Soft Comput (2011) 15:2303–2318
DOI 10.1007/s00500-010-0671-2

FOCUS

Multiobjective genetic fuzzy rule selection of single granularity-based fuzzy classification rules and its interaction with the lateral tuning of membership functions

Rafael Alcalá · Yusuke Nojima · Francisco Herrera · Hisao Ishibuchi

Published online: 25 November 2010

Abstract Multiobjective genetic fuzzy rule selection is based on the generation of a set of candidate fuzzy classification rules using a preestablished granularity or multiple fuzzy partitions with different granularities for each attribute. Then, a multiobjective evolutionary algorithm is applied to perform fuzzy rule selection. Since using multiple granularities for the same attribute has been sometimes pointed out as to involve a potential interpretability loss, a mechanism to specify appropriate single granularities at the rule extraction stage has been proposed to avoid it but maintaining or even improving the classification performance. In this work, we perform a statistical study on this proposal and we extend it by combining the single granularity-based approach with a lateral tuning of the membership functions, i.e., complete contexts learning. In this way, we analyze in depth the importance of determining the appropriate contexts for learning fuzzy classifiers. To this end, we will compare the single granularity-based approach with the use of multiple granularities with and without tuning. The results show that the performance of the obtained classifiers can be even improved by obtaining the appropriate variable contexts, i.e., appropriate granularities and membership function parameters.

Keywords Fuzzy rule-based classifiers · Multiobjective evolutionary algorithms · Granularity learning · Lateral tuning of membership functions

R. Alcalá (✉) · F. Herrera
Department of Computer Science and Artificial Intelligence,
University of Granada, 18071 Granada, Spain
e-mail: alcala@decsai.ugr.es

F. Herrera
e-mail: herrera@decsai.ugr.es

Y. Nojima · H. Ishibuchi
Department of Computer Science and Intelligent Systems,
Osaka Prefecture University, 1-1 Gakuen-cho, Naka-ku,
Sakai, Osaka 599-8531, Japan
e-mail: nojima@cs.osakafu-u.ac.jp

H. Ishibuchi
e-mail: hisaoi@cs.osakafu-u.ac.jp

1 Introduction

Many automatic techniques have been proposed in the literature to extract a proper set of fuzzy rules from numerical data. Most of these techniques usually try to improve the performance associated with the prediction error without paying a special attention to the system interpretability, an essential aspect of fuzzy rule-based systems. In the last years, the problem of finding the right interpretability–accuracy tradeoff, despite the original nature of fuzzy logic, has given rise to a growing interest in methods that take both aspects into account (Casillas et al. 2003). Of course, the ideal thing would be to satisfy both criteria to a high degree but since they are contradictory issues, generally it is not possible.

Evolutionary multiobjective optimization (EMO) algorithms (Coello et al. 2002; Deb 2001) generate a family of equally valid solutions, where each solution tends to satisfy a criterion to a higher extent than another. For this reason, EMO algorithms have been also applied to improve the accuracy–interpretability tradeoff of fuzzy rule-based systems (Alcalá et al. 2007b, 2009a; Botta et al. 2009; Cococcioni et al. 2007; Gacto et al. 2009, 2010; Ishibuchi et al. 1995, 1997; Ishibuchi and Nojima 2007; Ishibuchi and Yamamoto 2004; Pulkkinen and Koivisto 2008, 2010; Sánchez et al. 2009), where each solution in the Pareto front represents a different tradeoff between interpretability and accuracy (typically measured as complexity and prediction error). The application of EMO algorithms to fuzzy rule-based systems is often referred to as multiobjective genetic fuzzy systems.

Some of the most recognized works (Ishibuchi et al. 1995, 1997) were devoted to the application of EMO algorithms to perform a genetic fuzzy rule selection on an initial set of classification rules involving "*don't care*" conditions and considering two different objectives, classification accuracy and the number of rules. Then, a third objective was also included in order to minimize the length of the rules in Ishibuchi and Yamamoto (2004). In genetic fuzzy rule selection, a previously fixed granularity (Ishibuchi et al. 1995, 1997) or multiple granularities (Ishibuchi and Yamamoto 2004; Nojima et al. 2009) of triangular fuzzy membership functions have been used for the same attribute in the design of fuzzy classifiers, even regression models (Alcalá et al. 2003), since an appropriate granularity for each attribute is not known beforehand. By using multiple granularities, the number of fuzzy rules can be successfully reduced in a model.

Although using multiple granularities for the same attribute is one of the most promising approaches, its interpretability loss has often been pointed out. To solve this problem, we have proposed in Alcalá et al. (2009b) a single granularity specification approach for multiobjective genetic fuzzy rule selection. Multiobjective genetic fuzzy rule selection is a two-step method. In the first phase, a prespecified number of promising fuzzy rules are generated by a heuristic procedure. In the second phase, a multiobjective genetic algorithm is used to select a small number of fuzzy rules from the extracted ones in the first phase. A single granularity specification is an additional process before the second phase. After extracting a prespecified number of fuzzy rules with multiple granularities, a single granularity is specified for each attribute individually according to the frequency of employed partitions and the importance of the multiple granularity-based extracted rules. Then, a prespecified number of fuzzy rules are extracted again based on the specified granularity for each attribute. Following the same main idea, four different mechanisms were proposed and compared in Alcalá et al. (2009b).

In this work, we perform a statistical study focused on the best mechanism presented in Alcalá et al. (2009b), and we extend it by combining the single granularity-based approach with a lateral tuning of the membership functions, i.e., complete contexts learning. This is based on the linguistic 2-tuple representation model (Alcalá et al. 2007a; Herrera and Martínez 2000). The linguistic 2-tuple representation allows the lateral translation of a membership function by only considering one parameter (Alcalá et al. 2007a), and it represents an effective way to manage the size of the search space when both rule selection and tuning are combined. In this way, we can analyze in depth the importance of determining the appropriate contexts for learning fuzzy classifiers, which is the main aim of this paper.

To this end, we have compared the single granularity-based approach with the use of multiple granularities with and without tuning. We have tested the different approaches (with–without multiple granularities and with–without tuning) on 24 real-world problems. To assess the results obtained by the different algorithms, we have applied a non-parametric statistical test (Demšar 2006; García et al. 2008, 2009; García and Herrera 2008) for pair-wise comparisons, considering the means of the most accurate points of the Pareto fronts obtained from each algorithm. As well as the interpretability improvement that the use of a single granularity involves, the results show that the performance of the obtained classifiers can be even improved by obtaining the appropriate variable contexts, i.e., appropriate granularities and membership function parameters.

This contribution is arranged as follows. The next section introduces fuzzy rule-based classifiers by describing the rule structure and inference used in this paper. Section 3 presents the algorithm to generate single granularity-based fuzzy classification rules for multiobjective genetic fuzzy rule selection. Section 4 shows the experimental study on the best mechanism for this method. In Sect. 5, we propose the combination of the single granularity-based approach with the lateral tuning of membership functions in order to completely specify the variable contexts. Section 6 presents the experiments in combination with the lateral tuning. Section 7 points out some conclusions. Finally, Appendix describes the Wilcoxon signed-rank test for pair-wise comparisons.

2 Preliminaries: fuzzy rule-based classifiers structure and inference

Let us assume that we have m training (i.e., labeled) patterns $\mathbf{x}_p = (x_{p1}, \ldots, x_{pn}), p = 1, 2, \ldots, m$ from M classes in an n-dimensional pattern space where x_{pi} is the attribute value of the pth pattern for the ith attribute ($i = 1, \ldots, n$). For the simplicity of explanation, we assume that all the attribute values have already been normalized into real numbers in the unit interval [0, 1]. Thus, the pattern space of our classification problem is an n-dimensional unit-hypercube $[0, 1]^n$.

For our n-dimensional pattern classification problem, we use fuzzy rules of the following type:

$$\begin{aligned} R_q &: \text{If } x_1 \text{ is } A_{q1} \text{ and } \ldots \text{ and } x_n \text{ is } A_{qn} \\ &\quad \text{then Class } C_q \text{ with } CF_q, \end{aligned} \tag{1}$$

where R_q is the label of the qth fuzzy rule, $\mathbf{x} = (x_1, \ldots, x_n)$ is an n-dimensional pattern vector, A_{qi} is an antecedent fuzzy set ($i = 1, \ldots, n$), C_q is a class label, and CF_q is a rule weight. We denote the antecedent fuzzy sets of R_q as a fuzzy vector $\mathbf{A}_q = (A_{q1}, A_{q2}, \ldots, A_{qn})$.

Fourteen fuzzy sets are initially considered in four fuzzy partitions with different granularities. Figure 1 depicts these partitions. In addition to those 14 fuzzy sets, we also use the domain interval [0, 1] itself as an antecedent fuzzy set in order to represent a *don't care* condition.

Let S be a set of fuzzy rules of the form in Eq. 1. When an input pattern $\mathbf{x}_p$ is to be classified by S, first we calculate the compatibility grade of $\mathbf{x}_p$ with the antecedent part $\mathbf{A}_q = (A_{q1}, A_{q2}, \ldots, A_{qn})$ of each fuzzy rule R_q in S using the product operation as,

$$\mu_{\mathbf{A}_q}(\mathbf{x}_p) = \mu_{A_{q1}}(x_{p1}) \cdot \ldots \cdot \mu_{A_{qn}}(x_{pn}), \tag{2}$$

where $\mu_{A_{qi}}(\cdot)$ is the membership function of the antecedent fuzzy set A_{qi}. Then, a single winner rule R_w is identified using the compatibility grade and the rule weight of each fuzzy rule as

$$\mu_{\mathbf{A}_w}(\mathbf{x}_p) \cdot CF_w = \max\{\mu_{\mathbf{A}_q}(\mathbf{x}_p) \cdot CF_q | R_q \in S\}. \tag{3}$$

The input pattern $\mathbf{x}_p$ is classified as the consequent class C_w of the winner rule R_w. When multiple fuzzy rules with different consequent classes have the same maximum value in Eq. 3, the classification of $\mathbf{x}_p$ is rejected. If there is no compatible fuzzy rule with $\mathbf{x}_p$, its classification is also rejected.

