Genetic Tuning of Fuzzy Rule Deep Structures Preserving Interpretability and Its Interaction With Fuzzy Rule Set Reduction

Jorge Casillas, Oscar Cordón, María José del Jesus, and Francisco Herrera

***Abstract*—Tuning fuzzy rule-based systems for linguistic fuzzy modeling is an interesting and widely developed task. It involves adjusting some of the components of the knowledge base without completely redefining it. This contribution introduces a genetic tuning process for jointly fitting the fuzzy rule *symbolic representations* and the meaning of the involved *membership functions*. To adjust the former component, we propose the use of linguistic hedges to perform slight modifications keeping a good interpretability. To alter the latter component, two different approaches changing their basic parameters and using nonlinear scaling factors are proposed. As the accomplished experimental study shows, the good performance of our proposal mainly lies in the consideration of this tuning approach performed at two different levels of significance. The paper also analyzes the interaction of the proposed tuning method with a fuzzy rule set reduction process. A good interpretability-accuracy tradeoff is obtained combining both processes with a sequential scheme: first reducing the rule set and subsequently tuning the model.**

***Index Terms*—Complexity reduction, linguistic fuzzy modeling, linguistic hedges, surface and deep structures, tuning.**

I. Introduction

FUZZY modeling (FM), i.e., system modeling with fuzzy rule-based systems (FRBSs), may be considered as an approach used to model a system making use of a descriptive language based on fuzzy logic with fuzzy predicates. In this framework, one of the most important areas is *linguistic* FM, where the interpretability of the obtained model is the main requirement. This task is usually developed by means of linguistic FRBSs, which use fuzzy rules composed of linguistic variables [1] taking values in a term set with a real-world meaning. Thus, the linguistic fuzzy model consists of a set of linguistic descriptions regarding the behavior of the system being modeled [2].

Each of these linguistic fuzzy rules may be represented at two different levels of description by defining two different structures [3].

- *Surface structure*—It is a less specific description and involves defining the rule in its symbolic form as a relation between input and output linguistic variables. Fig. 1(a) shows the surface structure of a linguistic fuzzy rule.
- *Deep structure*—It is a more specific description and consists of the surface structure together with the definitions of the membership functions associated to the linguistic terms of the variables. Fig. 1(b) illustrates the deep structure of a linguistic fuzzy rule.

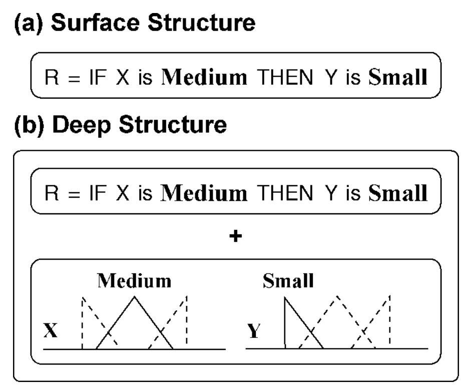


Fig. 1. Two different ways to define a linguistic fuzzy rule. (a) *Surface structure* = symbolic representation (b) *Deep structure* = symbolic representation + membership function shapes.

In a linguistic FRBS, the surface structure of each fuzzy rule is encoded in the rule base—constituted by the collection of rules in their symbolic forms—while the deep structure is contained in the knowledge base, which comprises both the rule base and the data base. The latter component contains the linguistic term sets and the membership functions defining their meanings.

One of the most important problems in the applications of fuzzy logic is the automatic derivation of these surface and deep structures from numerical information (input–output data pairs) representing the behavior of the real system. Numerous automatic methods—based on *ad-hoc* data-driven approaches [4], [5] or on different techniques such as neural networks [6], [7] and genetic algorithms (GAs) [8]—have been developed to perform this task.

A crucial problem emerges during this fuzzy model design: to obtain both an accurate and an understandable model. Indeed, FM usually comes with two contradictory requirements to the obtained model: the *interpretability*, capability to express the behavior of the real system in a comprehensible way, and the *accuracy*, capability to faithfully represent the real system. Of course, the ideal thing would be to satisfy both criteria to a high degree but, since they are contradictory issues, it is generally

Manuscript received January 27, 2001; revised April 23, 2002, February 18, 2003, and May 11, 2004. This work was supported in part by the Spanish Ministry of Science and Technology under Projects TIC2002-04036-C05-01 and TIC2003-00877, and by the European Regional Development Fund.

J. Casillas, O. Cordón, and F. Herrera are with the Department of Computer Science and Artificial Intelligence, University of Granada, Granada E-18071, Spain (e-mail: casillas@decsai.ugr.es; ocordon@decsai.ugr.es; herrera@decsai.ugr.es).

M. J. del Jesus is with the Department of Computer Science, University of Jaén, Jaén E-23071, Spain (e-mail: mjjesus@ujaen.es).

Digital Object Identifier 10.1109/TFUZZ.2004.839670

not possible. In that case, more priority is given to one of them (defined by the problem nature), leaving the other in the background.

Currently, one of the most promising research topics in FM relates with the quest of a good trade-off between interpretability and accuracy [9], [10]. This paper aims at proposing a method to automatically obtain well-balanced fuzzy models.

To do so, different mechanisms to improve the accuracy and the interpretability are used and, moreover, they are properly gathered to regulate the desired tradeoff. Thus, the surface structure is adjusted with linguistic modifiers while the rest of the deep structure is tuned by changing the membership functions. The fact of using linguistic modifiers guarantees to avoid an excessive interpretability loss since the changes they perform have associated a clear meaning. On the other hand, the membership functions are adjusted with a constrained optimization (that avoids the obtaining of excessively deformed fuzzy partitions) combined with nonlinear scaling factors that preserve the support sets of the fuzzy sets.

Additionally, the mentioned process is combined with a method to reduce the complexity of the fuzzy models by selecting the most representative fuzzy rules. This second process will significantly improve the interpretability of the obtained fuzzy models. The interaction between both processes is thoroughly studied. As will be shown, fuzzy models with a very good interpretability-accuracy tradeoff are obtained.

The paper is organized as follows. Section II shows a brief summary of different proposals to obtain a good balance between interpretability and accuracy. Section III introduces how the fuzzy rule deep structures of a linguistic fuzzy model may be tuned to improve its accuracy thus finding the desired balance. Section IV proposes a genetic tuning process to do so and shows experimental results when solving two different real-world applications. Section V analyzes the interaction of the proposed tuning method with a fuzzy rule set reduction process. Finally, Section VI outlines some conclusions. On the other hand, the Appendix A collects a description of the two real-world problems considered in the experimental studies of this paper.

II. Tuning Process as a Mechanism to Find the Interpretability-Accuracy Tradeoff

As it has been mentioned, FM faces two contradictory requirements: interpretability and accuracy. Since the improvement of one of them generally involves to worsen the other, there are two FM approaches depending on the main objective to be considered.

- *Linguistic FM*, mainly developed by linguistic (or Mamdani) FRBSs, is focused on the interpretability.
- *Precise FM*, mainly developed by Takagi–Sugeno FRBSs, is focused on the accuracy.

Regardless of the approach, a common scheme is followed to attain the desired balance between interpretability and accuracy (Fig. 2 graphically shows this operation mode).

1) First, the main objective (interpretability or accuracy) is tackled defining a specific model structure to be used, thus setting the FM approach, and deriving the model.

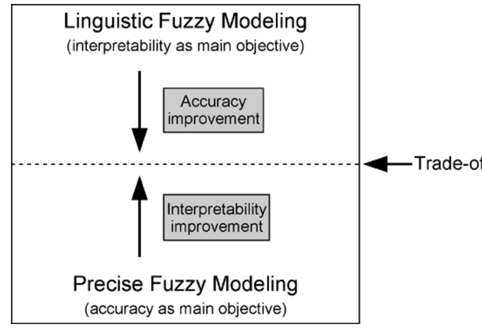


Fig. 2. Improvements of interpretability and accuracy in fuzzy modeling.

2) Then, the modeling components (model structure and/or modeling process) are improved by means of different mechanisms to compensate for the initial difference between both requirements. Thus, accuracy improvements are proposed in linguistic FM while interpretability improvements are proposed in precise FM.

Some examples found in the existing recent literature follow.

- *Linguistic FM with improved accuracy*—This approach has been performed by learning/tuning the membership functions by defining their shapes [11]–[17], their types (triangular, trapezoidal, etc.) [18], or their context (defining the whole semantic) [19], learning the granularity (number of linguistic terms) of the fuzzy partitions [20], or extending the model structure by using linguistic modifiers [14], [21], weights (importance factors for each rule) [22], or hierarchical architectures (mixing rules with different granularities) [23], among others.

  Additionally, although rule base reduction [23], [24] and input variable selection [25], [26] processes improve the interpretability, they can also be seen as accuracy improvements when redundancy and inconsistency criteria are considered.
- *Precise FM with improved interpretability*—This approach is usually developed by reducing the fuzzy rule set (usually with orthogonal transformations) [27]–[30], reducing the number of fuzzy sets (usually with similarity measures) with the subsequent merging of rules [31]–[33], reducing the number of input variables [34], or exploiting the local description of the rules (basically smoothing the consequent polynomial function of the Takagi–Sugeno rule or isolating the fuzzy rule actions) [35]–[37].

This topic, the interpretability-accuracy tradeoff, is a very important branch of research nowadays [9], [10]. Our aim in this contribution will be to attain this desired balance by increasing the accuracy of linguistic FRBSs preserving their interpretability. To achieve so, the optimization of the membership functions will be restricted and constrained nonlinear scaling factors will be used. Moreover, the rule structure will be flexibilized by adding linguistic modifiers to it. A fuzzy rule set reduction process will be also included to improve the interpretability.

III. Tuning Surface and Deep Structures

We can distinguish between two different approaches to automatically obtain a fuzzy model.

- *Learning process*—It relates to the task of directly obtaining the fuzzy rule surface [4], [38] or deep structures [32] from the available data.
- *Tuning process*—It assumes the existing of a previous definition for both structures—provided by a learning process or by experts—and adjusts them with *slight* modifications to increase the system performance.

Traditionally, the tuning process has been used to fit the deep structure by exclusively changing membership function meanings [11]–[13], [15], [16]. Besides, some proposals to develop this tuning with nonlinear scaling factors have also been introduced [14], [17]. On the other hand, recent contributions perform a tuning of the surface structures adjusting the symbolic representations [14], [21] with linguistic hedges [1].

Nevertheless, no proposals combining these different tuning approaches have been considered till now. This contribution aims at introducing a genetic method that *fully* tunes the deep structures using different ways for changing the linguistic term meanings together with the integration of linguistic hedges in the fuzzy rules.

Prior to proposing our tuning process in Section IV, this section introduces the way to adjust the FRBS by tuning its whole deep structures. Thus, the two following subsections, respectively, explain how to tune surface structures (symbolic representations) using linguistic hedges, and how to adapt the part of the deep structures related to the membership function shapes. The combined action of both processes will fully tune the fuzzy rule deep structures.

A. Tuning the Surface Structure by Using Linguistic Hedges

Certain operators may be included to slightly change the meaning of the linguistic labels involved in a specific linguistic fuzzy rule. As Zadeh highlighted in [1], a way to do so with a minor description loss is to use *linguistic hedges*.

A linguistic hedge (also known as linguistic modifier) is a function that alters the membership function of the fuzzy set associated to the linguistic label, obtaining a definition with a higher or lower precision depending on the case. Two of the most well known modifiers are the *concentration* linguistic hedge "very" and the *dilation* linguistic hedge "more-or-less." Their expressions are

$$\mu_T^{\text{very}}(x) = (\mu_T(x))^2$$
$$\mu_T^{\text{more-or-less}}(x) = \sqrt{\mu_T(x)}$$

and their effects on a triangular membership function are shown in Fig. 3.

The surface structure may be tuned by adding the mentioned linguistic hedges to the previously provided linguistic fuzzy rules, thus changing their symbolic form. For example, from the rule

IF X_1 is high and X_2 is good,
THEN Y is small

the following tuned rule would be obtained:

IF X_1 is very high and X_2 is good,
THEN Y is more-or-less small.

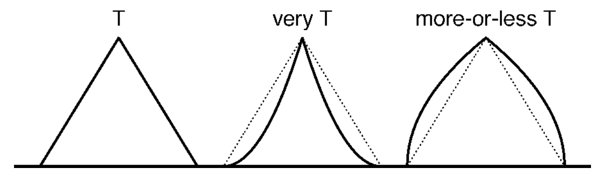


Fig. 3. Effects of the linguistic hedges "very" and "more-or-less."

