Increasing Fuzzy Rules Cooperation Based on Evolutionary Adaptive Inference Systems

Jesús Alcalá-Fdez,[1,†] Francisco Herrera,[1,*] Francisco Márquez,[2,‡]
Antonio Peregrín[2,§]
[1]*Department of Computer Science and Artificial Intelligence, University of Granada, 18071—Granada, Spain*
[2]*Department of Electronic Engineering and Automatics, University of Huelva, 21071—Huelva, Spain*

This article presents a study on the use of parametrized operators in the Inference System of linguistic fuzzy systems adapted by evolutionary algorithms, for achieving better cooperation among fuzzy rules. This approach produces a kind of rule cooperation by means of the inference system, increasing the accuracy of the fuzzy system without losing its interpretability. We study the different alternatives for introducing parameters in the Inference System and analyze their interpretation and how they affect the rest of the components of the fuzzy system. We take into account three applications in order to analyze their accuracy in practice.

1. INTRODUCTION

Two contradictory requirements are usually found in Fuzzy Modeling (FM) design: *interpretability* and *accuracy*. *Interpretability* is the capability to express the behavior of the real system in an understandable way. *Accuracy* is the capability to represent faithfully the real system. In practice, depending on the application details, one of the two properties normally prevails over the other, with higher interpretability with lower accuracy or lower interpretability with higher accuracy. Designers try to find a trade-off between the two edges, producing an increasing interest in the study of the aforementioned trade-off between interpretability and accuracy in FM.[1,2]

*Author to whom all correspondence should be addressed: e-mail: herrera@decsai.ugr.es.
†e-mail: jalcala@decsai.ugr.es.
‡e-mail: alfredo.marquez@diesia.uhu.es.
§e-mail: peregrin@diesia.uhu.es.

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 22, 1035–1064 (2007)
 Published online in Wiley InterScience
(www.interscience.wiley.com). • DOI 10.1002/int.20237

To develop more *reliable* linguistic fuzzy models, designers may choose between some mechanism's or design's aspects: membership function shape tuning, the integration of the Knowledge Base (KB) design into the whole fuzzy system derivation, rule base reduction, the extension of the model structure (by using linguistic modifiers, double-consequent rules, weighted rules, or hierarchical KBs), or tuning of the fuzzy model components.

We can distinguish between the KB elements and the remaining components. Regarding the latter ones, two components can be considered, the Inference System and the Defuzzification Methods. Difference studies have been developed analyzing their influence in FM (see Refs. 3 and 4, among others). In particular, the tuning of these components can be considered for getting more accurate linguistic fuzzy models while maintaining interpretability.

The Defuzzification Interface is the most typically tuned component.[5–10] Furthermore, there are some studies devoted to adaptive defuzzification methods tuning with evolutionary algorithms.[11–14] In Ref. 15, a study on the different evolutionary adaptive defuzzification approaches in linguistic fuzzy modeling is presented. On the other hand, the parametrization of the Inference System has also produced interest, some contributions about which can be found in Refs. 13 and 16–21.

This contribution develops a study into the use of parametrized expressions in the Inference System, sometimes called Adaptive Inference Systems, for getting more cooperation among the fuzzy rules and therefore more accurate fuzzy models without losing the interpretability. The different possibilities are shown and we also propose to tune the parameters with evolutionary algorithms. We call them *Evolutionary Adaptive Inference Systems (EAIS)*.

We analyze the behavior of the EAIS using two rule base learning methods. The well-known and simple rule learning method, the WM learning method,[22] for analyzing their behavior under a simple fuzzy rule base, and the evolutionary learning method of Thrift,[23] for analyzing their behavior under a more accurate rule base. Of course, the effect of introducing parameters into the inference system may have some bearing on other components of the fuzzy systems. The tuning of membership functions is one of the most important approaches for improving the fuzzy system's accuracy. In this article, we analyze the cooperation between the use of parametric conjunction operators and the tuning of the membership functions for getting a more accurate fuzzy model. The experiments show the positive cooperation between these two tuning approaches. Additionally, we examine the influence of the granularity in the fuzzy partitions and their use with the EAIS and the tuning of membership functions.

To achieve that, the article is organized as follows. Section 2 introduces the Adaptive Inference System, its components, and their meaning in conjunction with defuzzification methods. Section 3 is devoted to the basic study of the EAIS, analyzing its behavior with the three applications. Section 4 studies the combination of the EAIS with the tuning of the membership functions, and analyzes what happens when we increase the granularity of the fuzzy partitions. Finally, Section 5 presents some concluding remarks. An Appendix is also included presenting the description of the applications.

2. ADAPTIVE INFERENCE SYSTEM

In this section, the Adaptive Inference System will be analyzed. To do so, the two Inference System components are presented, and after that, we show how to introduce parameters inside the inference system components. Finally, the meaning of the Adaptive Inference System in conjunction with the defuzzification methods will be studied, which is an important part of this work.

2.1. Inference Components

In this contribution, we are considering linguistic-type IF–THEN rules of the following form:

$$R_i\text{: If } X_{i1} \text{ is } A_{i1} \text{ and} \ldots \text{and } X_{in} \text{ is } A_{in}, \text{ then } Y \text{ is } B_i$$

with $i = 1$ to M, and with X_{i1} to X_{in} and Y being the input and output variables, respectively, and with A_{i1} to A_{in} and B being the involved antecedents and consequent labels, respectively, of the rules.

The expression of the Compositional Rule of Inference in FM with punctual fuzzification is the following:

$$\mu_{\mathrm{B}'}(y) = I(C(\mu_{A1}(x_1), \ldots, \mu_{An}(x_n)), \mu_B(y))$$

where $\mu_{B'}(\cdot)$ is the membership function of the inferred consequent, $I(\cdot)$ is the rule connective, $C(\cdot)$ is the conjunction operator, $\mu_{Ai}(x_i)$ are the values of the matching degree of each input $(x_1, \ldots, x_n)$ of the system with the membership functions of the rule antecedents, and $\mu_B(\cdot)$ is the consequent of the rule.

Therefore, the Inference System performs the two following tasks:

(1) First, it computes $C(\mu_{A1}(x_1), \ldots, \mu_{An}(x_n))$ that is the *matching degree* of each rule, h_i. The conjunction operator $C(\cdot)$ is usually modeled with a t-norm.
(2) Second, it infers using the rule connective $I(\cdot)$ the matching degree and the consequent of the rule. Rule connectives can be classified into different families, the most known being the implication functions[24] and the t-norms.[25] T-norms are the most used in practical FM.

Hence, the Inference System employs two components: the conjunction $C(\cdot)$ and the rule connective $I(\cdot)$. The next subsection discusses the parametrization of these components.

2.2. Adaptive Components in the Inference System

The aforementioned two components, conjunction and rule connective, are suitable to be parameterized:

The parameter for the adaptive conjunction will be β; therefore the adaptive component is $C(\beta, \cdot)$ computed with $h_i(\beta)$.
The parameter for the adaptive rule connective will be α; therefore the adaptive component is $I(\alpha, \cdot)$.

We can consider two models of Adaptive Inference System depending on the number of parameters they use:

- a single parameter, α or β, to tune globally the behavior of the Adaptive Inference System, or
- individual parameters for every rule of the KB, α_i or β_i, having a local tuning mechanism of the behavior of the Inference System for every rule.

The parameters can be tuned with evolutionary algorithms. As mentioned previously, the learning process of the parameters can be considered as a global or local Inference System tuning process. This method of system tuning can be considered as an alternative or complementary one to other tuning methods[26] for improving accuracy in FM.

2.3. Meaning of the Adaptive Inference System in Conjunction with Defuzzification Methods

Adaptive Inference System components may be parametrized as we have described in Section 2.2, but the effects of the parametrization of its components cannot be studied in isolation. The Adaptive Inference System works jointly with the defuzzification method. Defuzzification methods always convert the inferred fuzzy sets into a crisp value.

Defuzzification methods operate in the way we describe below. We denote by means of B'_i the fuzzy set obtained as output when performing inference on rule R_i, and by means of y_0 the output of the fuzzy model for an input x_0. The *characteristic values* GC_i and MW_i, these being the Gravity Center and the Maximum Value of B'_i, respectively, are used. Defuzzification methods can be classified into two modes[4]:

- Mode A (*Aggregation First, Defuzzification After*): The defuzzification interface performs the aggregation of the individual fuzzy sets inferred, B'_i, to get the final output fuzzy set B'.
- Mode B (*Defuzzification First, Aggregation After)*: It avoids the computation of the final fuzzy set B' by considering the contribution of each rule output individually, obtaining the final output by taking a calculus (an average, a weighted sum, or a selection of one of them) of a concrete crisp characteristic value associated with each of them. This is the most used technique in practical implementation because it is easier to implement, it results in lower computational resource utilization, and it shows the best accuracy.[3]

The most used defuzzification methods acts in Mode B and employs the matching degree inside of their expression. The defuzzification method that uses the gravity center by means of a sum weighted by the matching degree has the following expression:

$$y_0 = \frac{\sum_i^N h_i \cdot GC_i}{\sum_i^N h_i}$$

Table I. Adaptive t-norms.

