Learning Cooperative Linguistic Fuzzy Rules Using the Best–Worst Ant System Algorithm

Jorge Casillas,* Oscar Cordón,† Iñaki Fernández de Viana,‡
Francisco Herrera§
Department of Computer Science and Artificial Intelligence,
E.T.S. de Ingeniería Informática, Periodista Daniel Saucedo Aranda s/n,
University of Granada, E-18071 Granada, Spain

Within the field of linguistic fuzzy modeling with fuzzy rule-based systems, the automatic derivation of the linguistic fuzzy rules from numerical data is an important task. In the last few years, a large number of contributions based on techniques such as neural networks and genetic algorithms have been proposed to face this problem. In this article, we introduce a novel approach to the fuzzy rule learning problem with ant colony optimization (ACO) algorithms. To do so, this learning task is formulated as a combinatorial optimization problem. Our learning process is based on the COR methodology proposed in previous works, which provides a search space that allows us to obtain fuzzy models with a good interpretability–accuracy trade-off. A specific ACO-based algorithm, the Best–Worst Ant System, is used for this purpose due to the good performance shown when solving other optimization problems. We analyze the behavior of the proposed method and compare it to other learning methods and search techniques when solving two real-world applications. The obtained results lead us to remark the good performance of our proposal in terms of interpretability, accuracy, and efficiency.

1. INTRODUCTION

Fuzzy rule-based systems (FRBSs) currently constitute one of the most important areas for the application of fuzzy set theory. These systems are an extension of classical rule-based systems, because they deal with fuzzy rules instead of classical logic rules. From this point of view, a very interesting application of FRBSs is *system modeling*,[1] which in this field may be considered as an approach used to model a system making use of a descriptive language based on fuzzy logic with

*Author to whom all correspondence should be addressed: e-mail: casillas@decsai.ugr.es.
†e-mail: ocordon@decsai.ugr.es.
‡e-mail: herrera@decsai.ugr.es.
§e-mail: ijfviana@ugr.es.

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 20, 433–452 (2005)
 Published online in Wiley InterScience
(www.interscience.wiley.com). • DOI 10.1002/int.20074

fuzzy predicates.[2] In this framework, an important area is *linguistic fuzzy modeling*, where the interpretability of the obtained model is the main requirement. This task is developed by means of linguistic FRBSs, which use fuzzy rules composed of linguistic variables[3] that take values in a term set with a real-world meaning. Thus, the linguistic model consists of a set of linguistic descriptions regarding the behavior of the system being modeled.[2]

The task of automatically defining the fuzzy rules included in an FRBS for a concrete application is considered as a hard problem and a large number of methods have been proposed to generate the involved rules from numerical data making use of different techniques such as ad hoc data-driven methods,[4] neural networks,[5] genetic algorithms,[6] fuzzy clustering,[7] and so forth.

We should say that the two most important requirements in fuzzy modeling are the following:[8,9]

- *Interpretability*—This refers to the capability of the fuzzy model to express the behavior of the system in a understandable way. This is a subjective property that depends on several factors, the number of fuzzy rules being one of the most important.
- *Accuracy*—This refers to the capability of the fuzzy model to faithfully represent the modeled system. The closer the model to the system, the higher its accuracy. As closeness we understand the similarity between the responses of the real system and of the fuzzy model.

Along with these two aspects, the speed of the fuzzy rule derivation process is also of main importance. Indeed, a quick learning has some interesting advantages like the capability of being used as a previous mechanism to understand the features of the problem being solved, being used as a first learning stage to subsequently refine the obtained results with a more complex postprocessing,[10] being integrated within a meta-learning process,[11] and so forth.

In this contribution, we propose a novel approach to the fuzzy rule learning problem based on the Cooperative Rules (COR) methodology[12] and making use of ant colony optimization (ACO) algorithms to find a good balance between the three mentioned requirements (interpretability and accuracy of the obtained fuzzy models, and quickness of the learning process).

ACO[13–15] is a new paradigm of bio-inspired algorithms that has shown very good behavior when solving hard combinatorial optimization problems such as the *traveling salesman problem* or the *quadratic assignment problem* (QAP). The main advantage of this technique is the global search guided by heuristic and memoristic information in a very efficient way that it applies.

To apply ACO algorithms to the fuzzy rule learning problem, it must be formulated as a combinatorial optimization problem and the features related to ACO algorithms—such as heuristic information, pheromone initialization, fitness function, solution construction, and pheromone update—must be introduced.

To do so, the COR methodology may be used. This methodology proposes to perform the learning task in two steps: first, *a search space construction* is performed and then, *a selection of the most cooperative fuzzy rule set* is carried out guided by a global error measure.

Because heuristic information on the local goodness of each consequent to each rule is available, ACO algorithms properly fit into this methodology, profiting from this kind of information to perform a better search and to increase the convergence speed, thus generating interpretable and accurate fuzzy models quickly. The Best–Worst Ant System (BWAS) algorithm[16] is used in this article due to the good performance shown when solving other optimization problems.[17,18]

The article is structured as follows. Section 2 describes the COR methodology. Section 3 briefly introduces ACO and BWAS. Section 4 describes all the aspects related to applying the BWAS model to the COR methodology. Section 5 analyzes the behavior of the proposed method when solving two real-world applications, comparing it to other well-known fuzzy rule generation processes. Finally, Section 6 points out some concluding remarks.