Actually, this tuning approach does not define a new meaning for the so-called *primary terms—high*, *good*, and *small* in our example-but it uses them as generators whose meaning is specified in the context. In other words, thanks to the *attributed-grammar semantics* [1] involved in linguistic variables, the final membership functions are computed from the knowledge of the membership functions of the primary terms.

Of course, the fact of using linguistic hedges will have a significant influence in the FRBS performance since the matching degree of the rule antecedents as well as the output fuzzy set obtained when applying the implication in the inference process will vary.

For some proposals that perform this kind of tuning with linguistic hedges, the interested reader can refer to [14] and [21].

B. Tuning the Deep Structure by Changing the Basic Membership Functions Parameters and Using Nonlinear Scaling Factors

Tuning the deep structure, moreover of adjusting the surface structure, involves fitting the characterization of the membership functions associated to the primary linguistic terms considered in the system. Thus, the meaning of the linguistic terms is changed from a previous definition, an initial data base in an FRBS.

To change the shapes of the membership functions (i.e., the meaning of the linguistic terms), the parameters defining them must be altered. We can mainly distinguish between two different kinds of tuning approaches.

- *Changing basic parameters* [11]–[13], [15], [16]—One of the most common ways of tuning membership functions is to change their basic parameters. For example, if the following triangular-shaped membership function is considered:

$$\mu_T(x) = \begin{cases} \frac{x-a}{b-a}, & \text{if } a \leq x < b \\ \frac{c-x}{c-b}, & \text{if } b \leq x \leq c \\ 0, & \text{otherwise} \end{cases}$$

  modifying the basic parameters—a, b, and c-will vary the shape of the fuzzy set associated to the membership function [see Fig. 4(a)], thus influencing the system performance.
- *Using nonlinear scaling factors* [14], [17]—Sometimes, more flexible alternative expressions for the membership functions are considered to vary the compatibility degree to the fuzzy sets. For example, a new membership function can be obtained raising the membership value to the power of α, a positive parameter that defines a nonlinear scaling function, i.e.,

$$\mu'_T(x) = (\mu_T(x))^\alpha, \qquad \alpha \in R^+.$$

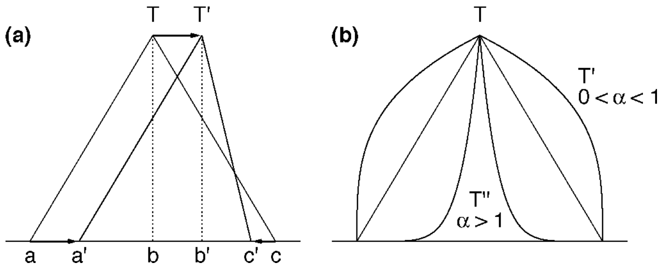


Fig. 4. Two kinds of tuning the membership function shapes. (a) Tuning by changing the basic membership function parameters. (b) Tuning by using nonlinear scaling factors.

In this case, the tuning process involves adjusting the α parameter to improve the system performance. Fig. 4(b) shows the effect of this tuning approach.

The latter approach (tuning with nonlinear scaling factors) has an important limitation with respect to the former (tuning of basic parameters): the support and core sets of the fuzzy set are not altered. Moreover, when a symmetrical fuzzy set is considered, its center of gravity is not changed. On the contrary, when tuning nonlinear scaling factors, the membership degree of a value to the fuzzy set increases in a nonlinear way as it gets closer to the core. According to this, we can assert that both tuning approaches are not incompatible but complementary.

IV. Genetic Tuning Process of Surface and Deep Structures for Linguistic Fuzzy Modeling

In this section, a tuning process based on GAs will be introduced to jointly fitting the membership functions by changing their basic and additional parameters and fitting the rule surface structure using linguistic hedges. The tuning involves starting from a previous knowledge base (rule base + data base) either derived by any learning method or provided by experts.

The following sections introduce the genetic process, its components, some dissertations about the interpretability of the tuned models, and an experimental study.

A. Genetic Process

Our proposal of FRBS genetic tuning is characterized as follows.

- The objective (fitness function) will be to minimize the well-known *mean square error* (MSE)

$$\mathrm{MSE} = \frac{1}{2 \cdot N} \sum_{l=1}^{N} \left(F(x^l) - y^l\right)^2$$

  with N being the data set size, $F(x^l)$ being the output obtained from the FRBS when the lth example is considered, and y^l being the known desired output.
- A threefold coding scheme $(CS_P + CS_A + CS_L)$ is used. CS_P will encode the basic membership function parameters, CS_A the α membership function parameters (i.e., the nonlinear scaling factors), and CS_L the linguistic hedges included in the different rules. Therefore, CS_P and CS_A are used to tune the semantics of the deep structures and CS_L to adjust the surface structures. Fig. 5 graphically shows such a scheme.
  - For the CS_P *part*, a 3-tuple of real values for each triangular membership function is used, thus being the data base encoded into a real-coded chromosome built by joining the membership functions involved in each variable fuzzy partition. A variation interval is defined for each basic parameter. It will be discussed in the next section.
  - For the CS_A *part*, a real-coded chromosome that encodes the value of the additional parameter α for each membership function is used. Each gene can take any value in the interval $[-1, 1]$ with the following mapping between alleles and actual value:

$$c_{ij}^A \in [-1, 0] \longleftrightarrow \alpha \in [\min, 1]$$
$$c_{ij}^A \in [0, 1] \longleftrightarrow \alpha \in [1, \max]$$

    and

$$\min = \frac{\log(0.5)}{\log(s_\alpha)} \quad \max = \frac{\log(0.5)}{\log(1 - s_\alpha)}$$

    with c_{ij}^A being the gene associated to the membership function for the jth linguistic term of the ith variable and $s_\alpha \in [0, 0.5]$ being a parameter that defines the flexibility degree allowed to tune the membership functions ($s_\alpha = 0$ for maximum flexibility and $s_\alpha = 0.5$ for minimum flexibility, i.e., no tuning).

    When using symmetrical triangular membership functions, this parameter can be interpreted as that the nonlinear scale factor allows membership degrees larger that 0.5 to the $s_\alpha \cdot 100\%$ (when contraction, $1 < \alpha$) or $(s_\alpha - 1) \cdot 100\%$ (when dilation, $0 < \alpha < 1$) of the corresponding support set. In this paper, we fix this value to $s_\alpha = 0.1$ [the extreme membership function shapes allowed with this value are depicted on Fig. 4(b)].
  - For the CS_L *part*, the coding scheme generates integer-coded strings of length $m \cdot (n + 1)$ (with m being the number of rules and n being the number of input variables). Each gene can take any value in the set {0,1,2} with the following correspondence to the linguistic hedge used:

$$c_{ij}^L = 0 \longleftrightarrow \text{the very linguistic hedge is used}$$
$$c_{ij}^L = 1 \longleftrightarrow \text{no linguistic hedge is used}$$
$$c_{ij}^L = 2 \longleftrightarrow \text{the more- or-less linguistic hedge is used}$$

    with c_{ij}^L being the gene associated to the linguistic term used in the jth variable of the ith rule.

Fig. 6 illustrates the proposed genetic tuning process. The GA may be used in different ways depending on the chromosome parts considered, thus performing different tuning processes. The most interesting ones are those that tune the whole deep structure and combine nonlinear scaling with membership function parameter changing, i.e., $CS_P + CS_L$ parts (PL-tun method) or $CS_P + CS_A + CS_L$ parts (PAL-tun method).

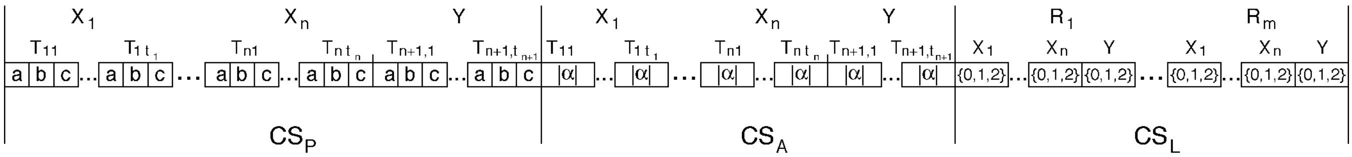


Fig. 5. Coding scheme for tuning FRBSs with n being the number of input variables, T_{ij} being the jth linguistic term of the ith variable (with $n+1$ being the output variable), t_i being the number of linguistic terms of the ith variable, and m being the number of linguistic fuzzy rules.

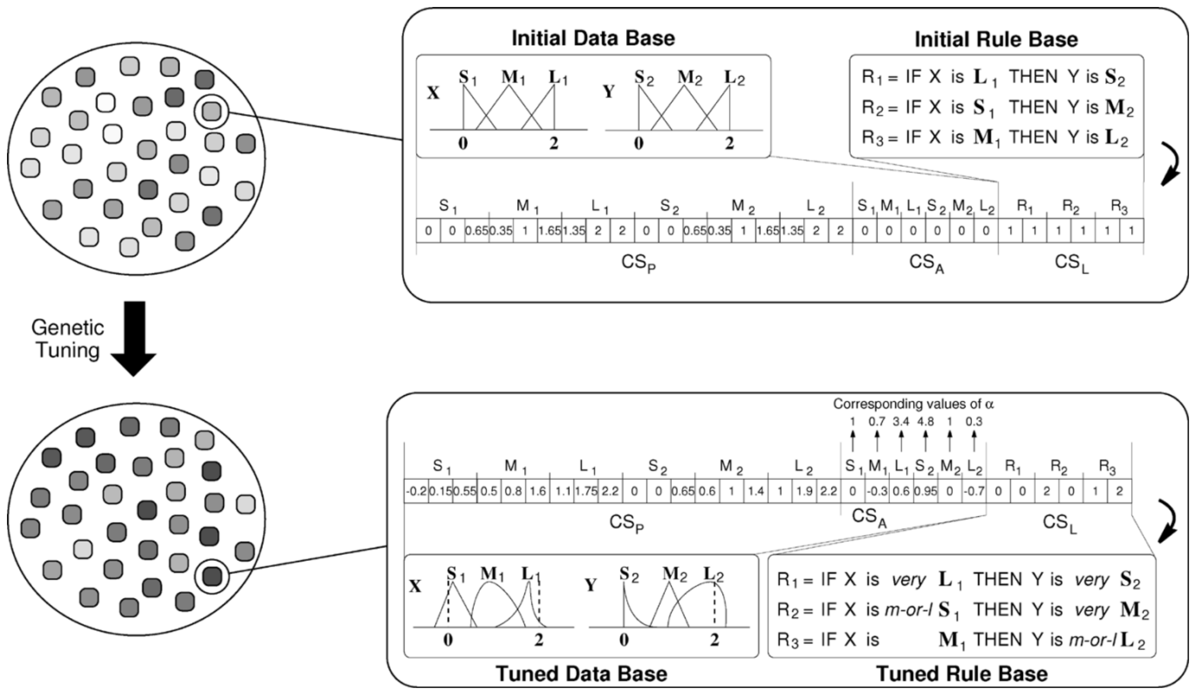


Fig. 6. Example of genetic tuning process for a single-input–single-output FRBS with three different linguistic terms for each variable and three linguistic fuzzy rules. S stands for *small*, M for *medium*, L for *large*, and m-or-l for *more-or-less*.

B. Genetic Components

The genetic tuning method has the following components.

- When generating the initial population, some of the original information in the initial knowledge base will be mixed up with random values.

  To include the original values in the CS_P part, the actual values will be directly included.

  For the CS_A part, the original values will depend on whether these parameters were used in the initial knowledge base or not. If so, the α parameters will be encoded following the said scheme; if not, the allele 0 (which means $\alpha = 1$) will be used.

  For the CS_L part, the modifiers used in the initial knowledge base are encoded with the said scheme. If no linguistic hedges were previously considered, alleles 1 will be used.

  The following four steps are considered to initialize the population.

1) A chromosome that represents the initial data base and rule base is included. Therefore, genes in the CS_P, CS_A, and CS_L parts will directly encode the values corresponding to the original knowledge base.
2) A third of the population is generated with the CS_P part at random (within the variation interval for each gene) while the alleles in CS_A and CS_L will encode the original values.
3) Another third of the population is generated with original values in CS_P, alleles at random (within the interval $[-1, 1]$) in CS_A, and original values in the CS_L part.
4) The remaining chromosomes are generated with the original values of the data base in the CS_P and CS_A parts, and alleles at random (within the set {0,1,2}) in the CS_L part.

