Available online at www.sciencedirect.com

ScienceDirect

International Journal of Approximate Reasoning
44 (2007) 45–64

INTERNATIONAL JOURNAL OF APPROXIMATE REASONING

www.elsevier.com/locate/ijar

Genetic learning of accurate and compact fuzzy rule based systems based on the 2-tuples linguistic representation ☆

Rafael Alcalá [a,*], Jesús Alcalá-Fdez [a], Francisco Herrera [a], José Otero [b]

[a] *Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain*
[b] *Department of Computer Science, University of Oviedo, Campus de Viesques, 33203 Gijón, Spain*

Received 19 July 2005; received in revised form 12 January 2006; accepted 6 February 2006
Available online 24 July 2006

Abstract

One of the problems that focus the research in the linguistic fuzzy modeling area is the trade-off between interpretability and accuracy. To deal with this problem, different approaches can be found in the literature. Recently, a new linguistic rule representation model was presented to perform a genetic lateral tuning of membership functions. It is based on the linguistic 2-tuples representation that allows the lateral displacement of a label considering an unique parameter. This way to work involves a reduction of the search space that eases the derivation of optimal models and therefore, improves the mentioned trade-off.

Based on the 2-tuples rule representation, this work proposes a new method to obtain linguistic fuzzy systems by means of an evolutionary learning of the data base *a priori* (number of labels and lateral displacements) and a simple rule generation method to quickly learn the associated rule base. Since this rule generation method is run from each data base definition generated by the evolutionary algorithm, its selection is an important aspect. In this work, we also propose two new *ad hoc* data-driven rule generation methods, analyzing the influence of them and other rule generation methods in the proposed learning approach. The developed algorithms will be tested considering two different real-world problems.

☆ Supported by the Spanish Ministry of Science and Technology under Projects TIC-2002-04036-C05-01 and TIN-2005-08386-C05-01.

* Corresponding author.
E-mail addresses: alcala@decsai.ugr.es (R. Alcalá), jalcala@decsai.ugr.es (J. Alcalá-Fdez), herrera@decsai.ugr.es (F. Herrera), jotero@lsi.uniovi.es (J. Otero).

0888-613X/$ - see front matter
doi:10.1016/j.ijar.2006.02.007

Keywords: Fuzzy rule-based systems; Linguistic 2-tuples representation; Learning; Interpretability–accuracy trade-off; Genetic algorithms

1. Introduction

One of the problems associated to linguistic fuzzy modeling (FM), modeling of systems building a linguistic model clearly interpretable by human beings, is its lack of accuracy when modeling some complex systems. It is due to the inflexibility of the concept of linguistic variable, which imposes hard restrictions to the fuzzy rule structure [1]. This drawback leads linguistic FM to sometimes move away from the desired trade-off between interpretability and accuracy, thus losing the usefulness of the model finally obtained.

Many different possibilities to improve the accuracy of linguistic FM while preserving its intrinsic interpretability have been considered in the specialized literature [2,3]. These approaches try to induce a better cooperation among the rules by acting on one or two different model components: the data base (DB) – containing the parameters of the linguistic partitions – and the rule base (RB) – containing the set of rules. An efficient way to do that is to obtain the whole knowledge base (KB) – RB and DB – by learning the DB *a priori* [4–10], i.e., considering a process that learns the DB and wraps a simple method to derive a set of rules for each DB definition. Most of the works based on these kinds of learning use genetic algorithms (GAs) for the learning of the DB parameters.

In fact, the automatic definition of fuzzy systems can be considered as an optimization or search process and nowadays, evolutionary algorithms, particularly GAs, are considered as the more known and used global search technique. Moreover, the genetic coding that they use allow them to include prior knowledge to lead the search up. For this reason, evolutionary algorithms have been successfully applied to learn fuzzy systems in the last years, giving way to the appearance of the so called genetic fuzzy systems [11,12].

On the other hand, to ease the genetic optimization of the DB parameters a new linguistic rule representation model was presented in [13]. It is based on the linguistic 2-tuples representation [14] that allows the lateral displacement of a label considering an unique parameter. This way to work involves a reduction of the search space that eases the derivation of optimal models.

In this work, we propose a new method to obtain whole KBs by means of an evolutionary learning of the DB *a priori* that is based on the linguistic 2-tuples rule representation [14]. This method consists of an evolutionary process that learns the optimal number of labels per variable and the lateral displacement of such labels. For each DB definition generated by the evolutionary algorithm, a quick rule generation process is run to obtain the RB. Additionally, in order to improve the generalization ability of the models so obtained we propose a new inference system considering non-covered input examples.

This way to work, makes the selection of the rule generation process become an important aspect. A preliminary study of the proposed technique was presented in [15] considering the Wang and Mendel's (WM) algorithm [16] as a first approach for the rule derivation. To perform a better study, we also propose two new *ad hoc* data-driven rule generation methods in this contribution, analyzing their influence in the proposed KB learning technique. Furthermore, these methods will be analyzed by solving two real-world problems from both, the accuracy and the interpretability point of view.

This contribution is arranged as follows. The next section describes the linguistic rule representation model based on the linguistic 2-tuples and proposes the new inference system. Section 3 introduces the learning scheme considered in this work and proposes the new evolutionary learning algorithm to obtain whole KBs. Section 4 presents two new *ad hoc* data-driven rule generation methods and explains how they can be integrated in the proposed evolutionary algorithm. Section 5 shows an experimental study considering two different real-world problems. Finally, Section 6 points out some concluding remarks.

2. Rule representation based on the linguistic 2-tuples

In [13], a new model of tuning of fuzzy rule-based systems (FRBSs) was proposed considering the linguistic 2-tuples representation scheme introduced in [14], which allows the lateral displacement of the support of a label and maintains a good interpretability associated to the obtained linguistic FRBSs. This tuning proposal also introduces a new model for rule representation based on the concept of symbolic translation [14] (the lateral displacement of a label).

Respect to the classical tuning [11,17–23], usually considering three parameters in the case of triangular membership functions (MFs), this way to work involves a reduction of the search space that eases the derivation of optimal models, preserving the original shape of the MFs.

The following subsections present the concept of symbolic translation, the linguistic 2-tuples rule representation and the new inference system proposed in this work to consider non-covered input examples.

2.1. The symbolic translation of a label

The symbolic translation of a linguistic term is a number within the interval $[-0.5, 0.5)$ that expresses the domain of a label when it is moving between its two lateral labels. Let us consider a set of labels S representing a fuzzy partition. Formally, we have the pair,

$$(s_i, \alpha_i), \quad s_i \in S, \ \alpha_i \in [-0.5, 0.5).$$

Fig. 1 depicts the symbolic translation of a label represented by the pair $(S_2, -0.3)$, considering a set S with five linguistic terms represented by their ordinal values ($\{0, 1, 2, 3, 4\}$).

Actually, the symbolic translation of a label involves the lateral displacement of the MF that represents such label. As an example, Fig. 2 shows the lateral displacement of the label M. The MF of the new label "y_2" is located between S and M, being still closer to M.

