ELSEVIER

Fuzzy Sets and Systems 89 (1997) 135–150

FUZZY sets and systems

Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems

Hisao Ishibuchi[a,*], Tadahiko Murata[a], I.B. Türkşen[b]

[a]*Department of Industrial Engineering, Osaka Prefecture University, Gakuen-cho 1-1, Sakai, Osaka 593, Japan*
[b]*Department of Industrial Engineering, University of Toronto, Toronto, Ont., Canada M5S 1A4*

Received October 1995; revised March 1996

Abstract

This paper proposes various methods for constructing a compact fuzzy classification system consisting of a small number of linguistic classification rules. First we formulate a rule selection problem of linguistic classification rules with two objectives: to maximize the number of correctly classified training patterns and to minimize the number of selected rules. Next we propose three methods for finding a set of non-dominated solutions of the rule selection problem. These three methods are based on a single-objective genetic algorithm. We also propose a method based on a multi-objective genetic algorithm for finding a set of non-dominated solutions. We examine the performance of the proposed methods by applying them to the well-known iris data. Finally we propose a hybrid algorithm by combining a learning method of linguistic classification rules with the multi-objective genetic algorithm. High performance of the hybrid algorithm is demonstrated by computer simulations on the iris data. © 1997 Elsevier Science B.V.

Keywords: Linguistic modeling; Pattern recognition; Data analysis; Genetic algorithms; Rule selection

1. Introduction

Fuzzy systems based on fuzzy if-then rules have been applied to various control problems [19, 28]. Fuzzy if-then rules in those fuzzy systems were usually derived from human experts. Recently, several approaches have been proposed for automatically generating fuzzy if-then rules from numerical data without domain experts (see, for example [29, 32, 35]). Genetic algorithms [3, 5] have been widely used for generating fuzzy if-then rules and tuning the membership functions of antecedent and consequent fuzzy sets. For example, Thrift [33], Feldman [1], and Kropp and Baitinger [18] employed genetic algorithms for generating fuzzy if-then rules. Membership functions were adjusted by genetic algorithms in Karr [15], Karr and Gentry [16], Surmann et al. [30], and Herrera et al. [4]. Both the generation of fuzzy if-then rules and the tuning of membership functions were performed by genetic algorithms in Kinzel et al. [17], Satyadas and Krishnakumar [26], Homaifar and McCormick [6], and Park et al. [24]. The number of fuzzy if-then rules was also determined by genetic

* Corresponding author. Fax: + 81-722-59-3340; e-mail: hisaoi@ie.osakafu-u.ac.jp.

0165-0114/97/$17.00 © 1997 Elsevier Science B.V. All rights reserved
PII S0165-0114(96)00098-X

algorithms in Nomura et al. [22], Liska and Melsheimer [21], Lee and Takagi [20], and Ishigami et al. [14]. Hierarchical structures of fuzzy if-then rules were determined by genetic algorithms in Shimojima et al. [27]. In those genetic-algorithm-based approaches, a rule set (i.e., a rule table) of fuzzy if-then rules was coded as an individual. On the other hand, a single fuzzy if-then rule was coded as an individual in fuzzy classifier systems of Valenzuela–Rendon [34] and Parodi and Bonelli [25].

The above-mentioned approaches were mainly applied to fuzzy control problems such as cart centering problems [1, 4, 6, 15, 17, 18, 20, 33], a pH control problem [16], a spacecraft attitude control problem [26], a truck backing problem [6], and a dc series motor control problem [24]. Some approaches were applied to function approximation problems [14, 21, 22, 25, 27, 30, 34].

Fuzzy systems based on fuzzy if-then rules have also been applied to pattern classification problems. Ishibuchi et al. [11] proposed a generation method of fuzzy if-then rules from numerical data for pattern classification problems, and Nozaki et al. [23] proposed a learning method of the generated fuzzy if-then rules. Genetic algorithms were used in Ishibuchi et al. [12, 13] for selecting a small number of fuzzy if-then rules with high classification performance. In genetic-algorithm-based rule selection methods [12, 13], the following fuzzy if-then rules were used as candidate rules for an n-dimensional pattern classification problem:

$$\text{Rule } R_j\text{: If } x_{p1} \text{ is } A_{j1} \text{ and } \ldots \text{ and } x_{pn} \text{ is } A_{jn} \text{ then } \boldsymbol{x}_p \text{ is Class } C_j \text{ with } CF = CF_j, \quad j = 1, 2, \ldots, N, \tag{1}$$

where R_j is a label of rule, $\boldsymbol{x}_p = (x_{p1}, \ldots, x_{pn})$ is an n-dimensional pattern vector, A_{ji} is an antecedent fuzzy set on the i-th axis of the pattern space, C_j is a consequent class, CF_j is the grade of certainty, and N is the total number of candidate rules. Because various fuzzy sets shown in Fig. 1 were used as antecedent fuzzy sets in the rule selection methods [12, 13], the linguistic interpretation of selected fuzzy if-then rules was not always easy.

In order to select fuzzy if-then rules that can always be interpreted linguistically, Ishibuchi et al. [7, 8] restricted the antecedent fuzzy sets of candidate fuzzy if-then rules to the six linguistic values shown in Fig. 2. That is, the antecedent fuzzy set A_{ji} was one of the six linguistic values in [7, 8].