- The crossover operator will depend on the chromosome part where it is applied.

— In the CS_P and CS_A *parts*, the max-min-arithmetical crossover [39] is considered. If $C_v^t = (c_1, \ldots, c_k, \ldots, c_H)$ and $C_w^t = (c'_1, \ldots, c'_k, \ldots, c'_H)$ are to be crossed, the following four sons are generated:

$$C_1^{t+1} = aC_w^t + (1-a)C_v^t$$
$$C_2^{t+1} = aC_v^t + (1-a)C_w^t$$
$$C_3^{t+1} \; with \; c_{3,k}^{t+1} = \min\{c_k, c'_k\}$$
$$C_4^{t+1} \; with \; c_{4,k}^{t+1} = \max\{c_k, c'_k\}.$$

The parameter $a \in [0, 0.5]$ is defined by the designer. We can notice that the variation interval of each gene will never be exceeded be-

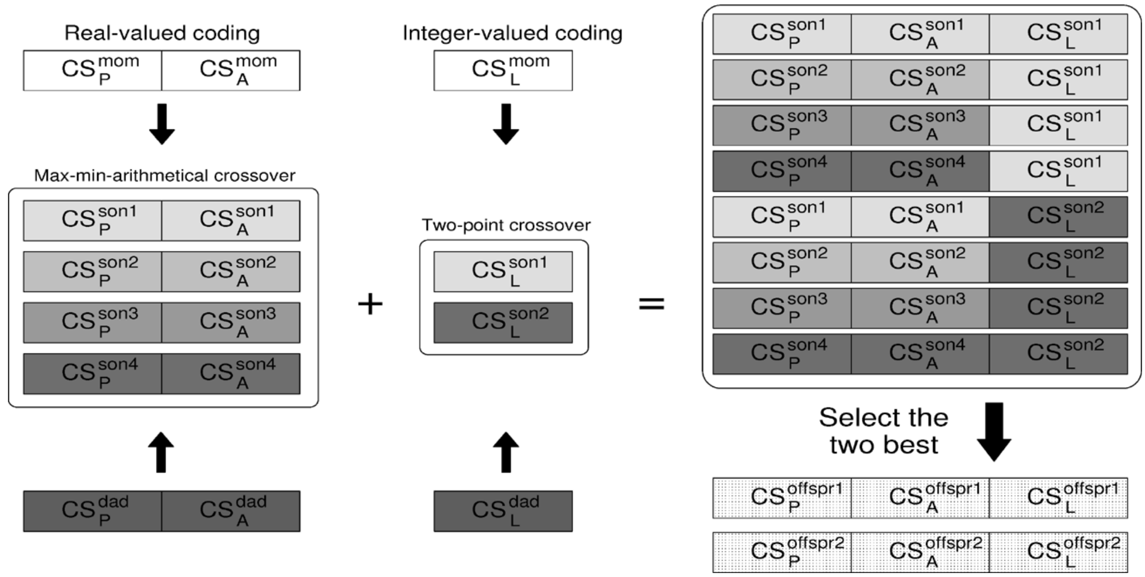


Fig. 7. Crossover operator.

cause of $\min\{c_k, c'_k\} \leq c^{t+1}_{i,k} \leq \max\{c_k, c'_k\}$, $\forall i \in \{1,2,3,4\}$.

— In the CS_L *part*, the standard two-point crossover is used.

After recombining each part, the two best chromosomes among the eight (*four* different CS_P and CS_A parts combined with *two* different CS_L parts) descendant obtained will be selected to replace their parents. Fig. 7 graphically shows it.

- The mutation operator will also depend on the chromosome part where it is applied.

— In the CS_P and CS_A *parts*, a uniform mutation operator is considered. It involves to change the value of the selected gene by other one randomly generated in the corresponding interval.

— In the CS_L *part*, the mutation operator changes the gene to the allele 1 when a gene with alleles 0 or 2 is to be mutated, and randomly to 0 or 2 when a gene with allele 1 is to be mutated.

Once a chromosome has been selected to be mutated, a randomly selected gene from each part is altered by its corresponding operator.

- A generational GA with Baker's stochastic universal sampling procedure [40] together with elitism (that ensures the selection of the best individual of the previous generation) is considered.

C. Interpretability Issues in the Proposed Tuning Method

Some issues on the interpretability preservation made by the proposed tuning method are analyzed in this section. Above all, we should remark that the tuning process is designed to make the rigid definition of linguistic fuzzy models more flexible with the aim of increasing the accuracy. The use of this structure, a far more interpretable than other formulations such as the Takagi–Sugeno one, gives to the final tuned model a good interpretability *per se*.

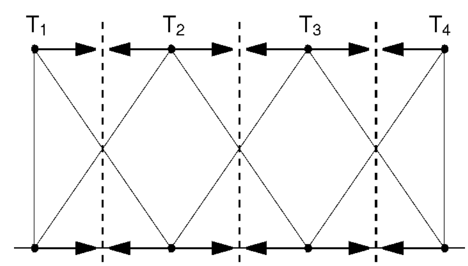


Fig. 8. Variation intervals for each membership function parameter to preserve meaningful fuzzy sets.

Nevertheless, some aspects on interpretability must be considered in order to preserve an appropriate legibility of the tuned linguistic models.

- *Comprehensibility of the membership functions*—A membership function is comprehensible insofar as the meaning of the associated linguistic term is easily understandable. Of course, this measure is highly subjective. We could say that the use of membership functions with a correct shape—meeting the constraint on that $a \leq b \leq c$—and a reasonable similarity with the initial definition (before tuning) ensures their comprehensibility.

The proposed tuning method addresses this issue constraining the optimization of every gene in the CS_P and CS_A parts. The membership function basic parameters (CS_P part) are constrained by using short variation intervals. These intervals are defined from the cross points between the initial fuzzy sets. Fig. 8 shows an example of the interval considered for each parameter.

On the other hand, the nonlinear scaling factors (CS_A part) are constrained to avoid excessively square-shaped (when $\alpha \to 0$) or singleton-shaped (when $\alpha \to \infty$) membership functions. Fig. 4(b) shows the maximum and minimum nonlinear scaling allowed by our method.

In the literature, this interpretability issue is traditionally preserved using unimodal Ruspini's (strong) fuzzy

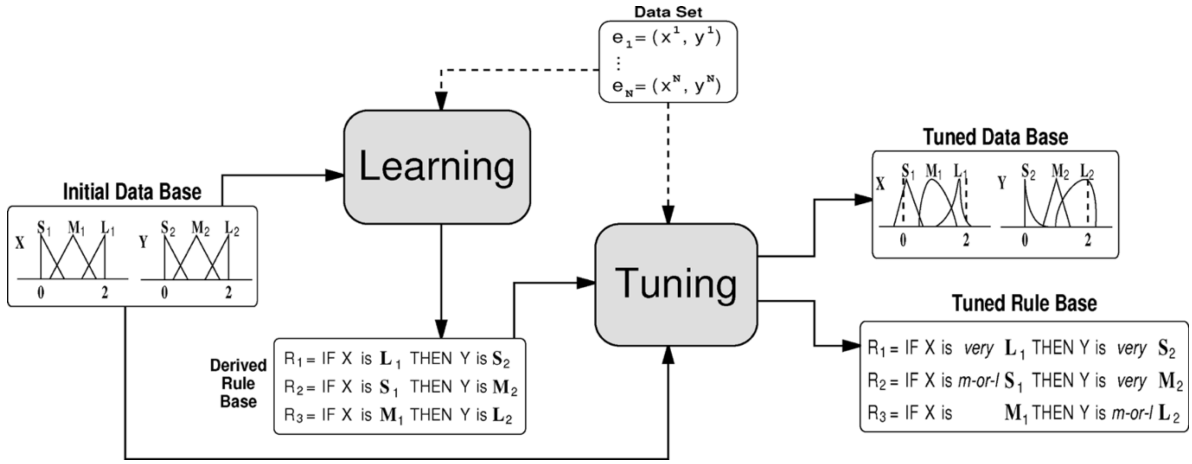


Fig. 9. Tuning process is performed in two stages. In the former one, a learning method is used to derive a rule base from an initial data base. Then, the tuning method adjusts the previously obtained rule base and/or the initial data base.

partitions [41], [42]. In this case, the sum of the membership degrees of a value to each linguistic term is always equal to one. Opposite to our proposal, this kind of partition is very inflexible thus losing accuracy ability.

Another proposal that ensures a good comprehensibility by adding the concept of conciseness (evaluated with fuzzy entropy and deviation measures) to the fitness function is proposed in [43].

- *Completeness of the deep structures*—A system is said to be complete when a matching degree greater than zero is obtained for any multidimensional input value. This issue affects to the whole fuzzy rule deep structures, i.e., the set of fuzzy rule surface structures composing the system and the fuzzy partitions of the linguistic variables.

  The proposed tuning method does not alter the completeness of the initial surface structures (defined before applying the tuning process) since the changes performed by the considered linguistic hedges ensure that the same primary terms are used and the supports of the associated fuzzy sets are not changed. Therefore, the completeness will be determined *a priori* either by the learning method used to generate the initial model or by the knowledge provided by experts.

  However, the changes of the basic membership function parameters made in the deep structures involve the chance of losing the completeness. Therefore, this case should be properly tackled by the tuning method. It may be basically addressed by penalizing bad solutions in the fitness function or by restricting the variations of the parameters. The proposed method considers this latter possibility by using the mentioned variation intervals. Other proposals that alter the fitness function to ensure a good completeness degree are introduced in [13] and [15].

- *Compactness of the surface structure set*—This is an important aspect that affects the interpretability of the linguistic fuzzy model. It involves the use of a reduced number of rules in order to make the model easily readable.

  Clearly, the proposed tuning process does not alter the number of linguistic fuzzy rules of the initial model. Nevertheless, we should point out some comments on that. Generally, the use of an excessive number of rules is caused by the need of attaining an acceptable accuracy. To do so, classical learning approaches increase the number of linguistic terms to improve the approximation to the data set, thus increasing the number of rules as well. However, the tuning method properly changes the shapes of the membership functions thus improving the accuracy. Therefore, in our case the use of an initial model with a low number of rules is recommended though it involves having a low accuracy degree initially.

  Nevertheless, a deeper analysis on the compactness is performed in Section V, where some combinations of the proposed tuning process with a fuzzy rule set reduction method are introduced.

- *Consistency of the surface structure set*—This concept relates with the lack of coherence in the definition of the surface structures when similar premise parts with different consequents are used.

  This matter is not significant in the proposed tuning method since the primary terms assigned to each linguistic fuzzy rule do not vary and, therefore, if the initial model is consistent, the tuned model will remain so.

  Of course, this is done provided that the involved membership functions do not become highly similar during the tuning process. Again, the use of the said variation intervals meets this condition when a consistent initial model is used.

D. Experimental Study of the Proposed Tuning Process

The experimental study will be focused on applying different tuning processes to a simple fuzzy model previously generated. The well-known Wang and Mendel (WM) method [4] will be used to derive initial rule bases. Therefore, the WM method will act as the learning module shown in Fig. 9, where the two-stage tuning operation mode considered in this experimental study is graphically shown.

This learning method was selected thanks to some interesting advantages that become a significant importance in our two-

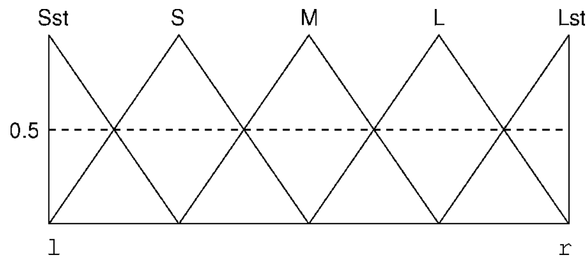


Fig. 10. Graphical representation of fuzzy partition, with Sst for *smallest*, S for *small*, M for *medium*, L for *large*, Lst for *largest*, and $[l, r]$ being the corresponding variable domain.

