Accuracy-Complexity Tradeoff Analysis by Multiobjective Rule Selection

Hisao Ishibuchi
Osaka Prefecture University
hisaoi@cs.osakafu-u.ac.jp

Yusuke Nojima
Osaka Prefecture University
nojima@cs.osakafu-u.ac.jp

Abstract

In this paper, we clearly demonstrate that genetics-based multiobjective rule selection can significantly improve the accuracy-complexity tradeoff curve of extracted rule sets for classification problems. First a prespecified number of rules are extracted from numerical data with continuous attributes using a heuristic rule extraction criterion. Then genetics-based multiobjective rule selection is applied to the extracted rule set to find a number of non-dominated rule subsets with respect to the classification accuracy and the number of rules. Experimental results clearly show that multiobjective rule selection finds a number of smaller rule subsets with higher accuracy than heuristically extracted rule sets. That is, the accuracy-complexity tradeoff curve is improved by multiobjective rule selection.

1. Introduction

Almost all real-world decision making problems involve multiple objectives. These objectives usually conflict with each other. In the case of knowledge extraction, we want to maximize the accuracy of extracted rules. At the same time, we want to minimize their complexity (i.e., maximize their interpretability). Evolutionary multiobjective optimization (EMO) is an active research area in the field of evolutionary computation (see Deb [1] and Coello et al. [2]). The main advantage of EMO approaches over conventional optimization techniques is that a number of non-dominated solutions are simultaneously obtained by their single run. The obtained non-dominated solutions help the decision maker to understand the tradeoff structure between conflicting objectives (e.g., through their visualization). Such knowledge about the tradeoff structure in turn helps the decision maker to choose the final solution from the obtained non-dominated ones.

In some conventional (i.e., non-evolutionary) approaches to multiobjective optimization, the decision maker is supposed to integrate multiple objectives into a single scalar objective function by assigning a relative weight to each objective. The assessment of the relative weight, however, is usually very difficult because the decision maker has no *a priori* information about the tradeoff between conflicting objectives. For example, it is very difficult to assign relative weights to the two major goals in knowledge extraction: accuracy maximization and complexity minimization. In other conventional approaches, the decision maker is requested to assign the target value to each objective. The specification of the target value is also difficult for the decision maker. For example, it is not easy to specify the target values for the classification accuracy and the number of extracted rules before the decision maker knows the tradeoff structure between the accuracy and the complexity of rule sets for a particular classification problem at hand.

Recently EMO approaches have been employed in some studies on modeling and classification. For example, Kupinski & Anastasio [3] used an EMO algorithm to generate non-dominated neural networks on a receiver operating characteristic curve. Gonzalez et al. [4] generated non-dominated radial basis function networks of different sizes. Abbass [5] used a memetic EMO algorithm (i.e., a hybrid EMO algorithm with local search) to speed up the back-propagation algorithm where multiple neural networks of different sizes were evolved to find an appropriate network structure. Non-dominated neural networks were combined into a single ensemble classifier in [6]-[8]. The use of EMO algorithms to design ensemble classifiers was also proposed in Ishibuchi & Yamamoto [9] where multiple fuzzy rule-based classifiers of different sizes were generated. In some studies on fuzzy rule-based systems [10]-[17], EMO algorithms were used to analyze the tradeoff between accuracy and interpretability.

In this paper, we intend to clearly demonstrate the effectiveness of EMO approaches to knowledge extraction from numerical data for classification problems with many continuous attributes. First we briefly explain some basic concepts in multiobjective optimization in Section 2. Next we explain our EMO approach to knowledge extraction. Our approach consists of two stages: heuristic rule extraction (i.e., data mining stage) and genetics-based multiobjective rule selection (i.e., optimization stage). These two stages are described in Section 3 and Section 4, respectively. In the second stage of our EMO approach, knowledge extraction is formulated as a two-objective rule

selection problem. The two objectives are to maximize the classification accuracy and to minimize the number of rules. An EMO algorithm is employed to efficiently find a number of non-dominated rule sets with respect to these two objectives for classification problems with many continuous attributes. In Section 5, obtained non-dominated rule sets are compared with heuristically extracted rule sets. Finally Section 6 concludes this paper.

2. Multiobjective Optimization

We explain some basic concepts in multiobjective optimization using the following *k*-objective problem:

$$\text{Minimize } \mathbf{z} = (f_1(\mathbf{y}), f_2(\mathbf{y}), ..., f_k(\mathbf{y})), \quad (1)$$

$$\text{subject to } \mathbf{y} \in \mathbf{Y}, \quad (2)$$

where **z** is the objective vector, **y** is the decision vector, and **Y** is the feasible region in the decision space. Since the *k* objectives usually conflict with each other, there is no absolutely optimal solution $\mathbf{y}^*$ ($\mathbf{y}^* \in \mathbf{Y}$) that satisfies the following relation with respect to all objectives:

$$\forall i \quad f_i(\mathbf{y}^*) = \min\{f_i(\mathbf{y}) : \mathbf{y} \in \mathbf{Y}\}. \quad (3)$$

In general, multiobjective optimization problems have a number of non-dominated (i.e., Pareto-optimal) solutions. Now we briefly explain the concept of Pareto-optimality. Let **a** and **b** be two feasible solutions of the *k*-objective problem in (1)-(2). When the following condition holds, **a** can be viewed as being better than **b**:

$$\forall i \quad f_i(\mathbf{a}) \le f_i(\mathbf{b}) \quad \text{and} \quad \exists j \quad f_j(\mathbf{a}) < f_j(\mathbf{b}). \quad (4)$$

In this case, we say that **a** dominates **b** (equivalently **b** is dominated by **a**). This dominance relation between **a** and **b** in (4) is sometimes denoted as $\mathbf{a} \prec \mathbf{b}$.

When **b** is not dominated by any other feasible solutions, **b** is referred to as a non-dominated (i.e., Pareto-optimal) solution of the *k*-objective problem in (1)-(2). That is, **b** is a Pareto-optimal solution when there is no feasible solution **a** that satisfies $\mathbf{a} \prec \mathbf{b}$. The set of all Pareto-optimal solutions forms a tradeoff surface in the *k*-dimensional objective space. This tradeoff surface in the objective space is referred to as the Pareto-front. Various EMO algorithms have been proposed to efficiently find a number of Pareto-optimal (or near Pareto-optimal) solutions that are uniformly distributed on the Pareto-front [1]-[2].