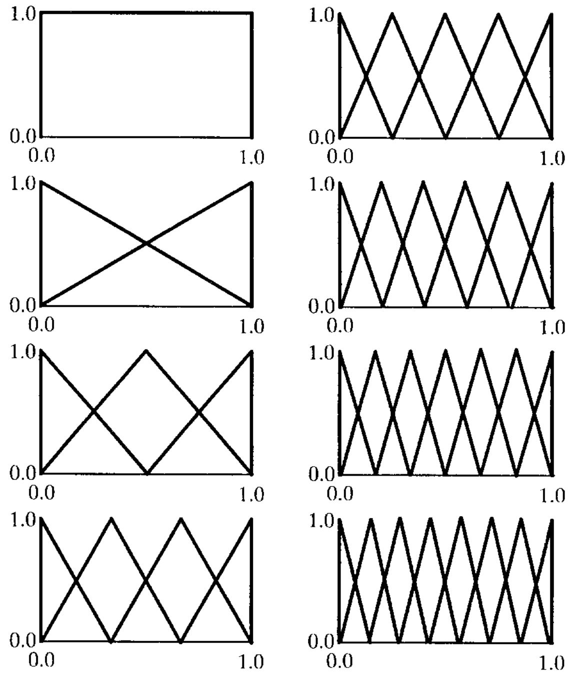

Fig. 1. Various antecedent fuzzy sets.

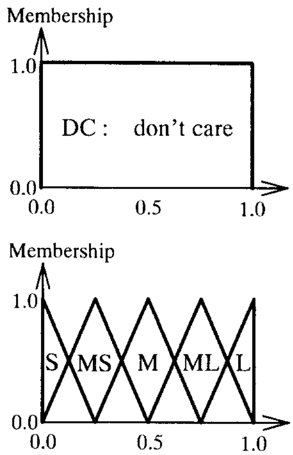

Fig. 2. Antecedent fuzzy sets of linguistic classification rules (DC: don't care, S: small, MS: medium small, M: medium, ML: medium large, and L: large).

Fuzzy if-then rules with linguistic values in their antecedent part were referred to as "linguistic classification rules" in [7, 8].

The rule selection problem of the linguistic classification rules in [7, 8], which is also discussed in this paper, has the following two objectives:

(i) To maximize the number of correctly classified training patterns by selected rules.

(ii) To minimize the number of selected rules.

These two objectives were combined into a single scalar fitness function using constant weights in [7, 8]. An idea of a multi-objective genetic algorithm was proposed to find a set of non-dominated solutions of the rule selection problem with the above two objectives in [9]. A fuzzy classifier system [10] was proposed to handle a rule selection problem with only the first objective for multi-dimensional pattern classification problems involving many features.

The main aim of this paper is to propose several methods for finding a set of non-dominated solutions of the rule selection problem with the above two objectives. First we briefly describe the formulation of the rule selection problem of linguistic classification rules. Next we propose three methods based on a single-objective genetic algorithm for finding the non-dominated solutions of the rule selection problem. We also propose a method based on a multi-objective genetic algorithm. Finally we propose a hybrid algorithm by combining a learning method [23] of linguistic classification rules with the multi-objective genetic algorithm. The performance of the proposed methods are examined by applying them to the classification problem of the iris data (see, for example, [2]).

2. Rule selection problem of linguistic classification rules

2.1. Pattern classification problem

We assume that m patterns $x_p = (x_{p1}, \ldots, x_{pn})$, $p = 1, 2, \ldots, m$ from c classes are given as training data in an n-dimensional pattern space $[0, 1]^n$. Thus, our pattern classification problem is a c-class problem in the n-dimensional hyper-cube $[0, 1]^n$. We show an example of the pattern classification problem in Fig. 3 where $m = 121$, $c = 2$ and $n = 2$. In Fig. 3, training patterns from Class 1 and Class 2 are shown by closed circles and open circles, respectively.

We also assume that linguistic values are given for each axis of the pattern space. While we use the six linguistic values in Fig. 2 for formulating the rule selection problem in this paper, we can use arbitrary linguistic values such as five values in Fig. 4 when we apply our approach to a specific pattern classification problem. The selection of linguistic values and the determination of their membership functions should be done by human experts. If no human experts can specify linguistic values for each axis of the pattern space, we may

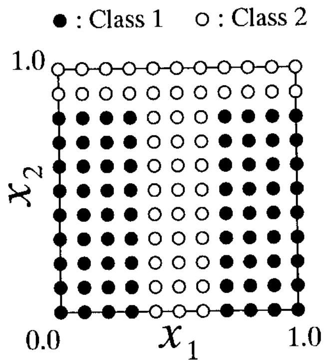

Fig. 3. An example of the pattern classification problem.

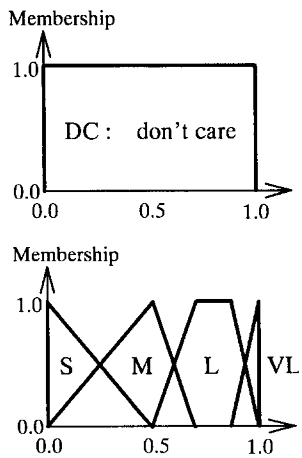

Fig. 4. Example of a set of alternative linguistic values (DC: don't care, S: small, M: medium, L: large, and VL: very large).

use typical linguistic values such as the six linguistic values in Fig. 2 for all the n axes of the pattern space.