TABLE I
TUNING PROCESSES CONSIDERED IN THIS EXPERIMENTAL STUDY

Method	Basic m.f. parameters	α m.f. parameter	Surface structure with linguistic hedges
P-tun	✓		
A-tun		✓	
L-tun			✓
PA-tun	✓	✓	
PL-tun	✓		✓
AL-tun		✓	✓
PAL-tun	✓	✓	✓

stage design approach. On the one hand, the WM method obtains the linguistic fuzzy model quickly but this model does not usually perform properly, leaving the tuning phase in charge of improving the accuracy. On the other hand, the WM method generates the linguistic fuzzy rules from examples (with a subsequent selection to solve the inconsistencies) instead of from fuzzy input subspaces, thus obtaining a compact rule base with a reduced number of rules [44]. Since the proposed tuning method does not reduce the rule base size, this fact fits well to the design approach.

An initial data base constituted by a primary fuzzy partition for each variable is employed in the WM method and the tuning processes. Every variable domain is partitioned into a number of equally distributed triangular-shaped fuzzy sets. Fig. 10 shows an example of the fuzzy partition considered with five linguistic terms.

With the aim of performing a rigorous analysis, every possible combination of the three different tuning approaches proposed in this paper (tuning the deep structure by changing the basic membership function parameters or using nonlinear scaling factors, or tuning the surface structure by using linguistic hedges) will be considered. Table I summarizes the different tuning processes. As said, the most interesting approaches seem to be the PL-tun and PAL-tun ones.

Two real-world modeling applications (the rice and electrical problems) are considered to analyze the performance of the different tuning processes. While the rice problem shows a simple distribution and it is included just because of the free availability of the data, the electrical problem becomes an interesting modeling application that presents more difficult relations among variables. Nevertheless, this paper does not aim to show solutions to these problems (which should be faced with a deeper data treatment) but it simply use them as benchmarks.

The data and experimental setup descriptions can be consulted in Appendix A. In the following subsections, the experimental results and their analysis are introduced.

1) Experimental Results: Table II collects the results obtained by the learning and tuning methods, where $\#R$ stands for the number of rules, MSE_{tra} and MSE_{tst} for the values of the MSE over the training and test data sets respectively, and h:m:s for the mean time in hours, minutes, and seconds expended by the runs (developed on a Pentium III 1-GHz). Because of normalized values in [0,1] are considered for the output variable in the rice problem, very small errors are obtained. Hence, the table shows these results multiplied by 10 000 to facilitate their reading. The values shown for $\mathrm{MSE}_{\mathrm{tra}}$ and $\mathrm{MSE}_{\mathrm{tst}}$ are rounded to the closer integer value in the electrical problem. The arithmetic mean $(\bar{x})$ over the 30 or 60 (depending on the problem) runs performed; the standard deviation over the five or ten mean values $(\sigma_{\bar{x}_i})$, one per data partition; and the arithmetic mean of the standard deviation values over the six runs for each data partition $(\bar{\sigma_{x_i}})$ are included. The best results for both applications are shown in boldface.

While $\sigma_{\bar{x}_i}$ stands for the differences existing among the data partitions, $\bar{\sigma_{x_i}}$ stands for the differences existing among the runs for each data partition. Therefore, the former value shows the robustness of the learning/tuning method to obtain similar results regardless the data partition, while the latter value shows the robustness of the probabilistic algorithm to obtain similar results regardless the followed pseudo-random sequence.

The values of the parameters used in the genetic tuning processes are as follows: a population size of 50 individuals, 10 000 and 50 000 evaluations respectively for the rice and electrical problems, 0.6 as crossover probability, 0.2 as mutation probability per chromosome, and 0.35 for the weight factor in the max–min–arithmetical crossover (parameter a).

On the other hand, the results obtained by three different learning methods on the same data sets are shown in Table III. Although they develop a different FRBS design task than our tuning process, they can be useful as a measure of the performance achieved by our proposal. The first method, proposed by Nozaki *et al.* in [5], uses linguistic fuzzy rules with double consequents and weights associated to them, moreover of considering an additional membership function parameter α to perform a nonlinear scaling over the membership functions. The second one, proposed by Thrift in [38], is a basic GA-based learning method that only defines the fuzzy rule surface structures. Finally, the third process, proposed by Liska and Melsheimer in [45], is a sophisticated learning method based on two stages that firstly designs the whole deep structures with a GA-based process and then performs a final tuning process.

As regards the values of parameters used in the Nozaki *et al.*'s method, the best results were obtained with $\alpha = 5$ in both problems, which are the results shown in the table. In the Thrift's method, a population size of 50 individuals, 10 000 and 50 000 evaluations for the rice and electrical problems respectively, 0.6 as crossover probability, and 0.2 as mutation probability per chromosome were used. In the Liska and Melsheimer's method, 50 individuals, 10 000 (rice) or 50 000 (electrical) evaluations, 32 (rice) or 625 (electrical) as maximum number of rules, 0.6 as

TABLE II
RESULTS OBTAINED BY THE PROPOSED TUNING METHODS ON THE INITIAL MODELS GENERATED BY THE WM METHOD

	Rice Problem									
	$\bar{x}$				$\sigma_{\bar{x}_i}$			$\overline{\sigma_{x_i}}$		
Method	**#R**	**MSE$^*_{tra}$**	**MSE$^*_{tst}$**	**h:m:s**	**#R**	**MSE$^*_{tra}$**	**MSE$^*_{tst}$**	**#R**	**MSE$^*_{tra}$**	**MSE$^*_{tst}$**
WM	**15**	132.84	131.19	0:00:00	**0.6**	59.87	42.39	—	—	—
WM+P-tun	15	12.88	21.26	**0:00:22**	0.6	2.12	4.22	—	1.21	2.59
WM+A-tun	15	24.97	22.80	0:00:40	0.6	5.28	2.81	—	1.33	3.56
WM+L-tun	15	18.10	**16.66**	0:00:26	0.6	5.59	3.73	—	**0.44**	**1.88**
WM+PA-tun	15	12.52	22.70	0:00:40	0.6	1.72	2.24	—	1.29	2.95
WM+PL-tun	15	**8.60**	17.75	0:00:26	0.6	0.70	2.74	—	0.85	2.63
WM+AL-tun	15	19.10	20.52	0:00:40	0.6	4.87	3.75	—	0.60	2.02
WM+PAL-tun	15	9.06	19.44	0:00:40	0.6	**0.58**	**1.99**	—	0.70	2.69

* multiplied by 10,000

	Electrical Problem									
	$\bar{x}$				$\sigma_{\bar{x}_i}$			$\overline{\sigma_{x_i}}$		
Method	**#R**	**MSE$_{tra}$**	**MSE$_{tst}$**	**h:m:s**	**#R**	**MSE$_{tra}$**	**MSE$_{tst}$**	**#R**	**MSE$_{tra}$**	**MSE$_{tst}$**
WM	**65**	56,135	56,359	0:00:00	**0.0**	1,498	4,685	—	—	—
WM+P-tun	65	18,395	22,136	**0:22:41**	0.0	778	3,200	**—**	1,110	1,988
WM+A-tun	65	37,243	38,837	0:33:58	0.0	455	1,816	**—**	**125**	**572**
WM+L-tun	65	20,967	23,420	0:25:16	0.0	632	3,207	—	336	1,439
WM+PA-tun	65	17,967	21,377	0:38:02	0.0	1,078	1,625	—	2,133	2,628
WM+PL-tun	65	**9,617**	**13,519**	0:25:33	0.0	**263**	3,153	**—**	694	1,509
WM+AL-tun	65	20,544	23,207	0:34:55	0.0	834	2,701	—	797	1,430
WM+PAL-tun	65	11,222	14,741	0:38:12	0.0	380	**1,315**	—	801	2,136

TABLE III
RESULTS OBTAINED BY OTHER METHODS

	Rice Problem									
	$\bar{x}$				$\sigma_{\bar{x}_i}$			$\overline{\sigma_{x_i}}$		
Method	**#R**	**MSE$^*_{tra}$**	**MSE$^*_{tst}$**	**h:m:s**	**#R**	**MSE$^*_{tra}$**	**MSE$^*_{tst}$**	**#R**	**MSE$^*_{tra}$**	**MSE$^*_{tst}$**
Nozaki [5]	64	**27.41**	**32.01**	**0:00:00**	**0.0**	3.63	**5.62**	**—**	**—**	**—**
Thrift [38]	**18.2**	53.10	63.55	0:00:18	1.2	3.68	11.06	1.8	**2.74**	**5.27**
Liska [45]	32	30.80	57.52	0:00:50	**0.0**	**3.37**	16.58	**0.0**	4.23	9.80

* multiplied by 10,000

	Electrical Problem									
	$\bar{x}$				$\sigma_{\bar{x}_i}$			$\overline{\sigma_{x_i}}$		
Method	**#R**	**MSE$_{tra}$**	**MSE$_{tst}$**	**h:m:s**	**#R**	**MSE$_{tra}$**	**MSE$_{tst}$**	**#R**	**MSE$_{tra}$**	**MSE$_{tst}$**
Nozaki [5]	**532**	**26,705**	**27,710**	**0:00:00**	**0.0**	**764**	**2,906**	—	—	—
Thrift [38]	565.3	31,228	37,579	3:13:25	2.6	1,018	7,279	6.1	**2,110**	**3,609**
Liska [45]	624.9	49,263	56,089	7:13:34	0.1	2,356	4,628	**0.1**	7,522	11,191

crossover probability, 0.1 as mutation probability per chromosome, and 0.1 as creep probability were used.

2) Analysis of Results: First of all, in view of the obtained results, we may notice that all tuning processes significantly improve the accuracy—both in approximation $(\mathrm{MSE}_{\mathrm{tra}})$ and prediction $(\mathrm{MSE}_{\mathrm{tst}})$—of the models derived by the WM learning method.

In a general view, we should say that when a full tuning of deep structures, i.e., including the surface ones, is performed and membership function parameter and nonlinear scaling tuning are combined (as our PL-tun and PAL-tun methods do), significantly more accurate fuzzy models are obtained.

It is particularly notable the excellent results generated by the PL-tun method that combines macroscopic and microscopic tuning effects [41] with two ways of changing the membership function shapes. This method obtains the best results in the $\mathrm{MSE}_{\mathrm{tra}} - \mathrm{MSE}_{\mathrm{tst}}$ balance among all the methods analyzed in the experimental study. Fig. 11 shows the tuned data base and rule base obtained by the PL-tun method for a run on a specific data set partition of the rice problem.

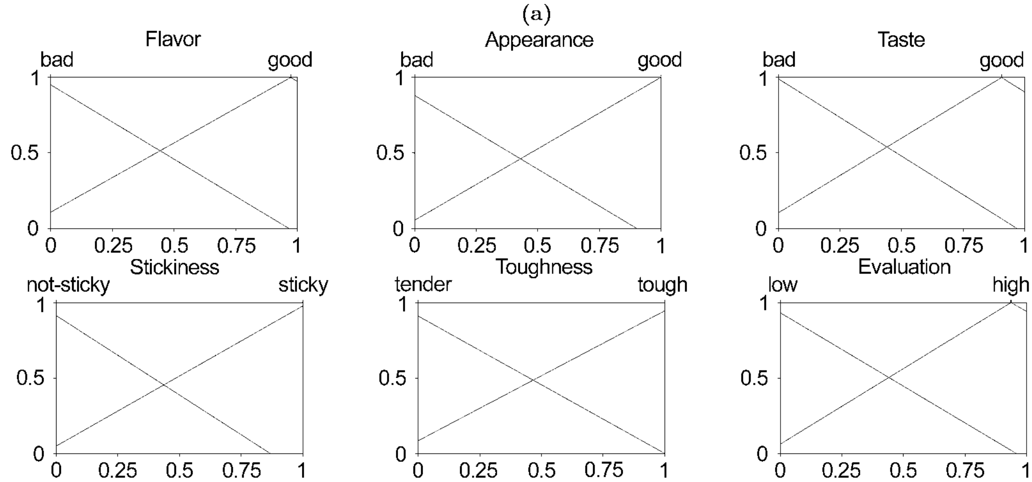
On the other hand, focusing on the P-tun and A-tun methods, the former clearly obtains more accurate models. We can deduce from these results that the fact of varying the support sets and moving the center of gravity of the fuzzy sets in the P-tun method (which is not possible in the A-tun method) allows the obtained linguistic fuzzy model to approximate better the training data set.

