Cooperative Coevolution for Learning Fuzzy Rule-Based Systems*

Jorge Casillas[1], O. Cordón[1], F. Herrera[1], and J.J. Merelo[2]

[1] Depto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Granada, E-18071 Granada, España
{casillas,ocordon,herrera}@decsai.ugr.es,
[2] GeNeura Team, Depto. Arquitectura y Tecnología de Computadores
Universidad de Granada, E-18071 Granada, España
jmerelo@geneura.ugr.es

Abstract. In the last few years, the coevolutionary paradigm has shown an increasing interest thanks to its high ability to manage huge search spaces. Particularly, the cooperative interaction scheme is recommendable when the problem solution may be decomposable in subcomponents and there are strong interdependencies among them.
The paper introduces a novel application of these algorithms to the learning of fuzzy rule-based systems for system modeling. Traditionally, this process is performed by sequentially designing their different components. However, we propose to accomplish a simultaneous learning process with cooperative coevolution to properly consider the tight relation among the components, thus obtaining more accurate models.

1 Introduction

Fuzzy rule-based systems (FRBSs) constitute an extension of classical rule-based systems, because they deal with IF-THEN rules where antecedents and/or consequents are composed of fuzzy logic statements, instead of classical logic rules. This consideration presents two essential advantages: the key features of knowledge captured by fuzzy sets involve handling uncertainty and inference methods become more robust and flexible with approximate reasoning methods of fuzzy logic. One of the most success applications of FRBSs is *system modeling* [17], 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 [23].

Several tasks have to be performed in order to design an FRBS for a concrete modeling application. One of the most important and difficult ones is to *derive* an appropriate knowledge base (KB) about the problem being solved. The KB stores the available knowledge in the form of fuzzy IF-THEN rules. It consists of the rule base (RB), comprised of the collection of rules in their symbolic forms, and the data base (DB), which contains the linguistic term sets and the membership functions defining their meanings.

* Partially supported by the Spanish CICYT, project PB98-1319

P. Collet et al. (Eds.): EA 2001, LNCS 2310, pp. 311–322, 2002.

Numerous automatic methods — based on ad hoc data-driven approaches [25] or on different techniques such as neural networks [15] or genetic algorithms (GAs) [4,20,21] — have been developed to perform the derivation task. When only the derivation of the RB is addressed, methods generally operate in only one stage [24,25]. In this case, the DB is usually obtained from the expert information (if it is available) or by a normalization process.

However, methods that design both RB and DB are preferable since the automation is higher. In this case, we can distinguish between two different approaches:

– *Simultaneous derivation*: It relates to the process of directly obtaining the whole KB (RB and DB) from the available data in a simultaneous way [12, 13]. This task is usually known as *learning process*.
– *Sequential derivation*: The task is divided into two or more stages, each of them performing a partial or complete derivation of the KB.

  Some methods learn the DB with a embedded approach [6,8] that may be used as one of the first stages.

  Generally, one of the last stages adjusts the previously learnt/obtained DB with slight modifications to increase the system performance [1,10,11]. This stage is known as *tuning process*.

  In most cases, a sequential process by firstly learning the RB and then tuning the DB is considered [3].

When the RB and the DB are simultaneously derived, the strong dependency of both components is properly addressed. However, the derivation process becomes significantly more complex because the search space grows and the selection of an appropriate search technique is crucial.

Recently, the coevolutionary paradigm [16] has shown an increasing interest thanks to its high ability to manage with huge search spaces and decomposable problems. The direct decomposition of the KB derivation process (thus obtaining two interdependent components, learning of the RB and DB) makes coevolutionary algorithms with a cooperative approach [19] very useful for this purpose.

In this paper, we propose a KB derivation method within this novel evolutionary paradigm. Actually, a method has been already proposed by Peña-Reyes and Sipper with this cooperative coevolutionary philosophy [18]. However, opposite to it, our proposal performs a more sophisticated learning of the RB based on the Cooperative Rules (COR) methodology [2], whose good performance is related to the consideration of cooperation among rules. Once the rule antecedents (defining fuzzy subspaces) have been obtained, COR generates a candidate consequent set for each subspace and searches the consequents with the best global performance.

