A Study on the Evolutionary Adaptive Defuzzification Methods in Fuzzy Modeling*

Oscar Cordón, Francisco Herrera

Dept. of Computer Science and Artificial Intelligence
University of Granada
18071 - Granada, Spain
ocordon,herrera@decsai.ugr.es

Francisco Alfredo Márquez, Antonio Peregrín

Dept. of Electronic Engineering, Computer Systems and Automatics
University of Huelva
Huelva, Spain
alfredo.marquez, peregrin@diesia.uhu.es

Abstract. Evolutionary Adaptive Defuzzification Methods are a kind of defuzzification methods based on using a parametrical defuzzification expression tuned with evolutionary algorithms. Their goal is to increase the accuracy of the fuzzy system without loosing its interpretability. They induce a kind of rule cooperation in the defuzzification interface.

1

This paper deals with Evolutionary Adaptive Defuzzification Methods. We study their common general expression, the different defuzzification methods that can be obtained from it, their interpretation, and their accuracy. We consider two applications in order to analyse their accuracy in practice. We get some useful results for practical fuzzy systems designed by means of this kind of Intelligent Hybrid System.

Keywords: fuzzy modeling, adaptive defuzzification methods, evolutionary algorithms.

1 Introduction

Fuzzy Modeling (FM) designers try to find a trade-off between two edges: higher interpretability with lower accuracy or lower interpretability with higher accuracy: Mamdani fuzzy systems versus TSK fuzzy systems. Nowadays, there is an increasing interest in the study of the trade-off between interpretability and accuracy in FM (Casillas, Cordón, Herrera and Magdalena 2003a, b).

A way to find a balance between interpretability and accuracy is to employ parametric expressions for the fuzzy system components in order to tune the behaviour of the system. Most of the parametric fuzzy system components literature are related to the Defuzzification Interface. This point is not singular since the interest in defuzzification methods in the framework of fuzzy systems design has been always significant. Nowadays, the interest is still considerable, see (Leekwijck and Kerre 1999, Kandel and Friedman 2000, Wang and Luoh 2000, Roychowdhury and Pedrycz 2001, Van Van Leekwijck and Kerre 2001, Grzegorzewski 2001, Patel and Mohan 2002, Oussalah 2002, Ma, Roventa and Spircu 2003). Historically, there have been many tendencies in the development of defuzzification methods: initially, most of the methods were based on the fuzzy set geometry (Hellendoorn and Thomas 1993) or statistical interpretations (Wierman 1997); later, parametric or adaptive defuzzification methods were proposed (Filev 1991, Filev and Yager 1993, Yager and Filev 1993, Bastian 1995, Jiang and Li 1996, Jin and von Seelen 1999, Kiendl 1997, Kandel and Friedman 1998, Esogbue, Song and Hearnes II 2000). Adaptive methods employ one or more parameters in their expression in order to modify the behavior of the defuzzifier or, in most cases, to get a higher accuracy. In the most of the previous contributions, the values of the mentioned parameters were fixed by the authors in order to get a well known behavior, empirically determined or tuned with simple algorithms. Recently, some

[1]*This work has been supported by Spanish MCyT under project TIC2002-04036-C05-01 and FEDER found.

Reference: O. Cordón, F. Herrera, F.A. Márquez, A. Peregrin, A Study on the Evolutionary Adaptive Defuzzification Methods in Fuzzy Modelling. International Journal of Hybrid Intelligent Systems 1:1 (2004) 36-48.

contributions consider evolutionary algorithms (Bäck, Fogel, and Michalewicz 1997) to adapt the parameters of the adaptive defuzzification methods (Hong 1995, Song and Leland 1996, Jin and von Seelen 1999, Kim 2000, Huang, Gedeon and Wong 2001, Kim, Choi and Lee 2002). We call them as *Evolutionary Adaptive Defuzzification (EAD) Methods*.

This work deals with EAD methods, their general expression, the EAD methods derived in the specific literature as well as other suitable ones, and presents an empirical study to analyse their accuracy with two different applications. We also study the interpretation of the parametric expressions.

In order to do that, this paper is organised as follows: Section 2 introduces the general common expression of adaptive defuzzification methods, the different defuzzification methods that can be obtained from it and an initial study of the interpretation of the parameters for the different approaches. Section 3 introduces the EAD methods and describes the evolutionary algorithms employed. Section 4 is devoted to the experimental study with two different applications. Finally, Section 5 points out some concluding remarks. Three appendices are also included presenting the rule connectives selected for the experimental study (Appendix A), the description of the two considered applications (Appendix B), and some partial empirical results (Appendix C).

2 Adaptive Defuzzification Methods

In this section, the general parametric EAD expression is shown as well as the defuzzification methods that could be considered. Some of these methods have been already introduced in the specialised literature by different authors. Finally, we present an interpretation of the parametric defuzzification approaches.

