COR Methodology: A Simple Way to Obtain Linguistic Fuzzy Models with Good Interpretability and Accuracy

Jorge Casillas, Oscar Cordon, and Francisco Herrera

Department of Computer Science and Artificial Intelligence,
Computer Engineering School, University of Granada,
E-18071 Granada, Spain
e-mail: {casillas,ocordon,herrera}@decsai.ugr.es

Abstract. The chapter introduces a simple learning methodology, the cooperative rules (COR) one, that improves the accuracy of linguistic fuzzy models preserving the highest interpretability. Its operation mode involves a combinatorial search of fuzzy rules performed over a set of previously generated candidate ones.

The accuracy is achieved by developing a smart search space reduction and by inducing the generation of a linguistic fuzzy rule set with good cooperation. COR also ensures a good interpretability by keeping the membership functions and the model structure unaltered, as well as generating a compact rule base.

1 Introduction

Fuzzy modeling, i.e., system modeling with fuzzy rule-based systems (FRBSs), may be considered as an approach used to model a system making use of a descriptive language based on fuzzy logic with fuzzy predicates. In this framework, one of the most important areas is *linguistic* fuzzy modeling, where the interpretability of the obtained model is the main requirement. This task is usually developed by means of linguistic (or Mamdani-type) FRBSs, which use fuzzy rules composed of linguistic variables [29] taking values in a term set with a real-world meaning. Thus, the linguistic fuzzy model consists of a set of linguistic descriptions regarding the behavior of the system being modeled [26].

An interpretable model has no sense if it does not faithfully represent the modeled system, i.e., if it is not accurate enough. Thus, a good trade-off between interpretability and accuracy is needed to perform a useful fuzzy modeling. When the main objective is the interpretability, this trade-off can be attained by using different tools to improve the accuracy.

This approach has been performed by learning/tuning the membership functions by defining their shapes [15,20,18], their types (triangular, trapezoidal, etc.) [25], or their context (defining the whole semantic) [23], learning the granularity (number of linguistic terms) of the fuzzy partitions [11], or extending the model structure by using linguistic modifiers [5,12], weights

(importance factors for each rule) [22], or hierarchical architectures (mixing rules with different granularities) [14], among others.

However, all these techniques change the membership functions or extend the original model structure, so that certain interpretability degree is lost. Opposite to it, our aim in this contribution will be to attain the desired balance by increasing the accuracy of linguistic FRBSs keeping the highest interpretability. To do so, a learning procedure that exclusively designs the fuzzy rule set will be introduced: the cooperative rules (COR) methodology. It is based on a combinatorial search of cooperative rules performed over a set of previously generated candidate rule consequents. Additionally, a rule base reduction is developed to improve the interpretability and the accuracy by obtaining compact models without redundancies and inconsistencies.

The chapter is organized as follows. Section 2 describes the COR methodology, Sect. 3 analyzes the behavior and main characteristics of the proposal, Sect. 4 introduces a specific method based on the COR methodology, Sect. 5 performs an experimental study applying our method and other ones to two real-world problems, and, finally, Sect. 6 points out some conclusions.

2 The COR Methodology

A family of efficient and simple methods to derive fuzzy rules guided by covering criteria of the data in the example set, called *ad hoc data-driven methods*, has been proposed in the literature in the last few years [3]. Their high performance, in addition to their quickness and easy understanding, make them very suitable for learning tasks.

However, ad hoc data-driven methods usually look for the fuzzy rules with the best individual performance (e.g. [28]) and therefore the global interaction among the rules of the RB is not considered. This sometimes causes KBs with bad cooperation among the rules to be obtained, thus not being as accurate as desired.

With the aim of addressing these drawbacks keeping the interesting advantages of ad hoc data-driven methods, a new methodology to improve the accuracy obtaining better cooperation among the rules is proposed in [3]: the COR methodology. Instead of selecting the consequent with the highest performance in each subspace like ad hoc data-driven methods usually do, the COR methodology considers the possibility of using another consequent, different from the best one, when it allows the FRBS to be more accurate thanks to having a KB with better cooperation.

COR consists of two stages:

1. *Search space construction* — It obtains a set of candidate consequents for each rule.
2. *Selection of the most cooperative fuzzy rule set* — It performs a combinatorial search among these sets looking for the combination of consequents with the best global accuracy.

A wider description of the COR-based rule generation process is shown in Fig. 1, whilst an example of the operation mode for a simple problem with two input variables and three labels for each linguistic variable is graphically illustrated in Fig. 2.

Since the search space tackled in step 2. is usually large, it is necessary to use approximate search techniques. In [3], accurate linguistic models have been obtained using simulated annealing. Other techniques such as tabu search, genetic algorithms, and ant colony optimization (ACO) have been also applied to COR methodology [4].

3 Analysis of COR Methodology

This section analyzes the behavior and main characteristics of the COR methodology.

3.1 Search Space Reduction

The COR methodology reduces the search space basing on heuristic information. This fact differences COR from other rule base learning methods [27] and allows it to be quicker and to make a better solution exploration.