In the following sections, an introduction to coevolutionary algorithms, the proposed KB derivation method, some experimental results, conclusions, and further work are shown.

2 Coevolutionary Algorithms

Evolutionary algorithms (EAs) [14] are general-purpose global search algorithms that use principles inspired by natural population genetics. In a EA, each individual in the population represents a candidate solution to the problem and has an associated *fitness* to determine which individuals are used to form new ones in the process of competition. The new individuals are created using genetic operators such as crossover and mutation.

Within this field, a new paradigm has been recently proposed, coevolutionary algorithms [16]. They involve two or more species (populations) that permanently interact among them by a coupled fitness. Thereby, in spite of each species has its own coding scheme and reproduction operators, when an individual must be evaluated, its goodness will be calculated considering some individuals of the other species. This coevolution makes easier to find solutions to complex problems.

Different kinds of interactions may be considered among the species according to the dependencies existing among the solution subcomponents. Generally, we can mention two different kinds of interaction:

- *Competitive coevolutionary algorithms* [22]: Those where each species competes with the remainder. In this case, increasing the fitness of an individual in a species implies decreasing the fitness of the ones other species, i.e., the success of somebody else entails the personal failure.
- *Cooperative* or *symbiotic coevolutionary algorithms* [19]: Those where all the species cooperate to build the problem solution. In this case, the fitness of an individual depends on its ability to cooperate with individuals from other species.

  Figure 1 illustrates the cooperative approach. As shown, a set of selected individuals (called cooperators) is built in each species to represent it. Each individual is evaluated constructing solutions with it and cooperators of the remaining species.

Therefore, the use of cooperative coevolutionary algorithms is recommendable when the following issues arise [18]:

1. the search space is huge,
2. the problem may be decomposable in subcomponents,
3. different coding schemes are used, and
4. there is strong interdependencies among the subcomponents.

They also arise in problems where the training set is not known in advance, but created by the solution to the problem themselves, e.g., when collision avoidance behavior for two planes is being evolved simultaneously [7]. In that cases, training sets are created by the other planes which are being evolved.

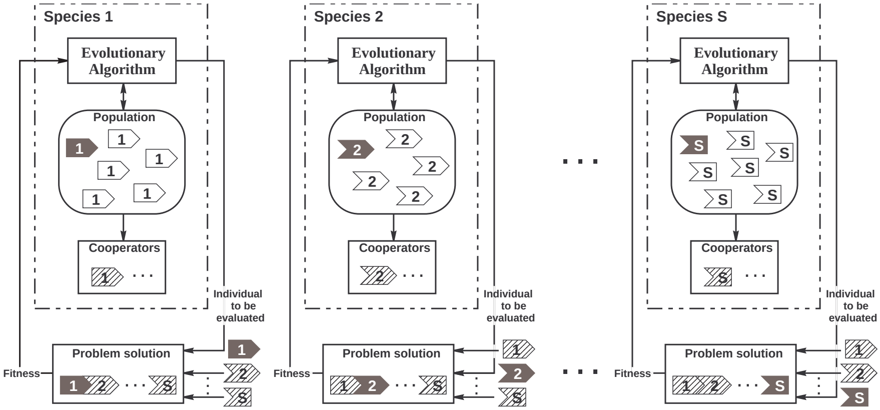


Fig. 1. Cooperative coevolutionary scheme

3 A Cooperative Coevolutionary Algorithm for Jointly Learning Fuzzy Rule Bases and Membership Functions

Intuitively, we may decompose the problem of deriving a proper KB for an FRBS into two subtasks: to obtain fuzzy rule symbolic representations (learning the RB) and to define membership function shapes (learning the DB). Therefore, our coevolutionary algorithm consists of two species that cooperate to build the whole solution.

In the following subsections, a formulation for both learning tasks and the components of the cooperative coevolutionary algorithm are introduced.

