H. Ishibuchi and Y. Nojima, “Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning,” *International Journal of Approximate Reasoning*, vol. 44, no. 1, pp. 4-31, January 2007.

Analysis of Interpretability-Accuracy Tradeoff of Fuzzy Systems by Multiobjective Fuzzy Genetics-Based Machine Learning

Hisao Ishibuchi*, Yusuke Nojima

Department of Computer Science and Intelligent Systems, Graduate School of Engineering,

Osaka Prefecture University, 1-1 Gakuen-cho, Sakai, Osaka 599-8531, Japan

* Corresponding Author: Prof. Hisao Ishibuchi

Phone: +81-72-254-9350 Fax: +81-72-254-9915 E-mail: hisaoi@cs.osakafu-u.ac.jp

Address: Department of Computer Science and Intelligent Systems,

Graduate School of Engineering, Osaka Prefecture University,

1-1 Gakuen-cho, Sakai, Osaka 599-8531, Japan

Abstract

This paper examines the interpretability-accuracy tradeoff in fuzzy rule-based classifiers using a multiobjective fuzzy genetics-based machine learning (GBML) algorithm. Our GBML algorithm is a hybrid version of Michigan and Pittsburgh approaches, which is implemented in the framework of evolutionary multiobjective optimization (EMO). Each fuzzy rule is represented by its antecedent fuzzy sets as an integer string of fixed length. Each fuzzy rule-based classifier, which is a set of fuzzy rules, is represented as a concatenated integer string of variable length. Our GBML algorithm simultaneously maximizes the accuracy of rule sets and minimizes their complexity. The accuracy is measured by the number of correctly classified training patterns while the complexity is measured by the number of fuzzy rules and/or the total number of antecedent conditions of fuzzy rules. We examine the interpretability-accuracy tradeoff for training patterns through computational experiments on some benchmark data sets. A clear tradeoff structure is visualized for each data set. We also examine the interpretability-accuracy tradeoff for test patterns. Due to the overfitting to training patterns, a clear tradeoff structure is not always obtained in computational experiments for test patterns.

Keywords: Classification, Fuzzy systems, Fuzzy data mining, Multiobjective optimization, Genetic algorithms, Genetics-based machine learning.

1. Introduction

There are two main goals in the design of fuzzy rule-based systems: One is the accuracy maximization and the other is the complexity minimization. In the 1990s, emphasis was placed on the accuracy maximization. Various approaches have been proposed to improve the accuracy of fuzzy rule-based systems using learning algorithms of neural networks and optimization techniques in evolutionary computation (e.g., see Cordon et al. [1] for a review on various evolutionary optimization techniques for the design of fuzzy rule-based systems). Those approaches usually improve the accuracy of fuzzy rule-based systems at the cost of their interpretability. That is, the complexity of fuzzy rule-based systems usually increases as a result of the accuracy maximization. Some researchers tried to simultaneously perform the accuracy maximization and the complexity minimization in order to design fuzzy rule-based systems with high accuracy and high interpretability [2]-[4]. It is, however, impossible to simultaneously optimize these two objectives. Thus the existence of the accuracy-complexity tradeoff in the design of fuzzy rule-based systems has been realized [5], [6]. Recently, the accuracy maximization and the complexity minimization have been often discussed as multiobjective optimization problems [5]-[9]. The accuracy-complexity tradeoff has been also studied for the design of neural networks [10], [11]. In the field of fuzzy rule-based systems, the accuracy-complexity tradeoff is often referred to as the interpretability-accuracy tradeoff. This is because high interpretability is the main advantage of fuzzy rule-based systems over other nonlinear systems such as neural networks.

One of the first attempts to simultaneously perform the accuracy maximization and the complexity minimization of fuzzy rule-based classifiers was GA-based rule selection of Ishibuchi et al. [12], [13] in the mid-1990s. They used the following fitness function for fuzzy rule selection:

$$fitness(S) = w_1 \cdot f_1(S) - w_2 \cdot f_2(S), \tag{1}$$

where S is a set of fuzzy rules (i.e., S is a fuzzy rule-based classifier), $f_1(S)$ is the number of correctly classified training patterns by S, $f_2(S)$ is the number of fuzzy rules in S, and w_1 and w_2 are prespecified positive constants. A standard single-objective genetic algorithm was

used to maximize the fitness function in (1). The GA-based rule selection method was extended to two-objective rule selection in [14] where a simple multiobjective genetic algorithm was used to find a large number of non-dominated rule sets with respect to the following two objectives:

$$\text{Maximize} \quad f_1(S) \quad \text{and minimize} \quad f_2(S). \tag{2}$$

This formulation was further extended to three-objective rule selection in [15], [16] as follows:

$$\text{Maximize} \quad f_1(S), \text{ minimize} \quad f_2(S), \text{ and minimize} \quad f_3(S), \tag{3}$$

where $f_3(S)$ is the total number of antecedent conditions (excluding "*don't care*" conditions) of fuzzy rules in *S*. Since the number of antecedent conditions of each fuzzy rule is often referred to as the rule length, $f_3(S)$ can be viewed as the total rule length. While only the number of fuzzy rules was considered as a complexity measure in the two-objective formulation in (2), the length of each fuzzy rule was also taken into account in the three-objective formulation in (3).

GA-based rule selection in [12]-[16] consists of two stages: heuristic generation of candidate rules and genetic rule selection. In the handling of low-dimensional problems such as the iris data set with four attributes, all possible fuzzy rules can be used as candidate rules [12]-[14]. Heuristic prescreening of candidate rules, however, is necessary in the candidate rule generation stage [15], [16] when a GA-based rule selection method is to be applied to high-dimensional problems such as the wine data set with 13 attributes and the sonar data set with 60 attributes. A data mining technique was used in [16] to generate a prespecified number of promising candidate rules. Genetic algorithms were used in the rule selection stage where each rule set was coded as a binary string. The string length is the same as the number of candidate rules.

A number of fuzzy genetics-based machine learning (GBML) algorithms have already been proposed in the literature [1] where fuzzy rule-based systems are generated from numerical data through genetic operations. Fuzzy GBML algorithms are usually classified into two categories as in the case of standard non-fuzzy GBML algorithms: Michigan approach and Pittsburgh approach. A single rule is handled as an individual in the Michigan approach while a rule set is

handled as an individual in the Pittsburgh approach. As a result, the evolution of fuzzy rule-based systems is indirectly driven by the fitness evaluation of each fuzzy rule in the Michigan approach while it is directly driven by the fitness evaluation of each rule set in the Pittsburgh approach. There is another category of fuzzy GBML algorithms: Iterative rule learning approach where a single rule is obtained from each execution of a GBML algorithm. A fuzzy rule-based system is constructed by its iterative execution (for details, see [1]).

In our former studies [17], [18], we showed that fuzzy rule-based classifiers can be generated for high-dimensional problems with many attributes by a Michigan-style GBML algorithm even when we use simple grid fuzzy partitions of the pattern space. Those studies demonstrated the necessity of the use of "*don't care*" conditions in the handling of high-dimensional problems. We also combined the Michigan and Pittsburgh approaches into a single hybrid algorithm [19]. In this paper, we extend the hybrid fuzzy GBML algorithm to the case of multiobjective optimization. Through computational experiments using a hybrid multiobjective fuzzy GBML algorithm on six benchmark data sets in the UCI Machine Learning Repository (http://www.ics.uci.edu/~mlearn/), we examine the interpretability-accuracy tradeoff in the design of fuzzy rule-based classifiers.

This paper is organized as follows. First we explain some basic concepts in multiobjective optimization in Section 2. We also explain the NSGA-II algorithm of Deb et al. [20] in Section 2. Next we explain fuzzy rule-based classifiers, three formulations of their multiobjective design, and our hybrid multiobjective fuzzy GBML algorithm in Section 3. Our GBML algorithm is implemented in the framework of the NSGA-II algorithm. Then we examine the interpretability-accuracy tradeoff in fuzzy rule-based classifiers in detail through computational experiments on benchmark data sets in Section 4. We also compare the three formulations of multiobjective fuzzy rule-based classifier design with each other in Section 4. Our GBML algorithm is compared with the multiobjective fuzzy rule selection method in Section 5. Finally we conclude this paper in Section 6.

2. Evolutionary multiobjective optimization

Evolutionary multiobjective optimization (EMO) is one of the most active research areas in

the field of evolutionary computation [21]-[24]. In this section, we briefly explain some basic concepts in multiobjective optimization and one of the most well-known and frequently-used EMO algorithms: NSGA-II of Deb et al. [20].

2.1. Multiobjective optimization problems

Let us consider the following *k*-objective maximization problem:

$$\text{Maximize} \quad \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_k(\mathbf{x})) \quad \text{subject to} \quad \mathbf{x} \in \mathbf{X}, \tag{4}$$

where $\mathbf{f}(\mathbf{x})$ is the objective vector, $f_i(\mathbf{x})$ is the *i*-th objective to be maximized, $\mathbf{x}$ is the decision vector, and $\mathbf{X}$ is the feasible region in the decision space.

When the following two conditions are satisfied, a feasible solution $\mathbf{x} \in \mathbf{X}$ is said to be dominated by another feasible solution $\mathbf{y} \in \mathbf{X}$ (i.e., $\mathbf{y}$ dominates $\mathbf{x}$: $\mathbf{y}$ is better than $\mathbf{x}$):

$$\forall i, \quad f_i(\mathbf{x}) \le f_i(\mathbf{y}) \quad \text{and} \quad \exists j, \quad f_j(\mathbf{x}) < f_j(\mathbf{y}). \tag{5}$$

If there is no feasible solution $\mathbf{y}$ that dominates $\mathbf{x}$, $\mathbf{x}$ is said to be a Pareto-optimal solution of the multiobjective optimization problem in (4). The set of all Pareto-optimal solutions is the Pareto-optimal solution set. The image of the Pareto-optimal solution set onto the objective space is the Pareto front. The Pareto dominance relation in (5) can be also applied to a solution set (i.e., a population). If there is no solution $\mathbf{y}$ in a population that dominates $\mathbf{x}$, $\mathbf{x}$ is said to be a non-dominated solution in that population.

2.2. Evolutionary multiobjective optimization algorithms

The task of evolutionary multiobjective optimization (EMO) algorithms is to find well-distributed Pareto-optimal or near Pareto-optimal solutions as many as possible. The main advantage of EMO algorithms over other multiobjective optimization methods is that many non-dominated solutions can be simultaneously obtained by their single run. A number of EMO algorithms have been proposed in the literature [21]-[24]. The NSGA-II algorithm of Deb et al. [20] is one of the most well-known and frequently-used EMO algorithms in the literature.

As in other evolutionary algorithms, first the NSGA-II algorithm generates an initial population. This is usually performed randomly. Then an offspring population is generated from

the current population by selection, crossover and mutation. The next population is constructed from the current and offspring populations. The generation of an offspring population and the construction of the next population are iterated until a prespecified stopping condition is satisfied. The NSGA-II algorithm has two features, which make it a high-performance EMO algorithm. One is the fitness evaluation of each solution based on Pareto ranking and a crowding measure, and the other is an elitist generation update procedure.

