Boosting of fuzzy models for high-dimensional imprecise datasets

Luciano Sánchez
Universidad de Oviedo
luciano@uniovi.es

José Otero
Universidad de Oviedo
jotero@lsi.uniovi.es

José Ramón Villar
Universidad de Oviedo
villarjose@uniovi.es

Abstract

Boosting and backfitting techniques are between the fastest fuzzy rule learning techniques, and therefore are well suited for high-dimensional datasets. In this paper we propose an extension of these last methods, that can be applied to learn rule-based models from interval and fuzzy valued data. The learning will depend on the minimum of a fuzzy valued fitness function, which will be optimized by means of a multicriteria, population based, simulated annealing method.

Keywords: Boosting, Imprecise Data, Backfitting.

1 Introduction

Genetic Learning of fuzzy rules produces linguistically understandable models, but the computational cost involved is much higher than the needed to obtain a statistical or neural model of comparable accuracy. This complexity increases with the size of the datasets, either with the number of examples or the number of features, preventing the use of GFSs in many practical problems.

In previous works, we have suggested to exploit the similarities between certain class of fuzzy models and Extended Additive Models [10] to apply boosting techniques in combination with GFSs [25]. The boosting of fuzzy models has an speed comparable to that of heuristic methods, and can be applied to a meaningful range of Additive Fuzzy Models. This kind of models has been long used by the fuzzy community [16], mostly in connection with neuro-fuzzy techniques [14, 18, 30] and more recently in Support Vector Machines related works [11, 3, 4].

An additional problem is rooted to the application of GFSs to *crisp* datasets. Given that one of the main objectives of the fuzzy techniques is to process imprecise information, in our opinion the fuzzy models should be learnt and evaluated over fuzzy data. There are also theoretical arguments favoring the use of a fuzzy valued fitness function in the context of these learning techniques [26, 28]. But the evaluation of a fuzzy valued function has an important overhead in terms of the number of calculations. We are not aware of optimization techniques than can find, in a reasonable time, the minimum of a fuzzy valued fitness function so complex as the one that originates in rule learning problems from high dimensional imprecise datasets.

This paper is dedicated to the study of this last problem, the efficient learning of fuzzy rules from imprecise data. In the next section, we will extend the basic backfitting algorithm to deal with fuzzy data. Then, in section 3, we will discuss the problem of optimizing a fuzzy-valued fitness function, and in section 4 a new extension of a population-based, muticriteria simulated annealing is introduced. The paper finishes with a benchmark analysis of the proposed algorithm and the concluding remarks.

2 Fuzzy Extended Additive Models

In this work, we will restrict ourselves to linguistic additive fuzzy rule-based models, comprising M rules as the one that follows:

$$\texttt{If } x \texttt{ is } A_m \texttt{ then } y \texttt{ is } B_m, \qquad (1)$$

where x and y are the feature and the output vectors, respectively, and A_m are conjunctions of linguistic labels, which in turn are associated to fuzzy sets. B_m can be either a singleton or a fuzzy number. In this paper, none of the sets A_m, B_m will be modified during the learning, to preserve the linguistic interpretability, but we will admit that each rule is assigned a weight w_m. Let B'_m be the result of the inference process for the preceding rule:

$$B'_m(x, y) = I(A_m(x), B_m(y)) \qquad (2)$$

where I is a fuzzy inference operator. The output y of the fuzzy model is then computed as

$$y = G_{m=1}^{M}(w_m B'_m) \qquad (3)$$

where G is an operator that combines all the sets B'_m to produce the final output. Let us define I to be the product, and G the sum of the centroids. Then,

$$y = \sum_{m=1}^{M} D(w_m A_m(x) B_m(y)) \qquad (4)$$

where D stands for "defuzzification." If all the B_m are singletons, and the input x is crisp, the output of the fuzzy model is a real number, which can be written as

$$y(x) = \sum_{m=1}^{M} f_m(x) \qquad (5)$$

where $f_m(x) = \beta_m A_m(x)$, and A_m is a conjunction of linguistic labels taken from a predetermined set, as mentioned before.