3.1 The Knowledge Base Derivation Process

Learning Fuzzy Rule Bases

The RB learning task is based on the COR methodology [2]. Let E be the input-output data set, $e_l = (x_1^l, \ldots, x_n^l, y^l)$ on of its elements (example), and n be the number of input variables. Let $\mathcal{A}_i$ be the set of linguistic terms of the i-th input variable and $\mathcal{B}$ the set of linguistic terms of the output variable. Its operation mode is the following:

1. Define a set of fuzzy input subspaces, $\{S_s \mid s \in \{1, \ldots, N_S\}\}$, with the antecedent combinations containing at least a positive example, i.e., $S_s = (A_1^s, \ldots, A_i^s, \ldots, A_n^s) \in \mathcal{A}_1 \times \ldots \times \mathcal{A}_n$ such that $E'_s \neq \emptyset$ (with A_i^s being a label of the i-th input variable, E'_s being the set of positive examples of the subspace S_s, and N_S the number of subspaces with positive examples). In this contribution, we will define the set of positive examples for the subspace S_s as follows:

$$E'_s = \{e_l \in E \;\mid\; \forall i \in \{1, \ldots, n\},$$
$$\forall A_{ij} \in \mathcal{A}_i,\; \mu_{A_i^s}(x_i^l) \geq \mu_{A_{ij}}(x_i^l)\} \;,$$

with A_{ij} being a label of the i-th input variable and μ_T the membership function of the label T.

2. For each subspace S_s, obtain a set of candidate consequents (i.e., linguistic terms of the output variable) $\mathbf{B^s}$ to build the corresponding fuzzy rule.
   In this contribution, we will define the set of candidate consequents for the subspace S_s as follows:

$$\mathbf{B^s} = \{B_k \in \mathcal{B} \mid \exists e_{l^s} \in E'_s \ \textit{where} \\ \forall B_l \in \mathcal{B},\ \mu_{B_k}(y^{l^s}) \geq \mu_{B_l}(y^{l^s})\}\ ,$$

   with B_k being a label of the output variable.
3. Perform a *combinatorial search* among these sets looking for the combination of consequents (one for each subspace) with the best global accuracy.

For example, from the subspace $S_s = (high, low)$ and the candidate consequent set in such a subspace $\mathbf{B^s} = \{small, medium, large\}$, we will obtain the fuzzy rule:

$$R_s = \textsf{IF } X_1 \textsf{ is } \textsf{\textit{high}} \textsf{ and } X_2 \textsf{ is } \textsf{\textit{low}} \ \textbf{\textsf{THEN}}\ Y \textsf{ is } B_s,$$

with $B_s \in \mathbf{B^s}$ being the label selected by the combinatorial search to represent to the subspace S_s.

Learning Fuzzy Membership Functions

In our case, the derivation of the DB involves determining the shape of each membership function. These shapes will have a high influence in the FRBS performance. In this contribution, we will consider triangular-shaped membership functions as follows:

$$\mu_T(x) = \begin{cases} \frac{x-a}{b-a}, \text{ if } a \leq x < b \\ \frac{c-x}{c-b}, \text{ if } b \leq x \leq c \\ 0, \quad \text{otherwise} \end{cases} ,$$

Therefore, different values of the parameters a, b, c will define different shapes of the membership function associated to the linguistic term T.

3.2 The Cooperative Coevolutionary Algorithm

Cooperative Interaction Scheme between Both Species

Let F_{ij} be the FRBS obtained by composing the subcomponents encoded in the chromosomes i and j of the species 1 (RBs) and 2 (membership functions), respectively. The objective will be to minimize the well-known *mean square error* (MSE):

$$\text{MSE}_{ij} = \frac{1}{2 \cdot N} \sum_{l=1}^{N} \left(F_{ij}(x^l) - y^l\right)^2,$$

