB coded by such chromosome. Since in high dimensional problems WM can take a long time deriving even thousands of rules, a cropping criterion has been added to this method. In this way, WM stops if the RB reach a maximum of 50 rules and mark the RB as not complete in order to penalize its objective values. This value (50 rules) was fixed thinking on readability and based on experimental results, since the models obtained do not present significant differences in the obtained errors by allowing a higher number of rules. In order to penalize such solutions (that should disappear when good complete solutions arise from the evolution), we estimate the number of rules as the product of the number of labels of the input variables in the DB decoded (this is a pessimist proportional estimation). In the case of the MSE, it is penalized multiplying it per 2.0 (if an example is not covered by the incomplete RB, the mid of the output domain is given as the estimated output).

C. 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_1 part, these solutions has been generated by considering all the possible combinations in the antecedent part, i.e., from 2 labels to 7 labels in all the input variables (6 combinations). For each of these combinations, all the possible combinations are generated in the consequent part (6 combinations per each input combination). Additionally two copies are included for each of the previous combinations. The first one with random values in $[-0.1, 0]$ in the C_2 part, and the second one with random values in $[0, 0.1]$. Thus, a total of 72 $(6 * 6 * 2)$ different solutions are generated.
- 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 case of problems with less than three input variables, an input variable v is removed at random, $L^v = 1$, in the first individual. This action is repeated until no more than 10 variables remains in this individual. This process is applied to all the individuals in the population in order to avoid the generation of solutions that has no sense (because of their exorbitant number of rules), therefore helping to increase the convergence of the algorithm without significantly affecting the final results.

D. Crossover and Mutation Operators

The crossover operator depends on the chromosome part where it is applied: A crossover point is randomly generated and the classical crossover operator is applied on this point for the C_1 part. On the other hand, the Parent Centric BLX (PCBLX) operator [12], which is based on the BLX-α, is applied for the C_2 part. PCBLX is described as follows. Let us assume that $X = (x_1 \cdots x_n)$ and $Y = (y_1 \cdots y_n)$, $(x_i, y_i \in [a_i, b_i] \subset \Re, i = 1 \cdots n)$, are two real-coded

chromosomes that are going to be crossed. The PCBLX operator generates the following two offspring:

- $O_1 = (o_{11} \cdots o_{1n})$, where o_{1i} is a randomly (uniformly) chosen number from 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.
- $O_2 = (o_{21} \cdots o_{2n})$, where o_{2i} is a randomly (uniformly) chosen number from the interval $[l_i^2, u_i^2]$, with $l_i^2 = max\{a_i, y_i - I_i\}$ and $u_i^2 = min\{b_i, y_i + I_i\}$.

Four offspring are generated by combining the two parts generated from C_1 with the two parts generated from C_2. Once each offspring is generated the mutation operator in 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. Besides, the same gene is changed at random in C_2. Finally, after considering mutation, only the two most accurate offspring are taken as descendant.

E. Incest Prevention

An incest prevention mechanism has been included for the C_2 part following the concepts of CHC [7]. Following the original CHC scheme (for binary coding), two parents are crossed if their hamming distance divided by 2 is over a predetermined threshold, L. Since C_2 makes use of a real coding scheme, we have to transform each gene considering a Gray Code (binary code) with a fixed number of bits per gene ($BITSGENE$), that is determined by the system expert. In this way, the threshold value is initialized as:

$$L = (\#GenesC_2 \cdot BITSGENE)/4.0.$$

Typically, L is decremented by one 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 two at each generation. Incest prevention represents a way to provide a good trade-off between exploration and exploitation, avoiding unnecessary crosses of very similar solutions at the earlier stages of the algorithm.

F. Restarting and Stopping Condition

In order to get away from local optima a restarting mechanism [7] (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. It is, only two exploration/exploitation stages are needed to reach convergence.

IV. Experiments and Analysis of Results

In order to evaluate the usefulness of the proposed approach, namely SPEA2-FL, in high dimensional problems, we have used five real-world problems with different number of variables and cases. Table I summarizes their main characteristics and shows the link to the KEEL software tool webpage [13] from which they can be downloaded.