Each solution in the current population is evaluated in the following manner. First, Rank 1 is assigned to all non-dominated solutions in the current population. All solutions with Rank 1 are tentatively removed from the current population. Next, Rank 2 is assigned to all non-dominated solutions in the reduced current population. All solutions with Rank 2 are tentatively removed from the reduced current population. This procedure is iterated until all solutions are tentatively removed from the current population (i.e., until ranks are assigned to all solutions). As a result, a different rank is assigned to each solution. Solutions with smaller ranks are viewed as being better than those with larger ranks. Among solutions with the same rank, an additional criterion called a crowding measure is taken into account. The crowding measure for a solution calculates the distance between its adjacent solutions with the same rank in the objective space (for details, see [20], [21]). Less crowded solutions with larger values of the crowding measure are viewed as being better than more crowded solutions with smaller values of the crowding measure. A pair of parent solutions are selected from the current population by binary tournament selection based on the Pareto ranking and the crowding measure.

When the next population is to be constructed, the current and offspring populations are combined into a merged population. Each solution in the merged population is evaluated in the same manner as in the selection phase of parent solutions using the Pareto ranking and the crowding measure. The next population is constructed by choosing a prespecified number (i.e., population size) of the best solutions from the merged population. Elitism is implemented in the NSGA-II algorithm in this manner.

An outline of the NSGA-II algorithm is written as follows:

[NSGA-II Algorithm]

Step 1: Generate an initial population with N_{pop} solutions where N_{pop} is the population size.

Step 2: Generate an offspring population by iterating the following procedures N_{pop} times:

(1) Choose a pair of parent solutions from the current population using binary tournament selection.

(2) Generate an offspring from the selected parent solutions by crossover and mutation.

Step 3: Combine the current population and the offspring population into a merged one. Then choose the best N_{pop} solutions from the merged population to construct the next population.

Step 4: If a prespecified stopping condition is not satisfied, return to Step 2. Otherwise terminate the execution of the algorithm. In the latter case, we choose all the non-dominated solutions in the merged population in Step 3 as the final solutions.

3. Multiobjective design of fuzzy rule-based classifiers

In this section, we explain fuzzy rule-based classifiers, three formulations of their multiobjective design, and a hybrid multiobjective fuzzy GBML algorithm.

3.1. Fuzzy rule-based classifiers

Let us assume that we have *m* training patterns $\mathbf{x}_p = (x_{p1}, ..., x_{pn})$, $p = 1, 2, ..., m$ from *M* classes where x_{pi} is the attribute value of the *p*-th training pattern for the *i*-th attribute ($i = 1, 2, ..., n$). We also assume that the *n*-dimensional pattern space has already been normalized into the *n*-dimensional unit hyper-cube $[0, 1]^n$. That is, we assume that we have an *M*-class pattern classification problem with *m* training patterns in the *n*-dimensional pattern space $[0, 1]^n$. For this problem, we use fuzzy rules of the following form:

$$\text{Rule } R_q: \text{If } x_1 \text{ is } A_{q1} \text{ and } ... \text{ and } x_n \text{ is } A_{qn} \text{ then Class } C_q \text{ with } CF_q, \tag{6}$$

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 fuzzy set, C_q is a class label, and CF_q is a rule weight.

We define the compatibility grade of each training pattern $\mathbf{x}_p$ with the antecedent part $\mathbf{A}_q = (A_{q1}, ..., A_{qn})$ of the fuzzy rule R_q in (1) using the product operator as

$$\mu_{\mathbf{A}_q}(\mathbf{x}_p) = \mu_{A_{q1}}(x_{p1}) \cdot \mu_{A_{q2}}(x_{p2}) \cdot ... \cdot \mu_{A_{qn}}(x_{pn}), \ p = 1, 2, ..., m, \tag{7}$$

where $\mu_{A_{qi}}(\cdot)$ is the membership function of A_{qi}.

To determine the consequent class C_q and the rule weight CF_q, we first calculate the confidence of the fuzzy rule "$\mathbf{A}_q \Rightarrow$ Class h" for each class h as follows (see the textbook on fuzzy data mining [25] for fuzzy versions of some basic concepts in data mining such as confidence and support):

$$c(\mathbf{A}_q \Rightarrow \text{Class } h) = \sum_{\mathbf{x}_p \in \text{Class } h} \mu_{\mathbf{A}_q}(\mathbf{x}_p) \Big/ \sum_{p=1}^{m} \mu_{\mathbf{A}_q}(\mathbf{x}_p), \quad h = 1, 2, ..., M. \tag{8}$$

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

$$c(\mathbf{A}_q \Rightarrow \text{Class } C_q) = \max\{c(\mathbf{A}_q \Rightarrow \text{Class } h) \mid h = 1, 2, ..., M\}. \tag{9}$$

Rule weights have a significant effect on the classification accuracy of a fuzzy rule-based classifier [26]. Several methods were examined to determine the rule weight of each fuzzy rule in [27] where good results were obtained from the following specification:

$$CF_q = c(\mathbf{A}_q \Rightarrow \text{Class } C_q) - \sum_{\substack{h=1 \\ h \neq C_q}}^{M} c(\mathbf{A}_q \Rightarrow \text{Class } h). \tag{10}$$

We use the above definition in this paper. When the rule weight is not positive, we do not generate the corresponding fuzzy rule.

Let S be a fuzzy rule-based classifier (i.e., a set of fuzzy rules). When an input pattern $\mathbf{x}_p$ is to be classified by the fuzzy rule-based classifier S, a single winner rule R_w is chosen from S as follows:

$$\mu_{\mathbf{A}_w}(\mathbf{x}_p) \cdot CF_w = \max\{\mu_{\mathbf{A}_q}(\mathbf{x}_p) \cdot CF_q \mid R_q \in S\}. \tag{11}$$

The input pattern $\mathbf{x}_p$ is assigned to the consequent class C_w of the winner rule R_w. When multiple rules with different consequent classes have the same maximum value in (11), the classification of the input pattern $\mathbf{x}_p$ is rejected. The classification of $\mathbf{x}_p$ is also rejected when there is no compatible fuzzy rules with positive compatibility grades for $\mathbf{x}_p$. In this case, all fuzzy rules have the same maximum value of zero in the right-hand side in (11). For other types of fuzzy rules and fuzzy reasoning in fuzzy rule-based classifiers, see [28], [29].

3.2. Multiobjective formulations of fuzzy rule-based classifier design

As we have already explained in Section 1, the following three objectives have been considered in fuzzy rule selection [12]-[16]:

$f_1(S)$: The number of correctly classified training patterns by *S*,

$f_2(S)$: The number of fuzzy rules in *S*,

$f_3(S)$: The total number of antecedent conditions of fuzzy rules (i.e., total rule length) in *S*.

It should be noted that "*don't care*" conditions are not counted in the calculation of the third objective.

In this paper, we use the following three formulations of multiobjective optimization problems (MOPs) for the interpretability-accuracy tradeoff analysis:

MOP-1: Maximize $f_1(S)$ and minimize $f_2(S)$,

MOP-2: Maximize $f_1(S)$ and minimize $f_3(S)$,

MOP-3: Maximize $f_1(S)$, minimize $f_2(S)$, and minimize $f_3(S)$.

For comparison, we also use the following three formulations of single-objective optimization problems (SOPs):

SOP-1: Maximize $w_1 \cdot f_1(S) - w_2 \cdot f_2(S)$,

SOP-2: Maximize $w_1 \cdot f_1(S) - w_3 \cdot f_3(S)$,

SOP-3: Maximize $w_1 \cdot f_1(S) - w_2 \cdot f_2(S) - w_3 \cdot f_3(S)$,

where w_1, w_2 and w_3 are prespecified non-negative weights.

The Pareto dominance relation in (5) in Section 2 is modified when it is applied to each of the three multiobjective optimization problems. For example, the Pareto dominance relation is modified for MOP-3 as follows: A rule set $S_{\mathbf{x}}$ is said to be dominated by another rule set $S_{\mathbf{y}}$ (i.e., $S_{\mathbf{y}}$ dominates $S_{\mathbf{x}}$: $S_{\mathbf{y}}$ is better than $S_{\mathbf{x}}$) when all the following inequalities hold:

$$f_1(S_{\mathbf{x}}) \le f_1(S_{\mathbf{y}}), \quad f_2(S_{\mathbf{x}}) \ge f_2(S_{\mathbf{y}}), \quad f_3(S_{\mathbf{x}}) \ge f_3(S_{\mathbf{y}}), \tag{12}$$

and at least one of the following inequalities holds:

$$f_1(S_{\mathbf{x}}) < f_1(S_{\mathbf{y}}),\ \ f_2(S_{\mathbf{x}}) > f_2(S_{\mathbf{y}}),\ \ f_3(S_{\mathbf{x}}) > f_3(S_{\mathbf{y}}). \tag{13}$$

Roughly speaking, when a rule set $S_{\mathbf{x}}$ has lower classification accuracy and higher complexity than another rule set $S_{\mathbf{y}}$, $S_{\mathbf{x}}$ is said to be dominated by $S_{\mathbf{y}}$ in all the three multiobjective optimization problems.

3.3. Hybrid multiobjective fuzzy GBML algorithm

We use a hybrid multiobjective fuzzy GBML algorithm to efficiently find a large number of non-dominated rule sets (i.e., fuzzy rule-based classifiers) of the three multiobjective optimization problems: MOP-1, MOP-2, and MOP-3. Our GBML algorithm can be viewed as a Pittsburgh-style algorithm except that a Michigan-style algorithm is applied to each rule set as a kind of mutation. Our GBML algorithm is implemented in the framework of the NSGA-II algorithm as follows:

[Hybrid Multiobjective Fuzzy GBML Algorithm]

Step 1: Generate an initial population of N_{pop} rule sets where N_{pop} is the population size.

Step 2: Generate an offspring population by iterating the following procedures N_{pop} times:

(1) Select a pair of parent rule sets from the current population using binary tournament selection.

(2) Generate an offspring from the selected pair of parent rule sets by crossover and mutation.

(3) Apply a single iteration of a Michigan-style GBML algorithm to the offspring rule set with a prespecified probability (0.5 in our computational experiments).

Step 3: Combine the current population and the offspring population into a merged one. Then choose the best N_{pop} rule sets from the merged population to construct the next population.

Step 4: If a prespecified stopping condition is not satisfied, return to Step 2. Otherwise terminate the execution of the algorithm. In the latter case, we choose all the non-dominated rule sets in the merged population in Step 3 as the final solutions.

Each rule set is evaluated based on the Pareto ranking and the crowding measure as

explained in Section 2 in the same manner as the NSGA-II algorithm. Our GBML algorithm has high search ability of Michigan approach to efficiently find good fuzzy rules as well as direct optimization ability of Pittsburgh approach to optimize rule sets. In the following, we explain our GBML algorithm in detail.

Each fuzzy rule is represented by its antecedent part $\mathbf{A}_q$ as an integer string of length n where n is the dimensionality of the pattern space (i.e., n is the number of attributes). The consequent class and the rule weight of each fuzzy rule are specified from training patterns as described in Subsection 3.1. For each attribute, we use 14 antecedent fuzzy sets in Fig. 1. We also use "*don't care*" as an additional antecedent fuzzy set. The membership value of this special antecedent fuzzy set is always unity for any input values ($\forall x$, $\mu_{don't\ care}(x) = 1$). We use 15 symbols (e.g., 0, 1, ..., 9, a, b, c, d, e) to represent *don't care* and the 14 antecedent fuzzy sets in Fig. 1. For example, an integer string "0102d0" denotes the fuzzy rule "If x_2 is S^2 and x_4 is L^2 and x_5 is ML^5 then Class C_q with CF_q" where *don't care* conditions on x_1, x_3 and x_6 represented by 0s in the string are omitted. It should be noted that the number of antecedent conditions (i.e., rule length) of this rule is three because *don't care* conditions are not counted. The total number of possible combinations of the antecedent part $\mathbf{A}_q$ is 15^n for our n-dimensional pattern classification problem.