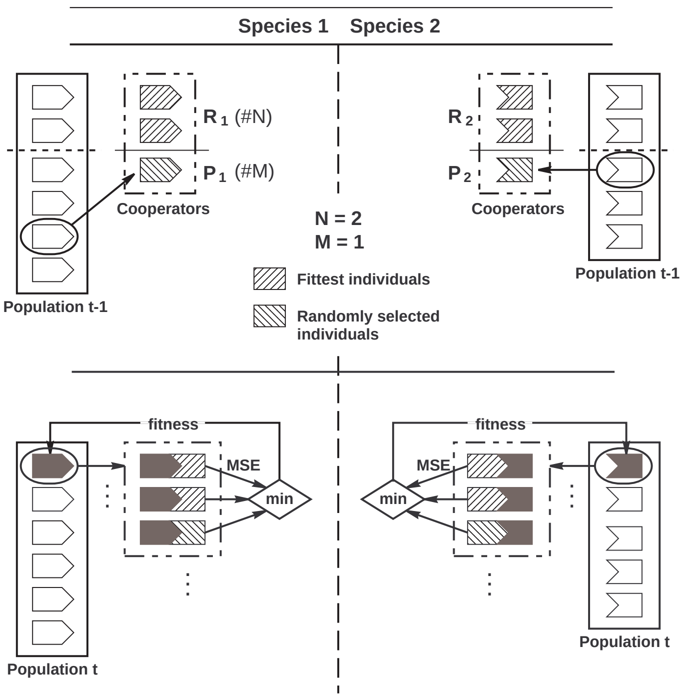


Fig. 2. Interaction scheme considered in the learning method

with N being the data set size, $F_{ij}(x^l)$ being the output obtained from the designed FRBS when the l-th example is considered, and y^l being the known desired output.

Each individual of species 1 or 2 is evaluated with the corresponding **fitness function** f_1 or f_2, which are defined as follows:

$$f_1(i) = \min_{j \in R_2 \cup P_2} \mathrm{MSE}_{ij}$$

$$f_2(j) = \min_{i \in R_1 \cup P_1} \mathrm{MSE}_{ij}$$

with i and j being individuals of species 1 and 2 respectively, R_1 and R_2 being the set of the fittest individuals in the previous generation of the species 1 and 2 respectively, and P_1 and P_2 being individual sets selected at random from the previous generation of the species 1 and 2 respectively. Figure 2 graphically shows the proposed interaction scheme.

Whilst the sets $R_{1|2}$ allow the best individuals to influence in the process guiding the search towards good solutions, the sets $P_{1|2}$ introduce diversity in the search. The combined use of both kinds of sets makes the algorithm have a trade-off between exploitation ($R_{1|2}$) and exploration ($P_{1|2}$). The cardinalities of the sets $R_{1|2}$ and $P_{1|2}$ are previously defined by the designer.

A **generational** [14] scheme is followed in both species. Baker's stochastic universal sampling procedure together with an elitist mechanism (that ensures to select the best individual of the previous generation) are used.

The specific operators used in every species are described in the following sections.

Species 1: Learning Fuzzy Rule Bases

An integer-valued vector (c) of size N_S (number of subspaces with positive examples) is employed as **coding scheme**. Each cell of the vector represents the index of the consequent used to build the rule in the corresponding subspace:

$$\forall s \in \{1, \ldots, N_S\},\ c[s] = k_s \ s.t.\ B_{k_s} \in \mathbf{B^s}.$$

The **initial pool** of this species is generated building the first individual as follows

$$\forall s \in \{1, \ldots, N_S\},$$
$$c_1[s] = \underset{k_s}{arg} \max_{B_{k_s} \in \mathbf{B^s}} CV(R^s_{k^s}),$$

with

$$CV(R^s_{k^s}) =$$
$$\max_{e_{l^s} \in E'_s} Min\left(\mu_{A^s_1}(x^{l^s}_1), \ldots, \mu_{A^s_n}(x^{l^s}_n), \mu_{B_{k^s}}(y^{l^s})\right),$$

and the remaining chromosomes generated at random:

$$\forall p \in \{2, \ldots, pool\_size\},\ \forall s \in \{1, \ldots, N_S\},$$
$$c_p[s] = \ some\ k_s\ s.t.\ B_{k_s} \in \mathbf{B^s}.$$

The standard two-point **crossover operator** is used. The **mutation operator** randomly selects a specific $s \in \{1, \ldots, N_S\}$ where $|\mathbf{B^s}| \geq 2$, and changes at random $c[s] = k^s$ by $c[s] = k^{s\prime}$ such that $B_{k^{s\prime}} \in \mathbf{B^s}$ and $k^{s\prime} \neq k^s$.

