Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @DIRECT®

Fuzzy Sets and Systems 138 (2003) 307–341

FUZZY sets and systems

www.elsevier.com/locate/fss

A hierarchical knowledge-based environment for linguistic modeling: models and iterative methodology☆

O. Cordón[a], F. Herrera[a,*], I. Zwir[b]

[a] *Department of Computer Science and Artificial Intelligence, E.T.S. de Ingeniería Informática, University of Granada, 18071 Granada, Spain*
[b] *Department of Computer Science, F.C.E. y N., University of Buenos Aires, 1428 Buenos Aires, Argentina*

Received 10 March 2000; received in revised form 2 July 2002; accepted 11 July 2002

Abstract

Although linguistic models are highly descriptive, they suffer from inaccuracy in some complex problems. This fact is due to problems related to the inflexibility of the linguistic rule structure that has been considered. Moreover, methods often employed to design these models from data are also biased by the former structure and by their nature, which is close to prototype identification algorithms.

In order to deal with these problems of linguistic modeling, an extension of the knowledge base of linguistic fuzzy rule-based systems was previously introduced, i.e., the hierarchical knowledge base (HKB) (IEEE Trans. Fuzzy Systems 10 (1) (2002) 2). Hierarchical linguistic fuzzy models, derived from this structure, are viewed as a class of local modeling approaches. They attempt to solve a complex modeling problem by decomposing it into a number of simpler linguistically interpretable subproblems. From this perspective, linguistic modeling using an HKB can be regarded as a search for a decomposition of a non-linear system that gives a desired balance between the interpretability and the accuracy of the model. Using this approach, we are able to effectively explore the fact that the complexity of the systems is usually not uniform.

We propose a well-defined hierarchical environment adopting a more general treatment than the typical prototype-oriented learning methods. This iterative hierarchical methodology takes the HKB as a base and performs a wide variety of linguistic modeling. More specifically, from fully interpretable to fully accurate, as well as intermediate trade-offs, hierarchical linguistic models.

With the aim of analyzing the behavior of the proposed methodology, two real-world electrical engineering distribution problems from Spain have been selected. Successful results were obtained in comparison with other system modeling techniques.

Keywords: Linguistic modeling; Fuzzy rule-based systems; Hierarchical linguistic partitions; Hierarchical knowledge base; Rule selection; Genetic algorithms

---

☆ This research has been supported by CICYT PB98-1319.
* Corresponding author. Tel.: +34-58-244019; fax: +34-58-243317.
E-mail addresses: ocordon@decsai.ugr.es (O. Cordón), herrera@decsai.ugr.es (F. Herrera), zwir@dc.uba.ar (I. Zwir).

PII: S0165-0114(02)00388-3

1. Introduction

One of the most important applications of fuzzy rule-based systems (FRBSs) is *system modeling* [2,24,36]. It is possible to distinguish between two types of modeling when working with FRBSs: *linguistic modeling* [31]—often represented by Mamdani FRBSs—and *fuzzy modeling* [2,31]—frequently represented by Takagi–Sugeno–Kang (TSK) FRBSs—according to the fact that the main requirement is the interpretability or the accuracy of the models, respectively. In fact, we usually find that these are two contradictory needs.

In this paper we focus on improving *linguistic modeling* [31]. Particularly, we make use of Mamdani-type FRBSs, which become the typical example of linguistic models that present the maximum description level, i.e., fuzzy rules that are globally interpretable. For the sake of simplicity, we will refer to the components of this kind of FRBSs—fuzzy linguistic rules and partitions—just as linguistic. A more detailed description of linguistic models and their differences with other fuzzy models can be found in [7,19,38].

Linguistic models, although descriptive, also suffer from inaccuracy in some complex problems. This fact is due to some problems related to the linguistic rule structure considered, which is a consequence of the inflexibility of the concept of linguistic variable [37]. Moreover, methods that usually learn these models from data are also biased by the former structure and by their nature, which is close to prototype identification algorithms [6,28,39].

In order to deal with these problems of linguistic modeling, we propose a hierarchical environment—model representation and learning methodology—as a strategy to improve simple linguistic models. This approach preserves the original model descriptive power and increases its accuracy by reinforcing those problem subspaces that are specially difficult.

We consider an extension of the knowledge base structure of linguistic or Mamdani FRBSs by which the concept of "*layers*" was introduced [11]. In this extension, which is also a generalization, the knowledge base is composed of a set of layers. Each layer contains linguistic partitions with different granularity levels and fuzzy rules, whose linguistic variables take values in these partitions. This knowledge base was called the hierarchical knowledge base (HKB), and it is formed by:

- A hierarchical data base (HDB), containing linguistic partitions of the said type.
- A hierarchical rule base (HRB), with the corresponding linguistic rules.

A previous approach to develop hierarchical models from a limited HKB [11], hierarchical systems of linguistic rules (HSLRs) of two levels, was focused on interpretability. In this paper, we extend the former model structure, i.e., the HKB, and propose an HSLR learning methodology (HSLR-LM) to learn it from examples. This methodology iteratively selects bad performance linguistic rules, which need more specificity, and expands them locally through different granularity levels. This fact produces a wide spectrum of solutions—from high interpretable to high accurate, and trade-off solutions—and avoids typical drawbacks of prototype-based linguistic rule generation methods (LRG-methods).

As a meta-methodology, the HSLR-LM works on simple models that have been previously obtained from different LRG-methods. Thus, for the sake of compatibility, its interpolation method activates independently each rule as a typical inference in fuzzy logic. Fuzzy set theory offers excellent tools for representing the uncertainty associated with the decomposition task, providing smooth transitions between individual local submodels. It also facilitates the interpolation of various types of knowledge within a common framework, giving a desired balance between the complexity and the accuracy of the model. Using this approach, we are able to effectively explore the fact that the complexity of the systems is usually not uniform.

In this contribution, accuracy and interpretability cannot be considered independently but as a trade-off interaction. Moreover, we empirically prove that *it is not always true that a set of rules with a higher granularity level performs a more accurate modeling of a problem than another with a lower one*. Interpretability condition emphasizes the generation of a low number of rules, thus, reducing the complexity of the model. However, this reduction also prevents possible model overfitting [21], i.e., like a pre-pruning strategy [23] which also improves the generalization and the accuracy of the results.

The relationship between accuracy and interpretability does not only depend on granularity and specificity, but also on other factors, for example, rule weights, flexible rule consequents, and moreover, compasity of and cooperation policies between the rules [7,11]. Therefore, different policies are considered for the methodology to find out the best way to perform the local hierarchical fuzzy decomposition and, afterwards, the corresponding integration in a compact HKB:

- Generation policies, considering weighted and double-consequent reinforced linguistic rules.
- Expansion policies, viewing the hierarchical process as a replacement or a reinforcement of bad performance linguistic rules.
- Selection policies, allowing different criteria—accuracy or trade-off accuracy-complexity oriented—to summarize the most compact set of linguistic rules by genetic algorithms.

The setup of this paper is as follows. In Section 2, the HKB philosophy is introduced and the lacks of LRG-methods are also highlighted. In Section 3, the local-oriented and iterative HSLR-LM is described in detail. Different policies concerning the algorithm performance are also studied. In Section 4, HSLR models obtained from the HSLR-LM are applied to solve previous problems on real-world applications. Analysis of results is performed by three different points of view: from the methodology performance, from the influence of the methodology parameters, and from the methodology policies. Results are also compared with other system modeling techniques. Finally, in Section 5, some concluding remarks are pointed out. Appendix A contains a brief description of the LRG-methods and acronyms used in the paper.

2. Framework

Our approach is oriented to produce a more general and well-defined structure, the HKB. This structure should be flexible enough to allow a wide variety of linguistic models, as said from very accurate to well interpretable ones. Our purpose is to preserve the descriptive capabilities of previous models, increasing their accuracy at different hierarchical levels. We simplify the inference mechanism adopted by previous hierarchical approaches [12,16,25,34], activating independently each rule as it is done in the conventional inference mechanism. Besides, we use summarization processes to obtain a compact set of rules that have good cooperation between them.

There are many reasons that encourage the use of hierarchical representations. From the theoretical point of view, the theory of fuzzy sets offers an excellent tool for representing the uncertainty associated with the hierarchical decomposition task and for providing smooth transitions between the individual local submodels [1]. Moreover, hierarchical rules are supported by the lack of truth functionality in many "logics of uncertainty". Results of the research on human plausible reasoning conducted by Michalski [20] show that people derive a combined certainty of a conclusion from uncertain premises by taking into consideration structural (or semantic) relations among the premises, based on a hierarchical knowledge representation.

From the practical point of view, it has been observed that the knowledge base structure, usually employed in the field of linguistic modeling, suffers from inaccuracy when working with very complex systems [3]. One way to solve many of the previous problems is to make the knowledge base more flexible, i.e., build a HKB. The basic philosophy of this structure will be described in Section 2.1.

Otherwise, there exists another problem related with linguistic modeling that concern by those learning methods usually employed to identify the knowledge base of an FRBS. Some of their lacks as prototype-identification algorithms (see Section 2.2) motivated the development of HSLR-LM.

2.1. HKB philosophy

The inaccuracy of linguistic models is due to some problems related to the linguistic rule structure considered in their knowledge base. This problem arises as a consequence of the inflexibility of the concept of linguistic variable [37], mostly caused by the rigid partitioning of the input and output spaces. A summary of these problems can be found in [3].

Therefore, we present a more flexible knowledge base structure that allows us to improve the accuracy of linguistic models without losing their interpretability to a high degree, the HKB [11]. It is composed of a set of layers, and each layer is defined by its components in the following way:

$$layer(t,n(t)) = DB(t,n(t)) + RB(t,n(t)),$$

where

- $n(t)$ is the number of linguistic terms that compose the partitions of layer t, and
- $DB(t,n(t))$ is the database which contains the linguistic partitions with granularity level $n(t)$ of layer t.