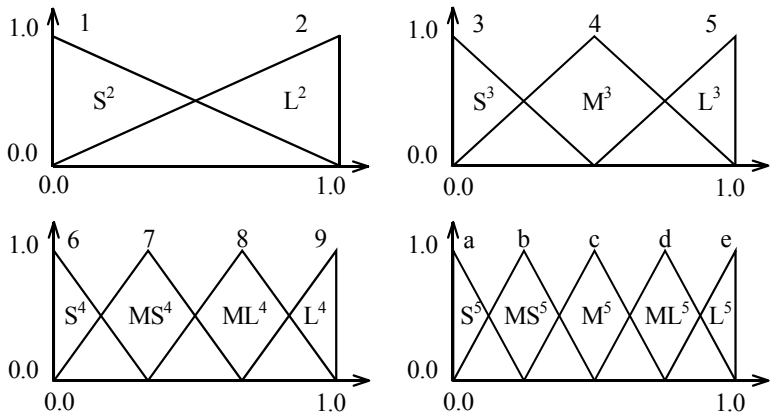


Fig. 1. Four fuzzy partitions used in our computational experiments. The superscript of each fuzzy set means the granularity of the fuzzy partition. Each of the 14 fuzzy sets is represented by one of the 14 symbols (i.e., 1, 2, ..., 9, a, b, c, d, e) as shown in this figure.

A rule set S is handled as an individual and coded as a concatenated integer string where each substring of length n represents a single fuzzy rule. It should be noted that the number of fuzzy rules in each rule set is not fixed in our hybrid multiobjective fuzzy GBML algorithm.

It was shown in [16], [17] that the search ability of Michigan-style fuzzy GBML algorithms was drastically improved by directly generating initial fuzzy rules from training patterns in a heuristic manner. We use a similar heuristic method to generate an initial population of rule sets in Step 1 of our GBML algorithm. First we randomly select a prespecified number of training patterns (say, N_{rule} training patterns). Next we generate a fuzzy rule R_q from each training pattern $\mathbf{x}_p = (x_{p1}, ..., x_{pn})$ by probabilistically choosing an antecedent fuzzy set A_{qi} for each attribute value x_{pi} from the 14 candidate fuzzy sets B_k ($k = 1, 2, ..., 9, \text{a}, \text{b}, \text{c}, \text{d}, \text{e}$) in Fig. 1. Each candidate fuzzy set B_k has the following selection probability for the attribute value x_{pi}:

$$P(B_k) = \frac{\mu_{B_k}(x_{pi})}{\sum_{j=1}^{\text{e}} \mu_{B_j}(x_{pi})}, \quad k = 1, 2, ..., 9, \text{a}, \text{b}, \text{c}, \text{d}, \text{e}. \tag{14}$$

That is, the antecedent part $\mathbf{A}_q = (A_{q1}, ..., A_{qn})$ is specified from $\mathbf{x}_p = (x_{p1}, ..., x_{pn})$ so that A_{qi} has a large compatibility grade with x_{pi}. Then each antecedent fuzzy set of the generated fuzzy rule is replaced with *don't care* using a prespecified probability $P_{don't\,care}$. In this manner, N_{rule} initial fuzzy rules are generated. An initial rule set consists of these fuzzy rules. By iterating this procedure, we generate N_{pop} initial rule sets (i.e., an initial population).

In Step 2 (1) of our GBML algorithm, a pair of parent rule sets are selected from the current population by binary tournament selection based on the Pareto ranking and the crowding measure as in the NSGA-II algorithm. Let the selected rule sets be S_1 and S_2. Some fuzzy rules are randomly selected from each parent to construct a new rule set by crossover in Step 2 (2). The number of fuzzy rules to be inherited from each parent to the new rule set is randomly specified. Let N_1 and N_2 be the number of fuzzy rules to be inherited from S_1 and S_2, respectively. We randomly specify N_1 and N_2 in the intervals $[1, |S_1|]$ and $[1, |S_2|]$, respectively, where $|S_i|$ is the number of fuzzy rules in the rule set S_i. In order to generate a new fuzzy rule, N_1 and N_2 fuzzy rules are randomly chosen from S_1 and S_2, respectively. The generated new rule set has $(N_1 + N_2)$ fuzzy rules. This crossover operation is applied to

the selected pair of parent rule sets using a prespecified crossover probability P_C. When the crossover operation is not applied, one of the two parent rule sets is viewed as an offspring. Each antecedent fuzzy set of the newly generated offspring rule set is randomly replaced with a different antecedent fuzzy set using a prespecified mutation probability P_M.

The point of our crossover operation is that the string length is not fixed. In our computational experiments, we use an upper limit on the number of fuzzy rules in each rule set to find compact rule sets with high interpretability. The upper limit is specified as 40 in this paper. When the number of fuzzy rules is more than 40 in the generated rule set (i.e., $N_1 + N_2 > 40$), we randomly select 40 fuzzy rules and remove the other rules from the rule set.

After the crossover and mutation operations in Step 2 (2), a single iteration of the following Michigan-style algorithm is applied to the newly generated offspring rule set in Step 2 (3):

[Single Iteration of Michigan-Style Fuzzy GBML Algorithm]

Step 1: An offspring rule set *S* is given by the main part of our GBML algorithm.

Step 2: Classify each training pattern by the rule set *S*. The fitness value of each rule is the number of correctly classified training patterns by that rule.

Step 3: Generate N_{replace} fuzzy rules from the existing rules in *S* by genetic operations and from misclassified and/or rejected training patterns.

Step 4: Replace the worst N_{replace} fuzzy rules in *S* with the newly generated N_{replace} rules.

Step 5: Return the updated rule set *S* to the main part of our GBML algorithm.

The fitness of each fuzzy rule R_q (i.e., $fitness(R_q)$) is the number of correctly classified training patterns by R_q in Step 2 of our Michigan-style algorithm. Since we use the single winner-based fuzzy reasoning method (i.e., since a single fuzzy rule is responsible for the classification of each training pattern), the following relation always holds:

$$f_1(S) = \sum_{R_q \in S} fitness(R_q). \tag{15}$$

In Step 3 of our Michigan-style algorithm, N_{replace} fuzzy rules are to be newly generated. We generate at least a half of new fuzzy rules (i.e., at least $N_{\text{replace}}/2$ rules) by genetic operations from the existing rules in *S*. The probabilistic specification of each antecedent fuzzy set by (14) and the replacement with *don't care* using the probability $P_{don't\,care}$ are also used to

generate new fuzzy rules. Let N_{MR} be the sum of the number of misclassified and rejected training patterns by the rule set *S*. When N_{MR} is less than or equal to $N_{\mathrm{replace}}/2$, all the N_{MR} training patterns are used to generate new fuzzy rules. In this case, N_{MR} fuzzy rules are generated from the N_{MR} training patterns. Other fuzzy rules (i.e., $(N_{\mathrm{replace}} - N_{\mathrm{MR}})$ rules) are generated by genetic operations. On the other hand, when N_{MR} is larger than $N_{\mathrm{replace}}/2$, $N_{\mathrm{replace}}/2$ training patterns are randomly chosen from the N_{MR} training patterns. Then $N_{\mathrm{replace}}/2$ fuzzy rules are directly generated from the chosen patterns. Other fuzzy rules are generated by genetic operations.

The number of replaced fuzzy rules (i.e., N_{replace}) is specified as $\lceil 0.2 \times |S| \rceil$ for each rule set *S* where $\lceil 0.2 \times |S| \rceil$ is the minimum integer not smaller than $0.2 \times |S|$. For example, one fuzzy rule is replaced when the number of fuzzy rules in *S* is less than or equal to five. In this case, the heuristic rule generation procedure and the genetic operation-based procedure are randomly evoked with the same probability when at least one training pattern is misclassified or rejected by the rule set *S*.

When we generate a new fuzzy rule by genetic operations, first a pair of parent fuzzy rules are selected from the rule set *S* using binary tournament selection. Then the standard uniform crossover operation is applied to the selected pair to generate a new fuzzy rule. Finally the same mutation operation as in the main part of our GBML algorithm is applied to each antecedent fuzzy set of the newly generated fuzzy rule. This procedure is iterated to generate a required number of new fuzzy rules.

A new rule set is generated in our GBML algorithm by selection, mutation, crossover and a single iteration of the Michigan-style algorithm. When a new rule set includes meaningless fuzzy rules with non-positive rule weights, those fuzzy rules are removed from the rule set. This procedure is iterated N_{pop} times to generate an offspring population of N_{pop} rule sets. The next population is constructed from the merged population of the current and offspring populations in the same manner as the NSGA-II algorithm. When a prespecified stopping condition is satisfied, our GBML algorithm returns all the non-dominated rule sets in the merged population.

When our GBML algorithm is applied to the single-objective optimization problems (i.e., SOP-1, SOP-2, and SOP-3), each rule set is evaluated by the scalar fitness function (i.e.,

weighted sum of multiple objectives) instead of the Pareto ranking and the crowding measure in the main part of our GBML algorithm. Except for this change in the fitness evaluation of each rule set, our GBML algorithm can be applied to each single-objective optimization problem with no modifications.

4. Computational experiments

In this section, we examine the interpretability-accuracy tradeoff of fuzzy rule-based classifiers through computational experiments on six benchmark data sets.

4.1. Conditions of computational experiments

We use six data sets with many numerical attributes: Wisconsin breast cancer, Diabetes, Glass, Cleveland heart disease, Sonar, and Wine, which are available from the UCI Machine Learning Repository (http://www.ics.uci.edu/~mlearn/). Table 1 shows the number of attributes, the number of patterns, and the number of classes in each data set. Some data sets include incomplete patterns with missing values. Those patterns are not used in our computational experiments. This is because the performance of classification methods usually depends on the choice of a handling method of missing values.

Table 1. Data sets used in our computer simulations.

Data set	Number of attributes (n)	Number of patterns (m)	Number of classes (M)	C4.5 in Elomaa & Rousu [30]	
				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 the last two columns of Table 1, we show benchmark results on these data sets. They are error rates reported in Elomaa and Rousu [30] where six variants of the C4.5 algorithm [31],

[32] were examined. The six variants were different from each other in their discretization methods of continuous attributes. The performance of each variant was evaluated by ten independent iterations (with different data partitions) of the whole ten-fold cross-validation (10-CV) procedure (i.e., 10×10-CV) in [30]. We use the same performance evaluation procedure (i.e., ten independent iterations of the whole 10-CV procedure) in our computational experiments when the interpretability-accuracy analysis is performed for test patterns.

We use the following parameter specifications in our GBML algorithm:

Number of fuzzy rules in each initial rule set: 20 rules,
Probability of *don't care* ($P_{don't\ care}$): 0.95 (Sonar data set) and 0.8 (the other data sets),
Population size: 200 rule sets,
Crossover probability in the main part: 0.9,
Crossover probability in the Michigan-style part: 0.9,
Mutation probability in the main part: $1/n$,
Mutation probability in the Michigan-style part: $1/n$,
Stopping condition: 5000 generations.