2.1 General expression of adaptive defuzzification methods

In this contribution, we consider Mamdani-type IF - THEN rules of the following form:

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

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.

In order to get the crisp output, the most employed technique in practice is to separately defuzzify every rule inferred fuzzy set and to compute then an average. This way of working is named Mode B – FITA (First Infer, Then Aggregate) (Cordon, Herrera and Peregrín 2000). The expression is:

$$y_0 = \frac{\sum_i^N h_i \cdot V_i}{\sum_i^N h_i}, \tag{1}$$

where h_i is the matching degree between the input variables and the rule antecedent fuzzy sets, and V_i represents a characteristic value of the fuzzy set inferred from rule R_i, the Maximum Value (MV_i) or the Gravity Center (GC_i). Their particular expressions are:

- MV of a fuzzy set B':

  $y_1 = \text{Min}\{z / \mu_{B'}(z) = \text{Max } \mu_{B'}(y)\}$

  $y_2 = \text{Max}\{z / \mu_{B'}(z) = \text{Max } \mu_{B'}(y)\}$

$$y_0 = \frac{y_1 + y_2}{2} \tag{2}$$

- GC of the fuzzy set B'.

$$y_0 = \frac{\int_Y y \cdot \mu_{B'}(y)\,dy}{\int_Y \mu_{B'}(y)\,dy} \tag{3}$$

The general expression that generates some parametric defuzzification methods is:

$$y_0 = \frac{\sum_i^N f(h_i) \cdot V_i}{\sum_i^N f(h_i)}, \tag{4}$$

where $f(h_i)$ is a functional of the matching degree.

The functional term can be defined with a single parameter, α or with a set of parameters α_i corresponding to one parameter for each rule R_i, i=1 to M, in the knowledge base. Moreover, the functional term could be defined as a product or as a power among other possible functions.

In this paper, combining both functional operators and the aforementioned single or several parameters fashion, the functional term could take any of these four forms:

$$f(h_i) = h_i \cdot \alpha, \quad f(h_i) = h_i^{\alpha} \tag{5}$$

$$f(h_i) = h_i \cdot \alpha_i, \quad f(h_i) = h_i^{\alpha_i} \tag{6}$$

However, it doesn't make sense to consider the form $f(h_i) = h_i \cdot \alpha$ as the effect of α is cancelled in the final expression.

2.2 Derived defuzzification methods

Combining the three aforementioned possibilities with the two characteristic values, MV or GC, we obtain six cases. Below, we show the expressions of the defuzzification methods obtained (noted as D_3 to D_8). We have added as D_1 and D_2 the well known non adaptive defuzzification methods expressions named MV and GC weighted by the matching respectively which can be taken as the non parametrical version of the adaptive methods considered.

- D_1, MV weighted by h_i:

$$y_0 = \frac{\sum_i^N h_i \cdot MV_i}{\sum_i^N h_i}, \tag{7}$$

- D_2, GC weighted by h_i:

$$y_0 = \frac{\sum_i^N h_i \cdot CG_i}{\sum_i^N h_i}, \tag{8}$$

- D_3:

$$y_0 = \frac{\sum_i^N h_i^{\alpha} \cdot MV_i}{\sum_i^N h_i^{\alpha}}, \tag{9}$$

- D_4:

$$y_0 = \frac{\sum_i^N h_i^{\alpha} \cdot CG_i}{\sum_i^N h_i^{\alpha}}, \tag{10}$$

- D_5:
- D_6:

$$y_0 = \frac{\sum_i^N h_i \cdot \alpha_i \cdot MV_i}{\sum_i^N h_i \cdot \alpha_i}, \qquad (11)$$

$$y_0 = \frac{\sum_i^N h_i^{\alpha_i} \cdot MV_i}{\sum_i^N h_i^{\alpha_i}}, \qquad (12)$$

- D_7:

$$y_0 = \frac{\sum_i^N h_i \cdot \alpha_i \cdot CG_i}{\sum_i^N h_i \cdot \alpha_i}, \qquad (13)$$

- D_8, *Accurate GC*:

$$y_0 = \frac{\sum_i^N h_i^{\alpha_i} \cdot CG_i}{\sum_i^N h_i^{\alpha_i}}. \qquad (14)$$

The functional term $f(h_i) = h_i \cdot \alpha_i$ has been employed in the specialised literature together with a learning algorithm based on gradient descent (Pal and Pal 1999). In the same way the *Accurate GC* (Kim 2000, Kim, Choi and Lee 2002) (D_8) was proposed together with an evolutionary algorithm to learn its parameters.

2.3 Interpretation of the parametric defuzzification methods. Relationship with other approaches

The role of the individual parameter α is interpreted as a modulation of the matching influence, which can be improved or attenuated. We should note that this modulation is only linear for the product case.

The interpretation is quite different when we use one parameter for each rule of the Knowledge Base. Instead of a global modulation of the matching influence, we are changing the local action of each rule defuzzified with a product or a power functional. We are going to show the difference between each of these functional terms as follows.

