Linguistic Modeling by Hierarchical Systems of Linguistic Rules

Oscar Cordón, Francisco Herrera, and Igor Zwir

***Abstract*—In this paper, we are going to propose an approach to design linguistic models which are accurate to a high degree and may be suitably interpreted. This approach will be based on the development of a Hierarchical System of Linguistic Rules learning methodology. This methodology has been thought as a refinement of simple linguistic models which, preserving their descriptive power, introduces small changes to increase their accuracy. To do so, we extend the structure of the Knowledge Base of Fuzzy Rule Base Systems in a hierarchical way, in order to make it more flexible. This flexibilization will allow us to have linguistic rules defined over linguistic partitions with different granularity levels, and thus to improve the modeling of those problem subspaces where the former models have bad performance.**

***Index Terms*—Genetic algorithms, hierarchical knowledge base, hierarchical linguistic partitions, linguistic modeling, Mamdani-type fuzzy rule-based systems, rule selection.**

I. Introduction

NOWADAYS, one of the most important areas for the application of Fuzzy Set Theory as developed by Zadeh [31] are fuzzy rule-based systems (FRBSs). These kinds of systems constitute an extension of classical rule-based systems, because they deal with linguistic rules instead of classical logic rules. Thanks to this, they have been successfully applied to a wide range of problems from different areas presenting uncertainty and vagueness in different ways [1], [13], [21], [19].

One of the most important applications of FRBSs is *System Modeling* [1], [21]. It is possible to distinguish between two types of modeling when we are working with FRBSs: *linguistic modeling* [25] and *fuzzy modeling* [1], according to the fact that the main requirement is the interpretability or the accuracy of the model, respectively. In fact, we usually find these two contradictory requirements, the accuracy and the interpretability of the model obtained. The choice between how interpretable and how accurate the model must be usually depends on the user's needs for a specific problem and will condition the kind of FRBS selected to model it.

Linguistic modeling has a problem associated, which is its lack of accuracy in some complex problems. In this paper, we are going to propose a simple linguistic modeling refinement approach—developed by means of linguistic FRBSs—which allows us to improve the accuracy of these kinds of models without losing its interpretability to a high degree. This approach considers the development of a hierarchical system of linguistic rules learning methodology (HSLR-LM), whose linguistic variables are defined on linguistic partitions with different granularity levels.

We extend the knowledge base (KB) structure of linguistic FRBSs by introducing the concept of "*layers*." In this extension, which is also a generalization, the KB is composed of a set of layers where each one contains linguistic partitions with different granularity levels and linguistic rules whose linguistic variables take values in these partitions. This KB is called hierarchical knowledge base (HKB), and it is formed by a hierarchical database (HDB) and a hierarchical rule base (HRB), containing linguistic partitions of the said type and linguistic rules defined over them, respectively.

To do so, this paper is set up as follows. In Section II, the balance between accuracy and interpretability in linguistic modeling is analyzed, as well as previous approaches to hierarchical fuzzy systems are discussed. In Section III, a description of the HKB and the relation between its components is regarded. In Section IV, a methodology to automatically design an HSLR from a generic linguistic rule generating method is introduced. In Section V, a linguistic modeling process obtained from the HSLR-LM and a well-known inductive linguistic rule generation process is applied to solve two different applications. In Section VI, we discuss some features of our methodology. Finally, in Section VII, some concluding remarks are pointed out.

II. Background and Framework

A. Balance Accuracy-Interpretability

As we have said, two types of modeling with FRBSs are distinguished according to the fact that the main requirement is the interpretability or the accuracy of the model: linguistic modeling and fuzzy modeling, respectively. These requirements are always contradictory.

The KB structure usually employed in the field of linguistic modeling has the drawback of its lack of accuracy when working with very complex systems. This fact is due to some problems related to the linguistic rule structure considered, which are a consequence of the inflexibility of the concept of linguistic variable [32]. A summary of these problems may be found in [2], [4], and it is briefly enumerated as follows.

- There is a lack of flexibility in the FRBSs because of the rigid partitioning of the input and output spaces.
- When the system input variables are dependent themselves, it is very hard to fuzzy partition the input spaces.

Manuscript received December 6, 2000; revised April 25, 2001. This work was supported by CICYT TIC96-0778 and PB98-1319.

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

I. Zwir is with the Department of Computer Science, FCEyN, University of Buenos Aires, 1428 Buenos Aires, Argentina (e-mail: zwir@dc.uba.ar).

Publisher Item Identifier S 1063-6706(02)01531-X.

- The homogenous partitioning of the input and output spaces when the input-output mapping varies in complexity within the space is inefficient and does not scale to high-dimensional spaces.
- The size of the rule base (RB) directly depends on the number of variables and linguistic terms in the system. Obtaining an accurate FRBS requires a significant granularity amount, i.e., it needs of the creation of new linguistic terms. This granularity increase causes the number of rules to rise significantly, which may take the system to lose the capability of being interpretable for human beings.

At least two things could be done to solve many of these problems and to improve the model accuracy. On the one hand, we can use fuzzy modeling, with the consequence of losing the model interpretability. On the other hand, we can refine a linguistic model trying not to change too much the meaning of the linguistic variables neither the descriptive power of the final FRBS generated.

In this paper, we will focus our attention on the second choice. Our methodology is proposed as an strategy to improve simple linguistic models, preserving their structure and descriptive power, and reinforcing only the modeling of those problem subspaces with more difficulties by a hierarchical treatment of the rules generated in these zones. In other words, we propose a refinement of simple linguistic models which introduces small changes to increase their accuracy.

The purpose of this extension is the flexibilization of the KB to become an HKB. This is possible by the development of a new KB structure, where the linguistic variables of the linguistic rules could take values from fuzzy partitions with different granularity levels. To do so, we will use an HKB of two layers, i.e., starting with an initial *layer* t, we produce *layer* $t+1$ in order to extract the final system of linguistic rules. This fact allows the HSLR to perform a significantly more accurate modeling of the problem space.

B. Previous Approaches to Hierarchical FRBSs

In this section, we will discuss the reach of our present methodology, comparing it with other previous approaches oriented to hierarchical processors, hierarchical fuzzy inference and rule extraction from a global hierarchical KB.

One of the previous hierarchical approaches have been directed to design a series of hierarchical fuzzy processors with a small number of input variables distributed in each processor [22]. While the computational efficiency of the distributed structure of the fuzzy processors is evident, the aggregation of those intermediate variables will contribute to lose the initial meaning of the model, diminishing its descriptive power.

Other works were also developed by Yager [28], [29], Gegov *et al.* [10] and Ishibuchi *et al.* [14], [15] in order to aggregate different priority levels of information in a hierarchical inference process.

As said, our approach is oriented to produce hierarchical rules, i.e., FRBSs whose RB is composed of linguistic rules defined on fuzzy partitions with different granularity levels. Our purpose is to preserve the descriptive power of the system of rules and to simplify the inference mechanism adopted by other previous hierarchical approaches, activating independently each rule as it is done in the conventional inference mechanism. Besides, we use a genetic selection process to obtain a compact set of rules that have good cooperation between them.

Finally, another approach generated in the same line have been performed by Ishibuchi et al. in [16]. Although it is not explicitly shown as an hierarchical methodology, because of its use of different granularity partitions, it can be adapted as such kinds of models by our present proposal. There, a genetic-algorithm-based method for removing unnecessary rules from fuzzy *if–then* rule sets corresponding to several fuzzy partitions is proposed. While this approach generates the whole set of fuzzy rules from each different granularity level fuzzy partition and then performs a genetic rule selection over all rules, we will focus our attention on those rules which model a subspace of the problem with significant error. That is, only these bad rules are expanded in a hierarchical way and then joined with the good ones, in order to perform a selection process which produces a good cooperation among them. All of these is done with the purpose of improving the system accuracy, preserving its description as far as possible.

III. Hierarchical Knowledge Base

Due the reasons described in Ssection II-A and to solve many of these problems, we present a new more flexible KB structure that allows us to improve the accuracy of linguistic models without losing their interpretability: the HKB, which is composed of a set of layers. We define a layer by its components in the following way:

$$\text{layer}(t, n(t)) = \text{DB}(t, n(t)) + \text{RB}(t, n(t)) \qquad (1)$$

with

- $n(t)$ being the number of linguistic terms that compose the partitions of layer t;
- $\text{DB}(t, n(t))$ being the Data Base (DB) which contains the linguistic partitions with granularity level $n(t)$ of layer t;
- $\text{RB}(t, n(t))$ being the RB formed by those linguistic rules whose linguistic variables take values in the former partitions.

At this point, we should note that, in this work, we are using *linguistic partitions* with the same number of linguistic terms for all input–output variables, composed of triangular-shaped, symmetrical and uniformly distributed membership functions.

From now on and for the sake of simplicity, we are going to refer to the components of a $\text{DB}(t, n(t))$ and $\text{RB}(t, n(t))$ as *t-linguistic partitions* and *t-linguistic rules*, respectively.

This set of layers is organized as a hierarchy, where the order is given by the granularity level of the linguistic partition defined in each layer. That is, given two successive layers t and $t+1$, then the granularity level of the linguistic partitions of layer $t+1$ is greater than the ones of layer t. This causes a refinement of the previous layer linguistic partitions.

As a consequence of the previous definitions, we could now define the HKB as the union of every layer t

$$\text{HKB} = \cup_t \text{layer}(t, n(t)). \qquad (2)$$

In the remainder of this Section, we are going to study the linguistic partitions and their extension to consider them as component parts of the $\text{DB}(t, n(t))$ of the $\text{layer}(t, n(t))$. Then, we are going to describe the relation between DBs from different layers (e.g., t and $t+1$), and to develop a methodology to build them under certain requirements. Finally, we will explain how to relate these DBs with linguistic rules, i.e., to create RBs from them.

A. Hierarchical Data Base

In this section, we are going to show how to build the HDB, bearing in mind that it is organized in a hierarchy, where the order is given by an increasing granularity level of the linguistic partitions.

To extend the classical linguistic partition, let us consider a partition P of the domain U of a linguistic variable A in the layer t:

$$P_A = \{S_1, \ldots, S_{n(t)}\} \tag{3}$$

with S_k $(k = 1, \ldots, n(t))$ being linguistic terms which describe the linguistic variable A. These linguistic terms are mapped into fuzzy sets by the semantic function M, which gives them a meaning: $M_U : S_k \rightarrow \mu_{S_k}(u)$ [32].

We extend this definition of P allowing the existence of several partitions, each one with a different number of linguistic terms, i.e., with a different granularity level. To do so, we add a parameter $n(t)$ to the definition of the linguistic partition P, which represents the granularity level of the partitions contained in the layer t where it is defined

$$P_A^{n(t)} = \left\{S_1^{n(t)}, \ldots, S_{n(t)}^{n(t)}\right\} \tag{4}$$

where $P_A^{n(t)} \in \text{DB}(t, n(t))$.

In order to build the HDB, we develop an strategy which satisfies two main requirements

- to preserve all possible fuzzy set structures from one layer to the next in the hierarchy;
- to make smooth transitions between successive layers.

On the one hand, we decided to preserve all the membership function modal points, corresponding to each linguistic term, through the higher layers of the hierarchy in order to fulfill the first requirement. On the other hand, and with the aim of building a new $t+1$-*linguistic partition*, we just add a new linguistic term between each two consecutive terms of the t-*linguistic partition*. To do so, we reduce the support of these linguistic terms in order to keep place for the new one, which is located in the middle of them. An example of the correspondence between a 1-*linguistic partition* and a 2-*linguistic partition*, with $n(1) = 3$ and $n(2) = 5$, respectively, is shown in Fig. 1.

As a result of the above considerations, Table I shows the number of linguistic terms which is needed in each t-*linguistic partition* in $\text{DB}(t, n(t))$ to satisfy the previous requirements. The values of parameter $n(t)$ represent the t-*linguistic partition* granularity levels and depend on the initial value of $n(t)$ defined in the first layer (e.g., 2 or 4 in Table I).

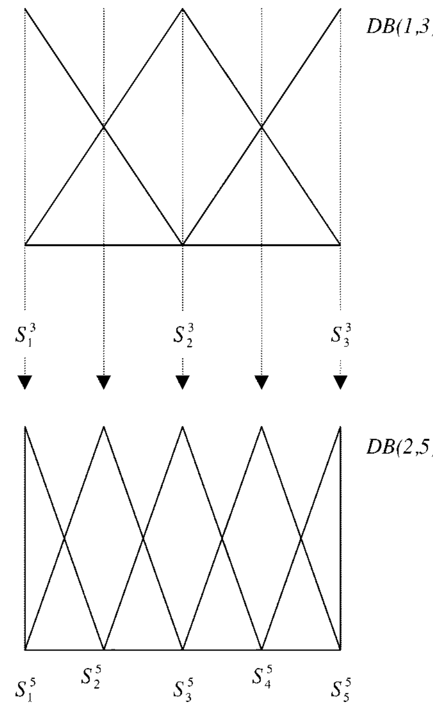

Fig. 1. Transition from a partition in $\text{DB}(1, 3)$ to another one in $\text{DB}(2, 5)$.