Figure		T-norm	
1	Dubois	$T_{\text{Dubois}}(x, y, \alpha) = \dfrac{x \cdot y}{Max(x, y, \alpha)}$	$(0 \leq \alpha \leq 1)$
2	Dombi	$T_{\text{Dombi}}(x, y, \alpha) = \dfrac{1}{1 + \sqrt[\alpha]{\left(\dfrac{1-x}{x}\right)^{\alpha} + \left(\dfrac{1-y}{y}\right)^{\alpha}}}$	$(\alpha > 0)$
3	Frank	$T_{\text{Frank}}(x, y, \alpha) = \log_{\alpha}\left[1 + \dfrac{(\alpha^{x}-1)(\alpha^{y}-1)}{\alpha - 1}\right]$	$(\alpha > 0), (\alpha \neq 1)$

where GC is computed with $[\int_Y y \cdot \mu'_B(y)\, dy]/[\int_Y \mu'_B(y)\, dy]$ (so-called standard WECOA).

In this article, we have selected the standard WECOA defuzzification method because of its significance and well-known behavior. Inferred fuzzy sets with higher matching degrees have more influence than the ones with lower matching degrees.

To study the meaning of the Adaptive Inference System in conjunction with the WECOA defuzzification method, we study the two parametrized components, adaptive rule connectives and adaptive conjunction operators.

2.3.1. Adaptive Rule Connective

If we employ $I(\alpha, \cdot)$, the parameter α allows the global rule connective tuning. If we employ $I((\alpha_i, i = 1 \ldots M), \cdot)$, the rule connective will be adapted individually. To discuss the meaning of this Adaptive Inference System, Table I exemplifies the classical parametric T-norms.[27]

Table II shows the relation between the five classical t-norms and the values of the parameter of the adaptive t-norms.

The inferred fuzzy set shape, with the three adaptive t-norms, is shown in Figures 1, 2, and 3.

In Ref. 19, the low influence of parametric rule connectives was shown. Observing Figures 1 to 3, the low influence of α can be understood. The inferred fuzzy set keeps its symmetry, so for that reason the Gravity Center or the Maximum Value (characteristic values of a fuzzy set) stand invariable. If the consequents

Table II. Relation between classical and parametrized t-norms depending on the α parameter.

	T_{Minimum}	T_{Hamacher}	$T_{\text{Algebraic}}$	T_{Einstein}	T_{Bounded}	T_{Drastic}
T_{Dubois}	0		1			
T_{Dombi}	∞	1				$\to 0$
T_{Frank}	$\to 0$		$\to 1$		∞	

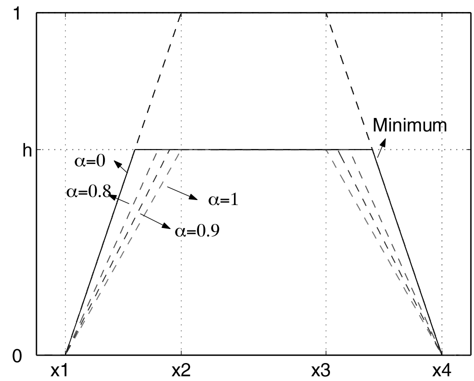

Figure 1. Inferred fuzzy set with Dubois adaptive t-norm.

do not show symmetry, then the value can be slightly displaced. Therefore, we will no longer consider employing parameterized rule connectives in this study.

2.3.2. *Adaptive Conjunction Operator*

The effect of the conjunctive parameter in the matching degree calculation is equivalent to one of the well-known mechanisms for modifying the linguistic meaning of the rule structure, the use of linguistic modifiers.[28,29] The goal of linguistic rule modifiers is also to improve the accuracy of the model, slightly relaxing the rule structure by changing the meaning of the involved labels. The inference parameter plays a similar role by changing the shape of the membership function associated with the linguistic label antecedents of the rule, as shown in Figure 4, where h is the matching for the trapezoidal fuzzy set when the input value is e, and h' and h'' are the values computed for $\beta = 0.2$ and $\beta = 0.1$, respectively. We must point

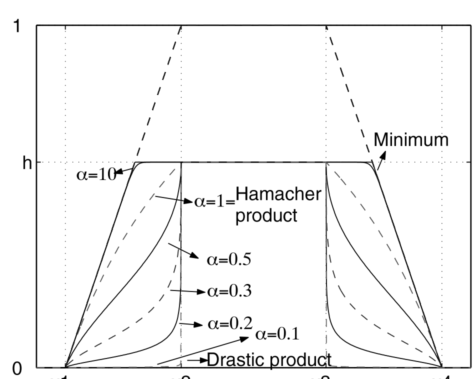

Figure 2. Inferred fuzzy set with Dombi adaptive t-norm.

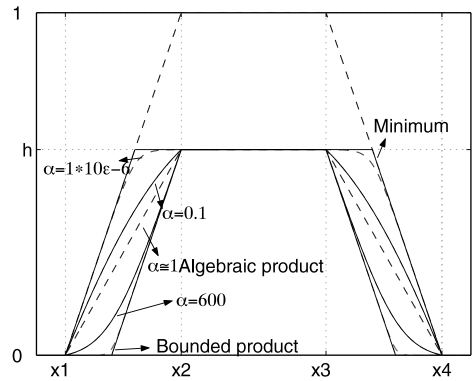

Figure 3. Inferred fuzzy set with Frank adaptive t-norm.

out that the effect of the adaptive t-norm playing the role of conjunction operator does not modify the shape of the inferred fuzzy set.

The t-norms only produce membership concentration effects. Examples of these kinds of linguistic modifiers are *absolutely*, *very*, *much more*, *more*, and *plus*.

The single parameter model globally adapts the conjunction operator between the classical t-norms. However, the benefits of this model will not yield remarkable improvements in accuracy. The reason is the low importance in choice of the conjunction operator in the design of linguistic fuzzy systems[3] with a behavior similar to the use of different t-norms.

Tables III and IV show the effects for the different Dombi t-norm values. Table III shows the results over two antecedents with matching degrees of 0.7 and 0.8. Table IV shows the results over matching degrees of 0.9 and 0.6. It must be observed that β values for both rules modify the matching proportionally.

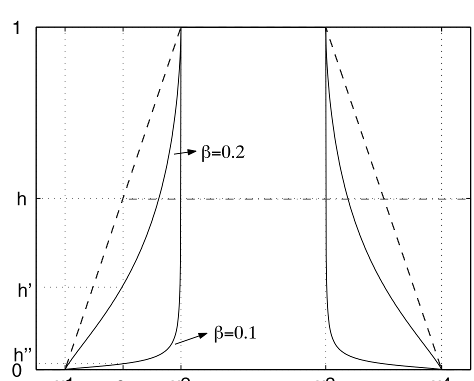

Figure 4. Graphical representation of the antecedent linguistic modification produced by different values of Dombi t-norm.

Table III. Dombi t-norm results using different values of β, for a rule with two antecedents and matching values of 0.7 and 0.8.

β	$h_i(\beta)$
0.1	0.00296
0.3	0.02307
1	0.59575
10	0.69990

Table IV. Dombi t-norm results using different values of β, for a rule with two antecedents and matching values of 0.9 and 0.6.

β	$h_i(\beta)$
0.1	0.00344
0.3	0.24451
1	0.56250
10	0.59999

Conversely, the model that employs individual parameters for each rule of the KB has a different meaning: The rules can be weighted individually. At the same time, the rule of Table III can have a β value of 10, whereas the rule of Table IV can have a β value of 0.1.

The aforementioned feature makes the Adaptive Conjunction Operator with individual parameters of each rule the most interesting case to study.

3. EVOLUTIONARY ADAPTIVE INFERENCE SYSTEMS

Our objective is to increase the fuzzy rules cooperation, getting a fuzzy-rule-based system with more accuracy, by means of the parameters optimization process. This optimization problem deals with the search for the values of a vector in order to get a set of rules that cooperates by means of their associated conjunction operators. Genetic Algorithms have been shown to be a very important tool in order to adapt Fuzzy Systems.[26]

In this section, first of all, we describe the benchmark selected, that is, the experiments employed and the measures that will be computed to show the improvements obtained with the EAIS. Second, we introduce the evolutionary algorithm used to adapt the parameters of the Adaptive Inference Systems, and finally we present the experimental results obtained and their analysis.

3.1. Benchmarks and Basic Results

We built several fuzzy models combining a representative set of EAIS with the WECOA defuzzification method to solve three different fuzzy model applications: *the estimation of the low voltage network real length in villages* (called E_1), *the estimation of the electrical medium voltage network maintenance cost in towns* (called E_2)[30] and the well-known *time series of sunspots*.[31] They are briefly described in the Appendix. The data sets E_1 and E_2 and a wide description can be found at http://decsai.ugr.es/∼casillas/fmlib.html.