The product functional term with a different parameter for each rule has the effect of weighted rules (Cho and Park 2000, Pal and Pal 1999). The value α_i associated to rule R_i gets the meaning of how significant or important is that rule for the inference process. An improved accuracy is the system modeling goal when using this kind of rule. The following is an example of a set of weighted rules, where the weights are w_i:

If X_{11} is A_{11} and ... and X_{1n} is A_{1n} then Y is B_1 with w_1
If X_{21} is A_{21} and ... and X_{2n} is A_{1n} then Y is B_2 with w_2
...
If X_{m1} is A_{m1} and ... and X_{mn} is A_{mn} then Y is B_m with w_m

The rule weight adaptation process will produce a rule subset with better cooperation among the rules composing it. This fact could be of special interest when the rule set have been generated employing a quick data-driven fuzzy rule generation method. These methods usually look for the best individual rule performance, and generate a rule base with a low cooperation degree. Employing the product functional and a parameter learning process will be equivalent to look for a subset of rules with the best global cooperation.

As regards the power functional case, the effect on defuzzification is equivalent to one of the well known mechanisms for modifying the linguistic meaning of the rule structure, the use of linguistic modifiers (Cordón, del Jesus and Herrera 1998, Liu, Chen and Tsao 2001). 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 defuzzifier parameter plays the same role changing the shape of the membership function associated with the linguistic label antecedents of the rule, as shown in Figure 1, where h is the matching for the trapezoidal fuzzy set when the input value is e. We must point out that this effect does not modify the shape of the inferred fuzzy set because the matching is only modified for defuzzification effects.

- When the fuzzy set is modified by power values greater than one, the membership function is concentrated (Zadeh 1973). The modified matching will be h' in Figure 1. Examples of these kind of linguistic modifiers are *absolutely*, *very*, *much more*, *more* and *plus* (Huang, Chen and Liu 1995, Huang, Chen and Liu 1999).

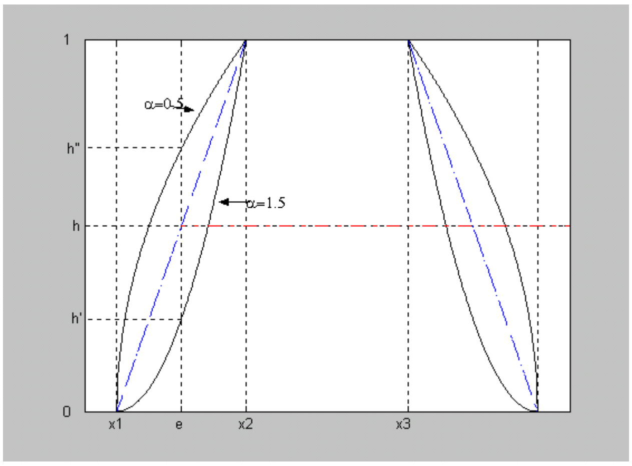

Figure 1. Graphical representation of the effect produced by the power based linguistic modifier on the defuzzification process.

- On the opposite, when the fuzzy set is modified with values below one, the membership function is expanded or dilated (Zadeh 1973). Observing Figure 1, the modified matching will now be h". Sometimes, these linguistic modifiers are named as minus, more or less and slightly (Huang, Chen and Liu 1995, Huang, Chen and Liu 1999).

Consequently, when considering a parameter for each rule, both functional terms taken into account change the structure of the individual rules of the Knowledge Base, using weights or modifying the linguistic meaning of the membership functions involved in the rules for defuzzification effects.

3. Evolutionary Adaptive Defuzzification Methods

As said, we call EAD Methods to adaptive defuzzification methods where evolutionary algorithms are employed tune the parameters.

In this contribution, two different evolutionary algorithms are considered to put this task into effect:

- (1+1)-Evolutionary Strategy, (1+1)-ES (Bäck and Schwefel 1995), for the single parameter functional case, and
- CHC algorithm (Eshelman 1991) with real coding for the functional that uses an individual parameter for each rule of the rule base.

Important differences between the amount of variables to optimise make us employ the two aforementioned different evolutionary algorithms. On the one hand, when we deal with the adaptation of a single parameter (D_3 and D_4), a local search with high efficiency is performed by the (1+1)-ES evolutionary model. On the other hand, when several α_i have to be derived, i.e. the search space is composed of M parameters, one for each rule in the rule base, (D_5, D_6, D_7 and D_8), the CHC algorithm is considered as an evolutionary model with a good trade-off between diversity and convergence in high-dimensional search spaces.