This search space reduction is performed by two constrains:

1. *Maximum Number of Fuzzy Input Subspaces* — The maximum number of fuzzy input subspaces, and therefore maximum number of fuzzy rules, is limited by the positive example sets. Depending on the approach followed (step 1.1.1. or 1.1.2. in Fig. 1), the selection of the subspaces will be more conservative or more generic.
 Step 1.1.1. divides the input space with a crisp grid bounded by the cross points between labels and, therefore, each example contributes to generate a single rule. On the contrary, step 1.1.2. divides the input space with a fuzzy grid where each example may contribute to generate several rules. Figure 3 graphically shows this fact. In Fig. 3(b), the examples lying in white zones have an influence on the generation of one rule, those lying in light grey zones influence two rules, and the ones lying in dark grey zones influence four rules.
 It is not possible to determine which approach is the best. The crisp grid approach always obtains an equal number or fewer rules than the fuzzy grid one – as in the fuzzy grid approach the examples have an influence on a wider region, thus generating more rules. However, this fact may make the model obtained with the crisp grid approach not to be as accurate as desired sometimes.
2. *Candidate Rule Set in each Subspace* — Once the fuzzy input subspaces are defined, a second search space reduction is made by constraining the set of possible consequents for each antecedent combination, i.e., the candidate rules in each subspace.

Inputs:

- An input-output data set – $E = \{e_1, \dots, e_l, \dots, e_N\}$, with $e_l = (x_1^l, \dots, x_n^l, y^l)$, $l \in \{1, \dots, N\}$, N being the data set size, and n being the number of input variables – representing the behavior of the problem being solved.
- A fuzzy partition of the variable spaces. In our case, uniformly distributed fuzzy sets are regarded. Let $\mathcal{A}_i$ be the set of linguistic terms of the i -th input variable, with $i \in \{1, \dots, n\}$, and $\mathcal{B}$ be the set of linguistic terms of the output variable, with $|\mathcal{A}_i|$ ($|\mathcal{B}|$) being the number of labels of the i -th input (output) variable.

Algorithm:

1. Search space construction:

- 1.1. Define the fuzzy input subspaces containing positive examples: To do so, we should define the positive example set ($E^+(S_s)$) for each fuzzy input subspace $S_s = (A_1^s, \dots, A_i^s, \dots, A_n^s)$, with $A_i^s \in \mathcal{A}_i$ being a label, $s \in \{1, \dots, N_S\}$, and $N_S = \prod_{i=1}^n |\mathcal{A}_i|$ being the number of fuzzy input subspaces. Two possibilities could be the following:

$$1.1.1. E^+(S_s) = \{e_l \in E \mid \forall i \in \{1, \dots, n\}, \forall A_i^s \in \mathcal{A}_i, \mu_{A_i^s}(x_i^l) \geq \mu_{A_i^s}(x_i^l)\}$$

$$1.1.2. E^+(S_s) = \{e_l \in E \mid \mu_{A_1^s}(x_1^l) \cdot \dots \cdot \mu_{A_n^s}(x_n^l) \neq 0\},$$

with $\mu_{A_i^s}(\cdot)$ being the membership function associated with the label A_i^s .

Among all the N_S possible fuzzy input subspaces, consider only those containing at least one positive example. To do so, the set of subspaces with positive examples is defined as $S^+ = \{S_j \mid E^+(S_j) \neq \emptyset\}$.

- 1.2. Generate the set of candidate rules in each subspace with positive examples: Firstly, the candidate consequent set associated with each subspace containing at least an example, $S_j \in S^+$, is defined. Two possibilities follow:

$$1.2.1. C(S_j) = \{B_k \in \mathcal{B} \mid \exists e_l \in E^+(S_j) \text{ with } \forall B' \in \mathcal{B}, \mu_{B_k}(y^l) \geq \mu_{B'}(y^l)\}.$$

$$1.2.2. C(S_j) = \{B_k \in \mathcal{B} \mid \exists e_l \in E^+(S_j) \text{ with } \mu_{B_k}(y^l) \neq 0\}.$$

Then, the candidate rule set for each subspace is defined as $CR(S_j) = \{R_k = [\text{IF } X_1 \text{ is } A_1^j \text{ and } \dots \text{ and } X_n \text{ is } A_n^j \text{ THEN } Y \text{ is } B_k] \mid B_k \in C(S_j)\}$.

In order to allow COR methodology to reduce the initial number for fuzzy rules, the special element $R_\emptyset$ (which means “do not care”) is added to each candidate rule set, i.e., $CR(S_j) = CR(S_j) \cup R_\emptyset$. If this element is selected, no rules are used in the corresponding fuzzy input subspace.

2. Selection of the most cooperative fuzzy rule set — This stage is performed by running a combinatorial search algorithm to look for the combination $\{R_1 \in CR(S_1), \dots, R_j \in CR(S_j), \dots, R_{|S^+|} \in CR(S_{|S^+|})\}$ with the best accuracy.