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

$$DB(t,n(t)) \rightarrow DB(t+1,2n(t)-1),$$

which means that a linguistic partition in $DB(t,n(t))$ with $n(t)$ linguistic terms becomes a linguistic partition in $DB(t+1,2n(t)-1)$ [11] (see Fig. 1 and Table 1). In order to satisfy this requirement, each linguistic term $S_k^{n(t)}$—term of order k from the linguistic partition in $DB(t,n(t))$—is mapped into $S_{2k-1}^{2n(t)-1}$. The former modal points are preserved and a set of $n(t)-1$ new terms is created, each one between $S_k^{n(t)}$ and $S_{k+1}^{n(t)}$ ($k=1,\dots,n(t)-1$) (see Table 2). In this view, we can generalize this

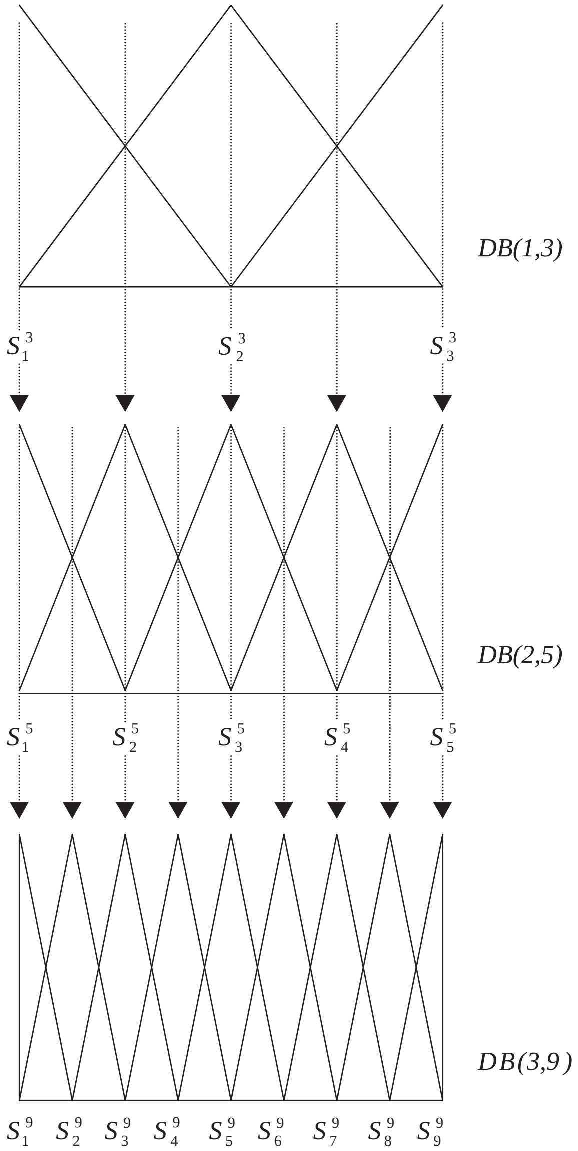

Fig. 1. Three layers of linguistic partitions which compose the HDB.

two-level successive layer definition for all layers t in the following way:

$$n(t) = (N-1)2^{t-1} + 1,$$

where $n(1) = N$, i.e., the number of linguistic terms in the initial layer partitions.

Remark 1. 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

Table 1
Hierarchy of *DB*s starting from two or four initial terms

$DB(t,n(t))$	$DB(t,n(t))$
$DB(1,2)$	$DB(1,4)$
$DB(2,3)$	$DB(2,7)$
$DB(3,5)$	$DB(3,13)$
$DB(4,9)$	$DB(4,25)$
$\vdots$	$\vdots$
$DB(6,33)$	$DB(6,97)$
$\vdots$	$\vdots$

Table 2
Mapping between terms from successive *DB*s

$DB(t,n(t))$		$DB(t+1,2n(t)-1)$
$S_{k-1}^{n(t)}$	$\rightarrow$	$S_{2k-3}^{2n(t)-1}$
		$S_{2k-2}^{2n(t)-1}$
$S_{k}^{n(t)}$	$\rightarrow$	$S_{2k-1}^{2n(t)-1}$
		$S_{2k}^{2n(t)-1}$
$S_{k+1}^{n(t)}$	$\rightarrow$	$S_{2k+1}^{2n(t)-1}$

membership functions at each level of the hierarchy. However, linguistic partitions for variables with global semantics can also be defined by expert knowledge.

- $RB(t,n(t))$ is the rule base formed by those linguistic rules whose linguistic variables take values in the former partitions. The main purpose of developing an HRB is to model the problem space in a more accurate way. To do so, those linguistic rules from $RB(t,n(t))$ that model a subspace with bad performance are expanded into a set of more specific linguistic rules, which become their image in $RB(t+1,2n(t)-1)$. This set of rules models the same subspace as the former ones and replaces them.

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

Remark 2. The t-linguistic rule structure is formed by a collection of well-known Mamdani-type linguistic rules:

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

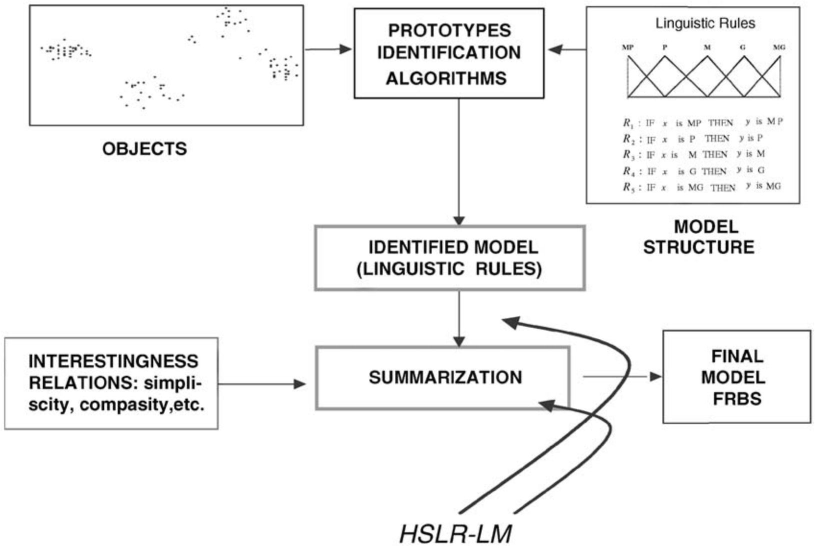

Fig. 2. LRG-methods as prototype identification algorithms.

where $x_1,\ldots,x_m$ and y are input and output linguistic variables, respectively; and $S_{i1}^{n(t)},\ldots,S_{im}^{n(t)}$, $B_i^{n(t)}$ are linguistic terms from different t-linguistic partitions contained in $DB(t,n(t))$, with fuzzy sets associated defining their meaning. In this contribution, we use the minimum t-norm in the role of conjunctive and implication operator. The fuzzy interpolation is performed by the defuzzification-strategy center of gravity weighted by the matching degree [8]. Each rule is independently activated as it is done in the conventional inference mechanism. Any other defuzzification method considering the matching degree of the fired rules may be used.

The described 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 $t+1$, 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 can now define the HKB as the union of all layers t:

$$HKB = \bigcup_t layer(t,n(t)).$$

2.2. Hierarchical methodology for learning an HKB

In order to characterize LRG-methods, and regarding [27,39], we can say that basically an LRG-method does its job as a prototype-identification algorithm. These algorithms perform the optimization of a functional $Q(F;Model(\gamma))$ that measures the extent to which the parameterized model $Model(\gamma)$ fits the subset F of the described object (see Fig. 2).

From this perspective, the linguistic rule identification problem is formulated as a clustering problem. More specifically, extracted subsets meet, to some extent, the requirements imposed by the model collection in the same way that elements of a clustering partition satisfy the constraint that their members be as similar as possible [27,39]. This point of view follows original ideas of Ruspini [26], later expanded by Bezdek introducing various methods centered upon the notion of prototype [4]. The basic idea of summarizing a data set by a number of representative prototypes—objects lying in the same space as the sample points—was later extended in many significant directions by relaxing this concept in a variety of ways, e.g., line segments, ellipsoids, etc. [5,18]. In this paper, we particularize these extensions by considering such prototypes as being linguistic rules [1].

Having the above concepts in mind, LRG-methods can be seen as identification algorithms with linguistic rule prototypes. That is, linguistic model builders whose main purpose is to extract the most suitable set of linguistic rules from an object (input–output data). This process is performed according to an optimization measure which evaluates the quality of the approximation. In addition, they organize and summarize results by interestingness criteria to provide a more compact and useful representation of the salient structures.

In order to illustrate this situation, consider for example the Wang and Mendel's Algorithm [33] described in Appendix A. It identifies linguistic rules from a set of input–output data (*object* F), building an approximate linguistic model ($Model(\gamma)$). The quality of the candidate substructures (rule premises) is measured based on a covering criterion ($Q(F; Model(\gamma))$).

All of these models generated by LRG-methods have the same drawbacks that prototype-identification methods have:

- Simple formulation of the prototype-identification problem as an optimization of a functional would simply result in a large collection of very specific rules. They used to have small extent and high accuracy, but poor generalization. Smaller, rather than larger, significative prototypes with high generalization power should be preferred.
- The determination of a complete clustering or a partition of the data set into a fixed number of prototypes has been a major issue for a long time.

All of these problems from LRG-methods and the use of a more complex structure like the HKB motivate a different treatment of the linguistic rule learning process. To do so, we will consider a hierarchical meta-methodology which modifies the framework shown in Fig. 2 by considering the following requirements:

- *Implementation of a sort of trade-off between the extensionality and the accuracy of the models*. Consider that rules which provide good explanations tend to be limited in extent. Conversely, those that are capable of describing large subsets of the data set are poorly accurate.
- *Adopt a more general treatment than that of a typical clustering problem*. Emphasize on the sequential isolation of individual clusters [18,27] rather than the determination of a full clustering. Furthermore, we do not want to assume a priori knowledge of the total number of clusters—rule prototypes—requiring that the set of all clusters be an exhaustive partition of the complete object.

Considering the former requirements, in the following section we will introduce an extended local-oriented HSLR-LM. This learning methodology will modify initial models identified by LRG-methods in an iterative way, performing gradual refinements on them.

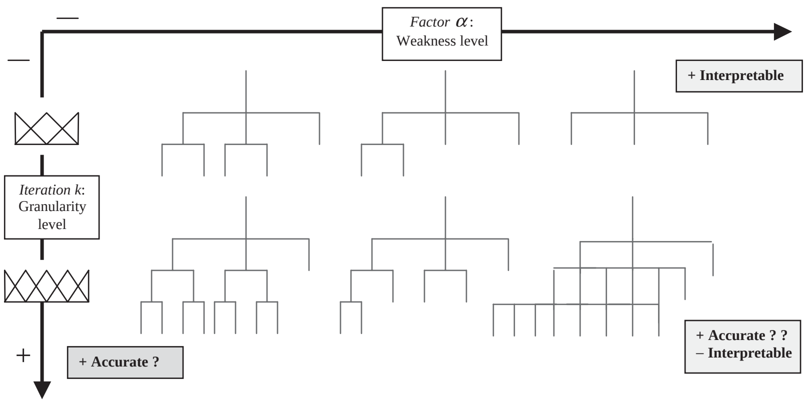

Fig. 3. Trade-offs between interpretability and accuracy.

3. An iterative localized HSLR-LM

We present the HSLR-LM as a methodology which performs a local hierarchical treatment of those problem subspaces which are badly modeled by conventional LRG-methods. This learning methodology performs a trade-off between the model extensionality and accuracy by implementing a local and iterative strategy. From this approach, a priori specifications of fixed number of linguistic rules can be avoided. Hopefully, this methodology will allow us to obtain a variety of linguistic models, from highly accurate to highly interpretable ones.

An HSLR-LM was developed as a parametrized methodology. The factor of expansion controls the level of bad performance that a rule should have to be expanded into more specific ones. Thus, a low factor implies a small expansion—smaller number of rules—and a more interpretable model. In this sense, our previous approach [11] is a special case which makes use of this parameter to obtain interpretable hierarchical models. Another parameter to be considered is the iteration of the algorithm. It is used to control the granularity level that more specific hierarchical rules, which replace those ones with bad performance, should have (see Fig. 3).

In the following we will first present the HSLR-LM algorithm. Afterwards, we will propose different design policies which could be combined with the basic local iterative strategy: HRB generation policies, HRB expansion policies, and HRB selection policies. All of these components compose a flexible hierarchical framework to deal with complex problems based on different requirements of accuracy and/or interpretability.

3.1. Algorithm

In this Subsection we present our iterative methodology to generate an HKB. We use an 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 $E_{TDS}=\{e^1,\ldots,e^l,\ldots,e^q\}$ is composed of q input–output data pairs $e^l=(ex_1^l,\ldots,ex_m^l,\ ey^l)$ which represent the behavior of the system modeled.