TABLE I
HIERARCHY OF DBs STARTING FROM TWO OR FOUR INITIAL TERMS

DB(t,n(t))		**DB(t,n(t))**
$DB(1,2)$		$DB(1,4)$
$DB(2,3)$		$DB(2,7)$
$DB(3,5)$	*or*	$DB(3,13)$
$DB(4,9)$		$DB(4,25)$
⋮		⋮
$DB(6,33)$		$DB(6,97)$
⋮		⋮

Generically, we could say that a DB from a layer $t+1$ is obtained from its predecessor as

$$\text{DB}(t, n(t)) \rightarrow \text{DB}(t+1, 2 \cdot n(t) - 1) \tag{5}$$

which means that a t-*linguistic partition* in $\text{DB}(t, n(t))$ with $n(t)$ linguistic terms becomes a $(t+1)$-*linguistic partition* in $\text{DB}(t+1, 2 \cdot n(t) - 1)$.

In order to satisfy the previous requirements, each linguistic term $S_k^{n(t)}$-term of order k from the t-*linguistic partition* in $\text{DB}(t, n(t))$—is mapped into $S_{2k-1}^{2 \cdot n(t)-1}$, preserving the former modal points, and a set of $n(t) - 1$ new terms is created, each one between $S_k^{n(t)}$ and $S_{k+1}^{n(t)}$ $(k = 1, \ldots, n(t) - 1)$. This mapping is clearly shown in Table II and a graphical example is to be found in Fig. 1.

TABLE II
MAPPING BETWEEN LINGUISTIC TERMS FROM SUCCESSIVE DBs

$\mathbf{DB(t,n(t))}$		$\mathbf{DB(t+1,2\cdot n(t)\text{-}1)}$
$S_{k-1}^{n(t)}$	$\longrightarrow$	$S_{2k-3}^{2\cdot n(t)-1}$
		$S_{2k-2}^{2\cdot n(t)-1}$
$S_{k}^{n(t)}$	$\longrightarrow$	$S_{2k-1}^{2\cdot n(t)-1}$
		$S_{2k}^{2\cdot n(t)-1}$
$S_{k+1}^{n(t)}$	$\longrightarrow$	$S_{2k+1}^{2\cdot n(t)-1}$

In this view, we can generalize this two-level successive layer definition for $n(t)$, for all layers t in the following way:

$$n(t) = (N-1)\cdot 2^{t-1} + 1 \qquad (6)$$

and:

$$n(1) = N$$

i.e., the number of linguistic terms in the initial layer partitions.

B. Hierarchical Rule Base

In this section, we explain how to develop an RB from layer $t+1$ based on RB$(t,n(t))$, DB$(t,n(t))$ and DB$(t+1, 2\cdot n(t)-1)$, in order to create an HRB. Later, in the following section, we are going to give a concrete method to perform this task for a two-layer HKB.

First, let us define the *t-linguistic rules* contained in RB$(t,n(t))$ as those rules whose linguistic variables take values from the *t-linguistic partitions* contained in DB$(t,n(t))$. The *t-linguistic rule* structure is formed by a collection of well-known Mamdani-type linguistic rules

$$R_i^{n(t)}: \text{IF } x_1 \text{ is } S_{i1}^{n(t)} \text{ and } \ldots \text{ and } x_m \text{ is } S_{im}^{n(t)} \text{ THEN } y \text{ is } B_i^{n(t)}$$

with $x_1,\ldots,x_m$ and y being the input linguistic variables and the output one, respectively, and with $S_{i1}^{n(t)},\ldots,S_{im}^{n(t)}, B_i^{n(t)}$ being linguistic terms from different *t-linguistic partitions* of DB$(t,n(t))$, with fuzzy sets associated defining their meaning. In this contribution, we will use the Minimum t-norm in the role of conjunctive and implication operator and the *Center of Gravity weighted by the matching degree* [5] as defuzzification strategy.

The main purpose of developing an HRB is to model the problem space in a more accurate way. To do so, those *t-linguistic rules* that model a subspace with bad performance are expanded into a set of $(t+1)$*-linguistic rules*, which become their image in RB$(t+1, 2\cdot n(t)-1)$. This set of rules models the same subspace that the former one and replaces it.

We should note that not all *t-linguistic rules* are to be expanded. Only those *t-linguistic rules* which model a subspace of the problem with a significant error, become the ones that are involved in this rule expansion process to build the RB$(t+1, 2\cdot n(t)-1)$. The remaining rules preserve their location in RB$(t,n(t))$. An explanation for this behavior could be found in the fact that it is not always true that a set of rules with a higher granularity level, performs a better modeling of a problem than another one, with a lower granularity level. Moreover, this is not true for all kinds of problems, and what is more, it is also not true for all linguistic rules that model a problem [8].

IV. SIMPLE LINGUISTIC MODELS REFINEMENT: A TWO-LEVEL HSLR LEARNING METHODOLOGY

Our methodology is proposed as an strategy to improve simple linguistic models preserving their structure and descriptive power, reinforcing only the modeling of those problem subspaces with more difficulties. Due to this reason, our HSLRs will be based on two hierarchical levels, i.e., two layers.

In the following, the structure of the learning methodology and its most important components are described in detail.

A. Structure of the Two-Level HSLR Learning Methodology

Our HSLR-LM is composed of three main processes which will be described in depth in the following subsections.

- The first process generates the HKB following the descriptions given in Section III. This process is presented in Section IV-B.
- The second process performs a genetic rule selection task that removes the redundant or unnecessary rules from the HRB, in order to select a subset of rules cooperating better. It is explained in Section IV-C.
- In the third process, a user evaluation process extends this approach to an iterative process, where he could adapt some parameters and re-execute the processes to achieve better results. It is described in Section IV-D.

It basically consists of the following steps which are listed in Table III and may be also graphically seen in Fig. 2.

B. Hierarchical Knowledge Base Generation Process

In this section, we present our methodology to generate an HKB. To do so, we use a linguistic rule generating (LRG) method, which, as an inductive method, is based on the existence of a set of input-output data E_{TDS} and a previously defined DB$(1,n(1))$. The data set $E_{\text{TDS}} = \{e^1,\ldots,e^l,\ldots,e^p\}$ is composed of p input-output data pairs $e^l = (ex_1^l,\ldots,ex_m^l, ey^l)$, which represent the behavior of the system being modeled.

Our HKB generation process has three main steps, that are listed below.

1) $\mathbf{RB(1,n(1))}$ **generation process**, where the rules of the initial layer $(t=1)$ are generated from the present DB$(1,n(1))$.
An LRG-method is run with the terms defined in the present partitions, that are in DB$(1,n(1))$, *denoted as:*

$$\text{RB}(1,n(1)) = \text{LRG}-\text{method}(\text{DB}(1,n(1)), E_{\text{TDS}}) \qquad (7)$$

with $n(1)=N$ and the initial DB$(1,n(1))$, given by an expert or by a normalization process considering a small number of terms.

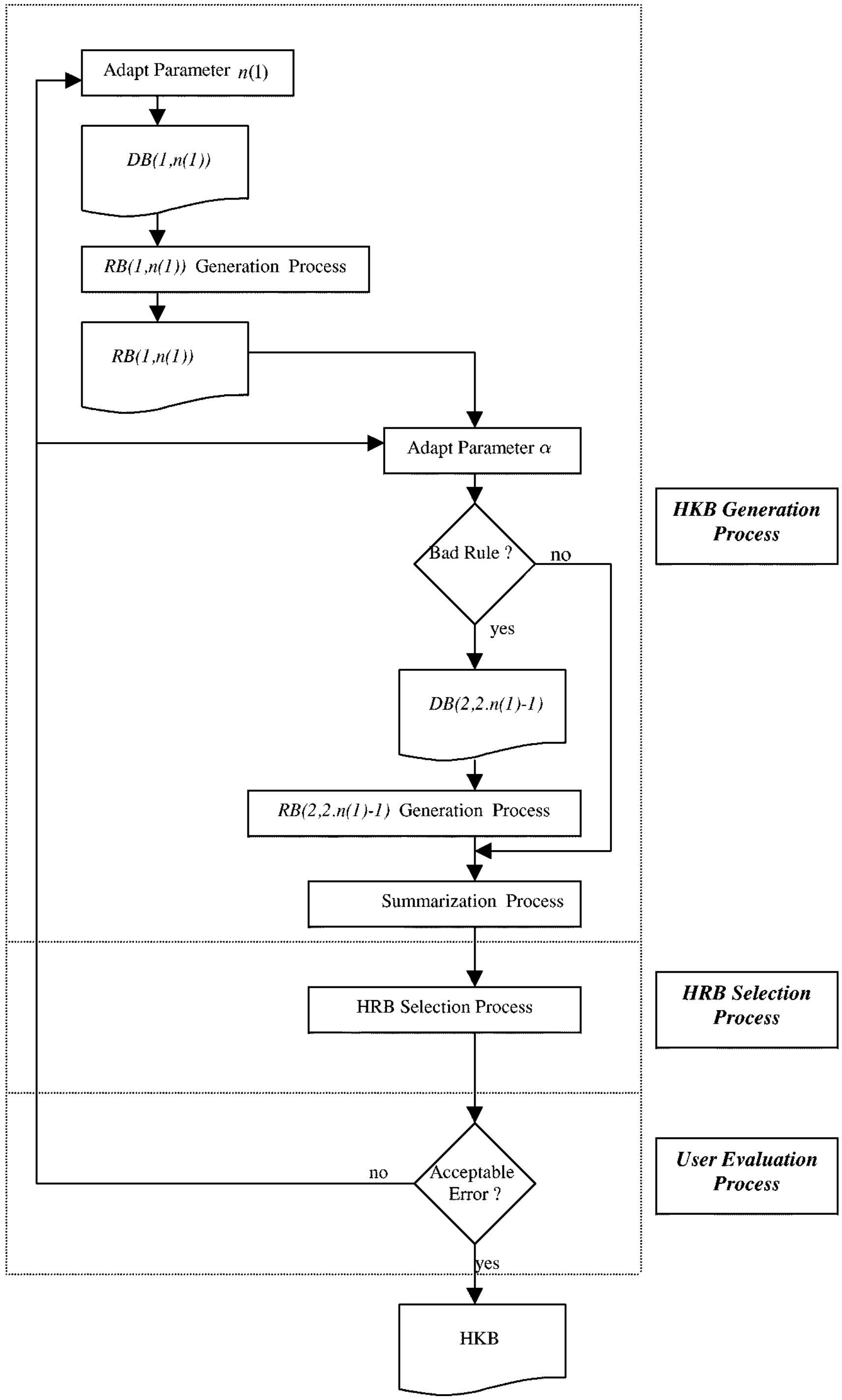

Fig. 2. HSLR-LM.