The L-tun method solves this drawback to a certain extent. Although, all in all, this method makes similar changes to the membership function shapes than the A-tun one, it has a better approximation capability thanks to perform them in the surface structures. Thus, the L-tun method overcomes the A-tun one and shows a better tradeoff between approximation and prediction than the P-tun method.

When two tuning approaches are combined, only the PL-tun method obtains significant improvements over the corresponding tuning methods under separate cover.

We can see that PA-tun and AL-tun methods do not obtain significant improvements. In the PA-tun case, even when both components perform a different tuning process, the membership

(a)



(b)

Rule	*Flavor*		*Appearance*		*Taste*		*Stickiness*		*Toughness*		*Evaluation*	
R_1	*very*	good	*m-or-l*	good	*very*	good	*m-or-l*	sticky		tender	*m-or-l*	high
R_2		good	*m-or-l*	bad	*very*	bad		not-sticky	*m-or-l*	tough	*very*	low
R_3	*m-or-l*	good	*m-or-l*	good	*m-or-l*	bad	*very*	not-sticky		tough	*very*	low
R_4		good		good		good		not-sticky		tender		high
R_5		good		good		good	*very*	not-sticky		tough		high
R_6	*m-or-l*	bad	*very*	bad	*very*	bad	*m-or-l*	not-sticky	*m-or-l*	tender		low
R_7		good	*very*	good	*very*	bad	*very*	not-sticky		tender		high
R_8	*very*	bad		good		good		sticky	*m-or-l*	tender	*m-or-l*	low
R_9	*very*	good	*m-or-l*	bad	*very*	bad	*m-or-l*	not-sticky	*very*	tender		low
R_{10}	*m-or-l*	bad		bad	*very*	bad	*m-or-l*	not-sticky	*very*	tough	*m-or-l*	low
R_{11}	*m-or-l*	good	*very*	good	*m-or-l*	bad	*very*	sticky	*very*	tender	*m-or-l*	high
R_{12}		good	*m-or-l*	bad		good	*very*	not-sticky	*very*	tender	*m-or-l*	low
R_{13}	*m-or-l*	bad		good	*m-or-l*	good	*very*	not-sticky	*very*	tender		low
R_{14}	*m-or-l*	good	*very*	good	*very*	good		sticky		tough		high
R_{15}	*m-or-l*	bad		good	*very*	bad		not-sticky		tender	*very*	low

Fig. 11. Knowledge base (fuzzy rule deep structures) generated by the WM + PL-tun method for a specific run and data set partition on the rice problem. $\mathrm{MSE}_{\mathtt{tra/tst}} \cdot 10\,000$ before tuning $= 147.04/167.00, \mathrm{MSE}_{\mathtt{tra/tst}} \cdot 10{,}000$ after tuning $= 10.26/13.81$. *m-or-l* stands for "more-or-less." (a) Tuned data base. (b) Tuned rule base (fuzzy rule surface structures).

function parameters tuning seems to overshadow the action of the nonlinear scaling factors. In the AL-tun case, the effects performed by both tuning approaches are redundant.

As for the behavior of the PAL-tun method, which groups the three tuning approaches regarded, we may notice that it obtains good approximation and prediction degrees, but worsen than the PL-tun method. Although the former method includes to the latter one, it seems that the slight improvement achieved by the CS_A part does not compensate for the search space size increasing. As regards the interpretability of the models obtained by this method, in Fig. 12 we can see how the combined use of constrained basic and additional membership function parameters does not disturb the legibility of the data base in the electrical problem.

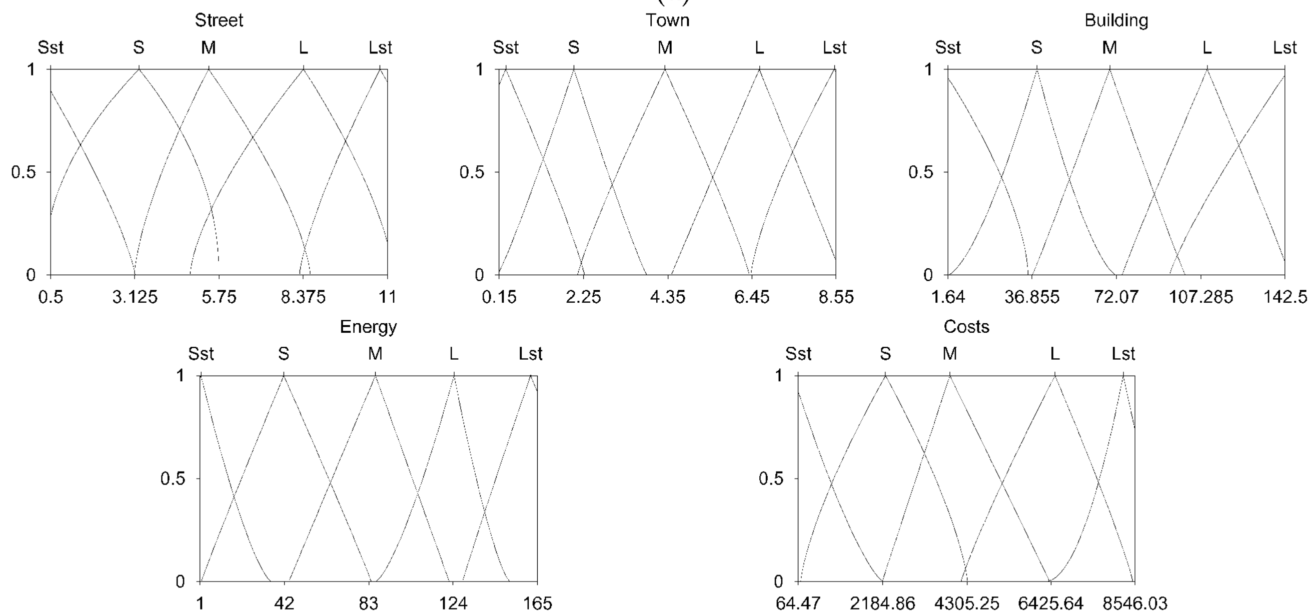
Comparing our results with the methods shown in Table III, we may observe that the proposed tuning methods significantly overcome the others, and models with better approximation and prediction degrees are obtained. On the other hand, the three compared methods obtain an excessive number of rules that seriously disturbs the interpretability of the generated models. This fact is caused by three different reasons (depending on the method) that are solved by our tuning proposal. The method of Nozaki *et al.* makes use of a more complex rule structure (with double consequents and weights) to obtain a good accuracy. Of course, this approach involves significantly increasing the number of rules. The method proposed by Thrift only designs the fuzzy rule surface structures and keeps unaltered the membership functions. Thus, a large number of rules is needed to obtain a high accuracy. Liska and Melsheimer's method performs a simultaneous learning of the surface structures and the basic membership functions parameters of the deep structures. This approach tremendously increases the tackled search space, thus making more difficult to obtain good solutions.

Finally, we may observe that the fact of performing a tuning over a compact initial fuzzy rule set allow these methods to be quicker than other approaches, like the Thrift and Liska's and Melsheimer's methods, even using the same number of evaluations in all the experiments. This is due to the inference process of the whole training data, which becomes the main bottleneck in fuzzy modeling, is quicker with a low number of rules. On the other hand, the increasing of computing time performed by A-tun, PA-tun, AL-tun, and PAL-tun methods with respect to P-tun, L-tun, and PL-tun methods is due to the expensive cost of computing real-valued power rising operations.

V. Interaction of the Proposed Tuning Process With Fuzzy Rule Set Reduction Processes

Once the tuning process has been presented, analyzed, and tested, it seems interesting to study its interaction with other well-known postprocessing methods used to refine initial fuzzy models: *fuzzy rule set reduction processes*. This task, also

(a)



(b)

Rule	*Street*		*Town*		*Building*		*Energy*		*Costs*	
R_{1}	*very*	Sst		Sst	*very*	Sst	*very*	Sst		Sst
R_{2}	*m-or-l*	Sst	*very*	Sst		Sst	*very*	S	*m-or-l*	Sst
R_{3}	*m-or-l*	Sst	*m-or-l*	S	*very*	Sst	*very*	Sst	*very*	Sst
R_{4}	*very*	Sst	*m-or-l*	S		Sst	*very*	S	*m-or-l*	S
R_{5}	*very*	Sst	*very*	S		S	*very*	Sst	*m-or-l*	Sst
R_{6}		Sst	*m-or-l*	S		S	*very*	S	*very*	S
R_{7}		S		Sst	*very*	Sst	*m-or-l*	Sst	*very*	Sst
R_{8}	*m-or-l*	S		Sst		Sst		S	*m-or-l*	S
R_{9}	*very*	S		Sst		S		Sst		S
R_{10}	*m-or-l*	S		Sst		S		S		S
R_{11}		S		S	*m-or-l*	Sst	*very*	Sst	*very*	Sst
R_{12}		S	*very*	S		Sst	*very*	S		S
R_{13}	*m-or-l*	S	*m-or-l*	S		S	*m-or-l*	Sst	*very*	S
R_{14}		S	*very*	S		S	*very*	S		S
R_{15}		S	*m-or-l*	M	*very*	S	*m-or-l*	Sst	*m-or-l*	S
R_{16}	*very*	S	*very*	M	*very*	S		S	*m-or-l*	S
R_{17}	*m-or-l*	S	*m-or-l*	M		M		Sst		M
R_{18}		S	*m-or-l*	M	*m-or-l*	M		S		M
R_{19}		M		S	*m-or-l*	Sst		Sst		Sst
R_{20}		M		S		Sst	*very*	S		S
R_{21}	*m-or-l*	M	*very*	S		Sst		M		S
R_{22}		M		S	*very*	S		Sst	*m-or-l*	S
R_{23}	*very*	M	*very*	S	*very*	S	*very*	S		S
R_{24}	*very*	M		S		S		M		M
R_{25}	*very*	M	*very*	M		S	*m-or-l*	Sst		S
R_{26}		M	*very*	M		S	*m-or-l*	S		S
R_{27}	*m-or-l*	M	*very*	M	*m-or-l*	S	*m-or-l*	M		M
R_{28}	*very*	M		M		M	*very*	S		M
R_{29}		M		M	*very*	M	*very*	M		M
R_{30}		M	*very*	L	*very*	M		S		M
R_{31}		M	*very*	L		M	*m-or-l*	M		M
R_{32}	*very*	M		L		L	*very*	S		M
R_{33}		M		L		L		M		L

Rule	*Street*		*Town*		*Building*		*Energy*		*Costs*	
R_{34}	*very*	L	*very*	S		S		Sst	*m-or-l*	S
R_{35}		L		S	*m-or-l*	S	*m-or-l*	S		S
R_{36}	*very*	L		S	*very*	S		M		M
R_{37}		L	*very*	S		S	*very*	L		M
R_{38}		L		M	*m-or-l*	S		Sst		S
R_{39}		L		M	*very*	S	*very*	S		M
R_{40}	*very*	L		M	*m-or-l*	S		M		M
R_{41}		L		M		S		L		M
R_{42}		L	*m-or-l*	M		M	*very*	S		M
R_{43}	*m-or-l*	L	*very*	M	*very*	M		M		L
R_{44}		L		M	*very*	M		L		L
R_{45}		L	*very*	L		M	*very*	Sst	*m-or-l*	M
R_{46}	*m-or-l*	L	*very*	L		M	*m-or-l*	S		M
R_{47}		L	*very*	L	*m-or-l*	M		M		L
R_{48}	*very*	L		L		M	*very*	L		L
R_{49}	*m-or-l*	L		L		L		S		L
R_{50}		L	*very*	L		L		M		L
R_{51}		L		L		L	*m-or-l*	L		Lst
R_{52}		L		Lst		L		S	*very*	M
R_{53}		L		Lst		L		M		L
R_{54}		L		Lst	*very*	L	*very*	L	*very*	Lst
R_{55}		L	*m-or-l*	Lst	*m-or-l*	Lst	*m-or-l*	S		Lst
R_{56}	*m-or-l*	L	*m-or-l*	Lst		Lst	*m-or-l*	M		Lst
R_{57}	*m-or-l*	Lst	*very*	S	*m-or-l*	S		S	*very*	S
R_{58}	*m-or-l*	Lst	*very*	S		S	*very*	L	*m-or-l*	M
R_{59}		Lst		S	*very*	S		Lst	*very*	L
R_{60}		Lst		S	*m-or-l*	M	*m-or-l*	S	*m-or-l*	M
R_{61}	*m-or-l*	Lst		S	*m-or-l*	M	*very*	L	*very*	M
R_{62}	*very*	Lst	*m-or-l*	S		M	*m-or-l*	Lst		L
R_{63}	*m-or-l*	Lst	*m-or-l*	L		M	*m-or-l*	S		M
R_{64}	*m-or-l*	Lst		L	*very*	M	*very*	L		L
R_{65}	*very*	Lst		L	*m-or-l*	M	*very*	Lst	*very*	Lst

Fig. 12. Knowledge base (fuzzy rule deep structures) generated by the WM + PAL-tun method for a specific run and data set partition on the electrical problem. $\mathrm{MSE}_{\mathtt{tra/tst}}$ before Tuning $= 58\,032/55\,150$, $\mathrm{MSE}_{\mathtt{tra/tst}}$ after Tuning $= 11\,395/14\,465$. *Street* stands for *street lengths, Town* for *town area, Building* for *building area, Energy* for *energy supply*, and *Costs* for *maintenance costs*. *Sst* stands for *smallest*, *S* for *small*, *M* for *medium*, *L* for *large*, *Lst* for *largest*, and *m-or-l* for "more-or-less." (a) Tuned data base. (b) Tuned rule base (fuzzy rule surface structures).

known as simplification or compaction, involves the decrease of the number of fuzzy rules with the main objective of improving the interpretability.

