Learning Fuzzy Rules Using Ant Colony Optimization Algorithms [1]

Jorge Casillas, Oscar Cordón, Francisco Herrera

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

Abstract

Within the Linguistic Modeling field, one of the most important applications of Fuzzy Rule-Based Systems, the automatic learning from numerical data of the fuzzy linguistic rules composing these systems is an important task. In this paper we introduce a novel way of addressing the problem making use of Ant Colony Optimization (ACO) algorithms. To do so, the learning task will be formulated as an optimization problem and the features necessary for an ACO algorithm will be introduced. The behavior of the proposed learning method will be analyzed, compared with other ones, when solving of two applications with different characteristics: a three-dimensional function and a real-world electric engineering problem.

1 Introduction

Nowadays, one of the most important areas for the application of Fuzzy Set Theory are Fuzzy Rule-Based Systems (FRBSs). These kinds of systems constitute an extension of classical Rule-Based Systems, because they deal with fuzzy rules instead of classical logic rules [2]. An important application of FRBSs is *Linguistic Modeling*, which in this field may be considered as an approach used to model a system making use of a descriptive language based on Fuzzy Logic with fuzzy predicates [11], where the interpretability of the obtained model is the main requirement. Thus, the linguistic model consists of a set of linguistic descriptions regarding the behavior of the system being modeled.

In this approach, fuzzy linguistic IF-THEN rules are formulated and a process of fuzzification, inference, and defuzzification leads to the final decision of the system. Although sometimes the fuzzy rules can be directly derived from expert knowledge, different efforts have been made to obtain an improvement on system performance by incorporating learning mechanisms guided by numerical information to define the fuzzy rules. This issue, known as *fuzzy rule learning* (FRL), is considered a hard problem and a large number of methods has been proposed to automatically generate fuzzy rules from numerical data making use of different techniques such as ad hoc data-driven methods, neural networks, genetic algorithms, fuzzy clustering, etc. For a review on some of them, refer to [1].

In this contribution we propose a novel way of facing the FRL problem making use of Ant Colony Optimization (ACO) algorithms [3, 7]. To do so, the FRL problem will be formulated as an optimization problem and the features related to these kinds of algorithms—such as heuristic information, pheromone initialization, fitness function, solution construction, and pheromone update—will be introduced.

With this aim, the paper is set up as follows. In Section 2, a brief introduction to FRBSs and the FRL problem is presented. Section 3 is devoted to introduce all the aspects related to

[1]This research has been supported by CICYT under project PB98-1319

ACO algorithms particularized to the FRL problem. In Section 4, the behavior of the proposed learning approach to solve two different applications is analyzed. Finally, in Section 5, some concluding remarks will be pointed out.

2 Fuzzy Rule-Based Systems and Fuzzy Rule Learning Problem

2.1 Introduction to Fuzzy Rule-Based Systems

An FRBS presents two main components: 1) the *Knowledge Base* (KB), representing the knowledge about the problem being solved in the form of fuzzy linguistic IF-THEN rules, and 2) the *Inference Engine*, which puts into effect the fuzzy inference process needed to obtain an output from the FRBS when an input is specified. The structure of a linguistic FRBS is shown in Figure 1.

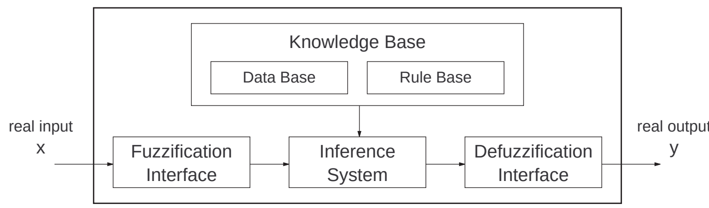

Figure 1: Generic structure of a linguistic Fuzzy Rule-Based System

- The KB is composed of the Rule Base (RB), constituted by the collection of linguistic rules themselves joined by means of the connective *also*, and of the Data Base (DB), containing the term sets and the membership functions defining their semantics. The fuzzy linguistic rule structure considered in linguistic FRBSs is the following:

  $$R_i: \text{ IF } X_1 \text{ is } A_{i1} \text{ and ... and } X_n \text{ is } A_{in} \text{ THEN } Y \text{ is } B_j \;\;,$$

  with $X_1, \ldots, X_n$ and Y being the input and output linguistic variables, respectively, and $A_{i1}, \ldots, A_{in}$ and B_j being linguistic labels, each one of them having associated a fuzzy set defining its meaning.