2.1 Backfitting and Boosting

According to Friedman [10], the *boosting* is a particular case of a backfitting algorithm, applied to a classification problem that has been transformed into a regression problem by means of a logistic transform. This procedure has been previously applied to learn fuzzy models and classifiers from crisp data [7, 23, 27], and shown to be as fast as some ad-hoc learning methods [25].

For clarity, let us repeat here the method proposed in [25]. To learn the model (5) from crisp data, we require an intermediate fitting algorithm, that can select the function $f_m = \beta_m A_m$ that best fits an arbitrary set of data. Let us name "`FitOneRule()`" to this intermediate algorithm. Then, we proceed as shown in the pseudocode that follows:

```
residual[1..N] = y[1..N]
rule base = emptyset
repeat
 f = FitOneRule(x[1..N],residual[1..N])
 do i=1..N
    residual[i] = residual[i] - f(x[i])
 end do
 rule base = rule base + f
until norm(residual)<epsilon
```

In words, we first fit one rule f_1 to the train set. Then, we replace the desired output by the residuals of the output of this rule, repeat the process to obtain f_2 and so on. Observe that, following the boosting nomenclature, the use of the residual is equivalent to the assignment of certain weight to each example in the dataset, and solve the corresponding weighted squares problem. These weights would range from a perfect fit (weight 0) to an uncovered example (weight 1), so the rationale of this process can also be explained as "fit a rule to the dataset, remove the examples that are well explained by this rule and repeat until all examples are explained."

Each rule f_m consists in a pair (β_m, A_m). A_m is a linguistic expression whose terms are labels of the linguistic variables defined over the input variables, connected by the operators "AND" and "OR". β_m is the product of the centroid of B_m and the weight w_m assigned to the rule. A fuzzy rule will be obtained in every iteration, and the process finishes when the best β_m is zero or the accuracy of the model is high enough, whichever come first. Because of the limitations of space, the reader

is refered to [25] for further details in the numerical procedure needed to obtain β_m, and the details of the algorithm `FitOneRule()`.

2.2 Interval and fuzzy valued data

It is intuitive that the output of a fuzzy model, when it is fed with a fuzzy input, must be a fuzzy set. In our case, this means that the measure of similarity between the output of $\beta_m A_m$ and the residual that is minimized by the function "fit one rule" is not longer a number, but a fuzzy set. Let us suppose that all the information we are given about the input x to our model is the fuzzy set X. Then, the most we can say about the output of the m-th rule is that it is contained in the fuzzy set F_m,

$$F_m = \beta_m \otimes A_m(X). \tag{6}$$

where $A_m(X)$ is the fuzzy set

$$[A_m(X)]_\alpha = \{A_m(x) \mid x \in [X]_\alpha\}. \tag{7}$$

Following the ideas introduced in [26, 28], given two fuzzy samples $\{X_1, \ldots X_N\}$ and $\{Y_1, \ldots Y_N\}$ of the input-output data, the best rule will be the one that minimices the *fuzzy valued* function

$$\bigoplus_{n=1}^{N} \mathrm{SQ}\left(Y_n - \bigoplus_{m=1}^{M} \beta_m \otimes A_m(X_i)\right) \tag{8}$$

where $[\mathrm{SQ}(X)]_\alpha = \{x^2 \mid x \in [X]_\alpha\}$. Two remarks are made: (a) the residual of the rule is also a difference between two fuzzy numbers, that must be computed by means of fuzzy arithmetic and (b) with this algorithm it is not longer needed that the consequents B_m of the fuzzy rules are singletons.