Finally, we should emphasize that the use of a rule weight or certainty grade can be interpreted as modifying the membership function of each antecedent fuzzy set as shown in Ishibuchi et al. (2000). Whereas it introduces a dimension of complexity to fuzzy if–then rules, it is widely used for fuzzy classification as it just affects the strength of each fuzzy if–then rule in the classification phase which does not change the position of the antecedent fuzzy sets (preserving the meaning of each linguistic value while the modified antecedent fuzzy set is not normal anymore).

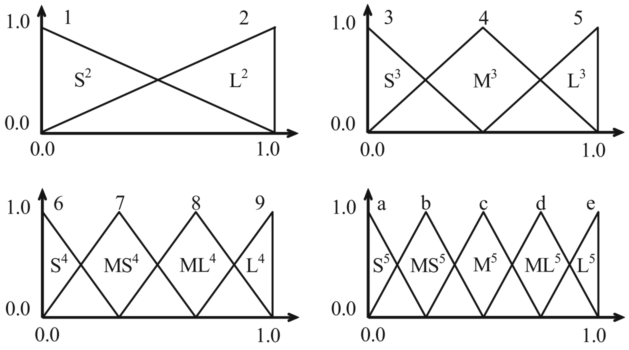


Fig. 1 The fourteen antecedent fuzzy sets considered

3 An algorithm for generating single granularity-based fuzzy classification rules

As we have already explained, multiobjective genetic fuzzy rule selection has been based on a previously fixed granularity (Ishibuchi et al. 1995, 1997) (five linguistic terms in all the attributes) or multiple granularities (Ishibuchi and Yamamoto 2004; Nojima et al. 2009). Based on this last approach (Ishibuchi and Yamamoto 2004), we describe in this section the mechanism that we proposed in Alcalá et al. (2009b) to generate single granularity-based fuzzy classification rules, which represents an approach closer to the interpretability. The proposed procedure is as follows:

Step 1 Rule extraction with multiple granularities.
Step 2 Specification of single granularity for each attribute based on the extracted rules.
Step 3 Rule extraction with selected single granularities.
Step 4 Multiobjective genetic fuzzy rule selection.

The original multiple granularities-based procedure (Ishibuchi and Yamamoto 2004) is composed of Steps 1 and 4. Steps 2 and 3 are additional procedures. In Step 1, we extract a fixed small number of rules for each class based on well-known data mining rule evaluation measures (Agrawal et al. 1996) and multiple granularities. In Step 2, we select a single granularity for each attribute based on the extracted rules. Then, we extract the final set of candidate rules for each class by using the selected single granularities in Step 3. Step 4 is the same as the original one to perform multiobjective genetic fuzzy rule selection. The next subsections present detailed explanations of these steps.

3.1 Rule extraction with multiple granularities (Step 1)

Since 14 antecedent fuzzy sets in Fig. 1 and an additional *don't care* fuzzy set [0, 1] are used for each attribute of the n-dimensional classification problem, the total number of possible fuzzy rules is 15^n. Among these possible rules, we examine only short fuzzy rules with a small number of antecedent conditions (i.e., short fuzzy rules with many *don't care* conditions) to generate an initial set of candidate rules. In this work, we specify the maximum number of antecedent conditions as three for datasets with less than 30 attributes and two for datasets with more than or equal to 30 attributes.

The consequent class C_q and the rule weight CF_q of each fuzzy rule R_q are specified from training patterns compatible with its antecedent part $\mathbf{A}_q = (A_{q1}, A_{q2}, \ldots, A_{qn})$ in the following heuristic manner (Ishibuchi et al. 2004). First, the confidence of each class for the antecedent part $\mathbf{A}_q$ is calculated as:

$$c(\mathbf{A}_q \Rightarrow \text{Class}\,h) = \frac{\sum_{\mathbf{x}_p \in \text{Class h}} \mu_{\mathbf{A}_q}(\mathbf{x}_p)}{\sum_{p=1}^{m} \mu_{\mathbf{A}_q}(\mathbf{x}_p)}, \quad h = 1, \ldots, M. \tag{4}$$

It should be noted that "$\mathbf{A}_q \Rightarrow \text{Class}\,h$" means the fuzzy rule with the antecedent part $\mathbf{A}_q$ and the consequent class h. Then, the consequent class C_q is specified by identifying the class with the maximum confidence:

$$c(\mathbf{A}_q \Rightarrow \text{Class } C_q) = \max_{h=1,2,\ldots,M} \{c(\mathbf{A}_q \Rightarrow \text{Class } h)\}. \tag{5}$$

In this manner, we generate the fuzzy rule R_q (i.e., $\mathbf{A}_q \Rightarrow \text{Class}\,C_q$) with the antecedent part $\mathbf{A}_q$ and the consequent class C_q. We do not generate any fuzzy rules with the antecedent part $\mathbf{A}_q$, if there is no compatible training pattern with $\mathbf{A}_q$.

The rule weight CF_q of each fuzzy rule R_q has a large effect on the performance of fuzzy rule-based classifiers. We use the following specification of CF_q, because good results were reported in the literature Ishibuchi and Yamamoto (2005):

$$CF_q = c(\mathbf{A}_q \Rightarrow \text{Class } C_q) - \sum_{\substack{h=1 \\ h \neq C_q}}^{M} c(\mathbf{A}_q \Rightarrow \text{Class } h). \tag{6}$$

We do not use the fuzzy rule R_q as a candidate rule, if the rule weight CF_q is not positive (i.e., if its confidence is not larger than 0.5).

In the above-mentioned heuristic manner, we can generate a large number of short fuzzy rules as candidate rules in multiobjective fuzzy rule selection (some of them with not interesting properties). In order to directly focus on the most interesting rules, a prescreening procedure is applied to decrease the number of candidate rules. Among short fuzzy rules satisfying these two threshold values, we choose a prespecified number of candidate rules for each class. As a rule evaluation criterion, we use the product of the support $s(R_q)$, which indicates the percentage of patterns covered by R_q, and the confidence $c(R_q)$. That is, we choose a prespecified number of the best candidate rules for each class with respect to the product $p(R_q) = s(R_q) \cdot c(R_q)$.

3.2 Single granularity specification and rule extraction (Steps 2 and 3)

Once a set of candidate rules is obtained based on multiple granularities (Step 1), the original approach (Ishibuchi and Yamamoto 2004) goes to Step 4 in order to apply multiobjective fuzzy rule selection. However, there is useful information in the extracted rules that could be used to specify an appropriate single granularity for each attribute. The frequency of the employed granularities in the extracted rules (weighted by the corresponding rule importance) has been used in Alcalá et al. (2009b) to fix the most promising granularities. For each attribute i $(i = 1, \ldots, n)$, we can specify the granularity with the highest sum of importance of the rules considering such granularity in the corresponding attribute:

$$Gr(i) = \operatorname*{argmax}_{g=2,\ldots,5} \left\{ \sum_{\text{Gran}(\mathbf{A}_{\text{qi}})=\text{g}} \text{Imp}(\mathbf{R}_q) \right\}, \tag{7}$$

where $\text{Gran}(\mathbf{A}_{qi})$ is the granularity of the partition containing the fuzzy set used in attribute i of rule R_q and $\text{Imp}(R_q)$ is a criterion associated to the importance of the rule in the sum. Many criteria can be considered involving different specification mechanisms:

- Frequency: $\text{Imp}(R_q) = 1, \forall q$.
- Confidence: $\text{Imp}(R_q) = c(R_q), \forall q$.
- Weight: $\text{Imp}(R_q) = CF_q, \forall q$.
- Support: $\text{Imp}(R_q) = s(R_q), \forall q$.
- Product: $\text{Imp}(R_q) = p(R_q), \forall q$.

However, the first three criteria are not recommended since they usually provoke overfitting. The last two criteria were studied in Alcalá et al. (2009b) as a way to extract more general rules instead of very specific ones, which helps to the generalization ability. In the same way, in order to preferably take into account more general rules, two approaches named 1-ALL approach and 1-2-3 approach, were examined in Alcalá et al. (2009b) with the two basic criteria (i.e., product and support). Both approaches give priority to granularities in the rules with a single condition, i.e., Eq. 7 is applied by only considering size one rules, if possible. The difference is only when there is no rule with a single condition in the corresponding attribute. Let us consider the product criterion and the next six rules, where g^i represents any fuzzy set of a partition with granularity i,

R_1 : If x_1 is g^2 and x_2 is g^4 and x_3 is g^3 then Class 1, $p(R_1)$: 0.4.
R_2 : If x_1 is g^4 then Class 2, $p(R_2)$: 0.8.
R_3 : If x_2 is g^3 then Class 2, $p(R_3)$: 0.3.
R_4 : If x_2 is g^2 then Class 1, $p(R_4)$: 0.8.
R_5 : If x_2 is g^3 and x_3 is g^4 then Class 1, $p(R_5)$: 0.6.
R_6 : If x_1 is g^2 and x_2 is g^2 and x_3 is g^3 then Class 1, $p(R_6)$: 0.3.

When we specify a granularity for the first attribute, we first check rule(s) with a single condition related to the first attribute by both approaches (1-ALL and 1-2-3). Since rule R_2 is the only rule in this situation, we select granularity 4 for the first attribute. Next, in the same manner, we can find two rules: R_3 and R_4 for the second attribute. We select granularity 2 for the second attribute because of the high product value obtained by both approaches. Finally, we select a granularity for the third attribute, but there is no rule with a single condition. In 1-ALL approach, we specify a single granularity from all the rules including the third attribute independently of the number of conditions they have (rules R_1, R_5 and R_6). The sum of product values for granularity 3 is 0.7 and 0.6 for granularity 4. From this comparison, we select granularity 3 for the third attribute. On the other hand, in 1–2–3 approach, we give priority to the rules with a smaller number of conditions (two conditions). That is, we select granularity 4 for the third attribute (if there are no rules with two conditions, then those with three are considered).