Species 2: Learning Fuzzy Membership Functions

As **coding scheme**, a 3-tuple of real values for each triangular membership function is used, thus being the DB encoded into a real-coded chromosome built by joining the membership functions involved in each variable fuzzy partition. A variation interval to every gene is associated to preserve meaningful fuzzy sets.

The **initial population** of this species is generated with a chromosome representing the original DB and the remaining chromosomes generated with the values at random within the corresponding variation interval.

The max-min-arithmetical **crossover** operator [10] is considered. If $C^t_v = (c_1, \ldots, c_k, \ldots, c_H)$ and $C^t_w = (c'_1, \ldots, c'_k, \ldots, c'_H)$ are to be crossed, the following four offspring are generated:

$$C^{t+1}_1 = aC^t_w + (1-a)C^t_v, \qquad C^{t+1}_2 = aC^t_v + (1-a)C^t_w,$$
$$C^{t+1}_3 \ \ with\ \ c^{t+1}_{3,k} = \min\{c_k, c'_k\}, \qquad C^{t+1}_4 \ \ with\ \ c^{t+1}_{4,k} = \max\{c_k, c'_k\}.$$

The parameter a is defined by the designer. The resulting descendents are the two best of the four aforesaid offspring. As may be observed, its formulation avoids the violation of the restrictions imposed by the variation intervals.

With respect to the **mutation operator**, it simply involves changing the value of the selected gene by other value obtained at random within the corresponding variation interval.

4 Experimental Results in the Electrical Maintenance Cost Estimating Problem

This experimental study will be devoted to analyze the behavior of the proposed derivation method — jointly learning the RB following the COR methodology and the membership functions — with cooperative coevolutionary algorithms (CORMF-CC). With this aim, we have chosen the problem of estimating the maintenance costs of the medium voltage electrical network in a town [5].

We will analyze the accuracy of the fuzzy models generated from the proposed process compared to the four following methods: the well-known ad hoc data-driven method proposed by Wang and Mendel (WM) [25]; a GA-based learning method following the COR methodology (COR-GA) [2]; and two sequential methods, WM+Tun and COR-GA+Tun, that firstly perform a learning of the RB with WM or COR-GA, respectively, and then adjust the membership functions with the tuning method proposed in [3].

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.

4.1 Problem Description

Estimating the maintenance costs of the medium voltage electrical network in a town [5] is a complex but interesting problem. Since an actual measure is very difficult to obtain, the consideration of models becomes useful. These estimations allow electrical companies to justify their expenses. Moreover, the model must be able to explain how a specific value is computed for a certain town. Our objective will be to relate the *maintenance costs of medium voltage line* with the following four variables: *sum of the lengths of all streets in the town*, *total area of the town*, *area that is occupied by buildings*, and *energy supply to the town*. We will deal with estimations of minimum maintenance costs based on a model of the optimal electrical network for a town in a sample of 1,059 towns.

To develop the different experiments in this contribution, the sample has been randomly divided in two subsets, the training and test ones, with an 80%-20% of the original size respectively. Thus, the training set contains 847 elements, whilst the test one is composed by 212 elements. *Five linguistic terms* for each variable are considered.

4.2 Experimental Results and Analysis

The following values have been considered for the parameters of each method:

- *COR-GA*: 61 individuals, 50 generations, 0.6 as crossover probability, and 0.2 as mutation probability.
- *Tuning stage of the WM+Tun and COR-GA+Tun methods*: 61 individuals, 300 generations, 0.6 as crossover probability, 0.2 as mutation probability, 0.35 for the weight factor in the max-min-arithmetical crossover, and 5 for the weight factor in the non-uniform mutation.
- *CORMF-CC*: 62 individuals (31 for each species), 300 generations, 0.6 and 0.2 for the crossover and mutation probabilities in both species respectively, 0.35 for the weight factor of the crossover operator in the species 2, the two fittest individuals ($|R_{1|2}| = 2$) and two random individuals ($|P_{1|2}|$=2) of each species are considered for the coupled fitness.

Ten different runs were performed for each probabilistic algorithm. The results obtained by the five methods analyzed are collected in Table 1, where MSE_{tra} and MSE_{tst} respectively stand for the error obtained over the training and test data sets. Arithmetic mean ($\bar{x}$) and standard deviation (σ) values of the 10 linguistic models generated by each method are included. The best mean results are shown in boldface. A total of 66 fuzzy rules were obtained in all cases.