Evolutionary Strategies (Bäck and Schewefel 1995) were initially developed by Rechenberg and Schwefel in 1964 as experimental optimisation techniques. The first ES algorithm, the so-called (1+1)-ES, was based on working with only two individuals per generation, one parent and one descendent. (1+1)-ES is based on encoding the possible optimisation problem solution into a real-coded string. This parent string is evolved by applying a mutation operator over each one of its components. The mutation strength is determined by a value σ, a standard deviation of a normally distributed random variable. This parameter is associated to the parent and it is evolved in each process step as well. If the evolution has been performed successfully, the offspring obtained by mutation is better adapted than its parent, then the descendent substitutes it in the next generation. The individual adaptation is measured by using a fitness function. The process is iterated until a determined finishing condition is satisfied.

On the other hand, during each generation, the CHC algorithm (Eshelman 1991) 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 being generated which are better than any member of the parent population), the population is reinitialised. The restart population completely consists of random individuals but one instance of the best individual found so far (Eshelman, Mathias and Schaffer 1997).

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 (α =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 taking into account if each character is different or not. Only those string pairs which 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.

4 Experimental Study

In this section, we present the experimental study developed in order to study the accuracy of the EAD methods considered. We will build several fuzzy models combining the defuzzification methods presented in Section 2.2.2 with a representative set of rule connectives (see Appendix A) to solve the two different applications described in Appendix B.

4.1 Comparison methodology

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

$$MSE\ (i,j) = \frac{\frac{1}{2} \sum_{k=1}^{M} (y_k - S[i,j](x_k))^2}{N} \tag{15}$$

where *S[i,j]* denotes the fuzzy model whose Inference System uses the rule connective I_i, and whose Defuzzification Interface is based on the defuzzification method D_j. This measure employs a set of system evaluation data formed by N pairs of numerical data $Z_k=(x_k, y_k)$, $k=1,...,N$, with x_k being the values of the input variables, and with y_k being the corresponding values of the associated output variables (see Appendix B).

We consider the Improvement Percentage (IP) index whose expression is:

$$IP(i,j) = 100 \times \left(1 - \frac{MSE(S(i,j))}{MSE(S_{NA}(i,k))}\right) \tag{16}$$

that is, the improvement shown by the MSE(·) of a fuzzy model S(·) built with a rule connective I_i and a specific defuzzification method D_j with respect to the system without tunable parameters, $S_{NA}(i,k)$, where *k* is 1 for the MV or 2 for the GC expressions according to the characteristic value used by the respective parametric one.

The fuzzy models considered employ the aforementioned eight defuzzification methods combined with the seven rule connectives described in the Appendix A. The conjunction operator was always the minimum t-norm, a robust conjunction operator as showed in (Cordón, Herrera and Peregrín 1997).

The FM applications used for this analysis are *the estimation of the low voltage network real length in villages* (called E_1) *and the estimation of the electrical medium voltage network maintenance cost in towns* (called E_2)

(Cordón, Herrera and Sánchez 1999). Both applications are briefly described in Appendix B. The fitness function employed was the aforementioned Mean Square Error. It has been applied over the training data set (see Appendix B).

For the (1+1)-ES, the stopping condition is not to improve the best solution found so far during 200 consecutive iterations. The CHC has been run during 20000 trials. The population size was 50 (randomly initialized with the exception of a single chromosome with all the genes initialised to 1), and a BLX-α crossover with α= 0,5 was employed as we mentioned before. The initial threshold was set to L/4, with L being the chromosome length (L=24 in E_1 and L=66 in E_2).

The searching interval for α *in* both evolutionary algorithms was fixed to [0,5]. This decision is justified by the following interval study:

- For the functional expression h_i^α (also valid for α):
  - h_i^α, $\alpha \in [1,\infty)$: soft attenuation of the h_i value,
  - h_i^α, $\alpha \in [0,1]$: soft enhancement of the h_i value.
- For the functional expression $\alpha \cdot h_i$:
  - $\alpha \cdot h_i$, $\alpha \in [1,\infty)$: strong enhancement of the h_i value,
  - $\alpha \cdot h_i$, $\alpha \in [0,1]$: strong attenuation of the h_i value.

The interval [0,5] allows us to attenuate as well as enhance the matching degree. In the (1+1)-ES, the attenuation is reasonably limited but the searching interval is also reduced, so the accuracy will be benefited with a lower number of iterations. The aforementioned interval reduces the searching interval, so the speed of convergence will be better.

The initial values for the parameters are equal to 1, that is, equivalent to the original non parametric defuzzification method.

We also achieved five trials for every fuzzy model tuning process, running them with five different seeds for the random number generator. Thus, the MSE considered was computed as the arithmetic mean of the five results.

4.2 Results and analysis

Tables 9 and 10 in Appendix C show the MSE obtained for applications E_1 and E_2 respectively, employing the non parametric defuzzification methods, D_1 and D_2. The values shown in those tables are the reference in order to measure the improvements of the EAD methods.

Tables 1 and 2 show the IP obtained for the single parameter defuzzification methods, for the training and test data sets.

Table 1. IP of the MSE for the FM of E_1 with single parameter EAD methods.