As we mentioned, taking into account the study developed in Subsection 2.3, we decided to concentrate the present study on the conjunction adaptation with individual parameter for each rule of the KB.

The analyzed Inference System was based on the three adaptive t-norms shown in Section 2.3 as conjunctive operators. We have selected the classical Minimum t-norm as rule connective for all models.

We consider a usual FM performance measure, the Mean Square Error ($MSE(\cdot)$):

$$MSE(i) = \frac{\frac{1}{2}\sum_{k=1}^{N}(y_k - S[i](x_k))^2}{N}$$

where $S[i]$ denotes the fuzzy model whose Inference System uses the conjunction operator C_i ($i = 1$ for Dubois, $i = 2$ for Dombi, and $i = 3$ for Frank t-norm), rule connective Minimum t-norm, and WECOA defuzzification method. This measure uses a set of system evaluation data formed by N pairs of numerical data $Z_k = (x_k, y_k)$, $k = 1, \ldots, N$, with x_k being the values of the input variables and with y_k being the corresponding values of the associated output variables.

We also computed the standard deviation of the *MSE*,

$$SD^2(S[i]) = \frac{\sum_{k=1}^{30}(MSE_k(S[i]) - \overline{MSE}(S[i]))^2}{30}$$

As we have mentioned, we have used two learning methods for getting the rule base, in order to study the behavior of the EAIS with two different approaches:

- the classical and simple Wang and Mendel (WM) method,[22] and
- the evolutionary learning algorithm of Thrift[23]: This algorithm employs a representation of the rule base based on a relational matrix. A decision table is encoded inside the chromosome using a positional code and establishing a mapping between the labels set associated to the system output variable. The GA employs integer coding and an elitist selection scheme. The crossover operator is the standard two-point crossover. The mutation operator is specifically designed for the process. The fitness function is based on a measure of error, computing how the output converges to the desired values by means of the mean square error.

We selected seven labels for application E_1, five labels for E_2, and three labels for sunspots. The aforementioned choices are commonly employed for the applications used and they perform a reasonable equilibrium between the rule base size and the number of variables and examples per application.

Table V and Table VI show the *MSE* together with their *SD* obtained for the three applications with the KBs obtained with the WM and Thrift methods, respectively. These values are reference ones so that we may later compute the EAIS improvements. They have been computed for every one of the three adaptive t-norms with a constant parameter value shown in Table VII, together with their equivalences: Dubois parameter initiated to 0 (equivalent to Minimum), Dombi

Table V. Reference initial *MSE* values for the FM of E_1, E_2, and sunspots with the KBs obtained with WM method.

	E_1		E_2		Sunspots
	MSE	*SD*	*MSE*	*SD*	*MSE*
Training					
$MSE_{NA}(S(C_{\text{Dubois}}))$	211,776.69	8046.29	56,135.74	1498.35	144.69
$MSE_{NA}(S(C_{\text{Dombi}}))$	210,179.94	8338.88	55,418.16	1005.09	136.30
$MSE_{NA}(S(C_{\text{Frank}}))$	210,277.99	9396.21	80,441.20	4097.47	121.34
Test					
$MSE_{NA}(S(C_{\text{Dubois}}))$	227,583.28	19,904.36	56,359.42	9647.82	418.06
$MSE_{NA}(S(C_{\text{Dombi}}))$	224,079.77	18,096.60	55,531.73	3544.82	410.29
$MSE_{NA}(S(C_{\text{Frank}}))$	221,198.61	17,883.71	80,597.13	6708.20	404.57

initiated to 1 (equivalent to Hamacher product), and Frank initiated to 0.5 (similar to Hamacher product).

The Thrift learning algorithm employed a population size of 61 and a crossover probability of 0.6. It was set with 1000 generations and it has been run for 100,000 evaluations, 6 runs per partition in E_1 and E_2 (5-fcv) and 30 run for Sunspots.

The greater *MSE* presented by application E_2 with the KBs generated with the evolutionary algorithm of Thrift is remarkable. It is usual because the application has a big number of variables and labels and the number of possible rules is high.

3.2. The Evolutionary Algorithm CHC

The evolutionary algorithm selected is the CHC.[32] It is considered as an evolutionary model with a good trade-off between diversity and convergence in high-dimensional search spaces in different applications.

Table VI. Reference initial *MSE* values for the FM of E_1, E_2, and sunspots with the KBs obtained with Thrift method.

	E_1		E_2		Sunspots	
	MSE	*SD*	*MSE*	*SD*	*MSE*	*SD*
Training						
$MSE_{NA}(S(C_{\text{Dubois}}))$	167,674.17	4152.83	68,153.13	3,403.875	118.27	6.62
$MSE_{NA}(S(C_{\text{Dombi}}))$	169,560.73	4564.82	63,673.24	518.17773	116.27	5.75
$MSE_{NA}(S(C_{\text{Frank}}))$	168,640.86	1010.12	80,190.06	20,820.210	95.27	7.21
Test						
$MSE_{NA}(S(C_{\text{Dubois}}))$	195,917.47	28,351.30	68,643.69	702.980	372.74	20.17
$MSE_{NA}(S(C_{\text{Dombi}}))$	227,208.32	32,746.41	76,446.69	4,946.988	433.12	31.28
$MSE_{NA}(S(C_{\text{Frank}}))$	212,803.66	19,477.25	94,050.93	18,467.355	437.95	29.16

Table VII. Initial values of the parametrized t-norms for evolutionary process.

Parametrized t-norms	Initial values	Classical equivalent t-norm
Dubois	0	Minimum
Dombi	1	Hamacher product
Frank	0.5	Similar to Hamacher product

3.2.1. CHC Algorithm

During each generation, the CHC algorithm[32] uses a parent population of size M to generate an intermediate population of M individuals, which are randomly paired and used to generate M potential offspring. Then, a survival competition is held, where the best M chromosomes from the parent and offspring populations are selected to form the next generation.

No mutation is applied during the recombination phase. Instead, when the population converges or the search stops making progress (i.e., the difference threshold has dropped to zero and no new offspring are been generated that are better than any member of the parent population), the population is reinitialized. The restarted population completely consists of random individuals except that one of them must be the best individual found so far.[33]

Although CHC was conceived for binary-coded problems, there are real-coded versions, like the one we employ in this work. In these cases, the BLX-α crossover ($\alpha = 0.5$) is employed in order to recombine the parent's genes. The Hamming distance is computed by translating the real-coded genes into strings and by taking into account whether each character is different or not. Only those string pairs that differ from each other by some number of bits (mating threshold) are mated. The initial threshold is set to $L/4$ where L is the length of the string. When no offspring is inserted into the new population, the threshold is reduced by 1.

The used fitness function was the aforementioned Mean Square Error, *MSE*.

3.2.2. Parameters for Experimentation

We achieved 30 trials for every fuzzy model tuning process, running them with six different seeds for the random number generator and five different data sets, fivefold cross validation approach for the two electrical problems, E_1 and E_2, and with 30 different seeds for the sunspots temporal time series. The considered real *MSE* was computed as the arithmetic mean of the 30 results.

The CHC algorithm has been run for 30,000 evaluations. The population size was 50 (randomly initialized with the exception of a single chromosome with all the genes initialized to the values showed in Table XVI, below). A BLX-α crossover with $\alpha = 0.5$ was used as mentioned earlier. The initial threshold was set to $L/4$, with L being the chromosome length.

The tuning interval (searching variable domain) for β was fixed to $[0,1]$ for Dubois t-norm, $(0,10]$ for Dombi t-norm and $(0,100]$ for Frank t-norm.

3.3. Experiments and Analysis Results

We also considered the Improvement Percentage (IP) index whose expression is

$$IP(i) = 100 \times \left(1 - \frac{MSE(S(i))}{MSE(S_{NA}(i))}\right)$$

that is, the improvement shown by the $MSE(\cdot)$ of a fuzzy model $S(\cdot)$ built with a conjunction operator C_i with regard to the system without tuned parameters (or, more exactly, with Dubois, Dombi, and Frank t-norms initialized to the values shown in Table VII), $S_{NA}(i)$, where i is 1 for Dubois, 2 for Dombi, and 3 for Frank.

Tables VIII and IX show the average of the MSE together with their SD obtained for applications E_1, E_2, and sunspots temporal time series with the EAIS for both types of KBs. Tables X and XI show the IP of the MSE obtained by the EAIS with the training and test data sets for the three applications.

Table XII presents an example of β_i values obtained at the end of the adaptation process with the Dombi t-norm for the applications E_1. It can be observed that rules 1, 5, 8, 18, and 21 have a high β value; this means that they form a high cooperation set of rules, whereas rules 10, 16, and 22 are specially constrained with very low values.

Table VIII. *MSE* for the FM of E_1, E_2, and sunspots using EAIS with the KBs obtained with the WM method.

