

# A Hybrid Learning Process for the Knowledge Base of a Fuzzy Rule-Based System

J. Casillas, O. Cordon, F. Herrera

Dept. Computer Science & AI  
E.T.S. Ing. Informática  
Universidad de Granada  
18071 - Granada, Spain  
casillas,ocordon,herrera@decsai.ugr.es

P. Villar

Dept. Informática  
E.S. de Ing. Informática  
Universidade de Vigo  
32004 - Ourense, Spain  
pvillar@uvigo.es

## Abstract

Within the field of linguistic fuzzy modeling with fuzzy rule-based systems, the automatic derivation of the knowledge base from numerical data is an important task. In this contribution, we propose a new approach to automatically learn the whole knowledge base, combining two different strategies for rules derivation and fuzzy partitions definition, working cooperatively in order to obtain accurate and interpretable models.

**Keywords:** linguistic fuzzy modeling, fuzzy rule-based systems, learning, genetic algorithms, ant colony optimization.

## 1 Introduction

An important application of Fuzzy Rule-Based Systems (FRBSs) is the *linguistic fuzzy modeling*, where the interpretability of the obtained model is the main requirement, together with its accuracy. This task is developed by means of linguistic FRBSs, which use fuzzy rules composed of linguistic variables that take values in a term set with a real-world meaning.

The Knowledge Base (KB) of a linguistic FRBS presents two main components: 1) the

Rule Base (RB), constituted by the collection of fuzzy rules, and 2) the Data Base (DB), that contains the membership functions of the fuzzy partitions associated to the linguistic variables. The composition of the KB of an FRBS directly depends on the problem being solved. The task of automatically defining the KB for a concrete application is considered as a hard problem. In the last few years, a large number of methods has been proposed to generate the involved rules (RB) making use of different techniques. Most of them need of the existence of a previous definition for the DB. This operation mode makes the DB have a significant influence on the FRBS performance [7]. For this reason, some approaches try to improve the preliminary DB definition considered once the RB has been derived by means of a tuning process to adjust the membership function parameters. In these cases, the obtained RB remains fixed.

Therefore, it would be desirable a greater degree of cooperation between these two task (RB and DB learning) in order to obtain models with a good interpretability-accuracy trade-off. With this aim, we propose a KB learning approach that evolves DB definitions and looks for compact RBs, working cooperatively to generate linguistic models with good behaviour. To do so, we consider a RB learning method based on the COR methodology together with an specific ant colony optimization-based algorithm, the *best-worst ant system*. For the DB derivation, we use a genetic algorithm that includes its three usual components: number of labels per variable, membership function definitions (ob-

---

<sup>1</sup>This research has been supported by CICYT under project TIC2002-04036-C05-01.

tained from a non-linear scaling function) and scaling factors.

This paper is organized as follows. Section 2 presents the COR-BWAS method. The GA that performs the proposed KB learning process is described in Section 3. Finally, in Section 4, some experimental results are shown and some conclusions are pointed out.

## 2 COR-BWAS learning process

COR-BWAS is an approach to the RB learning problem based on the Cooperative Rules (COR) methodology [3] and making use of ant colony optimization (ACO) algorithms to find a RB with good balance between its two main requirements: *interpretability* (less number of rules) and *accuracy* (better cooperation among rules). The two next subsections present the COR methodology and its integration with the specific ACO algorithm.

### 2.1 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 [3]. 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. [14]) 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 [3]: the COR methodology. Instead of selecting the consequent with the highest performance in each subspace like ad hoc data-driven meth-

ods 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: *search space construction*, where a set of candidate rules is obtained for each fuzzy input subspace and *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 contribution, we will also include an enhance to the original proposal [3] to allow it to eliminate badly defined and conflicting rules with the aim of improving the interpretability and the accuracy [2]. This approach, the fuzzy rule set reduction, is a regular practice in fuzzy modeling usually achieved by genetic algorithms. 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 that 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 to find good solutions increases.

### 2.2 Best-Worst Ant System to Learn Linguistic Fuzzy Rules

ACO algorithms [9] constitute a new family of global search bio-inspired algorithms that has been recently proposed. 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 *pheromone*. Ants have the ability of smelling 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 a probability defined by the amount of 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 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. 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: 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*.

One of these successful approaches is the Best-Worst Ant System (BWAS) model [4]. 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 (considering the global-best and the worst current solutions) or the pheromone trail mutation to introduce diversity in the search.