	Training		**Test**	
R.C.	**D_3**	**D_4**	**D_3**	**D_4**
I_1	0,90123	10,35227	4,66922	7,79387
I_2	1,66281	12,69176	6,72546	9,60671
I_3	1,69698	1,67122	6,78149	6,78725
I_4	1,69698	14,72528	6,78149	11,12842
I_5	1,69698	0,66757	6,78149	3,97103
I_6	1,66281	0,25658	6,72546	2,51457
I_7	1,66281	0,00555	6,72546	-0,26881

Tables 3 and 4 for application E_1 and Tables 5 and 6 for application E_2 show the IP of the MSE of multiple parameter EAD methods, while Tables 3 and 5 do so for the MV based methods and Tables 4 and 6 for the GC based ones.

Table 2. IP of the MSE for the FM of E_2 with single parameter EAD methods.

	Training		Test	
R.C.	**D_3**	**D_4**	**D_3**	**D_4**
I_1	8,06282	28,76866	9,29941	27,38372
I_2	3,88982	34,48842	6,05165	33,07418
I_3	7,77523	7,79119	10,15062	10,18228
I_4	7,77523	39,43767	10,15062	37,97787
I_5	7,77523	19,25580	10,15062	18,31686
I_6	3,88982	19,25580	6,05165	18,31686
I_7	3,88982	49,70895	6,05165	44,05140

Table 3. IP of the MSE for the FM of E_1 with multiple parameter AED methods based on the MV.

	Training		Test	
R.C.	**D_5**	**D_6**	**D_5**	**D_6**
I_1	14,18820	18,10597	7,86923	7,79620
I_2	14,44577	20,10201	8,80105	5,14748
I_3	14,15707	20,07103	2,65990	6,31867
I_4	14,15707	20,07103	2,65990	6,31867
I_5	14,15707	20,07103	2,65990	6,31867
I_6	14,44577	20,10201	8,80105	5,14748
I_7	14,44577	20,10201	8,80105	5,14748

Table 4. IP of the MSE for the FM of E_1 with multiple parameter AED methods based on the GC.

	Training		Test	
R.C.	**D_7**	**D_8**	**D_7**	**D_8**
I_1	2,18176	10,42146	1,92474	7,71026
I_2	3,00536	12,87297	2,85673	9,52972
I_3	14,17298	20,07499	7,55348	6,22153
I_4	3,80224	15,09013	3,79295	11,18154
I_5	14,40167	17,45240	8,27327	-1,01876
I_6	14,45836	16,95575	8,13563	-1,78539
I_7	16,03466	16,57746	6,91716	1,96117

Table 5. IP of the MSE for the FM of E_2 with multiple parameter EAD methods based on the MV.

	Training		Test	
R.C.	**D_5**	**D_6**	**D_5**	**D_6**
I_1	56,63170	61,39146	61,38098	65,17136
I_2	54,66390	58,43074	60,19666	63,73250
I_3	55,33593	59,83568	60,54732	64,61813
I_4	55,33593	59,83568	60,54732	64,61813
I_5	55,33593	59,83568	60,54732	64,61813
I_6	54,66390	58,43074	60,19666	63,73250
I_7	54,66390	58,43074	60,19666	63,73250

Table 6. IP of the MSE for the FM of E_2 with multiple parameter EAD methods based on the GC.

	Training		Test	
R.C.	**D_7**	**D_8**	**D_7**	**D_8**

I_1	24,06131	28,88662	21,65394	27,44477
I_2	29,12648	34,86772	27,00117	33,41484
I_3	55,24824	59,93250	60,59673	64,81639
I_4	33,72583	40,14250	31,75977	38,71634
I_5	61,24516	64,84529	64,39076	67,25277
I_6	61,23801	64,85135	64,42411	67,18371
I_7	72,04719	77,80156	72,00969	77,23946

Finally, Table 7 shows the global arithmetical means of the IP for the different functional terms for both applications.

Table 7. Arithmetical means of the IP for the different functional terms.

	Training		**Test**	
	E_1	**E_2**	**E_1**	**E_2**
h^{α}	3,66792	17,26889	6,19451	17,65781
$h \cdot \alpha_i$	12,00384	51,66596	5,83614	54,67494
h^{α_i}	17,71930	56,25130	5,42820	59,02082

Analysing the results, we can point out that:

- First, it is noticeable that the EAD methods improve the two classical well known defuzzification methods taken as references. Thus, they are a good option in order to get improved accuracy in Linguistic FM keeping an important interpretability level.

- The functional term with a single parameter for the whole expression presents worse results than the functional terms with a parameter for each rule. When a single parameter is considered, we are adapting the influence of the matching in the defuzzification method expression in a non linear way.

  The option with more degrees of freedom, a parameter per rule, allows us obtain more accuracy, but we must take into account that we are extending the Knowledge Base structure.

- The use of individual parameters for each rule could be accomplished with two different functional terms: power and product. Power has presented larger improvements than product. Therefore, in our experimentation with the two mentioned applications, the fact of modifying the linguistic meaning of the antecedents of each rule **for defuzzification effects** by the power function offers better improvements than the weighted rules associated to the product function.