	E_1		E_2		Sunspots	
	MSE	*SD*	*MSE*	*SD*	*MSE*	*SD*
Training						
$MSE(S(C_{\text{Dubois}}))$	190,526.06	11,806.79	34,371.65	613.18	98.03	32.24
$MSE(S(C_{\text{Dombi}}))$	164,825.37	16,841.05	28,105.26	1020.34	101.87	25.26
$MSE(S(C_{\text{Frank}}))$	185,188.14	12,810.12	34,975.09	690.48	96.58	20.40
Test						
$MSE(S(C_{\text{Dubois}}))$	218,261.72	11,517.53	36,845.51	2084.11	372.36	10.93
$MSE(S(C_{\text{Dombi}}))$	213,498.66	27,319.32	30,278.23	3233.26	378.02	7.87
$MSE(S(C_{\text{Frank}}))$	210,269.22	17,035.33	36,547.04	2514.89	387.60	4.19

Table IX. *MSE* for the FM of E_1, E_2, and sunspots using EAIS with the KBs obtained with the Thrift method.

	E_1		E_2		Sunspots	
	MSE	*SD*	*MSE*	*SD*	*MSE*	*SD*
Training						
$MSE(S(C_{\text{Dubois}}))$	157,554.64	3485.29	27,927.80	758.65	89.99	0.25
$MSE(S(C_{\text{Dombi}}))$	152,532.68	4896.65	19,986.34	1,549.19	97.39	0.44
$MSE(S(C_{\text{Frank}}))$	159,053.70	3455.67	28,061.20	7,218.04	92.71	0.07
Test						
$MSE(S(C_{\text{Dubois}}))$	190,950.35	29,884.91	33,110.45	1,641.40	368.47	10.24
$MSE(S(C_{\text{Dombi}}))$	200,106.16	20,641.79	28,753.16	9,094.28	441.31	49.75
$MSE(S(C_{\text{Frank}}))$	197,692.78	35,106.89	37,486.02	12,269.80	438.40	5.82

Analyzing the results obtained, we can point out that:

- First, it is noticeable that the use of EAIS improves the classical well-known Inference Systems based on nonadaptive classical t-norms taken as references. It does not depend on the KBs employed.
- The rule cooperation mechanism shows important accuracy improvements in the application E_2, and it presents a more irregular behavior in application E_1. In sunspots application we can find an overfitting problem for the Thrift KB.
- The EAIS are a good design option to improve the accuracy in Linguistic FM keeping an important interpretability level, but require a preliminary study for analyzing their behavior.

4. TUNING OF INFERENCE SYSTEMS AND MEMBERSHIP FUNCTIONS: ANALYSIS OF THEIR COOPERATION

The tuning of membership functions is one of the most important approaches for improving the fuzzy system's accuracy. As we mentioned, the effect of introducing parameters into the inference system may have some bearing on other

Table X. *IP* of the *MSE* for the FM of E_1, E_2, and sunspots using EAIS with the KBs obtained with the WM method.

	E_1	E_2	Sunspots
Training			
$IP(S(C_{\text{Dubois}}))$	10.03	38.73	32.24
$IP(S(C_{\text{Dombi}}))$	21.58	49.26	25.26
$IP(S(C_{\text{Frank}}))$	11.93	56.41	20.40
Test			
$IP(S(C_{\text{Dubois}}))$	4.10	24.57	10.93
$IP(S(C_{\text{Dombi}}))$	4.72	45.38	7.87
$IP(S(C_{\text{Frank}}))$	4.94	54.42	4.19

Table XI. *IP* of the *MSE* for the FM of E_1, E_2, and sunspots using EAIS with the KBs obtained with the Thrift method.

	E_1	E_2	Sunspots
Training			
$IP(S(C_{\text{Dubois}}))$	6.04	59.02	23.91
$IP(S(C_{\text{Dombi}}))$	10.04	68.61	16.24
$IP(S(C_{\text{Frank}}))$	5.68	65.01	2.68
Test			
$IP(S(C_{\text{Dubois}}))$	2.54	51.76	1.15
$IP(S(C_{\text{Dombi}}))$	11.93	62.39	−1.89
$IP(S(C_{\text{Frank}}))$	7.10	60.14	−0.10

components of the fuzzy systems. In this section, we analyze the cooperation between the use of parametric conjunction operators and the tuning of the membership functions for getting more accurate fuzzy models.

We present the results with a double combination:

- Sequential tuning: to tune the membership functions and then to use EAIS in a sequential mode.
- Cooperative tuning: to tune the membership functions and the conjunction parameters together by means of a genetic representation containing all the parameters.

In the following two subsections we present first the results of the tuning of membership functions, and then the results with the double combination (sequential and cooperative tuning).

Finally, we analyze what happens when we increase the granularity of the fuzzy partitions. This interest is based on the idea of getting better results when we increase the number of labels per variable in general and in the consequent in particular. The last subsection is devoted to this study.

4.1. Tuning of Membership Functions

Tables XIII and XIV show the *MSE* and *SD* obtained for applications E_1, E_2, and sunspots time series with the descriptive genetic tuning process of the membership functions.[34] Tables XV and XVI show the *IP* of the *MSE* relative to

Table XII. β_i values taken at the end of the evolutionary process for the Dombi t-norm.

$\beta_1 \ldots \beta_6$	10.000000	0.125972	1.543214	0.253670	10.000000	0.550866
$\beta_7 \ldots \beta_{12}$	0.270757	10.000000	0.790804	0.018632	0.223670	0.197948
$\beta_{13} \ldots \beta_{18}$	8.307211	0.241470	4.839832	0.012803	3.790283	10.000000
$\beta_{19} \ldots \beta_{24}$	0.252203	7.462416	9.999999	0.012594	0.371065	1.039599

Table XIII. *MSE* for the FM of E_1, E_2, and sunspots obtained tuning the membership functions with the KBs obtained with the WM method.

	E_1		E_2		Sunspots	
	MSE	*SD*	*MSE*	*SD*	*MSE*	*SD*
Training						
$MSE(S(C_{\text{Dubois}}))$	132,643.43	8499.62	14,320.62	994.21	67.49	1.16
$MSE(S(C_{\text{Dombi}}))$	133,073.24	7475.87	14,004.49	921.95	70.91	1.39
$MSE(S(C_{\text{Frank}}))$	132,423.36	7161.21	15,304.25	335.58	61.91	0.79
Test						
$MSE(S(C_{\text{Dubois}}))$	211,534.78	35,503.44	17,460.91	2677.19	306.04	55.99
$MSE(S(C_{\text{Dombi}}))$	230,220.45	62,525.71	17,933.84	3553.59	351.58	49.02
$MSE(S(C_{\text{Frank}}))$	231,232.76	59,411.83	17,921.22	1473.26	298.55	2.44

Tables V and VI obtained by the tuning process with the training and test data sets for both applications.

The tuning process of the membership functions has been carried out employing the three parametrized t-norms with their parameter values shown in Table VII.

If we compare the results presented by the EAIS with those presented by the tuning of membership functions, we can conclude that the improvement in the accuracy shown by the latter one is greater for both applications. Of course, the lose of interpretability is greater for the membership functions tuning than for EAIS. In the next subsection we analyze the possible cooperation between both approaches.

4.2. Adaptive Inference Systems and Tuning of Membership Functions

In this section we present the results of the double combination: sequential and cooperative tuning:

Table XIV. *MSE* for the FM of E_1, E_2, and sunspots obtained tuning the membership functions with the KBs obtained with the Thrift method.

	E_1		E_2		Sunspots	
	MSE	*SD*	*MSE*	*SD*	*MSE*	*SD*
Training						
$MSE(S(C_{\text{Dubois}}))$	129,188.82	3787.05	16,717.65	899.74	69.16	3.94
$MSE(S(C_{\text{Dombi}}))$	129,844.59	4064.47	17,997.58	1821.57	66.38	2.60
$MSE(S(C_{\text{Frank}}))$	131,016.08	3853.46	20,542.54	4478.10	60.29	0.63
Test						
$MSE(S(C_{\text{Dubois}}))$	252,325.48	70,558.43	23,820.59	3068.86	293.39	293.39
$MSE(S(C_{\text{Dombi}}))$	216,808.39	31,500.91	25,810.43	1505.34	274.25	274.25
$MSE(S(C_{\text{Frank}}))$	256,887.06	98,835.80	30,348.42	9121.56	293.24	293.24

Table XV. *IP* of the *MSE* of Table XIII for E_1, E_2, and sunspots obtained tuning the membership functions with the KBs obtained with the WM method.

	E_1	E_2	Sunspots
Training			
$IP(S(C_{\text{Dubois}}))$	37.37	74.49	53.36
$IP(S(C_{\text{Dombi}}))$	36.69	74.73	47.98
$IP(S(C_{\text{Frank}}))$	37.02	80.97	48.98
Test			
$IP(S(C_{\text{Dubois}}))$	7.05	69.02	26.79
$IP(S(C_{\text{Dombi}}))$	−2.74	67.71	14.31
$IP(S(C_{\text{Frank}}))$	−4.54	77.76	26.21

- First we use the EAIS to improve the *MSE* obtained by the membership function tuning in a sequential mode. The results are showed in Tables XVII to XX.
- Second, we present the results of the cooperative tuning. To do so, we employ a bigger chromosome in the CHC algorithm, with two parameter parts: membership function and EAIS parameters, respectively.
- The population size was 50, randomly initialized with the exception of a single chromosome with all the genes initialized to the original membership function values in the first part, and with the values shown in Table VIII for the second part. It was run for 35,000 evaluations.
- Tables XXI and XXII show the *MSE* and *SD* obtained for the applications and Tables XXIII and XXIV show the corresponding *IP*s.

