DECSAI

Department of Computer Science
and Artificial Intelligence

# Can Linguistic Modeling Be As Accurate As Fuzzy Modeling Without Losing Its Description To A High Degree?

**Jorge Casillas, Oscar Cordón, Francisco Herrera**

Technical Report #DECSAI-00-01-20
September, 2000

E.T.S. de Ingeniería Informática. Universidad de Granada.
E-18071 Granada, Spain – Phone +34-958-244019, Fax +34-958-243317

# Can Linguistic Modeling Be As Accurate As Fuzzy Modeling Without Losing Its Description To A High Degree? *

Jorge Casillas, Oscar Cordón, Francisco Herrera

*Department of Computer Science and Artificial Intelligence*
*E.T.S. de Ingeniería Informática, Avda. Andalucía 38*
*University of Granada, E-18071 Granada, Spain*

{casillas,ocordon,herrera}@decsai.ugr.es

**Abstract**

In system modeling with Fuzzy Rule-Based Systems (FRBSs), we may usually find two contradictory requirements, the interpretability and the accuracy of the model obtained. As known, Linguistic Modeling (LM)—where the main requirement is the interpretability—is developed by linguistic FRBSs, while Fuzzy Modeling (FM)—where the main requirement is the accuracy—is developed, among others, by approximate FRBSs. Whilst the fact of making linguistic FRBSs be highly interpretable involves establishing hard restrictions to the rule structure (due to the use of a global semantic) thus losing flexibility, relaxing such restrictions, as approximate FRBSs do (using a local semantic), can make more flexible models to be obtained but losing their interpretability.

The main objective of this contribution it to carry out a comparative analysis between LM and FM beyond the classical approach resigned to simply consider LM with a good interpretability but a bad accuracy. Some possibilities to significantly improve the precision of the linguistic models keeping good legibility will be introduced under the assumption that better accuracy could be obtained looking for a good balance between model flexibility and modeling simplicity.

The good performance of such improvements opposite to FM will be shown by means of a wide experimental study applying fourteen different learning methods, carefully selected, to four modeling problems with different nature.

**Keywords:** Linguistic Modeling, Fuzzy Modeling, linguistic and approximate Fuzzy Rule-Based Systems, learning, tuning, linguistic hedges, inconsistent rules, weighted rules

*This research has been supported by CICYT under project PB98-1319

# 1 Introduction

Fuzzy Rule-Based Systems (FRBSs) constitute an extension of classical Rule-Based Systems, because they deal with IF-THEN rules where antecedents and/or consequents are composed of Fuzzy Logic statements, instead of Classical Logic rules. This consideration presents two essential advantages: the key features of knowledge captured by fuzzy sets involve handling uncertainty, and inference methods become more robust and flexible with approximate reasoning methods of Fuzzy Logic.

One of the most usual applications of FRBSs is *system modeling* [3, 27], which in this field may be considered as an approach used to model a system making use of a descriptive language based on Fuzzy Logic with fuzzy predicates [31]. In this kind of modeling we may usually find two contradictory requirements, the *interpretability* and the *accuracy* of the model obtained. As known, traditionally, two main areas have been distinguished in system modeling with FRBSs: *Linguistic Modeling* (LM) and *Fuzzy Modeling* (FM). While the main requirement of the former is to obtain highly interpretable models, the latter is concerned with obtaining highly accurate models (losing descriptive power if necessary).

LM is developed by means of *linguistic* FRBSs (also known as Mamdani-type FRBS), which use fuzzy rules composed of linguistic variables [37] that take values in a term set with a real-world meaning, thus the linguistic model consists of a set of linguistic descriptions regarding the behavior of the system being modeled [31]. Opposite to it, FM is mainly put into effect by means of *approximate* FRBSs [1, 3, 6, 11, 13] or *Takagi-Sugeno-Kang* (TSK) FRBSs [29, 32]. These kinds of FRBSs aim at increasing the system accuracy making more flexible the model structure at the expense of losing interpretability. To do so, the former ones consider systems where the variables forming the rules do not take as a value a linguistic term with a fuzzy set associated but directly a fuzzy set. On the other hand, TSK FRBSs consider a rule structure where the consequent is a non interpretable but flexible linear combination of the inputs. In this contribution, only FM developed by approximate FRBSs will be considered.

Although approximate FRBSs are more flexible than linguistic ones, this does not always entail to obtain more accurate results since this freedom makes more difficult to accomplish the modeling process. Therefore, the main point in system modeling should not be as the capability of the model structure as the capability of generating the model. From this point of view, a trade-off between the freedom of the model and the good performance of the learning process would provide better behavior.

Improvements in the LM making more flexible the learning process and/or the rule structure can significantly increase the accuracy of the obtained model with a slight description loss. Taking into account such considerations, the following question arise:

*Can LM be as accurate as FM without losing its description to a high degree?*

The proposal of this contribution will be to answer this question showing possible improvements of LM and performing a wide experimental study to analyze the behavior of LM, improved LM, and FM facing different situations.

To do so, the paper is organized as follows. In Section 2, linguistic and approximate FRBSs to respectively develop LM and FM will be introduced, and a contrasted analysis of them will be performed. Section 3 will be devoted to introduce some approaches to improve LM increasing the system accuracy, considering the interpretability loss entailed by each of them. In Section 4, an experimental study will be performed applying some specific learning methods to the solving of four different problems. Finally, some interesting conclusions suggested along the paper will be pointed out in Section 5. The learning methods and problems considered in the study will be described in Appendices A and B, respectively.

## 2 Linguistic Modeling vs Fuzzy Modeling

Different FRBS types are used depending on the modeling kind considered. Some of the most usual for LM and FM are respectively linguistic and approximate FRBSs. These will be described in the following subsection. After that, an analysis on both modeling approaches will be made.

### 2.1 FRBSs Used To Develop Linguistic and Fuzzy Modeling

Following, linguistic and approximate FRBSs considered in LM and FM respectively will be introduced.

- LM based on Fuzzy Logic is considered as a system model constituting a linguistic description, being put into effect by means of a linguistic FRBS (also known as Mamdani-type FRBS). Thereby the concept of linguistic variable [37] plays a central role. A crucial reason why the linguistic rule-based approach is worth considering is that it may remain verbally interpretable. Linguistic FRBSs are formed by linguistic rules with the following structure:

  $$\begin{array}{l}\textbf{IF } X_1 \text{ is } A_1 \text{ and } \ldots \text{ and } X_n \text{ is } A_n \\ \textbf{THEN } Y_1 \text{ is } B_1 \text{ and } \ldots \text{ and } Y_m \text{ is } B_m \quad ,\end{array}$$

  with $X_i$ and $Y_j$ being input and output linguistic variables respectively, and with $A_i$ and $B_j$ being linguistic labels with fuzzy sets associated defining their meaning. These linguistic labels will be taken from a global *semantic* defining the set of possible fuzzy sets used for each variable (Figure 1 shows an example of a fuzzy partition with triangular membership functions). This structure provides a natural framework to include expert knowledge in the form of fuzzy rules.

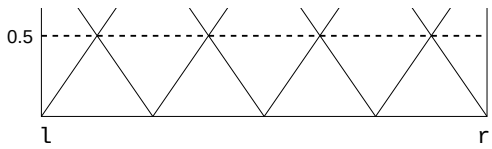



Figure 1: Graphical representation of an example of the semantic considered for a variable, standing S for *small,* M for *medium,* L for *large,* V for *very,* and E for *extremely*, being $[l, r]$ the corresponding variable domain

- On the contrary, FM has the ability of obtaining more flexible models thanks to being able to describe the system behavior without considering a good understanding. In this field, we can mainly found two kinds of FRBSs: the approximate FRBS [1, 3, 6, 11, 13] and the TSK FRBS [29, 32]. In this contribution, we will only focus on FM developed by approximate FRBSs.

  The approximate FRBS differs from the linguistic one in the direct use of fuzzy variables. Each fuzzy rule thus presents its own semantic, i.e., the variables take different fuzzy sets as values and not linguistic terms from a global term set. For this reason, we could respectively name linguistic and approximate FRBSs as *global-semantic* and *local-semantic* FRBSs. Since no global semantic is used in approximate FRBSs, fuzzy sets can not be interpreted. The fuzzy rule structure is as follows:

$$\textbf{IF } X_1 \text{ is } \widehat{A}_1 \text{ and } \ldots \text{ and } X_n \text{ is } \widehat{A}_n$$
$$\textbf{THEN } Y_1 \text{ is } \widehat{B}_1 \text{ and } \ldots \text{ and } Y_m \text{ is } \widehat{B}_m \quad ,$$

with $\widehat{A}_i$ and $\widehat{B}_j$ being fuzzy sets without a direct linguistic interpretation.

## 2.2 Linguistic or Fuzzy Modeling: Interpretability or Flexibility

Figure 2 illustrates the said FRBSs considered in the system modeling field. As shown, FRBS kinds have different *interpretability* and *flexibility* degrees according to the rule structure employed. The fact of making the model obtained by a linguistic FRBS be highly interpretable involves establishing hard restrictions to the rule structure thus losing flexibility [4]. On the contrary, relaxing such restrictions as approximate FRBSs do allows more flexible models to be obtained but losing their interpretability.