5 Concluding Remarks

Some authors have introduced EAD methods in order to improve the accuracy of Mamdani FM with a low lost of interpretability. In this work, we have studied their common expression, the interpretability of employing these parameters in the fuzzy system, and we have presented empirical results of the performance of several EAD methods.

Depending on the way the modifying parameters are introduced, they can modulate the influence of the matching degree, or extend the Knowledge Base structure in several ways.

It was empirically shown that the single parameter functional presents lower improvements in the accuracy than the cases with a parameter for each rule. Nevertheless, the single parameter model keeps a better interpretability.

The experimentation carried out in this study shows the high improvement of accuracy of the AED methods. Their use may improve the performance of any fuzzy system as the AED method specifically adapts the Defuzzification Interface to the specific problem.

As regards the two functional approaches, power and product, we got slightly better results with the power function, but it would be necessary to make a deeper study with more applications to get a sound conclusion.

References

Bäck, T. and Schwefel, H.-P. (New York 1995), "Evolution Strategies I: variants and their computational implementation", *J.Periaux, G. Winter, M. Galán, P. Cuesta (Eds.), Genetic Algorithms in Engineering and Computer Science, Wiley,* pp. 111-126.

Bäck T., Fogel, D. and Michalewicz, Z. (1997), "Handbook of evolutionary computation", *Oxford University Press*.

Bastian, A. (1995), "Handling the nonlinearity of a fuzzy logic controller at the transition between rules", *Fuzzy Sets and Systems,* vol. 71, pp. 369-387.

Casillas, J., Cordón, O., Herrera, F. and Magdalena, L. (Eds.) (2003), "Interpretability issues in fuzzy modeling", *Springer-Verlag*.

Casillas, J., Cordón, O., Herrera, F. and Magdalena, L. (Eds.) (2003), "Accuracy improvements in linguistic fuzzy modeling", *Springer-Verlag*.

Cho, J.-S. and Park, D.-J. (2000), "Novel fuzzy logic control based on weighting of partially inconsistent rules using neural network", *Journal of Intelligent and Fuzzy Systems,* vol. 8, pp. 99-100.

Cordón, O., Herrera, F. and Peregrín, A. (1997), "Applicability of the fuzzy operators in the design of fuzzy logic controllers", *Fuzzy Sets and Systems* vol. 86, pp. 15-41.

Cordón, O., del Jesus, M.J. and Herrera, F. (1998), "Genetic learning of fuzzy rule-based classification systems cooperating with fuzzy reasoning methods", *International Journal of Intelligent Systems* vol. 13, pp. 1025-1053.

Cordón, O., Herrera, F. and Sánchez, L. (1999), "Solving electrical distribution problems using hybrid evolutionary data analysis techniques", *Applied Intelligence* vol. 10, pp. 5-24.

Cordón, O., Herrera, F. and Peregrín, A. (2000), "Searching for basic properties obtaining robust implication operators in fuzzy control", *Fuzzy Sets and Systems,* vol. 111, pp. 237-251.

Dubois, D. and Prade, H. (1991), "Fuzzy sets in approximate reasoning, Part 1: Inference with possibility distributions", *Fuzzy Sets and Systems,* vol. 40, pp. 143-202.

Eshelman, L.J. (San Mateo 1991), "The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination", *Foundations of Genetic Algorithms-1, Rawlins,G.J.E. (Eds.) (Morgan Kaufmann Publishers*), pp. 265-283.

Eshelman, L.J., Mathias, K.E. and Schaffer, J.D. (San Mateo 1997), "Convergence controlled variation", *Foundations of Genetic Algorithms 4, R. Belew and M. Vose (Eds.) (Morgan Kaufmann Publishers)*, pp. 203-224.

Esogbue, A.O., Song, Q. and Hearnes II, W.E, (2000), "Defuzzification filters and applications to power system stabilization problems", *Journal of Mathematical Analysis and Applications,* vol. 251, pp. 406-432.

Filev, D.P., (1991), "A generalized defuzzification method via BAD distributions", *International Journal of Intelligent Systems,* vol. 6, pp. 687-697.

Filev, D.P. and Yager, R.R., (1993), "An adaptative approach to defuzzification based on level sets", *Fuzzy Sets and Systems,* vol. 54, pp. 355-360.

Grzegorzewski, P. (2001), "Fuzzy tests – defuzzification and randomization", *Fuzzy Sets and Systems,* vol. 118 - 3**,** pp. 437-446.

Gupta, M.M. and Qi, J. (1991), "Theory of T-norms and fuzzy inference methods", *Fuzzy Sets and Systems,* vol. 40, pp. 431-450.

Hellendoorn, H. and Thomas, C. (1993), "Defuzzification in Fuzzy Controllers", *Journal of Intelligent and Fuzzy Systems,* vol. 1, pp. 109-123.