- The Inference Engine is comprised by three components: a *Fuzzification Interface*, which has the effect of transforming crisp input data into fuzzy sets, an *Inference System*, that uses these together with the KB to perform the fuzzy inference process, and a *Defuzzification Interface*, that obtains the final crisp output from the individual fuzzy outputs inferred.

  The *Inference System* is based on the application of the Generalized Modus Ponens, extension of the classical logic Modus Ponens. It is done by means of the Compositional Rule of Inference, which in its simplest form is reduced to [5]:

  $$R_i(x_0, y) = \mu_{B'_i}(y) = I(\mu_{A_i}(x_0),\; \mu_{B_j}(y)) \;\;,$$

  with $x_0 = (x_1, \ldots, x_n)$ being the current system input, $\mu_{A_i}(x_0) = T(\mu_{A_{i1}}(x_1), \ldots, \mu_{A_{in}}(x_n))$ being the matching degree between the rule antecedent and the input —with $\mu_{A_{ik}}(\cdot)$ being the membership function of the label A_{ik} and T being a conjunctive operator (a *t-norm*)—, and I being a fuzzy implication operator.

2.2 The Fuzzy Rule Learning Problem

Several tasks have to be performed in order to design an FRBS for a concrete application. One of the most important and difficult ones is to obtain an appropriate KB about the problem being solved, in the following referred to as FRL problem. The difficulty presented by the human experts to express their knowledge in the form of fuzzy rules has made researchers develop automatic techniques for performing this task. For a review on some of them, refer to [1].

All these methods are based on working with an input-output data set $E = \{e_1, \ldots, e_N\}$, $e_l = (x_1^l, \ldots, x_n^l, y^l)$, representing the behavior of the problem being solved, and with a previous definition of the DB composed of the input and output primary fuzzy partitions. In our case, we will consider symmetrical fuzzy partitions with a number of triangular membership functions crossing at height 0.5 (as shown Figure 2 for the case of seven fuzzy sets). Therefore, our FRL problem will be restricted to obtain the rules combining the labels of the antecedents and assigning a specific consequent to each antecedent combination.

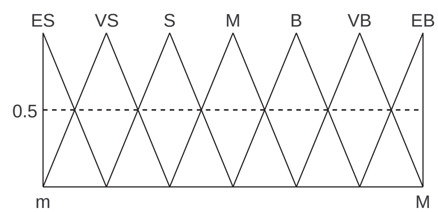

Figure 2: Graphical representation of a uniform fuzzy partition with seven labels

3 Ant Colony Optimization Algorithms for Learning Fuzzy Rules

To apply ACO algorithms to a specific problem, the following steps have to be performed:

1. Obtain a problem representation as a graph or a similar structure easily covered by ants.
2. 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. Establish an appropriate way of initializing the pheromone.
4. Define a fitness function to be optimized.
5. Select an ACO algorithm and apply it to the problem.

In the following subsections, these steps will be introduced to solve the FRL problem.

3.1 Problem Representation

To apply ACO algorithms to the FRL problem, 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 rules and interpreting the FRL problem as the way of assigning consequents (i.e., labels of the output fuzzy partition) to these rules with respect to an optimality criterion.

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) [3, 7], but with some peculiarities. We may draw an analogy between *rules* and facilities and between *consequents* and locations. However, unlike the QAP, the *set of possible consequents for each rule may be different* and *it is possible to assign a consequent to more than one rule* (two rules may have the same consequent). We can deduce from these characteristics that the order of selecting each rule to be assigned a consequent is not determinant, i.e., the assignment order is irrelevant.

To construct the graph, the following steps are taken:

1. *Determine the rules*: A rule R_i —$i = 1, \ldots, N_r$— defined by an antecedent combination,

$$R_i = IF\ X_1\ is\ A_{i1}\ and\ \ldots\ and\ X_n\ is\ A_{in}\ ,$$

   will take part in the graph if and only if:

$$\exists e_l = (x_1^l, \ldots, x_n^l, y^l) \in E\ such\ that\ \mu_{A_{i1}}(x_1^l) \cdot \ldots \cdot \mu_{A_{in}}(x_n^l) \neq 0\ .$$

   That is, there is at least one example located in the fuzzy input subspace defined by the antecedents considered in the rule.