These parameter specifications mean that the multiobjective evolution of rule sets is performed for 5000 generations from an initial population of 200 rule sets with 20 rules. During a single run of our GBML algorithm, 1,000,000 (i.e., 200×5000) rule sets are examined to find non-dominated rule sets. The relation between the number of generations and the performance of our GBML algorithm is discussed through computational experiments later.

When our GBML algorithm is applied to the single-objective optimization problems (i.e., SOP-1, SOP-2 and SOP-3), we specify the weight values as $w_1 = 10$ and $w_2 = w_3 = 1$. The dependency of the performance of our GBML algorithm on the choice of weight values is also discussed through computational experiments later.

4.2. Illustrative computational experiments

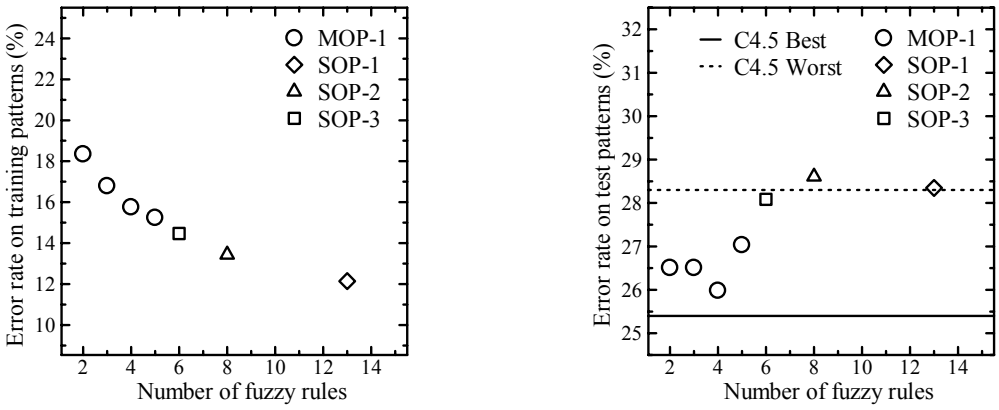
Before performing the interpretability-accuracy analysis of fuzzy rule-based classifiers in detail, we demonstrate how our GBML algorithm works for the single-objective and multiobjective optimization problems through illustrative computational experiments on the

diabetes data set. For illustration purpose, we randomly divided the 768 patterns of this data set into 384 training patterns and 384 test patterns. This setting of computational experiments is to easily show the error rates of each non-dominated fuzzy rule-based classifier on training patterns and test patterns. In the next subsection, we show experimental results based on the 10-fold cross-validation (10-CV) technique.

Our GBML algorithm was applied to each of the six formulations (SOP-1 ~ MOP-3) of fuzzy rule-based classifier design using the 384 training patterns of the diabetes data set. After 5000 generations in each trial, multiple non-dominated rule sets were obtained from each of the three multiobjective formulations. This is because our GBML algorithm was executed for each multiobjective formulation in the framework of evolutionary multiobjective optimization where a number of non-dominated solutions (i.e., non-dominated rule sets) were obtained as a result of optimization. On the other hand, a single rule set was obtained from each of the three single-objective formulations. This is because our GBML algorithm was executed for each single-objective formulation in the framework of evolutionary single-objective optimization where a single optimal solution (i.e., optimal rule set) with respect to the corresponding weighted sum scalar fitness function was obtained as a result of optimization.

The obtained non-dominated rule sets from MOP-1 with $f_1(S)$ and $f_2(S)$ are shown in Fig. 2 (a) together with the three rule sets obtained from the three single-objective formulations. The vertical axis of Fig. 2 (a) is the error rate on the training patterns. In Fig. 2 (a), the rejection of classification is counted as an error. Strictly speaking, the vertical axis is the sum of the error rate and the rejection rate. It should be noted that multiple non-dominated rule sets denoted by open circles in Fig. 2 (a) were obtained by a single run of our GBML algorithm. This result clearly demonstrates that our multiobjective GBML algorithm can find a number of non-dominated rule sets by its single run. One of the obtained non-dominated rule sets from MOP-1 includes only a single fuzzy rule (i.e., $f_2(S)=1$), which is not shown in Fig. 2 (a) because its error rate is out of the range of the vertical axis (i.e., 31.78% on the training patterns).

Fig. 2 (b) shows the error rates on the test patterns of the obtained rule sets in Fig. 2 (a). For comparison, the reported results in [30] by the C4.5 algorithm are also shown in Fig. 2 (b).



(a) Error rates on training patterns.

(b) Error rates on test patterns.

Fig. 2. Experimental results of a single run of our GBML algorithm on the diabetes data set. Our GBML algorithm was applied to MOP-1 with 50% training patterns and 50% test patterns. For comparison, experimental results from the three single-objective optimization problems are also shown.

We can observe a clear interpretability-accuracy tradeoff structure between the number of fuzzy rules and the error rate on the training patterns in Fig. 2 (a). That is, small rule sets with a few fuzzy rules are interpretable but not accurate. Fig. 3 shows the obtained rule set with two fuzzy rules, which has a 18.35% error rate on the training patterns in Fig. 2 (a). The rule set in Fig. 3 is interpretable but does not have high accuracy on the training patterns. On the other hand, larger rule sets with more fuzzy rules in Fig. 2 (a) are more accurate but not interpretable.

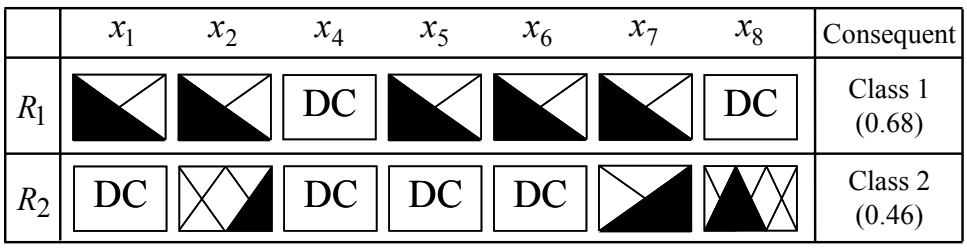
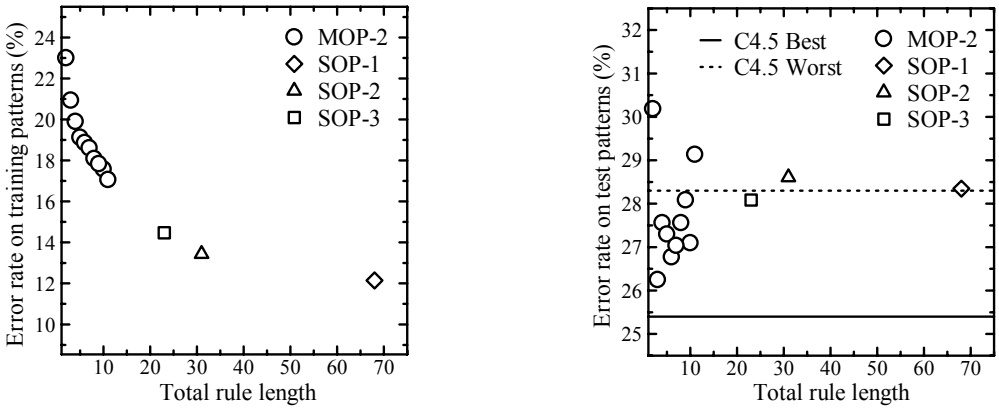


Fig. 3. An obtained non-dominated rule set with two fuzzy rules of MOP-1.

The interpretability-accuracy tradeoff structure of fuzzy rule-based classifiers is not so clear in Fig. 2 (b) with respect to their error rates on the test patterns. For example, the two fuzzy rules in Fig. 3 have high generalization ability (i.e., a 26.51% error rate on the test patterns) in Fig. 2 (b) while they have the worst error rate on the training patterns in Fig. 2 (a). The generalization ability (i.e., error rates on the test patterns) is somewhat degraded by the increase in the number of fuzzy rules due to the overfitting to the training patterns in Fig. 2 (b).

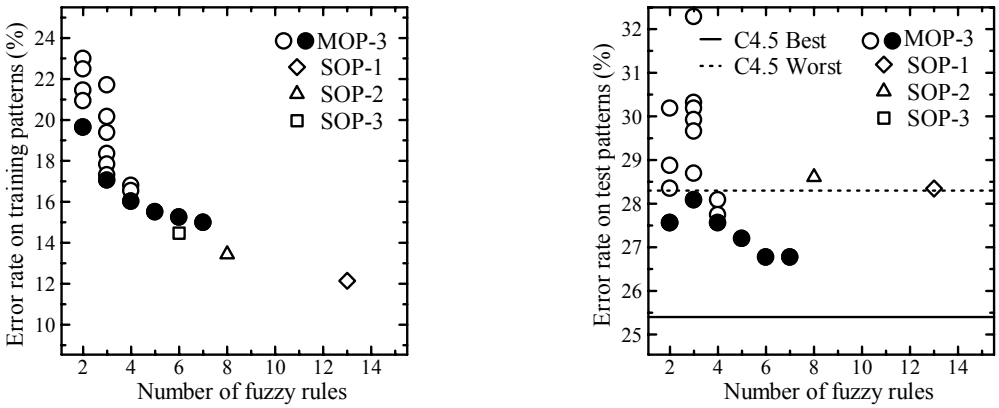
Fig. 4 shows experimental results by MOP-2 with $f_1(S)$ and $f_3(S)$. The horizontal axis is the total rule length (i.e., the total number of antecedent conditions of fuzzy rules) in Fig. 4 while it was the number of fuzzy rules in Fig. 2. We can observe a clear interpretability-accuracy tradeoff structure in Fig. 4 (a) with respect to error rates on the training patterns. On the other hand, such a tradeoff structure is not clear in Fig. 4 (b) with respect to error rates on the test patterns. From the comparison between Fig. 2 and Fig. 4, we can see that more non-dominated rule sets were obtained from MOP-2 than MOP-1.

Fig. 5 shows experimental results by MOP-3 with $f_1(S)$, $f_2(S)$, and $f_3(S)$. The horizontal axis in Fig. 5 is the number of fuzzy rules as in Fig. 2. Since we use both $f_2(S)$ and $f_3(S)$ as complexity measures in MOP-3, some non-dominated rule sets have the same number of fuzzy rules. Those rule sets are different from each other in the total rule length. For example, the total rule length is 1 in the simplest non-dominated rule set with two fuzzy rules (with a 26.87% error rate on the training patterns, which is out of the range of Fig. 5 (a)) while it is 6 in the most complicated non-dominated rule set with two fuzzy rules (with a 19.64% error rate on the training patterns in Fig. 5 (a)). The best rule set among those with the same number of fuzzy rules is depicted by a closed circle in Fig. 5 (a). The same rule set is also depicted by a closed circle in Fig. 5 (b). Whereas the lowest error rates on the test patterns are also obtained by the closed circles in Fig. 5 (b), this is not always the case as we will show later (see Fig. 18). That is, high accuracy on the training patterns among rule sets with the same number of fuzzy rules does not always mean high accuracy on the test patterns.