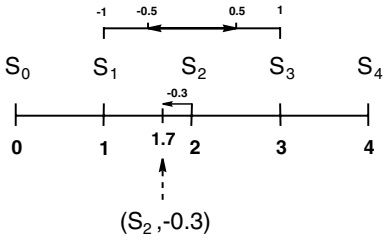


Fig. 1. Symbolic translation of a label.

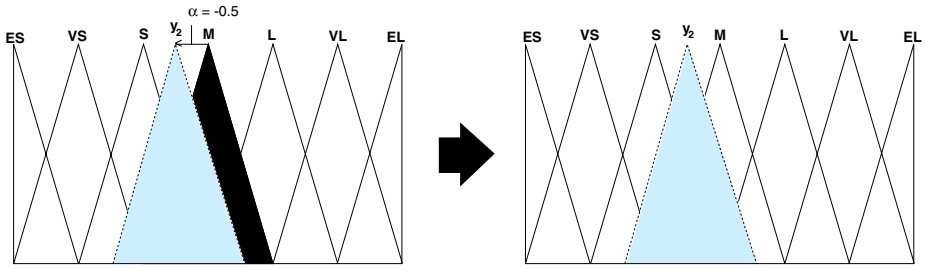


Fig. 2. Lateral displacement of the linguistic label M considering the set of labels $S = \{\text{ES}, \text{VS}, \text{S}, \text{M}, \text{L}, \text{VL}, \text{EL}\}$.

2.2. Rule representation

In [14], both the linguistic 2-tuples representation model and the needed elements for linguistic information comparison and aggregation are presented and applied to the decision making framework. In the context of the FRBSs, we are going to see its use in the linguistic rule representation. In the next we present this approach considering a simple control problem. Let us consider a control problem with two input variables, one output variable and a DB defined from experts determining the MFs for the following labels:

$$\mathit{Error},\ \nabla \mathit{Error} \rightarrow \{N, Z, P\},\ \mathit{Power} \rightarrow \{L, M, H\}.$$

Based on this DB definition, an example of classical rule and linguistic 2-tuples represented rule is:

Classical rule,
If **error** is Zero and ∇**Error** is positive then **Power** is high.
Rule with 2-tuples representation,
If **error** is $(\text{Zero}, 0.3)$ and ∇**Error** is $(\text{Positive}, -0.2)$ then **Power** is $(\text{High}, -0.1)$.

In [13], two different rule representation approaches were proposed, a global approach and a local approach. In our particular case, the learning is applied to the level of linguistic partitions (global approach). In this way, the pair (X_i, label) takes the same α value in all the rules where it is considered, i.e., a global collection of 2-tuples is considered by all the fuzzy rules. For example, X_i is $(\text{High}, 0.3)$ will present the same value for those rules in which the pair "X_i is High" was initially considered.

The main achievement is that, since the three parameters usually considered per label [11,17–23] are reduced to only one symbolic translation parameter, this proposal decreases the learning problem complexity easing indeed the derivation of optimal models. Other important issue is that, the learning of the displacement parameters keeps the original shape of the MFs (in our case triangular and symmetrical). In this way, from the parameters α applied to each label, we could obtain the equivalent triangular MFs, by which a FRBS based on linguistic 2-tuples could be represented as a classical Mamdani FRBS [24,25].

2.3. A new fuzzy inference system

Once the 2-tuples represented model is transformed to its equivalent classical Mamdani FRBS (obtaining the displaced MFs from the learned 2-tuples), a classical fuzzy reasoning

could be considered. In our case, the fuzzy reasoning method is 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 defuzzification operator [26] (FITA scheme).

However, since we are searching for models with the smallest possible number of rules (compact linguistic models) and the support of the final MFs comprising that rules can be displaced, there could be non-covered zones in the input space. Taking into account that the learning algorithm is biased by error measures, this fact should not be a problem (non-covered training data usually provokes high errors in the system and finally they would be covered). However, a good behavior of the obtained model is not ensured for the non-covered test data (i.e., the generalization of the final linguistic model could not be good for uncovered inputs). In this way, to consider non-covered input data for the system output computation, the following mechanism is applied when non-covered points are found:

(1) The nearest rule to the non-covered point is identified (normalized euclidean distance to the vertex of the labels). The non-covered coordinates of the point are set to the value of the vertex of the corresponding label.
(2) The second nearest rule is identified. Then, if the consequent labels of both rules present overlapping to some degree, we only infer with the nearest rule since it will be the most representative in a subspace that does not present strong changes in the output domain.
(3) In other case, the final FRBS output should be obtained by interpolation of both rules, since strong changes are detected in this subspace output domain. To do that, the coordinates of the point that are initially covered are displaced towards the second rule, ensuring a minimum covering degree of the nearest rule (nearing these coordinates to the corresponding label extreme at the 10% of the support size). As an example, let e_i be a coordinate of the non-covered point e that is initially covered by the corresponding label of the nearest rule $\{a_i^{1st}, b_i^{1st}, c_i^{1st}\}$ (left extreme, vertex and right extreme). And let $\{a_i^{2nd}, b_i^{2nd}, c_i^{2nd}\}$ be the definition points of the corresponding label of the second nearest rule. Then, the new value e'_i is computed as follows:

$$e'_i = \begin{cases} a_i^{1st} + (c_i^{1st} - a_i^{1st}) * 0.1, & \text{If } b_i^{2nd} < b_i^{1st}, \\ c_i^{1st} - (c_i^{1st} - a_i^{1st}) * 0.1, & \text{If } b_i^{2nd} > b_i^{1st}, \\ e_i, & \text{If } b_i^{2nd} = b_i^{1st}. \end{cases}$$

(4) Finally, we infer with the new input values considering the whole RB.

3. Evolutionary algorithm for learning of the knowledge base

This section presents the learning scheme and the specific evolutionary algorithm proposed in this work to obtain whole KBs based on the linguistic 2-tuples rule representation.

3.1. KB derivation by learning the DB a priori

As said, an efficient way to generate the whole KB of a FRBS consists of obtaining the DB and the RB separately, based on the DB learning *a priori* [4–10]. This way to work allows us to learn the most adequate context [5,8] for each fuzzy partition, which is

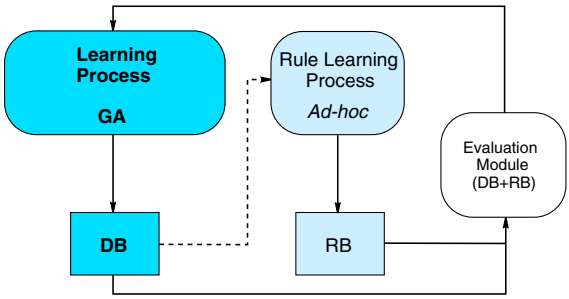


Fig. 3. Learning scheme of the KB.

necessary in different contextual situations (different applications) and for different fuzzy rule extraction models.

Although different optimization techniques could be considered for the learning of the DB parameters *a priori*, in this work, we consider an evolutionary algorithm for this task. In this way, the learning scheme considered for the learning of whole KBs is comprised of two main components (see Fig. 3):

- An evolutionary process to learn the DB, which allows to define:
  - The number of labels for each linguistic variable.
  - The lateral displacements of such labels.
- A quick *ad hoc* data-driven method to derive a RB from each DB definition generated by the evolutionary process. In this way, the cooperative action of both components allows to finally obtain the whole definition of the KB (DB and RB). The simple WM algorithm [16] will be considered for this task as a first approach.