2. COR: COOPERATIVE RULES 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.[12] Their high performance, in addition to their quickness and easy understanding, make them very suitable for fuzzy rule learning tasks.

However, ad hoc data-driven methods usually look for the fuzzy rules with the best individual performance (e.g., Ref. 19) and therefore the global interaction among the rules is not considered. This sometimes causes fuzzy rule sets to be obtained with bad cooperation among the rules composing them, making the fuzzy model not 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 Ref. 12: 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 fuzzy rule set with better cooperation.

COR consists of two stages:

1. *Search space construction*—where a set of candidate rules is obtained for each fuzzy input subspace.
2. *Selection of the most cooperative fuzzy rule set*—where a combinatorial search is performed among these sets looking for the combination of rules with the best global accuracy.

In this article, we also include an enhancement to the original proposal[12] to allow it to eliminate badly defined and conflicting rules with the aim of improving the interpretability (less number of rules) and the accuracy (better cooperation among rules). This approach, the fuzzy rule set reduction, is a regular practice in fuzzy modeling usually achieved by genetic algorithms.[10,20,21] These proposals generally perform the reduction with a postprocessing stage, once the rule set has been derived. Our proposal will achieve the reduction process at the same time as the learning one to address the existing interdependence between both processes.

To do so, the special element $R_\emptyset$ (which means "don't care") is added to the candidate rule set corresponding to each subspace. In this way, if such an element is selected for a specific subspace, this will mean that no rule belonging to this subspace will take part in the fuzzy rule set finally learned. This slight change in the COR methodology evidently involves increasing the search space with the known pros and cons: more accurate and interpretable solutions can be obtained but the difficulty of finding good solutions increases.

A wider description of the COR-based rule generation process including the fuzzy rule set reduction mechanism is shown in Figure 1.

Within the COR methodology, it is possible to follow two different approaches depending on the maximum number of rules and on the search space size we want to tackle:

- *Interpretability approach*—If the steps 1.1.1 and 1.2.1 in the algorithm shown in Figure 1 are considered, a lesser number of rules is generated and a narrower search space is explored. It involves making sacrifices in the accuracy for generating a reduced number of rules. According to the taxonomy performed in Ref. 12, this approach is guided by examples.
- *Accuracy approach*—However, if steps 1.1.2 and 1.2.2 are used, a wider search space is tackled with the subsequent accuracy improvement, but a higher number of rules will be obtained. According to the taxonomy performed in Ref. 12, this approach is guided by fuzzy grid.

Thus, depending on the problem nature (main requirement on the obtained fuzzy model, number of variables, number of linguistic terms, size and distribution of the example data set, difficulty to obtain accurate models, etc.), we should opt for one of the two mentioned approaches.

Regardless of which approach is followed, because the search space tackled in step 2 is usually large, it is necessary to use approximate search techniques (metaheuristics[22]). In this article, an efficient and accurate technique based on the BWAS algorithm is proposed.

3. INTRODUCTION TO ANT COLONY OPTIMIZATION

ACO algorithms[13] constitute a new family of global search bio-inspired algorithms that has been recently proposed. Since the first proposal, the ant system algorithm[23]—applied to the traveling salesman problem—numerous models have been developed to solve a wide set of optimization problems (refer to Refs. 14, 15, and 24 for a review on 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 finding and the return to the nest, ants deposit a substance called a *pheromone*. Ants have the ability to smell 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

Inputs:

- *An input-output data set*—$E = \{e_1, \ldots, e_l, \ldots, e_N\}$, with $e_l = (x_1^l, \ldots, x_n^l, y^l)$, $l \in \{1, \ldots, N\}$, 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 $i \in \{1, \ldots, n\}$, 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 $S_s = (A_1^s, \ldots, A_i^s, \ldots, A_n^s)$, with $A_i^s \in \mathcal{A}_i$ being a label, $s \in \{1, \ldots, N_S\}$, and $N_S = \prod_{i=1}^{n} |\mathcal{A}_i|$ being the number of fuzzy input subspaces. Two possibilities arise:

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

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

   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. Thus, 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. Again, there are two possibilities:

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

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

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

   To allow the COR methodology to reduce the initial number for fuzzy rules, the special element $R_\emptyset$ (which means "don't 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 rule is 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 of rules $\{R_1 \in CR(S_1), \ldots, R_j \in CR(S_j), \ldots, R_{|S^+|} \in CR(S_{|S^+|})\}$ 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

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

   with $F(x_1^l, \ldots, x_n^l)$ 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 higher the global performance and, thus, the better the rule cooperation.

Figure 1. COR algorithm.

a probability defined by the amount of pheromone existing in each trail. In this way, the depositions of the 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 makes the length of the path reduced step by step. The shortest paths are finally the more frequently visited ones and, therefore, the pheromone concentration is higher on them. Conversely, the longest paths are less visited and the associated pheromone trails are evaporated.

The basic operation mode of ACO algorithms is as follows:[23] At each iteration, a population of a specific number of ants progressively construct different tracks on a graph representing the problem instance (i.e., solutions to the problem) according to a *probabilistic transition rule* that depends on the available information (heuristic information 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 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.[25–27]

One of these successful approaches is the BWAS model.[16] It tries to improve the performance of ACO models using evolutionary algorithm concepts like the update rule based on that of the Population-Based Incremental Learning[28,29] (considering the global-best and the worst current solutions) or the pheromone trail mutation to introduce diversity in the search. A global scheme of the BWAS algorithm is shown in Figure 2.