An index measuring the cooperation degree of the encoded rule set is considered to evaluate the quality of each solution. In our case, the algorithm uses a global error function called *mean square error* (MSE), which is defined as

$$\text{MSE} = \frac{1}{2 \cdot N} \sum_{l=1}^N (F(x_1^l, \dots, x_n^l) - y^l)^2,$$

with $F(x_1^l, \dots, x_n^l)$ being the output obtained from the FRBS when the example e_l is used, and y^l being the known desired output. The closer to zero the measure, the greater the global performance and, thus, the better the rule cooperation.

Fig. 1. COR algorithm

Fig. 2. COR-based learning process for a simple problem with two input variables ($n = 2$) and three labels in the output fuzzy partition ($|\mathcal{B}| = 3$): (a) data set (E) and data base previously defined; (b) the six examples are located in four ($|S^+| = 4$) different subspaces that determine the antecedent combinations and candidate consequents of the rules; (c) set of possible consequents for each subspace, including the special element “don’t care” (dc); (d) combinatorial search accomplished within a space composed of 72 different combinations of consequents; (e) rule decision table for the fifth combination; (f) rule base generated from this combination

Fig. 3. (a) Crisp grid, an example only contributes to the generation of one rule. (b) Fuzzy grid, an example may contribute to the generation of several rules

Again, depending on the approach followed (step 1.2.1. or 1.2.2. in Fig. 1), the search space will be different. Step 1.2.1. has a more restrictive condition and generates lesser number of candidate rules than step 1.2.2..

We should remark that the positive example set previously generated will also influence in the candidate rule sets.

This search space reduction is graphically depicted in Fig. 4 when solving a real-world modeling problem (see Sect. 5.1 for a description of it) with two input variables and five linguistic terms for each fuzzy partition. The approaches 1.1.1+1.2.1 (COR-1) and 1.1.2+1.2.2 (COR-2) in the proposed methodology are compared with a well-known rule base learning method, the one proposed by Thrift in [27].

From the example distribution tackled – Fig. 4(a) –, the Thrift method considers all the available fuzzy input subspaces and all the possible consequents (including the “don’t care” symbol) in each of them – Fig. 4(c) –. However, the COR methodology only considers a subset of subspaces and candidate consequents for each – Fig. 4(d) –. This significative reduction decreases the solution space size – Fig. 4(b) – and ease the search process.

3.2 Interpretability and Accuracy Issues

This section analyzes some aspects included in the COR methodology that allow it to obtain linguistic fuzzy models with a good interpretability and accuracy degree.

- *Cooperation Among Rules to Improve Accuracy* — The cooperation induced by the COR methodology allows us to obtain linguistic fuzzy models with a better accuracy. This is due to the interpolative reasoning developed by FRBSs, which is one of the most interesting features of these kinds of systems and plays a key role in their high performance, being

Fig. 4. Comparison between the search space tackled by the Thrift method and the COR methodology in a real-world problem with five linguistic terms for each variable

a consequence of the cooperative action among the linguistic rules. On the other hand, the fact of globally processing these rules makes COR be more robust against noise.

- *Model Structure and Membership Functions Invariable for an Excellent Interpretability* — The COR methodology is an effort to exploit the accuracy ability of linguistic FRBSs by exclusively focusing on the rule base design. In this case, the membership functions and the model structure keep invariable, thus resulting in the highest interpretability. Indeed, instead of improve the accuracy by deriving the shape of the membership functions or by extending the model structure (weighted rules, linguistic hedges, hierarchical knowledge bases, etc.), COR methodology improves the accuracy inducing cooperation among the linguistic fuzzy rules.
- *Rule Base Reduction to Improve Interpretability and Accuracy* — A problem when defining a rule base is that one can not be sure whether the rules are correctly defined, i.e., without redundant rules or rules that generate

conflicts with others in certain situations. To face this problem, a *rule reduction process* can be developed by combining rules and/or selecting a subset of rules from a given rule base to achieve the goal of minimizing the number of rules used while maintaining – or even improving – the FRBS performance. The badly defined and conflicting rules are eliminated by the method because their existence degrades the system performance. Some methods have been proposed to search for an optimized subset of rules, usually achieved by genetic algorithms [6,13,14,16]. These proposals generally perform the reduction with a postprocessing stage, once the rule base has been derived.

The COR methodology, however, achieves the reduction process at the same time as the learning one with the aim of improving the accuracy (the cooperation among rules and thus the system performance can be improved by removing rules) and interpretability (a model with less rules is more interpretable) of the learned model.

This process is performed by adding the null rule ($R_\emptyset$) to the candidate rule set corresponding to each subspace, as shown in the step 1.2. of Fig. 1. In this way, if such a element is selected for a specific subspace, this will mean that no rules will take part for the corresponding antecedent combination. Although the addition of $R_\emptyset$ in each candidate rule set increases the search space, more accurate and interpretable solutions can be obtained.