3. Heuristic Extraction of Classification Rules

Our EMO approach to knowledge extraction consists of two stages: heuristic rule extraction and genetics-based multiobjective rule selection. In the first stage (i.e., data mining stage), a prespecified number of promising rules are efficiently extracted in a heuristic manner. Then a number of non-dominated rule sets, which are subsets of the extracted rules, are found by an EMO algorithm in the second stage (i.e., optimization stage). These two stages are explained in this section and the next section, respectively.

Let us assume that we have *m* training (i.e., labeled) patterns $\mathbf{x}_p = (x_{p1}, ..., x_{pn})$, $p = 1, 2, ..., m$ from *M* classes in the *n*-dimensional continuous pattern space where x_{pi} is the attribute value of the *p*-th training pattern for the *i*-th attribute ($i = 1, 2, ..., n$). We denote these training patterns by *D* (i.e., $D = \{\mathbf{x}_1, ..., \mathbf{x}_m\}$). We also denote training patterns from Class *h* as *D*(Class *h*) where $h = 1, 2, ..., M$.

For our *n*-dimensional *M*-class classification problem, we use the following classification rule:

$$\text{Rule } R_q: \text{If } x_1 \text{ is } A_{q1} \text{ and } ... \text{ and } x_n \text{ is } A_{qn} \text{ then Class } C_q \text{ with } CF_q, \quad (5)$$

where R_q is the label of the *q*-th rule, $\mathbf{x} = (x_1, ..., x_n)$ is an *n*-dimensional pattern vector, A_{qi} is an antecedent interval, C_q is a class label, and CF_q is a rule weight (i.e., certainty grade). Each antecedent condition "x_i is A_{qi}" means the inclusion relation $x_i \in A_{qi}$ (i.e., the inequality relation $A_{qi}^L \le x_i \le A_{qi}^U$ where $A_{qi} = [A_{qi}^L, A_{qi}^U]$). We denote the antecedent part of the classification rule R_q in (5) by the interval vector $\mathbf{A}_q$ where $\mathbf{A}_q = (A_{q1}, ..., A_{qn})$. Thus R_q is denoted as "$\mathbf{A}_q \Rightarrow$ Class C_q".

The first step to heuristic rule extraction is the discretization of the domain interval of each continuous attribute into antecedent intervals. Since we usually have no *a priori* information about an appropriate granularity of the discretization for each attribute, we simultaneously use multiple partitions with different granularities (i.e., from coarse partitions into a few intervals to fine partitions into many intervals). This is one characteristic feature of our approach to knowledge extraction. Since we simultaneously use multiple partitions with different granularities, we need no heuristic criteria to compare different granularities (i.e., to determine the number of intervals for each attribute). In computational experiments, we use five partitions into *K* intervals where $K = 1, 2, 3, 4, 5$ (see Fig. 1). It should be noted that $K = 1$ corresponds to the whole domain interval.

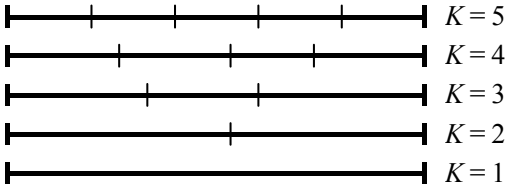

Fig. 1. Five partitions with different granularities used in our computational experiments.

As shown in Fig. 1, the whole domain interval is divided

into K intervals. To specify $(K-1)$ cutting points for each attribute, we use an optimal splitting method [18] based on the class entropy measure [19]:

$$H(A_1, ..., A_K) = -\sum_{j=1}^{K} \frac{|D_j|}{|D|} \sum_{h=1}^{M} \left(\frac{|D_{jh}|}{|D_j|} \cdot \log_2 \frac{|D_{jh}|}{|D_j|} \right), \quad (6)$$

where $(A_1, ..., A_K)$ is K intervals generated by the discretization of an attribute, D_j is the set of training patterns in the interval A_j, and D_{jh} is the set of training patterns from Class h in D_j. Using the optimal splitting method [18], we can efficiently find the optimal $(K-1)$ cutting points that minimize the class entropy measure in (6). In this manner, we can obtain multiple partitions for various values of K for each attribute.

When we use five partitions with $K = 1, 2, 3, 4, 5$ in Fig. 1, we have 15 antecedent intervals for each attribute. This means that we have 15^n combinations of the antecedent intervals for our n-dimensional classification problem. Such a combination corresponds to the antecedent part of each classification rule in (5).

The next step to heuristic rule extraction is the determination of the consequent class C_q and the rule weight CF_q for each combination $\mathbf{A}_q$ of the antecedent intervals. This is performed by calculating the confidence of the classification rule "$\mathbf{A}_q \Rightarrow$ Class h" for each class h (see [20] for the confidence measure). Let $D(\mathbf{A}_q)$ be the set of compatible training patterns with the antecedent part $\mathbf{A}_q$:

$$D(\mathbf{A}_q) = \{\mathbf{x}_p \mid x_{p1} \in A_{q1}, ..., x_{pn} \in A_{qn}\}. \quad (7)$$

When $D(\mathbf{A}_q)$ is empty, we do not generate any rule with the antecedent part $\mathbf{A}_q$.

The confidence of "$\mathbf{A}_q \Rightarrow$ Class h" is calculated as

$$c(\mathbf{A}_q \Rightarrow \text{Class } h) = \left| D(\mathbf{A}_q) \cap D(\text{Class } h) \right| \Big/ \left| D(\mathbf{A}_q) \right|, \quad h = 1, 2, ..., M. \quad (8)$$

The confidence of "$\mathbf{A}_q \Rightarrow$ Class h" in (8) is the ratio of compatible training patterns with $\mathbf{A}_q$ from Class h to all the compatible training patterns. Another measure called support has also been frequently used in the literature [20]. The support of "$\mathbf{A}_q \Rightarrow$ Class h" is calculated as

$$s(\mathbf{A}_q \Rightarrow \text{Class } h) = \left| D(\mathbf{A}_q) \cap D(\text{Class } h) \right| \Big/ \left| D \right|, \quad h = 1, 2, ..., M. \quad (9)$$

The consequent class C_q is specified as the class with the maximum confidence:

$$c(\mathbf{A}_q \Rightarrow \text{Class } C_q) = \max \{c(\mathbf{A}_q \Rightarrow \text{Class } h) \mid h = 1, 2, ..., M\}. \quad (10)$$

We have the same consequent class as in (10) even when we use the support in (9) instead of the confidence in (8). The consequent class C_q is the dominant class among the compatible training patterns with the antecedent part $\mathbf{A}_q$. As we have already mentioned, we do not generate any rule with the antecedent part $\mathbf{A}_q$ when there is no compatible training patterns with $\mathbf{A}_q$.