2.2. Fuzzy partition of a pattern space

When we use the six linguistic values in Fig. 2 for each axis of the n-dimensional pattern space, $N = 6^n$ linguistic classification rules can be generated from the training patterns $\boldsymbol{x}_p = (x_{p1}, \ldots, x_{pn})$, $p = 1, 2, \ldots, m$ because each antecedent fuzzy set A_{ji} in (1) may assume one of the six linguistic values. For example, $6^2 = 36$ linguistic classification rules can be generated for the two-dimensional pattern space $[0, 1]^2$ of the classification problem in Fig. 3. In this case, 36 fuzzy subspaces are generated in the pattern space $[0, 1]^2$ as shown in Fig. 5, and a linguistic classification rule is assigned to each fuzzy subspace. From Fig. 5, we can see that several linguistic rules are overlapping in the pattern space. This means that some of the 36 linguistic rules in Fig. 5 may be redundant for the classification task.

2.3. Rule generation

The consequent C_j and the grade of certainty CF_j of each linguistic classification rule in (1) can be determined by the given training patterns $\boldsymbol{x}_p = (x_{p1}, \ldots, x_{pn})$, $p = 1, 2, \ldots, m$ in the same manner as in the rule generation method of fuzzy if-then rules in [11]. First let us define the grade of compatibility of $\boldsymbol{x}_p$ to the jth linguistic classification rule R_j in (1) as

$$\mu_j(\boldsymbol{x}_p) = \mu_{A_{j1}}(x_{p1}) \cdot \cdots \cdot \mu_{A_{jn}}(x_{pn}), \tag{2}$$

where $\mu_{A_{ji}}(x_{pi})$ is the membership function of the antecedent fuzzy set A_{ji}. Thus the total grade of compatibility to the jth rule R_j is calculated for

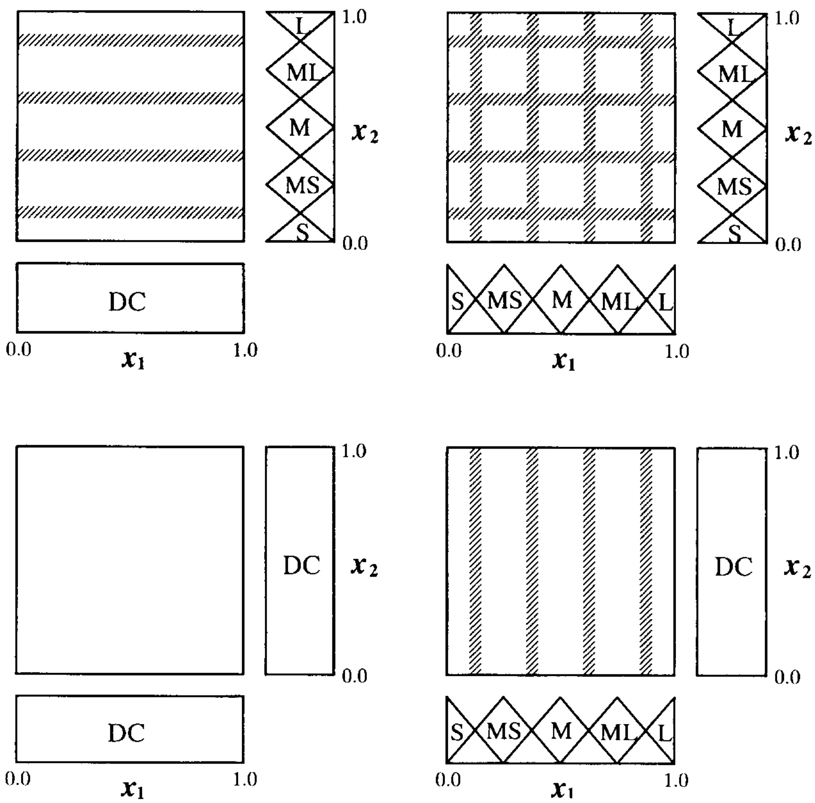

Fig. 5. Fuzzy partitions of the two-dimensional pattern space by the six linguistic values.

each class as

$$\beta_{\text{Class }h} = \sum_{x_p \in \text{Class }h} \mu_j(\boldsymbol{x}_p) = \sum_{x_p \in \text{Class }h} \mu_{A_{j1}}(x_{p1}) \cdot \cdots \cdot \mu_{A_{jn}} \cdot (x_{pn}), \quad h = 1, 2, \ldots, c, \tag{3}$$

where $\beta_{\text{Class }h}$ is the total grade of compatibility of the given patterns in Class h to the jth rule R_j in (1).