2. *Link the rules to consequents*: The rule R_i will be linked to the consequent B_j —$j = 1, \ldots, N_c$— (taken from the set of labels of the output fuzzy partition) if and only if it meets the following condition:

$$\exists e_l = (x_1^l, \ldots, x_n^l, y^l) \in E\ such\ that\ \mu_{A_{i1}}(x_1^l) \cdot \ldots \cdot \mu_{A_{in}}(x_n^l) \cdot \mu_{B_j}(y^l) \neq 0\ .$$

   That is, there is at least one example located in the fuzzy input subspace that is covered by such a consequent.

Figure 3 shows an example of a system with four rules and one output variable with three consequents. In Figure 3(a), the possible consequents for each antecedent combination are shown. 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}, as usual (see Figure 3(b)). As said, the order of selecting the rules is irrelevant. In Figure 3(c) we may see the possible paths that an ant can take in a specific example.

3.2 Heuristic Information

The heuristic information on the potential preference of selecting a specific consequent, B_j, in each antecedent combination (rule) is determined by considering covering criteria as follows (see Figure 4 for a graphical interpretation of the heuristic assignment):

For each rule defined by an antecedent combination, $R_i = IF\ X_1\ is\ A_{i1}\ and\ \ldots\ and\ X_n\ is\ A_{in}$ —$i = 1, \ldots, N_r$— do:

1. Build the set E'_i composed of the input-output data pairs that are located in the input subspace defined by R_i, i.e., $E'_i = \{e_l = (x_1^l, \ldots, x_n^l, y^l) \in E\ such\ that\ \mu_{A_{i1}}(x_1^l) \cdot \ldots \cdot \mu_{A_{in}}(x_n^l) \neq 0\}$.

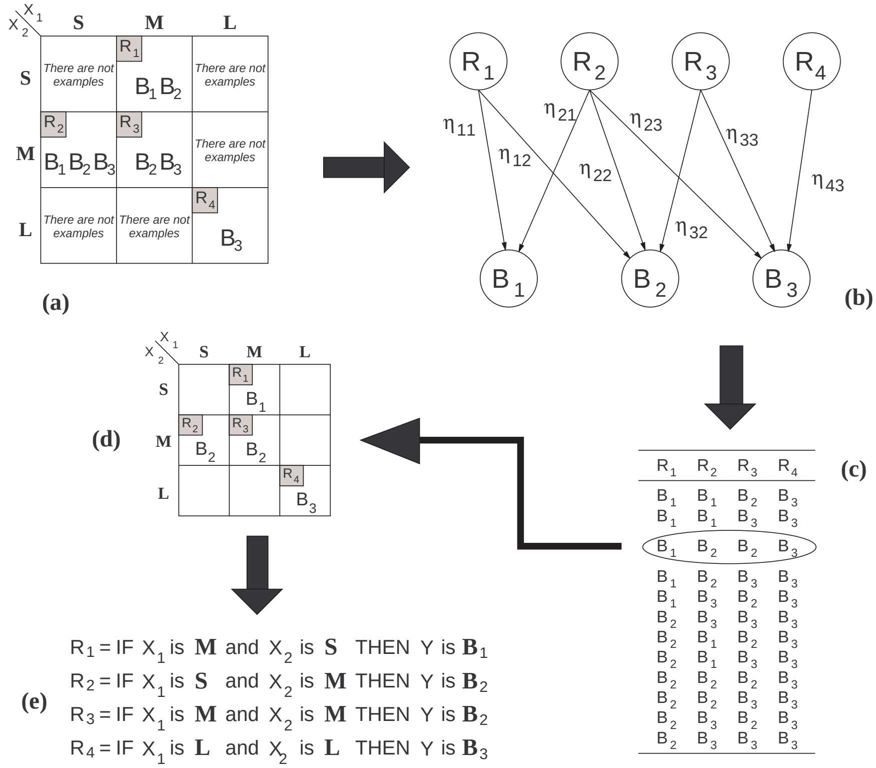

Figure 3: Learning process for a simple problem with two input variables ($n = 2$), four rules ($N_r = 4$), and three labels in the output fuzzy partition (N_c=3): (a) Set of possible consequent for each rule (only the rules where at least one example is located in the corresponding subspace are considered); (b) Graph of paths where $\eta_{ij} \neq 0$ except η_{13}, η_{31}, η_{41}, and η_{42}, which are zero; (c) It is possible to take twelve different paths (combinations of consequents); (d) Rule decision table for the third combination; (e) RB generated from the third combination

2. Make use of an initialization function based on covering criteria to give a heuristic preference degree to each election. Many different choices may be considered [4]. In this paper we will work with the *covering of the example best covered* criterion shown in Figure 4.