In Step 2, we can select a single granularity for each attribute in this way. Once single granularities are fixed, in Step 3, we have to apply again the candidate rule extraction procedure explained in Step 1 by only using the specified single granularities for each attribute.

Four different mechanisms have been defined: Support/1-ALL, Product/1-ALL, Support/1–2–3 and Product/1–2–3. However, we will focus on Product/1-ALL from now on, since this approach reported the best results in Alcalá et al. (2009b).

3.3 Multiobjective fuzzy rule selection (Step 4)

Let us assume that we have N candidate rules (i.e., N/M candidate rules for each of M classes). Any subset S of the N candidate rules can be represented by a binary string of length N:

$$S = s_1 s_2 \ldots s_N,$$

where $s_j = 1$ and $s_j = 0$ mean the inclusion and the exclusion of the jth candidate rule R_j in the subset S, respectively $(j = 1, \ldots, N)$. Such a binary string S is used as an individual in an EMO algorithm for multiobjective fuzzy rule selection.

It should be noted that S can be viewed as a fuzzy rule-based classifier. Each fuzzy rule-based classifier S is evaluated by the next three objectives:

$f_1(S)$: the number of correctly classified training patterns.
$f_2(S)$: the number of selected fuzzy rules.
$f_3(S)$: the total number of antecedent conditions.

That is, our multiobjective fuzzy rule selection problem is written as:

$$\text{Maximize}\, f_1(S),\ \text{and minimize}\, f_2(S)\ \text{and}\, f_3(S). \quad (8)$$

We use NSGA-II of Deb et al. (2002) to search for non-dominated fuzzy rule-based classifiers with respect to these three objectives. Uniform crossover and bit-flip mutation are used as genetic operations. The execution of NSGA-II was terminated at the prespecified number of generations.

In order to efficiently decrease the number of fuzzy rules in each rule set S, two heuristic techniques are used. One is biased mutation, where a larger mutation probability is assigned to the mutation from 1 to 0 than that from 0 to 1. The other is the removal of unnecessary fuzzy rules. Since we use the single winner-based scheme in Eq. 3 for classifying each training pattern by a fuzzy rule-based classifier S, some fuzzy rules in S may classify no training patterns. We can remove those unnecessary fuzzy rules from S without changing any classification results by S [i.e., without changing the first objective $f_1(S)$]. This heuristic procedure can be viewed as a kind of local search since $f_2(S)$ and $f_3(S)$ are improved without deteriorating $f_1(S)$.

4 Experiments on the learning of single granularities

In order to statistically examine the effects of the best granularity specification mechanism, Product/1-ALL, we have extended the experimental framework in Alcalá et al. (2007a) by considering up to 24 datasets selected from the UCI repository (Asuncion and Newman 2007). Since the algorithm has been not designed to consider nominal data (which is not the aim of the paper), we have only considered those available datasets with only continuous attributes in order to avoid wrong conclusions. On the other hand, in the case of presenting missing values (Bands, Cleveland, Dermatology, Hepatitis, Mammographic and Wisconsin), we have removed the instances with any missing value before partitioning. Table 1 summarizes the main properties of these datasets. It shows, for each dataset, the number of patterns, the number of attributes and the number of classes.

In order to analyze the performance of the single granularity specification, we compare the original approach (All Granularities) with the mechanism proposed to specify single granularities (Product/1-ALL). The parameter settings for both approaches are as follows (same conditions in all the cases):

- The number of fuzzy rules for each class: 300
- Optimizer: NSGA-II
- Population size: 200
- The number of generations: 5,000
- Crossover probability: 0.9 (uniform crossover)
- Mutation probability: 0.05 (from 1 to 0), 1/Lth (from 0 to 1, where Lth is the string length).

We consider a *tenfold cross-validation model*, i.e., ten random partitions of data each with 10%, and a combination of nine of them (90%) as training and the remaining

Table 1 Datasets considered for comparisons

Name	Patterns	Attributes	Classes
Appendicitis	106	7	2
Australian	690	14	2
Bands	365	19	2
Bupa	345	6	2
Cleveland	297	13	5
Dermatology	358	34	6
Glass	214	9	6
Haberman	306	3	2
Hayes-roth	132	4	3
Heart	270	13	2
Hepatitis	80	19	2
Ionosphere	351	34	2
Iris	150	4	3
Mammographic	830	5	2
Newthyroid	215	5	3
Pasture	36	22	3
Pima	768	8	2
Saheart	462	9	2
Sonar	208	60	2
Tae	151	5	3
Vehicle	846	18	4
Wdbc	569	30	2
Wine	178	13	3
Wisconsin	683	9	2

Table 2 Results obtained by the studied methods (most accurate)

Datasets	All Granularities				Product/1-ALL			
	#R	#C	Tr.	Ts.	#R	#C	Tr.	Ts.
Appendicitis	2.37	3.73	91.86	87.91	3.40	6.97	93.29	**88.21**
Australian	2.00	2.00	85.51	**85.51**	6.63	16.17	88.82	85.31
Bands	4.60	12.43	71.36	**68.73**	6.83	17.53	69.90	65.37
Bupa	6.73	16.73	69.50	63.99	12.80	31.27	78.06	**67.11**
Cleveland	20.17	56.13	73.11	**55.11**	28.67	77.77	76.72	52.83
Dermatology	11.40	19.33	99.07	**94.12**	13.53	23.47	99.51	93.26
Glass	12.63	32.37	78.65	60.48	19.30	44.73	83.97	**69.96**
Haberman	6.50	13.93	79.46	71.89	3.00	6.00	74.70	**73.19**
Hayes-roth	9.17	15.27	90.88	78.03	10.83	16.90	90.91	**79.14**
Heart	7.67	14.27	90.19	83.46	10.13	23.97	92.81	**84.32**
Hepatitis	3.80	9.30	96.10	**90.44**	4.33	9.40	98.20	87.00
Ionosphere	9.53	14.83	95.64	88.62	10.50	15.70	95.99	**90.79**
Iris	4.03	6.80	99.11	95.11	5.23	7.53	98.30	**95.33**
Mammogr.	6.97	14.70	83.07	**81.04**	11.67	25.27	83.30	79.35
Newthyroid	5.37	9.20	96.19	91.78	7.37	15.73	97.59	**93.01**
Pasture	3.70	5.87	98.05	**75.83**	4.43	8.13	100.00	73.61
Pima	6.63	14.33	77.80	**74.92**	10.63	25.87	79.04	73.79
Saheart	5.97	12.77	76.70	71.14	12.33	31.80	79.00	**71.22**
Sonar	6.73	10.50	86.54	**78.88**	6.87	12.60	87.82	77.48
Tae	7.77	18.93	66.55	54.57	11.33	25.43	65.98	**59.24**
Vehicle	13.77	35.77	69.34	62.81	15.60	43.10	70.80	**66.20**
Wdbc	5.67	9.83	97.12	**94.90**	7.27	12.40	97.26	93.96
Wine	3.90	8.23	100.00	**96.08**	6.37	12.10	99.92	95.11
Wisconsin	6.93	11.17	98.22	**96.07**	7.77	13.83	98.40	96.06
Mean	7.25	15.35	86.25	79.23	9.87	21.82	87.51	**79.62**

The bests in test are bold-faced

one as test.[1] For each one of the ten data partitions, the studied methods have been run three times, showing for each problem the averaged results of a total of 30 runs (10fcv x 3 different random seeds). Since these methods present a multiobjective nature, the averaged values are calculated considering the most accurate solution from each Pareto front obtained (the one with the highest classification rate in training). Our main aim following this approach is to obtain more reliable information at least in this part of the Pareto front, which in any case is comprised by quite simple models.

In order to assess whether significant differences exist among the results, we adopt statistical analysis (Demšar 2006; García et al. 2008, 2009, García and Herrera 2008) and in particular non-parametric tests, according to the recommendations made in Demšar (206) and García and Herrera 2008), where a set of simple, safe and robust non-parametric tests for statistical comparisons of classifiers has been analyzed. For a deep explanation on the use of non-parametric tests for data mining and computational intelligence, see the website at http://sci2s.ugr.es/sicidm/.

In our study, we will employ Wilcoxon's signed-rank test (Sheskin 2003, Wilcoxon 1945) for pair-wise comparison. Wilcoxon's test is based on computing the differences between two sample means (typically, mean test errors obtained by a pair of different algorithms on different datasets). A detailed description of this test is presented in Appendix. To perform the tests, we use a level of confidence $\alpha = 0.1$.

Table 2 shows the averaged number of rules/conditions (#R/#C) and classification percentages in training (Tr.) and test (Ts.) of the most accurate classifier from each of the obtained Pareto fronts. The overall mean values for each method are in the last row. The results show that Product/1-ALL presents the best overall mean value in the test classification percentage with approximately two more rules with respect to the original model considering multiple granularities. However, in Table 4, the application of the Wilcoxon test on the test classification percentage of the most accurate solutions shows that there is no statistical difference between both the approaches, All Granularities

[1] The corresponding data partitions (10-fcv) for these datasets are available at the KEEL project webpage (Alcalá-Fdez et al. 2009): http://sci2s.ugr.es/keel/datasets.php

Table 3 Results obtained by the studied method (equivalent complexity)

Datasets	Product/1-ALL (same complexity)			
	#R	#C	Tr.	Ts.
Appendicitis	2.17	4.10	92.24	**88.82**
Australian	3.50	6.40	87.13	**85.80**
Bands	4.67	11.37	68.93	**65.59**
Bupa	6.47	14.67	74.35	**67.93**
Cleveland	20.20	53.40	74.14	**53.48**
Dermatology	11.70	20.10	98.90	93.08
Glass	12.57	28.10	80.15	69.90
Haberman	3.00	6.00	74.70	73.19
Hayes-roth	9.57	13.40	90.07	**80.09**
Heart	7.20	15.73	91.29	83.46
Hepatitis	2.77	5.63	95.71	**90.20**
Ionosphere	9.27	13.37	95.67	90.33
Iris	4.23	5.30	97.56	94.44
Mammogr.	6.40	12.33	82.43	**79.67**
Newthyroid	5.63	11.20	96.54	92.23
Pasture	3.93	6.40	96.90	**73.89**
Pima	6.87	16.00	78.36	73.78
Saheart	6.20	14.90	76.71	70.49
Sonar	6.87	12.60	87.82	77.48
Tae	7.67	15.10	64.73	57.04
Vehicle	13.77	37.70	70.80	66.08
Wdbc	5.43	9.27	96.80	93.79
Wine	3.93	5.30	96.17	90.99
Wisconsin	6.97	11.50	98.23	96.02
Mean	7.12	14.58	86.10	79.49

Improvements with respect to the corresponding most accurate solution are bold-faced

vs. Product/1-ALL. The null hypothesis associated with Wilcoxon's test is accepted ($p \geq \alpha$), because the differences between R^+ and R^- are not significant. In any event, the proposed approach presents at least the same classification rates than the original one at the most accurate solution in general, avoiding the use of multiple granularities without any cost in the classification percentage.