4 Application of Ant Colony Optimization to COR Methodology

COR is characterized by its flexibility to be used with different techniques of optimization and search. In [3], successful linguistic models have been obtained using simulated annealing. Nevertheless, these results could be improved incorporating heuristic information to the learning process. This consideration would guide the algorithm in the search, making it quicker on finding good solutions. ACO [9] is a good support for such intention thanks to the inherent use of heuristic information. Therefore, this section describes the use of ACO in the COR methodology. The following subsection briefly introduces ACO algorithms and, after that, the following five subsections present the different components of the algorithm.

4.1 Introduction to Ant Colony Optimization

ACO algorithms [8] constitute a new family of global search bio-inspired algorithms that has recently appeared. Since the first proposal, the Ant System algorithm [10] – applied to the Traveling Salesman Problem –, numerous models have been developed to solve a wide set of optimization problems (refer to [8] for a review of models and applications).

ACO algorithms draw inspiration from the social behavior of ants to provide food to the colony. In the food search process, consisting of the food find and the return to the nest, the ants deposit a substance called *pheromone*. The ants have the ability of sniffing the pheromone and pheromone trails guide the colony during the search. When an ant is located at a branch, it decides to take the path according to the probability defined by the pheromone existing in each trail. In this way, the depositions of pheromone terminate in constructing a path between the nest and the food that can be followed by new ants. The progressive action of the colony members involves the length of the path is progressively reduced. The shortest paths are finally the more frequently visited ones and, therefore, the pheromone concentration is higher on them. On the contrary, the longest paths are less visited and the associated pheromone trails are evaporated.

The basic operation mode of ACO algorithms is as follows [10]: at each iteration, a population of a specific number of ants progressively construct different tracks on the graph (i.e., solutions to the problem) according to a *probabilistic transition rule* that depends on the available information (heuristic and pheromone trails). After that, the pheromone trails are updated. This is done by first decreasing them by some constant factor (corresponding to the evaporation of the pheromone) and then reinforcing the solution attributes of the constructed solutions considering their quality. This task is developed by the *global pheromone trail update rule*.

Several extensions to this basic operation mode have been proposed. Their improvements mainly consist of using different transition and update rules, introducing new components, or adding a local search phase [2,9,24].

To apply ACO algorithms to a specific problem, the five steps shown in Fig. 5 have to be performed. The following sections describe these aspects particularized to the COR methodology.

1. *Problem representation*: Interpret the problem to be solved as a graph or a similar structure easily traveled by ants.
2. *Heuristic information*: Define the way of assigning a heuristic preference to each choice that the ant has to take in each step to generate the solution.
3. *Pheromone initialization*: Establish an appropriate way of initializing the pheromone.
4. *Fitness function*: Define a fitness function to be optimized.
5. *ACO algorithm*: Select an ACO algorithm and apply it to the problem.

Fig. 5. Steps followed to apply ACO algorithms to a specific problem

4.2 Problem Representation

For applying ACO in the COR methodology, it is convenient to see it as a combinatorial optimization problem with the capability of being represented on a graph. In this way, we can face the problem considering a fixed number of subspaces and interpreting the learning process as the way of assigning consequents – i.e., labels of the output fuzzy partition – to these subspaces with respect to an optimality criterion (i.e., following the COR methodology).

Hence, we are in fact dealing with an assignment problem and the problem representation can be similar to the one used to solve the *quadratic assignment problem* (QAP) [1], but with some peculiarities. We may draw an analogy between *subspaces* and locations and between *consequents* and facilities. However, unlike the QAP, the *set of possible consequents for each subspace may be different and it is possible to assign a consequent to more than one subspace* (two rules may have the same consequent). We can deduce from these characteristics that the order of selecting each subspace to be assigned a consequent is not determinant since one assignment does not restrict the remaining ones, i.e., the assignment order is irrelevant.

Therefore, according to Fig. 1, each node $S_j \in S^+$ is assigned to each candidate consequent $B_k \in C(S_j)$ and the especial symbol “don’t care” that stands for absence of rules in such a subspace.

Figure 6 depicts the graph corresponding to the example represented in Fig. 2.

Fig. 6. Graph used in the COR-based ACO algorithm for the example shown in Fig. 2

4.3 Heuristic Information

The heuristic information on the potential preference of selecting a specific consequent, B_k , in each antecedent combination (subspace) is determined as described in Fig. 7.

For each subspace $S_j \in S^+$ do:

1. Build the sets $E^+(S_j)$ and $C(S_j)$ as shown in Fig. 1.
2. For each $B_k \in C(S_j)$, make use of an initialization function based on covering criteria to give a heuristic preference degree to each choice. Many different possibilities may be considered. Three of them are the following:
 - (a) $\eta_{jk} = H_1(S_j, B_k) = \max_{e_l \in E^+(S_j)} \text{Min } \mu_{A^j}(x^l), \mu_{B_k}(y^l)$.
 - (b) $\eta_{jk} = H_2(S_j, B_k) = \frac{1}{|E^+(S_j)|} \sum_{e_l \in E^+(S_j)} \text{Min } \mu_{A^j}(x^l), \mu_{B_k}(y^l)$.
 - (c) $\eta_{jk} = H_3(S_j, B_k) = H_1(S_j, B_k) \cdot H_2(S_j, B_k)$.
 with $\mu_{A^j}(x^l) = \text{Min } \mu_{A_1^j}(x_1^l), \dots, \mu_{A_n^j}(x_n^l)$.
3. For each $B_k \notin C(S_j)$, make $\eta_{jk} = 0$.
4. Finally, for the “don't care” symbol, make the following:

$$\eta_{j,|\mathcal{B}|+1} = \frac{1}{|\mathcal{B}|} \sum_{B_k \in \mathcal{B}} \eta_{jk}.$$