The consequent C_j is determined as the class with the maximum total grade of compatibility. That is, C_j is determined as Class $\hat{h}$ by

$$\beta_{\text{Class }\hat{h}} = \max\{\beta_{\text{Class }1}, \beta_{\text{Class }2}, \ldots, \beta_{\text{Class }c}\}. \tag{4}$$

If Class $\hat{h}$ is not determined uniquely (i.e., if two or more classes have the same maximum value in (4)), we assign ϕ to C_j where ϕ means an empty class. For example, the consequent class C_j is determined as Class 1 in Fig. 6(a)–(c) while ϕ is assigned to C_j in Fig. 6(d). The consequent C_j also becomes ϕ when $\beta_{\text{Class }h} = 0$ for all classes. This means that a linguistic classification rule with ϕ in the consequent part is generated when there are no patterns compatible with that rule. In this paper, linguistic classification rules with ϕ in the consequent part are referred to as "dummy rules" because those rules have no effect on the classification of new patterns.

The grades of certainty of all dummy rules are specified as $CF_j = 0$. For non-dummy rules, the grade of certainty CF_j is determined as

$$CF_j = \frac{\beta_{\text{Class }\hat{h}} - \bar{\beta}}{\sum_{h=1}^{c} \beta_{\text{Class }h}}, \tag{5}$$

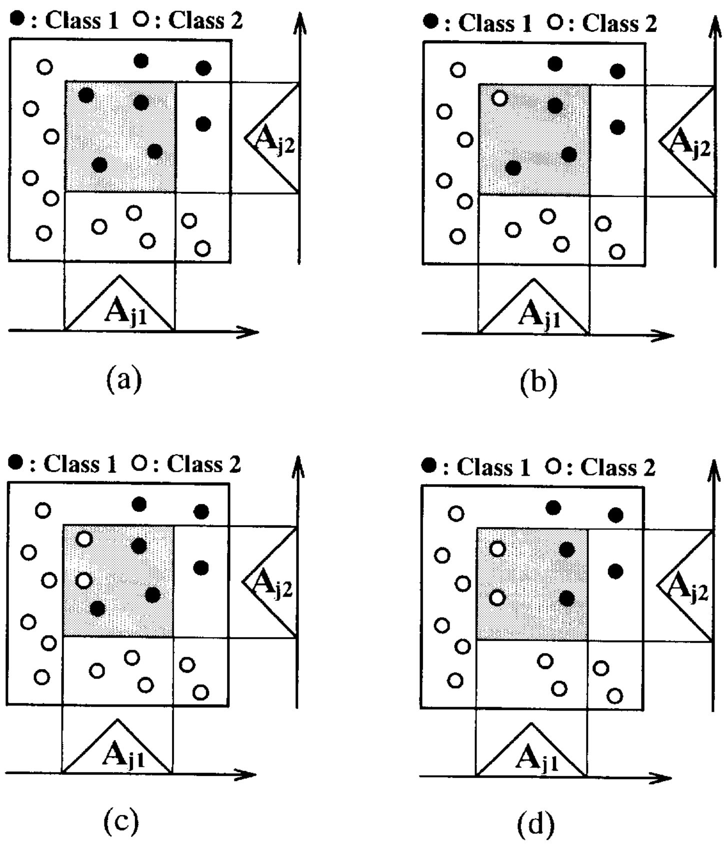

Fig. 6. Antecedent fuzzy sets and training patterns.

where

$$\bar{\beta} = \left(\sum_{h \neq \hat{h}} \beta_{\text{Class } h} \right) \Big/ (c - 1). \tag{6}$$

The grade of certainty CF_j is maximum (i.e., $CF_j = 1$) when $\beta_{\text{Class } \hat{h}} > 0$ and $\beta_{\text{Class } h} = 0$ for $h \neq \hat{h}$. That is, if all the patterns compatible with the jth linguistic classification rule R_j belong to the same class, the grade of certainty CF_j of this rule is equal to 1 (the maximum certainty). On the contrary, if the total grades of compatibility for the c classes are similar to one another (i.e., $\beta_{\text{Class } 1} \cong \cdots \cong \beta_{\text{Class } c}$), the grade of certainty is nearly equal to 0 (the minimum certainty). Among the four situations in Fig. 6, the grade of certainty CF_j is maximum in Fig. 6(a), and it is minimum in Fig. 6(d). The grade of certainty CF_j in Fig. 6(b) is larger than that in Fig. 6(c).

By applying the rule generation method described above to all the linguistic classification rules in (1), we have $N = 6^n$ linguistic classification rules including dummy rules. Let us denote the set of the generated N linguistic classification rules by S_{ALL}:

$$S_{\text{ALL}} = \{\text{Rule } R_j \mid j = 1, 2, \ldots, N\}. \tag{7}$$

All the linguistic classification rules in S_{ALL} are used in the rule selection problem as candidate rules.

2.4. Fuzzy reasoning

Let us denote a subset of S_{ALL} by S. That is, S is a set of linguistic classification rules. A new pattern $\boldsymbol{x}_p = (x_{p1}, \ldots, x_{pn})$ is classified by linguistic classification rules in S as follows:

Step 1: Calculate $\alpha_{\text{Class } h}$ for $h = 1, 2, \ldots, c$ as

$$\alpha_{\text{Class } h} = \max\{\mu_j(\boldsymbol{x}_p) \cdot CF_j \mid C_j = \text{Class } h \text{ and Rule } R_j \in S\}, \tag{8}$$

where $\mu_j(\boldsymbol{x}_p)$ is the grade of compatibility of $\boldsymbol{x}_p$ to the jth linguistic classification rule R_j, which is defined by (2).