We specify the rule weight CF_q by the confidence as

$$CF_q = c(\mathbf{A}_q \Rightarrow \text{Class } C_q). \quad (11)$$

The rule weight CF_q is used in the classification phase of new patterns in the following manner. When a new pattern is to be classified by a rule-based classification system, first all compatible rules with the new pattern are found. Then a single winner rule with the largest rule weight is identified among the compatible rules. Finally the new pattern is classified as the consequent class of the winner rule.

Using the above-mentioned rule generation procedure, we can generate a huge number of classification rules by examining the 15^n combinations of the antecedent intervals. For high-dimensional classification problems, it may be impractical to examine all the 15^n combinations. Thus we only examine short rules with a small number of antecedent conditions. It should be noted that the antecedent interval corresponding to $K = 1$ in Fig. 1 is actually equivalent to a *"don't care"* condition. Thus all *don't care* conditions with the antecedent interval for $K = 1$ can be omitted. In this paper, the number of antecedent conditions excluding *don't care* conditions is referred to as the rule length. We only examine short rules of length $L_{\max}$ or less (e.g., $L_{\max} = 3$). This restriction on the rule length is to find simple classification rules with high interpretability.

We further decrease the number of rules by choosing only good rules with respect to a heuristic rule extraction criterion. That is, we choose a prespecified number of short rules for each class using a heuristic criterion. In our computational experiments, we use the following three heuristic criteria:

Support with the minimum confidence level: Each rule is evaluated based on its support value when its confidence value is larger than the prespecified minimum confidence level. This criterion never extracts unqualified rules whose confidence values are smaller than the minimum confidence level. Five minimum confidence levels (0.5, 0.6, 0.7, 0.8, 0.9) are examined in computational experiments.

Product of confidence and support: Each rule is evaluated based on the product of its confidence and support values.

Difference in support: Each rule is evaluated based on the difference between its support value and the total support value of the other rules with the same antecedent condition and different consequent classes. More specifically, the rule R_q with the antecedent condition $\mathbf{A}_q$ and the consequent class C_q is evaluated as follows:

$$f(R_q) = s(\mathbf{A}_q \Rightarrow \text{Class } C_q) - \sum_{\substack{h=1 \\ h \neq C_q}}^{M} s(\mathbf{A}_q \Rightarrow \text{Class } h) . \quad (12)$$

This is a modified version of a heuristic rule evaluation criterion used in an iterative fuzzy genetics-based machine learning algorithm called SLAVE [21].

We choose a prespecified number of promising rules with the largest values of each criterion in a greedy manner for each class. As we have already mentioned, only short rules of length $L_{\max}$ or less are examined in the heuristic rule extraction stage in order to find interpretable rules.

4. Multiobjective Rule Selection

Let us assume that we have N rules extracted from numerical data by heuristic rule extraction in the previous section (i.e., N/M rules for each class). Genetics-based multiobjective rule selection is used to find non-dominated rule sets from these N rules with respect to the accuracy and the complexity (i.e., to find non-dominated subsets of the N rules). The accuracy maximization of a rule set S is performed by minimizing the error rate on training patterns by S. We include the rejection rate into the error rate (i.e., training patterns with no compatible rules in S are counted among errors in this paper). On the other hand, we measure the complexity of the rule set S by the number of rules in S. Thus our rule selection problem is formulated as follows:

$$\text{Minimize } f_1(S) \text{ and } f_2(S), \quad (13)$$

where $f_1(S)$ is the error rate on training patterns by the rule set S and $f_2(S)$ is the number of rules in S.

Any subset S of the N candidate rules can be represented by a binary string of length N as

$$S = s_1 s_2 \cdots s_N , \quad (14)$$

where $s_j = 1$ and $s_j = 0$ mean that the j-th candidate rule is included in S and excluded from S, respectively. Such a binary string is handled as an individual in our EMO approach.

Since feasible solutions (i.e., subsets of the extracted N rules) are represented by binary strings, we can directly apply almost all EMO algorithms to our rule selection problem in (13) using standard crossover and mutation operations. In this paper, we use an elitist non-dominated sorting genetic algorithm (NSGA-II) of Deb et al. [22] because it is a state-of-the-art well-known EMO algorithm with high search ability.

The NSGA-II algorithm randomly generates a prespecified number of binary strings of length N (say N_{pop} strings) as an initial population. Each string is evaluated using Pareto ranking and a crowding measure. N_{pop} new strings are generated by genetic operations (i.e., selection, crossover, and mutation). The generated offspring population is merged with the parent population. The next population is constructed by choosing N_{pop} best strings from the merged population with $2 \times N_{\text{pop}}$ strings using Pareto ranking and a crowding measure as in the selection of parent strings. In this manner, the generation update is iterated until a prespecified stopping condition is satisfied. Non-dominated strings are chosen from the merged population at the final generation. These strings are presented to the human user as non-dominated rule sets. See Deb et al. [22] for details of the NSGA-II algorithm.

In the application of the NSGA-II algorithm to our rule selection problem, we use two problem-specific heuristic tricks in order to efficiently find small rule sets with high accuracy. One trick is biased mutation where a larger probability is assigned to the mutation from 1 to 0 than that from 0 to 1. This is for efficiently decreasing the number of rules in each rule set. The other trick is the removal of unnecessary rules, which is a kind of local search. Since we use the single winner-based method for classifying each pattern by the rule set S, some rules in S may be chosen as winner rules for no training patterns. We can remove these rules without degrading the first objective (i.e., the number of correctly classified training patterns). At the same time, the removal of unnecessary rules leads to the improvement in the other objectives. Thus we remove all rules that are not selected as winner rules for any training patterns from the rule set S. The removal of unnecessary rules is performed after the first objective is calculated and before the second and third objectives are calculated.