(a) Error rates on training patterns. (b) Error rates on test patterns.

Fig. 4. Experimental results from MOP-2, SOP-1, SOP-2, and SOP-3.



(a) Error rates on training patterns. (b) Error rates on test patterns.

Fig. 5. Experimental results from MOP-3, SOP-1, SOP-2, and SOP-3.

Experimental results in Fig. 2, Fig. 4, and Fig. 5 also show that rule sets from the single-objective formulations do not always have high accuracy on the test patterns in the right plot of each figure while their accuracy is high on the training patterns in the left plot. Since only a single rule set can be obtained from a single run for a single-objective formulation, multiple runs with different weight specifications are required for the interpretability-accuracy tradeoff analysis of fuzzy rule-based classifiers. On the other hand, we can perform the interpretability-accuracy tradeoff analysis using experimental results from a single run for a

multiobjective formulation. This is the main advantage of multiobjective formulations of fuzzy rule-based classifier design over single-objective formulations.

4.3. Interpretability-accuracy analysis for each data set

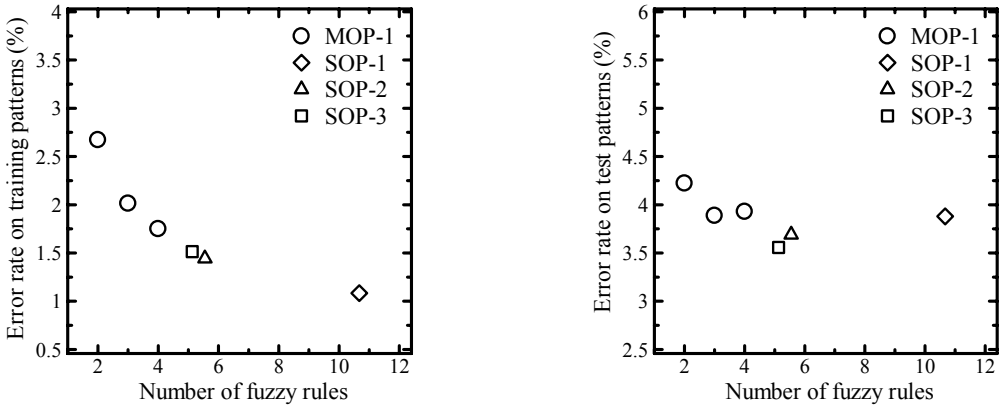
Since the number of fuzzy rules is an intuitively acceptable criterion to measure the interpretability of fuzzy rule-based classifiers, we performed the interpretability-accuracy analysis using MOP-1 with $f_1(S)$ and $f_2(S)$. That is, we applied our GBML algorithm to MOP-1. Our GBML algorithm was also applied to the three single-objective formulations for comparison. We used the 10-fold cross-validation (10-CV) method to evaluate the generalization ability of fuzzy rule-based classifiers for test patterns. Each data set was randomly divided into 10 subsets of the same size in the 10-CV method. Nine subsets were used as training patterns while one subset was used as test patterns. This training-test procedure was iterated 10 times so that each subset was used as test patterns once. We iterated the whole 10-CV procedure 10 times (i.e., 10×10-CV). That is, our GBML algorithm was executed 100 times for each data set.

When our GBML algorithm was applied to one of the three single-objective formulations, a single rule set was obtained from a single run. Thus the average result was simply calculated over 100 rule sets from 100 runs for each data set. On the other hand, multiple rule sets were obtained from a single run when our GBML algorithm was applied to MOP-1 as shown in Fig. 2. Different rule sets were usually obtained from different runs because different training patterns were used in each run of the 10-CV procedure and because our GBML algorithm is a stochastic search algorithm. The number of obtained rule sets from each run was not always the same over 100 runs, either. Thus the calculation of average results is not straightforward in the case of multiobjective design of fuzzy rule-based classifiers. We calculated the average error rate over rule sets with the same number of fuzzy rules when MOP-1 was used. For example, 100 rule sets with two fuzzy rules were obtained from 100 runs of our GBML algorithm on MOP-1 for the diabetes data set. The average error rate was calculated over those 100 rule sets. Other average error rates were also calculated for other values of the number of fuzzy rules. When rule sets with a particular number of fuzzy rules were obtained only from 50 or less runs, the average error rate over those rule sets is not reported in this paper since such an average result is not

reliable. That is, reported average error rates in this paper were always calculated over more than 50 runs.

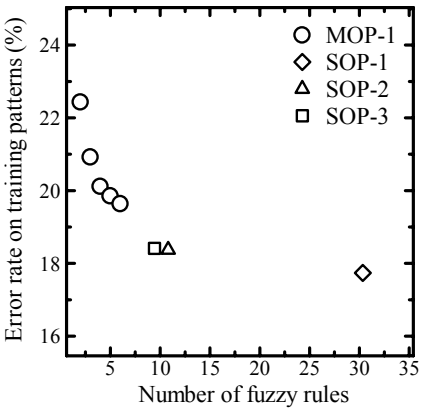
Over ten independent runs of the whole 10-CV procedure, we calculated average error rates on training patterns as well as on test patterns. Experimental results were summarized in Figs. 6-11. Left plots are results on training patterns while right plots are results on test patterns. From these figures, we can see that the six data sets have similar interpretability-accuracy tradeoff structures for training patterns in Figs. 6-11 (a). The six data sets, however, have totally different tradeoff structures for test patterns in Figs. 6-11 (b).

In Figs. 6-11 (a), rule sets with high accuracy on training patterns were obtained from the single-objective formulations for all the six data sets. Good rule sets in terms of generalization ability, however, were not always obtained from the single-objective formulations in Figs. 6-11 (b). This observation suggests an advantage of multiobjective formulations of fuzzy rule-based classifier design over single-objective formulations for some data sets.

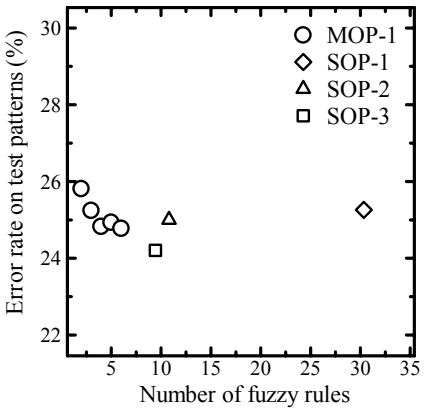


(a) Error rates on training patterns. (b) Error rates on test patterns.

Fig. 6. Experimental results on the Wisconsin breast cancer data set.

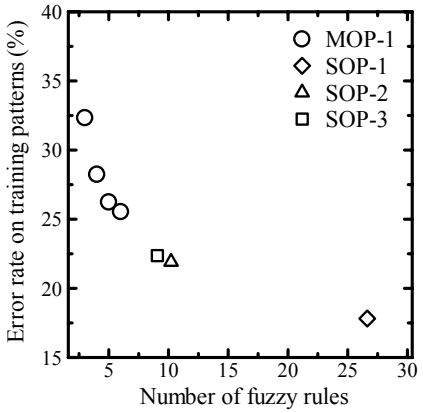


(a) Error rates on training patterns.

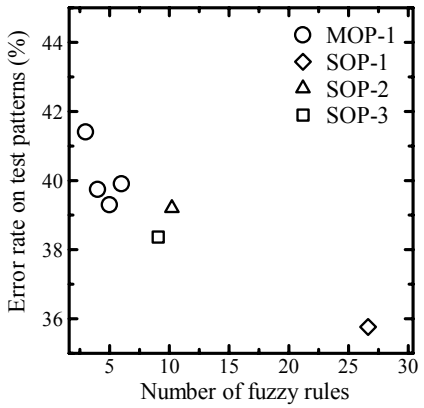


(b) Error rates on test patterns.

Fig. 7. Experimental results on the diabetes data set.

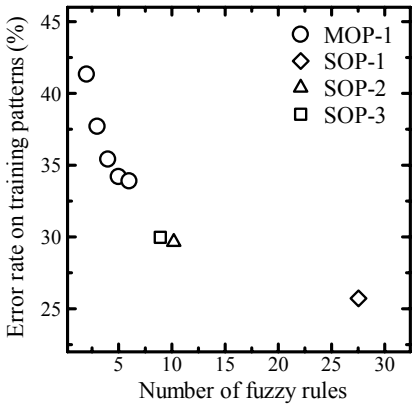


(a) Error rates on training patterns.

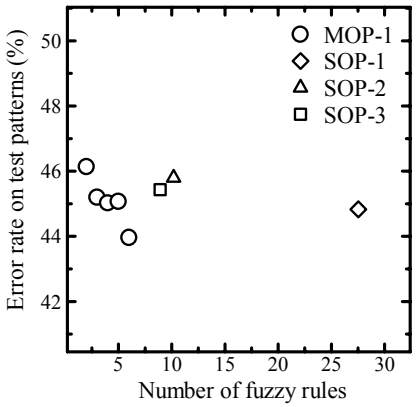


(b) Error rates on test patterns.

Fig. 8. Experimental results on the glass data set.

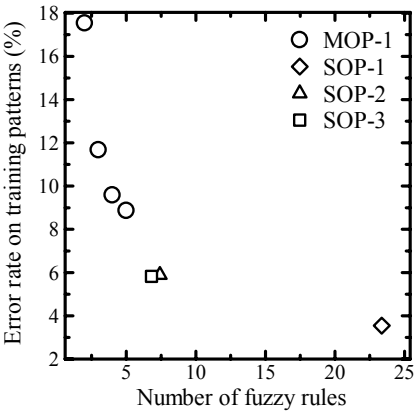


(a) Error rates on training patterns.

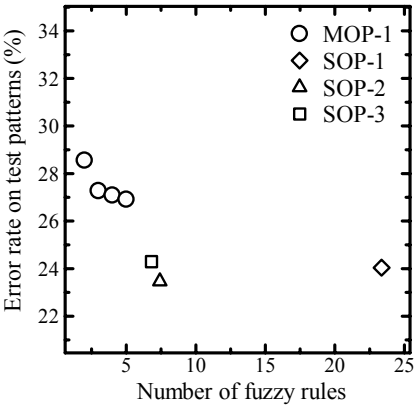


(b) Error rates on test patterns.

Fig. 9. Experimental results on the Cleveland heart disease data set.

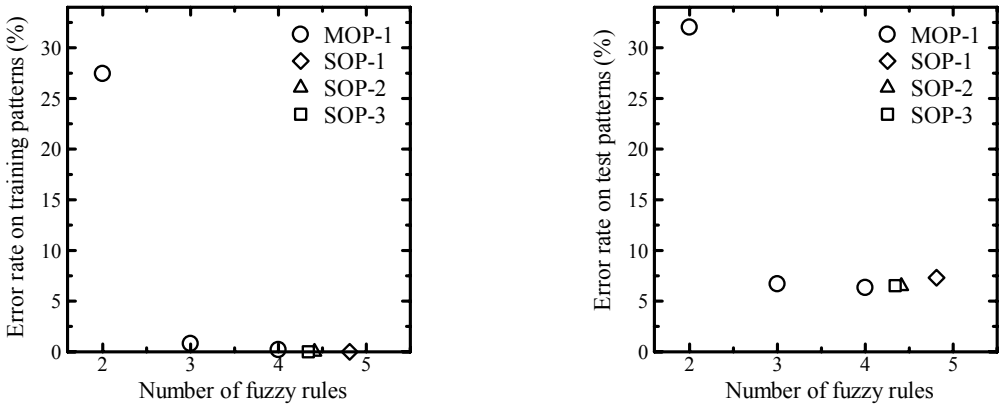


(a) Error rates on training patterns.