It basically consists of the following steps which can be also graphically seen in Fig. 6.

Initialization process

Step 0: *$RB(1,n(1))$ generation process. Generate $RB(1,n(1))$*, where the rules of the initial layer are generated from the terms defined in the present partitions—located in $DB(1,n(1))$—by an LRG-method:

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

where $n(1)=N$ and the initial $DB(1,n(1))$ is given by an expert or by a normalization process considering a small number of terms. The iteration and the last layer generated counters are initialized: $k=1$ and $p=1$, respectively.

Iteration process (*iteration k*)

Step 1: *HRB generation process. Generate HRB^{k+1}, where the linguistic rules from layer $(t+1)$ are generated taking into account $RB(t,n(t))$, $DB(t,n(t))$* and $DB(t+1, 2n(t+1)-1)$, and $1 \leqslant t \leqslant p \leqslant k+1$ (see Fig. 4).

(a) *Bad performance t-linguistic rule selection process*: This process performs the selection of those *t-linguistic rules* from HRB^k which will be expanded into $RB(t+1, 2n(t)-1)$, based on an error measure.

(i) *Calculate the error of HRB^k as a whole as MSE* (E_{TDS}, HRB^k): The mean square error (*MSE*) calculated over the training data set E_{TDS} is the error measure used in this work. Therefore, the *MSE* of the entire set of *t-linguistic rules* is represented by the following expression:

$$MSE(E_{\text{TDS}}, HRB^k) = \frac{\sum_{e^l \in E_{\text{TDS}}} (ey^l - s(ex^l))^2}{2|E_{\text{TDS}}|},$$

where $s(ex^l)$ is the output value obtained from the HRB^k, 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 t-linguistic rule as $MSE(E_i, R_i^{n(t)})$*: We need to define a subset E_i of E_{TDS} to calculate the error of the rule $R_i^{n(t)}$. E_i is a subset of the examples matching the antecedents of the rule i to a specific degree τ:

$$E_i = \{e^l \in E_{\text{TDS}} / \text{Min}(\mu_{S^{n(t)}_{i1}}(ex^l_1), \ldots, \mu_{S^{n(t)}_{im}}(ex^l_m)) \geqslant \tau\},$$

where $\tau \in (0,1]$. Then, we calculate the *MSE* of a *t-linguistic rule* $R_i^{n(t)}$ as

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

where $s_i(ex^l)$ is the output value obtained when inferring with $R_i^{n(t)}$. We should note that any other local error measure can be considered with no change in our methodology, such as the one shown in [35].

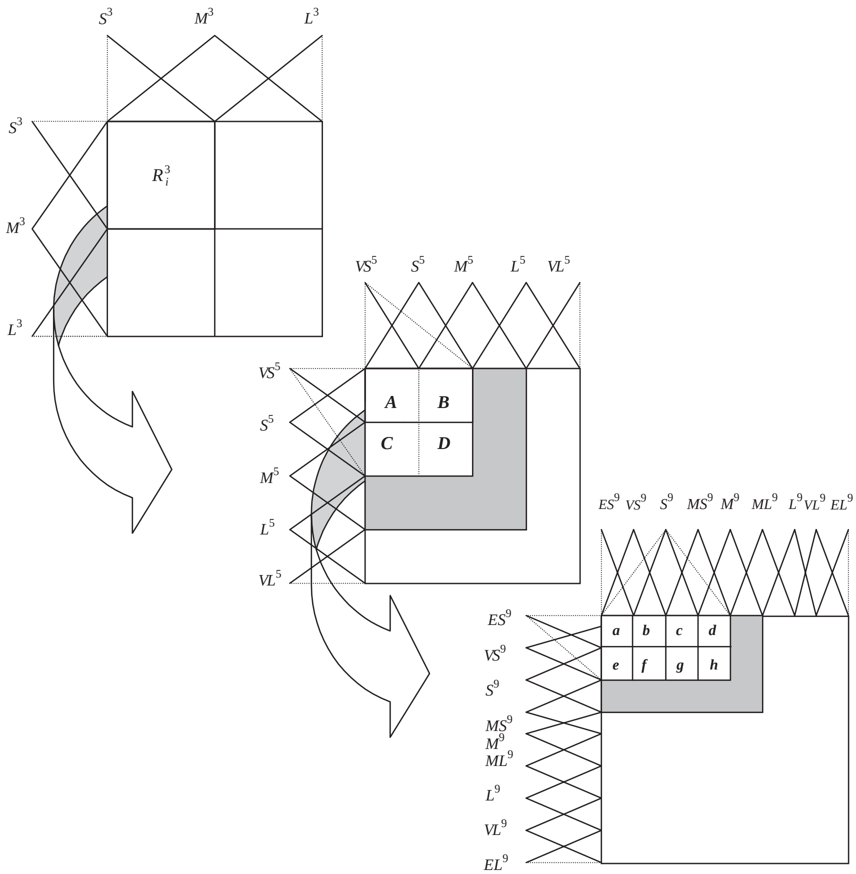

Fig. 4. Example of the HRB generation process. If $\delta = 0.5$, the problem subspace resulting from the bad *t-linguistic rule* expansion is the one represented by the small white square. If $\delta = 0.1$, it would be composed of the union of the former small white square and the gray one.

(iii) *Select the t-linguistic rules with bad performance which are going to be expanded, making the difference from the good ones*:

$$HRB^k_{\text{bad}} = \{R_i^{n(t)} / MSE(E_i, R_i^{n(t)}) \geqslant \alpha\, MSE(E_{\text{TDS}}, HRB^k)\},$$

$$HRB^k_{\text{good}} = \{R_i^{n(t)} / MSE(E_i, R_i^{n(t)}) < \alpha\, MSE(E_{\text{TDS}}, HRB^k)\},$$

where the threshold α represents a percentage of the error of the whole rule base which determines the expansion of a rule. For example, $\alpha = 1.1$ means that a *t-linguistic rule* with an MSE 10% higher than the MSE of the entire HRB^k should be expanded. The expansion of factor α 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 system modeled.*

Before describing the next step and for the sake of clarity, we are going to refer to $DB(t,n(t))$ as $DB_{x_j}(t,n(t))$ $(j=1,\dots,m)$, meaning that it contains the *t-linguistic partition* where the input linguistic variable x_j takes values, and as $DB_y(t,n(t))$ for the output variable y. Notice that, even if all *t-linguistic partitions* contained in a $DB(t,n(t))$ have the same number of linguistic terms, they are defined over different domains. Each one, corresponding to one linguistic variable or normalized by scaling factors.

Now, for each $R_i^{n(t)} \in HRB_{\text{bad}}^k$ perform the following processes:

(b) $DB(t+1, 2n(t)-1)$ *selection process*: If $(t=p)$, then: $p \leftarrow p+1$; create $DB_{x_j}(p,n(p))$, for all input linguistic variables x_j $(j = 1,\dots,m)$, and $DB_y(p,n(p))$, for the output linguistic variable y; $HDB^p \leftarrow HDB^{p-1} \cup DB(p,n(p))$. More specifically, if the *t-linguistic partitions*—corresponding to a bad rule—have reached the maximum granularity level available in the HDB, then generate the next layer database.

(c) *Bad performance t-linguistic rule expansion process*:

(i) *Select those* $(t+1)$*-linguistic partition terms from* $DB(t+1,2n(t)-1)$ *that will be contained in the* $(t+1)$*-linguistic rules. These rules are considered the image of the previous layer bad rules.*

For all linguistic terms considered in $R_i^{n(t)}$–$S_{ij}^{n(t)}$ defined in $DB_{x_j}(t,n(t))$ and associated to the linguistic variables x_j–, select those terms $S_h^{2n(t)-1}$ in $DB_{x_j}(t+1,2n(t)-1)$ which significantly intersect them. Consequently, for $B_i^{n(t)}$defined in $DB_y(t,n(t))$ and associated to the linguistic variable y. In other words, select those terms from the $(t+1)$*-linguistic partition* that describe approximately the same subspace as the terms included in $R_i^{n(t)}$, 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. Thus, the set of terms taken from $(t+1)$*-linguistic partitions* for the expansion of a *t-linguistic rule* $R_i^{n(t)}$ are selected in the following way:

$$I(S_{ij}^{n(t)}) = \left\{ S_h^{2n(t)-1} \in DB_{x_j}(t+1,2n(t)-1) \Big/ \operatorname*{Max}_{u\in U_j} \text{Min}\{\mu_{S_{ij}^{n(t)}}(u), \mu_{S_h^{2n(t)-1}}(u)\} \geqslant \delta \right\},$$

$$I(B_i^{n(t)}) = \left\{ B_h^{2n(t)-1} \in DB_y(t+1,2n(t)-1) \Big/ \operatorname*{Max}_{v\in V} \text{Min}\{\mu_{B_i^{n(t)}}(v), \mu_{B_h^{2n(t)-1}}(v)\} \geqslant \delta \right\},$$

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

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

$$I(R_i^{n(t)}) = I(S_{i1}^{n(t)}) \times \cdots \times I(S_{im}^{n(t)}) \times I(B_i^{n(t)}),$$

where $I(R_i^{n(t)}) \subset DB(t+1, 2n(t)-1)$. More specifically, create a fuzzy grid in the input fuzzy subspace of a bad performance rule that is being expanded.

(iii) *Extract* $(t+1)$*-linguistic rules from the selected* $(t+1)$*-linguistic partition terms, producing a set of L* $(t+1)$*-linguistic rules. This set represents the expansion of the bad t-linguistic rule* $R_i^{n(t)}$.

This task is performed by an LRG-method, which takes $I(R_i^{n(t)})$ and the set of input–output data E_i as its parameters:

$$CLR(R_i^{n(t)}) = LRG\text{-}method(I(R_i^{n(t)}), E_i) = \{R_{i_1}^{2n(t)-1}, \ldots, R_{i_L}^{2n(t)-1}\},$$

where $CLR(R_i^{n(t)})$ is the image of the expanded linguistic rule $R_i^{n(t)}$, i.e., the candidates to be in the HRB^{k+1} from rule i.

Step 2: *Summarization process. Obtain a joined set of candidate linguistic rules* ($JCLR$). Join the new candidate $(t+1)$-*linguistic rules* and the former good performance *t-linguistic rules:*

$$JCLR = HRB^k_{\text{good}} \cup \left(\bigcup_i CLR(R_i^{n(t)}) \right),$$

where $R_i^{n(t)} \in HRB^k_{\text{bad}}$.

Step 3: *HRB selection process. Simplify the set JCLR by removing the unnecessary rules from it and generating an HRB^{k+1} with good cooperation.* In this paper we consider an genetic process [11,15,17] to put this task into effect, but any other technique could be considered:

$$HRB^{k+1} = Selection\ Process(JCLR)$$

In the $JCLR$—where there are coexisting rules of different hierarchical layers—it may happen that a complete set of $(t+1)$-*linguistic rules*—which replaces an expanded *t-linguistic rule*—does not produce good results. However, a subset of this set of $(t+1)$-*linguistic rules* may work properly with less rules that cooperate better between them, and with those good rules preserved from the previous layer. Thus, the $JCLR$ set of rules may present redundant or unnecessary rules making the model using this HKB less accurate.