3.2. Evolutionary algorithm (the CHC approach)

Evolutionary algorithms in general and, GAs in particular, has been widely used to derive FRBSs. In this work, we will consider the use of a specific GA to design the proposed learning method, the CHC [27] algorithm. The CHC algorithm is a GA that presents a good trade-off between exploration and exploitation, being a good choice in problems with complex search spaces. This genetic model makes use of a mechanism of "selection of populations". M parents and their corresponding offspring are put together to select the best M individuals to take part of the next population (with M being the population size).

To provoke diversity in the population the CHC approach makes use of an incest prevention mechanism and a restarting approach, instead of the well-known mutation operator. This incest prevention mechanism is considered in order to apply the crossover operator, i.e., two parents are crossed if their distance (considering an adequate metric) divided by two is over a predetermined threshold, L. This threshold value is initialized as the maximum possible distance between two individuals divided by four. Following the original CHC scheme, L is decremented by one when there is no new individuals in the population in one generation. Furthermore, the algorithm restarts the population when L is below zero.

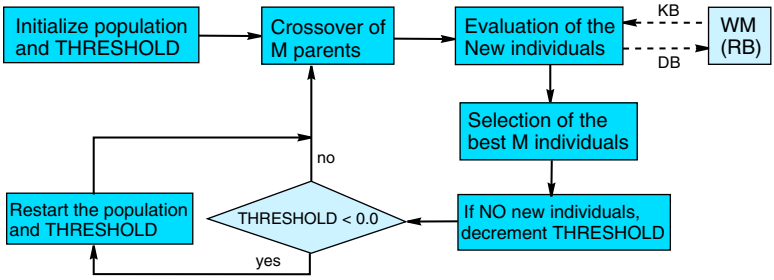


Fig. 4. Scheme of the algorithm considering the CHC approach.

Considering the learning scheme proposed in the previous subsection, the CHC algorithm have to define both, the granularity of the linguistic partitions and the lateral displacements of the involved labels. A global scheme of the proposed algorithm considering the CHC approach is shown in Fig. 4.

In the following, the components needed to design this process are explained. They are: DB codification, chromosome evaluation, initial gene pool, crossover operator (together with the considered incest prevention) and restarting approach.

3.3. DB codification

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

- Number of labels (C_1): This part is a vector of integer numbers with size N (being N the number of system variables). The possible numbers of labels depend on the problem being solved and are established by the system expert for each variable (usually the set $\{3,\ldots,9\}$ for the N variables):

$$C_1 = (L^1,\ldots,L^N).$$

- Lateral displacements (C_2): This part is a vector of real numbers with size $N * 9$ (N variables with a maximum of nine linguistic labels per variable) in which the displacements of the different labels are coded for each variable. Of course, if a chromosome does not have the maximum number of labels in one of the variables, the space reserved for the values of these labels is ignored in the evaluation process. In this way, the C_2 part has the following structure (where each gene is the tuning value of the corresponding label):

$$C_2 = \left(\alpha_1^1,\ldots,\alpha_{L^1}^1,\ldots,\alpha_1^N,\ldots,\alpha_{L^N}^N\right).$$

3.4. Chromosome evaluation

As said, to evaluate a determined chromosome we will apply the well-known rule generation method of Wang and Mendel [16] on the DB coded by such chromosome. To decode this DB, strong fuzzy partitions are defined considering the granularity values of C_1. After that, each MF is displaced to its new position considering the displacement

values of C_2. Once the whole KB is obtained and using the inference system presented in Section 2.3, the mean square error (MSE) is computed and the following function is minimized:

$$F_C = w_1 \cdot \mathrm{MSE} + w_2 \cdot \mathrm{NR},$$

where, NR is the number of rules of the obtained KB (to penalize a large number of rules), $w_1 = 1$ and w_2 is computed from the MSE and the number of rules of the KB generated from a DB considering the maximum number of labels (usually 9) and without considering the displacement parameters,

$$w_2 = \alpha \cdot (\mathrm{MSE}_{\text{max-lab}}/\mathrm{NR}_{\text{max-lab}})$$

with α being a weighting percentage given by the system expert that determines the trade-off between accuracy and complexity. Values higher than 1.0 search for linguistic models with few rules, and values lower than 1.0 search for linguistic models with high accuracy. A good neutral choice is for example 1.0 (good accuracy and not too many rules).

3.5. Initial gene pool

The initial population will be comprised of two different parts (with the same number of chromosomes):

- In the first part, each chromosome has the same random number of labels for all the system variables, setting all the translation parameters to zero.
- In the second part, the only change is that each variable could have a different number of labels.

Since CHC has no mutation operator, the translation parameters remain unchanged and the most promising number of labels is obtained for each linguistic variable. The algorithm works in this way until the first restarting is reached.

3.6. Crossover operator

Two different crossover operators are considered depending on the two parent's scope to obtain two offspring:

- *When the parents encode different granularity levels in any variable*, a crossover point is randomly generated in C_1 and the classical crossover operator is applied on this point in both parts, C_1 and C_2 (exploration).
- *When both parents have the same granularity level per variable*, an operator based on the concept of environments (the offspring are generated around one parent) is applied only on the C_2 part (exploitation). These kinds of operators present a good cooperation when they are introduced within evolutionary models forcing the convergence by pressure on the offspring (as the case of CHC). Particularly, we consider the Parent Centric BLX (PCBLX) operator [28], which is based on the BLX-α. Fig. 5 depicts the behavior of these kinds of operators.

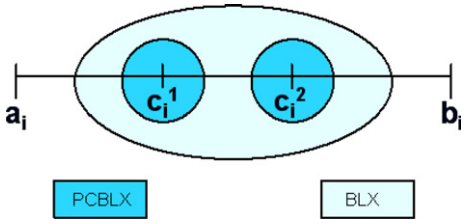


Fig. 5. Scheme of the behavior of the BLX and PCBLX operators.

The PCBLX is described as follows. Let us assume that $X = (x_1 \cdots x_n)$ and $Y = (y_1 \cdots y_n)$, $(x_i, y_i \in [a_i, b_i] \subset \Re,\ i = 1, \ldots, n)$, are two real-coded chromosomes that are going to be crossed. The PCBLX operator generates the two following offspring:

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

On the other hand, the incest prevention mechanism will be only considered in order to apply the PCBLX operator. In our case, two parents are crossed if their hamming distance divided by 2 is over a predetermined threshold, L. Since we consider a real coding scheme (the C_2 part is going to be crossed), we have to transform each gene considering a Gray Code (binary code) with a fixed number of bits per gene (*BITSGENE*), that is determined by the system expert. In this way, the threshold value is initialized as:

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

Following the original CHC scheme, L is decremented by one when there are no new individuals in the next generation. In order to avoid very slow convergence, in our case, L will be also decremented by one when no improvement is achieved respect to the best chromosome of the previous generation.