There are five steps to solve a specific problem by ACO algorithms:

1. *Problem representation*: Interpret the problem to be solved as a graph or a sim-

ilar 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.

COR is characterized by its flexibility to be used with different metaheuristics. In [3], 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 intention thanks to the inherent use of heuristic information. Therefore, we will describe the BWAS model in the COR methodology, presenting the different components of the algorithm according to the five steps previously commented (for a further description, refer to [2]):

**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—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 QAP [11], 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 since one assignment does not restrict the remaining ones, i.e., the assignment order is irrelevant.

**Heuristic Information:** The heuristic information on the potential preference of selecting a specific consequent, in each antecedent combination (subspace) can be found in [2].

**Pheromone Initialization:** 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).

**Fitness Function:** The fitness function will be the Mean Square Error (MSE) over the training data set.

**ACO Scheme:** BWAS algorithm.

### 3 Learning process of the FRBS knowledge base

In this Section, we propose a new process to automatically generate the KB of a FRBS based on a learning approach composed of two methods:

- A genetic learning process for the DB that allows us to define:
  - The number of labels for each linguistic variable.
  - The variable domain (working range).
  - The form of each fuzzy membership function in non-uniform fuzzy partitions, using a non-linear scaling function that defines different areas in the variable working range where the FRBS has a higher or a lower relative sensibility.

- The COR-BWAS learning method that derives the RB considering the DB previously obtained. This method is run from each DB definition generated by the GA, thus allowing the proposed hybrid learning process to finally obtain the whole definition of the KB (DB and RB) by means of the cooperative action of both methods.

All the components of the DB will be adapted throughout a genetic process. Since it is interesting to reduce the dimensionality of the search space for that process, the use of non-linear scaling functions is conditioned by the necessity of using parameterized functions with a reduced number of parameters. We consider the scaling function proposed in [10], that has a single sensibility parameter called  $a$  ( $a \in \mathbb{R}$ ). The function used is  $f : [-1, 1] \rightarrow [-1, 1]$

$$f(x) = \text{sign}(x) \cdot |x|^a, \quad \text{with } a > 0$$

The final result is a value in  $[-1, 1]$  where the parameter  $a$  produces uniform sensibility ( $a = 1$ ), higher sensibility for center values ( $a > 1$ ), or higher sensibility for extreme values ( $a < 1$ ). The range considered for the parameter  $a$  is the interval  $(0, 10]$ . In this paper, triangular membership functions are considered due to their simplicity. So, the non-linear scaling function will only be applied on the three definition points of the membership function. We should note that the previous scaling function is recommended to be used with symmetrical variables since it causes symmetrical effects around the center point of the interval. It can be desirable the possibility of producing higher sensibility in only one of the working range extents. Figure 1 shows a graphical representation of these five shape possibilities. For more details about the fuzzy partition building, refer to [5].

The evolutionary process that guides our approach for automatic learning of the whole KB of an FRBS is presented next:

Genetic Algorithms (GAs) [12] are search and optimization techniques that are based on a

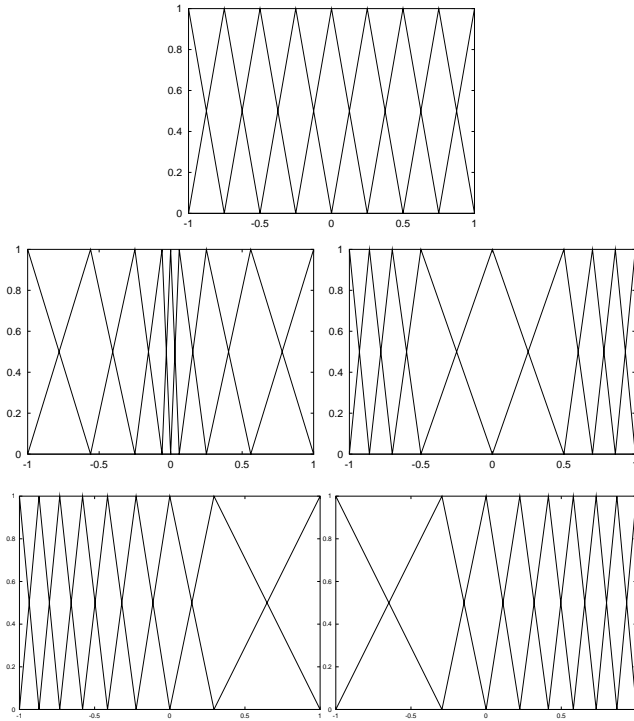


Figure 1: Fuzzy partitions shapes

formalization of natural genetics. The genetic process starts with a population of solutions called chromosomes, that constitutes the first generation ( $G(0)$ ), and undergoes evolution over it. While a certain termination condition is not met, each chromosome is evaluated by means of an evaluation function (a fitness value is assigned to the chromosome) and a new population is created ( $G(t+1)$ ), by applying a set of genetic operators to the individuals of generation  $G(t)$ .