Thus, this section is devoted to perform an experimental study on the combination of both postprocessing mechanisms: first tuning, then reduction; first reduction, then tuning; and tuning and reduction simultaneously. To do so, the first subsection introduces a brief revision of the literature related to fuzzy rule set reduction; the second one presents two GA-based processes to perform the reduction alone or in combination with the tuning; and finally, the third subsection shows and analyzes some experiments.

A. Fuzzy Rule Set Reduction Process

Sometimes, a large number of fuzzy rules must be used to reach an acceptable accuracy degree. However, this effect is often caused by a deficient fuzzy rule set learning process (sometimes advisedly) with tendency to generate too many rules. Besides worsening the accuracy, an excessive number of rules makes difficult to understand the model behavior.

Thus, we may find different kinds of rules in a fuzzy rule set: *irrelevant rules*, which do not contain significant information; *redundant rules*, whose actions are covered by other rules; *erroneous rules*, which are wrong defined and distort the FRBS performance; and *conflictive rules*, which perturb the FRBS performance when they coexist with others.

To face this problem, a fuzzy rule set reduction process can be developed to achieve the goal of minimizing the number of rules used while maintaining (or even improving) the FRBS performance. Fuzzy rule set reduction is generally applied as a post-processing stage, once an initial fuzzy rule set has been derived.

We may distinguish between two approaches to obtain a *compact* fuzzy rule set.

- *Selecting fuzzy rules*—It involves obtaining an optimized subset of rules from a previous fuzzy rule set by selecting some of them.

  We may find several methods to do so with different search algorithms that look for the most successful combination of fuzzy rules [23], [46]–[48]. In [49], an interesting heuristic rule selection procedure is proposed where, by means of statistical measures, a relevance factor is computed for each fuzzy rule composing the linguistic FRBSs to subsequently select the most relevant ones.

  The philosophy of ordering the rules with respect to an importance criterion and selecting a subset of them seems similar to the orthogonal transformation-methods considered by Takagi–Sugeno-type FRBSs [27]–[29]. This mechanism is used to give an importance degree to each fuzzy rule, thus obtaining a ranking of them. Once they have been sorted, the selection is achieved using only the most promising ones.
- *Merging fuzzy rules*—It is an alternative approach that reduces the fuzzy rule set by merging the existing rules.

  In [50], the authors propose to merge neighboring rules, i.e., fuzzy rules where the linguistic terms used by the same variable in each rule are adjacent. The merge is performed in three different ways: using a new fuzzy set that groups the adjacent linguistic terms, merging the adjacent fuzzy sets if they are very similar, or giving the set of rules in disjunctive normal form. Another proposal is presented in [24], where a special consideration to the merging order is made.

  In Takagi–Sugeno-type FRBSs, processes that simplify the fuzzy models by merging fuzzy rules have also been proposed [31]–[33]. They consists of two steps. First, those fuzzy sets with a high degree of similarity are merged to compose a unique fuzzy set that represent the collection of similar fuzzy sets and, on the other hand, irrelevant fuzzy sets are removed. Then, because of the fuzzy set reduction results in rules with equal antecedents, they are merged.

B. Combination of Fuzzy Rule Set Reduction and the Proposed Tuning

This section proposes two different ways of combining a reduction process with our tuning method: sequential and simultaneously. In the sequential combination, the reduction process acts independently of the tuning and simply takes a specific fuzzy model and refines it by reducing its initial number of rules by selecting a subset of them with good performance. In the simultaneous approach, the reduction process is integrated within the proposed tuning method with the aim of considering the interdependence existing between them. The following two subsections describe both methods.

1) Independent Reduction Process Considered: The following GA-based reduction process is considered.

- The objective (fitness function) will be to minimize the MSE.
- A binary coding scheme is used. Each gene, one per rule, can take the allele 0 (which means that the corresponding rule will not be used) or 1 (otherwise).
- To generate the initial pool, one chromosome with all the genes taking alleles 1 is generated, while the remaining chromosomes are randomly generated.
- The standard two-point crossover is used.
- The mutation operator changes the gene to the allele 1 when a gene with allele 0 must be mutated and *vice versa*.
- A generational GA with Baker's stochastic universal sampling procedure together with elitism (that ensures to select the best individual of the previous generation) is considered.

As we can see, with the fitness function considered previously, our reduction method mainly searches for improving the accuracy of the fuzzy model. Therefore, the method will remove those rules that worsen the performance of the model, i.e., *erroneous* and *conflictive* rules. Of course, the interpretability is indirectly improved.

Another possibility could be to consider some interpretability criteria in the fitness function or pool selection process. However, we think that an accuracy-oriented approach will interact better with the tuning method. Thus, *irrelevant* and *redundant* rules will not be directly removed by the reduction method since they could be used by the tuning process to improve the accuracy.

2) Simultaneous Tuning and Reduction Processes: From the algorithm defined in Section IV, the following changes must be performed to integrate the reduction process within the proposed tuning method.

- A fourth part CS_R is included in the coding scheme. It is a binary representation of length m (number of initial fuzzy rules). The allele 1 means that the corresponding rule is used, while 0 means that not.
- Initial pool: For the chromosome containing the initial knowledge base, alleles 1 are used in the CS_R part. The remaining chromosomes are divided into four groups. Each group contains one of the four parts of the coding scheme with the original values and the other three parts with random values.
- Crossover operator: The standard two-point crossover is used in the binary-valued CS_R part. Each son is joined to one of the sons generated from the CS_L part. After that, as shown in Fig. 7, for the case of three different parts, these two assembled sons are combined with the four real-valued sons obtained from the CS_P and CS_A

TABLE IV
RESULTS OBTAINED WHEN COMBINING TUNING AND REDUCTION PROCESSES

	Rice Problem									
	$\bar{x}$				$\sigma_{\bar{x}_i}$			$\bar{\sigma}_{x_i}$		
Method	**#R**	**MSE** $^*_{tra}$	**MSE** $^*_{tst}$	**h:m:s**	**#R**	**MSE** $^*_{tra}$	**MSE** $^*_{tst}$	**#R**	**MSE** $^*_{tra}$	**MSE** $^*_{tst}$
WM	15	132.84	131.19	0:00:00	**0.6**	59.87	42.39	—	—	—
WM+PL-tun	15	**8.60**	**17.75**	0:00:26	0.6	0.70	2.74	—	0.85	**2.63**
WM+PL-tun+Reduction	15	8.60	17.75	0:00:28	0.6	0.70	2.74	**0.0**	0.85	2.63
WM+PAL-tun	15	9.06	19.44	0:00:40	0.6	0.58	**1.99**	—	0.70	2.69
WM+PAL-tun+Reduction	15	9.05	19.45	0:00:43	0.6	0.57	2.01	0.1	**0.70**	2.68
WM+Reduction	**5.7**	40.92	44.60	0:00:01	1.1	3.79	9.00	1.1	2.74	6.16
WM+Reduction+PL-tun	5.7	10.35	24.12	**0:00:11**	1.1	1.06	7.36	1.1	1.06	6.23
WM+Reduction+PAL-tun	5.7	10.73	26.71	0:00:16	1.1	**0.49**	6.71	1.1	1.08	5.20
WM+(PL-tun&Reduction)	12.4	10.18	20.78	0:00:22	1.1	0.98	5.10	1.2	1.09	3.76
WM+(PAL-tun&Reduction)	12.3	10.46	21.51	0:00:37	0.8	0.86	2.87	1.5	1.12	3.27

* multiplied by 10,000

	Electrical Problem									
	$\bar{x}$				$\sigma_{\bar{x}_i}$			$\bar{\sigma}_{x_i}$		
Method	**#R**	**MSE** $_{tra}$	**MSE** $_{tst}$	**h:m:s**	**#R**	**MSE** $_{tra}$	**MSE** $_{tst}$	**#R**	**MSE** $_{tra}$	**MSE** $_{tst}$
WM	65	56,135	56,359	0:00:00	**0.0**	1,498	4,685	—	—	—
WM+PL-tun	65	9,617	13,519	0:25:33	0.0	263	3,153	**—**	694	1,509
WM+PL-tun+Reduction	64.9	**9,612**	**13,499**	0:30:36	0.1	259	3,149	**0.1**	689	1,540
WM+PAL-tun	65	11,222	14,741	0:38:12	0.0	380	1,315	—	801	2,136
WM+PAL-tun+Reduction	64.9	11,216	14,758	0:45:35	0.1	376	**1,311**	0.3	791	2,158
WM+Reduction	**42.7**	42,192	44,709	0:02:55	1.0	730	4,722	2.0	**465**	**1,256**
WM+Reduction+PL-tun	42.7	10,419	13,783	**0:19:27**	1.0	468	2,433	2.0	1,188	1,499
WM+Reduction+PAL-tun	42.7	10,993	14,564	0:26:48	1.0	**229**	2,549	2.0	840	2,101
WM+(PL-tun&Reduction)	57.2	11,690	15,394	0:27:55	0.4	416	2,014	3.5	827	1,515
WM+(PAL-tun&Reduction)	55.9	12,597	16,461	0:41:37	1.0	429	2,221	3.8	802	2,050

parts. The two best sons from the eight combinations are selected as offspring to replace their parents.

- Mutation operator: In the CS_R part, the operator changes the gene to the allele 1 when a gene with allele 0 must be mutated and *vice versa*.

C. Experimental Study of the Combination Tuning-Reduction

This section analyzes the behavior of combining the reduction process introduced in the previous section with our tuning method. To do so, three different possibilities have been considered: first, tune the initial fuzzy model and then reduce it; first, reduce the initial fuzzy model and then tune it; and jointly tune and reduce the initial fuzzy model. The results obtained and an analysis of them are shown as follows.

1) Experimental Results: Table IV collects the obtained results when combining tuning and reduction processes. PL-tun and PAL-tun methods have been selected as tuning processes for this experiment. The results of these methods and the initial fuzzy model (generated by the WM method) are also shown.

The values of parameters used are the following: 50 individuals, 0.6 as crossover probability, 0.2 as mutation probability per chromosome, and 0.35 for the weight factor in the max–min–arithmetical crossover (parameter a). With regard to the number of evaluations, it depends of the application and the method: 10 000 (rice problem) or 50 000 (electrical problem) evaluations for independent tuning; 1,000 (rice) or 10 000 (electrical) evaluations for independent reduction; and 11 000 (rice) or 60 000 (electrical) evaluations for simultaneous tuning and reduction.