To solve a specific problem by ACO algorithms, the five steps shown in Figure 3 have to be performed.[24] The following section describes these aspects particularized to the COR methodology.

1. Give an initial pheromone value, τ_0, to each edge.
2. While (*termination condition* is not satisfied) do:
    (a) Generate the track of each ant by the **solution construction process**.
    (b) Apply the **pheromone evaporation mechanism**.
    (c) Apply the **local search process** on the current-best solution.
    (d) Update $S_{global\ best}$ and $S_{current\ worst}$.
    (e) Apply the **Best-Worst pheromone trail update rule**.
    (f) Apply the **pheromone trail mutation**.
    (g) If (*stuck_condition* is satisfied) then apply **restart**.

Figure 2. BWAS algorithm.

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 scheme*: Select an ACO algorithm and apply it to the problem.

Figure 3. Steps followed to apply ACO algorithms to a specific problem.

4. BEST–WORST ANT SYSTEM TO LEARN LINGUISTIC FUZZY RULES

COR is characterized by its flexibility to be used with different metaheuristics. In Ref. 12, successful linguistic models were 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 more efficient and effective at finding good solutions. ACO is a good support for such an intention thanks to the inherent use of heuristic information. Therefore, this section describes the use of ACO, more specifically the BWAS model, in the COR methodology. To do so, the following five subsections present the different components of the proposed algorithm according to the scheme shown in Figure 3.

4.1. Problem Representation for Learning Cooperative Fuzzy Rules

To apply ACO in the COR methodology, it is convenient to see it as a combinatorial optimization problem with the capability of being represented on a weighted 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—that is, 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 QAP,[30] 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 draw from these characteristics that the order of selecting each subspace to be assigned a consequent is not determinant because one assignment does not restrict the remaining ones, that is, the assignment order is irrelevant.

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

Figure 4 shows an example of the learning process. In Figure 4c, the possible consequents for each antecedent combination are shown according to the data set and membership functions considered (Figure 4a). To construct a complete solution, an ant iteratively goes over each rule and chooses a consequent with a probability that depends on the pheromone trail τ_{ij} and the heuristic information η_{ij} associated to each decision, as usual (see Figure 4d). As said, the order of selecting the rules is irrelevant. Figure 4e, f shows the fuzzy rule set encoded by a specific solution.

4.2. 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 Figure 5.

4.3. Pheromone Initialization

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

$$\tau_0 = \frac{1}{|S^+|} \sum_{S_j \in S^+} \max_{B_k \in C(S_j)} \eta_{jk}$$

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).

4.4. Fitness Function

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

4.5. Ant Colony Optimization Scheme: Best–Worst Ant System Algorithm

Once the previous components have been defined, an ACO algorithm has to be given to solve the problem. In this contribution, the BWAS algorithm[16] is considered. The next subsections introduce its operation mode (see Figure 2) adapted to the fuzzy rule learning problem.