A way to analyze the results in other parts of the Pareto fronts is to compute the average Pareto fronts, composed by the average values of the different solutions of each of the thirty Pareto fronts (from the most accurate solution to the simplest one). This method represents an extension of the idea of analyzing the most accurate solutions in the Pareto fronts (average of the most accurate, on the second most accurate, etc.) presented in Alcalá et al. (2007b), Gacto et al. (2009) where the search is focused on the most accurate solutions only and extended in Alcalá et al. (2009a) for some representative points. We consider the next most accurate solution from each of the 30 fronts to compute a new averaged solution, until no more solutions remain in any of these fronts, obtaining a representation of the average Pareto fronts.

Table 4 Wilcoxon's test: All Granularities (R^+) versus Product/1-ALL (R^-) on the test error at the most accurate solution and at the same complexity

Comparison	R^+	R^-	Hypothesis ($\alpha = 0.1$)	p value
All Granularities versus Product/1-ALL	144	156	Accepted	0.864
All Granularities versus Product/1-ALL (same complexity)	150	150	Accepted	1.0

In Table 3, we show the average results of the proposed approach by considering the solution from the average Pareto fronts which is closest in terms of the number of rules (equivalent complexity) to the most accurate average solution from All Granularities approach. Even though it seems preferable to ensure a single granularity than to obtain a simpler solution, with approximately two rules of difference (between All Granularities and Product/1-ALL), we can observe in this table that very similar results are obtained when equivalent complexities are considered. In fact, it still presents the best overall mean values in test and Table 4 also shows no statistical differences between both approaches at the same complexity, All Granularities vs. Product/1-ALL (same complexity). This demonstrates that by fixing an appropriate single granularity at least equivalent results can be obtained from the point of view of the accuracy and the complexity. Further, it is highly preferable to avoid multiple granularities in terms of the global interpretability of the obtained models.

Figure 2 plots the average Pareto fronts on training and test sets for Bupa dataset. As we can see in this example, by determining single granularities the test data accuracy can be improved even in most of the solutions obtained. Additionally, as an example, three rule sets obtained by All Granularities, Product/1-ALL (most accurate solution in the Pareto front) and Product/1-ALL (with a similar complexity in the same Pareto front) are depicted in Fig. 3. These figures clearly show the same trend with the associated average Pareto fronts. The main difference between those obtained by Product/1-ALL and the one obtained from All Granularities is that we could easily represent the last two rule bases, b and c, in terms of the associated linguistic labels (if an expert is able to provide them), which is not possible with the first one, a.

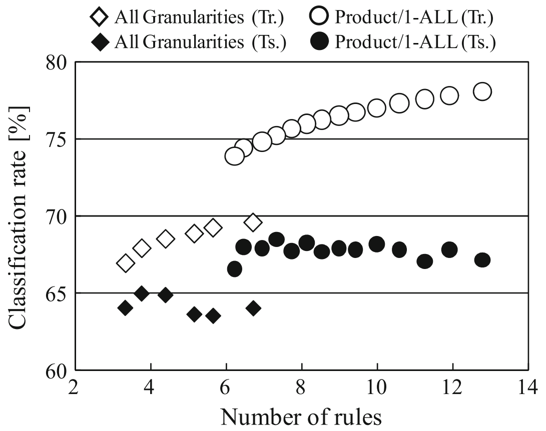


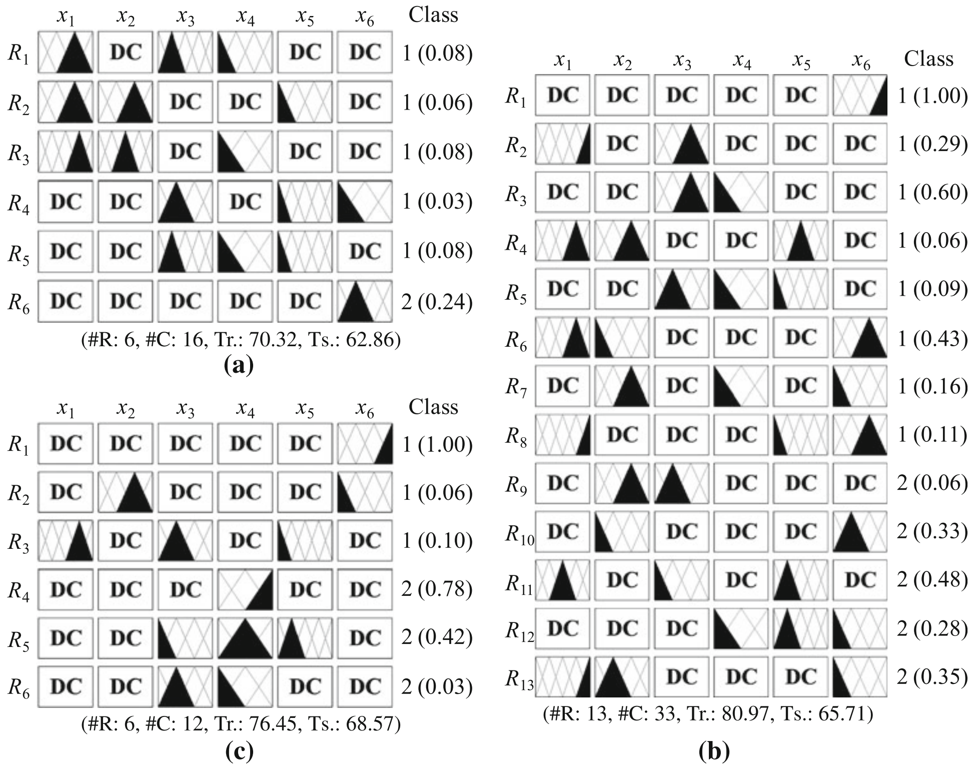
Fig. 2 Average Pareto fronts (training and test) for Bupa dataset with All Granularities and Product/1-ALL

5 On the combination of single granularity specification with the rule selection and the lateral tuning of membership functions: contexts learning

One of the most widely-used approaches to enhance the performance of fuzzy rule-based systems is the one focused on the definition of the membership functions, usually named *tuning of membership functions* (Alcalá et al. 2007a; Gürocak 1999; Herrera et al. 1995; Karr 1991). The tuning methods refine the parameters that identify the membership functions associated to the global linguistic labels. Classically, due to the wide use of the triangular-shaped membership functions, the tuning methods refine the three definition parameters that identify these kinds of membership functions. The tuning techniques could present a positive synergy with the multiobjective fuzzy rule selection, considering the existing dependencies between both parts, rules and membership functions, and representing a way to completely determine the model contexts when granularities are properly specified.

To this end, we will combine the tuning of membership functions with the multiobjective fuzzy rule selection within the Product/1-ALL approach (by modifying the evolutionary algorithm at step 4 of the said learning proposal). As said, evolving both parts concurrently represents a way to improve the accuracy of fuzzy rule-based classifiers; on the other hand, the search space becomes extremely complex to be handled by the state-of-the-art algorithms. In order to reduce the search space, we will exploit the linguistic 2-tuple representation (Alcalá et al. 2007a; Herrera and Martínez 2000), which represents a way to decrement the tuning problem complexity easing indeed the derivation of optimal models.

In the next subsection, we introduce the linguistic 2-tuple representation used for the lateral tuning of membership functions. Then, we stress the positive synergy between rule selection and tuning techniques as a way to enhance the capability of these methods to obtain more accurate and compact fuzzy rule-based classifiers. Finally, we propose the specific multiobjective evolutionary

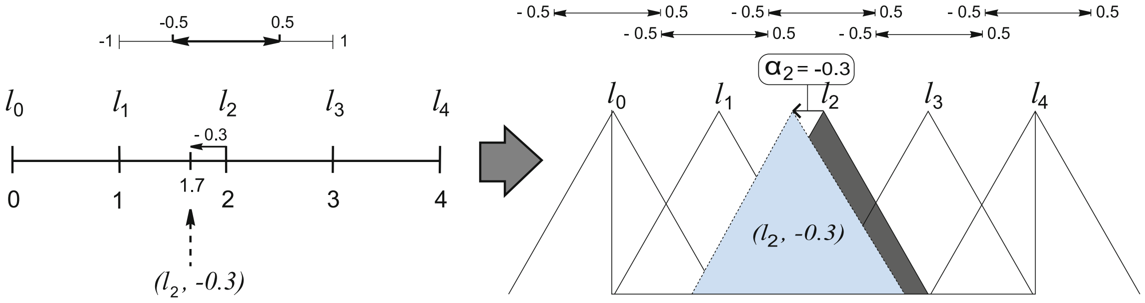

x_1 x_2 x_3 x_4 x_5 x_6 Class
R_1 DC DC DC 1 (0.08)
R_2 DC DC DC 1 (0.06)
R_3 DC DC DC 1 (0.08)
R_4 DC DC DC 1 (0.03)
R_5 DC DC DC 1 (0.08)
R_6 DC DC DC DC DC 2 (0.24)
x_1 x_2 x_3 x_4 x_5 x_6 Class
R_1 DC DC DC DC DC 1 (1.00)
R_2 DC DC DC DC 1 (0.29)
R_3 DC DC DC DC 1 (0.60)
R_4 DC DC DC 1 (0.06)
R_5 DC DC DC 1 (0.09)
R_6 DC DC DC 1 (0.43)
R_7 DC DC DC 1 (0.16)
R_8 DC DC DC 1 (0.11)
R_9 DC DC DC DC 2 (0.06)
R_{10} DC DC DC DC 2 (0.33)
R_{11} DC DC DC 2 (0.48)
R_{12} DC DC DC 2 (0.28)
R_{13} DC DC DC 2 (0.35)
x_1 x_2 x_3 x_4 x_5 x_6 Class
R_1 DC DC DC DC DC 1 (1.00)
R_2 DC DC DC DC 1 (0.06)
R_3 DC DC DC 1 (0.10)
R_4 DC DC DC DC DC 2 (0.78)
R_5 DC DC DC 2 (0.42)
R_6 DC DC DC DC 2 (0.03)