(b) Error rates on test patterns.

Fig. 10. Experimental results on the sonar data set.



(a) Error rates on training patterns. (b) Error rates on test patterns.

Fig. 11. Experimental results on the wine data set.

4.4. Comparison among three multiobjective formulations

In order to compare the three multiobjective formulations (i.e., MOP-1, MOP-2 and MOP-3) with each other, we also executed the whole 10-CV procedure 10 times for each data set using MOP-2 and MOP-3. In Table 2, we summarized the average number of obtained non-dominated rule sets from each formulation where bold face shows the largest average number for each data set. From this table, we can see that more non-dominated rule sets were obtained from MOP-2 and MOP-3 than MOP-1. That is, the use of the total rule length as a complexity measure increased the number of obtained non-dominated rule sets.

Table 2. Average number of obtained non-dominated rule sets.

Data set	MOP-1	MOP-2	MOP-3
Breast W	12.09	**13.32**	12.25
Diabetes	9.71	15.80	**17.06**
Glass	13.98	16.49	**27.09**
Heart C	11.56	**22.17**	18.59
Sonar	10.01	**20.47**	17.66
Wine	11.45	9.96	**11.81**

Table 3 summarizes the average value of the best error rate on training patterns among the obtained non-dominated rule sets from each run of our GBML algorithm over ten independent executions of the whole 10-CV procedure. For comparison, we also show the average result for each of the three single-objective formulations. From Table 3, we can see that the best results on training patterns were obtained by SOP-1 for all the six data sets. On the other hand, Table 4 summarizes the average value of the best error rate on test patterns among the obtained non-dominated rule sets from each run. Table 4 clearly demonstrates a potential advantage of multiobjective formulations over single-objective ones for some data sets with respect to the generalization ability of obtained fuzzy rule-based classifiers. We can also see from Table 4 that good results were obtained from different formulations for different data sets.

Table 3. Average best error rates on training patterns among the obtained rule sets from each formulation.

Data set	MOP-1	MOP-2	MOP-3	SOP-1	SOP-2	SOP-3
Breast W	1.59	1.71	1.74	**1.08**	1.44	1.51
Diabetes	19.48	19.79	19.59	**17.74**	18.37	18.41
Glass	25.11	27.08	25.94	**17.81**	21.92	22.36
Heart C	33.43	35.05	34.59	**25.72**	29.65	29.98
Sonar	8.55	8.69	8.42	**3.55**	5.89	5.82
Wine	0.01	0.10	0.03	**0.00**	**0.00**	**0.00**

Table 4. Average best error rates on test patterns among the obtained rule sets from each formulation.

Data set	MOP-1	MOP-2	MOP-3	SOP-1	SOP-2	SOP-3
Breast W	2.93	2.74	**2.66**	3.88	3.69	3.56
Diabetes	23.27	22.32	**21.80**	25.26	25.00	24.20
Glass	35.55	**33.93**	34.05	35.76	39.21	38.36
Heart C	**42.57**	42.85	42.64	44.83	45.80	45.44
Sonar	23.18	**17.32**	17.51	24.04	23.47	24.29
Wine	3.99	3.65	**3.04**	7.30	6.49	6.52

The three multiobjective formulations are compared with each other in terms of error rates on test patterns in Figs. 12-17. The horizontal axis is the number of fuzzy rules in the left plot of each figure while it is the total rule length in the right plot. The same experimental results were shown in the left and right plots of each figure using different coordinates. In the case of MOP-2, the average error rate on test patterns was calculated for rule sets with the same total rule length over 100 runs. When rule sets with a particular value of the total rule length were not obtained from more than 50 runs, the average error rate is not reported for that total rule length. In the case of MOP-3, the average error rate on test patterns was calculated for rule sets with the same number of fuzzy rules and the same total rule length over 100 runs. Only when rule sets with a particular combination of the number of fuzzy rules and the total rule length were obtained from more than 50 runs, the average error rate is reported as a result of MOP-3.

From Figs. 12-17 (as well as Table 4), we can see that good results were obtained from different formulations for different rule sets. The best rule sets with respect to the generalization ability were obtained from MOP-3 for three data sets in Figs. 12-17 (see Table 4). This observation suggests a potential usefulness of the use of both complexity measures (i.e., the number of fuzzy rules and the total rule length) as a safeguard against the overfitting of fuzzy rule-based classifiers to training patterns.

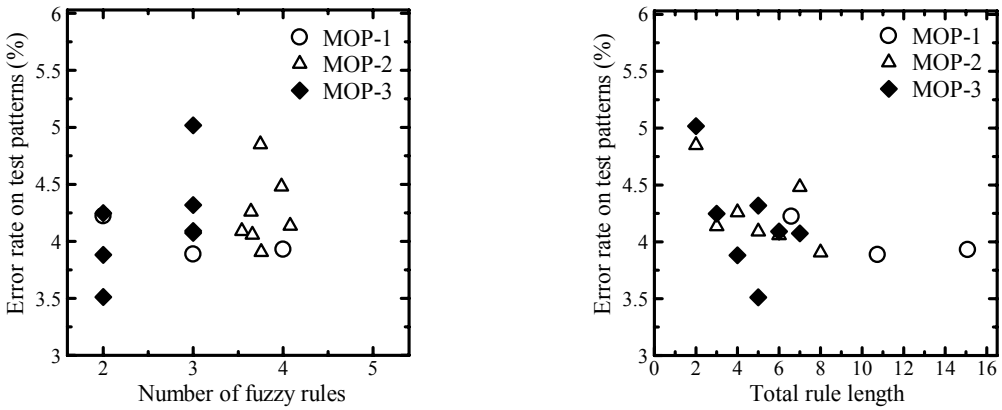


Fig. 12. Experimental results on the Wisconsin breast cancer data set.

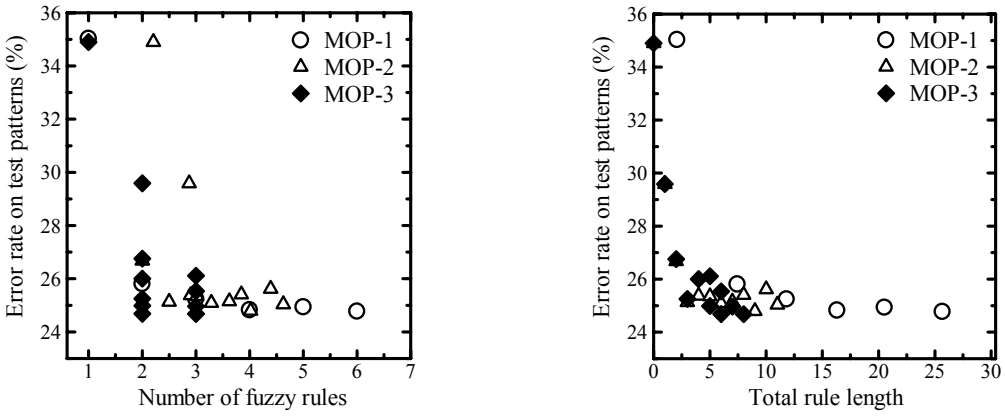


Fig. 13. Experimental results on the diabetes data set.

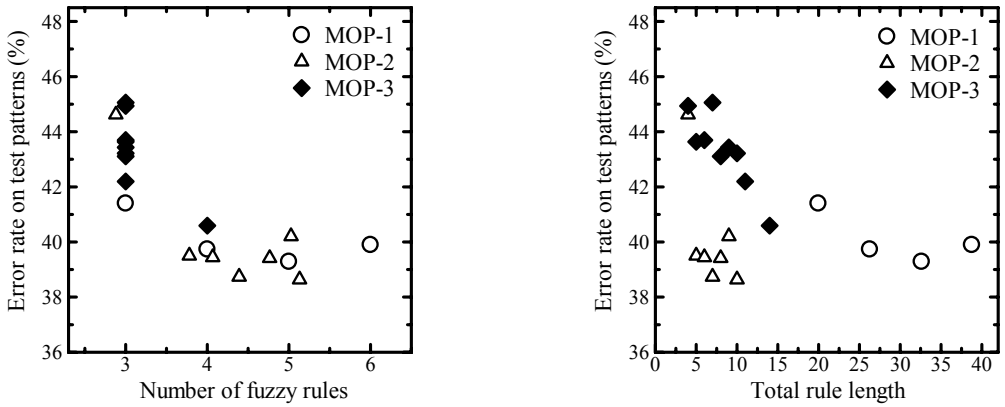


Fig. 14. Experimental results on the glass data set.

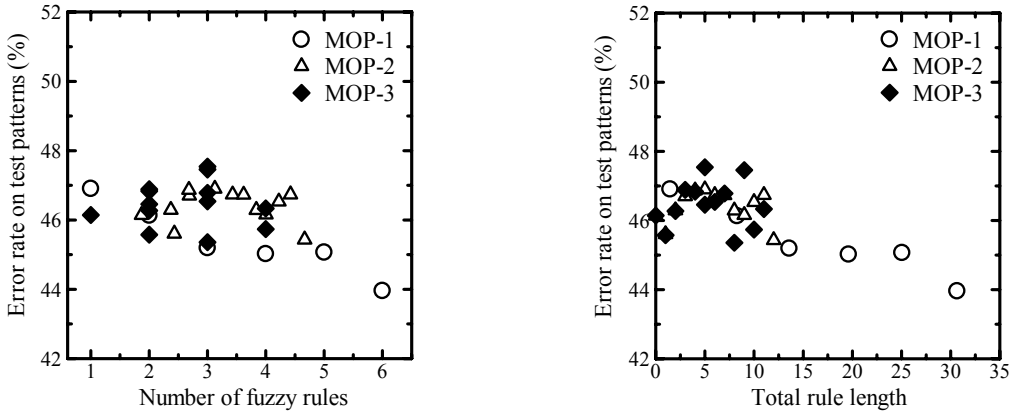


Fig. 15. Experimental results on the Cleveland heart disease data set.

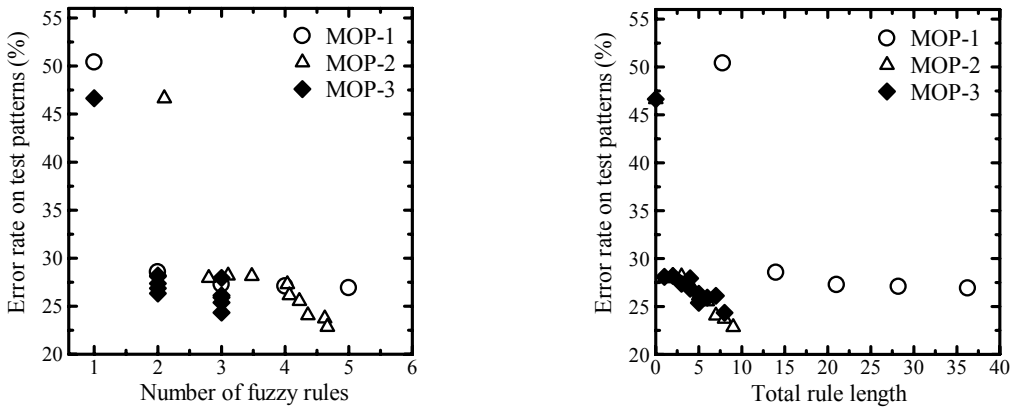


Fig. 16. Experimental results on the sonar data set.