5. Computational Experiments

5.1. Settings of Computational Experiments

We use six data sets in Table 1: Wisconsin breast cancer (Breast W), diabetes (Diabetes), glass identification (Glass), Cleveland heart disease (Heart C), sonar (Sonar), and wine recognition (Wine) data sets. These six data sets are available from the UC Irvine machine learning repository (http://www.ics.uci.edu/~mlearn/). Data sets with missing values are marked by "*" in the third column of Table 1. Since we do not use incomplete patterns with missing values, the number of patterns in the third column does not include those patterns with missing values. All attributes are handled as continuous attributes in this paper.

We evaluate the performance of our EMO approach in comparison with the reported results on the same data sets in Elomaa & Rousu [18] where six variants of the C4.5 algorithm were examined. The performance of each variant was evaluated by ten independent executions (with different data partitions) of the whole ten-fold cross-validation (10CV) procedure (i.e., 10×10CV) in [18]. We show in the

last two columns of Table 1 the best and worst error rates on test patterns among the six variants reported in [18] for each data set.

Table 1. Data sets used in our computational experiments.

Data set	Attributes	Patterns	Classes	C4.5 in [18]	
				Best	Worst
Breast W	9	683*	2	5.1	6.0
Diabetes	8	768	2	25.0	27.2
Glass	9	214	6	27.3	32.2
Heart C	13	297*	5	46.3	47.9
Sonar	60	208	2	24.6	35.8
Wine	13	178	3	5.6	8.8

* Incomplete patterns with missing values are not included.

In this section, we examine the accuracy of extracted rules by heuristic rule extraction and non-dominated rule sets obtained by genetics-based multiobjective rule selection for training patterns and test patterns. When the classification accuracy on training patterns is discussed, all the given patterns (excluding incomplete patterns with missing values) are used in heuristic rule extraction and multiobjective rule selection. On the other hand, we use the 10CV procedure (i.e., 90% training patterns and 10% test patterns) when we examine the accuracy on test patterns.

We first explain computational experiments for examining the accuracy on training patterns where all the given patterns are used as training patterns. The accuracy of rules is evaluated on the same training patterns.

As in Fig. 1, we simultaneously use five partitions for each attribute. In the heuristic rule extraction stage, various specifications are used as the number of extracted rules for each class in order to examine the relation between the number of extracted rules and their accuracy. The number of extracted rules is specified as 1, 2, 3, 4, 5, 10, 20, 30, 40, 50, and 100. The three heuristic criteria in Section 3 are used in the heuristic rule extraction stage. When multiple rules have the same value of a heuristic criterion, those rules are randomly ordered (i.e., random tie break). As we have already mentioned, the five specifications of the minimum confidence level (i.e., 0.5, 0.6, 0.7, 0.8, 0.9) are examined in the support criterion with the minimum confidence level.

The maximum rule length $L_{\max}$ is specified as $L_{\max} = 2$ for the sonar data set and $L_{\max} = 3$ for the other data sets. That is, candidate rules of length 2 or less are examined for the sonar data set while those of length 3 or less are examined for the other data sets. We use such a different specification because only the sonar data set involves a large number of attributes (i.e., it has a huge number of possible combinations of antecedent intervals).

For each specification of the heuristic rule extraction criterion, average results are calculated over 20 runs for each data set in order to decrease the possible effect of the random tie break. Then we choose the heuristic rule extraction criterion from which the best average error rate on training patterns is obtained among various criteria in the case of 100 rules for each class. The chosen heuristic rule extraction criterion is used to extract candidate rules for the genetics-based multiobjective rule selection stage. It should be noted that a different criterion is chosen for each data set.

As candidate rules in multiobjective rule selection, we extract 300 rules for each class from training patterns. Thus $300M$ rules are used as candidate rules where M is the number of classes. The NSGA-II algorithm is applied to the extracted $300M$ rules using the following parameter values to find non-dominated rule sets with respect to the two objectives of our rule selection problem:

Population size: 200 strings,
Crossover probability: 0.8 (uniform crossover),
Biased mutation probabilities:
$p_{\mathrm{m}}(0 \to 1) = 1/300M$ and $p_{\mathrm{m}}(1 \to 0) = 0.1$,
Stopping condition: 5000 generations.

The extraction of $300M$ rules and the application of the NSGA-II algorithm are executed 20 times for each data set. Multiple non-dominated rule sets are obtained from each run of the NSGA-II algorithm. We calculate the error rate of each rule set on training patterns. Then the average error rate is calculated over rule sets with the same number of rules among 20 runs. Only when rule sets with the same number of rules are found in all the 20 runs, we report the average error rate for that number of rules in this section.

On the other hand, the 10CV procedure is used for examining the accuracy of rules on test patterns. First the 10CV procedure is iterated three times (i.e., $3 \times 10\text{CV}$) using various criteria in heuristic rule extraction. The average error rates on test patterns are calculated over the three iterations of the 10CV procedure for various specifications of a heuristic rule extraction criterion and the number of extracted rules.

We choose the heuristic rule extraction criterion from which the best average error rate on test patterns is obtained among various criteria in the case of 100 rules for each class. The chosen heuristic rule extraction criterion is used to extract candidate rules for the genetics-based multiobjective rule selection stage as in the computational experiments for examining the accuracy on training patterns.

Using the chosen heuristic rule extraction criterion for each data set, the 10CV procedure is iterated three times (i.e., $3 \times 10\text{CV}$). In each run of $3 \times 10\text{CV}$ for each data set, 300 candidate rules are extracted for each class from training patterns. The NSGA-II algorithm is applied to the $300M$ candidate rules. The error rate on test patterns is calculated for each of the obtained non-dominated rule sets. The average error rate on test patterns is calculated for rule

sets with the same number of rules over 30 runs in 3×10CV . Only when rule sets with the same number of rules are obtained from all the 30 runs, we report the average error rate for that number of rules in this section.

5.2. Results on Training Patterns

In this subsection, we report experimental results on training patterns where average error rates are calculated on training patterns.

Wisconsin Breast Cancer Data: Experimental results by heuristic rule extraction are summarized in Table 2 where the average error rate over 20 runs is shown for each combination of a heuristic rule extraction criterion and the number of extracted rules for each class. The best error rate in each row is indicated by bold face. Since the best result for the case of 100 rules for each class is obtained by the support with the minimum confidence level 0.6 in Table 2 (see the last row), this heuristic rule extraction criterion is used in genetics-based multiobjective rule selection to extract 300 candidate rules for each class.