Hong, C.Z. (1995), "Handwritten Numeral Recognition Using a Small Number of Fuzzy Rules With Optimized Defuzzification Parameters", *Neural Networks,* vol. 8-5, pp. 821-827.

Huang, C.Y., Chen, C.Y. and Liu, B.D. (1995), "Current-mode linguistic hedge circuit for adaptive fuzzy logic controllers", *Electronic Letter,* vol. 31, pp. 1517-1518.

Huang, C.Y., Chen, C.Y. and Liu, B.D. (1999), "Current-mode fuzzy linguistic hedge circuits", *Analog Integrated Circuits Signal Process,* vol. 19, pp. 225-278.

Huang, Y., Gedeon, T.D. and Wong , P.M. (2001), "An integrated neural-fuzzy-genetic-algorithm using hyper-surface membership functions to predict permeability in petroleum reservoirs", *Engineering Applications of Artificial Intelligence,* vol. 14-1, pp. 15-21.

Jiang, T. and Li, Y. (1996), "Generalized defuzzification strategies and their parameter learning procedures", *IEEE Transactions on Fuzzy Systems,* vol. 4, pp. 64-71.

Jin, Y. and von Seelen, W. (1999), "Evaluating flexible fuzzy controllers via evolution strategies", *Fuzzy Sets and Systems,* vol. 108, pp. 243-252.
Kandel, A. and Friedman, M. (1998) "Defuzzification using most typical values", *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 28, pp. 901-906.
Kiendl, H. (Barcelona 1997), "Non - translation - invariant defuzzification", *Proceedings of the 6th FUZZ-IEEE'97,* pp. 737-741.
Kim, D. , (San Antonio 2000), "An accurate COG dDefuzzifier design by the co-adaptation of learning and evolution", *Proc. 9th IEEE International Conf. on Fuzzy Systems,* pp. 741-747.
Kim, D., Choi, Y. and Lee, S. (2002), "An accurate COG defuzzifier design using Lamarckian co-adaptation of learning and evolution", *Fuzzy Sets and Systems*, vol. 130-2, pp. 207-225.
Liu, B.-D., Chen, C.-Y. and Tsao, J.-Y. (2001), "Design of adaptative fuzzy logic controller based on linguistic-hedge concepts and genetic algorithms", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 31, pp. 32-53.
Ma, M., Kandel, A. and Friedman, M., (2000), "A new approach for defuzzification", *Fuzzy Sets and Systems,* vol. 111, pp. 351-356.
Magrez, P. and Smets, P. (1989), "Fuzzy modus ponens: A new model suitable for applications in knowledge-based systems", *International Journal of Intelligent Systems,* vol. 4, pp. 181-200.
Oussalah, M., (2002) "On the compatibility between defuzzification and fuzzy arithmetic operations", *Fuzzy Sets and Systems,* vol. 128, pp. 247-260.
Pal, N. R. and Pal, K., (1999), "Handling of inconsistent rules with an extended model of fuzzy reasoning", *Journal of Intelligent and Fuzzy Systems,* vol. 7, pp. 55-73.
Patel, A. V. and Mohan, B. M., "Some numerical aspects of center of area defuzzification method", *Fuzzy Sets and Systems*, vol. 132, pp. 401-409.
Roventa, E. and Spircu, T., "Averaging procedures in defuzzification processes", *Fuzzy Sets and Systems,* In Press.
Roychowdhury, S. and Pedrycz, W. (2001), "A survey of defuzzification strategies", *International Journal of Intelligent Systems,* vol. 16, pp. 679-695.
Song, Q. and Leland, R. P. (1996), "Adaptive learning defuzzification techniques and applications", *Fuzzy Sets and Systems,* vol. 81-3, pp. 321-329.
Van Leekwijck, W. and Kerre, E.E., (1999), "Defuzzification: criteria and classification", *Fuzzy Sets and Systems,* vol. 108, pp. 159-178.
Van Leekwijck, W. and Kerre, E.E. (2001), "Continuity focused choice of maxima: Yet another defuzzification method", *Fuzzy Sets and Systems,* vol. 122, pp. 303-314.
Wang, W.J. and Luoh, L. (2000), "Simple computation for the defuzzifications of center of sum and center of gravity", *Journal of Intelligent and Fuzzy Systems,* vol. 9, pp. 53-59.
Wang, L.X. and Mendel, J.M. (1992), "Generating fuzzy rules by learning from examples", *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 22, pp. 1414-1427.
Wierman, M.J., (1997) "Central values of fuzzy numbers", *Information Sciences*, vol. 100, pp. 207-215.
Yager, R.R. and Filev, D.P. (1993), "SLIDE: a simple adaptive defuzzification method", *IEEE Transa. Fuzzy Systems,* vol. 1, pp. 69-78.
Zadeh, L.A. (1973), "Outline of a new approach to the analysis of complex systems and decision processes", *IEEE Transactions on Systems, Man and Cybernetics,* vol. 3, pp. 28-44.