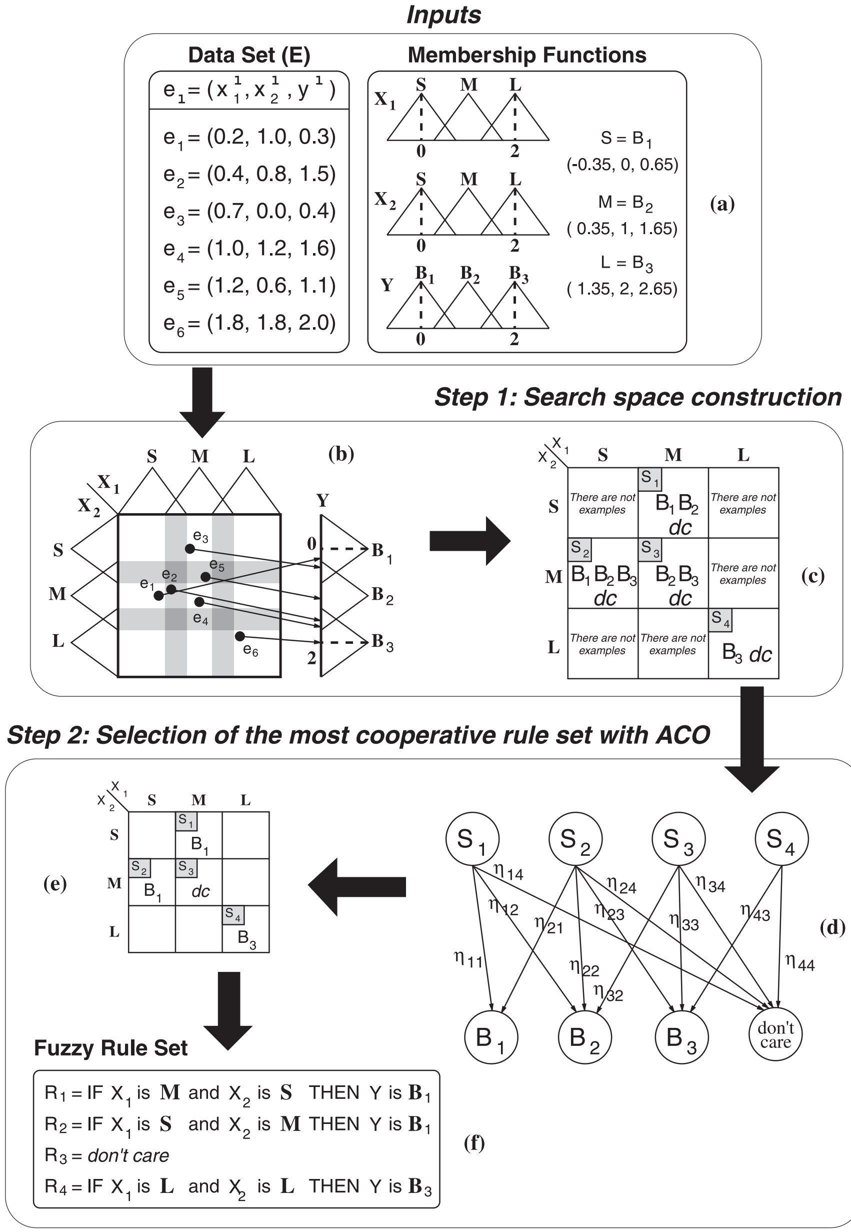

Figure 4. COR-based ACO learning process for a simple problem with two input variables ($n = 2$) and three labels in the output fuzzy partition ($|\mathcal{B}| = 3$): (a) Data set (E) and membership functions previously defined; (b) The six examples are located in four ($|S^+| = 4$) different subspaces 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) Graph constructed to explore the search space consisting of 72 different combinations of rules by means of the ACO algorithm; (e) Rule decision table for a possible solution to the problem; (f) Fuzzy rule set generated from this combination.

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

1. Build the sets $E^+(S_j)$ and $C(S_j)$ as shown in Figure 1.
2. For each $B_k \in C(S_j)$, make use of an initialization function based on a covering criterion to give a heuristic preference degree to each choice. Many different possibilities may be considered. Three of them are as follows [12]:

   (a) $\eta_{jk} = H_1(S_j, B_k) = \max_{e_l \in E^+(S_j)} Min\left(\mu_{A^j}(x^l), \mu_{B_k}(y^l)\right)$.

   (b) $\eta_{jk} = H_2(S_j, B_k) = \frac{1}{|E^+(S_j)|} \sum_{e_l \in E^+(S_j)} Min\left(\mu_{A^j}(x^l), \mu_{B_k}(y^l)\right)$.

   (c) $\eta_{jk} = H_3(S_j, B_k) = H_1(S_j, B_k) \cdot H_2(S_j, B_k)$.

   with $\mu_{A^j}(x^l) = Min\left(\mu_{A_1^j}(x_1^l), \ldots, \mu_{A_n^j}(x_n^l)\right)$.
3. For each $B_k \notin C(S_j)$, make $\eta_{jk} = 0$.
4. Finally, for the "don't care" symbol, make the following:

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

Figure 5. Heuristic assignment process.

4.5.1. Solution Construction Process

In the BWAS algorithm, at each construction step, the solution is built assigning a consequent to a rule with a probability given by the *transition rule*

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

with τ_{jk} being the pheromone of the edge (j,k) (i.e., the pheromone associated with the decision of assigning consequent B_k to the subspace S_j), η_{jk} being the heuristic information, and α and β being parameters that determine the relative influence of the pheromone trail and the heuristic information.

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, because the same consequent can be assigned to different rules.

4.5.2. Pheromone Evaporation Mechanism

The pheromone evaporation is done according to the following formula:

$$\tau_{jk} \leftarrow (1-\rho) \cdot \tau_{jk}, \forall j, \forall k$$

with $\rho \in [0,1]$ being the pheromone evaporation parameter.

4.5.3. Local Search Process

One of the most usual ways to improve the performance of ACO algorithms is the use of local search techniques.[26,27,30] This approach entails employing a local optimization technique to refine the solutions obtained after one or several iterations.

Despite the use of local search procedures that usually improves 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 the fuzzy rule learning problem, opposite to other applications, the time needed to evaluate a neighboring solution is greater than the one needed to construct a new solution.

In ACO, the local search technique is usually applied to the solution generated by each ant in each iteration. However, due to the time restrictions imposed in the fuzzy rule learning problem and to keep the high speed of the proposed learning approach, *the local search process will only be applied to the best solution generated at each iteration* in our case. After this process, the global pheromone trail is updated in the usual way (see Section 4.5.4).

The proposed local search will consist of the simple hill-climbing algorithm described in Figure 6.

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 generate at each iteration. Do the following:

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

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

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

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

- Obtain the solution T'_q applying a *neighbor generation mechanism* to T_{cur}, $T'_q \leftarrow N(T_{cur})$. This operator randomly selects a specific $j \in \{1, \ldots, |S^+|\}$ and changes R_j by $R'_j \in CR(S_j) - \{R_j\}$. Therefore, $T'_q = \{R_1, \ldots, R'_j, \ldots, R_{|S^+|}\}$.
- If $q = 1$, set $T \leftarrow T'_1$. Else, if T'_q is better than T, set $T \leftarrow T'_q$.

- If T is better than T_{cur}, set $T_{cur} \leftarrow T$ and continue. Otherwise, break the loop.

Set the best path to the optimized solution, $T_{best} \leftarrow T_{cur}$.

Figure 6. Local search process used in the BWAS algorithm.