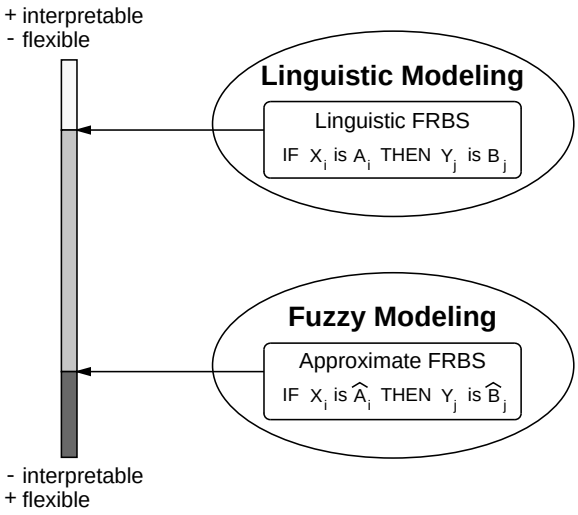



Figure 2: System modeling, interpretability and flexibility depending on the FRBS considered

The flexibility in a modeling process indicates the potential capability of approaching the problem being solved. However, it does not always entail to obtain accurate results since this freedom is a double-edge sword. A greater flexibility give approximation capability to the model obtained but both the complexity to accomplish the modeling process and the risk of overfitting the problem (excessive approximation degree with bad prediction capability) grow.

Therefore, the main point in system modeling should not be as the capability of the model as the capability of generating the model.

> *What good is the flexibility of the model if the learning process has difficulties to profit from it?*

Indeed, greater flexibility does not mean greater accuracy and a trade-off between the freedom of the model structure (*model flexibility*) and the good performance of the learning process (*modeling simplicity*) would provide better behavior (accuracy, see Figure 3). The easier the problem, the closer to LM this trade-off, and, similarly, the more complex the problem, the closer to FM the trade-off. To address this balance, either improvement in LM can be accomplished to make more flexible the learning and/or the model structure, or constrains in the learning process for FM can be made. In fact, the latter possibility has been analyzed in [1] for approximate

FRBSs, verifying that, in practice, best results are obtained when soft constrains (such as variation intervals for the fuzzy sets) are imposed to the learning process.

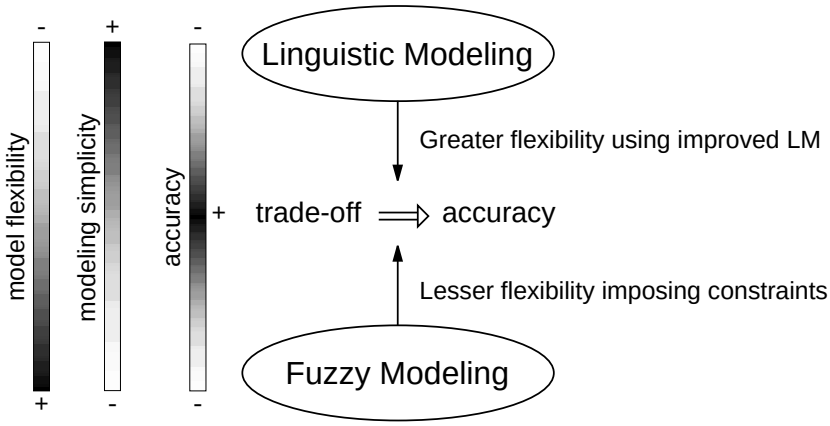



Figure 3: A trade-off between model flexibility and modeling simplicity would provide better accuracy

If the said trade-off is achieved starting from LM, more interpretable models will be obtained than starting from FM. Therefore, it seems to be very interesting to face the system modeling attempting to improve, as much as possible, LM accuracy without losing its description to a high degree. To put this into effect, some improvements performed on LM will be introduced in the following section. After that, experimental studies will be made in Section 4 comparing LM, improved LM, and FM to corroborate such hypotheses.

# 3 Improving Linguistic Modeling

As has been shown in Section 2, LM performed by linguistic FRBSs is very inflexible due to the use of a global semantic. However, it is possible to make some considerations to face this drawback. Basically two ways of improving LM can be considered performing the improvement in

- the *learning process*, giving more freedom to the learning by using a tuning process, or in
- the *rule structure*, slightly changing the rule structure to increase the accuracy.

Usually, all these improvements have the final goal of enhancing the *interpolative reasoning* the FRBS develops. This is one of the most interesting features of FRBSs and plays a key role in their high performance, being a consequence of the cooperative action of the linguistic rules.

In the following subsections, some examples of such improvements will be introduced, discussing later the interpretability of every approach.

## 3.1 Improving Linguistic Modeling Giving More Freedom Degrees By a Tuning Process

Improvements in the linguistic model learning process can be performed allowing it to have more freedom degrees different from the usually considered. In the last few years, many approaches have been presented to automatically learn the Rule Base (RB)—constituted by the collection of linguistic rules themselves joined by means of the connective *also*—from numerical information (input-output data pairs representing the system behavior) [2]. However, there is not much

information about the way to derive the Data Base (DB)—containing the term sets and the membership functions defining their semantics—and most of these RB learning methods need of the existence of a previous definition for it.

A common way to proceed involves considering uniform fuzzy partitions for all the linguistic variables existing in the problem. However, this operation mode makes the DB have a significant influence on the FRBS performance. This is why some proposals try to adjust the membership functions involved in the variable fuzzy partitions, which is known as *tuning process*. A taxonomy of this process can be made from two different angles:

- From the *effect generated in the membership function shapes* point of view (i.e., how to tune such membership functions), the most usual ways are the following:
  - *Changing the basic parameters defining the membership functions* [5, 11, 19, 20, 21, 22, 30, 34]. One of the most common ways of tuning the membership functions is to change the basic parameters defining such functions. For example, if the following triangular-shape membership function is considered:

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

    changing the basic parameters—$a$, $b$, and $c$—will vary the shape of the fuzzy set associated to the membership function (see Figure 4(a)), thus influencing the FRBS performance.
  - *Changing additional membership function parameters* [9, 16]. Sometimes, more flexible alternative expressions for the membership functions are considered to vary the compatibility degree to the fuzzy sets (since it also involves a different rule structure, these new expressions will be wider analyzed in next section). For example, a new membership function can be obtained raising the membership value to the power of $\alpha$, a positive parameter that defines a non-linear scaling function, i.e.,

    $$\mu'(x) = (\mu(x))^{\alpha}, \;\; 0 < \alpha \;\; ,$$

    In this case, the tuning process involves adjusting the additional $\alpha$ parameter to improve the FRBS performance (Figure 4(b) shows the effect of this tuning approach).


Figure 4: Two kinds of tuning for the membership function shapes: (a) changing the basic membership function parameters, (b) changing additional membership function parameters

- From the *operation mode in the modeling process* point of view (i.e., when to perform the tuning process), two main approaches are followed:

– *Embedded tuning*: In this approach the tuning is performed during the derivation of the RB. To do so, the learning process considers the whole Knowledge Base (KB)—the component of the FRBS that represents the knowledge about the problem being solved, which is composed of the RB and the DB—simultaneously optimizing RB and DB (e.g. [20, 21, 22, 34]).

– *A posteriori tuning*: This approach improves the preliminary DB definition considered once the RB have been derived. In this case, a tuning process considering the whole KB obtained by any method (the preliminary DB and the derived RB) is performed a posteriori (e.g. [5, 9, 11, 16, 19, 21, 30]).

Of course, both approaches can be also considered jointly, first developing a learning process with embedded tuning and subsequently performing an a posteriori tuning (e.g. [21]).

Finally, we should say that the tuning task gives more flexibility to the learning process but it runs the risk of overfitting the problem. To face this problem when changing basic membership function parameters, some slight constrains—such as using strong fuzzy partitions [5], using variation intervals for each fuzzy set [11], forcing to keep an established order among vertexes [11, 21], using symmetrical membership functions [19, 21], etc.—can be considered. When additional membership function parameters are adjusted, the tuning is sometimes restricted to a set of values (e.g. $\alpha \in \{0.5, 1, 2\}$) [9, 16] or an interval (e.g. $\alpha \in [0, 5]$), thus reducing furthermore the search space and improving the interpretability of the obtained model (see next section).

## 3.2 Improving Linguistic Modeling Using a Different Rule Structure

Another way to improve the LM performance is to extend the usual linguistic model structure to be more flexible. Three specific possibilities are the following:

### 3.2.1 To use linguistic or generalized hedges

A possibility to relax the model structure allowing it to be more flexible is to include certain operators to slightly change the meaning of the linguistic labels involved in the system when necessary [3, 8, 23].

- A way to do so without losing the description to a high degree is to use *linguistic hedges* (as Zadeh highlighted in [37]). A linguistic hedge (also known as linguistic modifier) is a function that lets us alter the membership functions for the fuzzy sets associated to the linguistic labels, giving a more or less precise definition as a result depending on the case. Two of the most well known modifiers are the *concentration* linguistic hedge "very" and the *dilation* linguistic hedge "more-or-less". Their expressions are

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

  and their effects on a normalized fuzzy set with a triangular membership function are shown in Figure 5.

- A generalization of these linguistic hedges is to raise the membership function to the power of any positive value, not necessarily 2 or 0.5 as the aforementioned modifiers do. In this

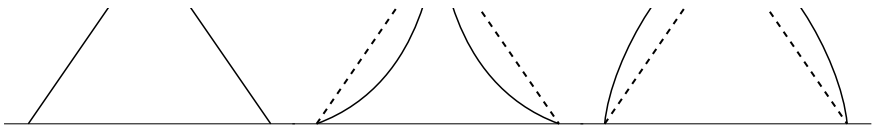


Figure 5: Effects of the linguistic hedges "very" and "more-or-less"

way, still more flexibility is given to the membership function. The general expression of this *generalized hedge* would be therefore the following:

$$\mu_T^{\alpha}(x) = (\mu_T(x))^{\alpha}, \;\; 0 < \alpha \;\; .$$

If ($\alpha > 1$), a concentration effect is performed and, on the contrary, if ($0 < \alpha < 1$), a dilation effect is performed (see Figure 4(b)). However, as opposed to the said linguistic hedges, these actions are now more difficult to be interpreted.