The important questions when using GAs are: how to code each solution (in this case, the DB of an FRBS), how to evaluate these solutions and how to create new solutions from existing ones. Next, we will briefly describe these aspects.

**Encoding the DB.** Each chromosome will be composed of four parts:

- Number of labels ( $C_1$ ): For a system with  $N$  variables (including input and output variables), the number of labels per variable is encoded into an integer array of length  $N$ . In this contribution, the possible values considered are the set

$\{2, \dots, 7\}$ .

- Shape parameters ( $C_2$ ): An integer array of length  $N$ , where the shape of the fuzzy partition  $S$  for each variable is stored. The five possible shapes previously commented are codified with a different number ( $\{0,1,2,3,4\}$ )
- Sensibility parameters ( $C_3$ ): A real array of length  $N$ , where the sensibility parameter ( $a$ ) for each variable is stored. In our case, the range considered for this parameter is the interval  $(0, 10)$ .
- Working ranges ( $C_4$ ): An array of  $N \times 2$  real values stores the variable working range ( $[v_{min}, v_{max}]$ ). If the initial domain of a variable is  $[v_{min}^i, v_{max}^i]$ , and  $d$  is the interval dimension ( $d = v_{max}^i - v_{min}^i$ ), the range considered for the variable domain lower limit is  $[v_{min}^i - (1/4 * d), v_{min}^i]$ , and the range for the upper limit is  $[v_{max}^i, v_{max}^i + (1/4 * d)]$ .

A graphical representation of the chromosome is shown next:

$$C_1 = (l_1, \dots, l_N)$$

$$C_2 = (S_1, \dots, S_N)$$

$$C_3 = (a_1, \dots, a_N)$$

$$C_4 = (r_1^{inf}, r_1^{sup}, \dots, r_N^{inf}, r_N^{sup})$$

$$C = C_1 C_2 C_3 C_4$$

**Evaluating the DB.** There are three steps that must be done to evaluate each chromosome:

- Generate the fuzzy partitions for all the linguistic variables using the information contained in the chromosome. First, each variable is linearly mapped from its working range  $[r_i^{inf}, r_i^{sup}]$ ,  $i = 1, \dots, N$  (fourth part of the chromosome) to  $[-1, 1]$ . Next, uniform fuzzy partitions for all the variables are created considering the number of labels per variable ( $l_i$ ). Finally, the non-linear scaling function with its shape ( $S_i$ ) and its sensibility parameter

$(a_i)$  is applied to the definition points of the membership functions, obtaining the fuzzy partition.

- Generate the RB, by running the COR-BWAS learning method considering the DB obtained in the previous step.
- Calculate the Mean Square Error over the training set using the KB obtained (DB + RB). In order to avoid the possible overfitting, thus improving the generalization capability of the final FRBS, we will lightly penalize FRBSs with a high number of rules ( $\#R$ ) to obtain more compact linguistic models. Therefore, once the RB has been generated and its MSE over the training set has been calculated, the fitness function is calculated as follows:

$$F_C = \omega_1 \cdot MSE + \omega_2 \cdot \#R$$

with  $\omega_1$  and  $\omega_2$  being weighting percentages ( $\omega_1 + \omega_2 = 1$ ). In this contribution we have considered that  $\omega_1 = 0.9$ .

**Genetic operators.** A set of genetic operators is applied to the genetic code of the DB contained in  $G(t)$ , to obtain  $G(t + 1)$ . Due to the special nature of the chromosomes involved in this DB definition process, the design of genetic operators able to deal with it becomes a main task. Since there is a strong relationship among the four chromosome parts, operators working cooperatively in  $C_1$ ,  $C_2$ ,  $C_3$  and  $C_4$  are required in order to make best use of the representation used. The reproduction operator is the Baker's stochastic universal sampling [1].