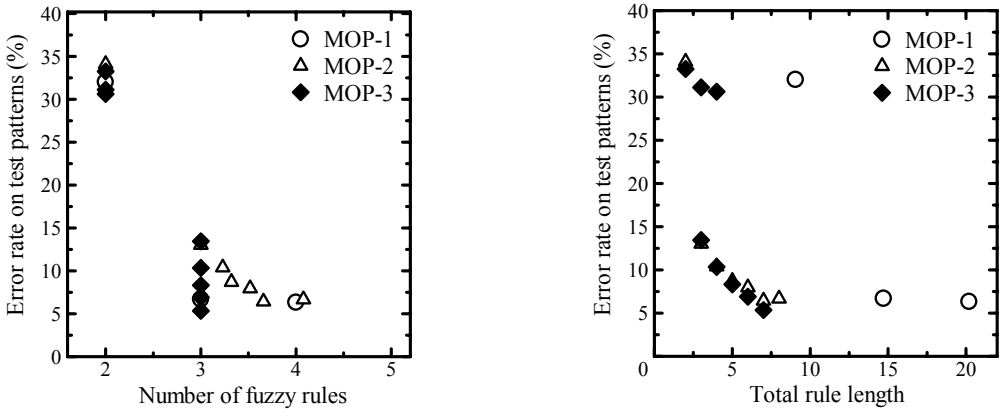
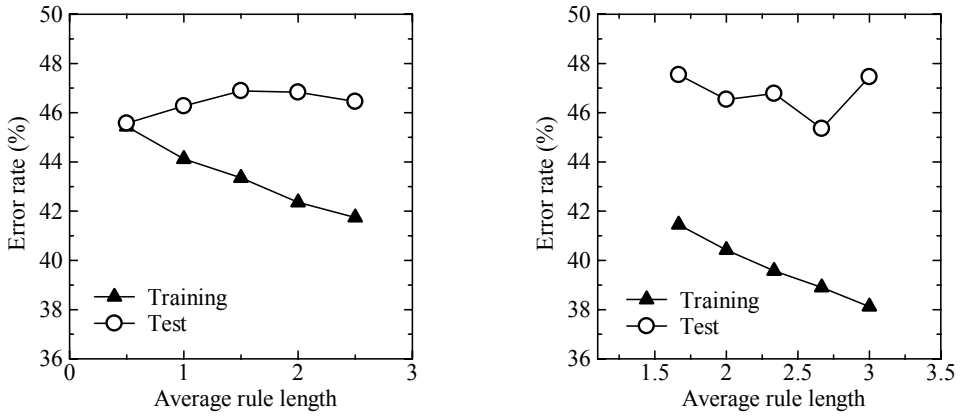


Fig. 17. Experimental results on the wine data set.

One advantage of the three-objective formulation MOP-3 over the two-objective formulations MOP-1 and MOP-2 is that MOP-3 enables us to examine the relation between the generalization ability and the average rule length among multiple rule sets with the same number of fuzzy rules. In Fig. 18, we show an example of such tradeoff analysis on the Cleveland heart disease data set. Fig. 18 is generated from Fig. 15 by concentrating on rule sets with two fuzzy rules and three fuzzy rules.

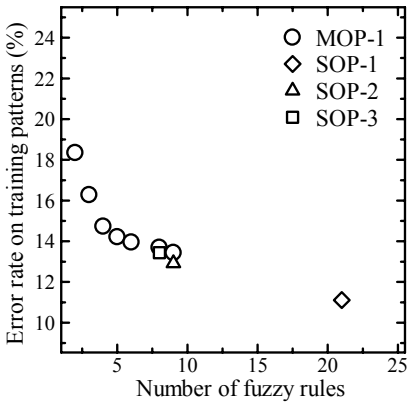


(a) Rule sets with two rules. (b) Rule sets with three rules.

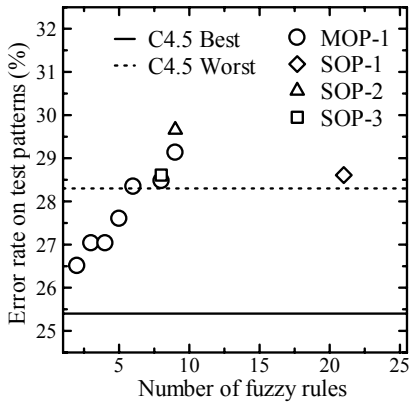
Fig. 18. Experimental results on the Cleveland heart disease data set.

4.5. Parameter specifications

In this subsection, we briefly discuss the specification of two parameters: the total number of generations in our GBML algorithm and the weight vector in each single-objective formulation. In our computational experiments, the total number of generations was specified as 5000. As we will show in Section 5, our GBML algorithm has a huge search space. Thus we may need more computation to obtain near-optimal non-dominated rule sets. In Fig. 19 and Fig. 20, we show experimental results of a single run on the diabetes data set after 20000 generations and 50000 generations, respectively. The corresponding results after 5000 generations were shown in Fig. 2 in Subsection 4.2. From Fig. 2 (a), Fig. 19 (a), and Fig. 20 (a), we can see that error rates on training patterns were improved further by increasing the number of generations. This means that 5000 generations were not enough to obtain near-optimal non-dominated rule sets. This also suggests the necessity to improve the search ability of our GBML algorithm while it is based on the state-of-the-art EMO algorithm: NSGA-II. On the other hand, we can see from Fig. 2 (b), Fig. 19 (b) and Fig. 20 (b) that the improvement in error rates on training patterns did not always lead to the improvement in error rates on test patterns (i.e., the improvement in the generalization ability of fuzzy rule-based classifiers).

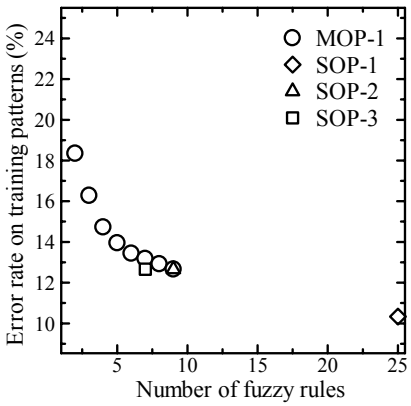


(a) Error rates on training patterns.

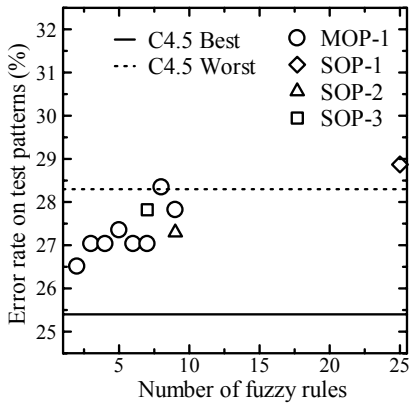


(b) Error rates on test patterns.

Fig. 19. Experimental results of a single run of our GBML algorithm on the diabetes data set after 20000 generations. Corresponding results after 5000 generations were shown in Fig. 2.



(a) Error rates on training patterns.



(b) Error rates on test patterns.

Fig. 20. Experimental results of a single run of our GBML algorithm on the diabetes data set after 50000 generations.

The weight values in the single-objective formulations were specified as $w_1 = 10$, $w_2 = 1$, and $w_3 = 1$ in our computational experiments. In general, obtained solutions strongly depend on weight values when we use the weighted sum of multiple objectives as a scalar fitness function to solve multi-objective optimization problems as single-objective ones. In Tables 5-7, we show

the dependency of experimental results by SOP-1 on the specification of the weight vector $\mathbf{w} = (w_1, w_2)$. In these tables, we examined six combinations of the weight values w_1 and w_2 using 10 independent executions of the whole 10-CV procedure (i.e., 10×10-CV) as in Tables 2-4. From these figures, we can see that totally different results were obtained from different specifications of the weight values. We can also see from these tables that the weight vector (10, 1) seems to be a good compromise between the accuracy and the complexity.

Table 5. Average number of fuzzy rules.

Data set	(1000, 1)	(100, 1)	(10, 1)	(1, 1)	(1, 10)	(1, 100)
Breast W	22.84	22.84	5.13	3.37	2.00	2.00
Diabetes	31.53	32.97	9.43	6.56	2.13	1.00
Glass	28.32	28.32	9.06	7.09	3.02	1.00
Heart C	28.37	28.65	8.91	7.33	1.04	1.00
Sonar	30.23	29.67	6.81	5.40	2.03	1.01
Wine	4.80	4.70	4.34	3.50	3.00	1.00

Table 6. Average best error rates on training patterns among the obtained rule sets.

Data set	(1000, 1)	(100, 1)	(10, 1)	(1, 1)	(1, 10)	(1, 100)
Breast W	1.01	1.01	1.51	1.87	2.67	2.69
Diabetes	17.86	17.78	18.41	18.79	22.12	34.90
Glass	17.91	17.91	22.36	23.77	35.19	64.98
Heart C	26.02	26.00	29.98	31.67	45.95	46.13
Sonar	3.75	3.56	5.82	6.49	18.02	46.34
Wine	0.00	0.00	0.00	0.49	1.60	60.11

Table 7. Average best error rates on test patterns among the obtained rule sets.

Data set	(1000, 1)	(100, 1)	(10, 1)	(1, 1)	(1, 10)	(1, 100)
Breast W	3.87	3.87	3.56	3.83	4.03	4.14
Diabetes	24.86	25.16	24.20	25.03	25.50	34.97
Glass	37.03	37.03	38.36	38.52	42.93	73.66
Heart C	46.06	45.79	45.44	46.10	46.48	46.25
Sonar	24.84	24.01	24.29	23.58	29.59	47.41
Wine	7.02	7.02	6.52	7.99	8.71	61.01

4.6. Choice of a single fuzzy rule-based classifier

As we have already explained, multiple fuzzy rule-based classifiers are obtained from a single run of our GBML algorithm when we use one of the three multiobjective formulations. In a real-world application of our GBML algorithm, we usually have to choose a single fuzzy rule-based classifier as a final solution from our GBML algorithm. In this case, our GBML algorithm may be used in the following manner.

First we divide the available training patterns into two subsets. One subset is used to generate multiple fuzzy rule-based classifiers by our GBML algorithm. The other subset is used to evaluate the generalization ability of each fuzzy rule-based classifier. By iterating this training-and-testing procedure several times using different partitions of the available training patterns, we can visualize the interpretability-accuracy tradeoff structure for test patterns. Next we choose an appropriate complexity level of fuzzy rule-based classifiers using the visualized tradeoff structure. In this stage, we can take into account both the interpretability and the generalization ability of fuzzy rule-based classifiers. Then our GBML algorithm is used to generate multiple non-dominated fuzzy rule-based classifiers using all the available training patterns. Finally one fuzzy rule-based classifier whose complexity is similar to the specified appropriate complexity level is chosen as the final solution from the obtained non-dominated fuzzy rule-based classifiers. In this manner, we can choose a single fuzzy rule-based classifier using the visualized interpretability-accuracy tradeoff structure for test patterns.