Step 2: Find the maximum value of $\alpha_{\text{Class } h}$ as

$$\alpha_{\text{Class } \hat{h}} = \max\{\alpha_{\text{Class } 1}, \ldots, \alpha_{\text{Class } c}\}. \tag{9}$$

If two or more classes take the same maximum value in (9), then the classification of $\boldsymbol{x}_p$ is rejected (i.e. $\boldsymbol{x}_p$ is left as an unclassifiable pattern), else assign $\boldsymbol{x}_p$ to Class $\hat{h}$ determined by (9).

In this procedure, a new pattern $\boldsymbol{x}_p = (x_{p1}, \ldots, x_{pn})$ is classified by the linguistic classification rule that has the maximum product of $\mu_j(\boldsymbol{x}_p)$ and CF_j.

2.5. Formulation of the rule selection problem

Our rule selection problem is to select a small number of linguistic classification rules from the rule set S_{ALL} to construct a compact classification system S with high classification performance. Therefore our problem can be written as follows:

$$\text{Maximize } NCP(S) \text{ and minimize } |S|, \tag{10}$$

$$\text{subject to } S \subseteq S_{\text{ALL}}, \tag{11}$$

where $NCP(S)$ is the number of correctly classified training patterns by linguistic classification rules in a rule set S, and $|S|$ is the number of the linguistic classification rules in S.

3. Single-objective genetic algorithm for the rule selection problem

3.1. A single-objective genetic algorithm

A single-objective genetic algorithm was applied to the rule selection problem (10), (11) for selecting a small number of linguistic classification rules from a large number of candidate rules in S_{ALL} in Ishibuchi et al. [7,8]. In their genetic algorithm, each rule set S is treated as an individual. A scalar fitness value of S is defined from the two objectives in (10) using constant weights as follows [7,8]:

$$f(S) = W_{NCP} \cdot NCP(S) - W_S \cdot |S|, \tag{12}$$

where W_{NCP} and W_S are constant positive weights assigned to the two objectives $NCP(S)$ and $|S|$, respectively.

Each individual (i.e. each rule set S) is represented by a string as $S = s_1 s_2 \ldots s_N$, where N is the number of the linguistic rules in S_{ALL} and $s_j = 1, -1$ or 0 denotes the following:

(i) $s_j = 1$ means that the jth rule R_j is included in the rule set S,

(ii) $s_j = -1$ means that the jth rule R_j is not included in the rule set S,

(iii) $s_j = 0$ means that the jth rule R_j is a dummy rule.

Since dummy rules have no effect on the classification of new patterns, they should be excluded from a rule set S. Therefore the special coding ($s_j = 0$) is assigned to dummy rules in order to prevent S from including them. A string $S = s_1 s_2 \ldots s_N$ is decoded as

$$S = \{\text{Rule } R_j \mid s_j = 1; j = 1, 2, \ldots, N\}. \quad (13)$$

A set of strings (i.e., a set of rule sets) is treated as a population (i.e., as a generation) in the genetic algorithm.

An extended version of the single-objective genetic algorithm in Ishibuchi et al. [7, 8] can be written as follows:

Step 0 (*Initialization*): Generate an initial population containing N_{pop} strings where N_{pop} is the number of strings in each population. In this operation, each string S is generated by assigning 0 to dummy rules and randomly assigning 1 or -1 to each of the other rules with the probability of 0.5.

Step 1 (*Rule elimination*): Classify all the given training patterns by linguistic classification rules included in each string S. Exclude non-active rules from S. That is, if a linguistic classification rule in S is not used for classifying any pattern, that rule is excluded from S. This rule elimination procedure is applied to all strings in the current population. Thus, every string consists of only active rules after this rule elimination procedure.

Step 2 (*Selection*): Select $\frac{1}{2}N_{\text{pop}}$ pairs of strings from the current population. The selection probability $P(S)$ of a string S in a population Ψ is specified as

$$P(S) = \frac{f(S) - f_{\min}(\Psi)}{\sum_{S \in \Psi}\{f(S) - f_{\min}(\Psi)\}}, \quad (14)$$

where

$$f_{\min}(\Psi) = \min\{f(S) \mid S \in \Psi\}. \quad (15)$$

Step 3 (*Crossover*): For each selected pair, randomly choose bit positions. Each bit position is chosen with probability 0.5. Interchange the bit values at the chosen positions in the selected pair.

Step 4 (*Mutation*): For each bit value of the generated strings by the crossover operation, apply the following mutation operation:

$$s_r = 1 \to s_r = -1 \quad \text{with probability } P_{\text{m}}(1 \to -1),$$

$$s_r = -1 \to s_r = 1 \quad \text{with probability } P_{\text{m}}(-1 \to 1).$$

Step 5 (*Elitist strategy*): Randomly remove one string from the N_{pop} strings generated by the above operations, and add the string with the maximum fitness value in the previous population to the current one.

Step 6 (*Termination test*): If a pre-specified stopping condition is not satisfied return to Step 1. The total number of generations is used as a stopping condition in this paper.