Fig. 7. Heuristic assignment process

4.4 Pheromone Initialization

The initial pheromone value of each assignment is obtained as follows:

$$\tau_0 = \frac{1}{|S^+|} \sum_{S_j \in S^+} \max_{B_k \in \mathcal{B}} \eta_{jk}.$$

In this way, the initial pheromone will be the mean value of the path constructed taking the best consequent in each rule according to the heuristic information (a greedy assignment).

4.5 Fitness Function

The fitness function will be the said MSE, defined in Fig. 1.

4.6 Ant Colony System with Local Search Algorithm

Once the previous components have been defined, an ACO algorithm has to be given to solve the problem. In this contribution, the well-known ant colony system [9] is considered. A local search procedure is added to improve the behavior. The components adapted to our problem is introduced in the following.

Solution Construction. The algorithm introduces a transition rule that establishes a balance between biased exploration and exploitation of the available information. The node k (i.e., the consequent B_k) is selected for the

subspace S_j as follows:

$$k = \begin{cases} \arg \max_{R_u \in CR(S_j)} \{(\tau_{ju})^\alpha \cdot (\eta_{ju})^\beta\}, & \text{if } q < q_0 \\ T, & \text{otherwise} \end{cases},$$

with τ_{jk} being the pheromone of the trail (j, k) ; η_{jk} being the heuristic information; α and β being parameters which determine the relative influence of the pheromone strength and the heuristic information; q being a random variable uniformly distributed over $[0, 1]$; $q_0 \in [0, 1]$ being a threshold defining the probability of selecting the more hopeful coupling (exploitation); and with T being a random node selected according to the following transition rule (biased exploration):

$$p(j, k) = \begin{cases} \frac{(\tau_{jk})^\alpha \cdot (\eta_{jk})^\beta}{\sum_{R_u \in CR(S_j)} (\tau_{ju})^\alpha \cdot (\eta_{ju})^\beta}, & \text{if } R_k \in CR(S_j) \\ 0, & \text{otherwise} \end{cases},$$

We should note that, as in the QAP, the transition rule becomes an assignment rule but, contrary to that problem, there is not a need for the ant to keep a tabu list with the previous assignments made, since the same consequent can be assigned to different rules.

Pheromone Trail Update Rule. The pheromone trail update rule is performed in two stages, global and local:

- *Global pheromone trail update rule:* Only an ant – the one who generated the best solution (T_{best}) till now – releases pheromone on a coupling. The formula is the following:

$$\tau_{jk} \leftarrow (1 - \rho) \cdot \tau_{jk} + \rho \cdot \Delta\tau_{jk},$$

with $\rho \in [0, 1]$ being the pheromone evaporation parameter, m being the number of ants, T_a being the solution constructed by the ant a , and with

$$\Delta\tau_{jk} = \begin{cases} \frac{1}{\text{MSE}(RB_{best})}, & \text{if } (j, k) \in T_{best} \\ 0, & \text{otherwise} \end{cases}.$$

- *Local pheromone trail update rule:* Each time an ant covers a coupling, a local pheromone update is done as follows:

$$\tau_{jk} \leftarrow (1 - \rho) \cdot \tau_{jk} + \rho \cdot \Delta\tau_{jk}.$$

In this paper, we will consider $\Delta\tau_{jk} = \tau_0$ [9].

Local Search. One of the most usual ways to improve the performance of ACO algorithms is the use of local search techniques [19,2,24]. This approach entails employing a local optimization technique to refine the solutions obtained after one or several iterations.

In spite of using local search procedures usually improve the efficacy of the ACO algorithm, it increases the number of evaluations at each iteration and therefore the runtime of the learning method, thus losing efficiency. Moreover, we must consider that in our problem, opposite to other applications, the time needed to evaluate a neighboring solution is greater than the one needed to construct a new solution.

The local search technique usually is applied to the solution generated by each ant. However, in order to accelerate the process, in our case *the local search process will be only applied to the best solution generated at each iteration*. After this process, the global pheromone trail is updated in the usual way.

The proposed local search will consist of the simple hill-climbing algorithm described in Fig. 8.

Let $C_i = \{B_1^i, \dots, B_{c_i}^i\}$ – with $c_i \in \{1, \dots, N_c\}$ – be the candidate consequent set of the i -th rule. Let LSi and LSn be two values previously given by the learning method designer to respectively define the maximum number of iterations and the number of neighbors to create at each iteration. Do the following:

Let $S_{best} = \{R_1, \dots, R_j, \dots, R_{|S^+|}\}$ be the solution corresponding to the best track found in the current ACO algorithm iteration.