3 Minimum of an imprecisely known function

As we pointed in [26, 28], the minimum of a fuzzy function is a problem that is actively being studied and that can not be regarded as solved. In [8] a review and a categorization of the most relevant approaches of fuzzy optimization problems is made. There are also many proposals of specific numerical fuzzy optimization algorithms (for example, Fuzzy Tabu Search [17], Evolutionary Algorithms [15] or Nonlinear Fuzzy Programming [19].) Some of these approaches are based on certain kinds of fuzzy ranking, or other heuristic criteria, that allow to compare any pair of fuzzy numbers and then to extend a scalar optimization algorithm to the fuzzy case. We will not use these criteria, but we will address the problem under the perspective of the α-Pareto dominance [1, 24], and transform the fuzzy optimization into a multicriteria problem. Our point of view will be made clear with the following example: Let us suppose that we want to compare the fuzzy error of two models, whose cuts at the α level are the intervals E_1 and E_2. Instead of applying an heuristic criteria (like, for instance, comparing the centers of E_1 and E_2) we observe that, if $E_1 \cap E_2 = \emptyset$, it is clear that either E_1 α-dominates E_2 or E_2 dominates E_1, because all the points contained in E_1 are lower that those of E_2 or vice versa. Otherwise ($E_1 \cap E_2 \neq \emptyset$), we can not know, without further assumptions, whether the unknown error value contained in E_1 is lower than that contained in E_2. It is also easy to see that a pure Pareto-based multicriteria optimization algorithm can be used to obtain a set of non-dominated solutions from which that contain the minimum of the imprecisely known function.

4 A fuzzy extension of the Simulated Annealing algorithm

We have decided to implement an hybrid between the Simulated Annealing and a Genetic Algorithm, combining the genetic operators of crossover and selection with the probabilistic search of SA, in order to obtain an evolutionary algorithm with greater possibilities of control of either the memory used and the speed of convergence. This way, we can balance the calculation time with the accuracy of the solution and better address the high-dimensional problems mentioned in the introduction.

To our knowledge, Pareto-based multiobjective simulated annealing techniques has not been widely applied. Nevertheless, multicri-

```
Select initial and final temperatures: T_0, T_1, and cooling factor :C
Select a starting point: x_0
Initialize the population of search paths: X = {x_0}
Initialize the set of elites (sample of Pareto front): P = {x_0}
T ← T_0
while T ≤ T_1
    // Initialize intermediate populations X'. P'
    X' ← X; P' ← P;
    for path ← 1 to size(X)
       x ← mutation(X_path)
       if x ≺ X_path then
          // The search point and the Pareto front are updated
          X'_path ← x; if x ≺ P_path then P'_path ← x
       else if X_path ≺ x then
          // The search point might be updated
          if rnd() < exp(-distance(X_path, x)/T) then X'_path ← x
       else
          // A new search path is generated
          X' ← X ∪ {x}; P' ← P ∪ {x};
       end if
    end for
    // If needed, the size of the set of paths is adjusted
    X ← selection(X'); P ← selection(P')
    T ← T · C
end while
```

Figure 1: Pseudocode of the MOSA algorithm

teria extensions of SA are an active research field [29, 20, 13]. In [6], Pareto-dominance was applied to guide the evolution of the simulated annealing, and the same approach was further extended in [12], where fuzzy numbers and uncertainty in dominance are used to decide whether an individual dominates other, as we will do in this paper. Similarly, in [21, 22], Pareto-dominance was also applied to judge how the multiobjective simulated annealing evolves. But, in all of the preceding algorithms, an aggregated function of objectives is used to evaluate each individual, thus the search is actually being carried in an scalar space.

A different approach to Pareto-based multiobjective simulated annealing is presented in [2], where a comparison of a Pareto-based evolutionary algorihtm and a population-based simulated annealing with dominance control approach is presented. In each simulated annealing iteration, a new individual is obtained by means of an heuristic, and it is included in the population if there is a non dominance relation with the current individual. If the new individual dominates the current one, then this last one is replaced. Otherwise, if the new individual is dominated by the current one, then it is accepted with temperature dependant probability. But, even in this last work, an aggregation function is used when the dominance is evaluated, therefore it can always be said that an heuristically generated individual either dominates or is dominated by the current one (thus collapsing the Pareto front to one point.) In this paper we drop this hypothesis and propose a new, population based, extension of the simulated annealing, where it may happen that an individual does not dominates neither is dominated by other different one.

4.1 The MOSA algorithm

The pseudocode of the Multi-Objective Simulated Annealing (MOSA) is shown in Figure 1. The algorithm is based in a variable sized population of search points. At each iteration, all the search points are mutated to form an intermediate population. Given a certain value of α, if the mutated individual

α-dominates the current search point, it replaces its parent in the intermediate population. If the mutated individual is dominated, then a random decision is made. Otherwise, the size of the intermediate population is increased, and the mutated point constitutes a new search point. A set of non-dominated solutions, of the same size as the population, is also maintained. Once all the individuals in the population have been mutated, the intermediate population is sampled to form the following generation. The codification of the solutions is the same binary coding that the one used in the GA in [25], and the operator `mutate` is the crossover with a randomly generated individual, also as mentioned in this last reference. The operators `distance`, and `select` are explained in the following subsections.