Table 2. Average error rates on training patterns of extracted rules by heuristic rule extraction (Breast W).

Rules for each class	Support with minimum confidence 0.5	0.6	0.7	0.8	0.9	Product	Diff.
1	8.78	8.78	8.78	8.78	7.91	**7.47**	7.50
2	8.71	8.71	8.71	7.03	**5.71**	6.59	6.59
3	7.03	7.03	7.03	5.56	6.15	**4.98**	6.44
4	5.83	5.71	5.71	5.71	**4.10**	4.39	4.74
5	**4.98**	**4.98**	**4.98**	5.27	5.71	6.00	5.23
10	**5.56**	**5.56**	**5.56**	5.83	6.37	6.73	6.73
20	**6.76**	6.84	6.92	7.04	8.82	8.78	8.78
30	10.40	10.40	10.40	10.40	9.66	9.46	**8.40**
40	**6.49**	6.56	6.52	6.60	7.38	7.61	7.61
50	8.17	8.18	8.13	8.20	8.20	**7.76**	7.78
100	7.22	**7.17**	7.22	7.26	7.47	7.47	7.47

In Fig. 2, we compare the average error rates between heuristic rule extraction and multiobjective rule selection. All the experimental results in Table 2 by heuristic rule extraction are depicted by closed circles whereas the average error rates of selected rules by multiobjective rule selection are shown by open circles. It should be noted that the horizontal axis in Fig. 2 is the total number of rules while the first column of Table 2 shows the number of rules for each class. From Fig. 2, we can see that smaller rule sets with lower error rates are found by multiobjective rule selection than heuristic rule extraction. That is, multiobjective rule selection improves the accuracy-complexity tradeoff curve in Fig. 2. We can observe a clear tradeoff structure between the average error rate and the number of rules from the experimental results by multiobjective rule selection (i.e., open circles in Fig. 2).

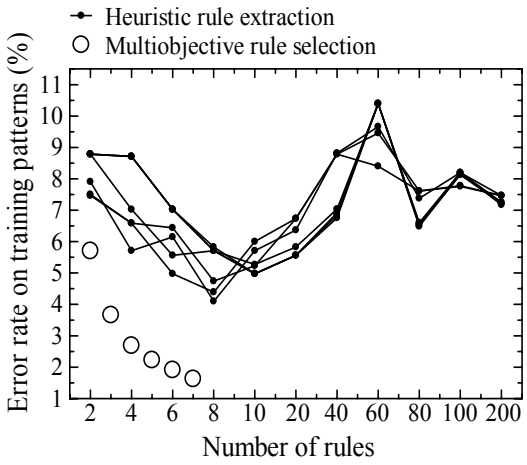

Fig. 2. Comparison between heuristic rule extraction and genetics-based multiobjective rule selection with respect to the average error rates on training patterns (Breast W).

Diabetes Data: Experimental results by heuristic rule extraction are summarized in Table 3. An interesting observation in Table 3 (and also in Table 2) is that the increase in the number of extracted rules does not always lead to the improvement in the average error rates. Another interesting observation from the comparison between Table 2 and Table 3 is that good results are obtained from different heuristic rule extraction criteria (e.g., see the sixth column with the label "0.9" of each table). That is, the choice of an appropriate criterion is problem-dependent.

Table 3. Average error rates on training patterns of extracted rules by heuristic rule extraction (Diabetes).

Rules for each class	Support with minimum confidence 0.5	0.6	0.7	0.8	0.9	Product	Diff.
1	32.68	36.59	30.99	36.33	62.63	**28.39**	29.43
2	31.38	32.16	**23.57**	36.33	59.90	23.96	28.39
3	31.77	**22.79**	23.28	36.33	58.33	23.18	28.39
4	32.68	**22.79**	23.44	36.33	49.87	**22.79**	24.09
5	33.85	23.05	23.44	30.86	49.78	**22.79**	23.44
10	29.69	27.60	**22.27**	30.73	47.33	26.95	22.46
20	31.51	24.74	**22.01**	28.36	38.63	24.48	22.14
30	30.83	24.55	**22.01**	26.43	33.52	22.79	22.87
40	28.65	**21.88**	22.79	24.35	32.29	22.66	22.79
50	27.47	**22.66**	22.79	24.48	30.08	**22.66**	23.31
100	26.95	23.70	23.70	**23.57**	26.56	23.96	23.78

In the same manner as Fig. 2, we compare heuristic rule extraction with multiobjective rule selection in Fig. 3.

Whereas multiobjective rule selection does not always outperform heuristic rule selection when the number of rule is small, it finds good rule sets with 8-20 rules. The relatively poor performance of multiobjective rule selection in the case of small rule sets with 2-6 rules is due to the use of candidate rules extracted by the support criterion with the minimum confidence level 0.8. As shown in Table 3, the performance of this criterion is not good when the number of extracted rules is small. Better results will be obtained from multiobjective rule selection if we use other criteria such as the product of confidence and support to extract candidate rules.

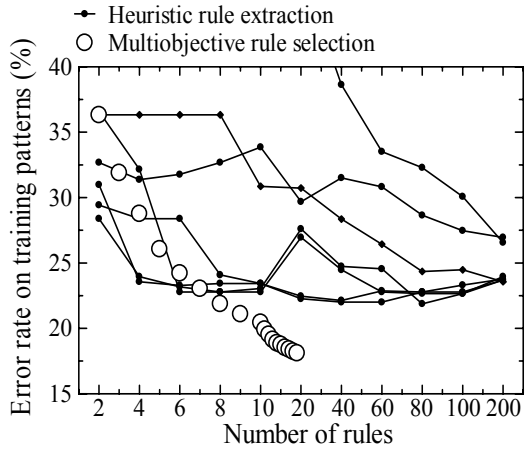

Fig. 3. Comparison between heuristic rule extraction and genetics-based multiobjective rule selection with respect to the average error rates on training patterns (Diabetes).

Glass Identification Data: In the same manner as Fig. 2 and Fig. 3, we compare heuristic rule extraction with multiobjective rule selection in Fig. 4. We can see from Fig. 4 that much better results are obtained from multiobjective rule selection than heuristic rule extraction. That is, much better tradeoffs between the accuracy and the complexity are obtained from multiobjective rule selection.