2) $\mathbf{RB(2, 2 \cdot n(1) - 1)}$ **generation process**, where the linguistic rules from layer 2 are generated taking into account $\mathrm{RB}(1, n(1)), \mathrm{DB}(1, n(1))$ and $\mathrm{DB}(2, 2 \cdot n(1) - 1)$.

a) *Bad performance 1-linguistic rule selection process.* This process performs the selection of those *1-linguistic rules* from $\mathrm{RB}(1, n(1))$ which will be expanded in $\mathrm{RB}(2, 2 \cdot n(1) - 1)$, based on an error measure. This measure analyzes the accuracy of the modeling performed by each *1-linguistic rule* in its definition subspace with respect to the global performance of the whole RB. These bad performance *1-linguistic rules* are going to be replaced by subsets of *2-linguistic rules*, which are going to be generated as their image. To do so, we have to follow the next steps:

i) *Calculate the error of* $\mathrm{RB}(1, n(1))$ *as a whole.* Compute $\mathrm{MSE}(E_{\mathrm{TDS}}, \mathrm{RB}(1, n(1)))$. The mean square error (MSE) calculated over a training data set, E_{TDS}, is the error measure used in this work. Therefore, the

TABLE III
HSLR LEARNING METHODOLOGY

1. **Hierarchical Knowledge Base Generation Process**
   (a) *RB(1,n(1)) Generation Process*
   (b) *RB(2,2·n(1)-1) Generation Process*
   (c) *Summarization Process*
2. **Hierarchical Rule Base Genetic Selection Process**
3. **User Evaluation Process**

MSE of the entire set of *1-linguistic rules* is represented by the following expression:

$$\text{MSE}(E_{\text{TDS}}, \text{RB}(1,n(1))) = \frac{\sum_{e^l \in E_{\text{TDS}}} (ey^l - s(ex^l))^2}{2 \cdot |E_{\text{TDS}}|} \quad (8)$$

with $s(ex^l)$ being the output value obtained from the $\text{RB}(1,n(1))$, when the input variable values are $ex^l = (ex^l_1, \ldots, ex^l_m)$, and ey^l is the known desired value.

ii) *Calculate the error of each individual 1-linguistic rule.* Compute $\text{MSE}(E_i, R_i^{n(1)})$. We need to define a subset of E_{TDS}, E_i, to be used to calculate the error of the rule $R_i^{n(1)}$. The set E_i is a set of the examples matching the antecedents of the rule i to a specific degree τ:

$$E_i = \left\{ e^l \in E_{\text{TDS}} / \text{Min}\left(\mu_{S_{i1}^{n(t)}}\left(ex_1^l\right), \ldots, \mu_{S_{im}^{n(t)}}\left(ex_m^l\right)\right) \geq \tau \right\} \quad (9)$$

where $\tau \in [0,1]$. Then, we calculate the MSE for a *1-linguistic rule* R_i^n as

$$\text{MSE}\left(E_i, R_i^{n(1)}\right) = \frac{\sum_{e^l \in E_i} (ey^l - s_i(ex^l))^2}{2 \cdot |E_i|} \quad (10)$$

with $s_i(ex^l)$ being the crisp output value obtained when the consequent of $R_i^{n(1)}$ is defuzzified. We should note that any other local error measure can be considered with no change in our methodology, such as the one shown in [30].

Remark 1: We should note that in this paper we consider $\tau = 0.5$ in order to emphasize the most influential examples responsability on the bad or good condition of the rule. That is, the neighbor examples which define the nearest decision surface induced by the rule prototype [20].

iii) *Select the 1-linguistic rules with bad performance.* Select those bad *1-linguistic rules* which are going to be expanded, making the difference from the good ones

$$\text{RB}_{\text{bad}}(1,n(1)) = \left\{ R_i^{n(1)} \big/ \text{MSE}\left(E_i, R_i^{n(1)}\right) \geq \alpha \cdot \text{MSE}(E_{\text{TDS}}, \text{RB}(1,n(1))) \right\} \quad (11)$$

$$\text{RB}_{\text{good}}(1,n(1)) = \left\{ R_i^{n(1)} \big/ \text{MSE}\left(E_i, R_i^{n(1)}\right) < \alpha \cdot \text{MSE}(E_{\text{TDS}}, \text{RB}(1,n(1))) \right\} \quad (12)$$

with α being a threshold that represents a percentage of the error of the whole RB, which determines the expansion of a rule. It may be adapted in order to have more or less expanded rules. It is noteworthy that this adaptation is not linear and, as a consequence, *the expansion of more rules does not ensure the decrease of the global error of the modeled system.* For example, $\alpha = 1.1$ means that a *1-linguistic rule* with an MSE a 10% higher than the MSE of the entire $\text{RB}(1,n(1))$ should be expanded.

Now, for each $R_i^{n(1)} \in \text{RB}_{\text{bad}}(1,n(1))$:

b) *Obtain the* $\text{DB}(2, 2\cdot n(1)-1)$. Create $\text{DB}_{x_j}(2, 2\cdot n(1)-1)$ for all input linguistic variables x_j $(j = 1,\ldots,m)$ and $\text{DB}_y\ (2, 2\cdot n(1)-1)$ for the output linguistic variable y.

i) *Select the 2-linguistic partition terms.* Select those terms from $\text{DB}(2, 2\cdot n(1)-1)$ that are going to be contained in the *2-linguistic rules* considered as the image of the previous layer bad rules.

Before describing this process and for the sake of clearness, we are going to refer to $\text{DB}(1,n(1))$ as $\text{DB}_{x_j}(1,n(1))$ $(j = 1,\ldots, m)$, meaning that it contains the *1-linguistic partition* where the input linguistic variable x_j takes values, and as $\text{DB}_y(1,n(1))$ for the output variable y. Even if all *1-linguistic partitions* contained in a $\text{DB}(1,n(1))$ have the same number of linguistic terms, they are defined over different domains corresponding to each linguistic variable.

For all linguistic terms considered in $R_i^{n(1)}$, i.e., $S_{ij}^{n(1)}$ defined in $\text{DB}_{x_j}(1,n(1))$ and associated to the linguistic variables x_j, select those terms $S_h^{2\cdot n(1)-1}$ in $\text{DB}_{x_j}(2, 2\cdot n(1) - 1)$ which significantly intersect them. Consequently, for $B_i^{n(1)}$ defined in $\text{DB}_y(1,n(1))$ and associated to the linguistic variable y, select those terms $B_i^{2\cdot n(1)-1}$ in $\text{DB}_y(2, 2\cdot n(1)-1)$ which significantly intersect them. That is, perform the selection of those terms of the *2-linguistic partition* that describe approximately the same subspace that the terms included in $R_i^{n(1)}$, but with a higher granularity level.

In this work we are going to consider that two linguistic terms have a "*significant intersection*" between each other, if the maximum cross level between their fuzzy sets in a linguistic partition overcomes a predefined threshold δ. In other words, the set of terms from the *2-linguistic partitions* for the expansion of the *1-linguistic rule* $R_i^{n(1)}$, are selected in the following way:

$$I\left(S_{ij}^{n(1)}\right) = \left\{ S_h^{2\cdot n(1)-1} \in \mathrm{DB}_{x_j}(2, 2\cdot n(1)-1) \Big/ \mathrm{Max}_{u\in U_j}\, \mathrm{Min}\left\{\mu_{S_{ij}^{n(1)}}(u), \mu_{S_h^{2\cdot n(1)-1}}(u)\right\} \geq \delta \right\} \quad (13)$$

$$I\left(B_i^{n(1)}\right) = \left\{ B_h^{2\cdot n(1)-1} \in \mathrm{DB}_y(2, 2\cdot n(1)-1) \Big/ \mathrm{Max}_{v\in V}\, \mathrm{Min}\left\{\mu_{B_i^{n(1)}}(v), \mu_{B_h^{2\cdot n(1)-1}}(v)\right\} \geq \delta \right\} \quad (14)$$

where $\delta \in [0, 1]$.

ii) *Combine the previously selected* m *sets* $I(S_{ij}^{n(1)})$ *and* $I(B_i^{n(1)})$ *by the following expression*:

$$I\left(R_i^{n(1)}\right) = I\left(S_{i1}^{n(1)}\right) \times \cdots \times I\left(S_{im}^{n(1)}\right) \times I\left(B_i^{n(1)}\right) \quad (15)$$

with $I(R_i^{n(1)}) \subset \mathrm{DB}(2, 2\cdot n(1)-1)$.

c) *Extract 2-linguistic rules from the combined selected 2-linguistic partition terms*. Produce a set of L *2-linguistic rules*, which are the expansion of the bad *1-linguistic rule* $R_i^{n(1)}$. This task is performed by an LRG-method, which takes $I(R_i^{n(1)})$ and the set of input–output data E_i as its parameters

$$\mathrm{CLR}\left(R_i^{n(1)}\right) = \mathrm{LRG-method}\left(I\left(R_i^{n(1)}\right), E_i\right) = \left\{R_{i_1}^{2\cdot n(1)-1}, \ldots, R_{i_L}^{2\cdot n(1)-1}\right\} \quad (16)$$

with $\mathrm{CLR}(R_i^{n(1)})$ being the image of the expanded linguistic rule $R_i^{n(1)}$, i.e., the candidates to be in the HRB from rule i.

3) **Summarization process**. *Obtain a joined set of candidate linguistic rules* (JCLR), performing the union of the group of the new generated *2-linguistic rules* and the former good performance *1-linguistic rules*

$$\mathrm{JCLR} = \mathrm{RB}_{\mathrm{good}}(1, n(1)) \bigcup \left(\bigcup_i \mathrm{CLR}\left(R_i^{n(1)}\right)\right) \quad (17)$$

with $R_i^{n(1)} \in \mathrm{RB}_{\mathrm{bad}}(1, n(1))$.

In the following, we show an example of the whole expansion process considering these linguistic partitions.

Let us consider $n(1) = 3$,

$$\mathrm{DB}_{x_1}(1,3) = \mathrm{DB}_{x_2}(1,3) = \mathrm{DB}_y(1,3) = \{S^3, M^3, L^3\};$$
$$\mathrm{DB}_{x_1}(2,5) = \mathrm{DB}_{x_2}(2,5) = \mathrm{DB}_y(2,5) = \{VS^5, S^5, M^5, L^5, VL^5\}$$

where

S Small;
M Medium;
L Large;
V Very.

Let us consider the following bad performance *1-linguistic rule* to be expanded:

$$R_i^3 : \text{IF } x_1 \text{ is } S_{i1}^3 \text{ and } x_2 \text{ is } S_{i2}^3 \text{ THEN } y \text{ is } B_i^3$$

where the linguistic terms are

$$S_{i1}^3 = S^3 \quad S_{i2}^3 = S^3 \quad B_i^3 = S^3$$

and the resulting sets I with $\delta = 0.5$ are

$$I\left(S_{i1}^3\right) = \{VS^5, S^5\}, \quad I\left(S_{i2}^3\right) = \{VS^5, S^5\},$$
$$I\left(B_i^3\right) = F(\cdot) \subseteq D_y(2,5)$$
$$I\left(R_i^3\right) = I\left(S_{i1}^3\right) \times I\left(S_{i2}^3\right) \times I\left(B_i^3\right).$$

Therefore, it is possible to obtain at most four *2-linguistic rules* generated by the LRG-method from the expanded rule R_i^3:

$$\mathrm{LRG}\left(I\left(R_i^3\right), E_i\right) = \left\{R_{i1}^5, R_{i2}^5, R_{i3}^5, R_{i4}^5\right\}.$$

This example is graphically shown in Fig. 3. In the same way, other bad performance neighbor rules could be expanded simultaneously.

Remark 2: We should note that in the latter example the value used for the parameter δ was 0.5. Each set $I(S_{ij}^3)$ that we consider in the example is a consequence of the use of this value in the expansion task of the rule R_i^3. Thus, the problem subspace resulting from that bad 1-linguistic rule expansion is the one represented by the small white square $(A \cup B \cup C \cup D)$ in Fig. 3. On the other hand, if we consider $\delta = 0.1$, the set of selected linguistic terms would be:

$$I\left(S_{i1}^3\right) = \{VS^5, S^5, M^5\}, \quad I\left(S_{i2}^3\right) = \{VS^5, S^5, M^5\}$$

and the said subspace would be composed of the union of the former small white square and the grey one.

C. Hierarchical Rule Base Selection Process

In the JCLR, where there are coexisting rules of two different hierarchical layers, it may happen that a complete set of *2-linguistic rules*, which replaces an expanded rule, does not produce good results. This means that there will be higher errors, as it is shown on the left hand side of Fig. 4. However, a subset of this set of *2-linguistic rules* may work properly, with less rules that have good cooperation between them, and with the good rules from the previous layer. This is shown on the right-hand side of Fig. 4. Thus, the JCLR set of rules generated may present redundant or unnecessary rules making the model using this HKB less accurate.