4.5.4. *Best–Worst Pheromone Trail Update Rule*

The pheromone trail update is done according to the following formula:

$$\tau_{jk} \leftarrow \tau_{jk} + \Delta\tau_{jk}$$

where

$$\Delta\tau_{jk} = \begin{cases} \dfrac{1}{\mathrm{MSE}(FRS_{global\ best})}, & \text{if } (j,k) \in T_{global\ best} \\ 0 & \text{otherwise} \end{cases}$$

with $T_{global\ best}$ being the global-best track solution and $FRS_{global\ best}$ being its corresponding fuzzy rule set.

Then, all the edges existing in the current-worst path solution, $T_{current\ worst}$, that are not present in the global-best one are penalized by another decay of the pheromone trail associated—an additional evaporation—performed as follows:

$$\forall (j,k) \in T_{current\ worst} \text{ and } (j,k) \notin T_{global\ best}, \tau_{jk} \leftarrow (1-\rho)\cdot\tau_{jk}$$

4.5.5. *Pheromone Trail Mutation*

Each row of the pheromone matrix is mutated—with probability P_m—by adding or subtracting the same amount of pheromone to the selected trail (a value that depends on the current iteration) as follows:

$$\tau'_{jk} = \begin{cases} \tau_{jk} + mut(it, \tau_{threshold}), & \text{if } a = 0 \\ \tau_{jk} - mut(it, \tau_{threshold}), & \text{if } a = 1 \end{cases}$$

with a being a random value in $\{0,1\}$, it being the current iteration, $\tau_{threshold}$ being the average of the pheromone trail on the edges composing the global-best solution, and with $mut(\cdot)$ being

$$mut(it, \tau_{threshold}) = \frac{it - it_r}{Nit - it_r}\cdot\sigma\cdot\tau_{threshold}$$

where Nit is the maximum number of iterations of the algorithm and it_r is the last iteration where a restart was performed.

We should mention that the $mut(\cdot)$ function does not prevent pheromone values from being negative. Hence, there is a need to check their correction after each application of this component.

4.5.6. *Restart of the Search Process When It Gets Stuck*

The algorithm will perform the restart by setting all the pheromone matrix components to τ_0, the initial pheromone value, when the global-best solution is not improved during a fixed number of iterations.

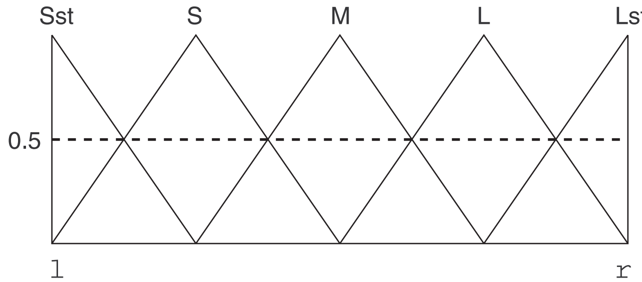

Figure 7. Graphical representation of a fuzzy partition with five linguistic terms, Sst standing for *smallest*, S for *small*, M for *medium*, L for *large*, and Lst for *largest*, and with $[l, r]$ being the corresponding variable domain.

5. EXPERIMENTAL STUDY

This section shows and analyzes some experimental results obtained by the previously presented COR-based method using the BWAS algorithm (COR-BWAS). We have selected five related methods to compare their performance with our proposal:

- The first one, proposed by Wang and Mendel,[19] is a simple algorithm that, although it does not obtain good accuracy results, is a traditional reference in the area.
- The second process, proposed by Nozaki et al.,[31] uses linguistic fuzzy rules with double consequents and weights associated to them to improve the fuzzy model performance.
- The third one, proposed by Thrift,[32] is a classical fuzzy rule set learning method based on genetic algorithms.
- Moreover, two COR-based methods based on a simulated annealing algorithm[12] and on the ant colony system algorithm[33] (the latter considering the same restart and local search procedures used in the COR-BWAS method) are also applied to analyze the performance of the BWAS-based approach.

In every analyzed method, we will consider uniformly distributed fuzzy partitions with symmetrical triangular membership functions crossing at height 0.5 (as shown in Figure 7) within the corresponding domain for each variable.

The analyzed methods have been applied to two different real-world problems. A fivefold cross validation is performed. Thus, each data set is divided into five subsets of (approximately) equal size. Each algorithm is applied five times for each problem, each time leaving out one of the subsets from training, but using only the omitted subset to compute the test error.[a] In the probabilistic algorithms, six runs with different seeds for the pseudorandom sequence are made for each data partition. Therefore, it involves 30 different runs of each algorithm for each problem. With this experimental setup that uses public real-world problems and cross-validation with multiple runs, we try to perform a sound experimental study,

[a]The data sets used in these experiments are available at http://decsai.ugr.es/~casillas/FMLib/

Table I. Summary of the two applications considered.

Application	#V	#E	#LT
Electrical line length	2	495	7
Electrical maintenance costs	4	1056	5

more rigorous than those usually performed by the fuzzy modeling community as remarked in Ref. 34.

Table I collects the main characteristics of the two analyzed real-world problems, where #V stands for the number of input variables, #E for the number of available examples, and #LT for the 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 Sections 5.4 and 5.5.

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.[35] This model was used to estimate the total length of line being maintained by an electrical company in 1999. 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 to more than 10,000 towns with these properties. We 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*. Seven labels are considered for each linguistic variable.

5.2. The Medium Voltage Electrical Network Maintenance Costs Problem

Estimating the maintenance costs of an electrical network in a town[35] is a complex but interesting problem. Because 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. That time, our objective was 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 dealt 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 1056 simulated towns. Five linguistic terms for each variable are considered.