3.7. Restarting approach

Since no mutation is performed, to get away from local optima a restarting mechanism is considered [27] when the threshold value L is lower than zero. In this case, all the chromosomes set up their C_1 parts to that of the best global solution, being the parameters of their C_2 parts generated at random within the interval $[-0.5, 0.5)$. Moreover, if the best global solution had any change from the last restarting point, this is included in the population (the exploitation only continues while there is convergence). This operation mode was initially proposed by the CHC authors as a possibility to improve the algorithm performance when it is applied to solve some kinds of problems [27].

4. Two new ad hoc data-driven rule generation methods and their integration in the evolutionary learning of the DB *a priori*

As said, the selection of the method considered for rule generation in the learning of the DB *a priori* becomes an important aspect. This method should allow to the learning

process to obtain accurate and, at the same time, compact KBs. Furthermore, since this method is run each time a DB is evaluated, its computation time must be as short as possible. In this section, we discuss about the kinds of methods that could favor this behavior, proposing two new *ad hoc* data-driven methods specifically designed for this task.

We can distinguish between two main possibilities to select this method:

(1) *The first possibility is the use of advanced methods to obtain rules with the best accuracy*. In [29], the authors analyzed different *ad hoc* data-driven methods to propose a new approach called mixed method (MM) that presents a better approximation ability. It is based on the combination of a method guided by examples (the WM [16] algorithm) and a method guided by fuzzy grid (the input space oriented strategy, ISS [30]), and consists of adding rules to the linguistic model obtained by WM in the fuzzy input subspaces that having examples do not still have a rule. Although at first, this approach could seem a good choice, the use of these kinds of advanced methods within our learning approach presents some important drawbacks that should be taken into account. On the one hand, the computational time needed by these methods is higher than that of simpler methods. Moreover, the accuracy improvement obtained by a more sophisticated approach is often achieved by increasing the final number of rules (less interpretable models). On the other hand, some studies [4,5,8] have shown that the system performance is much more sensitive to the learning of the DB than to the composition of the RB. In this way, it is not clear that the derivation of a more elaborated RB favors the learning of better DB definitions respect to other simpler RBs, since the RBs obtained could askew the learning of optimal DBs.

(2) *The second possibility is the use of simpler and faster algorithms that favors the learning of the MFs*. These kinds of methods quickly obtain a small set of basic rules based on the examples with the best covering degree in each fuzzy subspace. Therefore, the quality of the obtained rules directly depends on a successful DB definition to well cover the examples that better represent the system behavior. This way to work leads the DB learning *a priori* to obtain more optimal DBs and simpler RBs, i.e., more accurate and compact KBs. Furthermore, the derivation of simpler models is a way to reduce the overfitting, which eases the derivation of models also presenting a good generalization ability [31]. For these reasons, a basic and simple algorithm as WM performs so well when it is integrated in a method based on the *a priori* learning of the DB [4,5].

Since our main aim is the learning of accurate but also compact FRBSs and the computational time is also an important factor, we will focus our attention on methods fitting with the second possibility, i.e., simple methods that favors and guide the learning of the MFs. An example of these kinds of methods is the WM algorithm, considered in the previous section as a first approach to derive the RB. In the following subsections, we propose two new simple ad hoc data-driven methods that allow the derivation of simpler models maintaining the same or a similar accuracy. They are based on the selection of more general consequents considering a group of the best covered examples and not only the one with the best covering degree. The use of more general consequents also improves the generalization ability of the models so obtained, reducing the effect of noise points.

In any case, for the experiments and with comparative purposes, we will also consider the MM algorithm by directly replacing the WM algorithm in the method proposed in Section 3.

4.1. Rule generation method based on averaged outputs (AV algorithm)

This method tries to obtain more general consequents by means of a weighted average of the output of the examples matching the rule antecedents to a certain degree. The use of an averaged output decrements the influence of noise points. The method is based on the existence of a predefined DB and a set of input–output training data $E = \{e_1, \ldots, e_l, \ldots, e_m\}$ with $e_l = (x_1^l, \ldots, x_{N-1}^l, y^l)$, $l \in \{1, \ldots, m\}$, m being the data set size, and $N - 1$ being the number of input variables. The RB is generated by means of the following steps:

- Initially the RB is empty.
- For each example e_l in E:
  (1) Generate the rule antecedent with the labels best covering the input data $(x_1^l, \ldots, x_{N-1}^l)$.
  (2) If there is not a rule with the same antecedent in the RB:
    (a) Select the examples with a matching degree[1] higher than δ, where $\delta \in [0.5, 1]$ is a value provided by the system expert. If no examples can be selected, select all the examples covered to some degree.
    (b) Calculate the mean of the outputs of the selected examples weighted by their matching degrees, $\overline{M}$.
    (c) Generate the rule consequent with the label best covering $\overline{M}$.
    (d) Add the obtained rule to the RB.

The δ parameter determines how general or specific are the consequents obtained respect to the covered examples. Since it depends on the problem being solved, the granularity and the MFs positions, this parameter should be obtained together with the DB in the evolutionary process. At the end of this section we explain how the proposed methods are included in the evolutionary DB learning *a priori*. This method is a bit slower than the WM algorithm since for each rule antecedent, the matching of all the examples must be computed.

4.2. Rule generation method based on modal consequents (MO algorithm)

This method tries to obtain more general consequents obtaining the modal labels of those proposed by the examples matching the rule antecedents to a certain degree. Since noise points usually appear with a small frequency, these kinds of points would not be considered to compute the output. This method is also based on the existence of a predefined DB and a set E of input–output training data. This algorithm consists of the following steps:

- Initially the RB is empty.
- For each example e_l in E:
  (1) Generate the rule antecedent with the labels best covering the input data $(x_1^l, \ldots, x_{N-1}^l)$.
  (2) If there is not a rule with the same antecedent in the RB:

[1] Using the *minimum t-norm* as conjunctive operator on the obtained antecedent.

(a) Select the examples with a matching degree[1] higher than δ, where $\delta \in [0.5, 1]$ is a value provided by the system expert.
(b) If any example has been selected:
- Calculate the label best covering the output of each selected example, counting the number of times that each output label is obtained.
- Generate the rule consequent with the modal output label, i.e., the output label more times obtained.

Else:
- Generate the rule consequent exactly as WM (that of the rule obtained by the example with the highest covering degree on the N variables).

(c) Add the obtained rule to the RB.

As in the case of the previous method, the δ parameter is obtained together with the DB within the evolutionary process. This method can be implemented exactly as the WM algorithm but counting the frequency of the consequents proposed and finally selecting the modal labels. Therefore, it is faster than the AV algorithm and very similar to the WM algorithm.