In order to avoid this fact, we will use a genetic linguistic rule selection process with the aim of simplifying the initial linguistic rule set by removing the unnecessary rules from it and generating an HKB with good cooperation. In this paper, we consider a genetic process [6], [12], [16] to put this task into effect, but any other technique could be considered

$$\mathrm{HRB} = \mathrm{Selection\ Process(JCLR)}. \quad (18)$$

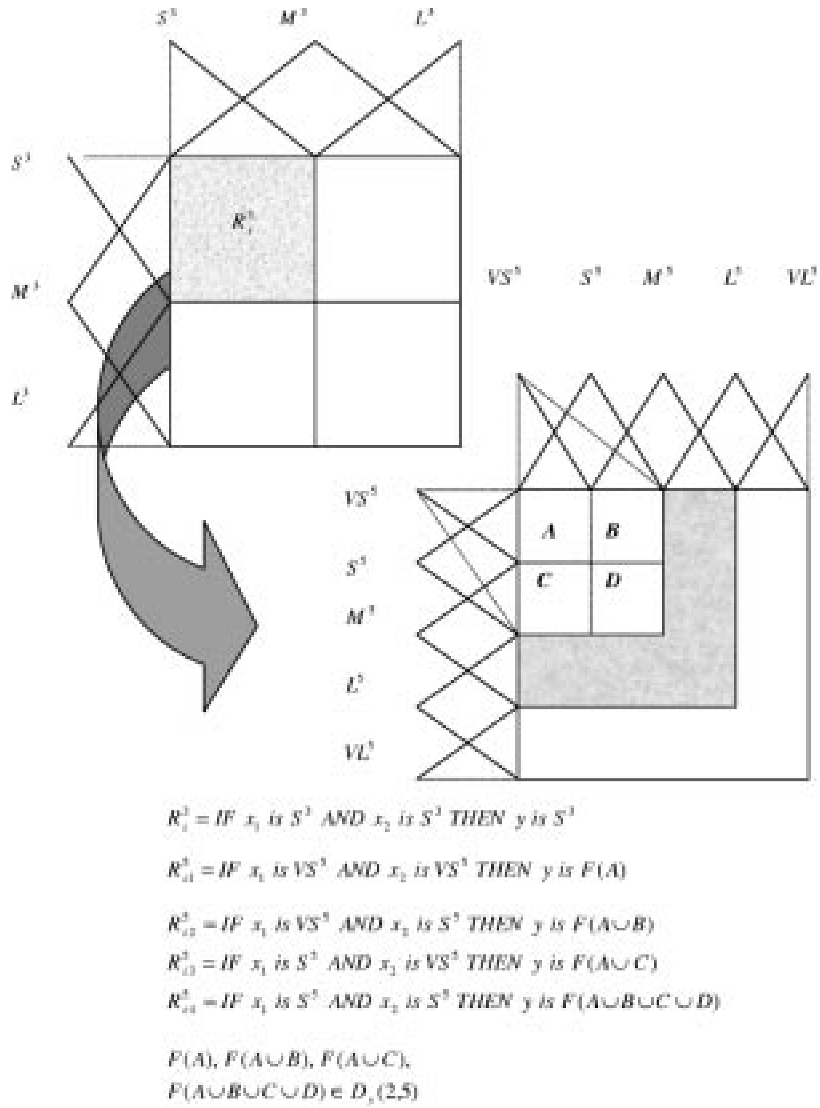

Fig. 3. Example of the HRB Generation Process.

Remark 3: It should be noted that the set JCLR is a unified set which contains all $(t+1)$-linguistic rules obtained from the simultaneous expansion of each bad t-linguistic rules, as well as the good ones from the same layer. Selection is performed once on this set during each iteration.

The genetic rule selection process [6], [12] is based on a binary coded genetic algorithm (GA), in which the selection of the individuals is performed using the stochastic universal sampling procedure together with an elitist selection scheme, and the generation of the offspring population is put into effect by using the classical binary multipoint crossover (performed at two points) and uniform mutation operators.

The coding scheme generates fixed-length chromosomes. Considering the rules contained in JCLR counted from 1 to z, an z-bit string $C = (c_1, \ldots, c_z)$ represents a subset of rules for the HRB, such that

$$\text{IF } c_i = 1 \text{ THEN } (R_i \in \text{HRB}) \, \text{ELSE} \, (R_i \notin \text{HRB}). \quad (19)$$

The initial population is generated by introducing a chromosome representing the complete previously obtained rule set, i.e., with all $c_i = 1$. The remaining chromosomes are selected at random.

As regards the fitness function $F(C_j)$, it is based on a global error measure that determines the accuracy of the FRBS encoded in the chromosome, which depends on the cooperation level of the rules existing in the HRB

$$\begin{aligned} F(C_j) &= \text{MSE}(E_{\text{TDS}}, \text{chromosome}) \\ &= \frac{\sum_{e^l \in E_{\text{TDS}}} (ey^l - s(ex^l))^2}{2 \cdot |E_{\text{TDS}}|} \end{aligned} \quad (20)$$

with $s(ex^l)$ being the output value obtained from RB encoded in the chromosome, when the input variable values are $ex^l = (ex_1^l, \ldots, ex_m^l)$, and ey^l is the known desired value.

We usually work with the MSE over a training data set, as it was defined in Section IV.B, although other measures may be used.

D. User Evaluation Process

It should be kept in mind that the level of precision which is obtained by applying the HSLR-LM is not fixed. However, this

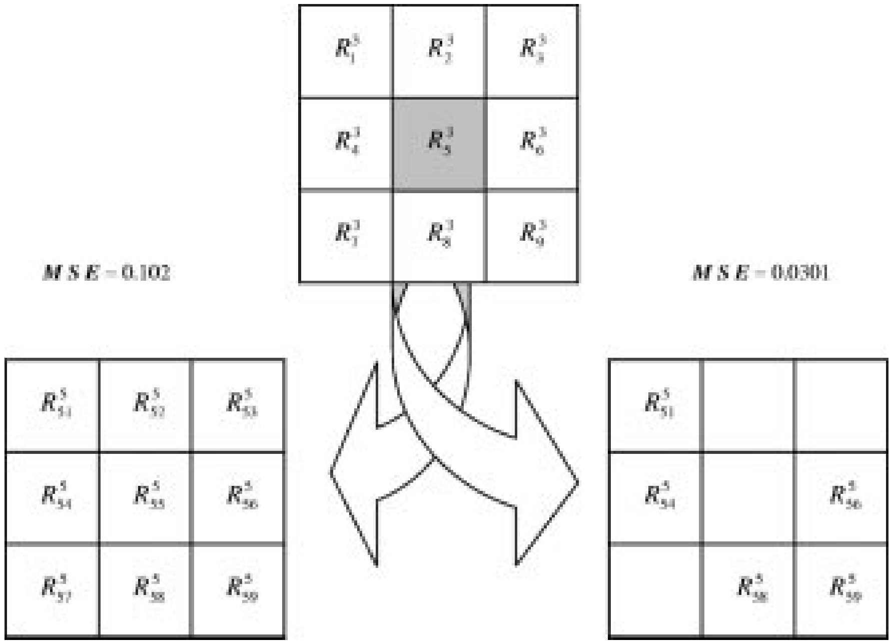

Fig. 4. Rule selection process.

methodology allows the user to adapt the level of precision to percentages of error suggested by an expert. This process depends on user's decisions, based on an error measure of the obtained model, and on the kind of problem to be modeled, to reach to a suitable set of rules which could perform the modeling task. From this point of view, the application of our methodology could also be considered as an user controlled iterative process. In this sense, the user could adapt the granularity of the initial linguistic partitions and/or the threshold which determines if an *1-linguistic rule* will be expanded into a set of *2-linguistic rules*, and apply again the methodology in order to obtain a better model.

This process works in this way: if the error measure of the obtained model (i.e., global error) does not satisfy the user requirements, then he can adapt the parameter α, item 2 in the HKB generation process, and/or reinitialize the process with a different granularity for the initial layer linguistic partition.

Finally, we want to point out that our methodology performs smooth refinements with small changes in order to improve the model. In the following, we will show an example of a difficult modeling real-world problem, in which the use of high granularity initial linguistic partitions does not improve the general error of the sample and what is more, gets it worse.

V. Examples of Application: Experiments and Analysis of Results

With the aim of analyzing the behavior of the proposed methodology, two real-world electrical engineering distribution problems in Spain have been selected [7], [23], [24].

The LRG-method considered for the previous experimentation is the one proposed by Wang and Mendel in [27], that we call as WM-method in the following. This method is briefly described in Appendix I.

As we have said, this methodology has been thought as a refinement of simple linguistic models, which uses an HKB of two layers, i.e., starting with an initial $\text{layer}(1, n(1))$, a $\text{layer}(2, 2 \cdot n(1) - 1)$ is created in order to extract the final system of linguistic rules.

For the sake of simplicity, in the following applications we are going to refer to those experiments produced by the HSLR-LM by the following notation:

$$\text{HSLR}(\text{LRG-method}, n(1), 2 \cdot n(1) - 1)$$

where $n(1)$, and $2 \cdot n(1) - 1$ are the initial and final granularity levels of the HKB, respectively, e.g., $\text{HSLR}(\text{WM}, 3, 5)$.

In addition, a reference to an application of WM is represented by the following expression:

$$\text{WM}(r)$$

with r being the granularity level of the linguistic partitions used in the method.

The results obtained in the experiments developed are collected in tables where $\#R$[1] stands for the number of rules of the corresponding HRB, MSE_{tra} and MSE_{tst} for the values obtained in the MSE measure computed over the training and test data sets, respectively. % indicates the relative error between two algorithms $((\epsilon_A - \epsilon_B/\epsilon_A) \cdot 100))$ [3], e.g., the percentage in which the WM-based model is improved by the HSLR. In the following experiments, we are going to compare the model generated by HSLR-LM, i.e., $\text{HSLR}(\text{WM}, n(1), 2 \cdot n(1) - 1)$, with the ones generated by $\text{WM}(n(1))$ and $\text{WM}(2 \cdot n(1) - 1)$.

A. The Electrical Engineering Distribution Problems

Two problems will be tackled: to relate some characteristics of certain village with the actual length of low voltage line contained in it, and to relate the maintenance cost of the network

[1]We should note the appearance of repeated rules—generated by the HKB generation process as a consequence of the overlapping produced in the selection of the $(t+1)$-*linguistic partition* terms—does not increase the computational cost of the process, because the rules are processed only once in the inference process and the result is multiplied by the number of times that it is repeated in the set of rules. As a consequence, those rules which are repeated are considered as a single one in the calculus of the complexity.

TABLE IV
NOTATION CONSIDERED FOR THE PROBLEM VARIABLES

Symbol	Meaning
x_1	Number of clients in population
x_2	Radius of i population in the sample
y	Line length, population i

installed in certain towns with some of their characteristics [7]. In both cases, it would be preferable that the solutions obtained verify another requirement: they have not only to be numerically accurate in the problem solving, but must be able to explain how a specific value is computed for a certain village or town. That is, it is interesting that these solutions are interpretable by human beings to some degree.

1) Computing the Length of Low Voltage Lines: Sometimes, there is a need to measure the amount of electricity lines that an electric company owns. This measurement may be useful for several aspects such us the estimation of the maintenance costs of the network, which was the main goal of the problem presented in Spain [7], [24]. High and medium voltage lines can be easily measured, but low voltage line is contained in cities and villages, and it would be very expensive to measure it. This kind of line used to be very convoluted and, in some cases, one company may serve more than 10 000 small nuclei. An indirect method for determining the length of line is needed.

Therefore, a relationship must be found between some characteristics of the population and the length of line installed on it, making use of some known data, that may be employed to predict the real length of line in any other village. We will try to solve this problem by generating different kinds of models determining the unknown relationship: linguistic, classical regression and neural models. To do so, we were provided with the measured line length, the number of inhabitants and the mean distance from the center of the town to the three furthest clients, considered as the radius of population i in the sample, in a sample of 495 rural nuclei [23], [24]. Our variables are named as shown in Table IV.

To compare regression techniques, neural modeling, and linguistic modeling, we have randomly divided the sample into two sets comprising 396 and 99 samples, labeled training and test, respectively.

The initial DB used for the HSLR-LM is constituted by three primary linguistic partitions formed by *three*, *four*, and *five linguistic terms* with triangular-shaped fuzzy sets giving meaning to them, i.e., $\mathrm{DB}(1,3)$, $\mathrm{DB}(1,4)$, and $\mathrm{DB}(1,5)$, respectively.