The genetic rule selection process [11,15] is based on a binary-coded genetic algorithm. The selection of the individuals is performed using the stochastic universal sampling procedure together with an elitist selection scheme. 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^{k+1} such that

$$IF\ \ c_i = 1\ \ THEN\ \ (R_i \in HRB^{k+1})\ \ ELSE\ \ (R_i \notin HRB^{k+1}).$$

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. This measure depends on the cooperation level of the rules existing in the $JCLR$. We usually work with the MSE over a training data set, as it was previously defined, although other measures may be used. The importance of this process is illustrated in Fig. 5.

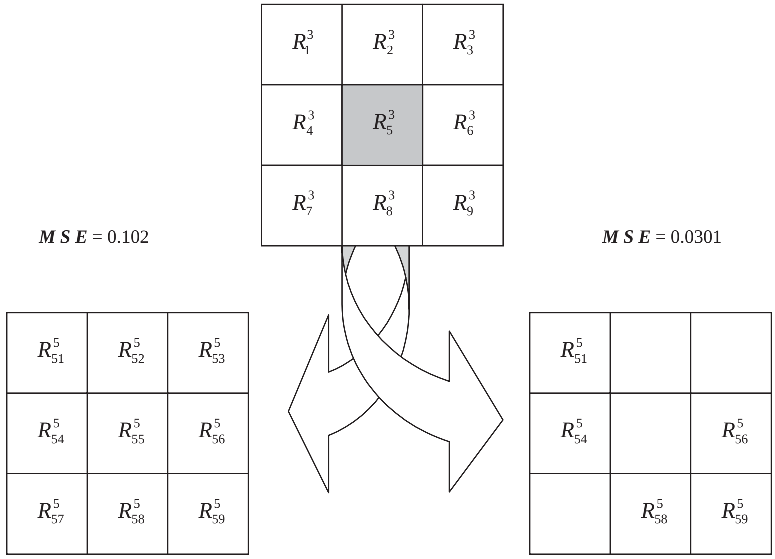

Fig. 5. HRB selection process.

Step 4: *Model validation process.* The final model is either accepted as a proper one for the given purpose, or it is rejected generating another iteration of the hierarchical process. Among the different indices that can be used to measure the quality of linear or non-linear systems after an identification loop [1], we consider a monotonic *MSE* measure on the training set computed as

$$\begin{aligned} &IF\ (MSE(HRB^{k+1}(E_{\mathrm{TDS}})) \leqslant MSE(HRB^{k}(E_{\mathrm{TDS}})) \text{ and } (k < K_{\max})) \\ &\quad THEN\ k \leftarrow k+1;\ \text{Goto Step 1,} \end{aligned}$$

where $K_{\max}$ is a previously defined maximum number of iterations. This value is based on a trade-off between the complexity and the accuracy of the desired model.

Finally, as a consequence of applying this algorithm, the HKB is redefined as

$$HKB = HDB^{p} + HRB^{k+1}.$$

As we referred, Fig. 6 graphically illustrates the HSLR-LM algorithm.

3.2. Generation policy

The $DB(t+1, 2n(t)-1)$ generation policy (see Step 1(c)(i)) was based on selecting those terms from $DB(t+1, 2n(t)-1)$ that significantly intersect the ones of the expanded bad rule. As a consequence of this policy, at least two different kinds of linguistic rules can be obtained from the HRB generation process. First, repeated $(t+1)$-*linguistic rules* can be generated as a consequence of the expansion of adjacent bad *t-linguistic rules*. Second, double-consequent $(t+1)$-*linguistic rules* can be derived in the same reason.

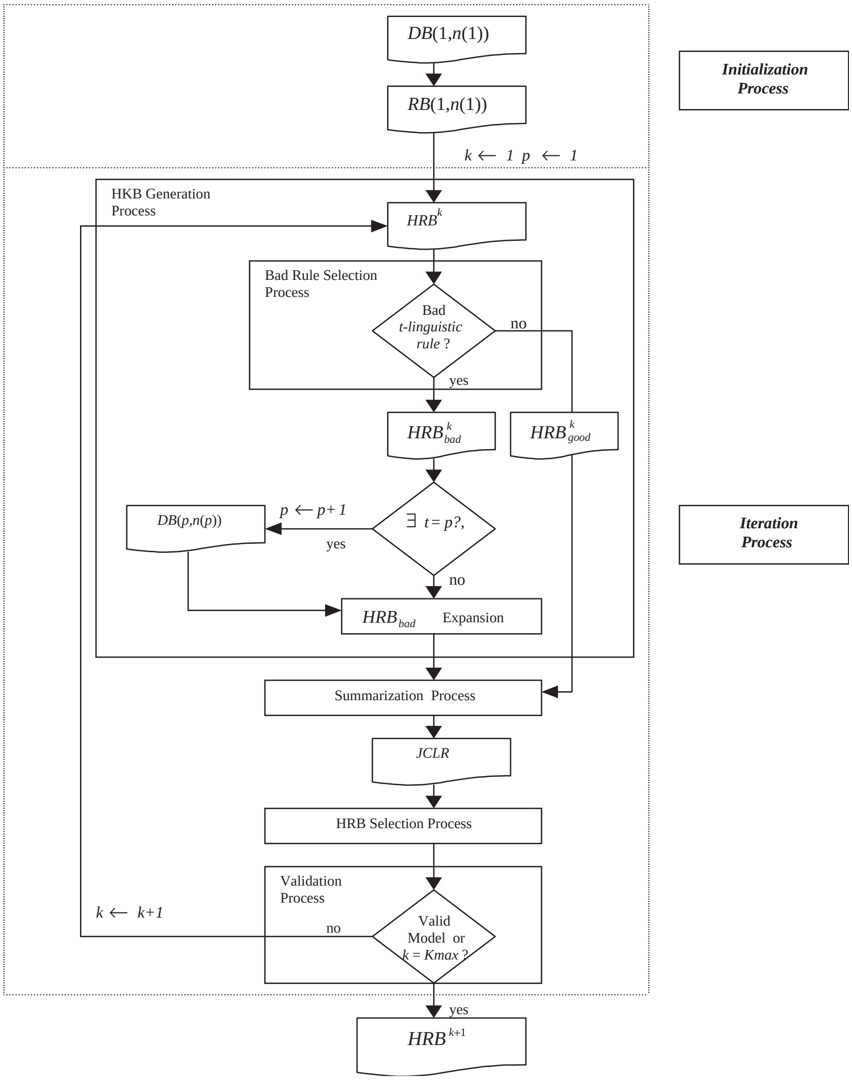

Fig. 6. Algorithm of the HSLR-LM design process.

What are the implications of these repeated and double-consequent rules in the hierarchical process? Are they usefully related with this process? Is the selection process powerful enough to take them away or to disambiguate them?

We will try to answer these questions after analyzing the mentioned consequences of applying the former policy and its influence in the obtained results.

3.2.1. Repeated $(t+1)$-linguistic rules

Consider the following situation where more than one copy of a rule can be produced by the generation process of the HSLR-LM in the same layer. This fact is illustrated in Fig. 7, where two

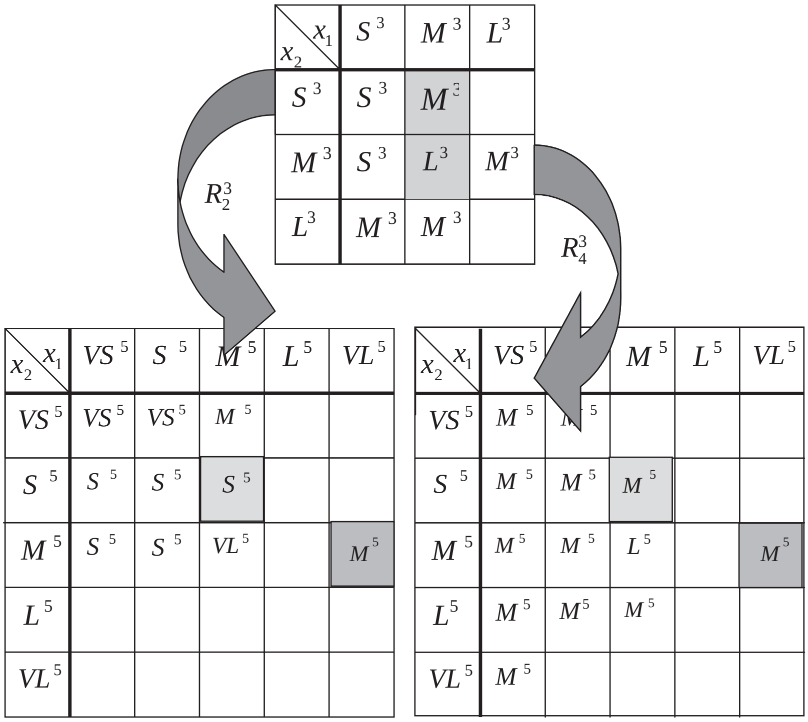

Fig. 7. Generation policy: repeated and double-consequent rules.

2-*linguistic rules* (see dark gray squares)

$$IF\ x_1 \text{ is } M^5 \text{ and } x_2 \text{ is } VL^5\ THEN\ y \text{ is } M^5$$

are both derived from the expansion of R_2^3 and R_4^3.

This happens because of the overlapping of the expanded rule images, which is produced by low values of the parameter δ (see Step 1(c)(i) in the algorithm). Once these repeated rules are generated, they are given to the selection process, which has the chance to eliminate all these redundant rules.

To answer the former questions and to decrease the computational complexity, we will experimentally compare the effect of excluding those repeated rules from the input of the selection process. We modify Step 3 of the HSLR-LM algorithm to extract repeated rules before the selection process takes place:

$$HRB^{k+1} = Selection(Extract\_Repeated(JCLR)).$$

In Section 4, we will evaluate and compare results obtained with and without considering repeated rules. From now on, models without repeated rules will be referred to as *NR-HSLRs*.

3.2.2. Double-consequent $(t+1)$-linguistic rules

As we have detected repeated linguistic rules in the last subsection, we can also observe that some of the learned rules have multiple consequents (see the two light gray squares in Fig. 7). As was introduced in [7,11,22], this phenomenon is an extension of the usual linguistic model structure where each combination of antecedents may have two or more consequents associated with it. We should note that this operation mode does not constitute an inconsistency from the interpolative reasoning point of view. However, it only corresponds to a shift of the consequent labels of the rules, producing final results lying in the intermediate zones between these fuzzy sets.

Consider the specific combination of antecedents of Fig. 7, "x_1 is S^5 and x_2 is M^5", which 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.

Double-consequent linguistic rules enrich the representational power of linguistic rules allowing different kinds of rules to belong to the HRB. Moreover, they postpone the selection of good rules until the summarization process is performed, considering the best cooperation between them.

3.3. Expansion policies: hierarchical replacement and hierarchical reinforcement

We have previously discussed in Fig. 3, more granularity implies more accuracy. As regards as the hierarchical process, the same question arises locally. The expansion policy followed by the algorithm in Step 1 locally replaces a bad modeling *t-linguistic rule* by a set of more specific $(t+1)$-*linguistic rules*. In this section we evaluate the performance of that policy and propose an alternative one.

In addition to the replacement criterion followed by the HSLR-LM, partial or incomplete solutions are also achieved as a consequence of the searching process implemented. The methodology implements a greedy strategy which makes the best available decision at every iteration. Therefore, the selection of *t-linguistic rules* at the current iteration is restricted by a maximum of p-hierarchical layers generated up to the last k iteration ($K_{\max}$), instead of having the complete set of rules generated from all possible HDBs. Moreover, some of the rules are not available because they were pruned by the former replacement strategy. From the above considerations, we affirm that HSLR-LM is not immune to the usual risk of hill-climbing searches without backtracking, i.e., converging to locally optimal solutions that are not globally optimal.