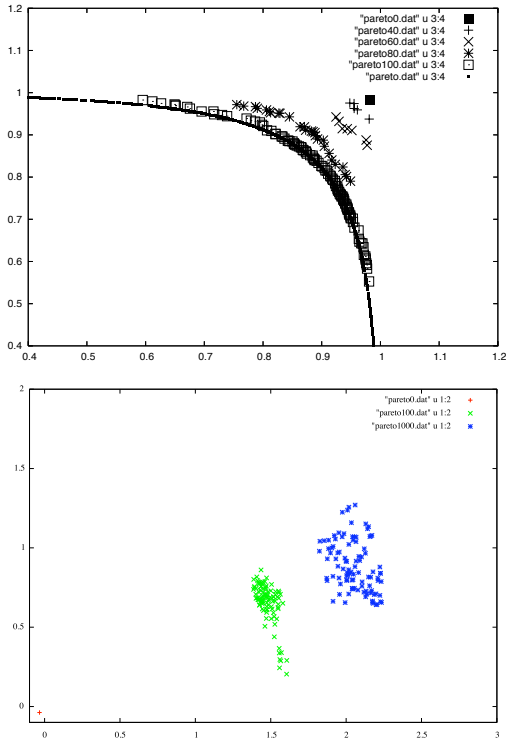

Figure 2: Evolution of the population in the MOSA algorithm, for a two-criteria problem (upper part,) and for a fuzzy optimization problem (lower part.)

4.1.1 The distance operator

The distance between individuals is not measured in the genotype space, but in the fitness landscape. The maximum of the differences of fitness at the level α is used, i.e.

$$d_\alpha(E_1, E_2) = \sup\{||x - y|| : x \in [E_1]_\alpha, \\ y \in [E_2]_\alpha\} \quad (9)$$

4.1.2 The selection operator

The size of the intermediate population can be twice as high as the the current population size, in the worst case. To control the maximum population size, at each iteration the set of elite points is revised and all the dominated solutions and duplicated points, along with their associated current search points, are removed. If the number of points is still too high, a random purge is performed until the size of the intermediate population is small enough.

4.2 Examples for crisp and fuzzy data

As an example, in Figure 2 it is shown the evolution of the MOSA algorithm for a two-criteria problem [9] and a fuzzy problem (fit the coefficients of a linear regression problem under data with an imprecision of the 10%). In the first problem, it can be observed that all the solutions are in the Pareto front after 100 iterations, and how the population size evolves. In the second problem, the set of non α-dominated solutions form a cloud that surrounds the exact solution –the point (2,1)– after 1000 iterations. Observe that the shape of the cloud can be used to study how the imprecision in the measures affects the tolerance of the solution.

5 Numerical analysis

This section has two parts: First: some benchmark problems are used to contrast the properties of the simulated annealing with those of the GA used in previous papers, and second: the effect of introducing some imprecision in the dataset is studied.

5.1 Benchmark problems

To separate the effects of the fuzzy data and the search method, we have fed our algorithm