The initial linguistic term sets for the mentioned DBs are shown in the following:

$$\mathrm{DB}(1,3) = \{S^3, M^3, L^3\}$$
$$\mathrm{DB}(1,4) = \{\mathrm{VS}^4, S^4, L^4, \mathrm{VL}^4\}$$
$$\mathrm{DB}(1,5) = \{\mathrm{VS}^5, S^5, M^5, L^5, \mathrm{VL}^5\}$$

where

$$S = \text{Small}$$
$$M = \text{Medium}$$
$$L = \text{Large}$$
$$\mathrm{VS} = \text{Very Small}$$
$$\mathrm{VL} = \text{Very Large.}$$

TABLE V
PARAMETERS

Parameter	Decision
Generation	
δ -*(2·n-1)-linguistic partition* terms selector-	0.1
τ -used to calculate $E_{\mathcal{F}}$	0.5
α -used to decide the expansion of rule-	1.1
GA Selection	
Number of generations	500
Population size	61
Mutation probability	0.1
Crossover probability	0.6

TABLE VI
RESULTS OBTAINED IN THE LOW VOLTAGE ELECTRICAL APPLICATION CONSIDERING $\mathrm{HSLR(WM}, 3, 5)$

Method	MSE_{tra}	MSE_{tst}	$\%_{tra}$	$\%_{tst}$	$\#R$
WM(3)	594276	626566	69.88	73.29	**7**
WM(5)	298446	282058	40.03	40.67	13
HSLR(WM,3,5)	**178950**	**167318**			12

TABLE VII
RESULTS OBTAINED IN THE LOW VOLTAGE ELECTRICAL APPLICATION CONSIDERING $\mathrm{HSLR(WM}, 4, 7)$

Method	MSE_{tra}	MSE_{tst}	$\%_{tra}$	$\%_{tst}$	$\#R$
WM(4)	301732	270747	43.72	28.82	**10**
WM(7)	222622	240018	23.72	19.70	23
HSLR(WM,4,7)	**169799**	**192714**			25

TABLE VIII
RESULTS OBTAINED IN THE LOW VOLTAGE ELECTRICAL APPLICATION CONSIDERING $\mathrm{HSLR(WM}, 5, 9)$

Method	MSE_{tra}	MSE_{tst}	$\%_{tra}$	$\%_{tst}$	$\#R$
WM(5)	298446	282058	47.14	36.01	**13**
WM(9)	197613	283645	20.16	36.36	29
HSLR(WM,5,9)	**158420**	**180488**			30

The parameters used in all of these experiments are listed in Table V.

The results obtained with our HSLR-LM starting from different granularities in the first layer DB partitions are shown in Tables VI, VII and VIII.

The structure of the simple model obtained from $\mathrm{HSLR(WM}, 3, 5)$ can be seen in Fig. 5, where the rules are numbered from left to right, and from up to down. The explanation of this figure can be found in Table IX (each learning process (P), rule condition (RC), hierarchy level (HL), rule number (RN) and corresponding mark (M) of Fig. 5 are explicit, $R_{i,j}^{2\cdot n(1)-1}$ is the rule j of layer 2, which is the image of the expanded rule $R_i^{n(1)}$ of layer 1.

Once we have shown the behavior of the linguistic models designed individually, we are going to compare their accuracy with the remaining techniques considered. Table X shows the results obtained by them and the best ones obtained by our HSLR-LM as well. To apply classical regression, the parameters of the polynomial models were fit by Levenberg–Marquardt, while exponential and linear models were fit by linear least squares. The

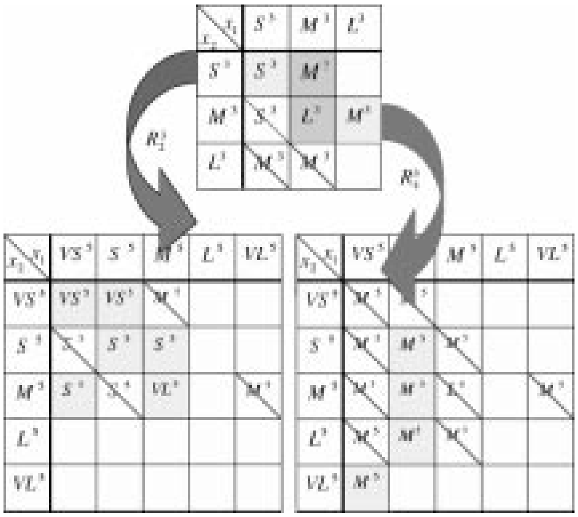

Fig. 5. $\mathrm{HSLR}(\mathrm{WM}, 3, 5)$.

TABLE IX
ACTIONS PERFORMED BY HSLR-LM ON THE RULE STRUCTURE

P	RC	HL	RN	M
HKB Gen. Proc.	not generated			white sqr.
	selected to be expanded	L 1	$R^3_2; R^3_4$	dark grey sqr.
HRB Sel. Proc.	selected to belong to the final HRB	L 1	$R^3_1; R^3_5$	light grey sqr.
		L 2/ R.2	$R^5_{2,1}; R^5_{2,2}; R^5_{2,5}; R^5_{2,6}; R^5_{2,7}; R^5_{2,9}$	
		L 2/ R.4	$R^5_{44}; R^5_{47}; R^5_{4,11}; R^5_{4,13}$	
HRB Sel. Proc.	discarded	L 1	$R^3_3; R^3_6; R^3_7$	cross sqr.
		L 2/ R 2	$R^5_{2,3}; R^5_{2,4}; R^5_{2,8}; R^5_{2,10}$	
		L 2/ R.4	$R^5_{4,1}; R^5_{4,2}; R^5_{4,3}; R^5_{4,5}; R^5_{4,6}; R^5_{4,8}; R^5_{4,9}; R^5_{4,10}; R^5_{4,12}$	

TABLE X
RESULTS OBTAINED IN THE LOW VOLTAGE ELECTRICAL APPLICATION COMPARED WITH OTHER TECHNIQUES

Method	MSE_{tra}	MSE_{tst}	*Complexity*
Linear	287775	209656	7 n., 2 par.
Exponential	232743	197004	7 n., 2 par
2th order pol.	235948	203232	25 n., 2 par.
3rd order pol.	235934	202991	49 n., 2 par.
Percep. (2-25-1)	**169399**	**167092**	102 par.
HSLR(WM,3,5)	178950	167318	12 rules

multilayer perceptron was trained with the QuickPropagation algorithm. The number of neurons in the hidden layer was chosen to minimize the test error [7], [24].

2) Computing the Maintenance Costs of Medium Voltage Line: We were provided with data concerning four different characteristics of the towns (see Table XI) and their minimum maintenance cost in a sample of 1059 simulated towns. In this case, our objective was to relate the last variable (maintenance costs) with the other four ones by applying the same modeling techniques considered for the previous problem. Numerical results will be compared next.

TABLE XI
NOTATION CONSIDERED FOR THE PROBLEM VARIABLES

Symbol	Meaning
x_1	Sum of the lengths of all streets in the town
x_2	Total area of the town
x_3	Area that is occupied by buildings
x_4	Energy supply to the town
y	Maintenance costs of medium voltage line

TABLE XII
RESULTS OBTAINED IN THE MEDIUM VOLTAGE ELECTRICAL APPLICATION CONSIDERING $\mathrm{HSLR}(\mathrm{WM}, 3, 5)$

Method	MSE_{tra}	MSE_{tst}	$\%_{tra}$	$\%_{tst}$	$\#R$
WM(3)	150545	125807	84.95	81.06	**27**
WM(5)	70908	77058	68.05	69.09	64
HSLR(WM,3,5)	**22653**	**23817**			97

TABLE XIII
RESULTS OBTAINED IN THE MEDIUM VOLTAGE ELECTRICAL APPLICATION CONSIDERING $\mathrm{HSLR}(\mathrm{WM}, 5, 9)$

Method	MSE_{tra}	MSE_{tst}	$\%_{tra}$	$\%_{tst}$	$\#R$
WM	70908	77058	82.95	82.10	**64**
WM	32191	33200	62.44	58.46	130
HSLR(WM,5,9)	**12089**	**13791**			524

TABLE XIV
RESULTS OBTAINED IN THE MEDIUM VOLTAGE ELECTRICAL APPLICATION COMPARED WITH OTHER TECHNIQUES

Method	MSE_{tra}	MSE_{tst}	*Complexity*
Linear	164662	36819	17 n., 5 par.
2th order pol.	103032	45332	77 n., 15 par.
Percep. (4-5-1)	86469	33105	35 par.
HSLR(WM,3,5)	**22653**	**23817**	97 rules

The sample have been randomly divided into two sets comprising 847 and 212 samples, 80 and 20 percent of the whole data set, labeled training and test, respectively. Our variables are named as shown in Table XI.

The initial DB used for the design methods is initialized as in the former problem for $\mathrm{DB}(1,3)$, and $\mathrm{DB}(1,5)$, as well as the other parameters which are listed in Table V. The different results obtained are shown in Tables XII, XIII, and XIV.

In view of the results obtained in the above experiments, we should remark some important conclusions. *From the accuracy point of view*, the different models generated from our process clearly outperform the WM-method ones in all granularity level linguistic partitions and in both electrical problems. They also outperform classical regression in the approximation of both data sets, training and test.

In the first problem, the linguistic model generated from $\mathrm{HSLR}(\mathrm{WM}, 3, 5)$ is less accurate than the neural one in the approximation of the training set, but we should note that they have almost the same value for the resulting test error. Therefore, this model approximates well the real system modeled and, moreover, it has the advantage of being much more interpretable than the neural model. In the second problem,

TABLE XV
RESULTS OBTAINED IN THE LOW VOLTAGE ELECTRICAL APPLICATION CONSIDERING $\mathrm{HSLR}(\mathrm{THR}, 3, 5)$

Method	MSE_{tra}	MSE_{tst}	$\%_{tra}$	$\%_{tst}$	$\#R$
THR(3)	266369	248257	34.66	29.73	**7**
THR(5)	218857	217847	20.48	19.93	25
HSLR(THR,3,5)	**174020**	**174428**			26

both training and test errors of the neural model are clearly outperformed by the HSLR model.

VI. Key Points of the Two-Level HSLR Learning Methodology

In this section, we will point out the most important features of HSLR-LM, highlighting its independence from the LRG-method to be used, the use of the α parameter to set the desired balance between accuracy and description of the generated HSLR, the importance of the selection process performed and some other aspects related to the methodology performance. Finally, its local treatment of the problem subspaces is linked with the accuracy and description paradigm.

A. Independence From the LRG-Method

As said, our methodology was thought as an strategy to improve simple linguistic models. In Section V, we have chosen the WM-method as an example of those kinds of simple methods. However, we could select any other inductive method, based on the existence of a set of input–output data E_{TDS} and a previously defined DB$(1, n(1))$. In order to illustrate this situation, we are going to show an experiment where an HSLR is obtained from another LRG-method. For the present application, we have selected the LRG-method proposed by Thrift [26], that we call as $\mathrm{THR}(r)$, with r being the granularity of the linguistic partitions considered. This method is briefly described in Appendix II.

The results obtained by the application of our methodology to the first electrical problem using the THR-method is shown in Table XV and % indicates the percentage in which the THR-based model is improved by the HSLR.

We can observe again that the HSLR-LM has outperformed the basic LRG-method, the THR-method in this case. This most accurate model was obtained by just adding one more rule to the model obtained by THR(5), with a significant improvement of the twenty percent both in $\mathrm{MSE}_{\mathrm{tra}}$ and $\mathrm{MSE}_{\mathrm{tst}}$.

In this view, we confirm the qualities of the HSLR-LM as a good strategy to obtain a refinement of simple models, based on performing few changes to the system structure.

B. Setting the Balance Between Accuracy and Description

In previous experiments, we have compared the accuracy and complexity of those linguistic models generated from our HSLR-LM based on an expansion factor α equal to 1.1. This means that those rules which overcomes the MSE of the whole RB$(1, n(1))$ in a 10% are considered as bad ones, and should be expanded. In this section, we are going to analyze the influence of other possible values for this factor and how it works as a regulator between the accuracy and the description of the system.

TABLE XVI
RESULTS OBTAINED IN THE LOW VOLTAGE ELECTRICAL APPLICATION BY $\mathrm{HSLR}(\mathrm{WM}, 3, 5)$ USING DIFFERENT VALUES FOR α

Method	α	MSE_{tra}	MSE_{tst}	$\#R$
WM(3)		594276	626566	**7**
WM(5)		298446	282058	13
HSLR(WM,3,5)	0.05	205120	187772	30
	0.5	204977	204467	22
	0.9	**178081**	170816	20
	1.1	178950	**167318**	12
	1.9	178950	**167318**	12

TABLE XVII
RESULTS OBTAINED IN THE MEDIUM VOLTAGE ELECTRICAL APPLICATION BY $\mathrm{HSLR}(\mathrm{WM}, 3, 5)$ USING DIFFERENT VALUES FOR α

Method	α	MSE_{tra}	MSE_{tst}	$\#R$
WM(3)		150545	125807	**27**
WM(5)		70908	77058	64
HSLR(WM,3,5)	1.1	**22653**	**23817**	97
	1.5	29336	29657	59
	1.9	29336	29657	59
	3.5	33117	37290	58

TABLE XVIII
RESULTS OBTAINED IN THE MEDIUM VOLTAGE ELECTRICAL APPLICATION BY $\mathrm{HSLR}(\mathrm{WM}, 5, 9)$ USING DIFFERENT VALUES FOR α

Method	α	MSE_{tra}	MSE_{tst}	$\#R$
WM(5)		70908	77058	**64**
WM(9)		32191	33200	130
HSLR(WM,5,9)	1.1	12089	13791	524
	1.5	**11643**	**11314**	420
	1.9	12387	12999	388
	5.5	32292	33151	121

In Tables XVI, XVII and XVIII, we analyze different values for α[2] in both electrical problems.