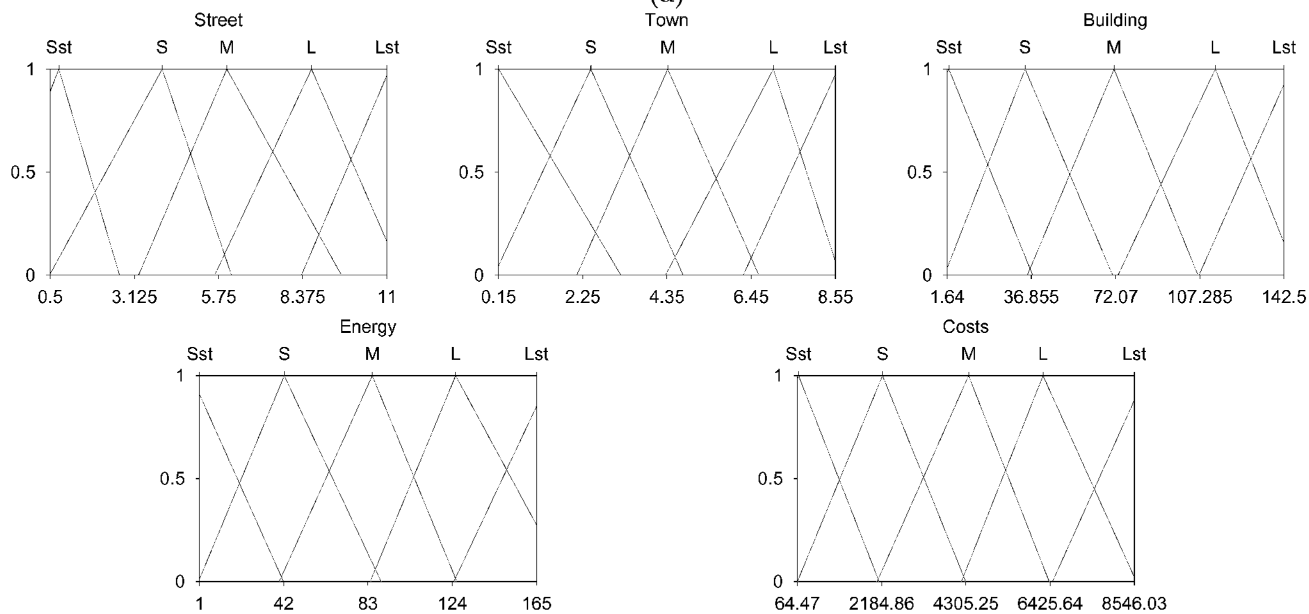
2) Analysis of Results: From the obtained results, we can see how tuning and reduction processes can significantly improve the accuracy of a fuzzy model. It is also interesting to verify that the order of performing both processes is a crucial question.

When reduction is applied after tuning (WM + Tuning + Reduction methods), the model is only slightly improved from both interpretability and accuracy points of view. Hence, no significant changes are obtained. To understand this behavior we should consider that our reduction method is guided by an accuracy measure (the MSE function) to optimize the solution. Thus, the results obtained show us that both tuning methods obtain a very accurate model profiting from all the rules provided by the WM method that can not be significantly improved by removing some of them with an a posteriori reduction.

Nevertheless, the reduction method has an important influence when it is directly applied over the initial model (the WM + Reduction method). In this case, the interpretability is improved by a significant reduction of the number of rules (62% for the rice problem and 34% for the electrical one). It is due to the method removes the erroneous and conflictive rules that worsen the accuracy of the model.

Once these rules are removed, the posterior tuning process (WM + Reduction + Tuning methods) improves even more the accuracy of the model adapting the parameters to the subset of rules selected in the reduction stage. Thus, the tuning process adapts itself to the new rule set extracting a good accuracy from it. This combination results in a fuzzy model with a good interpretability-accuracy tradeoff. Moreover, the fact of dealing with a lower number of rules allows the tuning process to behave quicker. Fig. 13 shows the linguistic fuzzy models generated by the sequential WM + Reduction + PL-tun method for a run on a specific data set partition (the same partition considered in Fig. 12) of the electrical problem.

(a)



(b)

Rule	*Street*	*Town*	*Building*	*Energy*	*Costs*	*Rule*	*Street*	*Town*	*Building*	*Energy*	*Costs*
R_1	Sst	*m-or-l* Sst	Sst	*very* Sst	Sst	R_{20}	L	M	*m-or-l* S	M	*very* M
R_2	*m-or-l* Sst	*m-or-l* S	*very* S	S	S	R_{21}	*very* L	M	S	L	M
R_3	S	*m-or-l* Sst	*m-or-l* Sst	*very* Sst	*m-or-l* Sst	R_{22}	*m-or-l* L	M	*very* M	*very* M	*very* L
R_4	S	*m-or-l* Sst	*m-or-l* Sst	S	S	R_{23}	L	*very* L	*very* M	S	M
R_5	S	Sst	S	*very* S	S	R_{24}	*m-or-l* L	*m-or-l* L	*very* M	*very* L	*m-or-l* L
R_6	S	*m-or-l* S	*very* Sst	*m-or-l* Sst	*m-or-l* Sst	R_{25}	L	*m-or-l* L	L	S	L
R_7	*m-or-l* S	S	*m-or-l* Sst	*very* S	S	R_{26}	L	L	*m-or-l* L	M	L
R_8	*m-or-l* S	S	*very* S	*m-or-l* S	*very* S	R_{27}	*m-or-l* L	L	L	L	Lst
R_9	*m-or-l* S	M	S	*m-or-l* Sst	*m-or-l* S	R_{28}	L	Lst	L	*very* S	*m-or-l* M
R_{10}	*very* S	*m-or-l* M	*m-or-l* M	*very* S	*very* M	R_{29}	L	*very* Lst	*very* L	M	*m-or-l* L
R_{11}	*very* M	S	Sst	S	*very* S	R_{30}	*very* L	Lst	*very* L	*very* L	*m-or-l* Lst
R_{12}	*m-or-l* M	S	Sst	*very* M	*m-or-l* S	R_{31}	*very* L	*m-or-l* Lst	Lst	S	*very* Lst
R_{13}	M	*m-or-l* S	S	*very* M	*m-or-l* M	R_{32}	L	*m-or-l* Lst	*m-or-l* Lst	*m-or-l* M	Lst
R_{14}	*m-or-l* M	*very* M	*very* S	*m-or-l* S	S	R_{33}	*very* Lst	*very* S	*very* S	S	S
R_{15}	M	M	*very* S	*m-or-l* M	*m-or-l* M	R_{34}	Lst	*m-or-l* S	*m-or-l* S	Lst	*m-or-l* L
R_{16}	*m-or-l* M	M	M	S	*m-or-l* M	R_{35}	Lst	*m-or-l* S	*m-or-l* M	*m-or-l* S	*m-or-l* M
R_{17}	*m-or-l* M	L	*very* M	*m-or-l* M	M	R_{36}	Lst	S	*m-or-l* M	Lst	L
R_{18}	M	*m-or-l* L	*m-or-l* L	M	*very* L	R_{37}	*very* Lst	*m-or-l* L	M	*m-or-l* L	*m-or-l* L
R_{19}	L	*m-or-l* M	S	*m-or-l* Sst	S	R_{38}	*very* Lst	*very* L	M	*m-or-l* Lst	*very* Lst

Fig. 13. Knowledge base (fuzzy rule deep structures) generated by the WM + Reduction + PL-tun method for a specific run and data set partition on the electrical problem. $\#\mathrm{R}/\mathrm{MSE}_{\mathrm{tra/tst}}$ before Reduction + Tuning $= 65/58\,032/55\,150$, $\#\mathrm{R}/\mathrm{MSE}_{\mathrm{tra/tst}}$ after Reduction $= 38/42\,409/37\,472$, $\#\mathrm{R}/\mathrm{MSE}_{\mathrm{tra/tst}}$ after Tuning $= 38/10\,243/12\,035$. *Street* stands for *street lengths, Town* for *town area, Building* for *building area, energy* for *energy supply*, and *Costs* for *maintenance costs*. *Sst* stands for *smallest*, *S* for *small*, *M* for *medium*, *L* for *large*, *Lst* for *largest*, and *m-or-l* for *more-or-less*. (a) Tuned data base. (b) Reduced and tuned rule base (fuzzy rule surface structures).

Finally, it is interesting to check up that the simultaneous approaches—WM + (Tuning and Reduction) methods—generate worsen results in number of rules and accuracy degrees than WM + Reduction + Tuning methods. The reduction part of the algorithm only removes those rules that can not be properly adapted by the tuning process for a better approximation. This simultaneous action deviates the search toward worse local optimum. Perhaps it could be solve with a different fitness function or pool selection that reward the rule number reduction. In any case, this method will tackle a higher search space than the sequential Reduction + Tuning approach.

VI. Concluding Remarks

In this contribution, we have introduced a genetic tuning process for jointly refining as the fuzzy rule symbolic representations as the meaning of the involved membership functions of a linguistic fuzzy model. For the former case, we propose the use of linguistic hedges to perform slight modifications keeping a good interpretability. For the latter case, two different ways considering basic or extended expressions are proposed.

Our proposal has shown very good results in terms of efficiency and accuracy. The good performance of our tuning method mainly lies in the consideration of tuning at two different levels of significance and modifying the fuzzy set shapes without changing their support sets if desired. Moreover, we have observed that the combination of a simple learning method to design the preliminary surface structures and number of rules with our tuning method behaves better than a more sophisticated one-stage learning process.

On the other hand, we have also analyzed the interaction between the proposed tuning method and a fuzzy rule set reduction process. We have shown that a good interpretability-accuracy tradeoff is obtained by firstly reducing the number of rules and then tuning the resulting model. In this case, the tuning method can profit from the selected rules adapting them for a good accuracy.

APPENDIX
REAL-WORLD MODELING PROBLEMS CONSIDERED IN THE EXPERIMENTAL STUDIES

Two real-world problems have been considered in the experimental studies performed in this paper. The following subsections describe them.

A. Rice Taste Evaluation Problem

Qualification of rice taste is usually put into effect by means of a subjective evaluation called the *sensory test*. In this test, a group of experts evaluates the rice according to a set of characteristics associated to it. These factors are: *flavor, appearance, taste, stickiness*, and *toughness*. The use of linguistic fuzzy modeling techniques becomes very interesting to represent the existing nonlinear relationships of the problem in a legible and precise way.

To do so, we are going to use the data set presented in [5]. This set is composed of 105 data vectors collecting subjective evaluations of the six variables in question (the five mentioned and the overall evaluation of the kind of rice), made up by experts on this number of kinds of rice grown in Japan.

With the aim of not biasing the learning because of the small size of the data set, we have randomly obtained ten different partitions of the said set, composed of 75 pieces of data in the training set and 30 in the test one. We follow this experimental setup in this problem to facilitate the comparison with the work where the data was used [5]. In the probabilistic algorithms, six runs with different seeds for the pseudorandom sequence are made for each data partition. Therefore, it involves 60 different runs of each algorithm for this problem.

Two linguistic terms are considered for each variable fuzzy partition.

B. The Estimation of Electrical Network Maintenance Costs Problem

Estimating the maintenance costs of an electrical network in a town [51] is a complex but interesting problem. Since an actual measure is very difficult to obtain when medium or low voltage lines are used, the consideration of models becomes useful. These estimations allow electrical companies to justify their expenses. Moreover, the model must be able to explain how a specific value is computed for a certain town. Our objective will be to relate the *maintenance costs* of medium voltage line with the following four variables: *sum of the lengths of all streets* in the town, *total area* of the town, *area occupied* by buildings, and *energy supply* to the town. We will deal with estimations of minimum maintenance costs based on a model of the optimal electrical network for a town. We were provided with a sample of 1,056 simulated towns.

To develop the different experiments for this problem, a 5-fold cross validation is performed. Thus, the data set is divided into five subsets of (approximately) equal size. Each algorithm is applied five times for each problem, each time leaving out one of the subsets from training, but using only the omitted subset to compute the test error. The training and test data partitions used in this problem are freely available at http://decsai.ugr.es/~casillas/FMLib/. In the probabilistic algorithms, six runs with different seeds for the pseudo-random sequence are made for each data partition. Therefore, it involves 30 different runs of each algorithm for this problem.

With this experimental setup that uses public real-world problems and cross-validation with multiple runs, we try to perform a sound experimental study, more rigorous than those usually performed by the fuzzy modeling community as remarked in [52] and [53].

Five linguistic terms are considered for each variable fuzzy partition.

REFERENCES

[1] L. A. Zadeh, "Concept of a linguistic variable and its application to approximate reasoning—1, 2, and 3," *Inform. Sci.*, vol. 8, 8, 9, no. 3, 4, 1, pp. 199–249, 301–357, 43–80, 1975/1976.

[2] M. Sugeno and T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 1, pp. 7–31, Feb. 1993.