	WM1	WM2	WM3	CH1	CH2	CH3	NIT	LIN	CUA	NEU	WLS	BFT	BMO
f1	5.65	5.73	5.57	5.82	8.90	6.93	5.63	130.5	**0.00**	0.17	0.09	0.45	**0.30**
f1-10	6.89	7.19	6.54	6.84	10.15	8.20	7.16	133.91	**1.40**	1.78	1.62	1.86	**1.71**
f1-20	11.07	10.99	11.06	11.33	13.45	12.42	10.63	135.6	**5.29**	6.42	5.90	6.04	**5.98**
f1-50	51.78	46.40	47.80	53.48	48.94	48.16	39.65	166.64	**33.53**	41.18	36.76	39.62	**38.66**
f2	0.41	0.48	0.45	0.40	0.59	0.45	0.43	1.54	1.61	1.48	**0.15**	**0.24**	0.26
f2-10	0.64	0.68	0.68	0.59	0.68	0.60	0.58	1.71	1.75	1.81	**0.29**	0.42	**0.41**
f2-20	1.27	1.16	1.17	1.29	1.15	1.17	0.97	2.04	2.09	0.90	**0.76**	**0.87**	**0.87**
f2-50	4.34	3.98	3.94	4.47	3.90	3.97	**3.59**	4.67	4.78	3.76	3.62	3.67	3.72
cable $\cdot 10^{-3}$	778	720	723	673	663	655	548	418	**393**	522	486	441	**437**
building $\cdot 10^{2}$	1.113	1.051	1.023	0.983	1.753	1.465	0.432	0.477	-	0.276	**0.246**	**0.375**	0.389

Table 1: Comparative results between additive regression + genetic backfitting (BFT), MOSA-based backfitting (BMO) and other approaches. BFT method was limited to 25 fuzzy rules. The best of WM, CH, NIT, BFT and BMO, plus the best overall model, were highlighted for every dataset.

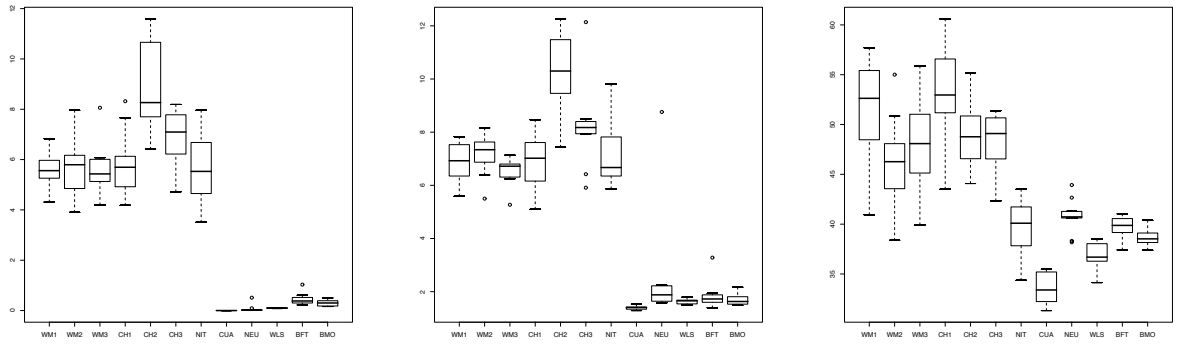

Figure 3: Comparative results between additive regression + backfitting (BFT) and multiobjective fuzzy backfitting (BFO). over f1, f1-20 and f1-50 datasets.

with crisp data (i.e., all fuzzy sets have a support of size 0,) and let the new algorithm to make at most the same number of evaluations than the genetic backfitting.

All the methods, the datasets and the statistical experimental setup used, are referenced in [25]. Wang and Mendel with importance degrees 'maximum' (WM1), 'mean' (WM2) and 'product maximum-mean' (WM3), the same three versions of Cordón and Herrera's method (CH1, CH2, CH3), Nozaki, Ishibuchi and Tanaka's (NIT), Linear (LIN) and Quadratic regression (QUA) , Neural Networks (NN) and TSK rules induced with Weighted Least Squares (WLS) are compared to genetic backfitting (BFT) and MOSA backfitting (BMO) over 8 synthetic problems and two real world problems. "f1" is $z = x^2 + y^2$ and "f2" is $10(x - xy)/(x - 2xy + y)$. "fx-y" is the function fx with y% of gaussian noise. "Building" and "Cable" are real-world problems, of moderate and small sizes, respectively. 5x2cv experimental framework was used: 50% of points were used to train the model, that was tested against the remaining 50%; roles of training and test sets were interchanged and the process repeated, and this was replicated 5 times for different permutations of the dataset, which gives 10 repetitions of the learning algorithm for every dataset. The mean of the test errors is shown in Table 1, and part of the boxplot of the results are shown in Figure 3. p-values assessing significance of the statistical contrasts as indicated in 5x2cv method are not included, but they can be deduced from the graphs: non overlapping boxes indicate that there exist a statistically significant difference between the algorithms involved. Observe that scalar SA-based backfitting is approximately equal to GA-based backfitting, and actually it seems