Two different mutation operators are used, each one of them acting on different chromosome parts. The mutation operator selected for  $C_1$  and  $C_2$  is similar to the one proposed by Thrift in [13]. Since  $C_3$  and  $C_4$  are based on a real-coding scheme, Michalewicz's non-uniform mutation operator is employed [12].

As regards the recombination process, two different crossover operators are considered depending on the two parents' scope. When

both parents have the same values in  $C_1$  and  $C_2$  (each variable has the same shape and equal number of labels in the two parents), then the genetic search has located a promising space zone that has to be adequately exploited. This task is developed by applying the Parent-Centric crossover operator[8] in  $C_3$  and  $C_4$  and obviously by maintaining the parent  $C_1$  and  $C_2$  values in the offspring. If the two parents encode different partition shapes or granularity levels, it is interesting the use of the information encoded by the parents for exploring the search space in order to discover new promising zones. In this way, an standard crossover operator is applied over the four parts of the chromosomes. This operator performs as follows: a crossover point  $p$  is randomly generated in  $C_1$  and the two parents are crossed at the  $p$ -th variable in the four chromosome parts,  $C_1$ ,  $C_2$ ,  $C_3$ , and  $C_4$ , thereby producing two meaningful descendants.

## 4 Experimental study

The proposed method have been applied to one real-world problem. A 5-fold 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<sup>1</sup>. In the GA, six runs with different seeds for the pseudo-random sequence are made for each data partition.

A problem with estimations of minimum maintenance costs which are based on a model of the optimal electrical network for a spanish town [6] will be considered to validate the KB learning process proposed. The problem has four input variables: *Sum of the lengths of all streets in the town*, *Total area of the town*, *Area that is occupied by buildings* and *Energy supply to the town* and one output variable: *Maintenance costs of medium voltage line*. These values are somewhat lower

---

<sup>1</sup>The data sets used in these experiments are available at <http://decsai.ugr.es/~casillas/FMLib/>

than the real ones, but companies are interested in an estimation of the minimum costs. Of course, real maintenance costs are exactly accounted but a model that relates these costs to any characteristic of simulated towns with the optimal installation is important for the electrical companies. We were provided with data concerning four different characteristics of the towns and their minimum maintenance costs in a sample of 1059 simulated towns. In this case, our objective was to relate the last variable (maintenance costs) with the other four ones.

The results obtained for our process (GA+COR-BWAS) will be compared with other two methods: the same process but considering the Wang and Mendel RB generation method[14] instead of COR-BWAS (GA+WM) and the original COR-BWAS learning process.

Table 1 shows the results. The columns named 1 to 5 of the first row represent the five subsets of the 5 fold cross validation. For each column four values are shown, corresponding to the average values of the six runs performed with different seeds. The last three columns show the arithmetic mean ( $\bar{x}$ ) over the 30 runs performed (in bold type); the standard deviation over the five mean 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. While  $\sigma_{\bar{x}_i}$  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 the data partition, while the latter value shows the robustness of the metaheuristic to obtain similar results regardless the followed pseudo-random sequence.

The four average values of each column for the three methods are:

- **MSE<sub>tra</sub>**: MSE over the training set.
- **MSE<sub>tst</sub>**: MSE over the test set.
- **#R**: The number of rules of the RB.

- **#L**: The sum of the number of labels considered in the DB for all the variables.

Since the best result with the original COR-BWAS for the problem were obtained considering five labels for each variable, the row **#L** is the sum of these labels (25) in all the columns.

As can be observed in Table 1, the proposed KB learning process obtains a significative improvement with respect to the other two methods, presenting more accurate prediction levels with lesser values for the MSE in training and test data set and more interpretable models with a smaller fuzzy rule set and a lesser number of linguistic terms. On the other hand, GA+COR-BWAS and GA+WM are slightly less sensitive to the data partition ( $\sigma_{\bar{x}_i}$ ) than the original COR-BWAS while the former ones are more sensitive to the pseudo-random sequence ( $\bar{\sigma}_{x_i}$ ) than the latter one.

The main drawback of GA+COR-BWAS is the long time required by the GA. Each time a new cromosome is evaluated, a complete run of COR-BWAS is performed. Our future work will be focused on adjusting the GA structure and parameters of the KB learning process proposed in order to reduce its high computational time keeping its good performance.