4.3. Integration of the proposed methods in the evolutionary learning of KBs

To consider these algorithms within the proposed approach for the DB learning *a priori*, the WM algorithm is directly replaced by the AV or the MO algorithms and the δ parameter should be obtained together with the DB. In this way, the method proposed in Section 3 must include the learning of the δ parameter. In the following, we will only explain the needed changes respect to this algorithm:

- *Coding scheme* – The *coding scheme* is modified by adding the new δ parameter that will be considered to obtain the RB:

$$C = C_1 + C_2 + \delta.$$

- *Initial gene pool* – It works in the same way, but setting the δ parameters at random in $[0.5, 1]$.
- *Crossover* – Considering the crossover operator presented in Section 3, when the C_1 part is crossed, the δ parameter is generated at random in $[0.5, 1]$. When only the C_2 part is crossed, the PCBLX is also applied on the δ parameters.
- *Restarting approach* – As in the original algorithm but setting up the δ parameters at random in $[0.5, 1]$. If the best global solution had any change from the last restarting point, this is included in the population considering the δ part.

The δ parameter is only needed to be considered in the rule generation process, but once the learning process ends and a final 2-tuple represented KB is obtained, this parameter is no more needed.

5. Experimental study

To evaluate the goodness of the proposed algorithms (DB learning *a priori* considering WM, AV or MO algorithms), two real-world electrical energy distribution problems [32] of different complexities are considered:

- *Estimating the length of low voltage lines in rural nuclei*. This problem with only two input variables involves a small search space (small complexity). However, it is still an interesting problem since the system is *strongly nonlinear* and the available data is limited to a low number of examples presenting *noise*. All of these drawbacks make the modeling surface complicated indeed and, in this case, produce a strong overfitting of the obtained models.
- *Estimating the maintenance costs of medium voltage lines in a town*. This problem consists of four input variables and the available data set is comprised of a representative number of well distributed examples. In this case, the learning methods are expected to obtain a considerable number of rules. Therefore, this problem involves *a larger search space* (high complexity).

To correctly solve both problems is a hard task since, in general, methods presenting a good approximation ability do not show a good generalization in real problems (similar to the first problem), since these kinds of methods can easily overfit the obtained models. In this way, the proposed methods present a good approximation ability (specially in the second problem) and at the same time a good generalization ability (specially in the first problem). In the following subsections these problems are introduced and solved to analyze the behavior of the proposed methods.

5.1. Experimental set-up

A brief description of the studied methods is presented in the next three paragraphs (Table 1 summarizes the main characteristics of these methods):

- The proposed methods are named as GLD-WM, GLD-AV and GLD-MO (presented in Sections 3 and 4.3 respectively). The GLD-MM is considered for comparison purposes directly replacing the WM algorithm by the MM method [29] in GLD-WM.
- The WM [16], COR [33] (with Best–Worst Ant System) and MM [29] algorithms are considered as a simple and two advanced rule generation methods to quickly obtain

Table 1
Methods considered for the experimental study

Ref., Year	Method	Type of learning
[16], 1992	WM	AHDD method
[29], 2004	MM	Improved AHDD method based on WM
[33], 2005	COR	Cooperative rules by using the BWAS algorithm
[13], 2004	WM + GL	Global lateral tuning from WM
Methods considering DB learning a priori		
[4], 2001	Gr-MF	Gr. + MF. parameters + RB by WM
[5], 2001	GA-WM	Gr. + Scaling factors + Domains + RB by WM
[8], 2004	GA-COR	Gr. + Scaling factors + Domains + RB by COR
Proposed algorithms		
—	GLD-MM	Gr. + Global lateral parameters + RB by MM
—	GLD-WM	Gr. + Global lateral parameters + RB by WM
—	GLD-AV	Gr. + Global lateral parameters + RB by AV
—	GLD-MO	Gr. + Global lateral parameters + RB by MO

AHDD: *Ad hoc* data-driven – BWAS: Best–Worst Ant System – Gr.: Granularity.

RBs from a predefined DB. We also show the results of the WM + GL tuning method [13] based on the linguistic 2-tuples representation. All of these methods will be considered as a reference since the proposed algorithms are based on some of them. The initial linguistic partitions for these methods are comprised by *five linguistic terms* with uniformly distributed triangular MFs giving meaning to them.

- On the other hand, three methods to obtain a complete KB (DB learning *a priori*) are considered for comparisons. The first one, Gr-WM [4], learns the granularity (number of labels) of the fuzzy partitions and the MFs parameters (their three definition points). GA-WM [5] and GA-COR [8] learn the granularity, scaling factors and the domains (i.e., the variable domain or working range to perform the fuzzy partitioning) for each system variable. These methods respectively obtain the corresponding RB by means of the WM and COR algorithms.

To develop the different experiments we consider a *5-folder cross-validation model*, i.e., five random partitions of data[2] with a 20%, and the combination of four of them (80%) as training and the remaining one as test. For each one of the five data partitions, the studied methods have been run six times, showing for each problem the averaged results of a total of 30 runs. Moreover, a *t-test* (with 95% confidence) was applied in order to ascertain if differences in the performance of the proposed approaches are significant.

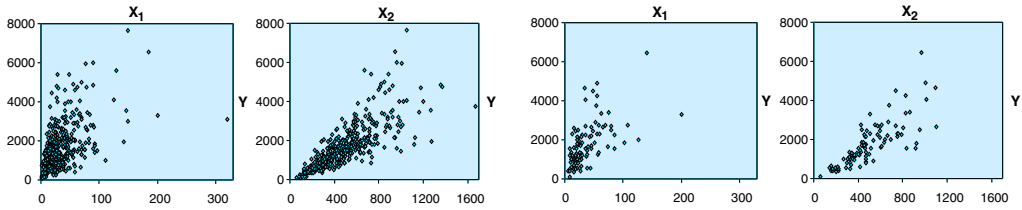
Finally, the following values have been considered for the parameters of each method:[3] 50 individuals, 50,000 evaluations, 30 bits per gene for the Gray codification and the set $\{3,\ldots,9\}$ as possible numbers of labels in all the system variables; 0.6 and 0.2 as crossover and mutation probabilities in the case of the Gr-MF, GA-WM and GA-COR algorithms; since the GA-COR algorithm spends too much time to derive the RB, the authors propose the use of only 2000 evaluations in both problems. The α factor for the fitness function of the GLD methods was set to 1 in both problems. Nevertheless, to obtain models with different levels of accuracy and simplicity, in the second problem (problem with more variables and rules) we also prove with $\alpha = 3$.

5.2. Estimating the length of low voltage lines

This problem consists of relating *the length of the low voltage line of a certain village* (as output variable) with the following two input variables: *the radius of the village* and *the number of users in the village*. A complete description of this problem can be found in [32]. To learn the different system models, we are provided with the measured line length, the number of inhabitants and the mean distance from the center of the town to the three furthest clients in a sample of 495 rural nuclei. Five partitions[2] considering an 80% (396) in training and a 20% (99) in test are considered for the experiments. The existing dependency of the two input variables with the output variable in the training and test data sets of one of the five partitions is shown in Fig. 6 (notice that they present strong non-linearities).

[2] These data sets are available at: http://decsai.ugr.es/~casillas/fmlib/.

[3] With these values we have tried to ease the comparisons selecting standard common parameters that work well in most cases instead of searching very specific values for each method. Moreover, we have set a large number of evaluations in order to allow the compared algorithms to achieve an appropriate convergence. No significant changes were achieved by increasing that number of evaluations.