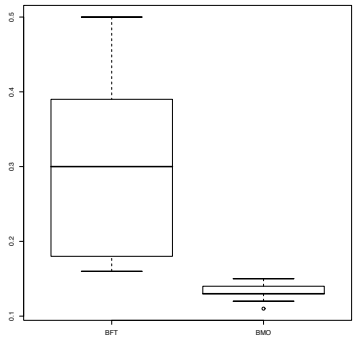

Figure 4: BFT compared to BFO when input data is *fuzzified* as mentioned in section 5.2.

to improve those results, while having less than 1/100th of the memory consumption of the GA.

5.2 Fuzzy data

In this second test, we have *fuzzified* the input data by covering each example with a triangular, symmetrical fuzzy set with a support of size 0.01, and centered in the precise data. As explained in [26, 28], we expect that the use of a fuzzy fitness have measurable consequences over the generalization properties of the models. In Figure 4 the dispersion of the results of BFT (over crisp data) and BMO (over fuzzified data) for the dataset f1 are shown. It is remarked that BMO produces a set of solutions; we have selected in all cases the rule with lower maximum error in the Pareto set. It is clear that there exist a significant different favouring the fuzzy algorithm. Nevertheless, since the time used by BFO was much higher in this case than the used by BFT, most restrictive tests must be made, but these preliminary results seem promising to us.

6 Concluding remarks

The learning of fuzzy rules from imprecise data is a heavy computational problem. In particular, for high-dimensional datasets, we need efficient algorithms that can be used to find the minimum of a fuzzy-valued function, which in turn is another hard problem.

In this work we have proposed to extend the backfitting of Additive Fuzzy Models to imprecise (fuzzy and interval valued) data, and also propose to use a population-based Simulated Annealing able to find a set of non α-dominated solutions of the problem. Compared to genetic backfitting, the memory consumption is much lower, and the execution time comparable, therefore this algorithm can be considered as a first step in the search for the methods proposed in [26, 28].

Acknowledgements

This work was supported by the Spanish Ministry of Education and Science, under grant TIN2005-08036-C05-05.

References

[1] Ali, F. M. (2001) A differential equation approach to fuzzy vector optimization problems and sensitivity analysis. Fuzzy Sets and Systems, Volume 119, Issue 1, 1 April 2001, Pages 87-95

[2] Burke, E.K. et al. (2001) Combining Hybrid Metaheuristics and Populations for the Multiobjective Optimisation of Space Allocation Problems. *Proc. GECCO'2001*, pages 1252–1259

[3] Chen, Y. and Wang, J. (2003) Support Vector Learning for Fuzzy Rule-Based Classification Systems, IEEE Transactions on Fuzzy Systems, vol. 11, no. 6, pp. 716-728, 2003.

[4] Chen, Y. and Wang, J. (2003) Kernel Machines and Additive Fuzzy Systems: Classification and Function Approximation. Proc. IEEE Int'l Conf. on Fuzzy Systems,pp. 789-795, 2003.

[5] Cordón, O., Herrera, F. (2000) A proposal for improving the accuracy of linguistic modeling. IEEE Transactions on Fuzzy Systems, 8, 3, pp. 335-344.

[6] Czyzak, P. and Jaszkiewicz, A. (1998) Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis*, 7:34–47, 1998.

[7] del Jesús, M. J., Hoffmann, F. , Junco, L., Sánchez, L. (2004) Induction of fuzzy-rule-based classifiers with evolutionary boosting

algorithms. IEEE T. Fuzzy Systems 12(3): 296-308

[8] Ekel, P., Pedrycz, W., Schinzinger, R. (1998) A general approach to solving a wide class of fuzzy optimization problems. Fuzzy Sets and Systems, Volume 97, Issue 1, 1 July 1998, Pages 49-66

[9] Fonseca, C., Fleming, P.J. (1995) An Overview of Evolutionary Algorithms in Multiobjective Optimization, *Evolutionary Computation* 3, 1-16, 1995.