Set $S_{cur} \leftarrow S_{best}$.

For $h = 1, \dots, LSi$ do:

For $q = 1, \dots, LSn$ do:

- Obtain the solution S'_q applying a *neighbor generation mechanism* to S_{cur} , $S'_q \leftarrow N(S_{cur})$. This operator randomly selects a specific $j \in \{1, \dots, |S^+|\}$ and changes R_j by $R'_j \in CR(S_j) - \{R_j\}$. Therefore, $S'_q = \{R_1, \dots, R'_j, \dots, R_{S^+}\}$.
- If $q = 1$, set $S \leftarrow S'_1$. Else, if S'_q is better than S , set $S \leftarrow S'_q$.

• If S is better than S_{cur} , set $S_{cur} \leftarrow S$ and continue. Otherwise, break the loop.

Set the best track to the optimized solution, $S_{best} \leftarrow S_{cur}$.

Fig. 8. Local search process used in the ant colony system algorithm

5 Experimental Study

This section shows and analyzes some experimental results obtained by the COR-based ACO algorithm previously presented.

Two different approaches within the COR methodology are used. The former one (COR-1), makes a very strong search space reduction that directly affects to the number of rules generated and the solution space explored (see Sect. 3.1 for more details). This version uses the approaches 1.1.1 and 1.2.1 in the algorithm shown in Fig. 1. The latter one (COR-2), however, tackles a wider search space that, although generates higher number of rules, leads the method to obtain more accurate solutions. This second version uses the approaches 1.1.2 and 1.2.2 in the algorithm of Fig. 1.

Moreover, four methods have been selected to compare their performance with our proposals. The first one, proposed by Wang and Mendel in [28], is a simple algorithm that, although does not obtain good accuracy results, is a traditional reference in the area. The second one, proposed by Nozaki *et al.* in [21], uses linguistic fuzzy rules with double consequents and weights associated to them, moreover of considering an additional membership function parameter α to perform a non-linear scaling over the membership functions. The third one, proposed by Thrift in [27], is a basic GA-based learning method that only defines the fuzzy rule set. Finally, the fourth process, proposed by Liska and Melsheimer in [17], is a sophisticated learning method based on two stages that firstly designs the fuzzy rule set and the corresponding membership functions with a genetic algorithm-based process and then performs a final tuning process to refine the model.

On the other hand, the analyzed methods have been applied to two different real-world problems¹. Table 1 collects their main characteristics.

Table 1. Summary on the two applications considered and their main characteristics

Application	#V	#Tra	#Tst	#LT
Electrical line length	2	396	99	7
Electrical maintenance costs	4	847	212	5

#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

These applications are briefly described in the two following subsections. After that, the obtained results and an analysis of them are introduced in Sect. 5.3.

¹ The data sets used in this experiment can be downloaded at the web page <http://decsai.ugr.es/~casillas/FMLib/>

5.1 The Electrical Low Voltage Line Length Problem

This problem involves finding a model that relates the total length of low voltage line installed in Spanish rural towns [7]. 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 are considered.

5.2 The Electrical Network Maintenance Costs Problem

Estimating the maintenance costs of an electrical network in a town [7] is a complex but interesting problem. Since an actual measure is very difficult to obtain when medium or low voltage lines are used, the consideration of models becomes useful. These estimations allow electrical companies to justify their expenses. Moreover, the model must be able to explain how a specific value is computed for a certain town. Our objective will be to relate the *maintenance costs of medium voltage line* with the following four variables: *sum of the lengths of all streets in the town*, *total area of the town*, *area that is occupied by buildings*, and *energy supply to the town*. We will deal with estimations of minimum maintenance costs based on a model of the optimal electrical network for a town. We were provided with a sample of 1,059 simulated towns.

The sample has been randomly divided into two subsets, the training one with 847 elements and the test one with 212 elements (80%-20%). Five linguistic terms for each variable are considered.

5.3 Results and Analysis

Table 2 collects the results obtained by the analyzed learning methods, where $\#R$ stands for the number of rules, MSE_{tra} and MSE_{tst} for the values of the MSE over the training and test data sets respectively, and EBS for the number of evaluations needed to obtain the best solution. The best results for both applications are shown in boldface.

As regards the values of parameters used in the Nozaki *et al.*'s method, the best results were obtained with $\alpha = 5$ in both problems, which are the results shown in the table. In the Thrift's method, a population size of 61

Table 2. Results obtained by the analyzed methods

Method	Electrical line length				Electrical maintenance costs			
	#R	MSE _{tra}	MSE _{tst}	EBS	#R	MSE _{tra}	MSE _{tst}	EBS
Wang [28]	24	222,654	239,962	—	66	71,294	80,934	—
Nozaki [21]	64	185,395	170,489	—	536	35,859	42,218	—
Thrift [27]	49	169,077	175,739	26,706	534	34,063	42,116	40,464
Liska [17]	49	167,014	167,383	47,672	625	57,911	69,277	43,409
COR-1	19	171,492	188,966	2,785	52	47,096	51,649	7,987
COR-2	29	173,823	160,753	417	234	30,348	39,435	12,837

individuals, 1,000 generations, 0.6 as crossover probability, and 0.2 as mutation probability per chromosome were used. In the Liska and Melsheimer's method, 61 individuals, 1,000 generations, 49 and 625 as maximum number of rules for the electrical line length and electrical maintenance costs problems respectively, 0.6 as crossover probability, 0.1 as mutation probability per chromosome, and 0.1 as creep probability were used.