(a) Training data (b) Test data

Fig. 6. (a) (X_1, Y) and (X_2, Y) dependency in the training data; (b) (X_1, Y) and (X_2, Y) dependency in the test data.

Table 2
Results obtained in the line length estimation problem with parameter $\alpha = 1$ for the fitness function

Method	$\#R$	MSE_{tra}	σ_{tra}	t-test	MSE_{tst}	σ_{tst}	t-test	$h{:}m{:}s$
WM	12.4	234712	32073	+	242147	24473	+	00:00:00.01
MM	19.2	232974	32471	+	244763	23141	+	00:00:00.2
COR	22.0	180995	7794	+	220320	32492	+	00:00:04
WM + GL	12.4	166674	11480	+	189216	14743	+	00:01:03
Gr-MF	21.9	157083	5426	+	242913	59205	+	00:01:31
GA-WM	15.8	160441	6616	+	210444	46773	+	00:01:24
GA-COR	12.6	**152313**	3590	☆	193052	25561	+	02:37:49
GLD-MM	11.2	160374	5020	+	182139	15531	+	00:01:47
GLD-WM	8.8	162295	4059	+	177507	16234	=	00:01:26
GLD-AV	8.7	159689	6324	+	172881	22449	=	00:01:54
GLD-MO	**7.9**	163696	4696	+	**172765**	18895	☆	00:01:25

The results obtained in this problem by the analyzed methods are shown in Table 2, where $\#R$ stands for the number of rules, MSE_{tra} and MSE_{tst} respectively for the averaged error obtained over the training and test data, σ for the standard deviation, $h{:}m{:}s$ for the averaged time of one run in an Intel Centrino (1.73 GHz, 512 MB of RAM) and where *t-test* represents the following information:

☆ represents the best averaged result.
+ means that the best result has better behavior than the one in the corresponding row.
= denotes that the results are statistically equal according to the *t-test*.

Analyzing the results presented in Table 2 we can point out the following conclusions:

- Although the GLD-based methods do not obtain the best training errors, the trade-off between approximation and generalization is pretty good in a problem with noise and poor example data. Taking into account this fact and the high test errors of the remaining methods, we could state that the remaining methods overfits while the GLD-based methods really learns the system behavior. Furthermore, GLD-WM, GLD-AV and GLD-MO obtain the models with the least number of rules.
- Respect to the use of the more advanced MM method, it slightly improves the training error of WM at the cost of adding much more rules. Fortunately, the GLD approach favors the derivation of more simple models, although GLD-MM still presents more

rules and less generalization ability than the remaining GLD-based methods. Therefore, in this problem, the use of this algorithm does not involves an advantage.

- We can see how the computational time of the rule derivation methods affects to the DB learning *a priori*. The most clear case is that of the COR algorithm, multiplying the time of WM per 400 and forcing the GA-COR to spend more than two hours to reach 2000 evaluations. In the following problem, this fact will be even more clear.

Fig. 7 depicts one of the 30 KBs obtained by GLD-MO in this problem. This figure shows how small variations in the MFs lead to important improvements in the behavior of the obtained FRBSs. In this way, the two input variables respectively present three and four labels whose MFs are more or less uniformly distributed, which makes easy to find their corresponding meanings for an expert. The output variable presents five labels that are balanced to the left, representing a higher concentration of examples with small outputs (see Fig. 6). However, since they are again more or less well distributed to the left and to the right of the middle label, we can still easily name these labels.

5.3. Estimating the maintenance costs of medium voltage lines

This problem consists of relating the *maintenance costs of the medium voltage line of a certain town* (as output variable) with the following four input variables: *sum of the lengths of all streets in the town*, *total area of the town*, *area that is occupied by buildings*, and *energy supply to the town*. A complete description of this problem can be found in [32]. In this case, we will deal with estimations of minimum maintenance costs based on a model of the optimal electrical network for a town in a sample of 1059 towns. Five partitions[2] considering an 80% (847) in training and a 20% (212) in test are considered for the experiments.

The results obtained in this problem by the analyzed methods are shown in Table 3 (these kinds of table was described in the previous subsection). Analyzing the results presented in Table 3 we can stress the following facts:

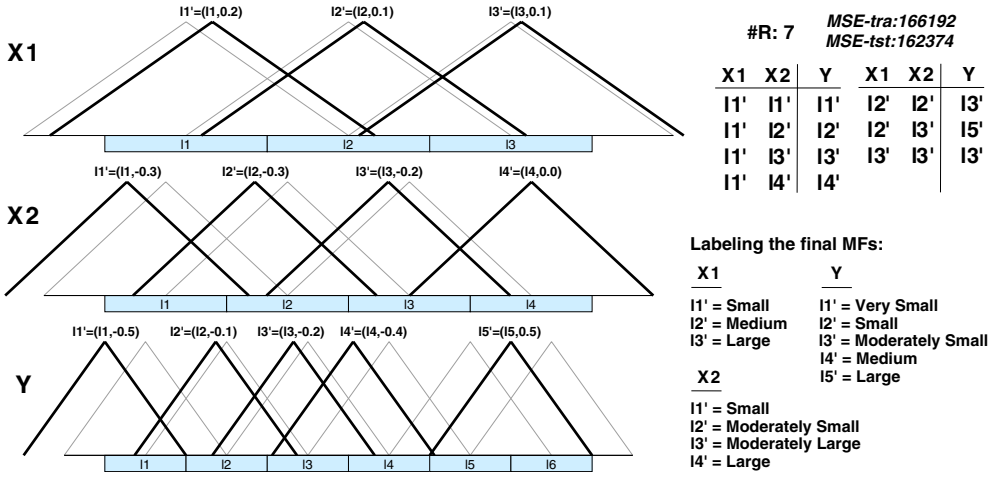
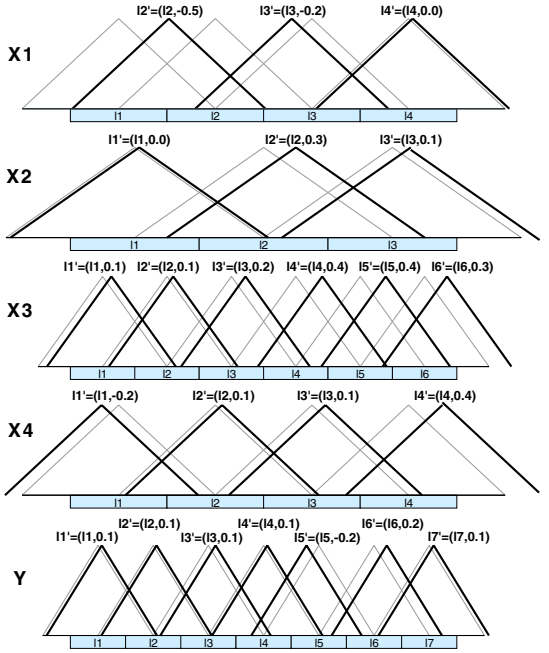


Fig. 7. DB with/without lateral displacements (black/gray), RB and displacements of a model obtained by GLD + MO (the unused labels were removed from this figure).