These modifiers, linguistic or generalized hedges, can be basically used in two different ways: different parameters for each of the membership functions composing the DB (*DB level*), or a different parameter for each membership function involved in each linguistic rule (*RB level*). A single parameter for the whole DB is also sometimes considered for generalized hedges [25] (*KB level*). An example of the use of linguistic hedges at RB level would be, for instance, the following rule:

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

### 3.2.2 To use more than one consequent for each rule

This approach involves allowing the RB to present rules where each combination of antecedents may have several consequents associated when it is necessary to improve the model accuracy [11, 12, 25]. It is clear that this will improve the capability of the model to perform the interpolative reasoning and, thus, its performance. The rule structure obtained, considering only an output variable, will be as follows:

**IF** $X_1$ is $A_1$ and ... and $X_n$ is $A_n$
**THEN** $Y$ is $\{B_1,\ldots,B_c\}$ ,

with $c$ being the number of consequents considered for a specific rule.

A particular case is the use of double-consequent rules (i.e., $c = 2$), which has been recently proposed [12, 25]. The use of several consequents has no influence on the linguistic model inference system. The only restriction imposed is that the defuzzification method must consider the matching degree of the rules fired, for example, the *center of gravity weighted by the matching degree* defuzzification strategy [14] may be used.

### 3.2.3 To consider weighted rules giving a certainty factor to each of them

This approach involves using an additional parameter for each rule that indicates its importance degree in the inference process [25, 26, 28, 36], instead of considering all rules equally important as in the usual case. Thus, more flexibility to improve the interpolative reasoning and, therefore, the model performance, is achieved. The rule structure will be the following one:

$$\textbf{IF } X_1 \text{ is } A_1 \text{ and } \ldots \text{ and } X_n \text{ is } A_n$$
$$\textbf{THEN } Y \text{ is } B \textit{ with } [w] \quad ,$$

with $w$ being the real-valued rule weight. In this approach, some changes in the classical inference system must be made to consider the weighted action of each rule.

The operator *with*, which attaches a weight to a rule, may be defined in different ways. One of the most usual options is to multiply the matching degree of the antecedent by the corresponding weight before applying the implication operator, which relates antecedent and consequent. Another possibility is to change the conclusion derived from the implication operator according to the corresponding weight (e.g., changing the support of the obtained fuzzy set).

### 3.3 Interpretability of Improved Linguistic Modeling

Following, the effects in the interpretability degree of the linguistic models obtained by the introduced improvements will be analyzed. Figure 6 shows the different interpretability losses depending on the improvement considered. We should say that the shown order must be considered at large, since a theoretically less interpretable slight action could be better than a theoretically more interpretable hard action. Of course, the order is ambiguous and the only purpose of Figure 6 is to allow the reader to gain an insight into the consequence in the loss of interpretability suffered when applying the different linguistic improvements.

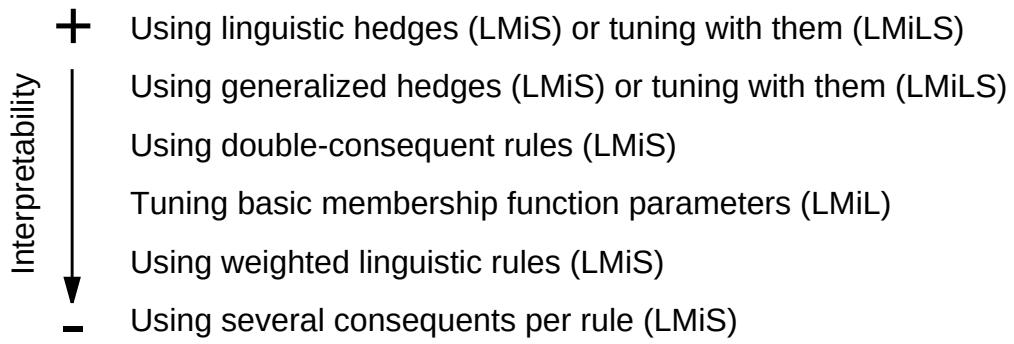



Figure 6: Interpretability loss depending on the linguistic improvement performed—LMiS stands for LM improved in the model structure, LMiL for LM improved in the learning process, and LMiLS for LM improved in both learning process and model structure—

- *Using linguistic hedges or tuning with them*: The use of linguistic hedges seems to be one of the improvements with less interpretability loss. It is due to the fact that the changes performed by the linguistic hedges have a clear meaning that allow this improvement to keep a similar interpretability to the classical approach.

  The interpretability loss will be different depending on the level where the improvement is applied. If the linguistic hedges are used in the RB, the model will be slightly less interpretable than using them in the DB, since in the latter one the changes performed to the membership functions have a global influence and the obtained model seems to be more easily interpreted than local changes in each rule.

  Of course, the interpretability consideration when tuning additional membership functions parameters using linguistic hedges is necessarily tied to the change in the rule structure needed to use these modifiers, thus having similar interpretability degree.

- *Using generalized hedges or tuning with them*: The use of generalized hedges is usually less interpretable than the consideration of linguistic ones, since now there is not a direct linguistic interpretation. However, this improvement also keeps a good description degree since the effect of applying a generalized hedge may be interpreted as a dilation or concentration action on the involved membership functions.

  The said consideration with respect to the interpretability loss depending on the level where the improvement is used can again be made. Moreover, if generalized hedges are used at KB level, the obtained model seems to be more interpretable since the modifier affects to the whole membership function set, and the changes could be considered transparent, only influencing in the FRBSs performance.

  The interpretability degree when tuning additional membership functions parameters using generalized hedges will be evidently similar.

- *Using double-consequent rules*: When using two consequents per rule, the interpretation of the action performed by every rule is confusing to some extent. However, we should note this fact does not constitute an inconsistency from the LM 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 both consequents. Indeed, let us suppose that a specific combination of antecedents, "$X_1$ is $A_1$ and ... and $X_n$ is $A_n$," has two different consequents associated, $B_1$ and $B_2$. The resulting double-consequent rule may be interpreted as follows [12]:

  **IF** $X_1$ is $A_1$ and ... and $X_n$ is $A_n$
  **THEN** $Y$ is *between* $B_1$ and $B_2$ .

- *Tuning basic membership function parameters*: This task may generate an intricate DB that could disturb the expert interpretation, thus losing some legibility degree. Figure 7 shows an example where a hard tuning of the membership function parameters may involve losing interpretability. To face this drawback, some constrains to the tuning process may be imposed (as mentioned in Section 3.1). Of course, these constrains make less flexible the learning process but, moreover of easing the legibility, the risk of overfitting the problem is reduced.

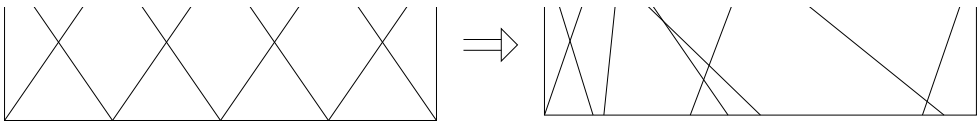


Figure 7: Certain interpretability is lost when a hard tuning is performed in the basic membership function parameters

- *Using weighted linguistic rules*: The interpretability loss with weighted rules lies in the difficulty to interpret the actual action performed by each rule in the interpolative reasoning. Moreover, to use rule weights finally involves the fact of changing the membership functions contained in the corresponding rule and this entails considering linguistic terms with a different meaning in each rule, which severely impairs the system interpretability [24].

- *Using several consequents per rule*: Finally, the interpretability consideration in this case is similar to the mentioned one with double-consequent rules. However, using several

consequents (more than two per rule) makes still more difficult to interpret the behavior of the system, becoming hardly interpretable when many different consequents or equal rules (same antecedent and consequent) are considered. The use of an indiscriminate number of rules without considering consistency criteria only can be justifiable from the accuracy outlook.

# 4 Experimental Comparative Study Among Linguistic, Improved Linguistic, and Fuzzy Modeling

In this section, an experimental comparison among LM, improved LM, and FM will be shown. To do so, several learning methods with different characteristics will be regarded to develop LM, improved LM, and FM making use of linguistic and approximate FRBSs. Table 1 shows the main characteristics of the methods considered (a short description to each of them can be consulted in Appendix A).

Table 1: Summary on the analyzed learning methods and their main characteristics

| Ref. | Method | Modeling Type | Algorithm Type | IL | IS | TBP | TAP | ET | PT | LH | GH | SC | DC | WR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| [35] | WM | LM | AHDD | | | | | | | | | | | |
| [7] | COR | LM | AHDD+SA | | | | | | | | | | | |
| [33] | T | LM | GA | | | | | | | | | | | |
| [11] | M-L-Tun | ILM | GA | ✓ | | ✓ | | | ✓ | | | | | |
| [21] | LMe | ILM | GA | ✓ | | ✓ | | ✓ | ✓ | | | | | |
| [11] | M-L | ILM | AHDD+GA | ✓ | ✓ | ✓ | | | ✓ | | | ✓ | | |
| [9] | LH-Tun | ILM | GA | ✓ | ✓ | | ✓ | | ✓ | ✓RB | | | | |
| [12] | ALM | ILM | AHDD+GA | | ✓ | | | | | | | | ✓s | |
| [25] | NIT | ILM | AHDD | | ✓ | | | | | | ✓KB | | ✓a | ✓ |
| — | WR | ILM | GA | | ✓ | | | | | | | | | ✓ |
| | | | | **Constrains in the learning** (see [1]) | | | | | | | | | | |
| [3] | WCA | FM | AHDD | Hard Constr. Learn./antec., Unconstr. Learn./cons. | | | | | | | | | | |
| [6] | P-FCS1 | FM | GA | Unconstrained Learning | | | | | | | | | | |
| [11] | M-ASCL | FM | GA | Soft Constrained Learning | | | | | | | | | | |
| [13] | M-AHCL | FM | GA | Hard Constrained Learning | | | | | | | | | | |

**ILM** = improved LM, **AHDD** = ad hoc data-driven method, **SA** = simulated annealing-based method, **GA** = genetic algorithm-based method, **IL** = LM improved in the learning, **IS** = LM improved in the model structure, **TBP** = tuning of basic membership function parameters, **TAP** = tuning of additional membership function parameters, **ET** = embedded tuning, **PT** = a posteriori tuning, **LH** = linguistic hedges, **GH** = generalized hedges, **SC** = several consequents per rule, **DC** = double-consequent rules, **WR** = weighted rules, ✓**RB** = linguistic hedges used at RB level, ✓**KB** = a generalized hedge used at KB level, ✓**s** = some rules composing RB have two consequents, ✓**a** = all rules composing RB have two consequents

The behavior of these learning methods will be analyzed when solving of four different applications (two "laboratory" problems and two real-world ones). Table 2 collects the main characteristics of these problems. A brief introduction to them is presented in Appendix B.