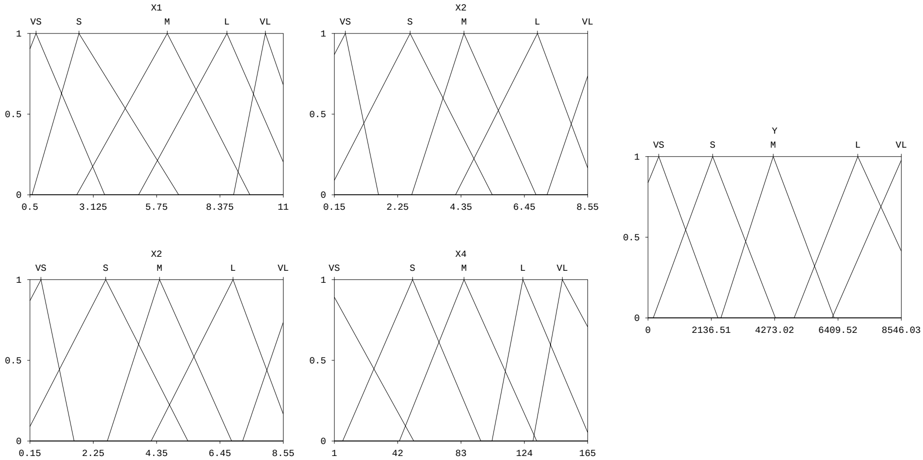
Table 1. Results obtained in the electrical problem

	$\textbf{MSE}_{tra}$		$\textbf{MSE}_{tst}$	
Method	$\bar{x}$	σ	$\bar{x}$	σ
WM	71,294	0	80,934	0
COR-GA	67,237	0	69,457	0
WM+Tun	24,667	1,350	34,143	2,452
COR-GA+Tun	24,255	1,349	31,393	2,831
CORMF-CC	**15,435**	1,094	**22,573**	1,557

In view of the obtained results, the CORMF-CC method shows the best performance combining both approximation (MSE_{tra}) and generalization (MSE_{tst}). Analyzing the two-stage methods (WM+Tun and COR-GA+Tun), we may observe how the tuning process significantly improve the accuracy degrees of the fuzzy models generated by the WM and COR-GA learning methods. However, when the derivation process is made in only one stage with the cooperative coevolutionary approach, the fuzzy model obtained overcomes the remainder thanks to the proper consideration of the dependency between the RB and the DB in the learning process. Moreover, the low standard deviations obtained show the robustness of the CORMF-CC algorithm.

Figure 3 illustrates the DB derived by the CORMF-CC method. Using the shown membership function shapes a good interpretability is kept up whilst the fuzzy model performance is improved.