## References

- [1] J.E. Baker, Reducing bias and inefficiency in the selection algorithm, *Proc. Second Int. Conference on Genetic Algorithms (ICGA '87)*, Hillsdale, 1987, pp. 14-21.
- [2] J. Casillas, O. Cordon, I. Fernández de Viana, F. Herrera, Learning Cooperative Linguistic Fuzzy Rules using the Best-Worst Ant System Algorithm, *Technical Report DECSAI-03-01-15*, University of Granada, Spain, 2003.
- [3] J. Casillas, O. Cordon, 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,*

Table 1: Results obtained

		1	2	3	4	5	$\bar{x}$	$\sigma_{\bar{x}_i}$	$\bar{\sigma}_{x_i}$
COR-BWAS	$MSE_{tra}$	38483	39106	39851	40746	40013	<b>39640</b>	966	566
	$MSE_{tst}$	46452	46050	40689	35850	39373	<b>41683</b>	4397	1599
	#R	42.3	40.0	40.5	40.2	42.0	<b>41.0</b>	1.8	1.4
	#L	25	25	25	25	25	<b>25</b>	-	-
GA + COR-BWAS	$MSE_{tra}$	19834	20678	20631	20640	20018	<b>20360</b>	402	1561
	$MSE_{tst}$	25076	24653	20919	23605	18899	<b>22830</b>	2302	3259
	#R	18.3	17.8	15.2	17.7	19.8	<b>17.8</b>	1.7	3.8
	#L	22.3	22.8	20.2	21.5	22.5	<b>21.9</b>	1.1	1.8
GA + WM	$MSE_{tra}$	22765	22682	24147	21697	23778	<b>23014</b>	971	2143
	$MSE_{tst}$	25826	23942	23817	24319	22549	<b>24090</b>	1176	3667
	#R	54.0	44.5	51.0	54.2	52.0	<b>51.1</b>	3.9	6.8
	#L	26.3	25.0	26.0	27.2	26.5	<b>26.2</b>	0.8	1.5

and *Cybernetics—Part B: Cybernetics*, 32(4), 2002, pp. 526-537.

- [4] O. Cordón, F. Herrera, I. Fernández de Viana, L. Moreno, A new ACO model integrating evolutionary computation concepts: the best-worst ant system, *Proceedings of the 2nd International Workshop on Ant Algorithms*, Brussels, 2000, pp. 22-29.
- [5] O. Cordón, F. Herrera, L. Magdalena, P. Villar, A genetic learning process for the scaling factors, granularity and contexts of the fuzzy rule-based system data base, *Information Science* 136, 2001, pp. 85-107.
- [6] O. Cordón, F. Herrera, L. Sánchez, Solving electrical distribution problems using hybrid evolutionary data analysis techniques, *Applied Intelligence* 10, 1999, pp. 5-24.
- [7] O. Cordón, F. Herrera, P. Villar, Analysis and guidelines to obtain a good uniform fuzzy partition granularity for fuzzy rule-based systems using simulated annealing, *International Journal of Approximate Reasoning* 25, 2000, pp. 187-215.
- [8] K. Deb, D. Joshi, A. Anand, Real-Coded Evolutionary Algorithms with Parent-Centric Recombination, *Proc. of the 2002 congress on evolutionary computation*, Piscataway, IEEE press, 2002, pp. 61-66.
- [9] M. Dorigo, T. Stützle, The ant colony optimization metaheuristic: algorithms, applications, and advances, in F. Glover and G. A. Kochenberger, editors, *Handbook of metaheuristics*, Kluwer Academic Publishers, 2003, pp. 251-285.
- [10] L. Magdalena, Adapting the gain of an FLC with genetic algorithms, *International Journal of Approximate Reasoning* 17(4), 1997, pp. 327-349.
- [11] V. Maniezzo, A. Colorni, The ant system applied to the quadratic assignment problem, *IEEE Transactions on Knowledge and Data Engineering*, 11(5), 1999, pp. 769-778.
- [12] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs* (Springer-Verlag, 1996).
- [13] P. Thrift, Fuzzy logic synthesis with genetic algorithms. Proceedings of Fourth International Conference on Genetic Algorithms (ICGA'91), 1991, pp. 509-513.
- [14] L.-X. Wang, J.M. Mendel, Generating fuzzy rules by learning from examples, *IEEE Transactions on Systems, Man, and Cybernetics*, 22(6), 1992, pp. 1414-1427.