As can be seen in the above results, the algorithm seems to be robust for any value of α, in the sense that good results are obtained considering many different values for this parameter.

Anyway, some special features could be remarked as regards the α setting. As a general rule, when α grows up, the system complexity decreases, i.e., less rules are finally obtained.

However, an increase on the number of rules does not always ensure a decrease on the model error, MSE. This fact is clearly seen in the results obtained in Tables XVI and XVIII.

As said in [8], it is not always true that a linguistic model with a high number of rules performs better than another with a lesser number of them, since the accuracy of the FRBS does not only depend on the number of rules in the RB but also on the cooperation among them.

From this point of view, parameter α can be considered to design models with different balance between accuracy and description (of course, the lower the number of rules, the more descriptive the system). For example, we find a good balance in Table XVI, where the most accurate model is obtained for the low voltage problem by means of the HSLR-LM, which is

[2]In Tables XVII and XVIII, we have not performed experiments with values of α lower than 1.1 because of the complexity of the problem.

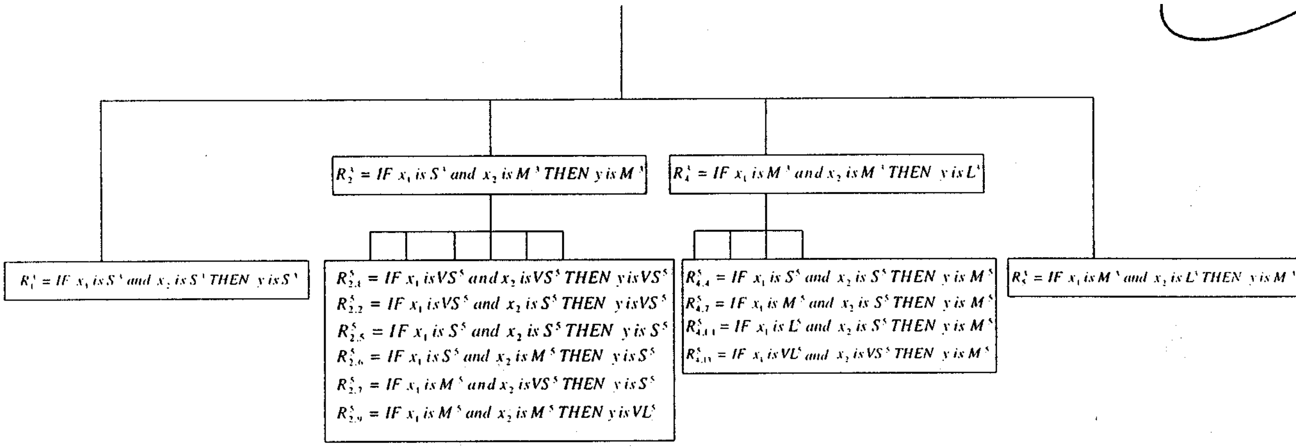

Fig. 6. Hierarchical clustering view of the HRB.

composed of only 12 rules. Notice that our model has one less rule than WM(5) while having a performance improvement of approximately a 40% both in $\mathrm{MSE_{tra}}$ and in $\mathrm{MSE_{tst}}$.

This idea can also be observed in the results shown in Table XVII as regards the medium voltage electrical application. Here, the user can decide between two models with a different treatment of the description-accuracy tradeoff. The model obtained when considering $\alpha = 1.1$ is the most accurate one, being more or less a 68% more accurate than the one obtained with WM(5). This would be the best choice when the accuracy is preferred to the description. However, HSLR-LM model is less interpretable than WM-method one since it has 33 more rules.

When a compromise solution between accuracy and description is preferred, the models obtained from HSLR-LM with $\alpha = 1.9$ and $\alpha = 3.5$ would be two very good solutions. They both are simpler than the model generated by WM(5) (59 and 58 rules, respectively, against 64) and outperform it by more or less a 60% and a 52%, respectively.

On the other hand, Table XVIII shows a different way to deal with the accuracy-description tradeoff. Signicantly more accurate models are obtained for the latter problem using higher granularity level initial partitions like five. Of course, the models generated by HSLR-LM starting from these partitions are very complex (from 121 to 524 rules) and thus very difficult to be interpreted. This would be the choice if the accuracy was definitively the only model requirement.

Finally, coming back to the discussion about the interpretability of the generated models, we should note that, when dealing with HSLRs, the system description level can not only be measured by the number of rules but also by the way they are represented. The HKB gives an order which can be used in the sense of interpretability. That is, human beings can not understand a hundred of different rules, but can associate a group of them with an specific task and deal with more general and subsumed rule sets. This basically suggests a hierarchical clustering point of view of the FRBSs, which gives a more interpretable view of HSLRs as it is ilustrated in Fig. 6 where the RB of the HSLR finally obtained from $\mathrm{HSLR(WM, 3, 5)}$ (see Fig. 5) is represented in the bottom level.

What is more, the order proposed in the HDB can be used to obtain an hierarchical extensionality measure of similarity to be used in grouping most undistinguishable fuzzy rules [17], [18], providing a theoretical background to the interpretability of HSLRs.

C. Influence of the Methodology Components

As was said by Goldberg [11], subtle integration of the abstraction power of fuzzy systems and the innovating power of genetic systems requires a design sophistication that goes further than putting everything together. That is, hybridizing in hierarchical models does not only involves putting rules with different granularities in the same bag. In this section, we will explore different aspects of the HSLR-LM which allow us to know why does it works and which are its future perspectives.

In the first section, we will consider the importance of the rule selection process in HSLR-LM, and in the second one, we will mine into HSLR-LM in order to discover what other things make it a successful methodology.

1) The Influence of the Rule Selection Process in the HSLR Summarization: One of most interesting features of an FRBS is the interpolative reasoning it develops, which is a consequence of the cooperation among the linguistic rules composing the KB. As said in Section IV-C, the set of rules generated by an LRG-method may present redundant or unnecessary rules which make the fuzzy model using this KB less accurate. This fact becomes more serious in an HSLR, where there are coexisting rules with different granularity levels. To deal with this problem, we have introduced an RB selection process in order to choose a subset of linguistic rules that properly work, i.e., with less rules that have good cooperation between them. This process organizes the incoming results from the rule generation tasks based on some "interestingness" criteria in order to provide a more understandable and compact representation.

To perform the said summarization, we have considered a GA although we could have chosen any other optimization method. The summarization criteria is represented in the fitness function $F(C_j)$, which is based on a measure of the global performance of the FRBSs. Basically, this measure shows the cooperation level of the candidate rules of the RB, which was the MSE on the training set in this work. Even so, any other proper measure

TABLE XIX
RESULTS OBTAINED BY HSLR-LM IN THE LOW VOLTAGE ELECTRICAL APPLICATION CONSIDERING $\alpha = 1.1$. WITH AND WITHOUT RULE SELECTION PROCESS

Method	MSE_{tra}	MSE_{tst}	$\#R$
HSLR(WM,3,5)/Selection	82422	68713	166
HSLR(WM,3,5)	**22653**	**23817**	**97**

TABLE XX
RESULTS OBTAINED BY HSLR-LM IN THE MEDIUM VOLTAGE ELECTRICAL APPLICATION CONSIDERING $\alpha = 1.1$. WITH AND WITHOUT RULE SELECTION PROCESS

Method	MSE_{tra}	MSE_{tst}	$\#R$
HSLR(WM,3,5)/Selection	501052	468037	26
HSLR(WM,3,5)	**178950**	**167318**	**12**

TABLE XXI
RESULTS OBTAINED IN THE MEDIUM VOLTAGE ELECTRICAL APPLICATION BY $\mathrm{HSLR}(\mathrm{WM}, 3, 5)$ USING DIFFERENT NUMBER OF GA GENERATIONS

α	MSE_{tra}	MSE_{tst}	$\#R$	*Iterations*
1.1	22653	23817	97	1000
1.1	**22358**	**23755**	**84**	2000
1.9	**29336**	29657	59	1000
1.9	**29336**	**29604**	**58**	2000

TABLE XXII
RESULTS OBTAINED IN THE LOW VOLTAGE ELECTRICAL APPLICATION USING AN IMPROVED FITNESS FUNCTION

Method	α	MSE_{tra}	MSE_{tst}	$\#R$
HSLR(WM,3,5)	1.1	**178950**	167318	12
(I)HSLR(WM,3,5)	1.1	180111	**166210**	**11**

TABLE XXIII
RESULTS OBTAINED IN THE MEDIUM VOLTAGE ELECTRICAL APPLICATION USING DIFFERENT VALUES FOR α AND AN IMPROVED FITNESS FUNCTION

Method	α	MSE_{tra}	MSE_{tst}	$\#R$
HSLR(WM,3,5)	1.1	**22358**	**23755**	84
(I)HSLR(WM,3,5)	1.1	22557	24679	**69**
HSLR(WM,3,5)	1.9	**29336**	**29604**	58
(I)HSLR(WM,3,5)	1.9	32223	34504	**51**

TABLE XXIV
RESULTS OBTAINED IN THE MEDIUM VOLTAGE ELECTRICAL APPLICATION USING DIFFERENT VALUES FOR α AND DISTINGUISHING REPEATED RULES

Method	α	MSE_{tra}	MSE_{tst}	*Complexity*	
				$\#r$	$\#d$
HSLR(WM,3,5)	1.1	**22358**	**23755**	172	84
(I)HSLR(WM,3,5)	1.1	22557	24679	148	**69**
HSLR(WM,3,5)	1.9	**29336**	**29604**	106	58
(I)HSLR(WM,3,5)	1.9	32795	33842	74	**49**

based on different interesting criteria could be used. The importance of this process is shown in Tables XIX and XX.

From the accuracy point of view, the hierarchical models with rule selection clearly outperform the ones without it in the approximation of both data sets. Considering the complexity of the models generated, the models which perform a rule selection task become the simpler ones.

Unfortunately, although GAs are a robust technique, sometimes they can not avoid to fall in local minima in strongly multimodal search surfaces like the one corresponding to multiple granularity fuzzy rules. On the one hand, this problem could be solved by relaxing some parameters of the algorithm, like the population size or the number of generations, as can be seen in Table XXI.

In fact, the latter table does not only shows a reduction in the MSE but also an interesting decrease in the complexity of the learned model which reveals that, sometimes, the GA does not select the minimum number of rules and that it could be improved. To do so, we introduce a modification of the fitness function of the GA which is a trade-off solution between complexity and accuracy of the system modeled [16].

Let consider the following function $F'(C_j)$ which penalizes those RBs with a high number of rules in the following way:

$$F'(C_j) = w_1 \cdot F(C_j) + w_2 \cdot N_{\text{rules}} \tag{21}$$

with $F(C_j)$ being the fitness function—based on the MSE—used in Subsection IV.C, N_{rules} being the number of rules of that RB, and with w_1 and w_2 being the weights of the terms of the function. In the present experiments, these constants are initialized in the following way [8]:

$$w_1 = 1.0; \quad w_2 = 0.1 \cdot \frac{\text{MSE}_{\text{initial}}}{N_{\text{initial rules}}}$$

with $\text{MSE}_{\text{initial}}$ and $N_{\text{initial rules}}$ being the error and the amount of rules of the original RB to be summarized, respectively.

Tables XXII and XXIII show some results obtained using the modified fitness function in the rule selection process with both examples considered in this paper. To do so, we introduce a new notation in order to make the difference between the former and these new experiments. It consists of adding the symbol (I) to the former notation, e.g., $(I)\,\mathrm{HSLR}(\mathrm{WM}, 3, 5)$.

As we expected, the new fitness function allows us to generate less complex models and performs a tradeoff between complexity and accuracy. Moreover, sometimes it also works as a prunning strategy that could prevent the system overfitting (see the HSLR obtained in Table XXII).

2) Learning More Features by Mining Into HSLRs: In the last section we corroborated that selection plays a fundamental role in systems with multiple granularity partitions. We have also seen that not all the rules were discarded by the GA process, and we showed at least two ways to improve its performance. In this section, we will analyze which are the rules that the GA discards and why some of them are still preserved.

Let us first consider what we noted at the begining of Section V about repeated rules. There, we said that some repeated rules generated by the rule generation process, specifically by the "significant intersection" criteria of terms selection, also belonged to the JCLR set of rule candidates. Surprisingly, some of them were not eliminated by the GA algorithm, even by the use of techniques like the one introduced in the last section. Consider Table XXIV as an example of that, where $\#r$ represents the extracted rules from the selection process and $\#d$ the corresponding number of different rules.