In the ACO algorithm of the COR method, the values of parameters were: $\rho = \{0.2, 0.4, 0.6, 0.8\}$ (any of these values obtained the same results), $\alpha = \{1, 2\}$ (the same results with both values), $\beta = 2$ and $\beta = 1$ for the electrical line length and the electrical maintenance costs problems respectively, $q_0 = 0.4$ and $q_0 = 0.2$ respectively for each problem, $LSi = 10$, and $LSn = 32$ and $LSn = 20$ respectively for each problem. With respect to the heuristic information considered, the H_3 and H_1 functions (see Sect. 4.3) was used for the electrical line length and the electrical maintenance costs problems, respectively. The number of iterations were 50 for all the cases.

From the obtained results, we can verify the good behavior of the COR methodology. The proposal COR-2 obtains the best generalization degrees (MSE_{tst}) in both problems. Moreover, this good accuracy is attained using lesser number of rules than the rest of analyzed methods, which improves the interpretability of the model.

Focusing on the two COR-based proposals shown, we can see the different interpretability and accuracy degrees obtained by them. The COR-1 method performs a very strong search space reduction that allows it to obtain fuzzy models with very compact rule sets, i.e., with small number of rules. Therefore, this approach is very suitable when the interpretability of the obtained model is an important issue. On the contrary, the COR-2 method generates fuzzy models with higher number of rules, thus obtaining better accuracy degrees. These results lead us to see the COR methodology as a useful tool to regulate the trade-off between both requirements.

Compared with the Nozaki *et al.*'s method, the COR-2 method obtains more accurate and interpretable models. While the former method needs using weighted double-consequents rules and a non-linear scaling factor to attain the shown accuracy, our method obtains more accurate method with simple, but cooperative linguistic fuzzy rules.

Opposite to the Thrift's method, COR-2 also obtains more accurate models in the second problem and better generalization degree in the former, with a significantly lesser number of rules. We must remark the difference of accuracy in the electrical maintenance costs, where in spite of being the search space of our method included into the one of the Thrift's method, a best solution is found with the COR-based method. This fact relates with the good exploration of the search space performed by our method by reducing the possible solution set and using heuristic information during the search.

Compared with the Liska and Melsheimer's method, one or both COR-based methods again obtain better accuracy degrees and significantly less number of rules. It is remarkable the fact that COR-1 method uses a fuzzy rule base 92% smaller than the one used by Liska and Melsheimer's method in the second problem, obtaining moreover better accuracy degrees. This turns out to be more surprising if we realize the latter method also derives the shape of the membership functions. The results seem to be related with the search spaces tackled by both approaches, which in the Liska and Melsheimer's method case is excessively large to be properly explored.

Finally, we may analyze the quickness of each learning process by comparing the number of evaluations of the fitness function needed to find the solution finally returned by the optimization algorithms. Thus, comparing the Thrift's method, the Liska and Melsheimer's one, and our proposals, we can verify that our methods not only obtain accurate and interpretable models, but they also generates them a far more quick than the other two methods. This aspect is interesting when several learning methods are hybridized to perform a more sophisticated modeling process and makes COR methodology very suitable for such purposes.

6 Concluding Remarks

This chapter has introduced a learning methodology to quickly generate accurate and simple linguistic fuzzy models, the COR methodology. It is based on a combinatorial search of fuzzy rules performed over a set of candidate ones to find those best cooperating. Therefore, instead of selecting the consequent with the highest performance in each fuzzy input subspace as other methods usually do, COR considers the possibility of using another consequent, different from the best one, when it allows the fuzzy model to be more accurate thanks to having a rule set with best cooperation.

The obtained experimental results lead us to think that the simple learning procedure performed by the COR methodology obtains linguistic fuzzy models with an excellent interpretability and good accuracy. The interpretability is ensured by keeping the membership functions and the model structure unaltered, as well as generating a compact rule base. The accuracy is achieved by developing a smart search space reduction and by inducing the generation of a linguistic fuzzy rule set with good cooperation.