(#R: 6, #C: 16, Tr.: 70.32, Ts.: 62.86)
(a)

(#R: 13, #C: 33, Tr.: 80.97, Ts.: 65.71)
(b)

(#R: 6, #C: 12, Tr.: 76.45, Ts.: 68.57)
(c)

Fig. 3 Three rule set examples in Bupa dataset. **a** by All Granularities, **b** most accurate by Product/1-ALL, **c** with a similar complexity by Product/1-ALL



(a) Simbolic Translation of a Label

(b) Lateral Displacement of a Membership Function

Fig. 4 Symbolic translation of a label and lateral displacement of the associated membership function

algorithm to perform together the rule selection and the lateral tuning of membership functions.

5.1 Lateral tuning of membership functions

In Alcalá et al. (2007a), a new model of tuning of fuzzy rule-based systems was proposed considering the linguistic 2-tuples representation scheme introduced in Herrera and Martínez (2000), which allows the lateral displacement of the support of a label and maintains the interpretability at a good level. This proposal introduces a new model for rule representation based on the concept of symbolic translation (Herrera and Martínez 2000). The symbolic translation of a label is a number in (−0.5, 0.5), expressing with this interval, the domain of a label when it is moving between its two adjacent lateral labels (see Fig. 4a). Let us consider a generic linguistic fuzzy partition $L = \{l_1, \ldots, l_{LB}\}$ (with LB representing the number of labels). Formally, we represent the symbolic translation of a label l_i in L by means of the 2-tuple notation,

$$(l_i, \alpha_i), \quad l_i \in L, \quad \alpha_i \in [-0.5, 0.5).$$

The symbolic translation of a label involves the lateral displacement of its associated membership function. Figure 4 shows the symbolic translation of a label represented by the 2-tuple $(l_2, -0.3)$ together with the associated lateral displacement of the corresponding membership function. Both the linguistic 2-tuples representation model and the elements needed for linguistic information comparison and aggregation were presented and applied to the Decision Making framework in Herrera and Martínez (2000).

In the context of fuzzy rule-based systems, the linguistic 2-tuples could be used to represent the membership functions comprising the linguistic rules. This way to work introduces a new model for rule representation that allows the tuning of the membership functions by learning their respective lateral displacements. The main achievement is that, since the three parameters usually considered per label (Babuška et al. 2002; Bonissone et al. 1996; Cordón et al. 2001; Cordón and Herrera 1997; Herrera et al. 1995; Jang 1993; Karr 1991; Zheng 1992) are reduced to only one symbolic translation parameter, this proposal decreases the learning problem complexity easing indeed the derivation of optimal models.

Notice that from the parameter α applied to each label, we could obtain the equivalent triangular membership functions. Thus, a fuzzy rule-based system based on linguistic 2-tuples can be represented as a classic Mamdani fuzzy rule-based system (Mamdani 1974; Mamdani and Assilian 1975). In this way, from the viewpoint of interpretability:

- the original shapes of the membership functions are maintained (in our case triangular and symmetrical) by laterally changing the location of their supports, and
- the lateral variation of the membership functions is restricted to a short interval, ensuring overlapping among adjacent labels to some degree but preventing their vertex points from crossing.

Finally, in order to avoid very specific parameters and to preserve, as much as possible, the original meanings of the membership functions, we propose the use of a short displacement interval. From different experiments with intervals (−0.5, 0.5), [−0.25, 0.25) and [−0.1, 0.1), we could observe that using values in [−0.1, 0.1) enabled obtaining the same or even better results in order to combine the lateral tuning with the Product/1-ALL approach. In this way, since it also represents a higher interpretability level, we will represent the translation of a linguistic label l_i as,

$$(l_i, \alpha_i), \quad l_i \in L, \quad \alpha_i \in [-0.1, 0.1).$$

5.2 Positive synergy between rule selection and the lateral tuning of membership functions

There are several reasons explaining the positive synergy between the rule selection and the lateral tuning of membership functions. Some of them are:

- Sometimes, redundant rules cannot be removed by only using a rule selection method, since these kinds of rules

could reinforce the action of poor rules improving the model accuracy. The tuning of membership functions can change the performance of these rules making the reinforcement action unnecessary, and therefore, helping the rule selection technique to remove redundant rules.
- The tuning process is affected when too much rules are included in the initial rule set. When the rule base of a model being tuned contains unnecessary rules, the tuning process also tries to improve these kinds of rules, adapting them and the remaining ones to get a good global performance. This way of working imposes strict restrictions, reducing the process ability to obtain precise linguistic models.
- This problem grows as the problem complexity grows (i.e., problems with a large number of variables and/or rules), and when the rule generation method does not ensure the generation of rules with the perfect cooperation, but a large set or candidate rules. In these cases, the tuning process is very complicated because the search ability is dedicated to reduce the bad cooperation of some rules instead of improving the performance of the remaining ones. In these cases, rule selection could help the tuning mechanism by removing the rules that really degrade the accuracy of the model.

Therefore, combining rule selection and tuning approaches could result in important improvements in the system accuracy, maintaining the interpretability at an acceptable level (Alcalá et al. 2006, 2007a; Gacto et al. 2009). However, in some cases, when both techniques are combined, the search space considered is too large, which could provoke the derivation of sub-optimal models (Alcalá et al. 2006).

In this context, the use of the 2-tuple representation can help to reduce the search space by allowing proper convergence to better global solutions that take the existing dependencies between rules and membership function parameters into account. In this section, we propose the selection of a small subset of cooperative rules from a candidate fuzzy rule set together with the learning of the symbolic translation parameters. This pursues the following aims:

- To improve the linguistic model accuracy selecting the set of rules best cooperating, while lateral tuning is performed to improve the location of the membership functions. Notice that it is perfectly compatible with the use of the certainty grades since, as explained in Section 2, using rule weights just affects the strength of each fuzzy if–then rule in the classification phase which does not change the position of the antecedent fuzzy sets. Therefore, they can be seen as complementary approaches in terms of accuracy improvement.
- To obtain simpler, and thus easily understandable, linguistic models by better removing unnecessary rules.
- To preserve the interpretability of the linguistic models since the 2-tuple representation approach does not modify the original shape of membership functions, and the lateral displacements are restricted to a short interval. Further, linguistic 2-tuples could be interpreted as a change in the linguistic meaning of the labels as indicated in Alcalá et al. (2007a).
- To favor the combined action of the tuning and selection strategies (which involves a larger search space) by considering the simpler search space of the lateral tuning (Alcalá et al. 2007a) (only one parameter per label).

5.3 Multiobjective evolutionary algorithm to jointly perform rule selection and lateral tuning (Step 4)

To select the subset of rules which cooperate best and to obtain the lateral translation parameters, we consider a multiobjective genetic algorithm which codes all of them (rules and parameters) in one chromosome. This method is based on the algorithm proposed in Sect. 3.3, again considering the genetic model of NSGA-II (Deb et al. 2002).

To this end, we must take into account the existence of binary genes (rule selection) and real values (lateral displacements) within the same chromosome. Therefore, the algorithm proposed in Sect. 3.3 is extended in order to consider a double coding scheme and to apply the appropriate genetic operators for each chromosome part. The following changes are considered in order to integrate the lateral tuning process within the said multiobjective genetic fuzzy rule selection algorithm:

- *Coding Scheme* A double *coding scheme* for both rule selection and lateral tuning is considered:

  $C = S + T.$

  In this case, the previous approach (part *S*) is combined with the lateral tuning by allowing an additional real vector T that represents the joining of the parameters of the fuzzy partitions. Let us consider the following number of labels per variable: $(m^1, m^2, \ldots, m^n)$, with n being the number of variables. Then, the real-coded vector T has the following form (where each gene is associated to the lateral displacement of the corresponding label),

  $T = (\alpha_1^1, \ldots, \alpha_{m^1}^1, \ldots, \alpha_1^n, \ldots, \alpha_{m^n}^n).$
- *Initial gene pool* The initial pool is obtained with individuals generated at random in $\{0, 1\}$ and $(-0.1, 0.1)$, respectively.
- *Crossover* The uniform crossover presented in subsection 3.3 for the S part combined with the BLX-0.5

(Eshelman and Schaffer 1993) crossover for the T part. The BLX-0.5 operator is applied twice considering T^1 and T^2 in order to obtain the T parts of both offsprings. Let us assume that $T^1 = (x_1, \ldots, x_g)$ and $T^2 = (y_1, \ldots, y_g), (x_i, y_i \in [a_i, b_i) = [-0.1, 0.1) \subset \Re, i = 1, \ldots, g)$, are the two real-coded chromosomes that are going to be crossed. Using the BLX-0.5 crossover, one descendant $Z = (z_1, \ldots, z_g)$ is obtained, where z_i is randomly (uniformly) generated within the interval $[l_i, u_i]$, with $l_i = max\{a_i, c_{min} - I\}, u_i = min\{b_i, c_{max} + I\}, c_{min} = min\{x_i, y_i\}, c_{max} = max\{x_i, y_i\}$ and $I = (c_{max} - c_{min}) \cdot 0.5$ (this 0.5 coming from BLX-0.5). Finally, two descendants are generated by joining the two from the S part with the two from the T part, one by one randomly.