As described in Appendix A.2, the M-L-Tun-method and the LH-Tun-method will be considered as a posteriori tuning processes independent of the learning method used to derive a previous KB. Therefore, in the experiments developed, these two tuning methods will be applied to the KBs previously generated by the three LM methods (the WM-method, the COR-method, and the T-method).

Table 2: Summary on the four applications considered and their main characteristics

| Application | Complexity | #V | #Tra | #Tst | #LT | #P |
|---|---|---|---|---|---|---|
| Three-dimensional function $F_1$ | Low | 2 | 674 | 67 | 7 | 1 |
| Three-dimensional function $F_2$ | High | 2 | 1,681 | 168 | 7 | 1 |
| Real-world rice evaluation | Medium | 5 | 75 | 30 | 2 | 10 |
| Real-world electrical distribution | Medium-High | 2 | 396 | 99 | 5/7 | 1 |

**#V** = number of input variables, **#Tra** = training data set size, **#Tst** = test data set size, **#LT** = number of linguistic terms considered for each fuzzy partition, **#P** = number of different partitions of the sample considered

In principle, the application of approximate FRBSs to simple problems as $F_1$ or linguistic FRBSs to complex problems as $F_2$ could seem aberrant, since they are not respectively designed for such cases [1]. However, we have considered appropriate these applications aiming at analyzing the behavior of such systems facing problems with different nature.

### 4.1 Experiments

In the experiments developed, we will consider the *mean square error* (MSE) to evaluate the quality of the results. We have defined MSE as

$$MSE = \frac{1}{2 \cdot N} \sum_{l=1}^{N} (Y^l - y^l)^2 \ ,$$

with $N$ being the data set size, $Y$ being the output obtained from the FRBS, and $y$ being the known desired output. The closer to zero the measure, the greater the model accuracy.

An initial DB constituted by a primary fuzzy partition for each variable will be considered when required. Every partition is formed by a specific number of labels with triangular-shaped equally distributed fuzzy sets giving meaning to them (as shown in Figure 1), and the appropriate scaling factors to translate the generic universe of discourse into the one associated with each problem variable.

As regards the FRBS reasoning method used, we have selected the *minimum t-norm* playing the role of the implication and conjunctive operators, and the *center of gravity weighted by the matching* strategy acting as the defuzzification operator [14].

Table 3 collects the results obtained by the considered methods when solving of the said applications. In that Table, #R stands for the number of rules, and $\text{MSE}_{tra}$ and $\text{MSE}_{tst}$ for the values obtained over the training and test data sets, respectively. In the rice taste evaluation problem, the values shown in these columns have been computed as the average obtained by the ten models generated for each method (see Appendix B.3). In the case of methods with several consequents per rule (the M-L-method, the ALM-method, and the NIT-method), the number of rules shown in the Table corresponds to the rules obtained after decomposing them in simple ones. The best results obtained by each learning method group (LM, improved LM, and FM) for each problem are shown in boldface. An additional column showing the modeling approach followed by each method is also included where LMiL stands for LM improved in the learning process and LMiS stands for LM improved in the model structure.

A global analysis of the obtained results will be made in the following section.

### 4.2 Analysis

From the obtained results, an analysis considering the behavior of the analyzed methods solving the different problems will be made focusing on three different angles: LM vs FM, LM vs im-

Table 3: Results obtained by the analyzed methods in the four problems considered

| | | $F_1$ | | | $F_2$ | | |
|---|---|---|---|---|---|---|---|
| **Modeling** | **Method** | **#R** | **MSE**$_{tra}$ | **MSE**$_{tst}$ | **#R** | **MSE**$_{tra}$ | **MSE**$_{tst}$ |
| LM | WM | 49 | 0.194386 | 0.044466 | 49 | 1.82425 | 2.099754 |
| LM | COR | 49 | 0.097397 | 0.043984 | 49 | **0.502518** | **0.521298** |
| LM | T | **47** | **0.067518** | **0.032441** | **48** | 0.503970 | 0.547687 |
| LMiL | WM + M-L-Tun | 49 | 0.060296 | 0.028621 | 49 | 0.882026 | 1.024698 |
| LMiL | COR + M-L-Tun | 49 | 0.050823 | 0.027383 | 49 | 0.391976 | 0.413784 |
| LMiL | T + M-L-Tun | 49 | 0.030928 | 0.021698 | 49 | 0.403650 | 0.432996 |
| LMiL | LMe | **43** | 0.105432 | 0.174930 | 49 | **0.330614** | **0.370192** |
| LMiLS | M-L | 96 | **0.011865** | **0.014518** | 253 | 0.370366 | 0.375440 |
| LMiLS | WM + LH-Tun | 49 | 0.071854 | 0.026771 | 49 | 1.563712 | 1.799811 |
| LMiLS | COR + LH-Tun | 49 | 0.042854 | 0.021180 | 49 | 0.454830 | 0.456347 |
| LMiLS | T + LH-Tun | 49 | 0.032462 | 0.022374 | 49 | 0.461015 | 0.459301 |
| LMiS | ALM | 55 | 0.019083 | 0.026261 | **32** | 0.548304 | 0.632155 |
| LMiS | NIT | 98 | 0.074073 | 0.015948 | 98 | 0.481266 | 0.519737 |
| LMiS | WR | 49 | 0.016953 | 0.020869 | 49 | 0.482345 | 0.490805 |
| FM | WCA | 49 | 0.288179 | 0.138663 | **49** | 0.494001 | 0.536226 |
| FM | P-FCS1 | **42** | **0.044838** | **0.038075** | 74 | 0.220314 | 0.222865 |
| FM | M-ASCL | 86 | 0.056008 | 0.055712 | 239 | **0.173270** | **0.168424** |
| FM | M-AHCL | 55 | 0.092998 | 0.066488 | 236 | 0.177442 | 0.183578 |

| | | Rice problem | | | Electrical problem | | |
|---|---|---|---|---|---|---|---|
| **Modeling** | **Method** | **#R** | **MSE**$_{tra}$ | **MSE**$_{tst}$ | **#R** | **MSE**$_{tra}$ | **MSE**$_{tst}$ |
| LM | WM | **15** | 0.013284 | 0.013119 | **24** | 222,654 | 239,962 |
| LM | COR | **15** | 0.007979 | 0.008244 | **24** | 182,945 | **170,789** |
| LM | T | 16 | **0.004949** | **0.005991** | 49 | **169,077** | 175,739 |
| LMiL | WM + M-L-Tun | 15 | 0.001106 | 0.002138 | 24 | 145,273 | 171,998 |
| LMiL | COR + M-L-Tun | 15 | 0.001292 | 0.002293 | 24 | 156,651 | **157,513** |
| LMiL | T + M-L-Tun | 16 | 0.001153 | 0.002927 | 49 | 147,844 | 164,993 |
| LMiL | LMe | 31 | **0.000810** | 0.002342 | 49 | 167,014 | 167,383 |
| LMiLS | M-L | 6 | 0.001075 | 0.002331 | 35 | 137,905 | 173,096 |
| LMiLS | WM + LH-Tun | 15 | 0.001758 | 0.001731 | 24 | 181,609 | 209,756 |
| LMiLS | COR + LH-Tun | 15 | 0.001047 | **0.001659** | 24 | 150,323 | 178,923 |
| LMiLS | T + LH-Tun | 16 | 0.000911 | 0.001902 | 49 | **139,013** | 188,945 |
| LMiS | ALM | **5** | 0.003416 | 0.003984 | **20** | 155,866 | 178,601 |
| LMiS | NIT | 64 | 0.002996 | 0.003520 | 64 | 173,230 | 190,808 |
| LMiS | WR | 15 | 0.002588 | 0.003099 | 24 | 149,303 | 182,249 |
| FM | WCA | 32 | 0.017729 | 0.020026 | 34 | 223,985 | 213,906 |
| FM | P-FCS1 | 51 | **0.000596** | 0.009055 | 71 | **141,023** | 197,941 |
| FM | M-ASCL | **4** | 0.003468 | **0.005290** | 20 | 142,109 | **166,579** |
| FM | M-AHCL | 5 | 0.001432 | 0.006094 | 16 | 149,129 | 168,210 |

proved LM, and improved LM vs FM. Considerations from the accuracy and the interpretability points of view will be regarded in each case.

An overall view combining accuracy and interpretability will be also made showing different guidelines on the best choice depending on the difficulty of the problem (as shown in Table 2). Certainly, generalizations from a specific problem can not be made to all problems having a similar complexity and the guidelines shown should be considered for guidance only.