References

1. E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm intelligence. From natural to artificial systems*. Oxford University Press, Oxford, UK, 1999.
2. B. Bullnheimer, R.F. Hartl, and C. Strauss. A new rank based version of the ant system: a computational study. *Central European Journal for Operations Research and Economics*, 7(1):25–38, 1999.
3. J. Casillas, O. Cordon, and F. Herrera. COR: A methodology to improve ad hoc data-driven linguistic rule learning methods by inducing cooperation among rules. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*. To appear.
4. J. Casillas, O. Cordon, and F. Herrera. Different approaches to induce cooperation in fuzzy linguistic models under the COR methodology. In B. Bouchon-Meunier, J. Gutiérrez-Ríos, L. Magdalena, and R.R. Yager, editors, *Techniques for constructing intelligent systems*, volume 1, pages 321–334. Physica-Verlag, Heidelberg, Germany, 2002.
5. O. Cordon, M.J. del Jesus, and F. Herrera. Genetic learning of fuzzy rule-based classification systems cooperating with fuzzy reasoning methods. *International Journal of Intelligent Systems*, 13:1025–1053, 1998.
6. O. Cordon and F. Herrera. A proposal for improving the accuracy of linguistic modeling. *IEEE Transactions on Fuzzy Systems*, 8(3):335–344, 2000.
7. O. Cordon, F. Herrera, and L. Sánchez. Solving electrical distribution problems using hybrid evolutionary data analysis techniques. *Applied Intelligence*, 10(1):5–24, 1999.
8. M. Dorigo and G. Di Caro. The ant colony optimization meta-heuristic. In D. Corne, M. Dorigo, and F. Glover, editors, *New ideas in optimization*, pages 11–32. McGraw-Hill, New York, NY, USA, 1999.
9. M. Dorigo and L.M. Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, 1997.
10. M. Dorigo, V. Maniezzo, and A. Coloni. The ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, 26(1):29–41, 1996.
11. J. Espinosa and J. Vandewalle. Constructing fuzzy models with linguistic integrity from numerical data-AFRELI algorithm. *IEEE Transactions on Fuzzy Systems*, 8(5):591–600, 2000.
12. A. González and 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):257–266, 1999.
13. F. Herrera, M. Lozano, and J.L. Verdegay. A learning process for fuzzy control rules using genetic algorithms. *Fuzzy Sets and Systems*, 100:143–158, 1998.
14. H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka. Selecting fuzzy if-then rules for classification problems using genetic algorithms. *IEEE Transactions on Fuzzy Systems*, 3(3):260–270, 1995.
15. Y. Jin, W. von Seelen, and B. Sendhoff. On generating FC³ fuzzy rule systems from data using evolution strategies. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, 29(4):829–845, 1999.
16. A. Krone, P. Krause, and T. Slawinski. A new rule reduction method for finding interpretable and small rule bases in high dimensional search spaces. In

- Proceedings of the 9th IEEE International Conference on Fuzzy Systems*, pages 693–699, San Antonio, TX, USA, 2000.
17. J. Liska and S.S. Melsheimer. Complete design of fuzzy logic systems using genetic algorithms. In *Proceedings of the 3rd IEEE International Conference on Fuzzy Systems*, pages 1377–1382, Orlando, FL, USA, 1994.
 18. B.-D. Liu, C.-Y. Chen, and J.-Y. Tsao. Design of adaptive fuzzy logic controller based on linguistic-hedge concepts and genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, 31(1):32–53, 2001.
 19. V. Maniezzo and A. Colomi. The ant system applied to the quadratic assignment problem. *IEEE Transactions on Knowledge and Data Engineering*, 11(5):769–778, 1999.
 20. D. Nauck and R. Kruse. Neuro-fuzzy systems for function approximation. *Fuzzy Sets and Systems*, 101(2):261–271, 1999.
 21. K. Nozaki, H. Ishibuchi, and H. Tanaka. A simple but powerful heuristic method for generating fuzzy rules from numerical data. *Fuzzy Sets and Systems*, 86(3):251–270, 1997.
 22. N.R. Pal and K. Pal. Handling of inconsistent rules with an extended model of fuzzy reasoning. *Journal of Intelligent and Fuzzy Systems*, 7:55–73, 1999.
 23. W. Pedrycz, R.R. Gudwin, and F.A.C. Gomide. Nonlinear context adaptation in the calibration of fuzzy sets. *Fuzzy Sets and Systems*, 88(1):91–97, 1997.
 24. M. Setnes and H. Hellendoorn. Orthogonal transforms for ordering and reduction of fuzzy rules. In *Proceedings of the 9th IEEE International Conference on Fuzzy Systems*, pages 700–705, San Antonio, TX, USA, 2000.
 25. Y. Shi, R. Eberhart, and Y. Chen. Implementation of evolutionary fuzzy systems. *IEEE Transactions on Fuzzy Systems*, 7(2):109–119, 1999.
 26. M. Sugeno and T. Yasukawa. A fuzzy-logic-based approach to qualitative modeling. *IEEE Transactions on Fuzzy Systems*, 1(1):7–31, 1993.
 27. P. Thrift. Fuzzy logic synthesis with genetic algorithms. In R.K. Belew and L.B. Booker, editors, *Proceedings of the 4th International Conference on Genetic Algorithms*, pages 509–513, San Mateo, CA, USA, 1991. Morgan Kaufmann Publishers.
 28. L.-X. Wang and J.M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(6):1414–1427, 1992.
 29. L.A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning. Parts I, II and III. *Information Science*, 8, 8, 9:199–249, 301–357, 43–80, 1975.