Appendix A: Rule Connectives

The rule connectives used are shown in this appendix. The different names for these operators are to be found in (Dubois and Prade 1991, Magrez and Smets 1989).

Implication Functions:

S-Implications :

Kleene-Dienes : $\mathbf{I_1\ (x,y)} = \text{Max}\ (1-x,\ y)$ (17)

Reichenbach-Mizumoto : $\mathbf{I_2\ (x,y)} = 1-x + x\cdot y$ (18)

R-Implications :

Gödel : $$I_3(x,y) = \begin{cases} 1, & \text{if } x \le y \\ y, & \text{otherwise} \end{cases} \quad (19)$$

S and R-Implications :

Lukasiewicz : $\mathbf{I_4}$ **(x,y)** = Min (1, 1-x+y) (20)

T-norms: We use the following t-norms as rule connectives (Cordón, Herrera and Peregrín 1997, Gupta and Qi 1991):

Logical Product (Minimum) : $\mathbf{I_5}$ **(x,y) =** Min (x, y) (21)

Hamacher Product : $I_6(x,y) = \dfrac{x \cdot y}{x+y-x \cdot y}$ (22)

Algebraic Product : $\mathbf{I_7}$ **(x,y)** = x·y (23)

Appendix B: Applications

Two electrical distribution problems described in (Cordón, Herrera and Sánchez 1999) have been selected to analyse the performance of the EAD 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.**

***E_1* Application**: *T*he 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 2, 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, and
EL is extremely large.

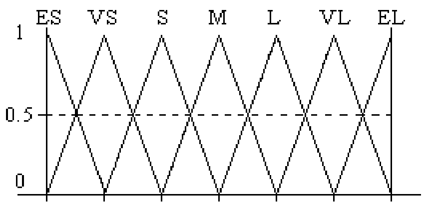

Figure 2. Fuzzy partition considered for the input and output variables of E_1

The rule base, composed of 24 linguistic rules shown in Table 8, has been obtained with the Wang and Mendel method (Wang and Mendel 1992), from a data training set of the 80% of the original available data, that is, 396 villages taken randomly.

Table 8. Rules of application E_1

x_2

x_1		ES	MS	S	M	L	ML	EL
		ES	*MS*	*MS*	*S*		*MS*	*M*
	MS	*ES*	*MS*	*MS*	*L*	*S*	*L*	
	S		*MS*	*M*	*S*	*S*	*L*	
	M		*MS*	*S*	*ML*	*EL*	*M*	

L			*M*				
M L							
EL				*S*			

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

***E_2* Application:** The second electrical distribution problem, E_2, has got 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, while the output variable domain is [0, 8546.030273]. The fuzzy partition employed for inputs and output has 5 labels {MP, P, M, G, MG} (see Figure 3), where:

MS is very small, L is large,
S is small, VL is very large.
M is middle,

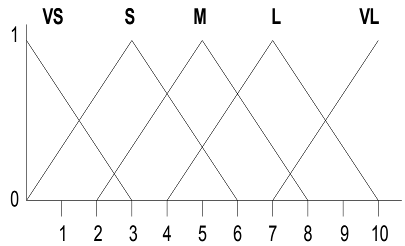

Figure 3. Fuzzy partition considered for the input and output variables of E_2

The rule base comprised by 66 linguistic rules has also been obtained with the Wang and Mendel method (Wang and Mendel 1992), from a training data set of the 80% of the original available data, that is, 847 cities taken randomly. The evaluation of the fuzzy models designed have been carried out with the remaining 20% of the initial data set, that is, with data from 212 cities.

Appendix C: Partial Results

Table 9. MSE for the FM of E_1 with the non adaptive defuzzification methods.

	Training		Test	
	D_1	**D_2**	**D_1**	**D_2**
I_1	222191,82	2427093,12	222191,82	2427093,12
I_2	221168,97	2217053,76	221168,97	2217053,76
I_3	222764,50	222484,08	222764,50	222484,08
I_4	222764,50	2037621,07	222764,50	2037621,07
I_5	222764,50	223017,45	222764,50	223017,45
I_6	221168,97	224996,06	221168,97	224996,06
I_7	221168,97	228513,82	221168,97	228513,82

Table 10. MSE for the FM of E_2 with the non adaptive defuzzification methods.

	Training		Test	
	D_1	**D_2**	**D_1**	**D_2**
I_1	75413,91	3241108,74	83764,27	3087677,08
I_2	72178,38	2942541,29	81879,53	2800832,95
I_3	71483,38	71381,04	81322,35	81151,32
I_4	71483,38	2691268,11	81322,35	2559428,33
I_5	71483,38	81133,58	81322,35	87556,85
I_6	72178,38	81133,58	81879,53	87556,85
I_7	72178,38	128974,71	81879,53	124407,99