5.3. Parameter Values Used

As regards the values of the parameters used in Nozaki et al.'s method, $\alpha = 1$ is considered to avoid membership function changes and make a fair comparison with the rest of fuzzy rule learning methods. In Thrift's method, a population size of 61 individuals, 1000 generations, 0.6 as crossover probability, and 0.2 as mutation probability per chromosome were used.

For the three analyzed COR-based methods, the *accuracy approach* (steps 1.1.2 and 1.2.2) was chosen for the electrical line length problem and the *interpretability approach* (steps 1.1.1 and 1.2.1) was used in the electrical maintenance costs problem. See Section 2 for more details. The decision was made attending to the maximum number of fuzzy input subspaces generated by each approach for each problem due to the training example data distribution. Although the interpretability approach generates 22 and 66 subspaces on average for the electrical line length and the maintenance costs problems, respectively, the accuracy approach generates 30 and 268 subspaces.

A previous experimentation with different combinations of parameter values over a specific data partition for each problem was developed in order to define the values producing the best behavior. In the COR-SA method, different initial temperatures between 50 to 5000 were tested. In the ACO-based methods, the following possibilities was considered: $\eta_{jk} = \{H_1, H_2, H_3\}$, $\rho = \{0.2, 0.4, 0.6, 0.8\}$, $\alpha = \{1,2\}$, $\beta = \{1,2\}$. The rest of the parameters were not varied. From this study, the following values were defined.

In the COR-SA method, the parameter values were: 2,000 as initial temperature, the maximum number of neighbors and the maximum number of acceptances were set to the number of initial fuzzy input subspaces generated by the COR methodology for each problem (30 on average in the electrical line length problem following the accuracy approach and 66 on average in the electrical maintenance costs problem following the interpretability approach), and 0.9 as exponential cooling factor. See Ref. 12 for the meaning of these parameters.

In the COR-ACS method, the values of parameters were: 50 iterations, a number of ants equal to the number of initial fuzzy input subspaces, H_3 as heuristic information (see Section 4.2), $\rho = 0.2$ and $\rho = 0.8$, respectively, for each problem, $\alpha = 1$ and $\alpha = 2$, respectively, for each problem, $\beta = 1$, $q_0 = 0.6$, $LSi = 10$, a value of LSn equal to the number of initial fuzzy input subspaces, and a number of iterations without improvement before performing restart equal to 5.

In the COR-BWAS method, the values selected for the parameters were: 50 iterations, a number of ants equal to the number of initial fuzzy input subspaces, H_2 and H_1 as heuristic information, respectively, for each problem, $\rho = 0.8$, $\alpha = 2$, $\beta = 2$, and $\beta = 1$, respectively, for each problem, $P_m = 0.3$, $\sigma = 4$, $LSi = 10$, a value of LSn equal to the number of initial fuzzy input subspaces, and a number of iterations without improving the results before performing restart equal to 5.

5.4. Results

Tables II and III collect the results obtained by the analyzed learning methods for each problem, 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 values shown for MSE_{tra}, MSE_{tst}, and EBS are rounded to the closer integer value. The arithmetic mean ($\bar{x}$) over the 30 runs performed, the standard deviation over the five mean

Table II. Results obtained by the analyzed methods in the electrical line length problem.

	$\bar{x}$				$\sigma_{\bar{x}_i}$				$\bar{\sigma}_{x_i}$			
Method	#R	MSE_{tra}	MSE_{tst}	EBS	#R	MSE_{tra}	MSE_{tst}	EBS	#R	MSE_{tra}	MSE_{tst}	EBS
WM	22.0	211,733	227,631	—	1.4	8,069	19,943	—	—	—	—	—
Nozaki et al.	60.8	182,297	205,779	—	1.9	2,764	29,132	—	—	—	—	—
Thrift	48.3	166,531	209,704	19,853	**0.8**	2,804	34,806	6,469	0.74	654	12,688	6,022
COR-SA	15.6	170,663	194,431	**333**	2.2	10,440	18,237	**137**	1.52	5,716	7,757	**117**
COR-ACS	28.4	172,349	198,416	786	2.2	3,381	22,810	543	1.28	1,061	4,370	393
COR-BWAS	**14.6**	**166,399**	**190,983**	4,166	2.1	**1,631**	**9,823**	3,206	**0.08**	**139**	**639**	2,409

Table III. Results obtained by the analyzed methods in the electrical maintenance costs problem.

	$\bar{x}$				$\sigma_{\bar{x}_i}$				$\bar{\sigma}_{x_i}$			
Method	#R	MSE_{tra}	MSE_{tst}	EBS	#R	MSE_{tra}	MSE_{tst}	EBS	#R	MSE_{tra}	MSE_{tst}	EBS
WM	65.0	56,135	56,359	—	**0.0**	1,498	4,685	—	—	—	—	—
Nozaki et al.	532.0	**26,705**	**27,710**	—	0.0	**764**	2,906	—	—	—	—	—
Thrift	565.3	31,228	37,579	50,204	2.6	1,018	7,279	**60**	6.11	2,110	3,609	**174**
COR-SA	43.7	41,086	42,907	**1,842**	0.9	1,173	**2,024**	149	2.54	2,387	3,230	280
COR-ACS	58.0	50,758	53,552	2,005	1.8	871	3,551	1,353	3.69	1,666	3,645	2,070
COR-BWAS	**41.0**	39,639	41,683	21,343	0.9	778	4,053	2,781	**1.40**	**566**	**1,599**	6,151

values ($\sigma_{\bar{x}_i}$), one per data partition, and the arithmetic mean of the standard deviation values over the six runs for each data partition ($\bar{\sigma}_{x_i}$) are included. The best results for each problem are shown in boldface.