To deal with the former issues and being inspired by Ishibuchi et al.'s method [17], we propose a different operation mode for the hierarchical process. It consists on preserving both the expanded rule and some of the rules composing its image in the next layer rule base. That is, to consider the expansion process as a *hierarchical reinforcement of a bad rule*.

Fig. 8 shows both kinds of rule expansion policies and allows us to illustrate how the reinforcement extension (b) modifies our previous replacement approach (a). This hierarchical reinforcement is basically characterized by the following points:

- The HSLR-LM reinforces the original rule with more specific rules defined over some of its subspaces. The main purpose of these finer rules is to correct the original rule in those places where it performs a bad modeling by locally reinforcing these zones.
- The reinforcement policy does not eliminate the concept of "replacement" of the expanded rule, but extends it allowing the selection process to eliminate this rule when it badly cooperates with the rest of the rules. Thus, it gives the selection process the chance to perform a more accurate

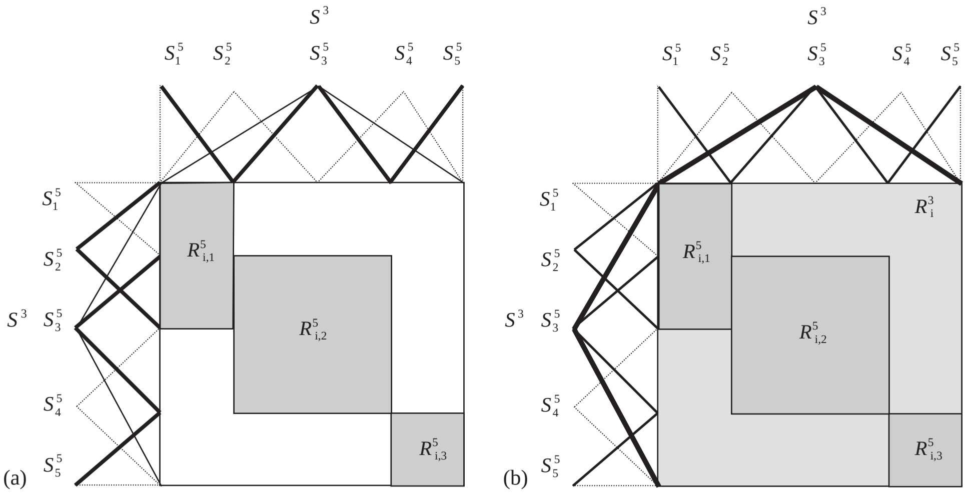

Fig. 8. Expansion policies: (a) replacement policy and (b) reinforcement policy.

search through the solution space in order to obtain the most accurate HRB. This fact may be seen as having a wider spectrum of possibilities to perform a selection decision.

- Reinforcement allows the methodology to backtrack to reconsider earlier choices which is impossible for the replacement approach. That is, if the global cooperation with the rules is not improved, *a decision which generates an expansion of a t-linguistic rule could be later corrected.* This happens by eliminating some or all of the $(t+1)$-*linguistic rules* generated. Then, *a bad rule is eliminated only if it is considered as bad by both processes*: *expanded by the generation process and discarded by the selection process.*
- The reinforcement approach allows the HSLR-LM to perform a sort of local bidirectional search which solves some of the problems of hierarchical clustering [14]. More specifically, a combined derivative and agglomerative clustering that can iteratively regulate how deep to search in the hierarchy.

In order to empirically prove the effect of this kind of refinement, we designed experiments by modifying Step 3 of the HSLR-LM algorithm (see Section 3.1) in the following way:

$$HRB^{k+1} = Selection(HRB^{k}_{\text{bad}} \cup JCLR).$$

Initially, this approach preserves the repeated rules considered in the previous subsection. Hence, we also designed experiments to appreciate the effect of excluding such rules by changing Step 3 of the algorithm:

$$HRB^{k+1} = Selection(Extract\_repeated(HRB^{k}_{\text{bad}} \cup \ JCLR)).$$

In Section 4, we will evaluate and compare results obtained for both types of expansion policies. From now on, models with hierarchical reinforcement will be referred as *HR-HSLRs*.

3.4. Selection policies: accuracy oriented and trade-off accuracy-complexity oriented

Unfortunately, although genetic algorithms constitute a robust technique, sometimes they cannot avoid to fall in local minima when strongly multimodal searching surfaces are considered. One of such complex environments is represented by the HKB which is composed of fuzzy rules defined on different granularity levels.

Models derived from non-optimal solutions are not accurate enough or/and contain redundant rules that make them more complex and thus, less interpretable. To partially avoid local minima solutions in the HKB, we modify the fitness function of the genetic algorithm. $F(C_j)$ was previously defined as an accuracy-oriented function that penalizes those rule bases which produce high errors. Now, it is updated by also penalizing rule sets with high number of rules. This new definition constitutes a trade-off solution between the complexity and the accuracy of the hierarchical model [17]. The fitness function can be re-written in the following way:

$$F(C_j) = w_1 \; MSE + w_2 \; N_{\text{rules}},$$

where *MSE* is the mean squared error produced by the current rule base encoded in the chromosome, N_{rules} is the number of rules of that rule base, and w_1 and w_2 are weights defining the relative importance of each objective. In our present experiments, these constants are initialized in the following way [10]:

$$w_1 = 1.0, \quad w_2 = 0.1\,\frac{MSE_{\text{initial}}}{N_{\text{initial rules}}},$$

where MSE_{initial} and $N_{\text{initial rules}}$ are the error and the amount of rules of the original rule base to be summarized, respectively.

It should be noted that the above definition of the fitness function does not only reduce the complexity of the model but, sometimes, increases its accuracy by working as a pruning strategy [21]. In addition to the present modification, other interestingness relations can also be implemented in HSLR-LM to enrich the summarization process (see Fig. 2).

In the following section, we will evaluate and compare results obtained from both types of selection policies. From now on, models using the accuracy-complexity-oriented strategy will be referred as *AC-HSLRs*.

4. Examples of application: experiments and analysis of results

With the aim of analyzing the behavior of the proposed iterative methodology, two real-world electrical engineering distribution problems from Spain [9,29,30] have been selected. The first one relates some characteristics of a certain village with the actual length of low-voltage line contained in it. The other relates the maintenance cost of the network installed in certain towns with some of their characteristics. In both cases, it would be better if the solutions obtained were not only numerically accurate, but also able to explain how these values are computed for certain villages or towns. In other words, it is important that solutions would be interpreted by human beings to some degree.

In order to do this, we have organized this section into four parts: a first part of notation and parameters; a second and third of experiments; and a final one with an analysis of the results. The analysis will be done from different perspectives: the methodology performance, the influence of its parameters, and the policies considered in its design.

4.1. Notation and parameters

As we have said, the learning methodology has been thought of as a refinement of simple linguistic models, which uses an HKB of some layers. For the sake of simplicity and interpretability, we will only consider the generation of HSLRs with up to three hierarchical levels.

In the following applications we are going to refer to these experiments produced by the HSLR-LM using the following notation:

$$\text{HSLR}(\textit{LRG-method}, n(1), n(p), K_{\max}),$$

where $n(1)$ and $n(p)$ are the initial and final granularity levels of the HKB, respectively; $K_{\max}$ is the number of iterations performed by this methodology; e.g., $HSLR$ (WM-method, 3, 9, 2). We should note that $HSLR(LRG\text{-method}, n(1), n(2), 1)$ represents a simple refinement of hierarchical models which is an interpretability-oriented approach of two levels.

The LRG-method considered for the previous experimentation is the one proposed by Wang and Mendel [33], noted by WM-method in the following. This method is briefly described in the Appendix A1. A reference to an application of WM-method is represented by $WM\text{-}method(r)$, where r is the granularity level of the linguistic partitions used in the method.

The initial databases used for the HSLR-LM have two primary linguistic partitions formed by *three* and *five linguistic terms* with triangular-shaped fuzzy sets, i.e. $DB(1,3)$ and $DB(1,5)$, respectively. The initial linguistic term sets for the mentioned databases are shown in the following:

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

where S = small, M = medium, L = large, VS = very small, VL = very large.

The parameters used in all of these experiments are listed in Table 3.

The results obtained in the experiments developed are collected in tables where MSE_{tra} and MSE_{tst} stand for the MSE values computed over the training and test data sets, respectively. $\#R$ stands for the number of simple rules in the corresponding HRB. $\#Dif$. represents a subset of $\#R$ with non-repeated or different rules, which becomes the real number of processed rules. Notice that these rules do not increase the computational cost of the process. They are processed only once in the inference process and the result is multiplied by their number of occurrences.

Different types of HSLR models will be evaluated by considering those parameter values which allow us to clarify some aspects of the methodology:

- *HSLR* models generated from different factors of expansion (α) to evaluate the proper levels of bad performance to be considered for a rule expansion (10%, 50% and 90% more than the entire MSE of the HRB^k).

Table 3
Parameter values

Parameter	Decision
Generation	
δ, $n(t+1)$-*linguistic partition* terms selector	0.1
τ, used to calculate E_i	0.5
α, used to decide the expansion of rule	1.1, 1.5, 1.9
Genetic algorithms selection	
Number of generations	600–2000
Population size	61
Mutation probability	0.1–0.2
Crossover probability	0.6

Table 4
HSLR-LM methods used in the experiments

Method	Policies		
	Generation	Expansion	Selection
HSLR	Repeated	Replacement	Accuracy
NW-HSLR	Non-repeated	Replacement	Accuracy
HR-HSLR	Repeated	Reinforcement	Accuracy
HR-NW-HSLR	Non-repeated	Reinforcement	Accuracy
AC-HSLR	Repeated	Replacement	Accuracy-complexity
AC-NW-HSLR	Non-repeated	Replacement	Accuracy-complexity
AC-HR-HSLR	Repeated	Reinforcement	Accuracy-complexity
AC-HR-NW-HSLR	Non-repeated	Reinforcement	Accuracy-complexity

- *HSLR* models designed from different number of iterations (K_{max}) to evaluate the effect of having hierarchical rules with different granularity levels ($K_{max} = 1, 2$).

In Table 4, we add some notation which suggests representative suffixes for the models generated in the experiments.

Finally, we will try to solve the former applications by generating different kinds of models: classical regression, neural models and a global linguistic approach based on adapting Ishibuchi et al.'s method for classification tasks [17] to learn an HKB:

- To apply classical regression, the parameters of the polynomial models were fit by Levenberg–Marquardt. Exponential and linear models were fit by linear least squares.
- The multilayer perceptron was trained with the QuickPropagation algorithm. The number of neurons in the hidden layer was chosen to minimize the test error [9,30].
- For the sake of simplicity, in this Subsection we will refer to the models obtained by a global linguistic approach as global HSLR (G-HSLR), in order to distinguish them from our local approach

Table 5
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

(HSLR-LM). The global approach obtains an HSLR by creating several hierarchical linguistic partitions with different granularity levels. It generates the complete set of linguistic rules in each of these partitions, takes the union of all of these sets, and finally, performs a genetic rule selection process on the whole rule set.