The use of EAIS yields a slight improvement in the accuracy of the earlier tuned KBs. The EAIS increases the rule cooperation of an accuracy improved KB.

Employing the EAIS together with the membership functions tuning, the cooperative tuning, it is possible to improve the accuracy presented by the tuning

Table XVI. *IP* of the *MSE* of Table XIV for E_1, E_2, and sunspots obtained tuning the membership functions with the KBs obtained with the Thrift method.

	E_1	E_2	Sunspots
Training			
$IP(S(C_{\text{Dubois}}))$	22.95	75.47	41.52
$IP(S(C_{\text{Dombi}}))$	23.42	71.73	42.91
$IP(S(C_{\text{Frank}}))$	22.31	74.38	36.72
Test			
$IP(S(C_{\text{Dubois}}))$	−28.79	65.30	21.29
$IP(S(C_{\text{Dombi}}))$	4.58	66.24	36.68
$IP(S(C_{\text{Frank}}))$	−20.72	67.73	33.04

Table XVII. *MSE* presented by the EAIS employing the previously tuned membership functions KB with the KBs obtained with the WM method.

	E_1		E_2		Sunspots	
	MSE	*SD*	*MSE*	*SD*	*MSE*	*SD*
Training						
$MSE(S(C_{\text{Dubois}}))$	130,622.64	8751.74	10,759.67	663.30	63.76	0.11
$MSE(S(C_{\text{Dombi}}))$	128,056.36	7168.05	9,022.45	451.44	64.69	0.11
$MSE(S(C_{\text{Frank}}))$	128,254.02	7832.27	9,435.29	445.51	59.38	0.01
Test						
$MSE(S(C_{\text{Dubois}}))$	224,588.05	49,825.32	14,228.34	2307.87	259.22	0.81
$MSE(S(C_{\text{Dombi}}))$	232,642.87	56,992.76	13,748.90	3833.72	298.70	7.19
$MSE(S(C_{\text{Frank}}))$	221,045.49	42,347.41	12,216.03	2009.04	305.69	0.01

Table XVIII. *MSE* presented by the EAIS employing the previously tuned membership functions KB with the KBs obtained with the Thrift method.

	E_1		E_2		Sunspots	
	MSE	*SD*	*MSE*	*SD*	*MSE*	*SD*
Training						
$MSE(S(C_{\text{Dubois}}))$	127,018.98	3,123.36	7770.98	138.02	57.80	0.23
$MSE(S(C_{\text{Dombi}}))$	134,199.07	16,700.86	7507.92	426.58	55.48	1.13
$MSE(S(C_{\text{Frank}}))$	128,489.48	4,530.78	8509.98	1669.17	53.87	0.26
Test						
$MSE(S(C_{\text{Dubois}}))$	252,411.04	71,061.56	12,267.53	2967.92	304.92	2.50
$MSE(S(C_{\text{Dombi}}))$	233,469.22	39,703.17	11,290.64	113.53	239.59	14.27
$MSE(S(C_{\text{Frank}}))$	253,108.35	100,067.61	12,380.31	977.12	327.15	19.80

Table XIX. *IP* of the *MSE* shown by Table XVII concerning Table V with the KBs obtained with the WM method.

	E_1	E_2	Sunspots
Training			
$IP(S(C_{\text{Dubois}}))$	38.32	80.83	55.93
$IP(S(C_{\text{Dombi}}))$	39.07	83.72	52.54
$IP(S(C_{\text{Frank}}))$	39.01	88.27	51.06
Test			
$IP(S(C_{\text{Dubois}}))$	1.32	74.75	38.00
$IP(S(C_{\text{Dombi}}))$	−3.82	75.24	27.20
$IP(S(C_{\text{Frank}}))$	0.07	84.84	24.44

Table XX. *IP* of the *MSE* shown by Table XVIII concerning Table VI with the KBs obtained with the Thrift method.

	E_1	E_2	Sunspots
Training			
$IP(S(C_{\text{Dubois}}))$	24.25	88.60	51.13
$IP(S(C_{\text{Dombi}}))$	20.85	88.21	52.28
$IP(S(C_{\text{Frank}}))$	23.81	89.39	43.46
Test			
$IP(S(C_{\text{Dubois}}))$	−28.84	82.13	18.19
$IP(S(C_{\text{Dombi}}))$	−2.76	85.23	44.68
$IP(S(C_{\text{Frank}}))$	−18.94	86.84	25.30

Table XXI. *MSE* presented tuning the membership functions together with the EAIS parameters with the KBs obtained with the WM method.

	E_1		E_2		Sunspots	
	MSE	*SD*	*MSE*	*SD*	*MSE*	*SD*
Training						
$MSE(S(C_{\text{Dubois}}))$	128,221.17	7871.52	8751.82	575.49	55.66	2.08
$MSE(S(C_{\text{Dombi}}))$	121,482.12	8099.42	7818.85	647.30	59.54	1.89
$MSE(S(C_{\text{Frank}}))$	127,069.59	8118.14	9526.95	587.66	50.79	0.89
Test						
$MSE(S(C_{\text{Dubois}}))$	195,568.21	26,216.73	11,912.64	2031.94	276.97	32.44
$MSE(S(C_{\text{Dombi}}))$	208,155.76	37,347.05	11,257.84	2646.91	370.33	57.60
$MSE(S(C_{\text{Frank}}))$	215,760.09	87,979.12	12,286.52	1781.26	296.72	15.58

Table XXII. *MSE* presented tuning the membership functions together with the EAIS parameters with the KBs obtained with the Thrift method.

	E_1		E_2		Sunspots	
	MSE	*SD*	*MSE*	*SD*	*MSE*	*SD*
Training						
$MSE(S(C_{\text{Dubois}}))$	125,608.51	3883.47	4897.92	353.39	51.78	138.02
$MSE(S(C_{\text{Dombi}}))$	121,275.28	2892.96	5756.68	691.45	54.44	426.58
$MSE(S(C_{\text{Frank}}))$	125,902.53	5432.17	7772.74	1939.89	50.54	1669.17
Test						
$MSE(S(C_{\text{Dubois}}))$	204,230.16	23,655.02	9,416.76	3809.34	279.30	2967.92
$MSE(S(C_{\text{Dombi}}))$	206,733.75	18,239.51	8,874.23	954.05	303.51	113.53
$MSE(S(C_{\text{Frank}}))$	210,823.18	24,547.44	11,124.18	2544.66	317.08	977.12

Table XXIII. *IP* of the *MSE* shown by Table XXI concerning Table V with the KBs obtained with the WM method.

	E_1	E_2	Sunspots
Training			
$IP(S(C_{\text{Dubois}}))$	39.45	84.41	61.53
$IP(S(C_{\text{Dombi}}))$	42.20	85.89	56.32
$IP(S(C_{\text{Frank}}))$	39.57	88.16	58.14
Test			
$IP(S(C_{\text{Dubois}}))$	14.07	78.86	33.75
$IP(S(C_{\text{Dombi}}))$	7.11	79.73	9.74
$IP(S(C_{\text{Frank}}))$	2.46	84.76	26.66

process of the membership functions, getting higher error reduction than considering both processes independently.

4.3. Increasing the Granularity

This section is dedicated to analyzing the effect of increasing the granularity of the variables, that is, the number of labels employed in the partitions. Sometimes, this method can be an easy way to improve the accuracy of the FMs, particularly when the bigger granularity is achieved in the consequent. It is an alternative possibility for improving the fuzzy system accuracy. We can compare both approximations tuning versus granularity increasing.

First, we will study the influence of other choices in the *MSE* presented by the original Rule Bases obtained with the WM method. The results are shown in the following tables:

- Tables XXV and XXVI belong to application E_1 whereas Tables XXVII to XXIX belong to application E_2. They show the *MSE* presented by the Knowledge Bases for every

Table XXIV. *IP* of the *MSE* shown by Table XXIII concerning Table VI with the KBs obtained with the Thrift method.

	E_1	E_2	Sunspots
Training			
$IP(S(C_{\text{Dubois}}))$	25.09	92.81	56.22
$IP(S(C_{\text{Dombi}}))$	28.48	90.96	53.18
$IP(S(C_{\text{Frank}}))$	25.34	90.31	46.95
Test			
$IP(S(C_{\text{Dubois}}))$	−4.24	86.28	25.07
$IP(S(C_{\text{Dombi}}))$	9.01	88.39	29.92
$IP(S(C_{\text{Frank}}))$	0.93	88.17	27.60

Table XXV. *MSE* and number of rules presented by the five different partitions used employing the WM method with 7/7 and 7/9 labels for E_1.