Whereas $\sigma_{\bar{x}_i}$ stands for the differences existing among the data partitions, $\bar{\sigma}_{x_i}$ stands for the differences existing among the runs for each data partition. Therefore, the former value shows the robustness of the learning method to obtain similar results regardless of the data partition, whereas the latter value shows the robustness of the metaheuristic to obtain similar results regardless of the followed pseudorandom sequence.

5.5. Analysis of the Obtained Results

From the obtained results, we can verify the good behavior of the COR-BWAS method. It obtains very good accuracy degrees in both problems. Moreover, this precision is attained using a lesser number of rules than the rest of the analyzed methods, which notably improves the interpretability of the obtained linguistic fuzzy models.

More specifically, we will analyze its performance from three different points of view: interpretability and accuracy, robustness, and quickness.

Interpretability and accuracy. Focusing on the COR-based proposals shown, we can see how the fact of using a specific search algorithm affects the interpretability and accuracy degrees of the obtained models. The BWAS technique shows the best behavior performing a better search process that allows it to access better solutions than the simulated annealing and ant colony system ones. Indeed, the COR-BWAS method obtains the models with the best interpretability (#R) and accuracy degrees (MSE_{tra} and MSE_{tst}) in both problems among the three COR-based methods.

Compared with the NIT method, the COR-BWAS method obtains more accurate and interpretable models in the electrical line length problem, whereas less accurate models are obtained in the electrical maintenance costs problem. However, our method obtains models with many fewer rules, which significantly improves the interpretability. Moreover, whereas the former method needs to use weighted double-consequents rules to attain the shown accuracy, our method obtains models with simple, but cooperative linguistic fuzzy rules.

Opposite to Thrift's method, COR-BWAS obtains significantly smaller fuzzy rule sets (69% and 93%, respectively, for each problem) and good accuracy. We must note the results obtained by our method in the electrical line length problem, where even as the search space tackled by Thrift's method includes that managed by our method, the best solutions are found with the latter.

This fact relates to the good exploration of the search space achieved by our method that performs a reduction of the possible solution set and includes a smart use of the heuristic information during the search. Nevertheless, in the electrical maintenance costs problem, the fuzzy models obtained by our method do not show accuracy degrees as good as the ones obtained by Thrift's method.

Robustness. As regards the robustness of the analyzed method, we can note the good $\sigma_{\bar{x}_i}$ and $\bar{\sigma}_{x_i}$ values obtained by the COR-BWAS method. It obtains the best degrees in #R, MSE_{tra}, and MSE_{tst} among the probabilistic methods (Thrift and COR-based methods). This fact shows that our method is less sensitive to the data partition $(\sigma_{\bar{x}_i})$ and to the pseudorandom sequence $(\bar{\sigma}_{x_i})$ than the remainder.

Quickness. Finally, we may analyze the quickness of each learning process by comparing the number of fitness function evaluations needed to find the solution finally returned by the optimization algorithm. Thus, comparing Thrift's method with the COR-based methods, we can verify that the latter ones not only obtain accurate and interpretable models, but they also generate them far more quickly than Thrift's method. The COR-BWAS method is the slowest among the three COR-based ones. This is due to the fact that our proposal makes a deeper search process that delays its convergence.

Nonetheless, the COR-BWAS method appears to be 79% and 57% quicker than the genetic algorithm-based Thrift's method for the electrical line length and maintenance costs problems, respectively. The quickness is an interesting aspect when several learning methods are hybridized to perform a more sophisticated modeling process and makes the COR methodology very suitable for such purposes.

6. CONCLUDING REMARKS

This contribution has presented a novel and interesting application of ACO to the fuzzy rule learning problem. To do so, the learning task has been formulated as a combinatorial optimization problem following the COR methodology. For a better search exploration, a method using the ACO-based BWAS technique has been proposed.

Its proper fit to the fuzzy rule learning problem has been shown when solving two real-world problems. Compared to other learning approaches, significantly good models have been obtained by the proposed method. Our proposal has been shown to properly address the interpretability–accuracy trade-off by obtaining very compact and accurate linguistic fuzzy models.

Acknowledgments

This work was partially supported by Spanish MCyT (projects TIC2002-04036-C05-01 and TIC2003-00877) and FEDER funds.

References

1. Pedrycz W, editor. Fuzzy modelling: Paradigms and practice. Norwell, MA: Kluwer Academic; 1996.
2. Sugeno M, Yasukawa T. A fuzzy-logic-based approach to qualitative modeling. IEEE Trans Fuzzy Syst 1993;1(1):7–31.
3. Zadeh LA. Concept of a linguistic variable and its application to approximate reasoning, 1, 2, and 3. Inform Sci 1975;8:199–249, 301–357; 1976;9:43–80.