5. Comparison with multiobjective fuzzy rule selection

In our GBML algorithm, each fuzzy rule for an *n*-dimensional classification problem is represented as an integer string of length *n* using 15 antecedent fuzzy sets. The total number of such strings is 15^n. Each rule set with *r* fuzzy rules is represented by an integer string of length $n \cdot r$. The total number of such strings is $15^{n \cdot r}$. In our computational experiments, we specified the upper bound on the number of fuzzy rules as 40. Thus the size of the search space is calculated as $\sum_{r=1}^{40} 15^{n \cdot r}$, which depends on the dimensionality of the pattern space (i.e., *n*).

On the other hand, first a prespecified number of candidate fuzzy rules are generated from training patterns in multiobjective fuzzy rule selection [15], [16]. Then an evolutionary

multiobjective optimization (EMO) algorithm is used to find non-dominated rule sets from the candidate fuzzy rules. Let N be the number of candidate fuzzy rules. In this case, each rule set is represented by a binary string of length N. Thus the size of the search space in multiobjective fuzzy rule selection is 2^N, which depends on the number of candidate fuzzy rules (i.e., N).

In Table 8, we compare the size of the search space between the two approaches to multiobjective design of fuzzy rule-based classifiers. We can see from Table 8 that the size of the search space exponentially increases with the dimensionality of the pattern space (i.e., n) in our GBML algorithm while it exponentially increases with the number of candidate fuzzy rules (i.e., N) in multiobjective fuzzy rule selection. We can also see from Table 8 that the size of the search space in multiobjective fuzzy rule selection is much smaller than that in our GBML algorithm for high-dimensional pattern classification problems.

Table 8. Comparison in the search space size between the two approaches to multiobjective design of fuzzy rule-based classifiers.

Multiobjective GBML Approach						
Dimensionality (n)	10	20	40	60	80	100
Search Space Size	2.7×10^{470}	7.5×10^{940}	5.6×10^{1881}	4.2×10^{2822}	3.1×10^{3763}	2.3×10^{4704}
Multiobjective Fuzzy Rule Selection						
Candidate Rules (N)	100	200	400	600	800	1000
Search Space Size	1.3×10^{30}	1.6×10^{60}	2.6×10^{120}	4.1×10^{180}	6.7×10^{240}	1.1×10^{301}

The difference in the size of the search space between the two approaches in Table 8 suggests the difficulty in searching for good fuzzy rule-based classifiers by our GBML algorithm. Experimental results in this paper, however, showed that good fuzzy rule-based classifiers were obtained by our GBML algorithm (whereas further improvement in the search ability of our GBML algorithm is required as shown in Subsection 4.5). As demonstrated in [19] for single-objective design of fuzzy rule-based classifiers, the search ability of fuzzy GBML algorithms strongly depends on the heuristic generation procedure of initial rules from training patterns. Since we used good initial rules generated from training patterns in a heuristic manner,

our GBML algorithm could efficiently find good fuzzy rule-based classifiers. If we use randomly generated initial fuzzy rules, it is very difficult to find good fuzzy rule-based classifiers in the huge search space (see [19]). In our computational experiments for the three single-objective formulations, we used the same generation update procedure as the NSGA-II algorithm. That is, the best rule sets were selected from the current and offspring populations. Slightly better results were obtained in this paper than Ishibuchi *et al.* [19] where the generation update procedure was based on a single elite solution.

On the other hand, the performance of non-dominated rule sets obtained from multiobjective rule selection strongly depends on the quality of candidate fuzzy rules. When the quality of candidate fuzzy rules is not high, it is very difficult for any EMO algorithms to find good non-dominated rule sets with high classification performance. Thus the smaller search space size in multiobjective rule selection does not always mean better non-dominated rule sets than the case of our GBML algorithm with the larger search space size.

6. Concluding remarks

In this paper, we examined the interpretability-accuracy tradeoff of fuzzy rule-based classifiers through computational experiments on some benchmark data sets using a hybrid multiobjective fuzzy GBML algorithm. Experimental results showed that there exists a clear interpretability-accuracy tradeoff structure in each data set with respect to error rates on training patterns. Such a tradeoff structure is not always clear with respect to error rates on test patterns. That is, our interpretability-accuracy tradeoff analysis showed that each data set has a different relation between the complexity of fuzzy rule-based classifiers and their generalization ability for test patterns. Our experimental results also suggested potential advantages of multiobjective formulation over single-objective ones. That is, the use of the number of fuzzy rules and/or the total rule length as a complexity measure worked well as a safeguard against the overfitting of fuzzy rule-based classifiers to training patterns.

In some computational experiments (e.g., Fig. 2 (a)), lower error rates were obtained from the single-objective formulations than the multiobjective ones. This observation suggests the necessity of the improvement in the search ability of our GBML algorithm. As shown in [33], it

is not easy for EMO algorithms to find a variety of Pareto-optimal (or near Pareto-optimal) solutions with a wide range of objective values of large-scale multiobjective combinatorial optimization problems. Multiobjective design of fuzzy rule-based classifiers is an example of such a large-scale multiobjective combinatorial optimization problem. It is left for future research to improve the search ability of EMO algorithms to find a variety of fuzzy rule-based classifiers with a wide range of objective values (especially to find fuzzy rule-based classifiers with high accuracy on training patterns).

Acknowledgements

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

References

[1] O. Cordon, F. Herrera, F. Hoffman, L. Magdalena, Genetic Fuzzy Systems, World Scientific, Singapore, 2001.

[2] Y. Jin, W. von Seelen, B. Sendhoff, On generating FC^3 fuzzy rule systems from data using evolution strategies, IEEE Trans. on Systems, Man, and Cybernetics: Part B 29 (6) (1999) 829-845.

[3] Y. Jin, Fuzzy modeling of high-dimensional systems: Complexity reduction and interpretability improvement, IEEE Trans. on Fuzzy Systems 8 (2) (2000) 212-221.

[4] M. Setnes, H. Roubos, GA-based modeling and classification: Complexity and performance, IEEE Trans. on Fuzzy Systems 8 (5) (2000) 509-522.

[5] J. Casillas, O. Cordon, F. Herrera, L. Magdalena (eds.), Interpretability Issues in Fuzzy Modeling, Springer, Berlin, 2003.

[6] J. Casillas, O. Cordon, F. Herrera, L. Magdalena (eds.), Accuracy Improvements in Linguistic Fuzzy Modeling, Springer, Berlin, 2003.

[7] F. Jimenez, A. F. Gomez-Skarmeta, G. Sanchez, H. Roubos, R. Babuska, Accurate, transparent and compact fuzzy models for function approximation and dynamic modeling through multi-objective evolutionary optimization, in: Proceedings of the 1st International

Conference on Evolutionary Multi-Criterion Optimization, 2001, Zurich, pp. 653-667.

[8] 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 35 (2) (2005) 143-155.

[9] 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 149 (1) (2005) 149-186.

[10] Y. Jin, T. Okabe, B. Sendhoff, Neural network regularization and ensembling using multi-objective evolutionary algorithms, in: Proceedings of 2004 Congress on Evolutionary Computation, 2004, Portland, pp. 1-8.

[11] L. S. Oliveira, M. Morita, R. Sabourin, F. Bortolozzi, Multi-objective genetic algorithms to create ensemble of classifiers, in: Proceedings of the 3rd International Conference on Evolutionary Multi-Criterion Optimization, 2005, Guanajuato, pp. 592-606.

[12] H. Ishibuchi, K. Nozaki, N. Yamamoto, H. Tanaka, Construction of fuzzy classification systems with rectangular fuzzy rules using genetic algorithms, Fuzzy Sets and Systems 65 (2/3) (1994) 237-253.

[13] H. Ishibuchi, K. Nozaki, N. Yamamoto, H. Tanaka, Selecting fuzzy if-then rules for classification problems using genetic algorithms, IEEE Trans. on Fuzzy Systems 3 (3) (1995) 260-270.

[14] H. Ishibuchi, T. Murata, I. B. Turksen, Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems, Fuzzy Sets and Systems 89 (2) (1997) 135-150.

[15] H. Ishibuchi, T. Nakashima, T. Murata, Three-objective genetics-based machine learning for linguistic rule extraction, Information Sciences 136 (1-4) (2001) 109-133.

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

[17] H. Ishibuchi, T. Nakashima, T. Murata, Performance evaluation of fuzzy classifier systems for multi-dimensional pattern classification problems, IEEE Trans. on Systems, Man, and Cybernetics: Part B 29 (5) (1999) 601-618.

[18] H. Ishibuchi, T. Nakashima, Improving the performance of fuzzy classifier systems for pattern classification problems with continuous attributes, IEEE Trans. on Industrial Electronics 46 (6) (1999) 157-168.

[19] H. Ishibuchi, T. Yamamoto, T. Nakashima, Hybridization of fuzzy GBML approaches for pattern classification problems, IEEE Trans. on Systems, Man, and Cybernetics: Part B 35 (2) (2005) 359-365.

[20] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. on Evolutionary Computation 6 (2) (2002) 182-197.

[21] K. Deb, Multi-objective optimization using evolutionary algorithms, John Wiley & Sons, Chichester, 2001.

[22] C. A. C. Coello, D. A. van Veldhuizen, G. B. Lamont, Evolutionary algorithms for solving multi-objective problems, Kluwer Academic Publishers, Boston, 2002.

[23] C. A. C. Coello, A comprehensive survey of evolutionary-based multiobjective optimization techniques, Knowledge and Information Systems 1 (3) (1999) 269-308.

[24] D. A. Van Veldhuizen, G. B. Lamont, Multiobjective evolutionary algorithms: Analyzing the state-of-the-art, Evolutionary Computation 8 (2) (2000) 125-147.

[25] H. Ishibuchi, T. Nakashima, M. Nii, Classification and modeling with linguistic information granules: Advanced approaches to linguistic data mining, Springer, Berlin, 2004.

[26] H. Ishibuchi, T. Nakashima, Effect of rule weights in fuzzy rule-based classification systems, IEEE Trans. on Fuzzy Systems 9 (4) (2001) 506-515.

[27] H. Ishibuchi, T. Yamamoto, Rule weight specification in fuzzy rule-based classification systems, IEEE Trans. on Fuzzy Systems 13 (4) (2005) 428-435.

[28] H. Ishibuchi, T. Nakashima, T. Morisawa, Voting in fuzzy rule-based systems for pattern classification problems, Fuzzy Sets and Systems 103 (2) (1999) 223-238.

[29] O. Cordon, M. J. del Jesus, F. Herrera, A proposal on reasoning methods in fuzzy rule-based classification systems, International Journal of Approximate Reasoning 20 (1) (1999) 21-45.

[30] T. Elomaa, J. Rousu, General and efficient multisplitting of numerical attributes, Machine Learning 36 (3) (1999) 201-244.

[31] J. R. Quinlan, C4.5: Programs for machine learning, Morgan Kaufmann, San Mateo, 1993.

[32] J. R. Quinlan, Improved use of continuous attributes in C4.5, Journal of Artificial Intelligence Research 4 (1996) 77-90.

[33] H. Ishibuchi, K. Narukawa, Comparison of evolutionary multiobjective optimization with reference solution-based single-objective approach, in Proceedings of 2005 Genetic and Evolutionary Computation Conference, 2005, Washington DC, pp. 787-794.