	7 labels ant./7 labels cons.			7 labels ant./9 labels cons.	
		MSE		*MSE*	
	#R	Training	Test	Training	Test
Partition 1	22	218,644.02	196,974.69	248,458.91	236,081.59
Partition 2	20	211,788.63	252,227.86	295,064.19	337,428.78
Partition 3	23	201,959.30	235,773.61	205,054.81	277,017.84
Partition 4	21	203,868.78	212,921.75	230,652.75	301,492.59
Partition 5	24	222,622.75	240,018.50	239,633.16	330,582.31
Mean *MSE*		211,776.69	227,583.28	243,772.76	296,520.63

partition as well as the number of rules obtained when we use some combinations of labels for the antecedents/consequent (5/5, 5/7, 5/9, 7/7, 7/9, and 9/9) of the variables.

- Tables XXX to XXXII show the *MSE* presented by the Knowledge Bases for the sunspots data set. The combinations of labels computed were 3/3, 3/5, 3/7, 5/5, 5/7, and 7/7.

Taking into account that the original partition employed for application E_1 was seven labels for the antecedent and seven labels for the consequent, we can observe in Tables XXV and XXVI that we do not get better results when we increase the granularity than with the initial granularity (7/7).

Application E_2 was partitioned with 5/5, that is, five labels for the antecedents and five labels for the consequents. We have usually gotten better results than for the initial ones when we increase the granularity. In all cases these new results are far from those that we obtain via the tuning. Therefore the tuning may be considered a better mechanism for improving the accuracy.

Finally, the sunspot time series originally employed with three labels for the antecedents and the consequent (results in Tables XXX to XXXII). The results

Table XXVI. *MSE* and number of rules presented by the five different partitions used employing the WM method with 9/9 labels for E_1.

	9 labels ant./9 labels cons.		
		MSE	
	#R	Training	Test
Partition 1	30	204,625.59	234,941.27
Partition 2	28	282,412.03	339,344.97
Partition 3	30	200,226.84	293,158.28
Partition 4	28	293,592.38	312,416.34
Partition 5	29	197,613.66	283,645.19
Mean *MSE*		235,694.10	292,701.21

Table XXVII. *MSE* and number of rules presented by the five different partitions used employing the WM method with 5/5 and 5/7 labels for E_2.

	5 labels ant./5 labels cons.			5 labels ant./7 labels cons.		
		MSE			*MSE*	
	#R	Training	Test	#R	Training	Test
Partition 1	65	54,148.87	62,719.16	65	44,575.02	54,985.05
Partition 2	65	57,562.54	49,048.29	65	59,793.17	55,354.45
Partition 3	65	54,853.61	59,896.96	65	46,324.49	47,978.95
Partition 4	65	58,031.76	55,149.77	65	46,506.70	47,249.28
Partition 5	65	56,081.95	54,982.91	65	47,577.53	42,982.64
Mean *MSE*		56,135.75	56,359.42		48,955.38	49,710.07

Table XXVIII. *MSE* and number of rules presented by the five different partitions used employing the WM method with 5/9 and 7/7 labels for E_2.

	5 labels ant./9 labels cons.			7 labels ant./7 labels cons.		
		MSE			*MSE*	
	#R	Training	Test	#R	Training	Test
Partition 1	65	64,177.76	81,890.68	102	54,514.89	70,262.36
Partition 2	65	67,726.49	67,679.17	104	53,003.75	51,968.31
Partition 3	65	68,599.83	62,963.25	103	49,334.38	61,391.48
Partition 4	65	65,391.13	69,897.65	104	54,405.44	51,091.46
Partition 5	65	70,052.98	58,417.45	104	54,205.27	42,762.89
Mean *MSE*		67,189.64	68,169.64		53,092.75	55,495.30

Table XXIX. *MSE* and number of rules presented by the five different partitions used employing the WM method with 7/9 and 9/9 labels for E_2.

	7 labels ant./9 labels cons.			9 labels ant./9 labels cons.		
		MSE			*MSE*	
	#R	Training	Test	#R	Training	Test
Partition 1	102	32,968.54	45,989.63	126	30,428.66	34,888.71
Partition 2	104	34,283.67	29,444.22	130	31,621.75	32,045.95
Partition 3	103	44,066.95	54,232.24	130	33,616.13	29,731.99
Partition 4	104	31,851.72	38,270.32	129	31,883.31	42,614.27
Partition 5	104	35,671.00	26,431.35	130	32,279.15	33,531.31
Mean *MSE*		35,768.38	38,873.55		31,965.80	34,562.45

Table XXX. *MSE* and number of rules employing the WM method with 3/3 and 3/5 labels for sunspots.

3 labels ant./3 labels cons.			3 labels ant./5 labels cons.		
	MSE			*MSE*	
#R	Training	Test	#R	Training	Test
21	144.69	418.06	21	190.53	254.69

have been improved by the combination 3/5, but they have not improved by the other combination with a high number of labels per antecedent and consequent. Of course, we get better results in the training but not in the test, where we get very high error.

The application of tuning of membership functions together with EAIS offers in general better results than the approximation based on increasing the granularity.

Finally, we have decided to apply the tuning to the 3/5 partition Knowledge Base of sunspots with the EAIS and tuning of membership functions:

- Table XXIII shows the initial values for the KB obtained with the WM method using the three adaptive t-norms with the parameters of Table VII.
- Table XXXIV presents the values obtained using the EAIS, and Table XXXV presents the *IP* reached using EAIS. There are only improvements over the training data set, but the Dombi t-norm rule connective reveals an important overfitting.

The next step was to tune the membership functions. Table XXXVI shows the *MSE* of the tuned database, and Table XXXVII presents the *IP*s of Table XXXVI. Database tuning gets important improvements, but overfitting appears using Dubois and Dombi t-norms.

We used EAIS with the tuned KBs. Table XXXVIII includes the *MSE* results of the EAIS process, whereas Table XXXIX shows the *IP* of Table XXXVIII. We again find overfitting in two cases.

Finally, we made use of a model with both adaptive elements together: EAIS and database tuning. It learns parameters of the rule connectives and database values jointly, in the same evolutionary process. Table XL shows the *MSE* for the adaptation of both conjunction connective and membership functions. Table XLI

Table XXXI. *MSE* and number of rules employing the WM method with 3/7 and 5/5 labels for sunspots.

3 labels ant./7 labels cons.			5 labels ant./5 labels cons.		
	MSE			*MSE*	
#R	Training	Test	#R	Training	Test
21	199.97	349.03	60	147.95	600.39

Table XXXII. *MSE* and number of rules employing the WM method with 5/7 and 7/7 labels for sunspots.

5 labels ant./7 labels cons.			7 labels ant./7 labels cons.		
	MSE			*MSE*	
#R	Training	Test	#R	Training	Test
60	145.06	591.69	93	113.59	664.96

Table XXXIII. *MSE* values for sunspots with the KB obtained with the WM method using three labels in the antecedents and five labels in the consequents.

Training	
$MSE_{NA}(S(C_{\text{Dubois}}))$	190.53
$MSE_{NA}(S(C_{\text{Dombi}}))$	185.00
$MSE_{NA}(S(C_{\text{Frank}}))$	167.68
Test	
$MSE_{NA}(S(C_{\text{Dubois}}))$	254.69
$MSE_{NA}(S(C_{\text{Dombi}}))$	246.91
$MSE_{NA}(S(C_{\text{Frank}}))$	223.92

Table XXXIV. *MSE* for sunspots using EAIS with the KB obtained with the WM method using three labels in the antecedents and five labels in the consequents.

	MSE	*SD*
Training		
$MSE_{NA}(S(C_{\text{Dubois}}))$	138.38	0.06
$MSE_{NA}(S(C_{\text{Dombi}}))$	134.82	0.22
$MSE_{NA}(S(C_{\text{Frank}}))$	129.69	0.02
Test		
$MSE_{NA}(S(C_{\text{Dubois}}))$	236.09	2.02
$MSE_{NA}(S(C_{\text{Dombi}}))$	459.89	2.94
$MSE_{NA}(S(C_{\text{Frank}}))$	238.58	0.35

Table XXXV. *IP* of Table XXXIV.

Training	
$MSE_{NA}(S(C_{\text{Dubois}}))$	27.37
$MSE_{NA}(S(C_{\text{Dombi}}))$	27.12
$MSE_{NA}(S(C_{\text{Frank}}))$	22.66
Test	
$MSE_{NA}(S(C_{\text{Dubois}}))$	7.30
$MSE_{NA}(S(C_{\text{Dombi}}))$	−86.26
$MSE_{NA}(S(C_{\text{Frank}}))$	−6.54

presents the corresponding *IP*s. The training improvements are important, but it is easy to reach overfitting again.

As a final conclusion of this subsection, we must say that, in general, with the tuning approaches we get better results than increasing the granularity. When we increase the granularity we cannot be sure of improving the accuracy (see examples E_1 and sunspots).