This fact drives us to analyze what other factors, different from granularity, make influence in the development of hierarchical models. To do so, we mined into HSLRs and discovered some other interesting features of their components as the ones that we list in the following.

- *Weighted reinforced linguistic rules*
  As said, repeated rules appear because of the overlapping of the expanded rule images, which is produced by low values of the parameter δ. Considering an HRB and our present methodology, this rule repetition is produced by the generation of more than one copy of a rule in the same layer, as shown in Fig. 5 with the *t-linguistic rule*

  IF x_1 is M^5 and x_2 is VL^5 THEN y is M^5

  which is both derived from the expansion of R_2^3 and R_4^3.
  Once those repeated rules are generated, they are given to the selection process. This process has the chance to eliminate all those redundant rules but it has been seen that sometimes it preserves some of them. Although in the previous case the repeated rules were discarded, some other times this kind of rules are preserved reinforcing their importance in those subspaces where they take place (see Table XXIV).
- *Double-consequent linguistic rules*
  As a result of the use of our approach, we can observe that some of the learned rules have multiple consequents (Fig. 5). As was introduced in [9], this phenomenon is an extension of the usual linguistic model structure which allows the KB to present rules where each combination of antecedents may have two or more consequents associated. We should note that this operation mode does not constitute an inconsistency from the interpolative reasoning point of view but only a shift of the main labels making that the final output of the rule lie in an intermediate zone between them both. Hence, it may have the following linguistic interpretation. Let us consider that the specific combination of antecedents of Fig. 5, "x_1 is S^5 and x_2 is M^5", has two different consequents associated, S^5 and M^5. From a linguistic modeling point of view, the resulting double-consequent rule may be interpreted as follows:

IF x_1 is S^5 and x_2 is M^5 THEN y is between S^5 and M^5.

These approaches enrich the representational power of fuzzy rules allowing different kinds of rules to belong to the HRB. Moreover, they postpone the selecting rule decisions until the sumarization process is performed, considering the best cooperation between them.

As seen, not only the different granularity rules make influence in the model performance. There are many other complementary improvements that should be taken into account in order to obtain more accurate models. In the next section, we will complement the current features by considering a new reinforcement strategy. There, we will evaluate these models and analyze the future extensions of the methodology.

D. Local Processing in HSLRs

Finally, in this section, we will explore the locality of HSLR-LM in the expansion of the linguistic rules, comparing its operation mode with the global approach introduced by Ishibuchi *et al.* in [16]. To do so, this section is divided in two parts. First, HSLR-LM in its current form is directly compared with the other approach, in order to analyze the influence of the local or global processing. Then, a new capability that is present in Ishibuchi *et al.*'s process and not in HSLR-LM will be introduced in the latter in order to improve its performance.

1) Local Versus Global Rule Expansion: As was pointed out in Section II-B, there is another method which also performs a multigranular treatment of linguistic rules. This method, introduced by Ishibuchi et al. in [16], obtains a set of fuzzy rules by creating several linguistic partitions with different granularity levels, generating the complete set of linguistic rules in each of these partitions, taking the union of all of these sets, and finally performing a genetic rule selection process on the whole rule set. For the sake of simplicity, even if it was not presented as a hierarchical process, in this section we will adapt it and refer to this method as a global HSLR learning methodology (G-HSLR-LM), in order to distinguish it from our local approach (HSLR-LM).

Although G-HSLR-LM was designed to construct a fuzzy classification system, and the main purpose of the HSLR-LM proposed in this paper is to perform linguistic modeling, some interesting coincidences and differences have been found between them. Let us first consider Table XXV which shows a common notation for both hierarchical methodologies in order to clarify their similarities and differences. We should remember that $\mathrm{CLR}(R_i^{n(1)})$ stands for the image of the expanded bad linguistic rule $R_i^{n(1)}$, which joined to the former good performance *1-linguistic rules* constitute the set of candidate linguistic rules to be in the final HRB.

In the following, we will consider both methodologies in order to study their features and evaluate their performance.

- While HSLR-LM locally expands those rules which perform a bad modeling in some subspaces of the problem, G-HSLR-LM performs the same task in a global way, i.e., it expands all rules in all granularity levels.
- Both methods perform a genetic rule selection to extract the set of rules which best cooperates between them, i.e., the HRB, but on a different rule set. We should note that, in order to allow the comparison between both hierarchical methods, the fitness defined in Section IV-C was used in the GA for both approaches.

Tables XXVI and XXVII show results obtained by the global method with and without the rule selection process, in order to evaluate its influence. % indicates the percentage in which G-HSLR-LM is improved by HSLR-LM.

In view of the results obtained, it can be seen than our hierarchical methodology, HSLR-LM, which is based on a local rule expansions, obtains better results than G-HSLR-LM in terms of accuracy in both applications.

As regards the complexity of the models obtained, and thus, its interpretability, HSLR-LM generates the simplest model for the low voltage application, with three less rules than G-HSLR model (12 against 15), while the model obtained from the latter methodology is seven rules simpler (51 against 58) than ours

TABLE XXV
LOCAL AND GLOBAL SELECTION PROCESSES

HSLR-LM	$HRB = Selection\ Process$ $(RB_{good}(t,n(t)) \cup (\cup_i CLR(R_i^{n(1)})))$
G-HSLR-LM	$HRB = Selection\ Process$ $(RB(t,n(t)) \cup RB(t+1,n(t+1))))$

TABLE XXVI
RESULTS OBTAINED IN THE LOW VOLTAGE ELECTRICAL APPLICATION CONSIDERING $\alpha = 1.1$

Method	MSE_{tra}	MSE_{tst}	$\%_{tra}$	$\%_{tst}$	$\#R$
G-HSLR(WM,3,5)	**177735**	180721	-0.68	7.41	15
HSLR(WM,3,5)	178950	**167318**			**12**

TABLE XXVII
RESULTS OBTAINED IN THE MEDIUM VOLTAGE ELECTRICAL APPLICATION CONSIDERING $\alpha = 1.9$

Method	MSE_{tra}	MSE_{tst}	$\%_{tra}$	$\%_{tst}$	$\#R$
G-HSLR(WM,3,5)	34197	35827	14.21	17.36	**51**
HSLR(WM,3,5)	**29336**	**29604**			58

in the medium voltage one. However, this low complexity increase is justified by a very significant modeling error decrease: our model is approximately a 14% better in $\mathrm{MSE_{tra}}$ and a 17% better in $\mathrm{MSE_{tst}}$.

Moreover, we must have in mind another advantage of our methodology versus G-HSLR-LM: the fact that in HSLR-LM there is a parameter available, α, that allows the user to establish the desired balance between accuracy and description in the generated model.

2) Introducing Ishibuchi et al.'s Rule Reinforcement in HSLR-LM: Analyzing more deeply the operation mode of Ishibuchi et al.'s method, we can observe that G-HSLR-LM allows the HSLR derived from it to present both the expanded rule and some of the rules composing its image in the next layer RB. This is a consequence of the global expansion it performs and results in a reinforcement of the expanded rule. A rule reinforcement is a refinement of the action of a rule in the subspace where it is defined, allowing the maintenance of the rule itself, which produces a more flexible HRB structure.

Since HSLR-LM directly substitutes the expanded rule by its image, there is no possibility for the previous kind of reinforcement. As introduced in Section VI-C–2, we found two different reinforcements in HSLR-LM: *weighted reinforced linguistic rules* and *double-consequent linguistic rules*. These reinforcements were applied on the whole subspace of the rule and produced a global refinement action. This suggests that only the same layer linguistic rules participate on the reinforcement process, i.e., same layer rules could model a specific subspace of the problem.

However, a different kind of reinforcement, as a consequence of combining the global and local approaches, can be obtained by performing a local refinement in a specific part of the rule subspace. That is, *hierarchical reinforced linguistic rules* are obtained where the reinforcement is produced by allowing not only the image of the expanded rule but also the expanded rule itself to be considered in the selection process (as done in G-HSLR-LM). Thus, it gives the selection process the chance to perform a more accurate search in the solution space in order to obtain the most accurate HRB. This approach does not eliminate the concept of "replacement" of the expanded rule, but extends it allowing the selection process to eliminate that rule when it cooperates bad with the rest of the rules.

This approach, resulting from incorporating a capability of G-HSLR-LM that was not previously present in HSLR-LM, extends the former reinforcements allowing different granularity rules to model specific subspaces of the problem, i.e., it allows the system to perform local refinement actions.

To evaluate the different alternatives described in the last and the present subsections, Tables XXVIII, XXIX and XXX show results for the two applications considered comparing both local (including the new capability) and global methodologies. To do so, the following notation is considered to refer to the use of hierarchically reinforced rules $\mathrm{HSLR-HR(LRG-method},$ $n(1), 2\cdot n(1)-1)$.

Some conclusions related to the features of the HSLR-LM can be drawn from the obtained results.

- As may be seen *from the accuracy point of view*, the linguistic models generated from $(I)\mathrm{G-HSLR(WM},3,5)$ and $\mathrm{G-HSLR(WM},3,5)$ are clearly outperformed by the the local hierarchical linguistic models in the approximation of the test sets in every case and in most of the

TABLE XXVIII
RESULTS OBTAINED IN THE LOW VOLTAGE ELECTRICAL APPLICATION CONSIDERING $\alpha = 1.1$

Method	MSE_{tra}	MSE_{tst}	$\#R$
HSLR(WM,3,5)	178950	167318	**12**
HSLR-HR(WM,3,5)	**175619**	**162873**	**12**
G-HSLR(WM,3,5)	177735	180721	15
(I)HSLR(WM,3,5)	180111	166210	11
(I)HSLR-HR(WM,3,5)	**176781**	**161764**	**10**
(I)G-HSLR(WM,3,5)	179016	185805	12

TABLE XXIX
RESULTS OBTAINED IN THE MEDIUM VOLTAGE ELECTRICAL APPLICATION CONSIDERING $\alpha = 1.1$.

Method	MSE_{tra}	MSE_{tst}	$\#R$
HSLR(WM,3,5)	22358	23755	84
HSLR-HR(WM,3,5)	**20588**	**22583**	86
G-HSLR(WM,3,5)	34197	35827	**51**
(I)HSLR(WM,3,5)	22557	24679	69
(I)HSLR-HR(WM,3,5)	**20752**	**21005**	67
(I)G-HSLR(WM,3,5)	32976	32508	**40**

TABLE XXX
RESULTS OBTAINED IN THE MEDIUM VOLTAGE ELECTRICAL APPLICATION CONSIDERING $\alpha = 1.9$

Method	MSE_{tra}	MSE_{tst}	$\#R$
HSLR(WM,3,5)	29336	**29604**	58
HSLR-HR(WM,3,5)	**29119**	31949	66
G-HSLR(WM,3,5)	34197	35827	**51**
(I)HSLR(WM,3,5)	32795	33842	49
(I)HSLR-HR(WM,3,5)	**29923**	**31681**	49
(I)G-HSLR(WM,3,5)	32976	32508	**40**

training sets. On the one hand, we can observe that the global approach overfits in the complex low voltage application. On the other hand, the same global approach can not improve the accuracy of the results once it achieves a specific level. In contrast, HSLR have achieved significantly much accurate error levels.

It can be seen that the accuracy obtained by HSLR-LM does not only depend on the granularity level of the linguistic partitions and on the corresponding rules but there are many other interesting features that can be also exploited. As an example, consider the use of hierarchically reinforced rules, which shows a great improvement over the aproximation of both training and test sets.

- *From the complexity point of view*, the models generated by the HSLR-LM for the low voltage problem approximate properly the real system modeled and, what is more, they have the advantage of being simpler than the global ones. Moreover, the HSLR-HR models show great improvements with least complexity. The simplest model composed of only 10 rules is obtained using this capability.

  The models obtained by G-HSLR for the second electrical problem are simpler than the local ones, but the latter are almost a 35% more accurate. On the one hand, HSLR-LM can deal with these kinds of problems by making use of its capability of performing a tradeoff between accuracy and description, i.e., setting the factor of expansion in a more proper way. As an example, see the results shown in Table XXX considering $\alpha = 1.9$, where the global models are also seven rules simpler, when considering Ishibuchi's fitness function, and nine, when using ours.

In view of the former results, we can conclude that the HSLR-LM it is not a closed and static methodology. As said, it is open and the detected features suggest us that it could still be improved. Moreover, HSLR-LM is not based on simply grouping together different granularity level linguistic rules but it composes a methodology supported by many interesting features which, in different ways, allow us to generate more accurate models with an appropriate description level.