Since the heuristic information is based on covering criteria, it will be zero for a specific consequent when no examples located in the fuzzy input subspace are covered by it. This means that for a rule, only those links to consequents whose heuristic information is greater than zero will be considered. In Figure 3(b) we can observe the consequent B_3 can not be assigned to the rule R_1, the consequent B_1 can not be assigned to the rule R_3, and the consequents B_1 and B_2 can not be assigned to the rule R_4 because their heuristic informations (covering degrees) are zero.

3.3 Pheromone Initialization

The initial pheromone value of each assignment is obtained as follows: $\tau_0 = \frac{\sum_{i=1}^{N_r} \max_{j=1}^{N_c} \eta_{ij}}{N_r}$. 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).

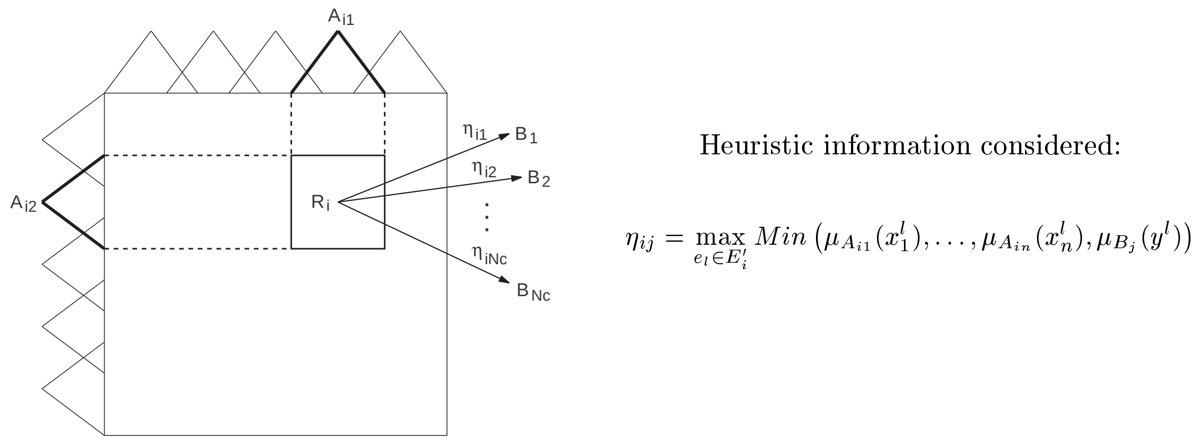

Figure 4: Heuristic assignment from the rule R_i to each consequent in a system with two input variables, five labels for each of them, and N_c labels (consequents) in the output fuzzy partition. The *covering of the example best covered* is considered to be the heuristic information

3.4 Fitness Function

The fitness function establishes the quality of a solution. The measure considered will be the function called *mean square error* (MSE), which defined as $MSE(RB_k) = \frac{1}{2 \cdot |E|} \sum_{e_l \in E} (y^l - F_k(x_0^l))^2$, with $F_k(x_0^l)$ being the output obtained from the FRBS (built using the RB generated by the ant k, RB_k) when receives the input x_0^l (input component of the example e_l), and y^l being the known desired output. The closer to zero the measure is, the better the solution is.

3.5 Ant Colony Optimization Algorithm

Once the previous components have been defined, an ACO algorithm has to be given to solve the problem. In this contribution, two well-known ACO algorithms will be considered: the Ant System (AS) [9] and the Ant Colony System (ACS) [8]. Depending on the ACO algorithm followed, two methods arise: the AS-FRL and the ACS-FRL ones. The so-known *solution construction* and *pheromone trail update rule* considered by these ACO algorithms will be used. Only some adaptations will be needed to apply them to the FRL problem:

- The set of nodes attainable from R_i (set of feasible neighborhood of node R_i) will be $J_k(i) = \{j \text{ such that } \eta_{ij} \neq 0\}$ in the transition rules considered by both ACO algorithms when constructing the solution.
- The amount of pheromone ant k puts on the couplings belonging to the solution constructed by it will be $1/MSE(RB_k)$, with RB_k being the RB generated by ant k.
- In the *local pheromone trail update rule* of the ACS algorithm, the most usual way of calculating $\Delta\tau_{ij}$, $\Delta\tau_{ij} = \tau_0$, we will be used, thus considering the simple-ACS algorithm.