TABLE I
Data sets considered

Problem	Abbr.	Variables	Cases
Analcat	ANA	7	4052
Weather-Ankara	WAN	9	1609
MV Artificial Domain	MV	10	40768
Baseball	BAS	16	337
Computer-Activity	CA	21	8192

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

Two single objective-based methods to obtain complete KBs are considered for comparisons, GR-MF [3] (learning Granularities and the three MF parameters) and GA-WM [2] (learning Granularities, scaling factors and domains). They are also based on the DB learning *a priori*. WM [11] is also considered as a reference since the proposed algorithm and the algorithms considered for comparisons are based on it. We will refer to this method as WM(L), with L being the number of labels used in the initial DB.

The input parameters considered for these algorithms are: the set $\{2, \ldots, 7\}$ as possible numbers of labels in all the system variables, population size of 61 (external population and standard population sizes of 61 and 200 in the case of SPEA2-FL), 100,000 evaluations, 0.6 as crossover probability (originally SPEA2 has not crossover probability) and 0.2 as mutation probability per chromosome. Further, SPEA2-FL uses 30 bits per gene for the Gray coding.

In all the experiments, we adopted a *5-fold cross-validation model*, i.e., we randomly split the data set into 5 folds, each containing the 20% of the patterns of the data set, 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 data set, we therefore consider the average results of 30 runs. In the case of SPEA2-FL, the averaged values are calculated considering the most accurate solution from each obtained Pareto front.

The results obtained by the considered 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 shows the average number of rules and used variables (R/V). The second and third columns show the average accuracy (MSE) in training and test data (Tra./Tst.) with their respective standard deviations (SDs).

[1]The corresponding data partitions (5-fold) for these datasets are available at the KEEL project webpage [13]: `http://sci2s.ugr.es/keel/datasets.php`

TABLE II
AVERAGE RESULTS OF THE DIFFERENT ALGORITHMS. RESULTS IN THIS TABLE (TRA./TST. AND SD) SHOULD BE MULTIPLIED BY 10^5 IN THE CASE OF BAS. GR-MF AND GA-WM WERE NOT APPLICABLE TO MV AND CA BECAUSE OF THE LARGE NUMBER OF VARIABLES AND CASES PROVOKED MEMORY OVER FLOW ERRORS.

Data set	Measure	WM(3)			WM(5)			WM(7)			GR-MF			GA-WM			SPEA2-FL		
NAME$_{(V/Size)}$		R/V	Tra.	Tst.	R/V	Tra.	Tst.	R/V	Tra.	Tst.	R/V	Tra.	Tst.	R/V	Tra.	Tst.	R/V	Tra.	Tst.
ANA$_{(7/4052)}$	Mean	72/8	0.187	0.189	124/8	0.027	0.030	171/8	0.012	0.017	148/8	0.005	0.017	150/8	**0.003**	0.008	**25/3**	0.006	**0.006**
	SD		0.001	0.005		0.000	0.002		0.000	0.003		0.001	0.008		0.001	0.005		0.000	0.001
WAN$_{(9/1609)}$	Mean	156/10	16.063	16.393	457/10	4.878	6.305	853/10	2.692	6.538	397/10	**1.406**	7.381	279/10	1.522	2.820	**12/2**	1.810	**1.823**
	SD		0.961	1.700		0.405	1.052		0.110	2.101		0.067	5.404		0.065	2.825		0.060	0.143
BAS$_{(16/337)}$	Mean	181/17	1.921	3.695	253/17	0.782	6.198	264/17	0.316	10.598	262/17	0.255	12.439	262/17	**0.202**	11.706	**33/6**	1.673	**2.575**
	SD		0.109	0.739		0.047	0.686		0.006	1.339		0.020	2.177		0.031	2.562		0.103	0.521
MV$_{(10/40768)}$	Mean	3812/11	12.404	12.620	24472/11	4.031	5.019	30616/11	1.963	24.831	-	-	-	-	-	-	**20/3**	**0.531**	**0.531**
	SD		0.245	0.228		0.027	0.0756		0.002	1.352								0.060	0.062
CA$_{(21/8192)}$	Mean	425/22	40.384	40.956	1539/22	8.449	12.440	2774/22	**5.327**	19.143	-	-	-	-	-	-	**28/5**	6.046	**6.135**
	SD		3.115	4.637		0.351	1.148		0.060	2.807								0.456	0.474

Analyzing the results shown in Table II we can highlight that SPEA2-FL obtained so simple and accurate solutions without significant overfitting, i.e., highly correlated values in training and test with respect to the other approaches. Another interesting aspect is the number of variables it is keeping in the different datasets. Thus, SPEA2-FL seems good even in the case that variables without interesting additional information are initially included in the datasets. That property makes the method scalable for high dimensional problems, in which it is still able to obtain good solutions from the point of view of the accuracy-interpretability trade-off (quite simple models with equidistributed strong fuzzy partitions). On the contrary, no values were obtained by GR-MF and GA-WM in MV and CA because of the large number of variables and cases provoked memory over flow errors after a large number of hours running without finishing the evaluation of the initial population.