- *Mutation* The bit-flip biased mutation (see Sect. 3.3) for the binary part and a random mutation in one gene of the real part.

The rest of the components of the algorithm remains unchanged.

6 Experiments on the combination with the lateral tuning

In order to analyze the performance of the tuning of membership functions, when it is applied together with the single granularity specification or even together with the original approach using multiple granularities, we compare all the possible combinations among them. They are the original approach without and with tuning (namely All Granularities and All Gr.-TUN), and the mechanism proposed to specify single granularities without and with tuning (namely Product/1-ALL and Product/1-ALL-TUN). For this study, we will follow the same experimental framework presented in Sect. 4. The parameter settings for all these approaches are as follows (same conditions again in all the cases):

- The number of fuzzy rules for each class: 300
- Optimizer: NSGA-II
- Population size: 200
- The number of generations: 5,000
- Crossover probability: 0.9 (uniform crossover for the rule coding, combined with BLX-0.5 for the membership function parameter coding when tuning is performed).
- Mutation probability: 0.05 (from 1 to 0 for the rule coding, 1/Lth (from 0 to 1, where Lth is the string length), combined with randomly changing a value for the membership function parameter coding when tuning is performed.

The results obtained by the different methods are shown in Table 5 (we include again the results from All Granularities and Product/1-ALL in order to ease the comparative analysis). In this case, the approach with the best overall mean results is Product/1-ALL-TUN, presenting more or less the same number of rules than the original/classic approach (therefore, decreasing the complexity with respect to the single granularity-based algorithm without tuning). In order to assess whether we can conclude that completely specifying the model contexts, i.e. using Product/1-ALL-TUN, statistically outperforms the remaining approaches in terms of test classification percentage, we apply Wilcoxon's test to the results achieved by this approach and the remaining algorithms in the most accurate solutions. Table 6 shows these results. The null hypothesis associated with the Wilcoxon's test is now rejected ($p < \alpha$) in all the cases in favor of Product/1-ALL-TUN due to the differences between R^+ and R^-. Thus, we can conclude that performing tuning on the prespecified granularities represents a way to obtain better classifiers with at least the same complexity than the previous approaches.

With respect to the tuning on the multiple granularity-based approach, we can observe that taking into account the results in Table 6, Product/1-ALL-TUN also outperforms this approach statistically. In fact, it presents almost the same overal mean on the test classification percentage than its respective counterpart, and it is even worse than Product/1-ALL, obtaining probably too simple models. Table 7 shows the results of the Wilcoxon's test on the test values for this method and both approaches without tuning. The null hypothesis is now accepted for both methods, showing that the tuning is not useful on the approach based on multiple granularities. Therefore, it seems that the tuning is still able to provide some kind of improvement in the single granularity model but not too much improvement, when the data is more or less well covered by using multiple granularities.

If we closely look at experimental results on each dataset by the four methods, we can observe the following general facts on the application of the tuning:

- Whereas the differences in the classification accuracy among the four methods are not so large on many data sets, they are large on some data sets.
- Whereas the tuning improved the classification accuracy on many data sets, it deteriorated not only the test data accuracy (probably due to the overfitting) but also the training data accuracy (due to the complexity increasing) on some data sets.

Figures 5 and 6 plot the average Pareto fronts on training and test sets at Pima and Sonar datasets on the four methods.

Table 5 Results obtained by the studied methods (most accurate)

Datasets	All Granularities				Product/1-ALL				All Gr.-TUN				Product/1-ALL-TUN			
	#R	#C	Tr.	Ts.	#R	#C	Tr.	Ts.	#R	#C	Tr.	Ts.	#R	#C	Tr.	Ts.
Appendicitis	2.37	3.73	91.86	87.91	3.40	6.97	93.29	**88.21**	2.27	3.43	91.79	87.00	3.13	6.33	93.47	87.30
Australian	2.00	2.00	85.51	85.51	6.63	16.17	88.82	85.31	2.23	5.47	86.73	85.60	5.13	12.17	89.18	**85.65**
Bands	4.60	12.43	71.36	68.73	6.83	17.53	69.90	65.37	4.73	12.87	72.68	**68.82**	6.63	15.70	71.18	65.80
Bupa	6.73	16.73	69.50	63.99	12.80	31.27	78.06	67.11	4.93	12.23	70.53	63.20	9.60	21.97	78.59	**67.19**
Cleveland	20.17	56.13	73.11	55.11	28.67	77.77	76.72	52.83	17.70	49.03	74.62	54.53	19.43	52.53	77.21	**58.80**
Dermatology	11.40	19.33	99.07	94.12	13.53	23.47	99.51	93.26	9.13	14.93	99.10	**95.23**	10.63	17.43	99.28	94.48
Glass	12.63	32.37	78.65	60.48	19.30	44.73	83.97	69.96	10.53	25.73	78.75	64.31	15.37	34.80	83.68	**71.28**
Haberman	6.50	13.93	79.46	71.89	3.00	6.00	74.70	**73.19**	5.50	12.30	79.82	71.02	3.37	6.27	76.82	71.88
Hayes-roth	9.17	15.27	90.88	78.03	10.83	16.90	90.91	**79.14**	8.20	12.83	90.49	77.48	9.97	15.90	90.99	78.88
Heart	7.67	14.27	90.19	83.46	10.13	23.97	92.81	**84.32**	6.50	11.90	90.01	82.96	8.77	18.77	91.87	82.84
Hepatitis	3.80	9.30	96.10	**90.44**	4.33	9.40	98.20	87.00	3.10	7.70	96.71	90.38	2.97	6.83	97.88	88.53
Ionosphere	9.53	14.83	95.64	88.62	10.50	15.70	95.99	90.79	6.70	9.97	95.47	88.63	8.13	10.67	96.25	**90.79**
Iris	4.03	6.80	99.11	95.11	5.23	7.53	98.30	95.33	4.10	6.90	99.43	94.00	4.03	4.60	98.30	**97.33**
Mammographic	6.97	14.70	83.07	81.04	11.67	25.27	83.30	79.35	5.37	10.30	82.96	**81.05**	7.13	14.97	83.90	80.49
Newthyroid	5.37	9.20	96.19	91.78	7.37	15.73	97.59	93.01	4.67	8.33	97.79	94.12	5.43	10.40	98.02	**94.60**
Pasture	3.70	5.87	98.05	75.83	4.43	8.13	100.00	73.61	3.47	5.13	99.39	78.33	3.90	6.47	99.27	**80.56**
Pima	6.63	14.33	77.80	74.92	10.63	25.87	79.04	73.79	5.70	12.83	78.51	75.97	5.20	11.00	79.06	**77.05**
Saheart	5.97	12.77	76.70	71.14	12.33	31.80	79.00	**71.22**	4.60	9.57	76.93	70.70	6.80	16.30	77.73	70.13
Sonar	6.73	10.50	86.54	78.88	6.87	12.60	87.82	77.48	5.10	7.40	87.04	73.77	5.27	9.03	87.91	**78.90**
Tae	7.77	18.93	66.55	54.57	11.33	25.43	65.98	59.24	7.47	17.50	69.78	55.69	9.77	22.10	71.21	**60.78**
Vehicle	13.77	35.77	69.34	62.81	15.60	43.10	70.80	**66.20**	10.77	26.87	69.44	63.91	11.93	32.03	71.11	66.16
Wdbc	5.67	9.83	97.12	94.90	7.27	12.40	97.26	93.96	4.87	7.67	97.30	**95.49**	5.13	8.63	97.33	94.90
Wine	3.90	8.23	100.00	**96.08**	6.37	12.10	99.92	95.11	3.90	7.13	100.00	94.14	5.80	10.17	99.92	93.03
Wisconsin	6.93	11.17	98.22	96.07	7.77	13.83	98.40	96.06	5.57	9.00	98.19	95.82	6.30	10.97	98.33	**96.35**
Mean	7.25	15.35	86.25	79.23	9.87	21.82	87.51	79.62	6.13	12.79	86.81	79.26	7.49	15.67	87.85	**80.57**

The bests in test are bold-faced

Table 6 Wilcoxon's test: Approaches without tuning or tuning on All Granularities (R^+) versus Product/1-ALL-TUN (R^-) on the test error at the most accurate solution

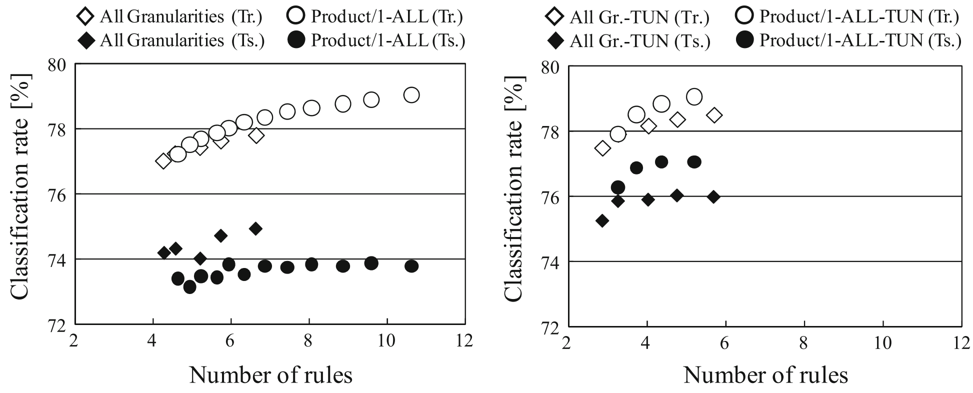
Comparison	R^+	R^-	Hypothesis ($\alpha = 0.1$)	p value
All Granularities versus Product/1-ALL-TUN	85	215	Rejected	0.063
Product/1-ALL versus Product/1-ALL-TUN	74	226	Rejected	0.030
All Gr.-TUN versus Product/1-ALL-TUN	76	224	Rejected	0.034