**LM vs FM**

- *Accuracy*: Whilst better precision degrees in $F_1$ are obtained with LM, more accurate models are obtained in FM solving a complex problem as $F_2$, thanks to the use of local semantics that introduce additional degrees of freedom allowing the obtained fuzzy models to be more accurate in these cases. On the other hand, similar accuracy results are obtained by both LM and FM in the rice taste evaluation problem while slightly more accurate models are obtained by FM methods in the electrical problem. However, the approximate FRBSs generated in FM show, in the two real-world problems, a clear tendency to overfit them (an approximation degree significantly greater than the generalization one) because of the high capability of approximation characterizing these kinds of systems that brings the models to be very close to the training data performing then a bad prediction.

  Two methods, the WM-method in LM and the WCA-method in FM, do not generate proper models. In the former case, these results perhaps are related to the fact of generating linguistic rules considering local criteria, without taking into account the global performance of the system [7]. In the latter case, the bad performance developed by the method shows the logical difficulty of designing approximate FRBSs with ad hoc data-driven algorithms.

- *Interpretability*: As regards the legibility of the generated models, the ones obtained by LM methods are significantly more interpretable than the ones obtained in FM, thanks to the use of a global semantic (as mentioned in Section 2.1). Nevertheless, the fuzzy models with only four and five rules (on average for the ten models) obtained by the M-ASCL-method and the M-AHCL-method in the rice problem are interesting results—Table 4(c) shows a Fuzzy Rule Base (FRB) with only two fuzzy rules generated by the M-ASCL-method in this problem—.

- *Guidelines depending on the problem*:

  - *Problems with simple complexity*: In simple problems as $F_1$, LM shows to be clearly better than FM since more accurate models are obtained being, moreoever, more interpretable.
  - *Problems with medium complexity*: All accuracy results being similar between LM and FM in the rice problem, the LM developed by linguistic FRBSs should be clearly preferable to the FM with approximate FRBSs, since the former has, in addition, a high interpretability. Therefore, the models obtained by the linguistic learning T-method are recommended in the rice taste evaluation problem although the fuzzy models generated by the M-ASCL-method and the M-AHCL-method are also good results in spite of the difficulty to understand the fuzzy rule actions (a generated FRB is shown in Table 4(c)).
  - *Problems with medium-high complexity*: In the electrical distribution network problem, the preference between one or another approach will depend on the main requirement to be satisfied: interpretability or accuracy. More accurate models are obtained

Table 4: Some FRBs generated for a specific data set partition of the rice taste evaluation problem—$MSE_{tra}$ and $MSE_{tst}$ values are shown in brackets—

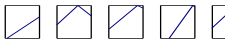
**(a)** WM-method's RB (0.014704 – 0.016700)

| *Rule* | *Flavor* | *Appear.* | *Taste* | *Stickiness* | *Tough.* | *Eval.* |
|---|---|---|---|---|---|---|
| $R_1$ | bad | bad | bad | not-sticky | tender | low |
| $R_2$ | bad | bad | bad | not-sticky | tough | low |
| $R_3$ | bad | good | bad | not-sticky | tender | low |
| $R_4$ | bad | good | good | not-sticky | tender | low |
| $R_5$ | bad | good | good | sticky | tender | low |
| $R_6$ | good | bad | bad | not-sticky | tender | low |
| $R_7$ | good | bad | bad | not-sticky | tough | low |
| $R_8$ | good | bad | good | not-sticky | tender | low |
| $R_9$ | good | good | bad | not-sticky | tough | low |
| $R_{10}$ | good | good | bad | not-sticky | tender | high |
| $R_{11}$ | good | good | bad | sticky | tender | high |
| $R_{12}$ | good | good | good | not-sticky | tender | high |
| $R_{13}$ | good | good | good | not-sticky | tough | high |
| $R_{14}$ | good | good | good | sticky | tender | high |
| $R_{15}$ | good | good | good | sticky | tough | high |

**(b)** ALM-method's RB (0.003838 – 0.002852)

| *Rule* | *Flavor* | *Appear.* | *Taste* | *Stickiness* | *Tough.* | *Eval.* |
|---|---|---|---|---|---|---|
| $R_1$ | bad | bad | bad | not-sticky | tough | low |
| $R_2$ | bad | good | bad | not-sticky | tender | low |
| $R_3$ | good | bad | bad | not-sticky | tender | low |
| $R_4$ | bad | good | good | not-sticky | tender | high |
| $R_5$ | good | good | good | sticky | tender | high |
| $R_6$ | good | good | good | sticky | tough | high |

**(c)** M-ASCL-method's FRB (0.002518 – 0.003290)

| *Rule* | *F* | *A* | *Ta* | *S* | *To* | *E* |
|---|---|---|---|---|---|---|
| $R_1$ |  | | | | | |
| $R_2$ |  | | | | | |

**(d)** WM+LH-Tun-method's RB (0.002230 – 0.000901)

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

by FM methods at expense of losing the capability of interpreting their behavior to a high degree. In Tables 5(a) and 5(d), we may see two models respectively obtained by an LM and an FM method. Indeed, although the model generated by the M-ASCL-method (Table 5(d)) is more accurate than the one obtained by the COR-method (Table 5(a)), the difficulty to understand the former one is significantly higher.

– *Problems with high complexity*: FM is clearly recommended in a complex problem as $F_2$ since it obtains significantly more accurate models.

**LM vs Improved LM**

- *Accuracy*: We may note the great result shown by using different improvements in LM, generally obtaining more accurate linguistic models. Nevertheless, some improved linguistic models present certain overfitting of the problem losing generalization, as the case of the ones generated by the LMe-method, the T+M-L-Tun-method and the T+LH-method in the rice problem, or the ALM-method, the NIT-method and the WR-method in the electrical problem. This worse prediction is one of the risks when excessive flexibility is given to the model and shows the difficulty to overcome LM in some cases.

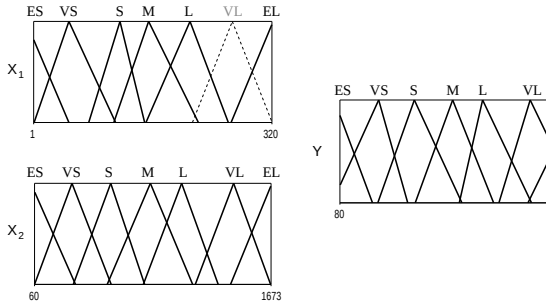
- *Interpretability*: The improvements in LM performed by some learning methods, in spite of obtaining good accuracy degrees, involve the use of a high number of rules in some cases (e.g., the LMe-method in the rice problem, the M-L-method in $F_1$ and $F_2$, or the

Table 5: Some FRBs generated in the electrical application problem—$\text{MSE}_{tra}$ and $\text{MSE}_{tst}$ values are shown in brackets—

**(a)** COR-method's RB (182,945 – 170,789)

| $X_2$ \ $X_1$ | ES | VS | S | M | L | VL | EL |
|---|---|---|---|---|---|---|---|
| ES | ES | VS | | | | | |
| VS | ES | VS | VS | S | | | |
| S | VS | S | S | M | S | | |
| M | VS | M | M | EL | | | S |
| L | | S | M | EL | | | |
| VL | S | L | L | S | | | |
| EL | M | | | | | | |

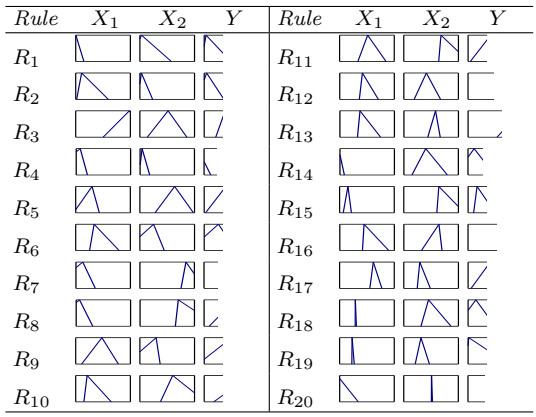
**(b)** COR+M-L-Tun-method's DB (156,651 – 157,513)




**(c)** ALM-method's RB (155,866 – 178,601)

| $X_2$ \ $X_1$ | ES | VS | S | M | L | VL | EL |
|---|---|---|---|---|---|---|---|
| ES | ES | ES | | | | | |
| VS | ES<br>VS | VS | VS | | | | |
| S | VS | VS<br>M | | S | M | | |
| M | VS | S<br>L | | VL | | | S |
| L | | | | | | | |
| VL | VS | L | | M | | | |
| EL | M | | | | | | |

**(d)** M-ASCL-method's FRB (142,109 – 166,579)




NIT-method in the four problems). Moreover of the corresponding interpretability loss derived from such improvements (tuning of basic membership functions parameters, several consequents per rule, weighted rules, etc.), so many rules make the obtained model be still more hardly understandable, which is an important requirement in LM and should be taken into account. Opposite to these cases, some methods achieve to significantly improve the behavior keeping a good interpretability, as mentioned in Section 3.3.

For example, Tables 5(a) and 5(b) respectively show the RB generated by the COR-method and the corresponding DB changed by the M-L-Tun-method in the electrical problem. As notice, using the same RB and practically keeping the original DB (a uniformly generated one)—thus having a similar interpretability—a significantly more accurate model is obtained by the improved LM method.

- *Guidelines depending on the problem*: Keeping in mind the said requirements in terms of accuracy (with good trade-off between approximation and generalization degrees) and interpretability, the following methods may be considered to be the ones obtaining the best solutions in each problem:

  - *Problems with simple complexity*: The T+M-L-Tun-method and the COR+LH-Tun-method obtain good models for $F_1$. Although the M-L-method, the NIT-method, and the WR-method also obtains good accuracy results, their models are difficult to

be interpreted due to the excessive number of rules (in the two former cases) and the consideration of rules with several consequents and/or weights.