Cleveland Heart Disease Data: In the same manner as Figs. 2-4, experimental results are summarized in Fig. 5. Multiobjective rule extraction does not always outperform heuristic rule extraction when the number of rules is small. Multiobjective rule selection, however, finds much better rule sets than heuristic rule selection when the number of rules is large (e.g., 15-50 rules). We obtained a similar observation in Fig. 3 for the Diabetes data set.

Sonar Data: Experimental results are summarized in Fig. 6. We can see that much lower error rates are obtained by multiobjective rule selection than heuristic rule extraction for those rule sets with 6-15 rules.

Wine Data: Experimental results are summarized in Fig. 7. All the 20 runs of the NSGA-II algorithm find rule sets with only 5 rules that can correctly classify all the given patterns. On the other hand, 30 rules can not correctly classify all the given patterns in the case of heuristic rule extraction.

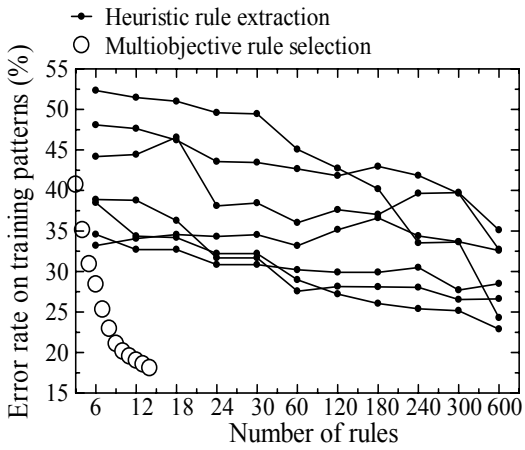

Fig. 4. Comparison between heuristic rule extraction and genetics-bases multiobjective rule selection with respect to the average error rates on training patterns (Glass).

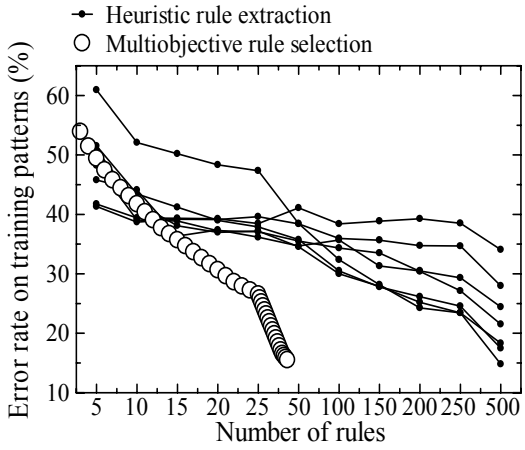

Fig. 5. Comparison between heuristic rule extraction and genetics-bases multiobjective rule selection with respect to the average error rates on training patterns (Heart C).

5.3. Results on Test Patterns

In this subsection, we report experimental results on test patterns where average error rates on test patterns are calculated by three iterations of the 10CV procedure. Heuristic rule extraction and genetics-based multiobjective rule selection are compared with each other. Our experimental results are also compared with the reported results of the C4.5 algorithm in Elomaa and Rousu [18].

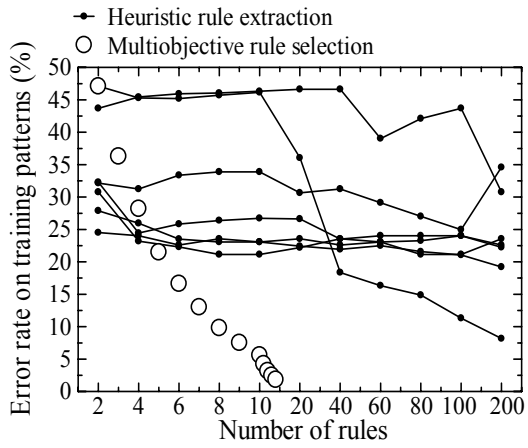

Fig. 6. Comparison between heuristic rule extraction and genetics-bases multiobjective rule selection with respect to the average error rates on training patterns (Sonar).

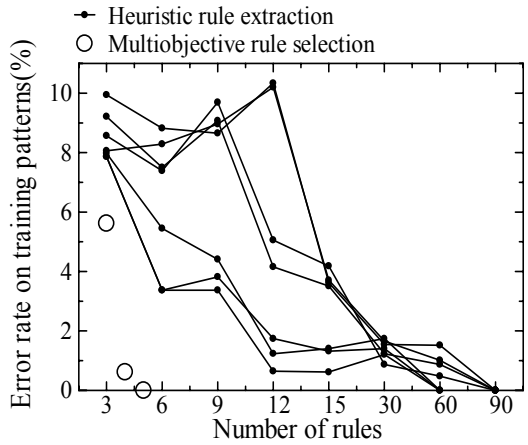

Fig. 7. Comparison between heuristic rule extraction and genetics-bases multiobjective rule selection with respect to the average error rates on training patterns (Wine).

Wisconsin Breast Cancer Data: Experimental results by heuristic rule extraction are summarized in Table 4 where the average error rate on test patterns over three iterations of the 10CV procedure is shown for each combination of a heuristic rule extraction criterion and the number of extracted rules for each class. The best error rate in each row is indicated by bold face. The best result for the case of 100 rules for each class is obtained by the difference criterion in support in Table 3. So we use this heuristic rule extraction criterion in genetics-based multiobjective rule selection to extract 300 candidate rules for each class from training patterns in each run of the 10CV procedure.

In Fig. 8, we compare heuristic rule extraction with multiobjective rule selection by depicting the average error rates on test patterns. Much better results are obtained by multiobjective rule selection. The dotted and dashed lines show the worst and best results of the C4.5 algorithm in Elomaa and Rousu [18], respectively. We can see that multiobjective rule selection outperforms the best result of the C4.5 algorithm with respect to the generalization ability.

Table 4. Average error rates on test patterns of extracted rules by heuristic rule extraction (Breast W).