The rule elimination procedure in Step 1 is added to the genetic algorithm in our former work [7, 8]. The crossover operation in Step 3 was called the uniform crossover in Syswerda [31]. In Step 4, different mutation probabilities $P_{\text{m}}(1 \to -1)$ and $P_{\text{m}}(-1 \to 1)$ are assigned to the mutations from 1 to -1 and from -1 to 1, respectively. A larger probability is usually assigned to $P_{\text{m}}(1 \to -1)$ than to $P_{\text{m}}(-1 \to 1)$ in order to reduce the number of linguistic classification rules in each individual.

The genetic algorithm was applied to the classification problem in Fig. 3 with the following parameter specifications:

Weights in the fitness function: $W_{NCP} = 5$, $W_S = 1$,

Population size: $N_{\text{pop}} = 20$,

Crossover probabilities: $P_{\text{m}}(1 \to -1) = 0.1$,
$P_{\text{m}}(-1 \to 1) = 0.001$,

Stopping condition: 1000 generations.

First we generated $6^2 = 36$ linguistic classification rules corresponding to the fuzzy partitions in Fig. 5. Then the genetic algorithm was applied to the rule selection problem for selecting a small number of significant rules from the generated 36 rules. By the genetic algorithm, the following three linguistic classification rules were selected:

If x_{p1} is *don't care* and x_{p2} is *don't care*

then Class 1 with $CF = 0.19$,

If x_{p1} is *don't care* and x_{p2} is *large*

then Class 2 with $CF = 0.84$,

If x_{p1} is *medium* and x_{p2} is *don't care*

then Class 2 with $CF = 0.75$.

The classification boundary obtained by the selected three linguistic classification rules is shown in Fig. 7. From Fig. 7, we can see that all the given patterns are correctly classified by the selected rules.

Since the grade of certainty of the first linguistic classification rule is very small (i.e., 0.19), this rule is employed in the classification of a new pattern only when other rules do not have large grades of compatibility to the new pattern. Therefore we have the following classification rule from the above three rules by ignoring "don't care" attributes.

If x_1 is *medium* or x_2 is *large*

then Class 2, else Class 1.

From the configuration of the given patterns in Fig. 7, we can see that this classification rule agrees with our intuitive recognition of the given patterns.

We also applied the genetic algorithm to the well-known iris data (see, for example, [8]) for selecting linguistic classification rules. The pattern classification problem of the iris data is a three-class problem with four attributes. In each class, 50 patterns are given (total 150 patterns) as training patterns. Since the iris data has four attributes, $6^4 = 1296$ linguistic classification rules were generated as candidate rules. Thus our rule selection problem is to find a compact rule set from the 1296 rules. The total number of possible rule sets is $2^{1296} \cong 1.36 \times 10^{390}$.

By the genetic algorithm with the same parameter specifications as the above computer simulation, five linguistic rules in Fig. 8 were selected. The last column (# of patterns) in Fig. 8 shows the number of training patterns that were correctly classified by each rule. Therefore we can see that 147 patterns (98% of the given 150 patterns) are correctly classified by the selected five rules. By ignoring "don't care" attributes denoted by rectangles in Fig. 8, we have the following linguistic classification rules from the selected rules:

If x_3 is *small* then Class 1 with $CF = 1.00$,

If x_3 is *medium* and x_4 is *medium*

then Class 2 with $CF = 0.95$,

If x_2 is *medium small* and x_4 is *medium large*

then Class 3 with $CF = 0.78$,

If x_1 is *medium* and x_2 is *medium* and x_4 is large

then Class 3 with $CF = 1.00$,

If x_1 is *large* and x_2 is *medium*

then Class 3 with $CF = 1.00$.

3.2. *Searching for non-dominated solutions using variable weights*

In the single-objective genetic algorithm described in the last subsection, the weights W_{NCP} and

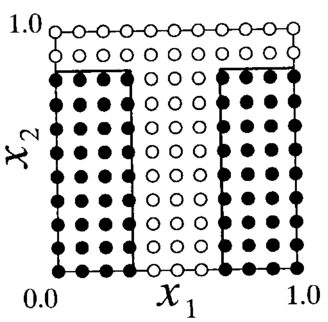

Fig. 7. Classification boundary by the selected three rules.

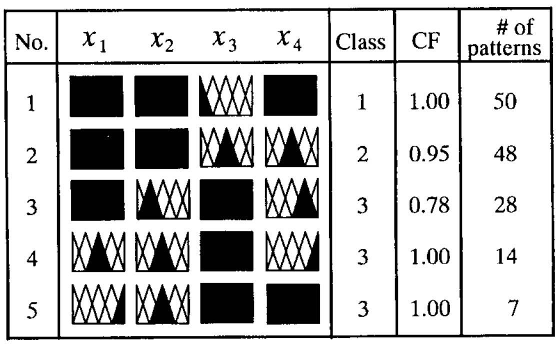

Fig. 8. Selected linguistic classification rules for the iris data.