4.2. Computing the length of low-voltage lines

With the aim of measuring the amount of electricity lines that an electric company owns, a relationship was searched in [9,29,30] between the variables of Table 5. To compare different models we have randomly divided the original sample of 495 rural nuclei into two sets comprising 396 and 99 samples, labeled training and test, respectively. The results obtained with our HSLR-LM with different values for the expansion factor α are shown in Tables 6 and 7. Finally, comparisons with other techniques are shown in Table 8.

4.3. Computing the maintenance costs of medium-voltage line

We were provided with data concerning four different characteristics of the towns (see Table 9) related to their minimum maintenance cost in a sample of 1059 simulated towns [9,30]. The samples have been randomly divided into two sets comprising 847 and 212 samples, 80% and 20% of the whole data set, labeled training and test, respectively.

The results obtained with our HSLR-LM with different values for the expansion factor α are shown in Tables 10 and 11. Comparisons with other techniques are shown in Table 12.

4.4. Analysis of results

In view of the results obtained in the experiments, we should remark some important conclusions from different perspectives. First, the general results of the methodology performance are discussed. Second, an analysis of the influence of the parameters is performed. Finally, a more detailed description and interpretation of the results obtained from different policies are done.

4.4.1. Analysis of the methodology performance

Let us analyze the obtained results from different points of view:

- *From the accuracy point of view*: The different models generated from HSLR-LM in both electrical problems clearly outperform in MSE_{tra} and MSE_{tst} those ones obtained by the WM-method, in all iteration and factor of expansion levels (see Tables 6, 7, 10 and 11). They also outperform classical

Table 6
Results obtained in the low-voltage electrical application considering $\alpha = 1.1$

Method	#R	#Dif.	MSE_{tra}	MSE_{tst}
WM-method(3)	**7**	**7**	594 276	626 566
WM-method(5)	13	13	298 446	282 058
WM-method(9)	29	29	197 613	283 645
HSLR(WM-method,3,5,1)	12	12	178 950	167 318
NR-HSLR(WM-method,3,5,1)	13	13	178 950	167 318
HR-HSLR(WM-method,3,5,1)	12	12	175 619	162 873
HR-NR-HSLR(WM-method,3,5,1)	12	12	175 619	162 873
AC-HSLR(WM-method,3,5,1)	11	11	180 111	166 210
AC-NR-HSLR(WM-method,3,5,1)	11	11	180 111	166 210
AC-HR-HSLR(WM-method,3,5,1)	10	10	176 781	161 764
AC-HR-NR-HSLR(WM-method,3,5,1)	10	10	176 781	161 764
HSLR(WM-method,3,9,2)	44	35	153 976	165 458
NR-HSLR(WM-method,3,9,2)	31	31	155 423	171 241
HR-HSLR(WM-method,3,9,2)	41	35	**153 237**	171 606
HR-NR-HSLR(WM-method,3,9,2)	25	25	154 411	**156 197**
AC-HSLR(WM-method,3,5,2)	30	25	157 761	165 411
AC-NR-HSLR(WM-method,3,5,2)	23	23	158 478	171 546
AC-HR-HSLR(WM-method,3,5,2)	27	25	158 775	163 774
AC-HR–NR-HSLR(WM-method,3,5,2)	22	22	158 935	163 723

regression, neural networks and global linguistic models in the approximation of both data sets, training and test (see Tables 8 and 12).

- *From the complexity point of view*: HSLR-LM has obtained relatively simple models for the problems with respect to the accuracy improvements achieved ($\%_{tra}, \%_{tst}$) over the initial models generated by the WM-method. In most of the cases, the models obtained from HSLR-LM are even simpler than the WM-method ones while having an important improvement in both errors (see Tables 6 and 7 with $K_{max} = 1, 2$).

  The high-order electrical problem, with much more accurate and complex results, can be an exception when more than a single iteration is performed (see Tables 10 and 12 with $K_{max} = 2$). However, alternative solutions that also outperform the models generated from the remaining techniques with a simpler structure were proposed (see options AC in Tables 10 and 11).

  Moreover, even having a higher number of rules, the HKB gives a hierarchical order which can be used in the sense of interpretability. In other words, human beings cannot understand hundreds of different rules, but can associate a group of them with a 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.
- *From the scalability point of view*: Although we have shown experiments with a simple LRG-method like the WM-method, more complex fuzzy rule learning methods can be used.

Table 7
Results obtained in the low-voltage electrical application considering $\alpha = 1.5$

Method	#R	#Dif.	MSE_{tra}	MSE_{tst}
WM-method(3)	**7**	**7**	594 276	626 566
WM-method(5)	13	13	298 446	282 058
WM-method(9)	29	29	197 613	283 645
HSLR(WM-method,3,5,1)	12	12	178 950	167 318
NR-HSLR(WM-method,3,5,1)	13	13	178 950	167 318
HR-NR-HSLR(WM-method,3,5,1)	12	12	175 619	162 873
HR-HSLR(WM-method,3,5,1)	12	12	175 619	162 873
AC-HSLR(WM-method,3,5,1)	11	11	180 111	166 210
AC-NR-HSLR(WM-method,3,5,1)	11	11	180 111	166 210
AC-HR-HSLR(WM-method,3,5,1)	10	10	176 781	161 764
AC-HR-NR-HSLR(WM-method,3,5,1)	10	10	176 781	161 764
HSLR(WM-method,3,9,2)	34	28	**153 962**	164 377
NR-HSLR(WM-method,3,9,2)	28	28	156 935	173 396
HR-HSLR(WM-method,3,9,2)	42	34	154 820	167 110
HR-NR-HSLR(WM-method,3,9,2)	24	24	156 378	**158 065**
AC-HSLR(WM-method,3,5,2)	25	22	157 722	161 510
AC-NR-HSLR(WM-method,3,5,2)	22	22	158 839	165 190
AC-HR-HSLR(WM-method,3,5,2)	26	25	158 929	168 667
AC-HR–NR-HSLR(WM-method,3,5,2)	23	23	161 071	165 091

Table 8
Results obtained in the low-voltage electrical application compared with other techniques

Method	MSE_{tra}	MSE_{tst}	Complexity
Linear	287 775	209 656	7 nodes, 2 par.
Exponential	232 743	197 004	7 nodes, 2 par
Second order polynomial	235 948	203 232	25 nodes, 2 par.
Third order polynomial	235 934	202 991	49 nodes, 2 par.
Three layer perceptron 2–25–1	169 399	167 092	102 par.
G-HSLR(WM-method,3,9,2)	159 851	189 119	31 rules
AC-HR-HSLR(WM-method,3,5,1)	176 781	161 764	10 rules
HR-NR-HSLR(WM-method,3,9,2)	**154 411**	**156 197**	25 rules

In the Appendix A2, we show results using another inductive LRG-method proposed by Thrift [32]. These results confirm the quality of the HSLR-LM which, as a meta-methodology, obtains accurate refinements from simple models generated by different LRG-methods.

- *From the locality point of view*: The linguistic models generated from HSLR-LM overcome the ones performed by G-HSLR-LM in the approximation of the training and test sets. We should

Table 9
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 10
Results obtained in the medium-voltage electrical application considering $\alpha = 1.1$

Method	#R	#Dif.	MSE_{tra}	MSE_{tst}
WM-method(3)	**27**	**27**	150 545	125 807
WM-method(5)	64	64	70 908	77 058
WM-method(9)	130	130	32 191	33 200
HSLR(WM-method,3,5,1)	193	84	22 358	23 755
NR-HSLR(WM-method,3,5,1)	79	79	28 087	27 495
HR-HSLR(WM-method,3,5,1)	205	86	20 588	22 583
HR-NR-HSLR(WM-method,3,5,1)	79	79	28 087	27 495
AC-HSLR(WM-method,3,5,1)	159	69	22 557	24 679
AC-NR-HSLR(WM-method,3,5,1)	44	44	29 182	28 236
AC-HR-HSLR(WM-method,3,5,1)	185	67	20 752	21 005
AC-HR-NR-HSLR(WM-method,3,5,1)	54	54	30 445	32 897
HSLR(WM-method,3,9,2)	1628	556	11 229	12 650
NR-HSLR(WM-method,3,9,2)	369	369	12 677	13 767
HR-HSLR(WM-method,3,9,2)	1900	573	**9 843**	10 998
HR-NR-HSLR(WM-method,3,9,2)	393	390	11 769	**10 703**
AC-HSLR(WM-method,3,5,2)	1367	555	10 450	10 710
AC-NR-HSLR(WM-method,3,5,2)	167	167	12 807	13 390
AC-HR-HSLR(WM-method,3,5,2)	1430	486	10 334	10 954
AC-HR-NR-HSLR(WM-method,3,5,2)	143	143	15 881	18 168

note that the global approach, which was inspired in [17], has been described as a limited strategy (see high errors in Tables 8 and 12) derived from directly putting rules with different granularity in the same bag and making a selection on it. Hierarchical and hybrid fuzzy systems and genetic algorithms require more than simple combinations derived from putting everything together, but a more sophisticated analysis and design of the system components and their features [13]. HSLR becomes a generalization of G-HSLR and an open methodology that can still be improved in many ways by adding and properly combining different interestingness relations (see Fig. 2).

Table 11
Results obtained in the medium-voltage electrical application considering $\alpha = 1.9$

Method	#R	#Dif.	MSE_{tra}	MSE_{tst}
WM-method(3)	**27**	**27**	150 545	125 807
WM-method(5)	64	64	70 908	77 058
WM-method(9)	130	130	32 191	33 200
HSLR(WM-method,3,5,1)	107	59	29 336	29 657
NR-HSLR(WM-method,3,5,1)	53	53	34 870	39 367
HR-HSLR(WM-method,3,5,1)	115	66	29 119	31 949
HR-NR-HSLR(WM-method,3,5,1)	53	53	34 870	39 367
AC-HSLR(WM-method,3,5,1)	83	53	32 623	33 924
AC-NR-HSLR(WM-method,3,5,1)	40	40	42 826	42 100
AC-HR-HSLR(WM-method,3,5,1)	78	51	35 139	38 497
AC-HR-NR-HSLR(WM-method,3,5,1)	45	45	37 750	42 152
HSLR(WM-method,3,9,2)	688	347	14 825	15 016
NR-HSLR(WM-method,3,9,2)	294	294	16 717	16 941
HR-HSLR(WM-method,3,9,2)	969	462	**12 051**	**12 922**
HR-NR-HSLR(WM-method,3,9,2)	281	279	14 999	14 497
AC-HSLR(WM-method,3,5,2)	258	155	16 221	17 630
AC-NR-HSLR(WM-method,3,5,2)	121	121	17 658	18 378
AC-HR-HSLR(WM-method,3,5,2)	292	189	13 428	13 457
AC-HR-NR-HSLR(WM-method,3,5,2)	167	167	16 983	20 064

Table 12
Results obtained in the medium-voltage electrical application compared with other techniques

Method	MSE_{tra}	MSE_{tst}	Complexity
Linear	164 662	36 819	17 nodes, 5 par.
2th order polynomial	103 032	45 332	77 nodes, 15 par.
3 layer perceptron 4-5-1	86 469	33 105	35 par.
G-HSLR(WM-method,3,9,2)	24 335	21 714	135 rules
AC-HR-HSLR(WM-method,3,5,1)	20 752	21 005	67 rules
HR-HSLR(WM-method,3,9,2)	**9 843**	**10 998**	573 rules

4.4.2. Analysis of the influence of the methodology parameters

Now let us go deeply into the analysis of results by considering the effects of applying those different parameter values described in Section 3 (see Fig. 3). Let us first analyze how different values for the factor of expansion (α) and the number of iterations (K_{max}) make their influence on the final hierarchical models:

- *Iteration level* (K_{max}). We should note that all those models obtained by the use of more than one iteration perform the best approximation in MSE_{tra}. However, we should make a difference between both examples when generalization is considered:

- ◦ In simple problems with a quite similar distribution between training and test sets, like the medium-voltage electrical application, the MSE_{tst} is also overcome by the more iterative models (see Tables 10 and 12).
- ◦ In other more complex problems, such as low-voltage electrical application, a more iterative configuration can overfit the system modeled (see Tables 6 and 7 with $K_{max} = 2$). We should note that initial partitions with a higher granularity do not ensure more accuracy (see also WM-method(9) and WM-method(5) in Table 6).