Rules for each class	Support with minimum confidence 0.5	0.6	0.7	0.8	0.9	Product	Diff.
1	11.19	11.19	11.19	11.19	7.66	**7.42**	7.45
2	8.30	8.52	8.67	8.12	**5.77**	6.53	6.50
3	6.71	6.71	6.71	**5.55**	5.89	5.95	6.23
4	5.31	5.50	5.29	5.53	5.29	**5.17**	5.78
5	**4.97**	5.09	5.02	5.36	5.83	5.74	5.53
10	5.70	**5.69**	5.73	6.05	6.09	6.69	6.61
20	6.50	**6.45**	6.47	6.75	8.64	8.40	8.17
30	9.52	9.51	9.52	9.66	8.91	8.09	**7.54**
40	7.49	7.52	7.48	7.61	7.14	7.18	**7.13**
50	7.56	**7.55**	**7.55**	7.56	7.73	**7.55**	7.56
100	7.53	7.52	7.54	7.54	7.38	**7.09**	**7.09**

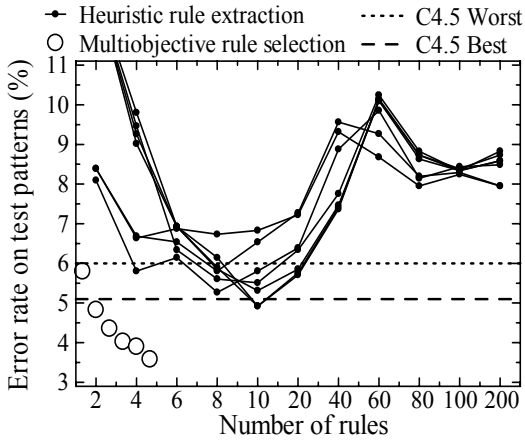

Fig. 8. Experimental results of the 10CV procedure (Breast W).

Diabetes Data: In the same manner as Fig. 8, we compare the average error rates on test patterns between heuristic rule extraction and multiobjective rule selection in Fig. 9. Experimental results show that multiobjective rule selection does not outperform heuristic rule extraction in terms of error rates on test patterns for the diabetes data set.

Glass Identification Data: Experimental results are summarized in Fig. 10. Fig. 10 clearly shows that better results are obtained from multiobjective rule selection than heuristic rule extraction.

Cleveland Heart Disease Data: Experimental results are summarized in Fig. 11 where multiobjective rule selection does not always outperform heuristic rule extraction.

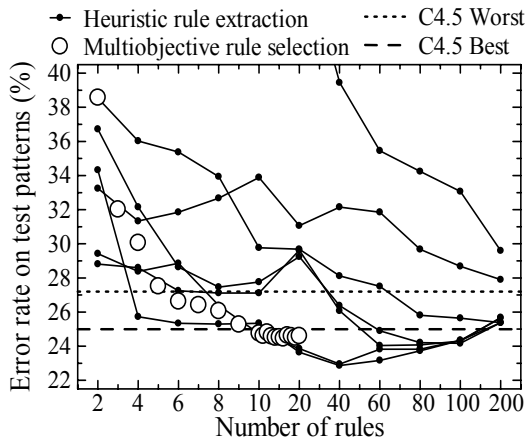

Fig. 9. Experimental results of the 10CV procedure (Diabetes).

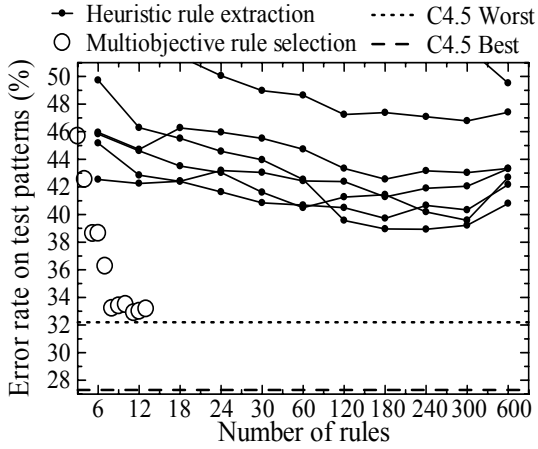

Fig. 10. Experimental results of the 10CV procedure (Glass).

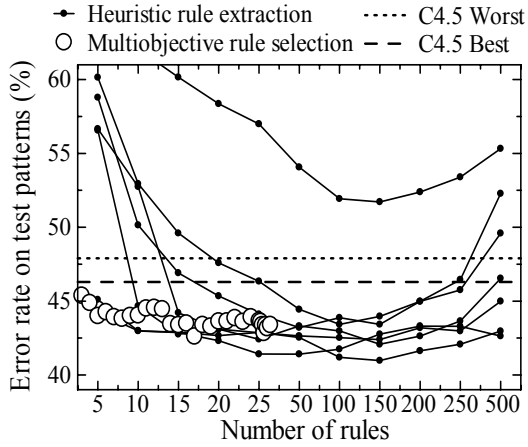

Fig. 11. Experimental results of the 10CV procedure (Heart C).

Sonar Data: Experimental results are summarized in Fig. 12. We can see from Fig. 12 that lower error rates are obtained by multiobjective rule selection than heuristic rule extraction when the number of rules is 9-12.

Wine Data: Experimental results are summarized in Fig. 13. We can see from Fig. 13 that very small rule sets of only 3 or 4 rules obtained by multiobjective rule selection have almost the same generalization ability as much larger rule sets of 30-150 rules obtained by heuristic rule extraction.

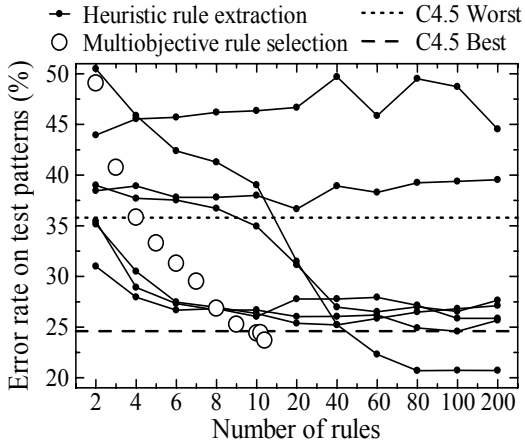

Fig. 12. Experimental results of the 10CV procedure (Sonar).