Table XXXVI. *MSE* for sunspots obtained tuning the membership functions with the KB obtained with the WM method using three labels in the antecedents and five labels in the consequents.

	MSE	*SD*
Training		
$MSE_{NA}(S(C_{\text{Dubois}}))$	73.22	0.38
$MSE_{NA}(S(C_{\text{Dombi}}))$	73.18	1.97
$MSE_{NA}(S(C_{\text{Frank}}))$	67.98	0.13
Test		
$MSE_{NA}(S(C_{\text{Dubois}}))$	304.31	63.66
$MSE_{NA}(S(C_{\text{Dombi}}))$	274.56	44.77
$MSE_{NA}(S(C_{\text{Frank}}))$	223.77	12.56

Table XXXVII. *IP* of Table XXXVI.

Training	
$MSE_{NA}(S(C_{\text{Dubois}}))$	61.57
$MSE_{NA}(S(C_{\text{Dombi}}))$	60.44
$MSE_{NA}(S(C_{\text{Frank}}))$	59.46
Test	
$MSE_{NA}(S(C_{\text{Dubois}}))$	−19.48
$MSE_{NA}(S(C_{\text{Dombi}}))$	−11.20
$MSE_{NA}(S(C_{\text{Frank}}))$	0.07

Table XXXVIII. *MSE* of EAIS method employing the previously tuned membership functions KB.

	MSE	*SD*
Training		
$MSE_{NA}(S(C_{\text{Dubois}}))$	68.68	0.27
$MSE_{NA}(S(C_{\text{Dombi}}))$	61.81	1.01
$MSE_{NA}(S(C_{\text{Frank}}))$	62.54	0.23
Test		
$MSE_{NA}(S(C_{\text{Dubois}}))$	368.12	6.98
$MSE_{NA}(S(C_{\text{Dombi}}))$	227.10	2.25
$MSE_{NA}(S(C_{\text{Dombi}}))$	253.93	0.63

Table XXXIX. *IP* of Table XXXVIII.

Training	
$MSE_{NA}(S(C_{\text{Dubois}}))$	63.95
$MSE_{NA}(S(C_{\text{Dombi}}))$	66.59
$MSE_{NA}(S(C_{\text{Frank}}))$	62.70
Test	
$MSE_{NA}(S(C_{\text{Dubois}}))$	−44.54
$MSE_{NA}(S(C_{\text{Dombi}}))$	8.02
$MSE_{NA}(S(C_{\text{Frank}}))$	−13.40

Table XL. *MSE* presented tuning the membership functions together with the EAIS parameters.

	MSE	*SD*
Training		
$MSE_{NA}(S(C_{\text{Dubois}}))$	57.94	0.48
$MSE_{NA}(S(C_{\text{Dombi}}))$	64.47	3.82
$MSE_{NA}(S(C_{\text{Frank}}))$	60.91	2.72
Test		
$MSE_{NA}(S(C_{\text{Dubois}}))$	225.24	94.74
$MSE_{NA}(S(C_{\text{Dombi}}))$	331.16	80.06
$MSE_{NA}(S(C_{\text{Frank}}))$	258.26	106.26

Table XLI. *IP* of Table XL.

Training	
$MSE_{NA}(S(C_{\text{Dubois}}))$	69.59
$MSE_{NA}(S(C_{\text{Dombi}}))$	65.15
$MSE_{NA}(S(C_{\text{Frank}}))$	63.68
Test	
$MSE_{NA}(S(C_{\text{Dubois}}))$	11.56
$MSE_{NA}(S(C_{\text{Dombi}}))$	−34.12
$MSE_{NA}(S(C_{\text{Frank}}))$	−15.33

In particular, we have focused attention on the sunspots problem for three labels in the antecedent and five labels in the consequent, with good results. In this case, when we use the evolutionary tuning, we get better results in training but only slight improvements in test for some conjunction operators. We get high *SD* values in the test; this show a high dispersion of the results per run.

It would be of interest to analyze this behavior of the evolutionary algorithms and to design a more robust evolutionary algorithm for the tuning of membership functions and EAIS. On the other hand, according to these results we cannot be sure of getting the best results via tuning and EAIS if we have used a high granularity due to possible overfitting.

5. CONCLUDING REMARKS

In the framework of improving the accuracy of linguistic fuzzy systems, the tuning of system components is a helpful alternative. EAIS are a valuable option. In this work we have studied their models, types, and relationship with other system components, presenting empirical results of several EAIS.

The improvement shown by the combination of the EAIS tuning process together with the membership function tuning is important. It gets improvements when being used with KBs easily obtained or with those learned with more advanced methods.

Acknowledgment

This article was supported by the MCyT under project TIC2002-04036-C05-01.

References

1. Casillas J, Cordón O, Herrera F, Magdalena L, editors. Interpretability issues in fuzzy modeling. Berlin: Springer-Verlag; 2003.
2. Casillas J, Cordón O, Herrera F, Magdalena L, editors. Accuracy improvements in linguistic fuzzy modeling. Berlin: Springer-Verlag; 2003.
3. Cordón O, Herrera F, Peregrín A. Applicability of the fuzzy operators in the design of fuzzy logic controllers. Fuzzy Set Syst 1997;86:15–41.
4. Cordón O, Herrera F, Peregrín A. Searching for basic properties obtaining robust implication operators in fuzzy control. Fuzzy Set Syst 2000;111:237–251.
5. Filev DP, Yager RR. An adaptative approach to defuzzification based on level sets. Fuzzy Set Syst 1993;54:355–360.
6. Yager RR, Filev DP. SLIDE: A simple adaptive defuzzification method. IEEE Trans Fuzzy Syst 1993;1:69–78.
7. Bastian A. Handling the nonlinearity of a fuzzy logic controller at the transition between rules. Fuzzy Set Syst 1995;71:369–387.
8. Jiang T, Li Y. Generalized defuzzification strategies and their parameter learning procedures. IEEE Trans Fuzzy Syst 1996;4:64–71.
9. Kandel A, Friedman M. Defuzzification using most typical values. IEEE Trans Syst Man Cybern 1998;28:901–906.
10. Esogbue AO, Song Q, Hearnes WE II. Defuzzification filters and applications to power system stabilization problems. J Math Anal Appl 2000;251:406–432.

11. Hong CZ. Handwritten numeral recognition using a small number of fuzzy rules with optimized defuzzification parameters. Neural Netw 1995;8:821–827.
12. Song Q, Leland RP. Adaptive learning defuzzification techniques and applications. Fuzzy Set Syst 1996;81:321–329.
13. Jin Y, von Seelen W. Evaluating flexible fuzzy controllers via evolution strategies. Fuzzy Set Syst 1999;108:243–252.
14. Kim D, Choi Y, Lee S. An accurate COG defuzzifier design using Lamarckian co-adaptation of learning and evolution. Fuzzy Set Syst 2002;130:207–225.
15. Cordon O, Herrera F, Marquez FA, Peregrín A. A study on the evolutionary adaptive defuzzification methods in fuzzy modeling. Int J Hybrid Intell Syst 2004;1:36–48.
16. Berger M. A new parametric family of fuzzy connectives and their application to fuzzy control. Fuzzy Set Syst 1998;93:1–16.
17. Batyrshin I, Kaynak O. Parametric classes of generalized conjunction and disjunction for fuzzy modeling. IEEE Trans Fuzzy Syst 1999;7:586–596.
18. Batyrshin I, Kaynak O. Fuzzy modeling based on generalized conjunction operators. IEEE Trans Fuzzy Syst 2003;10:678–683.
19. Herrera F, Marquez FA, Peregrín A. Genetic adaptation of rule connectives and conjunction operators in fuzzy rule based systems: An experimental comparative study. In: Proc Third Int Conf of the European Society for Fuzzy Logic and Technology; 2003. pp 100–104.
20. Moraga C, Trillas E, Guadarrama S. Multiple-valued logic and artificial intelligence fundamentals of fuzzy control revisited. Artif Intell Rev 2003;20:169–197.
21. Whalen T. Parametrized R-implications. Fuzzy Set Syst 2003;134:231–281.
22. Wang LX, Mendel JM. Generating fuzzy rules by learning from examples. IEEE Trans Syst Man Cybern 1992;22:1414–1427.
23. Thrift P. Fuzzy logic synthesis with genetic algorithms. In: Proc Fourth Int Conf on Genetic Algorithms. San Francisco, CA: Morgan Kaufmann; 1991. pp 509–513.
24. Trillas E, Valverde L. On implication and indistinguishability in the setting of fuzzy logic. In: Kacpryzk J, Yager R, editors. Management decision support systems using fuzzy sets and possibility theory. Cologne: Verlag TÜV Rheinland; 1985. pp 198–212.
25. Gupta M, Qi J. Theory of t-norms and fuzzy inference methods. Fuzzy Set Syst 1991; 40:431–450.
26. Cordón O, Herrera F, Hoffmann F, Magdalena L. Genetic fuzzy systems: Evolutionary tuning and learning of knowledge bases. Singapore: World Scientific; 2001.
27. Mizumoto M. Pictorial representations of fuzzy connectives, Part I: Cases of t-norms, t-conorms and averaging operators. Fuzzy Set Syst 1989;31:217–242.
28. Cordón O, del Jesus MJ, Herrera F. Genetic learning of fuzzy rule-based classification systems cooperating with fuzzy reasoning methods. Int J Intell Syst 1998;13:1025–1053.
29. Liu BD, Chen CY, Tsao JY. Design of adaptative fuzzy logic controller based on linguistic-hedge concepts and genetic algorithms. IEEE Trans Syst Man Cybern 2001;31:32–53.
30. Cordón O, Herrera F, Sánchez L. Solving electrical distribution problems using hybrid evolutionary data analysis techniques. Appl Intell 1999;10:5–24.
31. Foukal PV. The variable sun. Sci Am 1990;262:34–41.
32. Eshelman LJ. The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In: Rawlins GJE, editor. Foundations of genetic algorithms. San Mateo, CA: Morgan Kaufman; 1991. pp 265–283.
33. Eshelman LJ, Mathias KE, Schaffer JD. Convergence controlled variation. In: Belew R, Vose M, editors. Foundations of genetic algorithms, 4th ed. San Mateo, CA: Morgan Kaufmann; 1997. pp 203–224.
34. Cordón O, Herrera F. A three-stage process for learning descriptive and approximate fuzzy logic controller knowledge bases from examples. Int J Approx Reason 1997;17:369–407.
35. Marple SL. Digital spectral analysis with applications. Englewood Cliffs, NJ: Prentice Hall; 1997.
36. Angeline PJ. Evolving predictors for chaotic time series. In: Proc SPIE: Application and Science of Computational Intelligence; 1998. pp 170–180.