Table 3
Results obtained in the maintenance costs estimation problem

Method	#*R*	MSE_{tra}	σ_{tra}	*t*-test	MSE_{tst}	σ_{tst}	*t*-test	*h*:*m*:*s*
WM	65	56136	1498	+	56359	4686	+	00:00:00.02
MM	266	44958	1926	+	45598	6553	+	00:00:00.4
COR	41	39640	566	+	41683	1599	+	00:01:00
WM + GL	65	23064	1479	+	25654	2611	+	00:08:15
Gr-MF	93.9	16726	2403	+	18824	3651	+	00:07:53
GA-WM	51.1	23014	2143	+	24090	3667	+	00:10:26
GA-COR	**17.8**	20360	1561	+	22830	3259	+	36:45:41
Proposed methods with parameter $\alpha = 1$ *in the fitness function*								
GLD-MM	261.7	**9617**	1025	☆	**11825**	2168	☆	14:19:02
GLD-WM	57.5	10218	1044	=	12088	1972	=	00:09:10
GLD-AV	60.4	11856	2014	+	13604	2969	+	00:22:06
GLD-MO	54.9	10568	1017	=	12718	2001	=	00:09:20
Proposed methods with parameter $\alpha = 3$ *in the fitness function*								
GLD-MM	154.1	12093	1995	+	14020	2606	+	12:41:02
GLD-WM	41.2	13074	2040	+	15196	2757	+	00:09:23
GLD-AV	36.4	14868	2939	+	16885	4095	+	00:13:32
GLD-MO	34.4	13687	2108	+	16050	2095	+	00:07:55

- In this problem, the drawbacks of the use of more advanced rule derivation methods are even more obvious. In this case, the MM method presents significant improvements in training and test respect to WM at the cost of obtaining an excessive number of rules and increasing the computational time. This makes the GLD-MM method to take more than 14/12 h to obtain a model with more than two/one hundred rules and without significant improvements respect to the use of more simple models. On the other hand, although we do not consider the COR algorithm in our methods, a second analysis could be done about its use for the DB learning *a priori*, GA-COR. The main problem of COR is the long computational time it takes to obtain a RB (approximately 3000 times more than WM), which makes the GA-COR algorithm to take more than one day to reach a total of 2000 evaluations. The main achievement, of this method respect to its homologous, GA-WM, is the derivation of a linguistic model with less number of rules. It is due to the rule simplification performed by COR during the RB learning, which results in linguistic models with too few rules and therefore, with no much better accuracy.
- The GLD-based methods proposed in this work show an important reduction of the mean squared error over the training and test data in a problem with a large search space. It is due to the use of the linguistic 2-tuple representation that reduces the search space respect to the classical learning of MFs, easing the derivation of more optimal models. We must take into account that the Gr-MF method theoretically could obtain at least the same results than GLD-WM, since Gr-MF learns the three definition points of the MFs, being a generalization of GLD-WM.
- GLD-AV and GLD-MO performs so well when we search for simpler models with a similar accuracy to those obtained by GLD-WM or GLD-MM. Furthermore, the linguistic models so obtained are interpretable in a high level since the original shapes of the initial MFs are maintained. In this way, we can highlight the GLD-MO method because of the low number of rules, the errors and the computational times obtained in both problems.



#R: 33 *MSE-tra:12805*
MSE-tst:12381

X1	X2	X3	X4	Y	X1	X2	X3	X4	Y	X1	X2	X3	X4	Y
I2'	I1'	I1'	I1'	I1'	I3'	I2'	I2'	I3'	I4'	I4'	I2'	I2'	I4'	I6'
I2'	I1'	I1'	I2'	I2'	I3'	I2'	I3'	I1'	I3'	I4'	I2'	I3'	I1'	I3'
I2'	I1'	I2'	I1'	I2'	I3'	I2'	I3'	I2'	I4'	I4'	I2'	I3'	I2'	I4'
I2'	I1'	I2'	I2'	I3'	I3'	I2'	I3'	I3'	I5'	I4'	I2'	I3'	I3'	I5'
I3'	I1'	I1'	I1'	I1'	I3'	I2'	I4'	I2'	I5'	I4'	I2'	I3'	I4'	I6'
I3'	I1'	I1'	I2'	I2'	I3'	I2'	I4'	I3'	I6'	I4'	I2'	I4'	I2'	I5'
I3'	I1'	I2'	I1'	I2'	I3'	I2'	I5'	I2'	I6'	I4'	I2'	I4'	I3'	I7'
I3'	I1'	I2'	I2'	I3'	I3'	I2'	I5'	I3'	I7'	I4'	I3'	I4'	I2'	I5'
I3'	I1'	I2'	I3'	I4'	I4'	I2'	I2'	I1'	I3'	I4'	I3'	I4'	I3'	I6'
I3'	I2'	I2'	I1'	I2'	I4'	I2'	I2'	I2'	I3'	I4'	I3'	I6'	I2'	I6'
I3'	I2'	I2'	I2'	I3'	I4'	I2'	I2'	I3'	I4'	I4'	I3'	I6'	I3'	I7'

Labeling the final MFs:

X1
I2' = Small
I3' = Medium
I4' = Large

X2
I1' = Small
I2' = Medium
I3' = Large

X3
I1' = Very Small
I2' = Small
I3' = Moderately Small
I4' = Moderately Large
I5' = Large
I6' = Very Large

X4
I1' = Small
I2' = Moderately Small
I3' = Moderately Large
I4' = Large

Y
I1' = Very Small
I2' = Small
I3' = Moderately Small
I4' = Medium
I5' = Moderately Large
I6' = Large
I7' = Very Large

Fig. 8. DB with/without lateral displacements (black/gray), RB and displacements of a model obtained by GLD + MO (the unused labels were removed from this figure).

Fig. 8 presents the KB obtained by GLD-MO from one of the 30 runs performed in this problem with $\alpha = 3$. Analyzing this linguistic model, we can observe a similar DB configuration to that obtained in the previous problem. The MFs are more or less well distributed which allows us to easily give a meaning to the corresponding labels.

6. Conclusions

This work presents a new method for learning KBs by means of an *a priori* evolutionary learning of the DB (granularity and translation parameters) that uses the linguistic 2-tuples rule representation model and a new inference system. Furthermore, two new *ad hoc* data-driven rule generation methods have been proposed to analyze the influence of them and other rule generation methods in the proposed learning approach. In the following, we present our conclusions and further works:

- The used learning scheme together with the 2-tuples rule representation model and the new inference system allows an important reduction of the search space that eases the derivation of more precise and compact linguistic models.
- The use of simple rule derivation methods searching for basic rules better covering the example data, favors the learning of a better DB and the derivation of RBs with a smaller number of rules. Since the DB learning has more influence in the system behavior than the RB composition, these kinds of methods also eases the derivation of more precise and compact models.

- Moreover, since a global approach is considered and the shapes of the initial MFs are preserved, the interpretability of the obtained models is maintained to a high level respect to the classical learning of fuzzy systems.

The use of different α values to penalize the number of rules in the second problem has demonstrated the existence of optimal models with different levels of accuracy and simplicity. An interesting further work could be the use of multiobjective genetic algorithms to obtain the pareto front with these solutions. In this way, we could easily select a solution with the desired accuracy–interpretability trade-off considering two main objectives, the training error and the number of rules.