Table 7 Wilcoxon's test: Approaches without tuning (R^+) versus All Gr.-TUN (R^-) on the test error at the most accurate solution

Comparison	R^+	R^-	Hypothesis ($\alpha = 0.1$)	p value
All Granularities versus All Gr.-TUN	143	157	Accepted	0.841
Product/1-ALL versus All Gr.-TUN	172	128	Accepted	0.530

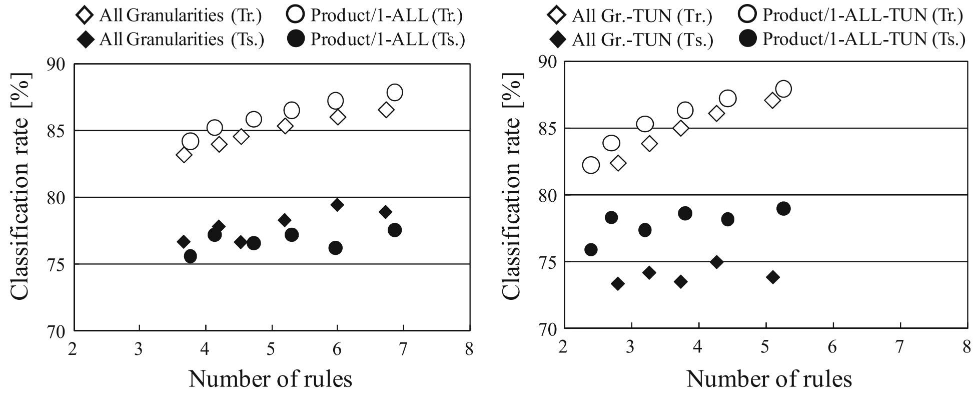
As we can see, the complexities are significantly decreased giving way to Pareto fronts with similar lengths to those obtained by the original approach. In the first case, Pima, we can observe that both approaches improve the test classification percentages when the tuning is applied, making the single granularity-based approach to overtake the multiple granularities-based one. In the second case, Sonar, we can observe that the tuning overfits when it is applied on All Granularities, whereas the test still improves when it is applied on the single granularities-based approach.

Additionally, two rule sets obtained by both approaches with tuning (most accurate solutions) are depicted in



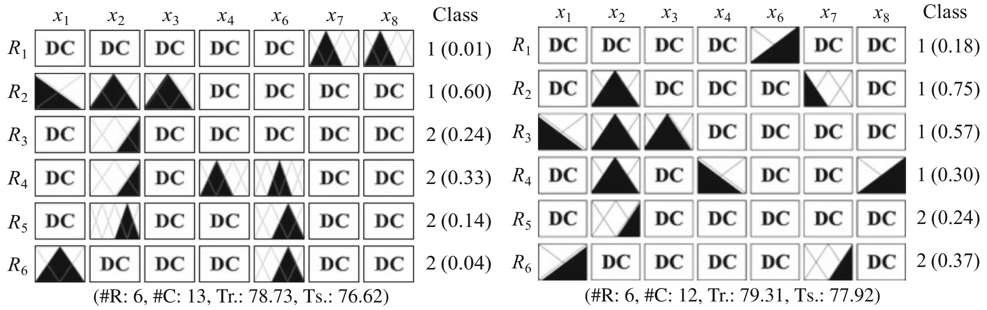
(a) Without lateral tuning **(b)** With lateral tuning

Fig. 5 Average Pareto fronts (training and test) on Pima dataset from All Granularities and Product/1-ALL without and with tuning



(a) Without lateral tuning **(b)** With lateral tuning

Fig. 6 Average Pareto fronts (training and test) on Sonar dataset from All Granularities and Product/1-ALL without and with tuning


x1 x2 x3 x4 x6 x7 x8 Class
R1 DC DC DC DC DC 1 (0.01)
R2 DC DC DC DC 1 (0.60)
R3 DC DC DC DC DC DC 2 (0.24)
R4 DC DC DC DC 2 (0.33)
R5 DC DC DC DC DC 2 (0.14)
R6 DC DC DC DC DC 2 (0.04)
(#R: 6, #C: 13, Tr.: 78.73, Ts.: 76.62)
x1 x2 x3 x4 x6 x7 x8 Class
R1 DC DC DC DC DC DC 1 (0.18)
R2 DC DC DC DC DC 1 (0.75)
R3 DC DC DC DC 1 (0.57)
R4 DC DC DC DC 1 (0.30)
R5 DC DC DC DC DC DC 2 (0.24)
R6 DC DC DC DC DC 2 (0.37)
(#R: 6, #C: 12, Tr.: 79.31, Ts.: 77.92)

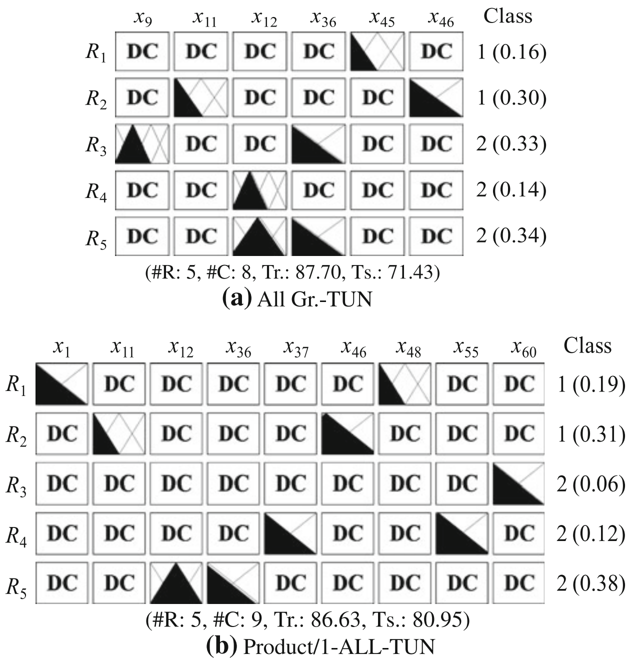
(a) All Gr.-TUN **(b)** Product/1-ALL-TUN

Fig. 7 Two tuned rule sets with multiple (**a**) and with single (**b**) granularity for Pima dataset

Figs. 7 and 8, respectively, at Pima and Sonar. Thanks to the fact that no strong changes are required on the membership functions to reach the highest level of accuracy, we can observe that in all the cases the membership functions are very close to the original ones. In this way, we can consider that the interpretability is highly preserved. Therefore, the main differences are again that we could easily represent the rule bases obtained from Product/1-ALL-TUN in terms of the associated linguistic labels (if an expert is able to provide them), which is not possible with the All Granularities-based approach.

7 Concluding remarks

In this work, we have analyzed and extended a method to generate single granularity-based fuzzy classification rules for multiobjective genetic fuzzy rule selection. After extracting a prespecified number of fuzzy rules with multiple granularities, a single granularity is specified for each attribute individually according to the frequency of employed partitions and the importance of the multiple granularity-based extracted rules. Then, multiobjective genetic fuzzy rule selection is applied in order to obtain a


x_9 x_{11} x_{12} x_{36} x_{45} x_{46} Class
R_1 DC DC DC DC DC 1 (0.16)
R_2 DC DC DC DC 1 (0.30)
R_3 DC DC DC DC 2 (0.33)
R_4 DC DC DC DC DC 2 (0.14)
R_5 DC DC DC DC 2 (0.34)
x_1 x_{11} x_{12} x_{36} x_{37} x_{46} x_{48} x_{55} x_{60} Class
R_1 DC DC DC DC DC DC DC 1 (0.19)
R_2 DC DC DC DC DC DC DC 1 (0.31)
R_3 DC DC DC DC DC DC DC DC 2 (0.06)
R_4 DC DC DC DC DC DC DC 2 (0.12)
R_5 DC DC DC DC DC DC DC 2 (0.38)

(#R: 5, #C: 8, Tr.: 87.70, Ts.: 71.43)
(a) All Gr.-TUN

(#R: 5, #C: 9, Tr.: 86.63, Ts.: 80.95)
(b) Product/1-ALL-TUN

Fig. 8 Two tuned rule sets with multiple (**a**) and with single (**b**) granularity for Sonar dataset

set of non-dominated solutions with a good tradeoff between complexity and accuracy. Additionally, it has been combined with the lateral tuning of membership functions as a way to analyze the importance of contexts learning in the derivation of linguistic fuzzy rule-based classification systems.

The different approaches have been statistically compared on a set of 24 well-known datasets. As well as the interpretability improvement that the use of a fixed single granularity involves, the results show that the performance of the obtained classifiers can be maintained with respect to the use of multiple granularities. On the other hand, we can conclude that combining single granularity specification and a slight tuning of the membership functions gives way to more accurate models with similar complexities to the original/classic approach based on multiple granularities. By contrast, the tuning is not effective when it is combined with the use of multiple granularities, which is only showing slight improvements on the complexities.

Taking into account the results obtained, we can highlight the following general tendencies:

- With respect to the complexity (in terms of the number of fuzzy rules and the total rule length), all complexity-based approaches are able to obtain lower complexities but similar in the best cases. Further, the application of a tuning clearly helps to decrement the complexity with respect to the corresponding counterparts.
- With respect to the classification accuracy, the differences in the classification accuracy among the four methods are not so large on many data sets if compared with differences in the complexity.

Finally, to sum-up, we can conclude that the tuning is highly promising on single granularities but not on multiple granularities. This demonstrates the importance of completely determining appropriate contexts, i.e., appropriate granularities and membership function parameters.

Acknowledgments This work was supported by the Spanish CICYT Project TIN2008-06681-C06-01 and the Japanese Grant-in-Aid for Young Scientists (B): KAKENHI (22700239).