4. Bárdossy A, Duckstein L. Fuzzy rule-based modeling with application to geophysical, biological and engineering systems. Boca Raton, FL: CRC Press; 1995.
5. Nauck D, Klawonn F, Kruse R. Foundations of neuro-fuzzy systems. New York: Wiley; 1997.
6. Cordón O, Herrera F, Hoffmann F, Magdalena L. Genetic fuzzy systems: Evolutionary tuning and learning of fuzzy knowledge bases. Singapore: World Scientific; 2001.
7. Babuška R. Fuzzy modeling for control. Norwell, MA: Kluwer Academic; 1998.
8. Casillas J, Cordón O, Herrera F, Magdalena L, editors. Interpretability issues in fuzzy modeling. Heidelberg: Springer; 2003.
9. Casillas J, Cordón O, Herrera F, Magdalena L, editors. Accuracy improvements in linguistic fuzzy modeling. Heidelberg: Springer; 2003.
10. Cordón O, Herrera F. A three-stage evolutionary process for learning descriptive and approximate fuzzy logic controller knowledge bases from examples. Int J Approx Reason 1997; 17:369–407.
11. Cordón O, Herrera F, Villar P. Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base. IEEE Trans Fuzzy Syst 2001;9:667–674.
12. Casillas J, Cordón O, Herrera F. COR: A methodology to improve ad hoc data-driven linguistic rule learning methods by inducing cooperation among rules. IEEE Trans Syst Man Cybern B 2002;32:526–537.
13. Dorigo M, Stützle T. Ant colony optimization. Cambridge, MA: The MIT Press; 2004.
14. Dorigo M, Stützle T. The ant colony optimization metaheuristic: Algorithms, applications, and advances. In: Glover F, Kochenberger GA, editors. Handbook of metaheuristics. Norwell, MA: Kluwer Academic; 2003. pp 251–285.
15. Dorigo M, Di Caro G. The ant colony optimization meta-heuristic. In: Corne D, Dorigo M, Glover F, editors. New ideas in optimization. New York: McGraw-Hill; 1999. pp 11–32.
16. Cordón O, Herrera F, Fernández de Viana I, Moreno L. A new ACO model integrating evolutionary computation concepts: The best–worst ant system. In: Proc Second Int Workshop on Ant Algorithms, Brussels, Belgium, 2000. pp 22–29.
17. Cordón O, Fernández de Viana I, Herrera F. Analysis of the best–worst ant system and its variants on the TSP. Mathware Soft Comput 2002;9:177–192.
18. Cordón O, Fernández de Viana I, Herrera F. Analysis of the best–worst ant system and its variants on the QAP. In Dorigo M, Di Caro G, Sampels M, editors. Ant algorithms. LNCS 2463. Heidelberg: Springer-Verlag; 2002. pp 228–234.
19. Wang L-X, Mendel JM. Generating fuzzy rules by learning from examples. IEEE Trans Syst Man Cybern 1992;22:1414–1427.
20. Ishibuchi H, Nozaki K, Yamamoto N, Tanaka H. Selecting fuzzy if-then rules for classification problems using genetic algorithms. IEEE Trans Fuzzy Syst 1995;3:260–270.
21. Krone A, Krause P, Slawinski T. A new rule reduction method for finding interpretable and small rule bases in high dimensional search spaces. In: Proc Ninth IEEE Int Conf on Fuzzy Systems, San Antonio, TX, 2000. pp 693–699.
22. Glover F, Kochenberger GA, editors. Handbook of metaheuristics. Norwell, MA: Kluwer Academic; 2003.
23. Dorigo M, Maniezzo V, Colorni A. The ant system: Optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern B 1996;26:29–41.
24. Cordón O, Herrera F, Stützle T. A review on the ant colony optimization metaheuristic: Basis, models and new trends. Mathware Soft Comput 2002;9:141–175.
25. Dorigo M, Gambardella LM. Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Trans Evol Comput 1997;1:53–66.
26. Bullnheimer B, Hartl RF, Strauss C. A new rank based version of the ant system: A computational study. Centr Eur J Operat Res Econ 1999;7:25–38.
27. Stützle T, Hoos H. Max-min ant system. Future Generat Comput Syst 2000;8:889–914.
28. Baluja S. Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Technical Report CMU-CS-94-163. Pittsburgh, PA: Carnegie Mellon University; 1994.

29. Baluja S, Caruana R. Removing the genetics from standard genetic algorithm. In: Prieditis A, Russell S, editors. Proc Int Conf on Machine Learning. San Mateo, CA: Morgan Kaufmann; 1995. pp 38–46.
30. Maniezzo V, Colorni A. The ant system applied to the quadratic assignment problem. IEEE Trans Knowl Data Eng 1999;11:769–778.
31. Nozaki K, Ishibuchi H, Tanaka H. A simple but powerful heuristic method for generating fuzzy rules from numerical data. Fuzzy Set Syst 1997;86:251–270.
32. Thrift P. Fuzzy logic synthesis with genetic algorithms. In: Belew RK, Booker LB, editors. Proc Fourth Int Conf on Genetic Algorithms. San Mateo, CA: Morgan Kaufmann; 1991. pp 509–513.
33. Casillas J, Cordón O, Herrera F. Learning fuzzy rules using ant colony algorithms. In: Proc Second Int Workshop on Ant Algorithms, Brussels, Belgium, 2000. pp 13–21.
34. Cordón O, Herrera F. Author's reply [to comments on "A proposal for improving the accuracy of linguistic modeling"]. IEEE Trans Fuzzy Syst 2003;11:866–869.
35. Cordón O, Herrera F, Sánchez L. Solving electrical distribution problems using hybrid evolutionary data analysis techniques. Appl Intell 1999;10:5–24.