4 Examples of Application

With the aim of analyzing the behavior of the proposed ACO processes, we have chosen two different applications: the fuzzy modeling of a three-dimensional function and a real-world electric engineering problem [6]. We will compare them with two well-known ad hoc rule learning methods whose high performance has been clearly demonstrated: the method proposed by Wang and Mendel (WM-method) [12] and the one proposed by Nozaki, Ishibuchi, and Tanaka (NIT-method) [10]. Two new methods have also been developed with the aim of comparing the ACO approach with other optimization ones. These two methods are based on the same problem representation presented in this paper (combinatorial search of consequents among a set of candidates for each rule) but using a Simulated Annealing algorithm (SA-FRL) and a Genetic Algorithm (GA-FRL) to accomplish the search. Finally, a greedy algorithm directly based on the heuristic information (HI-FRL) by taking the consequent with the highest value for each rule, which was proposed in [4], will be also considered. The results presented for each algorithm have been taken after a hard experimentation with the parameters in order to look for the best behavior.

An initial DB constituted by a primary fuzzy partition for each variable will be considered in each case. Every partition is formed by seven labels with triangular-shaped equally distributed fuzzy sets giving meaning to them (as shown in Figure 2), and the appropriate scaling factors to translate the generic universe of discourse into the one associated with each problem variable. With respect to the FRBS reasoning method used, we have selected the *minimum t-norm* playing the role of the implication and conjunctive operators, and the *center of gravity weighted by the matching* strategy acting as the defuzzification operator [5].

Concerning the parameters used in the ACO algorithms, the number of ants will be the number of rules in each case, the number of iterations will be 50, and for the rest of parameters (ρ, α, and β, for both AS-FRL and ACS-FRL, and q_0 for ACS-FRL) an experimental study has been performed, showing in the tables the best results.

4.1 Linguistic Modeling of a Simple Three-Dimensional Function

For this first experiment, a simple unimodal three-dimensional mathematical function is considered to be modeled, $F(x_1, x_2) = x_1^2 + x_2^2$, with $x_1, x_2 \in [-5, 5]$ and hence $F(x_1, x_2) \in [0, 50]$. A set with $1,681$ values has been generated for the training data set. Another set with 168 values (the ten percent of the training set) has been generated for its use as test set to evaluate the performance of the learning methods, avoiding any possible bias related to the data in the training set.

The results obtained by the seven methods analyzed are collected in Table 1, where #R stands for the number of rules, MSE_{tra} and MSE_{tst} for the values obtained over the training and test data sets respectively, and EBS for the number of evaluations needed to obtain the best solution. The best results are shown in boldface.

Analyzing these results, we may note the high performance of the ACO methods. Opposite to the three ad hoc learning methods, the models generated by AS-FRL and ACS-FRL are clearly better in both approximation (MSE_{tra}) and generalization (MSE_{tst}). Focusing on the methods based on combinatorial search, the ACS-FRL is the algorithm that performs the best search process obtaining the most accurate model regarding approximation, and with a good generalization. However, the four methods obtain similar results (being slightly worse the approximation degree of the model generated by AS-FRL) and is in the convergence speed where the ACO approaches stand out. As notice, ACS-FRL found the best solution three times quicker than the SA approach and seventeen times quicker than the GA. In AS-FRL, the differences are still more significant. This fact is due to the use of heuristic information that guides the ACO algorithms in the search process.

Table 1: Results obtained in the modeling of F

Method	**#R**	**MSE**$_{tra}$	**MSE**$_{tst}$	**EBS**	**Parameters**
WM-method	**49**	2.048137	2.255928	0	—
NIT-method	98	2.465487	1.768125	0	—
HI-FRL	**49**	2.048137	2.255928	0	—
SA-FRL	**49**	1.609891	**1.213388**	3,528	Init. temp. = 40, No. of neighbors = 98
GA-FRL	**49**	1.606097	1.514651	20,555	500 gen., 61 indiv., $P_c = 0.6$, $P_m = 0.2$
AS-FRL	**49**	1.660622	1.419587	**686**	$\alpha = 1$, $\beta = 2$, $\rho = 0.2$
ACS-FRL	**49**	**1.601071**	1.350340	1,225	$\alpha = 1$, $\beta = 1$, $\rho = 0.2$, $q_0 = 0.4$

4.2 The Electrical Distribution Network Problem

Sometimes, there is a need to measure the amount of electricity lines that an electric company owns. This measurement may be useful for several aspects such as the estimation of the maintenance costs of the network, which was the main goal in this application [6]. The problem involves finding a model that relates the total length of low voltage line installed in a rural town with the number of inhabitants in the town and the mean of the distances from the center of the town to the three furthest clients in it. This model will be used to estimate the total length of line being maintained.