With respect to the 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 only using one of the four cores). Table III shows the running times of the fast WM algorithm (*Ad-Hoc* method) and the evolutionary-based approaches. Of course WM is practically instantaneous in some of the datasets. However, it is very interesting to see that, in the case of CA and MV, a run of this simple method can take about $1/2$ and 6 minutes, respectively. It is why the cropping strategy included in SPEA2-FL is actually needed. Except for CA and MV, the proposed method is able to obtain solutions taking only some seconds. The times for CA and MV are also very good taking into account the kinds of problems they represent and the evolutionary nature of this algorithm.

In order to show the behavior of the Pareto fronts provided by SPEA2-FL, in Figure 3 we show a representative Pareto front (the results of a single trial) on CA. This figure shows the very high correlation among the values in training and test, which represents an interesting property of the proposed method.

TABLE III
AVERAGE TIMES OF A RUN — HOURS, MINUTES AND SECONDS (H:M:S)

Method	ANA	WAN	BAS	MV	CA
WM(3)	0 aprox.	0 aprox.	0 aprox.	0:00:47	0:00:18
WM(5)	0 aprox.	0 aprox.	0 aprox.	0:05:10	0:00:19
WM(7)	0 aprox.	0 aprox.	0 aprox.	0:06:12	0:00:35
GR-MF	2:58:27	1:19:12	0:13:29	-	-
GA-WM	1:46:41	1:16:59	0:16:46	-	-
FS-MGFS	0:00:50	0:00:33	0:00:22	0:17:31	0:08:28

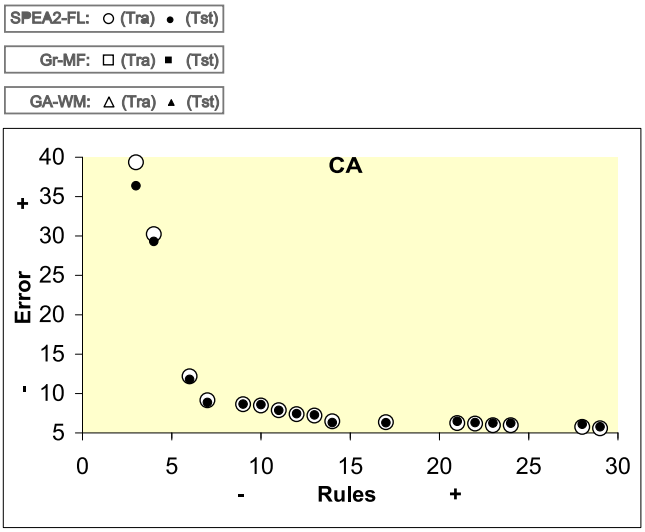

Fig. 3. Pareto front example in CA.

V. CONCLUSIONS

In this work, we have proposed an effective MOEA for the learning of linguistic KBs in high dimensional regression problems, namely SPEA2-FL. This method, based on the DB learning *a priori*, allows a slight uniform displacement of the linguistic fuzzy partitions and includes some effective

mechanisms in order to make able the derivation of simple and accurate linguistic FRBSs, in problems that are difficult to be solved by standard evolutionary methods.

We have shown that SPEA2-FL is able to obtain promising linguistic models, avoiding overfitting and keeping equidistributed strong fuzzy partitions. The scalability of SPEA2-FL is also an interesting characteristic of this method, that is able to solve problems with more than 40,000 cases or more than 20 variables in a very fast way.

ACKNOWLEDGMENT

Supported by the Spanish Ministry of Education and Science under grant no. TIN2008-06681-C06-01.

REFERENCES

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

[2] 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," *Information Sciences*, vol. 136, pp. 85–107, 2001.

[3] 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, 2001.

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

[5] R. Alcalá, J. R. Cano, O. Cordón, F. Herrera, P. Villar, and I. Zwir, "Linguistic modeling with hierarchical systems of weighted linguistic rules," *International Journal of Approximate Reasoning*, vol. 32, no. 2-3, pp. 187–215, 2003.

[6] 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.

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

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

[9] 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.

[10] 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.

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

[12] F. Herrera, M. Lozano, and A. 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.

[13] 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.