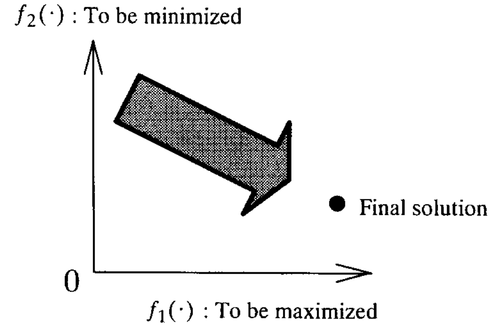

Fig. 9. Search direction of the single-objective genetic algorithm.

W_S were constant. Thus the search direction of the genetic algorithm was fixed as shown in Fig. 9. This means that the choice of the weight values in (12) has a significant effect on the final solution (i.e., selected linguistic classification rules) obtained by the genetic algorithm. Because the importance of each objective in the rule selection problem depends on the preference of human users, it is not easy to assign constant values to the weights W_{NCP} and W_S in advance.

The basic approach to multi-objective optimization problems is to try to find not a single solution but a set of non-dominated solutions. The final solution should be determined by decision makers (i.e., human users in our rule selection problem) from the non-dominated solutions depending on their preference. Thus we propose several methods for searching for the non-dominated solutions of the rule selection problem.

One simple method for searching for non-dominated solutions is to employ variable weights. That is, the execution of the single-objective genetic algorithm is repeated using various values of the weights W_{NCP} and W_S. The single-objective genetic algorithm was applied to the iris data with the same parameter specification as in the last subsection except for the weight values. The following ten pairs of the weight values were employed:

$$(W_{NCP}, W_S) = (0.1, 1), (0.5, 1), (1, 1), (5, 1), (10, 1), (50, 1), (100, 1), (500, 1), (1000, 1), (5000, 1).$$

Table 1
Solutions obtained by the single-objective genetic algorithm with various weight values

W_{NCP}	W_S	$NCP(S)$	$\|S\|$
0.1	1	142	3
0.5	1	146	4
1	1	147	5
5	1	147	5
10	1	147	7
50	1	147	5
100	1	147	6
500	1	146	4
1000	1	146	4
5000	1	146	4

The single-objective genetic algorithm described in the last subsection was applied to the iris data using each pair of the weight values. From these ten trials, ten solutions in Table 1 were obtained. From Table 1, we can see that the following solution are non-dominated:

$$\{(NCP(S), |S|)\} = \{(142, 3), (146, 4), (147, 5)\}.$$

The final solution should be selected from these three non-dominated solutions by human users depending on their preference.

3.3. Introducing a constraint condition on the number of rules

We can also search for the non-dominated solutions of the rule selection problem by introducing a constraint condition on the number of rules (i.e., a constraint condition on $|S|$). For example, if we want to maximize the number of correctly classified training patterns (i.e., to maximize $NCP(S)$) using five linguistic classification rules at best, the rule selection problem can be written as

$$\text{Maximize} \quad NCP(S), \tag{16}$$

$$\text{subject to} \quad |S| \leqslant 5, \tag{17}$$

$$S \subseteq S_{\text{ALL}}. \tag{18}$$

We formulate the following fitness function by introducing a large penalty when the constraint condition (17) is not satisfied:

$$f(S) = W_{NCP} \cdot NCP(S) - W_S \cdot \max\{0, |S| - 5\}, \quad (19)$$

where the weights W_{NCP} and W_S are specified as $W_{NCP} \ll W_S$ in order to attach a large penalty to the fitness function when the constraint condition (17) is not satisfied. Using different values in the right-hand side of the constraint condition (17), we can search for the non-dominated solution of the rule selection problem in (10), (11).

Let N_{rule} be the right-hand side constant of the constraint condition (17), then we have the following fitness function:

$$f(S) = W_{NCP} \cdot NCP(S) - W_S \cdot \max\{0, |S| - N_{\text{rule}}\}. \quad (20)$$

The non-dominated solutions of the rule selection problem in (10), (11) can be obtained using this fitness function with various values of N_{rule}.

The single-objective genetic algorithm described in Subsection 3.1 was applied to the iris data using each of the following ten values of N_{rule}:

$$N_{\text{rule}} = 3, 4, 5, 6, 7, 8, 9, 10, 11, 12.$$

By the ten trials of the genetic algorithm with $W_{NCP} = 1$ and $W_S = 100$, ten solutions in Table 2 were obtained. We can see that the following solutions are non-dominated in Table 2.

$$\{(NCP(S), |S|)\} = \{(142, 3), (146, 4), (147, 5)\}.$$

Table 2
Solutions obtained by the single-objective genetic algorithm with a constraint condition on the number of selected linguistic classification rules

Constraint	$NCP(S)$	$\lvert S\rvert$
$\lvert S\rvert \leqslant 3$	142	3
$\lvert S\rvert \leqslant 4$	146	4
$\lvert S\rvert \leqslant 5$	147	5
$\lvert S\rvert \leqslant 6$	147	6
$\lvert S\rvert \leqslant 7$	147	5
$\lvert S\rvert \leqslant 8$	147	8
$\lvert S\rvert \leqslant 9$	147	6
$\lvert S\rvert \leqslant 10$	147	5
$\lvert S\rvert \leqslant 11$	146	6
$\lvert S\rvert \leqslant 12$	146	10

3.4. *Introducing a constraint condition on the number of correctly classified patterns*