- *Problems with medium complexity*: In the rice problem, the COR+LH-Tun-method (with a very good accuracy) and the ALM-method (though it does not have a good accuracy, it have an excellent interpretability where the original semantic is kept and only six simple rules are used) may be considered as the methods that obtain the best solutions. Their corresponding models are respectively shown in Tables 4(b) and 4(d), respectively.
- *Problems with medium-high complexity*: The COR+M-L-Tun-method generates a model with an excellent balance between approximation and prediction degrees in the electrical problem, having moreover a proper interpretability.
- *Problems with high complexity*: The COR+M-L-Tun-method and the LMe-method are clearly the best options in the $F_2$ problem. Although the M-L-method also obtains an accurate model, its interpretability with the large number of rules used is very bad.

As notice, excellent models result of combining simple methods with a subsequent tuning process. This combination present similar or better results than complex rule generation mechanism as the LMe-method and the M-L-method, which also perform an a posteriori tuning, being moreover significantly simpler and quicker.

**Improved LM vs FM**

- *Accuracy*: The models obtained with improved LM methods significantly overcome the ones obtained with FM methods in $F_1$ and the rice problems. In the electrical problem, similar or slightly better results are also obtained by improved LM learning methods. Only in the complex problem $F_2$, FM methods generate clearly more accurate models.
- *Interpretability*: Regarding the facility to understand the behavior of the obtained model, in spite of the slight loss of legibility suffered by improved LM, improved linguistic models are a far more legible than approximate fuzzy models thanks to the consideration of a global semantic (as said in Section 2.1). For example, we can see how the model obtained by the COR+M-L-Tun-method (whose RB and DB are shown in Tables 5(a) and 5(b), respectively) is notably more interpretable than the fuzzy model obtained by the M-ASCL-method (Table 5(d)) in the electrical problem.
- *Guidelines depending on the problem*:
  - *Problems with simple complexity*: The improved LM learning methods are clearly better in terms of accuracy and interpretability in $F_1$.
  - *Problems with medium complexity*: In the rice evaluation taste problem, the improved LM shows to be better than FM since the best prediction degrees are obtained by the former approach, being furthermore interpretable. Nevertheless, the model obtained by the FM M-ASCL-method with only four rules (as average for the ten partitions considered in this problem) is an interesting result, although its accuracy is worse than, for example, the model obtained by the improved LM ALM-method with only five rules. Two models generated by these methods are shown in Tables 4(b) and 4(c). As notice, a slightly more accurate and significantly more interpretable solution is obtained by the improved LM method.
  - *Problems with medium-high complexity*: Taking into account the accuracy and interpretability of the obtained models, the improved LM learning methods would be

preferable to the FM ones in the electrical problem. As may be observed in Table 5(b), the tuning performed to the global semantic over the KB generated by the COR-method (Table 5(a)) makes the obtained model be more accurate and interpretable than the best analyzed fuzzy model (obtained by the M-ASCL-method, Table 5(d)).

– *Problems with high complexity*: In $F_2$ problem, to settle on one or another option will depend on which requirement is mainly considered: while in improved LM satisfactory accuracy with good interpretability is obtained, in FM very good precision is achieved sacrificing the interpretability.

# 5 Concluding Remarks

The main objective of this contribution has been to carry out a comparative analysis between LM and FM beyond the classical approach resigned to simply consider LM with a good interpretability but a bad accuracy. Some possibilities to significantly improve the precision of the linguistic models keeping good legibility have been introduced under the assumption that better accuracy could be obtained looking for a good balance between model flexibility and modeling simplicity. The good performance of such improvements opposite to FM has been shown by means of a wide experimental study where fourteen different learning methods, carefully selected, were applied to four modeling problems with different nature.

The conclusions shown through the paper can be summarized answering the following four questions:

### What possibilities are found in system modeling with FRBSs?

In system modeling with FRBSs, we may usually find two contradictory requirements, the interpretability and the accuracy of the model obtained. As known, LM (where the main requirement is the interpretability) is developed by linguistic FRBSs, while FM (where the main requirement is the accuracy) is developed, among others, by approximate FRBSs. Whilst the fact of making linguistic FRBSs be highly interpretable involves establishing hard restrictions to the rule structure (due to the use of a global semantic) thus losing flexibility, relaxing such restrictions, as approximate FRBSs do (using a local semantic), can make more flexible models to be obtained but losing their interpretability.

### Is flexibility equal to accuracy?

This question may be answered with another one: What good is the flexibility of the model if the learning process has difficulties to profit from it? Indeed, greater flexibility does not involve greater accuracy, since the complexity to accomplish the modeling process grows, and a trade-off between the freedom of the model (model flexibility) and the good performance of the learning process (modeling simplicity) would provide better behavior.

### How can be obtained more accurate models with good interpretability?

To address the balance between model flexibility and modeling simplicity, either

- improvement in the LM can be accomplished to make more flexible the learning and/or the rule structure, or
- constrains in the learning process for the FM can be imposed.

If the said trade-off is achieved starting from LM, more interpretable models will be obtained than starting from FM. Therefore, it seems to be very interesting to face system modeling attempting to improve, as much as possible, LM without losing its description to a high degree. To do so, two different kinds of linguistic improvements have been introduced in this contribution:

- Improvement in the *learning process* by tuning the membership function parameters associated to the linguistic terms.
- Improvement in the *model structure* by using linguistic or generalized hedges, using several consequents per rule, or using weighted rules.

Very interesting results have been obtained in this paper combining simple linguistic learning methods, which do not make use of complex rule generation mechanisms, with a tuning process a posteriori (either adjusting basic membership function parameters or using linguistic hedges). Therefore, it seems to be a good idea to investigate into new tuning processes with good criteria of prediction and interpretation quality.

### Can LM be as accurate as FM without losing its description to a high degree?

In this contribution, we have seen that in spite of LM is generally less accurate than FM, the former can be properly improved making it more flexible. With improved LM, the performance field of LM is extended covering problems with medium or medium-high complexity, where similar or better accuracy than FM is obtained with more interpretable models. Therefore, this leads us to think that, if not always, for the most of cases, properly improved LM can be as accurate as FM without losing its description to a high degree.

# A Learning Methods Considered in the Experimental Study

## A.1 Linguistic Model Learning Methods

Three specific methods to develop linguistic models have been considered:

- In [35], Wang and Mendel propose a simple ad hoc data-driven learning method to generate linguistic rules (**WM-method**). The algorithm consists of dividing the input and output spaces into fuzzy regions, generating the rule best covering each example, and finally selecting among the rules with common antecedent part (conflicting rules) the one with the highest importance degree (computed according to the covering degree) for each fuzzy input region.

- Casillas, Cordón, and Herrera propose in [7] the Cooperative Rules methodology (**COR-method**) to induce good cooperation among linguistic rules benefiting from the interpolative reasoning. To do so, firstly a set of candidate rules is generated in each group (conflicting rules) following some specific learning method (in this contribution, the WM-method will be used). Then, instead of directly selecting the best rule among the conflicting ones, as the previous method does, a combinatorial process (performed by a Simulated Annealing algorithm in this case) is developed to obtain the combination of rules (one per group) with the best global performance, even though they are not the best in the corresponding groups.

- The method proposed by Thrift (**T-method**) [33] is based on a Genetic Algorithm (GA) that encodes all the cells of the complete decision table in the chromosome. In this way, the method establishes a mapping between the set of linguistic terms associated to the output variable and an ordered integer set (containing one more element encoding the null value) representing the allele set. Each of the chromosomes is constituted by joining the partial coding associated to each of the linguistic labels contained in the decision table cells. A gene containing the null allele will represent the absence of the linguistic rule located in the corresponding cell in the RB.

## A.2 Improved Linguistic Model Learning Methods

Seven specific methods to improve LM by different ways will be considered. These methods cover the different approaches presented in Section 3 as Table 1 shows: a method to perform only an a posteriori tuning; an embedded tuning method; a whole method with learning and a posteriori tuning processes where several consequents per rule are allowed; an a posteriori tuning of additional membership function parameters; a method considering the double-consequent rule structure; a method with weighted double-consequent rules and a generalized hedge; and a method to learn weighted rules.

- The a posteriori tuning process introduced in [11] by Cordón and Herrera (**M-L-Tun-method**) is based on a GA where each chromosome forming the genetic population encodes a complete DB definition that will be combined with the existing RB in order to evaluate the individual adaptation. Since each of the triangular membership functions has an associated parametric representation based on a 3-tuple of real values, a primary fuzzy partition can be represented by an array composed by $3 \cdot N_t$ real values, with $N_t$ being the number of terms forming the linguistic variable term set. The DB is encoded into a fixed length real-coded chromosome built by joining the partial representations of each variable fuzzy partition. An interval of performance (defining the intervals of adjustment)

is associated to every gene (membership function parameter) in the chromosome. A fitness function penalizing the lack of completeness and appropriate genetic operators are employed.

- Liska and Melsheimer present in [21] a learning method (**LMe-method**) that performs an embedded tuning process during the learning stage. The KB is represented as a chromosome composed of three substrings: a real-coded one with the membership functions associated to all system variables, each of them being represented by two parameters (center and width); an integer-coded one containing the structure of each rule, i.e., the terms used in each variable, with the possibility of rejecting some variable if the null term is used; and another integer-coded substring with the consequent associated to each rule, also having the possibility of rejecting the rule if the corresponding consequent is null. The genetic operators (crossover and mutation) will act in a different way depending on what substring of the chromosome they are applied on. The performance of each chromosome in a population is evaluated by an exponential ranking technique based on the error function. Once the genetic learning process has finished, the method applies a second phase tuning the membership functions for ensuring the optimum adjustment.