(a) DB



(b) RB

Rule	X_1	X_2	X_3	X_4	Y	*Rule*	X_1	X_2	X_3	X_4	Y	*Rule*	X_1	X_2	X_3	X_4	Y
R_1	VS	VS	VS	VS	VS	R_{23}	M	S	VS	S	S	R_{45}	L	L	L	S	L
R_2	VS	VS	VS	S	S	R_{24}	M	S	VS	M	S	R_{46}	L	L	L	M	L
R_3	VS	S	S	VS	S	R_{25}	M	M	M	S	M	R_{47}	L	L	L	L	VL
R_4	VS	S	S	S	S	R_{26}	M	M	M	M	L	R_{48}	L	L	M	S	M
R_5	VS	S	VS	VS	VS	R_{27}	M	M	S	S	S	R_{49}	L	L	M	M	M
R_6	VS	S	VS	S	S	R_{28}	M	M	S	M	M	R_{50}	L	L	M	L	L
R_7	S	VS	VS	VS	VS	R_{29}	M	M	S	VS	S	R_{51}	L	L	M	VS	M
R_8	S	VS	VS	S	S	R_{30}	M	L	L	S	M	R_{52}	L	VL	VL	S	VL
R_9	S	VS	S	VS	VS	R_{31}	M	L	L	M	L	R_{53}	L	VL	VL	M	VL
R_{10}	S	VS	S	S	S	R_{32}	M	L	M	S	M	R_{54}	L	VL	VL	L	VS
R_{11}	S	S	S	VS	S	R_{33}	M	L	M	M	M	R_{55}	L	VL	L	S	M
R_{12}	S	S	S	S	M	R_{34}	L	S	S	VS	S	R_{56}	L	VL	L	M	L
R_{13}	S	S	VS	VS	VS	R_{35}	L	S	S	S	M	R_{57}	L	VL	L	L	VL
R_{14}	S	S	VS	S	S	R_{36}	L	S	S	M	M	R_{58}	VL	S	M	S	M
R_{15}	S	M	M	VS	M	R_{37}	L	S	S	L	M	R_{59}	VL	S	M	L	M
R_{16}	S	M	M	S	M	R_{38}	L	M	M	S	M	R_{60}	VL	S	M	VL	L
R_{17}	S	M	S	VS	S	R_{39}	L	M	M	M	L	R_{61}	VL	S	S	S	S
R_{18}	S	M	S	S	S	R_{40}	L	M	M	L	L	R_{62}	VL	S	S	L	M
R_{19}	M	S	S	VS	S	R_{41}	L	M	S	VS	S	R_{63}	VL	S	S	VL	L
R_{20}	M	S	S	S	S	R_{42}	L	M	S	S	M	R_{64}	VL	L	M	S	M
R_{21}	M	S	S	M	M	R_{43}	L	M	S	M	M	R_{65}	VL	L	M	L	L
R_{22}	M	S	VS	VS	VS	R_{44}	L	M	S	L	M	R_{66}	VL	L	M	VL	VL

Fig. 3. KB derived by the CORMF-CC method, where VS stands for *very small*, S for *small*, M for *medium*, L for large, and VL for *very large*

5 Concluding Remarks and Further Work

A KB derivation method that jointly learns the fuzzy rules and membership functions involved in an FRBS has been proposed. The fact of performing these tasks together allows the method to consider the tight relation between both components, thus obtaining better fuzzy models. However, this joint consideration becomes more difficult since the search space is significantly increased, thus being crucial the selection of a proper technique.

As David Goldberg stated, the integration of single methods into hybrid intelligent systems goes beyond simple combinations. For him, the future of Computational Intelligence "*lies in the careful integration of the best constituent technologies*" and subtle integration of the abstraction power of fuzzy systems and the innovating power of genetic systems requires a design sophistication that goes further than putting everything together [9].

In this contribution, this issue is addressed by using a cooperative coevolutionary approach with a sophisticated rule learning component based on the cooperation among the fuzzy rules derived. The good performance of the method compared with other classical hybridizations has been shown when solving a real-world problem. Nevertheless, the proposed modeling approach can be applied to other system modeling problems.

As further work, we propose to extend the components of the KB to be derived (number of labels, more flexible fuzzy rules, etc.), to consider other metaheuristics to adapt each species, and to improve the interaction scheme for a better interdependency consideration and scalability to more than two species.

References

1. P. P. Bonissone, P. S. Khedkar, and Y. Chen. Genetic algorithms for automated tuning of fuzzy controllers: a transportation application. In *Proceedings of the 5th IEEE International Conference on Fuzzy Systems*, pages 674–680, New Orleans, LA, USA, 1996.
2. J. Casillas, O. Cordón, and F. Herrera. Different approaches to induce cooperation in fuzzy linguistic models under the COR methodology. In B. Bouchon-Meunier, J. Gutiérrez-Ríos, L. Magdalena, and R. R. Yager, editors, *Techniques for constructing intelligent systems*. Springer-Verlag, Heidelberg, Germany, 2001. In press.
3. O. Cordón and F. Herrera. A three-stage evolutionary process for learning descriptive and approximate fuzzy logic controller knowledge bases from examples. *International Journal of Approximate Reasoning*, 17(4):369–407, 1997.
4. O. Cordón, F. Herrera, F. Hoffmann, and L. Magdalena. *Genetic fuzzy systems: evolutionary tuning and learning of fuzzy knowledge bases*. World Scientific, Singapore, Singapore, 2001.
5. O. Cordón, F. Herrera, and L. Sánchez. Solving electrical distribution problems using hybrid evolutionary data analysis techniques. *Applied Intelligence*, 10(1):5–24, 1999.
6. O. Cordón, F. Herrera, and P. Villar. Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base. To appear in *IEEE Transactions on Fuzzy Systems*. Draft available at `http://decsai.ugr.es/~herrera/`.
7. N. Durand, J.-M. Alliot, and F. Medioni. Neural nets trained by genetic algorithms for collision avoidance. *Applied Intelligence*, 13(3):205–213, 2000.