[10] Friedman, J., Hastie, T., Tibshirani, R. (1998) "Additive Logistic Regression: A Statistical View of Boosting". Machine Learning.

[11] Gan, H. M., Tan, A. H. (2005) From a Gaussian mixture model to additive fuzzy systems. IEEE Trans FS Volume 13, Issue 3, June 2005 Page(s) : 303 - 316

[12] Hapke, M., Jaszkiewicz, A., Slowinski, R.. (2000) Pareto Simulated Annealing for Fuzzy Multi-Objective Combinatorial Optimization. *Journal of Heuristics*, 6(3):329–345, August 2000.

[13] Mulet, R. Hernández-Guía, M. Rodríguez-Pérez, S. (2005) A new simulated annealing algorithm for the multiple sequence alignment problem: The approach of polymers in a random media. *Physical Review E*, 72(3), 2005.

[14] Huang, C. Moraga, C.(2005). Extracting fuzzy if-then rules by using the information matrix technique. J. Comput. Syst. Sci. 70(1): 26-52 2005

[15] Jiménez, F. Cadenas, J. M. ,Verdegay J. L. and Sánchez, G. (2003) Solving fuzzy optimization problems by evolutionary algorithms Information Sciences, Volume 152, June 2003, Pages 303-311

[16] van den Berg, D., Ettes, (1998) Representation and Learning Capabilities of Additive Fuzzy Systems, Proceedings of INES'98, pp. 121-126, Vienna, Austria, September 1998.

[17] Li, C., Liao, X. and Yu, J. (2004) Tabu search for fuzzy optimization and applications. Information Sciences, Volume 158, January 2004, Pages 3-13

[18] Lin, C.C. (2004) A weighted max-min model for fuzzy goal programming. Fuzzy Sets and Systems, 142,3,2004, pp. 407-420.

[19] Loetamonphong, J., Fang, S. and Young, R. (2002) Multi-objective optimization problems with fuzzy relation equation constraints Fuzzy Sets and Systems, Volume 127, Issue 2, 16 April 2002, Pages 141-164

[20] Matos, M.A. Melo, P. (1999) Multiobjective Reconfiguration for Loss Reduction and Service Restorating Using Simulated Annealing. In *Proc. PowerTech Budapest 99*, pages 213–218, Budapest, Hungary, 1999

[21] Nam, D. Park, C. H. (2000) Multiobjective Simulated Annealing: A Comparative Study to Evolutionary Algorithms. *International Journal of Fuzzy Systems*, 2(2):87–97, 2000.

[22] Nam, D., Park, C. H. (2002) Pareto-Based Cost Simulated Annealing for Multiobjective Optimization. *Proceedings (SEAL'02)*, volume 2, pages 522–526, Orchid Country Club, Singapore, November 2002.

[23] Otero, J., Sánchez, L. (2006) Induction of descriptive fuzzy classifiers with the Logitboost algorithm. Soft Computing. In press

[24] Sakawa, M. Fuzzy Sets and Interactive Multiobjective Optimization. Plenum. 1993

[25] Sánchez, L. Otero, J. (2004) A fast genetic method for inducting descriptive fuzzy models. Fuzzy Sets and Systems 141(1): 33-46

[26] Sánchez, L., Couso, I. (2005) Advocating the use of Imprecisely Observed Data in Genetic Fuzzy Systems. I International Workshop on Genetic Fuzzy Systems, GFS 2005.

[27] Sánchez, L., Otero, J. (2006) Boosting fuzzy rules in classification problems under single-winner inference. International Journal of Intelligent Systems, 2006. In press

[28] Sánchez, L., Couso, I. (2006) Advocating the use of Imprecisely Observed Data in Genetic Fuzzy Systems". IEEE Trans Fuzzy Sys, 2006. Under review.

[29] Smith, K. I. Everson, R. M. and Fieldsend, J. E. (2004) Dominance Measures for Multi-Objective Simulated Annealing. In *Proc. CEC'2004*, volume 1, pages 23–30, Portland, Oregon, USA, June 2004.

[30] Zhang, D., Bai, X., Cai, K. (2004) Extended neuro-fuzzy models of multilayer perceptrons. Fuzzy Sets and Systems, 142, 2, 2004, 221-242.