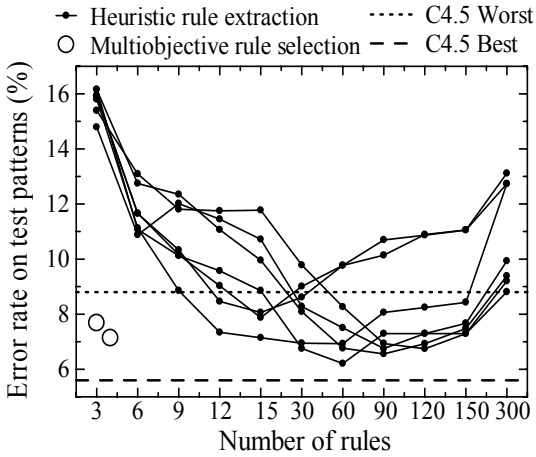

Fig. 13. Experimental results of the 10CV procedure (Wine).

6. Conclusions

We compared heuristic rule extraction with genetics-based multiobjective rule selection through computational experiments on six data sets from the UC Irvine machine learning repository. Experimental results showed that

multiobjective rule selection improved the accuracy-complexity tradeoff curve of heuristically extracted rules by searching for good combinations of a small number of rules. This improvement was observed in all experiments with respect to the accuracy on training patterns and most experiments with respect to the accuracy on test patterns. Except for the glass data set, multiobjective rule selection was comparable to or outperformed the C4.5 algorithm in terms of the generalization ability of obtained rule sets.

Since a large number of rules are usually obtained from data mining, multiobjective rule selection seems to be a promising direction to decrease the complexity of extracted rules. One difficulty of our EMO approach is its large computational load when it is applied to large data sets.

Acknowledgement

This work was partially supported by Japan Society for the Promotion of Science (JSPS) through Grand-in-Aid for Scientific Research (B): KAKENHI (17300075).

References

[1] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Chichester, 2001.

[2] C. A. Coello Coello, D. A. van Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer Academic Publishers, Boston, 2002.

[3] M. A. Kupinski and M. A. Anastasio, "Multiobjective genetic optimization of diagnostic classifiers with implications for generating receiver operating characteristic curve," *IEEE Trans. on Medical Imaging*, vol. 18, no. 8, pp. 675-685, August 1999.

[4] J. Gonzalez, I. Rojas, J. Ortega, H. Pomares, F. J. Fernandez, and A. F. Diaz, "Multiobjective evolutionary optimization of the size, shape, and position parameters of radial basis function networks for function approximation," *IEEE Trans. on Neural Networks*, vol. 14, no. 6, pp. 1478-1495, November 2003.

[5] H. A. Abbass, "Speeding up back-propagation using multiobjective evolutionary algorithms," *Neural Computation*, vol. 15, no. 11, pp. 2705-2726, November 2003.

[6] H. A. Abbass, "Pareto neuro-evolution: Constructing ensemble of neural networks using multi-objective optimization," *Proc. of Congress on Evolutionary Computation*, pp. 2074-2080, Canberra, Australia, December 8-12, 2003.

[7] A. Chandra and X. Yao, "DIVACE: Diverse and accurate ensemble learning algorithm," *Lecture Notes in Computer Science 3177: Intelligent Data Engineering and Automated Learning - IDEAL 2004*, Springer, Berlin, pp 619-625, August 2004.

[8] A. Chandra and X. Yao, "Evolutionary framework for the construction of diverse hybrid ensemble," *Proc. of the 13th European Symposium on Artificial Neural Networks - ESANN 2005*, pp 253-258, Brugge, Belgium, April 27-29, 2005.

[9] H. Ishibuchi and T. Yamamoto, "Evolutionary multiobjective optimization for generating an ensemble of fuzzy rule-based classifiers," *Lecture Notes in Computer Science*, vol. 2723, *Genetic and Evolutionary Computation - GECCO 2003*, pp. 1077-1088, Springer, Berlin, July 2003.

[10] H. Ishibuchi, T. Murata, and I. B. Turksen, "Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems," *Fuzzy Sets and Systems*, vol. 89, no. 2, pp. 135-150, July 1997.

[11] H. Ishibuchi, T. Nakashima, and T. Murata, "Three-objective genetics-based machine learning for linguistic rule extraction," *Information Sciences*, vol. 136, no. 1-4, pp. 109-133, August 2001.

[12] O. Cordon, M. J. del Jesus, F. Herrera, L. Magdalena, and P. Villar, "A multiobjective genetic learning process for joint feature selection and granularity and contexts learning in fuzzy rule-based classification systems," in J. Casillas, O. Cordon, F. Herrera, and L. Magdalena (eds.), *Interpretability Issues in Fuzzy Modeling*, pp. 79-99, Springer, Berlin, 2003.

[13] F. Jimenez, A. F. Gomez-Skarmeta, G. Sanchez, H. Roubos, and R. Babuska, "Accurate, transparent and compact fuzzy models by multi-objective evolutionary algorithms," in J. Casillas, O. Cordon, F. Herrera, and L. Magdalena (eds.), *Interpretability Issues in Fuzzy Modeling*, pp. 431-451, Springer, Berlin, 2003.

[14] H. Ishibuchi and T. Yamamoto, "Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining," *Fuzzy Sets and Systems*, vol. 141, no. 1, pp. 59-88, January 2004.

[15] H. Ishibuchi, T. Nakashima, M. Nii, *Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining*, Springer, Berlin, November 2004.

[16] H. Wang, S. Kwong, Y. Jin, W. Wei, and K. F. Man, "Agent-based evolutionary approach for interpretable rule-based knowledge extraction," *IEEE Trans. on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, vol. 35, no. 2, pp. 143-155, May 2005.

[17] H. Wang, S. Kwong, Y. Jin, W. Wei, and K. F. Man, "Multi-objective hierarchical genetic algorithm for interpretable fuzzy rule-based knowledge extraction," *Fuzzy Sets and Systems*, vol. 149, no. 1, pp. 149-186, January 2005.

[18] T. Elomaa and J. Rousu, "General and efficient multisplitting of numerical attributes," *Machine Learning*, vol. 36, no. 3, pp. 201-244, September 1999.

[19] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[20] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo, "Fast discovery of association rules," in U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (eds.), *Advances in Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, pp. 307-328, 1996.

[21] A. Gonzalez and R. Perez, "SLAVE: A genetic learning system based on an iterative approach," *IEEE Trans. on Fuzzy Systems*, vol. 7, no. 2, pp. 176-191, April 1999.

[22] K. Deb, A. Pratap, S. Agrawal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, April 2002.