8. P. Glorennec. Constrained optimization of FIS using an evolutionary method. In F. Herrera and J. L. Verdegay, editors, *Genetic algorithms and soft computing*, pages 349–368. Physica-Verlag, Heidelberg, Germany, 1996.
9. D. E. Goldberg. A meditation on the computational intelligence and its future. Technical Report Illigal 2000019, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, Illinois, IL, USA, 2000. Available at `http://www-illigal.ge.uiuc.edu/`.
10. F. Herrera, M. Lozano, and J. L. Verdegay. Tuning fuzzy controllers by genetic algorithms. *International Journal of Approximate Reasoning*, 12:299–315, 1995.
11. C. L. Karr. Genetic algorithms for fuzzy controllers. *AI Expert*, 6(2):26–33, 1991.
12. K. KrishnaKumar and A. Satyadas. GA-optimized fuzzy controller for spacecraft attitude control. In J. Periaux, G. Winter, M. Galán, and P. Cuesta, editors, *Genetic algorithms in engineering and computer science*, pages 305–320. John Wiley & Sons, New York, NY, USA, 1995.
13. L. Magdalena and F. Monasterio-Huelin. A fuzzy logic controller with learning through the evolution of its knowledge base. *International Journal of Approximate Reasoning*, 16(3):335–358, 1997.
14. Z. Michalewicz. *Genetic algorithms + data structures = evolution programs.* Springer-Verlag, Heidelberg, Germany, 3rd edition, 1996.
15. D. Nauck, F. Klawonn, and R. Kruse. *Fundations of neuro-fuzzy systems.* John Wiley & Sons, New York, NY, USA, 1997.
16. J. Paredis. Coevolutionary computation. *Artificial Life*, 2:355–375, 1995.
17. W. Pedrycz, editor. *Fuzzy modelling: paradigms and practice.* Kluwer Academic, Norwell, MA, USA, 1996.
18. C. A. Peña-Reyes and M. Sipper. Fuzzy CoCo: a cooperative coevolutionary approach to fuzzy modeling. To appear in *IEEE Transactions on Fuzzy Systems.* Draft version available at `http://lslwww.epfl.ch/~penha/`.
19. M. A. Potter and K. A. De Jong. Cooperative coevolution: an architecture for evolving coadapted subcomponents. *Evolutionary Computation*, 8(1):1–29, 2000.
20. V. M. Rivas, J. J. Merelo, I. Rojas, G. Romero, P. A. Castillo, and J. Carpio. Evolving 2-dimensional fuzzy logic controllers. To appear in *Fuzzy Sets and Systems*, 2001.
21. I. Rojas, J. J. Merelo, J. L. Bernier, and A. Prieto. A new approach to fuzzy controller designing and coding via genetic algorithms. In *Proceedings of the 6th IEEE International Conference on Fuzzy Systems*, 1997.
22. C. D. Rosin and R. K. Belew. New methods for competitive coevolution. *Evolutionary Computation*, 5(1):1–29, 1997.
23. M. Sugeno and T. Yasukawa. A fuzzy-logic-based approach to qualitative modeling. *IEEE Transactions on Fuzzy Systems*, 1(1):7–31, 1993.
24. P. Thrift. Fuzzy logic synthesis with genetic algorithms. In R. K. Belew and L. B. Booker, editors, *Proceedings of the 4th International Conference on Genetic Algorithms*, pages 509–513, San Mateo, CA, USA, 1991. Morgan Kaufmann Publishers.
25. L.-X. Wang and J. M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(6):1414–1427, 1992.