- The method to generate linguistic FRBSs proposed by Cordón and Herrera (**M-L-method**) [11] follows the MOGUL paradigm presented in [10]. The method is composed of three stages: an *iterative RB generation process*, a *genetic simplification process*, and a *genetic tuning process* (this third stage is the aforementioned M-L-Tun-method). In the first stage, the RB is derived rule by rule, selecting the most accurate one at each step of the algorithm. Once this rule is obtained, its covering over the training set examples is taken into account. Those examples covered in a degree higher than a specific value are removed from the training set. Each time the best rule has to be selected in the generation process, the accuracy of the candidates is measured by using a multicriteria fitness function that allows the method to ensure the completeness and consistency of the final KB generated. Since the same antecedent combination can be generated for several rules, the method implicitly makes more flexible the rule structure allowing it to have several consequents associated.

  The second stage, the simplification one (proposed in [18]), combines rules and eliminates redundant rules, selecting the most cooperative set of them. It is based on a binary coded GA (one bit for each rule belonging to the previous RB) where each gene indicates the consideration or not of the corresponding rule to belong to the final RB. Appropriate selection and recombination operators are used. The method uses the same fitness function considered in the M-L-Tun-method.

- The method to tune additional membership function parameters by using linguistic hedges at RB level presented in [9] (**LH-Tun-method**) will be also considered. Following the a posteriori tuning approach, the method involves starting from a previous RB generated by any linguistic learning method. After that, by means of a simple GA-based algorithm, the LH-Tun-method tunes the different membership functions considered in each linguistic rule using the linguistic hedges "very" and "more-or-less". To do so, the coding scheme generates integer-coded chromosomes of length $N_r \cdot (n+1)$ (with $N_r$ being the number of rules and $n$ being the number of input variables). Each gene can take three possible alleles—0, 1, or 2—that respectively indicate the three possibilities for each membership function—to use the "more-ore-less" linguistic hedge, not use linguistic hedge, or to use the "very" linguistic hedge—.

  The initial population is generated by introducing a chromosome that represents the previously obtained RB, i.e., all genes taking the allele 1. The remaining chromosomes are generated at random. The standard two-point crossover is used. The mutation operator

changes the gene to the allele 1 when a gene with alleles 0 or 2 must be mutated, and randomly to 0 or 2 when a gene with allele 1 must be mutated. Finally, the Baker's stochastic universal sampling procedure together with an elitist selection scheme and the MSE fitness function are considered.

- The method proposed by Cordón and Herrera according to the Accurate Linguistic Modeling methodology (**ALM-method**) [12] is based on the following two aspects. Firstly, the model structure is extended allowing a combination of antecedents to have two consequents associated when necessary. Therefore, two rules, the primary and secondary in importance, are obtained in each combination considering a specific generation process. In this contribution, the generation process based on the WM-method will be considered. Then, after decomposing each double-consequent rule into two independent simple ones, the selection process proposed in [18] and previously described in the M-L-method is employed to select a subset of the rules best cooperating.

- The proposal of Nozaki, Ishibuchi, and Tanaka (**NIT-method**) in [25] also employs two consequents in each combination of antecedents to improve the performance. In this case, moreover of using always double-consequent rules, these rules are weighted to consider different importance degrees during the inference process. Each rule is derived obtaining a preliminary rule with the consequent taking a real value (instead a linguistic term) and generating from it the two terms best covering such a value (giving a weight to each of the two obtained rules according to the membership degree). The method also makes use of a generalized hedge at KB level (with the $\alpha$ parameter defined by the designer) to give greater flexibility to the learning process.

- A method to learn weighted rules (**WR-method**) will be also considered in this contribution. Once the RB has been generated by any linguistic learning method (COR-method will be considered in this paper), WR-method performs a search by a GA to find the weight of each rule that obtains the best FRBS performance. The operator *with*, which attaches the weight to the corresponding rule, will be applied multiplying it by the matching degree of the antecedent (as said in Section 3.2).

  Each chromosome is a real-coded string with a length equal to the number of rules. Each gene takes a value in the range $[0, 1]$ that represents the weight considered for the corresponding rule. The initial population is generated by introducing a chromosome where all genes take value 1 (original RB) and with the remaining chromosomes generated at random. The two-point crossover operator is used. The mutation simply consists of changing the selected gene by another value randomly generated in the range $[0, 1]$. The Baker's stochastic universal sampling procedure together with an elitist selection scheme and the MSE fitness function are again considered.

### A.3 Approximate Fuzzy Model Learning Methods

Four learning methods to generate approximate FRBSs will be considered. A wider description of them, among others, can be found in [1], where some dissertations on the different constrains to make less flexible the learning process and their effects are also discussed.

- The Weighted Counting Algorithm (**WCA-method**), proposed by Bárdossy and Duckstein in [3], is based on the principle of firstly generating the antecedents of the rule and then obtaining the consequents taking the examples in the training data set matching the antecedents of the rule to any degree as a base. A previous definition of the antecedent fuzzy set supports for each rule, as well as the number of rules, are required. The algorithm generates the vertex of the triangular membership functions thus determining the

fuzzy set shapes used for the antecedents, and subsequently identifies the corresponding consequents.

- The Pittsburgh-style Fuzzy Classifier System #1 (**P-FCS1-method**), proposed by Carse, Fogarty, and Munro [6], is based on a GA characterized by an interesting coding scheme and crossover operator. The rule representation employed by this method is a set of terms that encode the centers and widths of fuzzy set membership functions over the range of input and output variables using a real coding scheme. A chromosome is a variable length concatenated string of such fuzzy rules. The crossover operator is based on the classical two-point crossover but with an $n$-dimensional consideration (being $n$ the number of input variables). A mutation operator, as well as operators to create and delete rules, are also used. This method does not consider any restriction on the fuzzy set parameters since both the gene pool initialization and the genetic operators freely generate these values, thus performing an unconstrained learning process [1].

- Two different methods proposed by Cordón and Herrera, the MOGUL-based approximate Soft Constrained Learning method (M-ASCL-method) [11] and the Hard Constrained Learning Method (M-AHCL-method) [13], will be considered. These methods follow the MOGUL paradigm [10] mentioned in M-L-method (Section A.2), with the first stage being an *evolutionary generation process* and keeping the same simplification process. The tuning stage, originally proposed in [17], is performed in a different way to the mentioned M-L-Tun-method, since no global semantic is considered in approximate FRBSs and, therefore, each fuzzy set involved in each fuzzy rule is independently tuned.

  In **M-ASCL-method**, the generation process draws inspiration from the previously said M-L-method. Opposite to the previous one, once a rule is obtained, an (1+1)-Evolution Strategy locally tunes such a rule making it get an approximate nature. Variation intervals for each fuzzy set are considered, thus imposing soft constrains to the learning process [1].

- On the contrary, the **M-AHCL-method** performs a different generation process based on a GA with hard constrains considering an interval where each point defining the membership functions (instead the whole fuzzy set) may take value [1]. It is accomplished thanks to the use of special genetic operators that maintain the constrains.

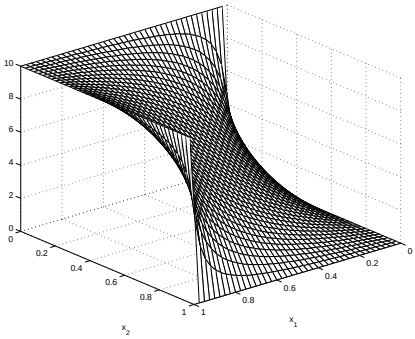
# B Applications Considered in the Experimental Study

The four problems considered in the experimental study will be briefly introduced in the following subsections (Table 2 in Section 4 collects a summary on them).

## B.1 Simple Three-Dimensional Function: $F_1$

The aim in this first problem will be to model the surface of a smooth three-dimensional mathematical function ($F_1$) [11], shown in Figure 8, which presents discontinuities at $(0, 0)$ and $(1, 1)$, as can be observed in its graphical representation. To do so, a training data set uniformly distributed in the three-dimensional definition space has been obtained experimentally. In this way, a set with 674 values has been generated for the function $F_1$ taking 26 values for each one of the two input variables considered to be uniformly distributed in their intervals (it is composed of 674 values instead of 676 because it is not defined in two space points). Another data set has been generated for its use as test set to evaluate the performance of the learning method, avoiding any possible bias related to the data in the training set. The size of this data set (67) is the ten percent of the training set one. The data is obtained generating the input variable values at random in the concrete universes of discourse for each one of them, and computing

the associated output variable value. Seven labels will be used for each variable in those methods that need to know this aspect. A uniform partition with such terms for each variable (see Figure 1) will be considered when a preliminary DB is required.