To compare the methods, we have randomly divided the sample, composed of 495 pieces of real data obtained from direct measures in this number of villages, into two sets comprising 396 and 99 samples, labeled training and test. The results obtained with the considered methods are collected in Table 2.

Table 2: Results obtained in the electrical application

Method	**#R**	**MSE**$_{tra}$	**MSE**$_{tst}$	**EBS**	**Parameters**
WM-method	**24**	222,654	239,962	0	—
NIT-method	64	185,395	170,489	0	—
HI-FRL	32	239,393	275,953	0	—
SA-FRL	32	**174,295**	161,261	1,248	Init. temp. = 500, No. of neighbors = 32
GA-FRL	32	175,122	187,605	20,512	500 gen., 61 indiv., $P_c = 0.6$, $P_m = 0.2$
AS-FRL	32	178,119	**158,662**	**384**	$\alpha = 1$, $\beta = 2$, $\rho = 0.6$
ACS-FRL	32	175,096	165,561	576	$\alpha = 1$, $\beta = 2$, $\rho = 0.2$, $q_0 = 0.2$

From the obtained results, we may again note the good performance of the ACO approaches that outperform the three ad hoc learning methods. Among the four combinatorial search algorithms, the AS-FRL performs a search a little worse than the rest but obtains the best model with respect to generalization. ACS-FRL obtains a very good model only overcame to a lesser extent by the SA-FRL method. Again, the main advantage of the ACO algorithms lies in the convergence speed, which in the case of the ACS-FRL method is twice quicker than the SA approach and thirty five times quicker than the GA-FRL method, moreover obtaining a most accurate model in this latter case.

5 Concluding Remarks

In this paper, a novel and interesting application, the FRL problem (which involves automatically learning from numerical data the RB composing an FRBSs), has been proposed to be solved by the ACO meta-heuristic. In this way, two specific ACO-based learning methods have been presented. Their high performance has been shown in the solving of two problems. Comparing with other ad hoc learning algorithms, the models obtained by the ACO methods are clearly better. Moreover, opposite to other kinds of optimization techniques as SA and GAs, the ACO approach performs a quick convergence and sometimes obtains better results. The former is due to the use of heuristic information to guide the global search. As further work, we propose to apply new ACO approaches to the FRL problem using new features such as the local search to improve the performance of the models designed.

References

[1] R. Alcalá, J. Casillas, O. Cordón, F. Herrera, and S.J.I. Zwir. Learning and tuning fuzzy rule-based systems for linguistic modeling. In C.T. Leondes (Ed.), *Knowledge-Based Systems*, vol. 3, ch. 29. Academic Press, 2000.

[2] A. Bárdossy and L. Duckstein. *Fuzzy rule-based modeling with application to geophysical, biological and engineering systems*. CRC Press, 1995.

[3] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, 1999.

[4] O. Cordón and F. Herrera. A proposal for improving the accuracy of linguistic modeling. *IEEE Trans. on Fuzzy Systems*, 8(3), 2000.

[5] O. Cordón, F. Herrera, and A. Peregrín. Applicability of the fuzzy operators in the design of fuzzy logic controllers. *Fuzzy Sets and Systems*, 86: 15–41, 1997.

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

[7] M. Dorigo and G. Di Caro. The ant colony optimization meta-heuristic. In D. Corne, M. Dorigo, and F. Glover (Eds.), *New Ideas in Optimization*, pp. 11–32. McGraw-Hill, 1999.

[8] M. Dorigo and L.M. Gambardella. Ant colony system: a cooperative learning approach to the travelling salesman problem. *IEEE Trans. on Evolutionary Computation*, 1(1): 53–66, 1997.

[9] M. Dorigo, V. Maniezzo, and A. Colorni. The ant system: optimization by a colony of cooperating agents. *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, 26(1): 29–41, 1996.

[10] 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: 251–270, 1997.

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

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