Appendix: Wilcoxon signed-rank test

The Wilcoxon signed-rank test is a pair-wise test that aims to detect significant differences between two sample means: it is the analogous to the paired t test in non-parametric statistical procedures. If these means refer to the outputs of two algorithms, then the test practically assesses the reciprocal behavior of the two algorithms (Sheskin 2003; Wilcoxon 1945). Let d_i be the difference between the performance scores of the two algorithms on the ith out of N_{ds} datasets. The differences are ranked according to their absolute values; average ranks are assigned in case of ties. Let R^+ be the sum of ranks for the datasets on which the first algorithm outperformed the second, and R^- the sum of ranks for the contrary outcome. Ranks of $d_i = 0$ are split evenly among the sums; if there is an odd number of them, one is ignored:

$$R^+ = \sum_{d_i > 0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i),$$

$$R^- = \sum_{d_i < 0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i).$$

Let T be the smaller of the sums, $T = \min(R^+, R^-)$. If T is less than, or equal to, the value of the distribution of Wilcoxon for N_{ds} degrees of freedom [Table B.12 in Zar (1999)], the null hypothesis of equality of means is rejected.

The Wilcoxon signed-rank test is more sensible than the t test. It assumes commensurability of differences, but only qualitatively: greater differences still count for more, which is probably desired, but the absolute magnitudes are ignored. From the statistical point of view, the test is safer since it does not assume normal distributions. Also, the outliers (exceptionally good/bad performances on a few datasets) have less effect on the Wilcoxon test than on the t test. The Wilcoxon test assumes continuous differences d_i; therefore, they should not be rounded to one or two decimals, since this would decrease the test power due to a high number of ties.

When the assumptions of the paired *t* test are met, the Wilcoxon signed-rank test is less powerful than the paired *t* test. On the other hand, when the assumptions are violated, the Wilcoxon test can be even more powerful than the *t* test. This allows us to apply it to the means obtained by the algorithms in each dataset, without any assumption about the distribution of the obtained results.

References

Agrawal R, Mannila H, Srikant R, Toivonen H, Verkamo AI (1996) Fast discovery of association rules. In: Fayyad UM, Piatetsky-Shapiro G, Smyth P, Uthurusamy R (eds) Advances in knowledge discovery and data mining. AAAI, Menlo Park, pp 307–328

Alcalá R, Cano JR, Cordón O, Herrera F, Villar P, Zwir I (2003) Linguistic modeling with hierarchical systems of weighted linguistic rules. Int J Approx Reason 32(2–3):187–215

Alcalá R, Alcalá-Fdez J, Casillas J, Cordón O, Herrera F (2006) Hybrid learning models to get the interpretability–accuracy trade-off in fuzzy modeling. Soft Comput 10(9):717–734

Alcalá R, Alcalá-Fdez J, Herrera F (2007a) A proposal for the genetic lateral tuning of linguistic fuzzy systems and its interaction with rule selection. IEEE Trans Fuzzy Syst 15(4):616–635

Alcalá R, Gacto MJ, Herrera F, Alcalá-Fdez J (2007b) A multi-objective genetic algorithm for tuning and rule selection to obtain accurate and compact linguistic fuzzy rule-based systems. Int J Uncertain Fuzziness Knowl Based Syst 15(5):539–557

Alcalá R, Ducange P, Herrera F, Lazzerini B, Marcelloni F (2009a) A multi-objective evolutionary approach to concurrently learn rule and data bases of linguistic fuzzy rule-based systems. IEEE Trans Fuzzy Syst 17(5):1106–1122

Alcalá R, Nojima Y, Herrera F, Ishibuchi H (2009b) Generating single granularity-based fuzzy classification rules for multiobjective genetic fuzzy rule selection. In: Proceedings of the 2009 IEEE International Conference on Fuzzy Systems, Jeju (Korea), pp 1718–1723

Alcalá-Fdez J, Sánchez L, García S, del Jesus M, Ventura S, Garrell J, Otero J, Romero C, Bacardit J, Rivas V, Fernández J, Herrera F (2009) KEEL: a software tool to assess evolutionary algorithms to data mining problems. Soft Comput 13(3):307–318

Asuncion A, Newman DJ (2007) UCI machine learning repository. http://www.ics.uci.edu/~mlearn/MLRepository.html

Babuška R, Oosterhoff J, Oudshoorn A, Bruijn PM (2002) Fuzzy self-tuning PI control of pH in fermentation. Eng Appl Artif Intell 15(1):3–15

Bonissone PP, Khedar PS, Chen YT (1996) Genetic algorithms for automated tuning of fuzzy controllers, a transportation aplication. In: Proceedings of the Fifth IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'96). Nueva Orleans, LA, EE. UU, pp 674–680

Botta A, Lazzerini B, Marcelloni F, Stefanescu DC (2009) Context adaptation of fuzzy systems through a multi-objective evolutionary approach based on a novel interpretability index. Soft Comput 13(5):437–449

Casillas J, Cordón O, Herrera F, Magdalena L (eds) (2003) Interpretability issues in fuzzy modeling, Studies in Fuzziness and Soft Computing, vol 128. Springer, Heidelberg

Cococcioni M, Ducange P, Lazzerini B, Marcelloni F (2007) A pareto-based multi-objective evolutionary approach to the identification of mamdani fuzzy systems. Soft Comput 11:1013–1031

Coello CA, Veldhuizen DAV, Lamont GB (eds) (2002) Evolutionary algorithms for solving multi-objective problems. Kluwer, Boston

Cordón O, Herrera F (1997) A three-stage evolutionary process for learning descriptive and approximative fuzzy logic controller knowledge bases from examples. Int J Approx Reason 17(4):369–407

Cordón O, Herrera F, Hoffmann F, Magdalena L (2001) Genetic fuzzy systemes. Evolutionary tuning and learning of fuzzy knowledge bases. In: Advances in fuzzy systems—applications and theory, vol 19. World Scientific, Singapore

Deb K (2001) Multi-objective optimization using evolutionary algorithms. Wiley, New York

Deb K, Pratab A, Agrawal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197

Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30

Eshelman LJ, Schaffer JD (1993) Real-coded genetic algorithms and interval-schemata. Found Genet Algorithms 2:187–202

Gacto MJ, Alcalá R, Herrera F (2010) Integration of an index to preserve the semantic interpretability in the multi-objective evolutionary rule selection and tuning of linguistic fuzzy systems. IEEE Trans Fuzzy Syst 18(3):515–531

Gacto MJ, Alcalá R, Herrera F (2009) Adaptation and application of multi-objective evolutionary algorithms for rule reduction and parameter tuning of fuzzy rule-based systems. Soft Comput 13(5):419–436

García S, Fernández A, Luengo J, Herrera F (2009) A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability. Soft Comput 13(10):959–977

García S, Herrera F (2008) An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons. J Mach Learn Res 9:2677–2694

García S, Molina D, Lozano M, Herrera F (2009) A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization. J Heuristics 15(6):617–644

Gürocak HB (1999) A genetic-algorithm-based method for tuning fuzzy logic controllers. Fuzzy Sets Syst 108(1):39–47

Herrera F, Lozano M, Verdegay JL (1995) Tuning fuzzy logic controllers by genetic algorithms. Int J Approx Reason 12:299–315

Herrera F, Martínez L (2000) A 2-tuple fuzzy linguistic representation model for computing with words. IEEE Trans Fuzzy Syst 8(6):746–752

Ishibuchi H, Ishibuchi H, Nakashima T, Nakashima T (2000) Effect of rule weights in fuzzy rule-based classification systems. IEEE Trans Fuzzy Syst 9:506–515

Ishibuchi H, Murata T, Turksen IB (1995) Selecting linguistic classification rules by two-objective genetic algorithms. In: Proceedings of the IEEE International Conference on Systems, Man, and Cybernatics. Vancouver, Canada, pp 1410–1415

Ishibuchi H, Murata T, Türksen IB (1997) Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems. Fuzzy Sets Syst 89(2):135–150

Ishibuchi H, Nakashima T, Nii M (2004) Classification and modeling with linguistic information granules: advanced approaches to linguistic data mining. Springer, Berlin

Ishibuchi H, Nojima Y (2007) Analysis of interpretability–accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning. Int J Approx Reason 44(1):4–31

Ishibuchi H, Yamamoto T (2004) Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining. Fuzzy Sets Syst 141(1):59–88

Ishibuchi H, Yamamoto T (2005) Rule weight specification in fuzzy rule-based classification systems. IEEE Trans Fuzzy Syst 13:428–435

Jang JSR (1993) ANFIS: Adaptive network based fuzzy inference system. IEEE Trans Syst Man Cybern 23(3):665–684

Karr C (1991) Genetic algorithms for fuzzy controllers. AI Expert 6(2):26–33

Mamdani E (1974) Application of fuzzy algorithms for control of simple dynamic plant. In: Proc. IEEE, vol 121, pp 1585–1588

Mamdani E, Assilian S (1975) An experiment in linguistic synthesis with a fuzzy logic controller. Int J Man Mach Stud 7:1–13

Nojima Y, Ishibuchi H, Kuwajima I (2009) Parallel distributed genetic fuzzy rule selection. Soft Comput 13(5):511–519

Pulkkinen P, Koivisto H (2008) Fuzzy classifier identification using decision tree and multiobjective evolutionary algorithms. Int J Approx Reason 48:526–543

Pulkkinen P, Koivisto H (2010) A dynamically constrained multiobjective genetic fuzzy system for regression problems. IEEE Trans Fuzzy Syst 18(1):161–177

Sánchez L, Otero J, Couso I (2009) Obtaining linguistic fuzzy rule-based regression models from imprecise data with multiobjective genetic algorithms. Soft Comput 13(5):467–479

Sheskin D (2003) Handbook of parametric and nonparametric statistical procedures. Chapman & Hall/CRC, Boca Raton

Wilcoxon F (1945) Individual comparisons by ranking methods. Biometrics 1:80–83

Zar J (1999) Biostatistical analysis. Prentice-Hall, Upper Saddle River

Zheng L (1992) A practical guide to tune proportional and integral (pi) like fuzzy controllers. In: Proceedings of the First IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'92). San Diego, pp 633–640