References

[1] A. Bastian, How to handle the flexibility of linguistic variables with applications, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 3 (4) (1994) 463–484.
[2] R. Alcalá, J. Alcalá-Fdez, J. Casillas, O. Cordón, F. Herrera, Hybrid learning models to get the interpretability–accuracy trade-off in fuzzy modeling, Soft Computing 10 (9) (2006) 717–734.
[3] J. Casillas, O. Cordón, F. Herrera, L. Magdalena (Eds.), Accuracy Improvements in Linguistic Fuzzy Modeling, Springer-Verlag, 2003.
[4] O. Cordón, F. Herrera, P. Villar, Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base, IEEE Trans. Fuzzy Syst. 9 (4) (2001) 667–674.
[5] O. Cordón, F. Herrera, L. Magdalena, P. Villar, A genetic learning process for the scaling factors, granularity and contexts of the fuzzy rule-based system data base, Information Sciences 136 (2001) 85–107.
[6] B. Filipic, D. Juricic, A genetic algorithm to support learning fuzzy control rules from examples, in: F. Herrera, J.L. Verdegay (Eds.), Genetic Algorithms and Soft Computing, Physica-Verlag, 1996, pp. 403–418.
[7] D. Simon, Sum normal optimization of fuzzy membership functions, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10 (4) (2002) 363–384.
[8] J. Casillas, O. Cordón, F. Herrera, P. Villar, A hybrid learning process for the knowledge base of a fuzzy rule-based system, in: Proceedings of the 2004 International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Perugia, Italy, 2004, vol. 3, pp. 2189–2196.
[9] W. Pedrycz, Associations and rules in data mining: a link analysis, International Journal of Intelligent Systems 19 (7) (2004) 653–670.
[10] Y. Teng, W. Wang, Constructing a user-friendly ga-based fuzzy system directly from numerical data, IEEE Transactions on Systems, Man, and Cybernetics B 34 (5) (2004) 2060–2070.
[11] O. Cordón, F. Herrera, F. Hoffmann, L. Magdalena, GENETIC FUZZY SYSTEMS. Evolutionary tuning and learning of fuzzy knowledge bases, Advances in Fuzzy Systems – Applications and Theory, vol. 19, World Scientific, 2001.
[12] O. Cordón, F. Gomide, F. Herrera, F. Hoffmann, L. Magdalena, Ten years of genetic fuzzy systems: current framework and new trends, Fuzzy Sets and Systems 41 (1) (2004) 5–31.
[13] R. Alcalá, F. Herrera, Genetic tuning on fuzzy systems based on the linguistic 2-tuples representation, in: Proceedings of the 2004 IEEE International Conference on Fuzzy Systems, Budapest, Hungary, 2004, vol. 1, pp. 233–238.
[14] F. Herrera, L. Martínez, A 2-tuple fuzzy linguistic representation model for computing with words, IEEE Transactions on Fuzzy Systems 8 (6) (2000) 746–752.
[15] R. Alcalá, J. Alcalá-Fdez, F. Herrera, J. Otero, A new genetic fuzzy system based on linguistic 2-tuples to learn knowledge bases, in: Proceedings of the First International Workshop on Genetic Fuzzy Systems (GFS 2005), Granada, Spain, 2005, pp. 107–112.
[16] L. Wang, J. Mendel, Generating fuzzy rules by learning from examples, IEEE Transactions on Systems, Man, and Cybernetics 22 (6) (1992) 1414–1427.
[17] O. Cordón, F. Herrera, A three-stage evolutionary process for learning descriptive and approximative fuzzy logic controller knowledge bases from examples, International Journal of Approximate Reasoning 17 (4) (1997) 369–407.

[18] R. Babuška, J. Oosterhoff, A. Oudshoorn, P.M. Bruijn, Fuzzy self-tuning PI control of pH in fermentation, Engineering Applications of Artificial Intelligence 15 (1) (2002) 3–15.
[19] P.P. Bonissone, P.S. Khedar, Y.-T. Chen, Genetic algorithms for automated tuning of fuzzy controllers, a transportation aplication, in: Proceedings of the Fifth IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'96), Nueva Orleans, LA, EE. UU., 1996, pp. 674–680.
[20] F. Herrera, M. Lozano, J.L. Verdegay, Tuning fuzzy logic controllers by genetic algorithms, International Journal of Approximate Reasoning 12 (1995) 299–315.
[21] J.S.R. Jang, ANFIS: adaptive network based fuzzy inference system, IEEE Transactions on Systems, Man, and Cybernetics 23 (3) (1993) 665–684.
[22] C. Karr, Genetic algorithms for fuzzy controllers, AI Expert 6 (2) (1991) 26–33.
[23] L. Zheng, A practical guide to tune proportional and integral (pi) like fuzzy controllers, in: Proceedings of the First IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'92), San Diego, 1992, pp. 633–640.
[24] E. Mamdani, Application of fuzzy algorithms for control of simple dynamic plant, in: Proceedings of the IEEE, vol. 121, 1974, pp. 1585–1588.
[25] E. Mamdani, S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, International Journal of Man–Machine Studies 7 (1975) 1–13.
[26] O. Cordón, F. Herrera, A. Peregrín, Applicability of the fuzzy operators in the design of fuzzy logic controllers, Fuzzy Sets and Systems 86 (1) (1997) 15–41.
[27] L. Eshelman, The chc adaptive search algorithm: how to have safe search when engaging in nontraditional genetic recombination, in: G. Rawlin (Ed.), Foundations of Genetic Algorithms, vol. 1, Morgan Kaufman, 1991, pp. 265–283.
[28] F. Herrera, M. Lozano, A. Sánchez, A taxonomy for the crossover operator for real-coded genetic algorithms: an experimental study, International Journal of Intelligent Systems 18 (2003) 309–338.
[29] P. Carmona, J. Castro, J. Zurita, Strategies to identify fuzzy rules directly from certainty degrees: a comparison and a proposal, IEEE Transactions on Fuzzy Systems 12 (5) (2004) 631–640.
[30] K. Nozaki, H. Ishibuchi, H. Tanaka, A simple but powerful heuristic method for generating fuzzy rules from numerical data, Fuzzy Sets and Systems 86 (3) (1997) 251–270.
[31] H. Ishibuchi, T. Yamamoto, Effects of three-objective genetic rule selection on the generalization ability of fuzzy rule-based systems, in: Proceedings of the Second International Conference on Evolutionary Multi-Criterion Optimization (EMO'03), LNAI, 2632, Springer-Verlag, Faro, Portugal, 2003, pp. 608–622.
[32] O. Cordón, F. Herrera, L. Sánchez, Solving electrical distribution problems using hybrid evolutionary data analysis techniques, Applied Intelligence 10 (1999) 5–24.
[33] J. Casillas, O. Cordón, I.F. de Viana, F. Herrera, Learning cooperative linguistic fuzzy rules using the best–worst ant systems algorithm, International Journal of Intelligent Systems 20 (2005) 433–452.