To deal with this problem, at least two different pruning techniques are implemented in the methodology [21]:

- ◦ Pre-pruning strategy, where the regulation of factor α is performed [23].
- ◦ Post-pruning strategy, where non-repeated rules as a generation policy and the accuracy-complexity orientation as a selection policy are used in an iterative way.

- *Factor of expansion* (α). 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 different values for this parameter. However, some special features could be remarked regarding the α setting. As a general rule, when α grows up, the system complexity decreases, i.e., less rules are expanded and thus a simpler HRB is finally obtained. However, when accuracy is considered, an increase in the number of rules does not always ensure a decrease in the model *MSE*. A good cooperation among such rules is also needed.

  The parameter α can be considered to design models with a different balance between accuracy and description (as said, the higher its value, the lower the number of rules, and hence the more descriptive the system). In this sense, different situations are illustrated in Figs. 9–11 for $K_{max} = 1$.

  For example, we find a good trade-off solution between accuracy and interpretability in Fig. 9. In this graph we can observe that the most accurate model for the low-voltage problem is obtained by means of the HSLR(WM-method,3,5,1), which is composed of only 12 rules. This idea can also be observed in the results of the medium-voltage electrical application as shown in Fig. 10. Here, the user can also decide between models with a different treatment of the description-accuracy trade-off:

  - ◦ When accuracy is preferred to description, the best choice would be the model obtained when considering $\alpha = 1.1$, i.e., the most accurate one.
  - ◦ When a compromise solution between accuracy and description is preferred, the models obtained from HSLR(WM-method,3,5,1) with $\alpha = 1.9$ and 3.5 would be two very good solutions; both are the simpler models (59 and 58 rules, respectively) with lesser rules than WM-method(5).
  - ◦ Finally, when accuracy is definitively the only modeling requirement, there would be another choice for some kinds of problems. Fig. 11 shows a different way to deal with the accuracy-description trade-off. Significantly, more accurate models are obtained for the latter problem using initial partitions with a higher granularity level like five. Of course, the models generated by HSLR(WM-method,5,9,1) starting from these partitions are very complex and thus very

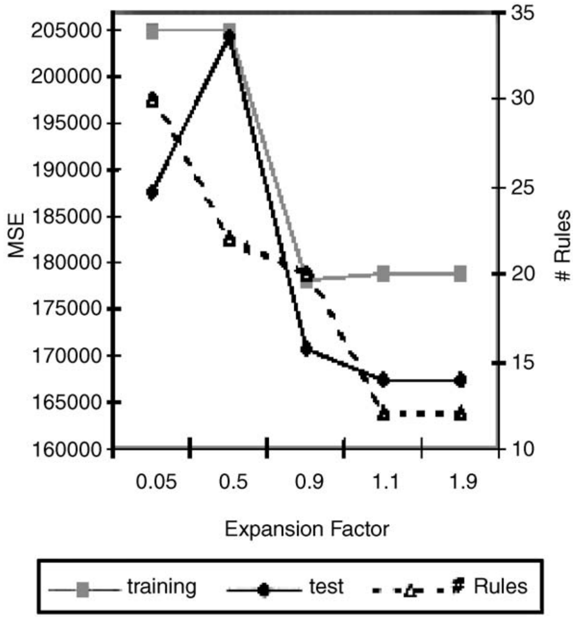

Fig. 9. *HSLR*(WM-method,3,5,1) *MSE* tendency using different values for α and their complexity in the low-voltage application.

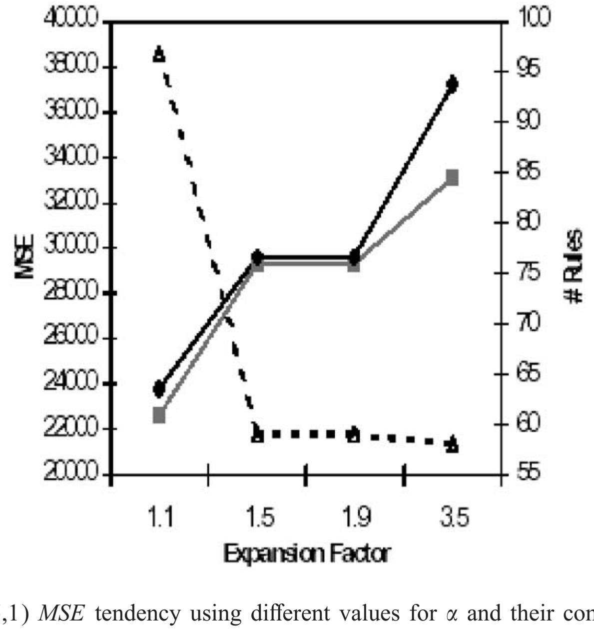

Fig. 10. *HSLR*(WM-method,3,5,1) *MSE* tendency using different values for α and their complexity in the medium-voltage application.

difficult to be interpreted. Even in this case, still, a simpler and more accurate model than WM-method(9) can be found with $\alpha = 5.5$ (121 against 130 rules).

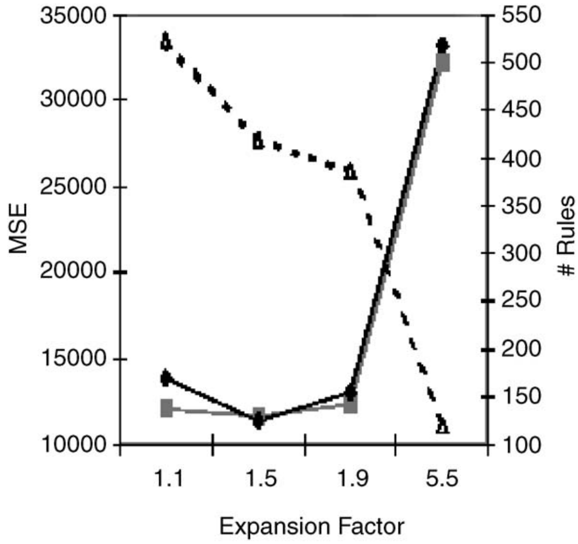

Fig. 11. *HSLR*(WM-method,5,9,1) *MSE* tendency using different values for α and their complexity.

From the generalization point of view, the factor of expansion also serves as a pre-pruning strategy that can be used to prevent overfitting. We could choose higher values of α in order to expand a lesser number of rules. This will cause a worse model fitting on the training examples, but a better one on the test set (compare the same models in Tables 6 and 7 and in Tables 10 and 11).

4.4.3. *Analysis of the methodology policies*

Let us analyze the influence of the different policies considered in the hierarchical process.

- *Generation policy.*
  - *Weighted and non-weighted rules*: We should note that all of those models obtained by the use of repeated rules perform the best approximation in MSE_{tra}. Moreover, some of them perform a significant reduction in the error in comparison with those models that eliminate repeated rules.

    As mentioned in Section 3.2.1, 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 is observed that sometimes it preserves some of them (see the difference between $\#R$ and $\#Dif.$ in the tables). This fact produces a sort of *reinforcement* on the whole subspace of the rule—a global refinement action—and can be interpreted as a *weight* on that rule. We should note some important aspects of weighted rules:
    - Weighted rules do not excessively increase the computational cost of the process because they are processed only once when the inference takes place. More specifically, the defuzzified value of a rule is multiplied by its occurrences, i.e., its *weight*.

– The use of weighted rules produces good approximation results, and its influence lies on the defuzzification method. In our case, we use the minimum t-norm in the role of conjunctive and implication operation and the *center of gravity weighted by the matching degree* [8] as a defuzzification strategy. Any other defuzzification method considering the matching degree of the rules fired may be used. Thus, we modify the computation of the final output given by the system y^*. This value is calculated by aggregating the partial actions obtained by means of the matching degree weighted average:

$$y^* = \frac{w_1 h_1 y_1 + \cdots + w_j h_j y_j + \cdots + w_T h_T y_T}{w_1 h_1 + \cdots + w_j h_j + \cdots + w_T h_T},$$

where w_j is the number of times that the rule j is repeated, h_j is the matching degree of the rule j, and y_j the center of gravity for each individual fuzzy set B_j.

– In spite of their good performance, weighted rules can overfit the system modeled when they are combined with a hierarchical process and a *reinforcement expansion policy* (see results with $K_{\max} = 2$ in Tables 6, 7, and 10). We can decide not to use them, and thus, allow the algorithm to perform an iterative post-pruning strategy which could produce the most proper generalization in some kinds of problems.

◦ *Double-consequent rules*: As we have detected weighted reinforced rules in HSLRs, we can also observe that some of the learned rules have multiple consequents. Again, this kind of rules can also be interpreted as a reinforcement performed in the whole space of the rule.

• *Expansion policies*: *replacement and reinforcement*. We should note that almost all of those models obtained by the use of the reinforcement policy with more than two hierarchical levels and initial partitions with low granularity levels—by using or deleting weighted rules to avoid overfitting—perform the best approximation in MSE_{tra} and MSE_{tst}. Moreover, more independence from the parameters of the algorithm could also be achieved:

◦ *Independence from the granularity of the initial partitions*: As we have previously seen in Fig. 11, it may happen that proper initial partitions with higher granularity levels could generate more accurate results for a specific problem. As mentioned in [10], finding these partitions is a very hard task. The obtained results show that a reinforcement policy combined with a hierarchical process is the best competitive strategy to deal with the former situations. More specifically, this approach makes HSLRs *more independent from the initial partitions*, by starting with low granular ones and continuously performing gradual and iterative improvements on them (see Tables 10 and 11).

◦ *Independence from the factor of expansion* α: Complex real problems, such as the low-voltage electrical application, present anomalies due to its high non-linearity which requires a proper factor of expansion (e.g. low values tends to overfit the system modeled). The use of a reinforcement policy implements a sort of revocable strategy in the HSLR-LM that would make α less important, allowing the process to be performed in a more accurate way.

Finally, the reinforcement expansion policy can also be seen as a sort of *default reasoning* (see Fig. 8(b)). That is, a general and less specific *t-linguistic rule* can be always activated. However, some of those more specific $(t+1)$-*linguistic rules*, which reinforce the former rule, sometimes participate in the activation. Thus, these more specific rules joined with the former default one perform the final system inference as an exception approach.

- *Selection policy*: *accuracy-oriented and accuracy-complexity oriented.* As we expected, the fitness function introduced in Section 3.4 allows us to generate simpler models and perform a trade-off between complexity and accuracy (see AC options in Tables 6, 7, 10 and 11). Moreover, sometimes it also works as a pruning strategy that could prevent the system overfitting (see AC option with $K_{max} = 1$ in Table 6). That is, a kind of post-pruning [21] rule selection process which, in the methodology context, does not only consider the quality of the approximation performed by each rule but also its global cooperation with the whole set.