VII. Concluding Remarks

In this paper, an HSLR-LM has been proposed, which is a new approach to design linguistic models accurate to a high degree and suitably interpretable by human beings. An HKB learning process capable of automatically generating linguistic models following the said approach has been introduced as well, and its behavior has been compared to other modeling techniques in solving two different problems. The proposed process has obtained very good results.

On the one hand, a new approach to understand linguistic partitions has been shown, the HDB. This concept does not change the meaning of the linguistic variables neither their descriptive power, it just allows us to represent the information in a more accurate way with more granularity. As was said, the HKB structure, allowing each rule to be expanded and replaced by its hierarchical image, has demostrated to improve the model accuracy in some specific space zones presenting a higher complexity. We have shown that although more accurate systems can be obtained from a bigger number of rules, a small proper increase can still produce accurate results. Moreover, HSLR-LM provides a way to perform a tradeoff regulation between the accuracy and interpretability of the systems modeled.

As well as that, HSLR-LM can still be improved by: finding the best and more proper weight of each rule and/or of its multiple consequents, introducing more layers and an iterative process to deal with them, studying the effects of considering hierarchical reinforced rules as a partially revocable extension of the methodology search algorithm and evaluating different criteria to expand rules. On the other hand, we can also consider it as an iterative design method, from the user point of view. It is possible to develop an automatic method which iteratively search through different levels of the HKB (i.e., more than two levels). All of these things will be treated as extensions of the methodology in a future work.

Finally, as was said by Goldberg [11], if the future of computational inteligence "lies in the careful integration of the best constituent technologies," hierarchical and hybrid fuzzy systems and GAs require more than simple combinations derived from putting everything together, but a more sofisticated analysis and design of the system components and their features. This paper present progresses in a program of research devoted to find the most proper integration forms and to explore the HSLRs capabilities. As said, we have shown an open methodology and the obtained results encourage us to continue working in future extensions and validations for the HSLR-LM.

Appendix I
WM Rule Generation Method

The inductive RB generation process proposed by Wang and Mendel in [27] is widely known because of its simplicity and good performance. It is based on working with an input-output training data set, E_{TDS}, representing the behavior of the problem being solved and with previous definition of the DB composed of the input and output primary linguistic partitions used. The linguistic rule structure considered is the usual Mamdani-type rule with m input variables and one output variable presented in Section III.

The generation of the linguistic rules of this kind is performed by putting into effect the following three steps.

1) *To generate a preliminary linguistic rule set*: This set will be composed of the linguistic rule best covering each example (input-output data pair) existing in the input–output data set E_{TDS}. The structure of these rules is obtained by taking a specific example, i.e., an $m+1$-dimensional real array (m input and 1 output values) and setting each one of the rule variables to the linguistic label associated to the fuzzy set best covering every array component.
2) *To give a degree of importance to each rule*: Let R = IF x_1 is S_1 and ... and x_m is S_m THEN y is B be the linguistic rule generated from the example $e_l = (x_1^l, \ldots, x_m^l, y^l), l = 1, \ldots, |E_{\mathrm{TDS}}|$. The degree of

importance associated to it will be obtained as follows: $G(R) = \mu_{S_1}(x_1^l) \ldots \mu_{S_m}(x_m^l) \cdot \mu_B(y^l)$.

3) *To obtain a final RB from the preliminary linguistic rule set*: If all rules presenting the same antecedent values have associated the same consequent one in the preliminary set, this linguistic rule is automatically put (only once) into the final RB. On the other hand, if there are conflictive rules, i.e., rules with the same antecedent and different consequent values, the rule considered for the final RB will be the one with higher importance degree.

APPENDIX II
THR RULE GENERATION METHOD

This method is based on encoding all the cells of the complete decision table in the chromosomes. In this way, Thrift [26] establishes a mapping between the label set associated to the system output variable and an ordered integer set (containing one more element and taking 0 as its first element) representing the allele set. An example is shown to clarify the concept. Let $\{\mathrm{NB}, \mathrm{NS}, \mathrm{ZR}, \mathrm{PS}, \mathrm{PB}\}$ be the term set associated to the output variable, and let us note the absence of value for the output variable by the symbol "–." The complete set formed joining this symbol to the term set is mapped into the set $\{0, 1, 2, 3, 4, 5\}$. Hence the label NB is associated with the value 0, NS with $1, \ldots, \mathrm{PB}$ with 4 and the blank symbol "–" with 5.

Therefore, the GA employs an integer coding. Each one of the chromosomes is constituted by joining the partial coding associated to each one of the linguistic labels contained in the decision table cells. A gene presenting the allele "–" will represent the absence of the fuzzy rule contained in the corresponding cell in the RB.

The GA proposed considers an elitist selection scheme and the genetic operators used are of different nature. While the crossover operator is the standard two-point crossover, the mutation operator is specifically designed for the process. When it is applied over an allele different from the blank symbol, it changes its value one level either up or down or to the blank code. When the previous gene value is the blank symbol, it selects a new value at random.

Finally, the fitness function is based on an application specific measure. The fitness of an individual is determined by computing the use of the FRBS considering the RB coded in its genotype.

ACKNOWLEDGMENT

We would like to thank L. Sánchez, from Oviedo University, for the Electrical Engineering application from Hidroeléctrica del Cantábrico and for solving it by means of classical and neural techniques. We would also like to thank to the anonymous referees of this paper for their valuable contributions.

REFERENCES

[1] A. Bardossy and L. Duckstein, *Fuzzy Rule-Based Modeling with Application to Geophysical, Biological and Engineering Systems*. Boca Raton, FL: CRC, 1995.

[2] A. Bastian, "How to handle the flexibility of linguistic variables with applications," *Int. J. Uncertainty Fuzziness Knowledge-Based Syst.*, vol. 2, no. 4, pp. 463–484, 1994.

[3] E. Bauer and R. Kohavi, "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants," *Machine Learning*, vol. 5, pp. 1–38, 1998.

[4] B. Carse, T. C. Fogarty, and A. Munro, "Evolving fuzzy rule based controllers using genetic algorithms," *Fuzzy Sets Syst.*, vol. 80, pp. 273–294, 1996.

[5] O. Cordón, F. Herrera, and A. Peregrín, "Applicability of the fuzzy operators in the design of fuzzy logic controllers," *Fuzzy Sets Syst.*, vol. 86, pp. 15–41, 1997.

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

[7] O. Cordón, F. Herrera, and L. Sánchez, "Solving electrical distribution problems using hybrid evolutionary data analyis techniques," *Appl. Intel.*, vol. 10, pp. 5–24, 1999.

[8] O. Cordón, F. Herrera, and P. Villar, "Analysis and guidelines to obtain a good uniform fuzzy partition granularity for fuzzy rule-based systems using simulated annealing," *Int. J. Approx. Reasoning*, vol. 25, no. 3, pp. 187–215, 2000.

[9] O. Cordón and F. Herrera, "A proposal for improving the accuracy of linguistic modeling," *IEEE Trans. Fuzzy Syst.*, vol. 8, pp. 335–344, Aug. 2000.

[10] A. E. Gegov and P. M. Frank, "Hierarchical fuzzy control of multivariable systems," *Fuzzy Sets Syst.*, vol. 72, pp. 299–310, 1995.

[11] D. E. Goldberg, "A Meditation on the Computational Intelligence and Its Future," Dept. of General Engineering, Univ. of Illinois at Urbana-Champaign, Tech. Rep. #2 000 019, 2000.

[12] 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.

[13] K. Hirota, Ed., *Industrial Applications of Fuzzy Technology*. New York: Springer-Verlag, 1993.

[14] H. Ishibuchi, K. Nozaki, and H. Tanaka, "Distributed representation of fuzzy rules and its applications to pattern classification," *Fuzzy Sets Syst.*, vol. 52, pp. 21–32, 1992.

[15] ——, "Efficient fuzzy partition of pattern space for classification problems," *Fuzzy Sets Syst.*, vol. 59, pp. 295–304, 1993.

[16] 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, pp. 260–270, June 1995.

[17] F. Klawon, "Fuzzy sets and vague environments," *Fuzzy Sets Syst.*, vol. 66, pp. 207–221, 1994.

[18] F. Klawon and J. L. Castro, "Similarity in fuzzy reasoning," *Math. Soft Comput.*, vol. 2, pp. 197–228, 1995.

[19] C. T. Leondes, Ed., *Fuzzy Theory Systems, Techniques and Applications*. New York: Academic, 2000.

[20] T. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.

[21] W. Pedrycz, Ed., *fuzzy modeling: Paradigms and Practice*. Boston, MA: Kluwer, 1996.

[22] C. V. S. Raju and J. Zhou, "Adaptative hierarchical fuzzy controller," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, pp. 973–980, Aug. 1993.

[23] L. Sánchez, "Study of the Asturias Rural and Urban Low Voltage Network," Hidroeléctrica del Cantábrico Research and Development Department, Asturias, Spain, Tech. Rep. (in spanish), 1997.

[24] ——, "Interval-valued GA-P algorithms," *IEEE Trans. Evol. Comput.*, vol. 4, pp. 64–72, 2000.

[25] M. Sugeno and T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling," *IEEE Trans. Fuzzy Syst.*, vol. 1, pp. 17–31, Feb. 1993.

[26] P. Thrift, "Fuzzy logic synthesis with genetic algorithms," in *Proc. Fourth Int. Conf. Genetic Algorithms (ICGA'91)*, 1991, pp. 509–513.

[27] L. X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, pp. 1414–1427, 1992.

[28] R. R. Yager, "On a hierarchical structure for fuzzy modeling and control," *IEEE Trans. Systems, Man, Cybern.*, vol. 23, pp. 1189–1197, Aug. 1993.

[29] ——, "On the construction of hierarchical fuzzy systems model," *IEEE Trans. Syst., Man, Cybern.*, vol. 28, pp. 55–66, Feb. 1998.

[30] 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, pp. 530–537, Aug. 1998.

[31] L. A. Zadeh, "Fuzzy sets," *Inform. Control*, vol. 8, pp. 338–353, 1965.

[32] ——, "The concept of a linguistic variable and its application to aproximate reasoning," *Inform. Sci.*, pt. I, vol. 8, pp. 199–249, 1975.

Oscar Cordón was born in Cadiz, Spain, in 1972. He received the M.S. and Ph.D. degrees in computer science, both from the University of 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 is a member of the Approximate Reasoning and Artificial Intelligence research group. He has written more than 30 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). His current main research interests are in the fields of: fuzzy rule-based systems, fuzzy and linguistic modeling, fuzzy classification, genetic fuzzy systems, evolutionary algorithms, ant colony optimization, and information retrieval.

Dr. Cordón has coorganized special sessions on "Genetic Fuzzy Systems" in the Eight IPMU'2000 Conference and in the Joint IFSA-NAFIPS'2001 Conference, and has coedited two special issues on the same topic in the *Information Sciences and Fuzzy Sets and Systems Journals*.

Francisco Herrera was born in 1966. He received the M.S. and Ph.D. degrees in mathematics, both from the University of Granada, Spain, in 1988 and 1991, respectively.

He is an Associate Professor in the Department of Computer Science and Artificial Intelligence, University of Granada. He coedited the book *Genetic Algorithms and Soft Computing* (New York: Springer-Verlag, 1996) and four journal special issues, "Genetic Fuzzy Systems for Control and Robotics" (*International Journal of Approximate Reasoning*, 17:4, 1997), "Genetic Fuzzy Systems" (*International Journal of Intelligent Systems*, 13:10–11, 1998), "New Developments in Genetic Fuzzy Systems" (*Information Sciences*, 136:1-4, 2001), and "Computer with Words: Foundations and Applications" (*International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 2001). He is coauthor of the book *Genetic Fuzzy Systems. Evolutionary Tuning and Learning of Fuzzy Knowledge Bases* (Singapore: World Scientific, 2001). His research interests include decision making problems in fuzzy environment, fuzzy rule-based systems, machine learning, genetic algorithms, and genetic fuzzy systems.

Igor Zwir was born in 1967. He received the B.S. degree in computer science and the M.S. degree, both from the University of Buenos Aires, Argentina, and the Ph.D. degree in computer science from the University of Granada, Spain, in 1992, 1997, and 2001, respectively.

He is an Assistant Professor of Computer Science at the University of Buenos Aires. His current research interests are fuzzy rule-based systems, evolutionary computation, pattern recognition, and knowledge discovery in complex objects, all of these topics in the field of computational biology and economics.