[3] L. A. Zadeh, "Soft computing and fuzzy logic," *IEEE Software*, vol. 11, no. 6, pp. 48–56, 1994.

[4] 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, Dec. 1992.

[5] K. Nozaki, H. Ishibuchi, and H. Tanaka, "A simple but powerful heuristic method for generating fuzzy rules from numerical data," *Fuzzy Sets Syst.*, vol. 86, no. 3, pp. 251–270, 1997.

[6] D. Nauck, F. Klawonn, and R. Kruse, *Fundations of Neuro-Fuzzy Systems*. New York: Wiley, 1997.

[7] R. Fullér, *Introduction to Neuro-Fuzzy Systems*. Heidelberg, Germany: Springer-Verlag, 2000.

[8] O. Cordón, F. Herrera, F. Hoffmann, and L. Magdalena, *Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*. Singapore: World Scientific, 2001.

[9] J. Casillas, O. Cordón, F. Herrera, and L. Magdalena, Eds., *Interpretability Issues in Fuzzy Modeling*. Heidelberg, Germany: Springer-Verlag, 2003.

[10] J. Casillas, O. Cordón, F. Herrera, and L. Magdalena, Eds., *Accuracy Improvements in Linguistic Fuzzy Modeling*. Heidelberg, Germany: Springer-Verlag, 2003.

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

[12] 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.

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

[14] O. Cordón, M. J. del Jesus, and F. Herrera, "Genetic learning of fuzzy rule-based classification systems cooperating with fuzzy reasoning methods," *Int. J. Intell. Syst.*, vol. 13, pp. 1025–1053, 1998.

[15] Y. Jin, W. von Seelen, and B. Sendhoff, "On generating FC^3 fuzzy rule systems from data using evolution strategies," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 29, no. 4, pp. 829–845, Aug. 1999.

[16] D. Nauck and R. Kruse, "Neuro-fuzzy systems for function approximation," *Fuzzy Sets Syst.*, vol. 101, no. 2, pp. 261–271, 1999.

[17] B.-D. Liu, C.-Y. Chen, and J.-Y. Tsao, "Design of adaptive fuzzy logic controller based on linguistic-hedge concepts and genetic algorithms," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 31, no. 1, pp. 32–53, Feb. 2001.

[18] Y. Shi, R. Eberhart, and Y. Chen, "Implementation of evolutionary fuzzy systems," *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 2, pp. 109–119, Apr. 1999.

[19] W. Pedrycz, R. R. Gudwin, and F. A. C. Gomide, "Nonlinear context adaptation in the calibration of fuzzy sets," *Fuzzy Sets Syst.*, vol. 88, no. 1, pp. 91–97, 1997.

[20] J. Espinosa and J. Vandewalle, "Constructing fuzzy models with linguistic integrity from numerical data-AFRELI algorithm," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 5, pp. 591–600, Oct. 2000.

[21] A. González and R. Pérez, "A study about the inclusion of linguistic hedges in a fuzzy rule learning algorithm," *Int. J. Uncertainty, Fuzziness, Knowledge-Based Syst.*, vol. 7, no. 3, pp. 257–266, 1999.

[22] N. R. Pal and K. Pal, "Handling of inconsistent rules with an extended model of fuzzy reasoning," *J. Intell. Fuzzy Syst.*, vol. 7, pp. 55–73, 1999.

[23] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, "Selecting fuzzy IF–THEN rules for classification problems using genetic algorithms," *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 3, pp. 260–270, Jun. 1995.
[24] T.-P. Hong and C.-Y. Lee, "Effect of merging order on performance of fuzzy induction," *Intell. Data Anal.*, vol. 3, no. 2, pp. 139–151, 1999.
[25] T.-P. Hong and J.-B. Chen, "Finding relevant attributes and membership functions," *Fuzzy Sets Syst.*, vol. 103, no. 3, pp. 389–404, 1999.
[26] H.-M. Lee, C.-M. Chen, J.-M. Chen, and Y.-L. Jou, "An efficient fuzzy classifier with feature selection based on fuzzy entropy," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 31, no. 3, pp. 426–432, Jun. 2001.
[27] J. Yen and L. Wang, "An SVD-based fuzzy model reduction strategy," in *Proc. 5th IEEE Int. Conf. Fuzzy Systems*, New Orleans, LA, 1996, pp. 835–841.
[28] ——, "Simplifying fuzzy rule-based models using orthogonal transformation methods," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 29, no. 1, pp. 13–24, Feb. 1999.
[29] Y. Yam, P. Baranyi, and C.-T. Yang, "Reduction of fuzzy rule base via singular value decomposition," *IEEE Trans. Fuzzy Syst.*, vol. 7, pp. 120–132, Apr. 1999.
[30] T. Taniguchi, K. Tanaka, H. Ohtake, and H. O. Wang, "Model construction, rule reduction, and robust compensation for generalized form of Takagi-Sugeno fuzzy systems," *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 4, pp. 525–538, Aug. 2001.
[31] M. Setnes, R. Babuška, U. Kaymak, and H. R. van Nauta Lemke, "Similarity measures in fuzzy rule base simplification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 28, no. 3, pp. 376–386, Jun. 1998.
[32] M. Setnes and J. A. Roubos, "GA-fuzzy modeling and classification: Complexity and performance," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 5, pp. 509–522, Oct. 2000.
[33] J. A. Roubos and M. Setnes, "Compact and transparent fuzzy models and classifiers through iterative complexity reduction," *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 4, pp. 516–524, Aug. 2001.
[34] M. L. Hadjili and V. Wertz, "Takagi-Sugeno fuzzy modeling incorporating input variables selection," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 6, pp. 728–742, Dec. 2002.
[35] J. Yen, L. Wang, and C. W. Gillespie, "Improving the interpretability of TSK fuzzy models by combining global learning and local learning," *IEEE Trans. Fuzzy Syst.*, vol. 6, no. 4, pp. 530–537, Aug. 1998.
[36] M. Bikdash, "A highly interpretable form of Sugeno inference systems," *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 6, pp. 686–696, Dec. 1999.
[37] A. Fiordaliso, "A constrained Takagi–Sugeno fuzzy system that allows for better interpretation and analysis," *Fuzzy Sets Syst.*, vol. 118, no. 2, pp. 307–318, 2001.
[38] P. Thrift, "Fuzzy logic synthesis with genetic algorithms," in *Proc. 4th Int. Conf. Genetic Algorithms*, R. K. Belew and L. B. Booker, Eds., San Mateo, CA, 1991, pp. 509–513.
[39] F. Herrera, M. Lozano, and J. L. Verdegay, "Fuzzy connectives based crossover operators to model genetic algorithms population diversity," *Fuzzy Sets Syst.*, vol. 92, no. 1, pp. 21–30, 1997.
[40] J. E. Baker, "Reducing bias and inefficiency in the selection algorithm," in *Proc. 2nd Int. Conf. Genetic Algorithms*, Hillsdale, NJ, 1987, pp. 14–21.
[41] P. P. Bonissone, P. S. Khedkar, and Y. Chen, "Genetic algorithms for automated tuning of fuzzy controllers: A transportation application," in *Proc. 5th IEEE Int. Conf. Fuzzy Systems*, New Orleans, LA, 1996, pp. 674–680.
[42] F. Cheong and R. Lai, "Constraining the optimization of a fuzzy logic controller using an enhanced genetic algorithm," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 30, no. 1, pp. 31–46, Jan. 2000.
[43] T. Suzuki, T. Kodama, T. Furuhashi, and H. Tsutsui, "Fuzzy modeling using genetic algorithms with fuzzy entropy as conciseness measure," *Inform. Sci.*, vol. 136, no. 1–4, pp. 53–67, 2001.
[44] J. Casillas, O. Cordón, and F. Herrera, "COR: A methodology to improve ad hoc data-driven linguistic rule learning methods by inducing cooperation among rules," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 32, no. 4, pp. 526–537, 2002.
[45] J. Liska and S. S. Melsheimer, "Complete design of fuzzy logic systems using genetic algorithms," in *Proc. 3rd IEEE Int. Conf. Fuzzy Systems*, Orlando, FL, 1994, pp. 1377–1382.
[46] H. Ishibuchi, T. Murata, and I. B. Türkşen, "Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems," *Fuzzy Sets Syst.*, vol. 89, no. 2, pp. 135–150, 1997.
[47] F. Herrera, M. Lozano, and J. L. Verdegay, "A learning process for fuzzy control rules using genetic algorithms," *Fuzzy Sets Syst.*, vol. 100, pp. 143–158, 1998.
[48] O. Cordón and F. Herrera, "A proposal for improving the accuracy of linguistic modeling," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 3, pp. 335–344, Jun. 2000.
[49] A. Krone and H. Taeger, "Data-based fuzzy rule test for fuzzy modeling," *Fuzzy Sets Syst.*, vol. 123, no. 3, pp. 343–358, 2001.
[50] A. Klose, A. Nurnberger, and D. Nauck, "Some approaches to improve the interpretability of neuro-fuzzy classifers," in *Proc. 6th Eur. Congr. Intelligent Techniques and Soft Computing*, Aachen, Germany, 1998, pp. 629–633.
[51] O. Cordón, F. Herrera, and L. Sánchez, "Solving electrical distribution problems using hybrid evolutionary data analysis techniques," *Appl. Intell.*, vol. 10, no. 1, pp. 5–24, 1999.
[52] J. A. Roubos and R. R. Babuška, "Comments on the benchmarks in 'A proposal for improving the accuracy of linguistic modeling' and related articles," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 6, pp. 861–865, Dec. 2003.
[53] O. Cordón and F. Herrera, "Authors' reply," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 6, pp. 866–869, Dec. 2003.

Jorge Casillas received the M.Sc. degree in computer engineering and the Ph.D. degree in computer science, both from the University of Granada, Granada, Spain, in 1998 and 2001, respectively.

He is an Assistant 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. He has worked on several research projects supported by the Spanish government and the European Union. He has coedited two books on the topic of interpretability-accuracy tradeoff in fuzzy modeling. His research interests include fuzzy modeling, knowledge discovery, and metaheuristics.

Oscar Cordón was born in Cadiz, Spain, on August 11, 1972. He received the M.S. and Ph.D. degrees in computer science, both from the University of Granada, Granada, Spain, in 1994 and 1997, respectively.

He is an Associate Professor in the Department of Computer Science and Artificial Intelligence of the University of Granada, where he is a Member of the Soft Computing and Intelligent Information Systems Research Group, and heads the Virtual Learning Center of the University (CEVUG). He has written more than 40 papers in international journals and is coauthor of the book Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases (Singapore: World Scientific, 2001). He has co-organized special sessions on genetic fuzzy systems, ant colony optimization, and soft computing applications to intelligent information retrieval in several national and international conferences, and has co-edited four special issues on the previous topics in *Information Sciences*, *Mathware and Soft Computing*, the *International Journal of Approximate Reasoning*, and *Fuzzy Sets and Systems*, as well as three books. He has also worked on eleven research projects (being the coordinator of three of them) supported by the Spanish and Andalusian governments, the University of Granada, and the European Union under the JOULE project, concerning several aspects in the area of genetic algorithms, fuzzy systems, ant colony optimization and other metaheuristics, and eLearning. His current main research interests are in the fields of fuzzy rule-based systems, fuzzy and linguistic modeling, genetic fuzzy systems, soft computing and information retrieval, and evolutionary algorithms, ant colony optimization and other metaheuristics, and their applications to different topics (image registration, bioinformatics, etc.).

María José del Jesus received the M.Sc. and Ph.D. degrees in computer sciences from the University of Granada, Granada, Spain, in 1994 and 1999, respectively.

Currently, she is an Associate Professor with the Department of Computer Sciences, University of Jaén, Jaén, Spain. Her research interests are in the fields of fuzzy rule-based systems, fuzzy and linguistic modeling, fuzzy classification, machine learning, and genetic fuzzy systems.

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

He is an Associate Professor with the Department of Computer Science and Artificial Intelligence, University of Granada. He is coauthor of the book *Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases* (Singapore: World Scientific, 2001). He has co-edited three international books, co-organized several special sessions in international conferences, and co-edited some special issues in international journals on preference modeling, computing with words and genetic fuzzy systems. His research interests include decision making problems in fuzzy environment, fuzzy rule-based systems, machine learning, ant colony optimization, genetic algorithms, and genetic fuzzy systems.