5. Concluding remarks

In this paper, hierarchical linguistic models are viewed as a class of local modeling approaches which attempt to solve a complex modeling problem by decomposing it into a number of simpler subproblems. Fuzzy set theory offers excellent tools for representing the uncertainty associated with the decomposition task, providing smooth transitions between individual local submodels. From this perspective, HSLRs have been proposed as a parameterized solution that achieves a desired balance between the complexity and the accuracy of the systems modeled, effectively exploring their non-linearity and non-uniformity.

We designed HSLR-LM as a learning meta-methodology for identifying hierarchical linguistic models. It performs gradual and local-oriented refinements on problem subspaces that are badly modeled by previous models—rather than in the whole problem domain. Moreover, it integrates the improved local behavior with the whole model by summarization processes which ensure a good global performance.

Finally, as Goldberg said [13], if the future of Computational Intelligence "*lies in the careful integration of the best constituent technologies*", hierarchical and hybrid fuzzy systems and genetic algorithms require more than simple combinations derived from putting everything together. However, they need a more sophisticated analysis and design of the system components and their features. This paper presents progresses in a research program devoted to find the most proper system integration and to explore the HSLRs capabilities.

Appendix A

A.1. WM rule generation method

The inductive rule base generation process proposed by Wang and Mendel [33] 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 a previous definition of the database—input–output primary linguistic partitions. The linguistic rule structure considered is the usual Mamdani-type rule with m input variables and one output variable.

The generation of the linguistic rules of this kind is performed by putting into effect the three following 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 $R = IF\ x_1\ is\ S_1\ and \ldots\ and\ x_m\ is\ S_m\ THEN\ y\ is\ B$ 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 matching degree associated to it will be obtained as follows: $G(R) = \mu_{S_1}(x_1^l) \ldots \mu_{S_m}(x_m^l)\mu_B(y^l)$.
3. *To obtain a final rule base from the preliminary linguistic rule set*. If all rules presenting the same antecedent values have associated the same consequent in the preliminary set, this linguistic rule is automatically put (only once) into the final rule base. Otherwise, if there are conflicting rules with the same antecedent and different consequent values, the rule considered for the final rule base will be the one with the highest matching degree.

A.2. 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 [32] 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 $\{NB, NS, ZR, PS, PB\}$ be the term set associated with 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 $1, \ldots, PB$ with 4 and the blank symbol "–" with 5.

Therefore, the genetic algorithms employ 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 rule base.

The genetic algorithms 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 rule base coded in its genotype.

As said, HSLR-LM was thought of as a meta-methodology designed to operate on different LRG-methods. In Table 13 we present results using the LRG-method proposed by Thrift [32] to evaluate its behavior.

We can observe again that the HSLR-LM has outperformed the basic LRG-method, the THR-method in this case. The conclusions drawn in the analysis of results performed in the main part of the paper, remain in view of the results shown in Table 13.

Table 13
Results obtained in the low-voltage electrical application considering THR-method and $\alpha = 1.1$

Method	#R	#Dif.	MSE_{tra}	MSE_{tst}
THR-method(3)	**7**	**7**	266 369	248 257
THR-method(5)	25	25	218 857	217 847
HSLR(THR-method,3,5,1)	32	27	174 020	174 428
NR-HSLR(THR-method,3,5,1)	24	24	178 434	178 759
HR-HSLR(THR-method,3,5,1)	35	29	173 161	169 272
HR-NR-HSLR(THR-method,3,5,1)	22	22	171 005	170 638
HSLR(THR-method,3,9,2)	62	53	154 524	**153 765**
NR-HSLR(THR-method,3,9,2)	51	51	163 613	167 700
HR-HSLR(THR-method,3,9,2)	63	53	**153 245**	157 236
HR-NR-HSLR(THR-method,3,9,2)	36	36	153 773	181 680

Table 14
Description of the acronyms

Acronym	Meaning
AC	Trade off accuracy-complexity-oriented policy
FRBS	Fuzzy rule-based system
G-HSLR	Global hierarchical systems of linguistic rules
HKB	Hierarchical knowledge base
HDB	Hierarchical database
HR	Hierarchical reinforcement policy
HRB	Hierarchical rule base
HSLR-LM	Hierarchical systems of linguistic rules learning methodology
LRG-methods	Linguistic rule generation methods
NR	Non-weighted rules policy
THR	Thrift’s linguistic rule generation method
WM	Wang and Mendel’s linguistic rule generation method

A.3. Acronyms

In Table 14, we list the acronyms used in this paper and their corresponding meanings.

References

[1] R. Babuška, Fuzzy Modeling for Control, Kluwer Academic Publishers, Dordrecht, 1998.
[2] A. Bardossy, L. Duckstein, Fuzzy Rule-Based Modeling with Application to Geophysical, Biological and Engineering Systems, CRC Press, Boca Raton, FL, 1995.
[3] A. Bastian, How to handle the flexibility of linguistic variables with applications, Internat. J. Uncertainty, Fuzziness Knowledge-Based Systems 2 (4) (1994) 463–484.
[4] J.C. Bezdek, Fuzzy Mathematics in pattern classification, Ph.D. Thesis, Cornell University, 1973.

[5] J.C. Bezdek, Fuzzy clustering, in: E.H. Ruspini, P.P. Bonisone, W. Pedrycz (Eds.), Handbook of Fuzzy Computation, Institute of Physics Press, 1998, pp. f6.1:1–f6.6:19.
[6] J.C. Bezdek, S. Pal (Eds.), Fuzzy Models for Pattern Recognition: Methods that Search for Structures in Data, IEEE Press, New York, 1992.
[7] O. Cordón, F. Herrera, A proposal for improving the accuracy of linguistic modeling, IEEE Trans. Fuzzy Systems 8 (4) (2000) 335–344.
[8] O. Cordón, F. Herrera, A. Peregrín, Applicability of the fuzzy operators in the design of fuzzy logic controllers, Fuzzy Sets and Systems 86 (1997) 15–41.
[9] O. Cordón, F. Herrera, L. Sánchez, Solving electrical distribution problems using hybrid evolutionary data analysis techniques, Appl. Intell. 10 (1999) 5–24.
[10] O. Cordón, F. Herrera, P. Villar, A genetic learning process for the scaling factors, granularity and contexts of the fuzzy rule-based system data base. Inform. Sci., 2001, to appear.
[11] O. Cordón, F. Herrera, I. Zwir, Linguistic modeling by Hierarchical systems of linguistic rules, IEEE Trans. Fuzzy Systems 10 (1) (2002) 2–20.
[12] A.E. Gegov, P.M. Frank, Hierarchical fuzzy control of multivariable systems, Fuzzy Sets and Systems 72 (1995) 299–310.
[13] D.E. Goldberg, A meditation on the computational intelligence and its future, Illigal Report #2000019, Department of General Engineering, University of Illinois at Urbana-Champaign, 2000.
[14] D.J. Hand, Discrimination and Classification, Wiley, New York, 1992.
[15] F. Herrera, M. Lozano, J.L. Verdegay, A learning process for fuzzy control rules using genetic algorithms, Fuzzy Sets and Systems 100 (1998) 143–158.
[16] H. Ishibuchi, K. Nozaki, H. Tanaka, Efficient fuzzy partition of pattern space for classification problems, Fuzzy Sets and Systems 59 (1993) 295–304.
[17] H. Ishibuchi, K. Nozaki, N. Yamamoto, H. Tanaka, Selecting fuzzy if–then rules for classification problems using genetic algorithms, IEEE Trans. Fuzzy Systems 3 (3) (1995) 260–270.
[18] R. Krishnapuram, J. Keller, A possibilistic approach to clustering, IEEE Trans. Fuzzy Systems (1993) 98–110.
[19] C.T. Leondes (Ed.), Fuzzy Theory Systems, Techniques and Applications, Academic Press, New York, 2000.
[20] R.S. Michalski (Ed.), Multistrategy Learning, Kluwer Academic Press, Dordrecht, 1993.
[21] T. Mitchell, Machine Learning, McGraw-Hill, New York, 1997.
[22] K. Nozaki, H. Ishibuchi, H. Tanaka, A simple but powerful heuristic method for generating fuzzy rules from numerical data, Fuzzy Sets and Systems 86 (1997) 251–270.
[23] T. Oates, D. Jensen, The effects of training set size on decision tree complexity, Proc. 14th Internat. Conf. on Machine Learning 1997, pp. 254–262.
[24] W. Pedrycz (Ed.), Fuzzy Modelling: Paradigms and Practice, Kluwer Academic Press, Dordrecht, 1996.
[25] C.V.S. Raju, J. Zhou, Adaptative hierarchical fuzzy controller, IEEE Trans. Systems Man Cybernet. 23 (4) (1993) 973–980.
[26] E.H. Ruspini, A new approach to clustering, Inform. and Control 15 (1) (1969) 22–32.
[27] E.H. Ruspini, Recent developments in fuzzy clustering, in: R.R. Yager (Ed.), Fuzzy Sets and Possibility Theory, Recent Developments, Pergamon Press, Oxford, 1982, pp. 133–147.
[28] E.H. Ruspini, I. Zwir, Automated qualitative description of measurements, Proc. 16th IEEE Instrumentation and Measurement Technology Conf., Venice, Italy, 1999.
[29] L. Sánchez, Study of the Asturias rural and urban low-voltage network, Technical Report, Hidroeléctrica del Cantábrico Research and Development Department, Asturias, Spain, 1997 (in spanish).
[30] L. Sánchez, Interval-valued GA-P algorithms, IEEE Trans. Evolutionary Comput. 4 (1) (2000) 64–72.
[31] M. Sugeno, T. Yasukawa, A Fuzzy-logic-based approach to qualitative modeling, IEEE Trans. Fuzzy Systems 1 (1) (1993) 7–31.
[32] P. Thrift, Fuzzy logic synthesis with genetic algorithms. Proc. 4th Internat. Conf. on Genetic Algorithms (ICGA'91), Morgan Kauffman Pub., Los Altos, CA, 1991, pp. 509–513.
[33] L.X. Wang, J.M. Mendel, Generating fuzzy rules by learning from examples, IEEE Trans. Systems Man Cybernet. 22 (1992) 1414–1427.
[34] R.R. Yager, On the construction of hierarchical fuzzy systems model, IEEE Trans. Systems Man Cybernet. 28 (1) (1998) 55–66.

[35] J. Yen, L. Wang, C. Wayne Gillespie, Improving the interpretability of TSK fuzzy models by combining global learning and local learning, IEEE Trans. Fuzzy Systems 6 (4) (1998) 530–537.
[36] L.A. Zadeh, Fuzzy sets, Inform. and Control 8 (1965) 338–353.
[37] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning, Inform. Sci. Part I: 8 (1975) 199–249; Part II: 8 (1975) 301–357; Part III: 9 (1975) 43–80.
[38] L.A. Zadeh, Toward a theory of information granulation and its centrality in human reasoning and fuzzy logic, Fuzzy Sets and Systems 90 (1997) 111–127.
[39] I. Zwir, E.H. Ruspini, Qualitative object description: initial reports of the exploration of the frontier, Proc. EUROFUSE-SIC' 99, Budapest, Hungary, 1999, pp. 485–490.