APPENDIX: APPLICATION BENCHMARKS

Two electrical distribution problems described in Ref. 30 and a classical problem of a time series of sunspots have been selected to analyze the performance of the EAIS methods in FM. The first application, E_1, is the estimation of the low voltage network real length in rural villages, the second application, E_2, is the estimation of the electrical medium voltage network maintenance cost in a town, and the third application, sunspots, is a time series prediction used in the literature to measure the performance of prediction and modeling systems.

E_1 Application

The data set has two inputs and a single output from 495 villages. The domains of the input variables are [1, 320] and [60, 1673], respectively. The output variable takes values in the interval [80, 7675]. The input and output variable domains have been partitioned with seven labels {ES, VS, S, M, L, VL, EL} as shown in Figure A.1, with the following meaning:

ES is extremely small
VS is very small
S is small
M is medium
L is large
VL is very large
EL is extremely large.

Two kinds of rule bases have been obtained: on the one hand, the rule bases are composed of 20 to 24 linguistic rules depending on the partition, obtained with the Wang and Mendel method.[22] On the other, the rule bases are composed of 42 to 47 rules achieved with the Thrift evolutionary learning method.[23] Both have been obtained from data training sets of 80% of the original available data, that is, 396 villages taken randomly. We have considered fivefold cross validation; therefore we get five rule bases associated with the five training sets.

The evaluation of the different fuzzy models composed of the EAIS methods have been carried out with the remaining 20% of the initial data set, that is, with data from 99 villages.

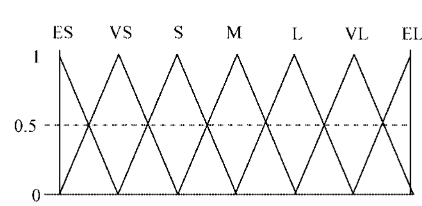

Figure A.1. Fuzzy partition considered for the input and output variables of E_1.

E_2 Application

The second electrical distribution problem, E_2, has a data set of 1059 cities with four input variables and a single output. The input variable domains are [0.5, 11], [0.15, 8.55], [1.64, 142.5], and [1, 165], respectively, whereas the output variable domain is [0, 8546.030273]. The fuzzy partition employed for inputs and output has five labels {MP, P, M, G, MG}, where

VS is very small
S is small
M is middle
L is large
VL is very large.

We have two kinds of rule bases: those composed of 65 linguistic rules achieved with the Wang and Mendel method[22] and those obtained with the Thrift[23] method, with about around 520 rules. Both have also been obtained from training data sets of 80% of the original available data, that is, 847 cities taken randomly. Evaluation of the fuzzy models has been carried out with the remaining 20% of the initial data set, that is, with data from 212 cities. In the same way, we have considered a five-fold cross validation.

The Sunspots Time Series

Sunspots, often larger in diameter than the earth, are dark blotches on the sun. They were first observed around 1610, shortly after the invention of the telescope.[31] Yearly averages have been recorded since 1700. The sunspot numbers are defined as $k(10g + f)$, where g is the number of sunspot groups, f is the number of individual sunspots, and k is used to reduce different telescopes to the same scale.[35] The observations are shown as black squares in Figure A.2. The average time between maxima is 11 years. Notice, however, that the time between maxima ranges from 7 to 15 years.

The underlying mechanism for sunspot appearances is not exactly known. No first-principles theory exists, although it is known that sunspots are related to other solar activities. For example, the magnetic field of the sun changes with an average period of 22 years. Sunspots usually appear in pairs, corresponding to magnetic dipoles. Sunspot pairs reverse their polarity from one cycle to the next, reflecting the underlying magnetic cycle. The sunspot series has served as a benchmark in the statistics literature.

The goal of time series prediction can be stated succinctly as follows: given a sequence $y(1), y(2), \ldots, y(N)$ up to time N, find the continuation $y(N+1)$, $y(N+2)\ \ldots$. The series may arise from sampling of a continuous time system and be either stochastic or deterministic in origin.

In time-series prediction, there are two forms to measure the output errors:

- *Single-step prediction*: where external inputs to the method are true observed data.
- *Iterative prediction*: where external inputs to the method are predicted outputs from previous iterations.

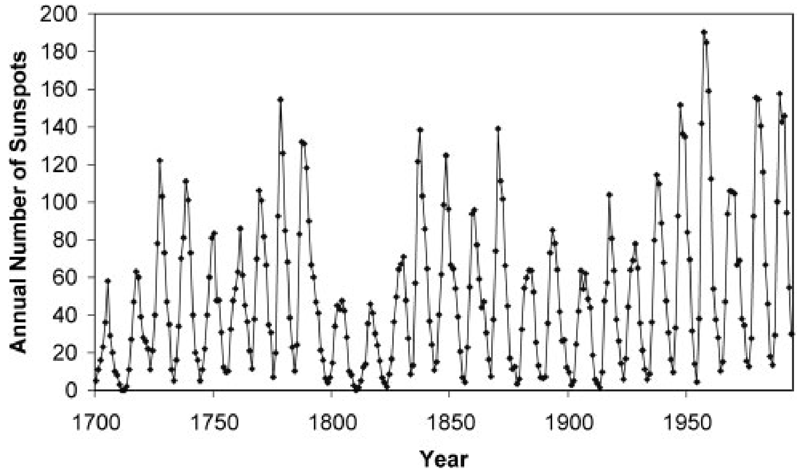

Figure A.2. The sunspot data.

Single-step prediction has been usually used in the literature. In this way, we will use single-step prediction to analyze the methods. To estimate the performance of the methods we check the *MSE*. This kind of fitting error describes how well the points are approximated by the surface over the input space.

Many works try to analyze the sunspot data using linear and nonlinear methods. The sunspot of years 1700 through 1920 were chosen to be the training set, and the sunspots of years 1921 through 1994 were chosen for single-step prediction, where the error is computed.

In this work we have a temporal window with four delays. These delays are the 1, 2, 4, and 8 delay from the predicted data. In Figure A.3, we can see the windows considered.

Therefore, we will only use four points, but we will cover eight delays (as was proposed in Ref. 36). Because this problem will be solved with fuzzy modeling-based techniques, each delay should be considered as an input variable

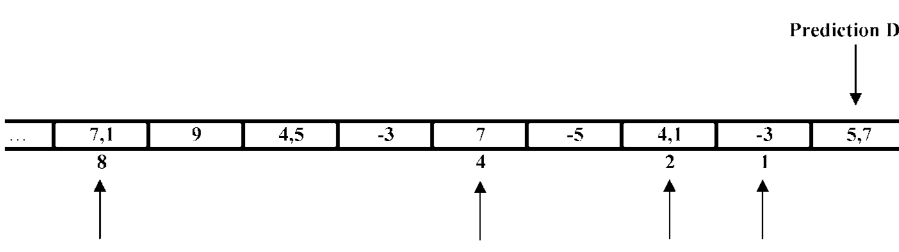

Figure A.3. Temporal window.

for these kinds of methods. Following this approach, all the methods consider four input variables.

The fuzzy partition employed for inputs and output has three labels. Two kinds of rule bases have been employed: the one obtained with the Wang and Mendel method[22] composed of 21 rules, and the one composed of 62 rules achieved with Thrift learning method.[23] Both have been obtained from a training data set of 74 examples. Evaluation of the fuzzy models has been carried out with data set of 213 examples.