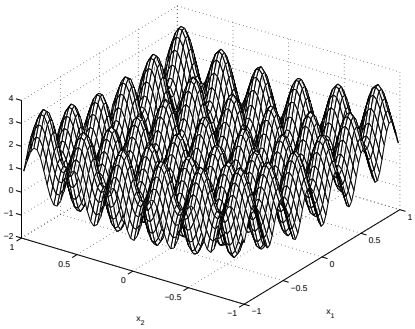

$$F_1(x_1, x_2) = 10 \cdot \frac{x_1 - x_1 x_2}{x_1 - 2x_1 x_2 + x_2}$$

$$x_1, x_2 \in [0, 1], \quad F_1(x_1, x_2) \in [0, 10]$$

Figure 8: Graphical representation, mathematical expression, and variable universes of discourse of the simple three-dimensional function $F_1$

## B.2 Complex Three-Dimensional Function: $F_2$

The *generalized Rastrigin function*, $F_2$, a strongly multimodal one, will be considered as complex three-dimensional function [18]. Its graphical representation, mathematical expression, and variable universes of discourse are shown in Figure 9. The function data is obtained like in the previous case. Training and test sets, with 1,681 (taking 41 values for each of the two input variables) and 168 (ten percent) values, respectively, have been generated. Seven labels will be also considered in this case.




$$F_2(x_1, x_2) = x_1^2 + x_2^2 - cos(18x_1) - cos(18x_2)$$

$$x_1, x_2 \in [-1, 1], \quad F_2(x_1, x_2) \in [-2, 3.5231]$$

Figure 9: Graphical representation, mathematical expression, and variable universes of discourse of the complex three-dimensional function $F_2$

## B.3 The Rice Taste Evaluation Problem

Subjective qualification of food taste is a very important but difficult problem. In the case of the rice taste qualification, it is usually put into effect by means of a subjective evaluation called the *sensory test*. In this test, a group of experts evaluates the rice according to a set of characteristics associated to it. These factors are: *flavor*, *appearance*, *taste*, *stickiness*, and *toughness* [25]. Because of the large quantity of relevant variables, the problem of rice taste analysis becomes very complex, thus leading to solve it by means of modeling techniques capable of obtaining a model representing the non-linear relationships existing in it. In order to do so, we are going to use the data set presented in [25]. This set is composed of 105 data vectors collecting subjective evaluations of the six variables in question (the five mentioned and the

overall evaluation of the kind of rice), made up by experts on this number of kinds of rice grown in Japan.

With the aim of not biasing the learning, we have randomly obtained ten different partitions of the mentioned set, composed by 75 pieces of data in the training set and 30 in the test one, to generate ten models in each experiment. Two labels, or uniform partitions with two linguistic terms, will be considered when required.

### B.4 The Electrical Distribution Network Problem

This problem involves finding a model that relates the total length of low voltage line installed in Spanish rural towns [15]. This model will be used to estimate the total length of line being maintained by an electrical company. We were provided with a sample of 495 towns in which the length of line was actually measured and the company used the model to extrapolate this length over more than 10,000 towns with these properties. We will limit ourselves to the estimation of the *total length of low voltage line installed in a town*, given the inputs *number of inhabitants of the town* and *distance from the center of the town to the three furthest clients*. To develop the different experiments in this contribution, the sample has been randomly divided in two subsets, the training and test ones, with an 80%-20% of the original size respectively. Thus, the training set contains 396 elements, whilst the test one is composed by 99 elements. Seven labels, or the corresponding uniform partitions, will be considered when necessary excepting for the M-AHCL-method and the M-ASCL-method, where five labels will be considered.

## References


[1] R. Alcalá, J. Casillas, O. Cordón, F. Herrera, Approximate Mamdani-type fuzzy rule-based systems: features and taxonomy of learning methods, Technical Report #DECSAI-99-01-17, *Department of Computer Science and Artificial Intelligence*, University of Granada, Granada, Spain, October 1999.

[2] R. Alcalá, J. Casillas, O. Cordón, F. Herrera, S.J.I. Zwir, Learning and tuning fuzzy rule-based systems for linguistic modeling, in C.T. Leondes (Ed.), *Knowledge-Based Systems: Techniques and Applications*, Vol. 3 (Academic Press, San Diego/London, USA/UK, 2000) 889–941.

[3] A. Bárdossy, L. Duckstein, *Fuzzy rule-based modeling with application to geophysical, biological and engineering systems* (CRC Press, Boca Raton, FL, USA, 1995).

[4] A. Bastian, How to handle the flexibility of linguistic variables with applications, *International Journal of Uncertainty, Fuzziness and Knowlegde-Based Systems* **2**:4 (1994) 463–484.

[5] P.P. Bonissone, P.S. Khedkar, Y. Chen, Genetic algorithms for automated tuning of fuzzy controllers: a transportation application, *Proceedings of the 5th IEEE International Conference on Fuzzy Systems*, New Orleans, USA (1996) 674–680.

[6] B. Carse, T.C. Fogarty, A. Munro, Evolving fuzzy rule based controllers using genetic algorithms, *Fuzzy Sets and Systems* **80** (1996) 273–294.

[7] J. Casillas, O. Cordón, F. Herrera, A methodology to improve ad hoc data-driven linguistic rule learning methods by inducing cooperation among rules, Technical Report #DECSAI-00-01-01, *Department of Computer Science and Artificial Intelligence*, University of Granada, Granada, Spain, February 2000.

[8] Z. Chi, H. Yan, T. Pham, *Fuzzy algorithms with application to image processing and pattern recognition* (World Scientific, Singapore, 1996).

[9] O. Cordón, M.J. del Jesus, F. Herrera, Genetic learning of fuzzy rule-based classification systems cooperating with fuzzy reasoning methods, *International Journal of Intelligent Systems* **13** (1998) 1025–1053.

[10] O. Cordón, M.J. del Jesus, F. Herrera, M. Lozano, MOGUL: A methodology to obtain genetic fuzzy rule-based systems under the iterative rule learning, *International Journal of Intelligent Systems* **14**:11 (1999) 1123–1153.

[11] O. Cordón, F. Herrera, A three-stage evolutionary process for learning descriptive and approximate fuzzy logic controller knowledge bases from examples, *International Journal of Approximate Reasoning* **17**:4 (1997) 369–407.

[12] O. Cordón, F. Herrera, A proposal for improving the accuracy of linguistic modeling, *IEEE Transactions on Fuzzy Systems* **8**:3 (2000) 335–344.

[13] O. Cordón, F. Herrera, Hybridizing genetic algorithms with sharing scheme and evolution strategies for designing approximate fuzzy rule-based systems, *Fuzzy Sets and Systems* (to appear).

[14] 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.

[15] O. Cordón, F. Herrera, L. Sánchez, Solving electrical distribution problems using hybrid evolutionary data analysis techniques, *Applied Intelligence* **10** (1999) 5–24.

[16] A. González, R. Pérez, A study about the inclusion of linguistic hedges in a fuzzy rule learning algorithm, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* **7**:3 (1999) 257–266.

[17] F. Herrera, M. Lozano, J.L. Verdegay, Tuning fuzzy controllers by genetic algorithms, *International Journal of Approximate Reasoning* **12** (1995) 299–315.

[18] 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.

[19] C.L. Karr, Genetic algorithms for fuzzy controllers, *AI Expert* **6**:2 (1991) 26–33.

[20] K. KrishnaKumar, A. Satyadas, GA-optimized fuzzy controller for spacecraft attitude control, in J. Periaux, G. Winter, M. Galán, P. Cuenta (Eds.), *Genetic algorithms in engineering and computer science* (John Wiley & Sons, New York, USA, 1995) 305–320.

[21] J. Liska, S.S. Melsheimer, Complete design of fuzzy logic systems using genetic algorithms, *Proceedings of the 3rd IEEE International Conference on Fuzzy Systems*, Orlando, USA (1994) 1377–1382.

[22] L. Magdalena, F. Monasterio-Huelin, A fuzzy logic controller with learning through the evolution of its knowledge base, *International Journal of Approximate Reasoning* **16**:3 (1997) 335–358.

[23] S. Mitra, S.K. Pal, Fuzzy self organization, inferencing, and rule generation, *IEEE Transactions on Systems, Man, and Cybernetics* **26** (1996) 608–619.

[24] D. Nauck, R. Kruse, How the learning of rule weights affects the interpretability of fuzzy systems, *Proceedings of the 7th IEEE International Conference on Fuzzy Systems*, Anchorage, AK, USA (1998) 1235–1240.

[25] 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.

[26] N.R. Pal, K. Pal, Handling of inconsistent rules with an extended model of fuzzy reasoning, *Journal of Intelligent and Fuzzy Systems* **7** (1999) 55–73.

[27] W. Pedrycz (Ed.), *Fuzzy modelling: paradigms and practice* (Kluwer Academic, Dordrecht/Norwell, MA, USA, 1996).

[28] S.P. Stoeva, A weight-learning algorithm for fuzzy production systems with weighting coefficients, *Fuzzy Sets and Systems* **48** (1992) 87–97.

[29] M. Sugeno, G.T. Kang, Structure identification of fuzzy model, *Fuzzy Sets and Systems* **28** (1988) 15–33.

[30] H. Surmann, A. Kanstein, K. Goser, Self-organizing and genetic algorithms for an automatic design of fuzzy control and decision systems, *Proceedings of the 1st European Congresss on Fuzzy and Intelligent Technologies*, Aachen, Germany (1993) 1097–1104.

[31] M. Sugeno, T. Yasukawa, A fuzzy-logic-based approach to qualitative modeling, *IEEE Transactions on Fuzzy Systems* **1**:1 (1993) 7–31.

[32] T. Takagi, M. Sugeno, Fuzzy identification of systems and its application to modeling and control, *IEEE Transactions on Systems, Man, and Cybernetics* **15** (1985) 116–132.

[33] P. Thrift, Fuzzy logic synthesis with genetic algorithms, in R.K. Belew, L.B. Booker (Eds.), *Proceedings of the 4th International Conference on Genetic Algorithms*, San Mateo, CA, USA (Morgan Kaufmann Publishers, 1991) 509–513.

[34] C. Wang, T. Hong, S. Tseng, Integrating fuzzy knowledge by genetic algorithms, *IEEE Transactions on Evolutionary Computation* **2**:4 (1998) 138–149.

[35] L.X. Wang, J.M. Mendel, Generating fuzzy rules by learning from examples, *IEEE Transactions on Systems, Man, and Cyber netics* **22**:6 (1992) 1414–1427.

[36] D.S. Yeung, E.C.C. Tsang, Weighted fuzzy production rules, *Fuzzy Sets and Systems* **88** (1997) 299–313.

[37] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning, Part I: *Information Science* **8** (1975) 199–249, Part II: *Information Science* **8** (1975) 301–357, Part III: *Information Science* **9** (1975) 43–80.