In the last subsection, we introduce a constraint condition on the number of selected linguistic classification rules. In a similar manner, we can introduce a constraint condition on the number of correctly classified training patterns. Let us assume that the number of correctly classified training patterns should be larger than or equal to N_{pattern} (e.g., $N_{\text{pattern}} = 145$ in the application to the iris data). In this case, our rule selection problem can be written as

$$\text{Minimize} \quad |S|, \quad (21)$$

$$\text{subject to} \quad NCP(S) \geqslant N_{\text{pattern}}, \quad (22)$$

$$S \subseteq S_{\text{ALL}}. \quad (23)$$

We formulate the following fitness function by introducing a large penalty when the constraint condition (22) is not satisfied:

$$f(S) = -W_{NCP} \cdot \max\{0, N_{\text{pattern}} - NCP(S)\} - W_S \cdot |S|, \quad (24)$$

where the weights W_{NCP} and W_S are specified as $W_{NCP} \gg W_S$ in order to attach a large penalty to the fitness function when the constraint condition (22) is not satisfied. The non-dominated solutions of the rule selection problem in (10), (11) can be obtained using this fitness function with various values of N_{pattern}.

The single-objective genetic algorithm described in Subsection 3.1 was applied to the iris data using each of the following ten values of N_{pattern}:

$$N_{\text{pattern}} = 141, 142, 143, 144, 145, 146, 147, 148, 149, 150.$$

By the ten trials of the genetic algorithm with $W_{NCP} = 100$ and $W_S = 1$, ten solutions in Table 3 were obtained. We can see that the following solutions are non-dominated in this table.

$$\{(NCP(S), |S|)\} = \{(142, 3), (146, 4), (147, 6)\}.$$

Table 3
Solutions obtained by the single-objective genetic algorithm with a constraint condition on the number of correctly classified training patterns

Constraint	$NCP(S)$	$\|S\|$
$NCP(S) \geqslant 141$	141	3
$NCP(S) \geqslant 142$	142	3
$NCP(S) \geqslant 143$	144	4
$NCP(S) \geqslant 144$	144	4
$NCP(S) \geqslant 145$	145	4
$NCP(S) \geqslant 146$	146	4
$NCP(S) \geqslant 147$	147	6
$NCP(S) \geqslant 148$	147	6
$NCP(S) \geqslant 149$	147	6
$NCP(S) \geqslant 150$	147	6

4. Two-objective genetic algorithm for the rule selection problem

In the last section, we have proposed three methods for searching for the non-dominated solutions of the rule selection problem by the single-objective genetic algorithm. The single-objective genetic algorithm was repeated with different parameter specifications (e.g., different weight values) in each method. In this section, we propose a multi-objective genetic algorithm for searching for the non-dominated solutions more directly.

A rule set S is treated as a string $S = s_1 s_2 \dots s_N$ in the multi-objective genetic algorithm as in the single-objective algorithm described in the last section. Crossover and mutation operators in the multi-objective genetic algorithm are also the same as those of the single-objective algorithm. Our multi-objective genetic algorithm differs from the single-objective algorithm in its selection procedure and elitist strategy. In our multi-objective genetic algorithm, the selection probability $P(S)$ in (14) is determined by the fitness function $f(S)$ in (12) with randomly specified weight values. That is, when a pair of parent strings are selected, the values of the weights W_{NCP} and W_S are assigned as

$$W_{NCP}\text{: a random real number in } [0, 1], \tag{25}$$

$$W_S\text{: } W_S = 1 - W_{NCP}. \tag{26}$$

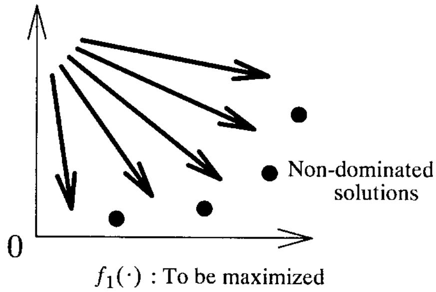

Fig. 10. Search direction of the multi-objective genetic algorithm.

The random weight values are given by (25), (26) for each selection of a pair of parent strings. That is, because $\frac{1}{2}N_{\text{pop}}$ pairs of parent strings are selected in each generation, $\frac{1}{2}N_{\text{pop}}$ values are randomly specified to each weight in each generation. Thus we can see that the selection procedure in each generation of our multi-objective genetic algorithm drives the search of the algorithm in various directions in Fig. 10.

In the execution of the multi-objective genetic algorithm, a tentative set of non-dominated solutions is externally preserved. This means that there are two sets of strings in each generation: one is a current population and the other is a tentative set of non-dominated solutions. A certain number of strings (say, N_{elite} strings) are randomly selected from the tentative set of non-dominated solutions, and the selected strings are added to the current population as elite solutions in our multi-objective genetic algorithm.

Our multi-objective genetic algorithm can be written as follows:

Step 0 (*Initialization*): Generate an initial population containing N_{pop} strings in the same manner as the single-objective genetic algorithm in the last section.

Step 1 (*Rule elimination*): Classify all the given training patterns by linguistic classification rules included in each string S. Exclude non-active rules from S. This rule elimination procedure is applied to all strings in the current population.

Step 2 (*Evaluation*): Calculate the values of the two objectives $NCP(S)$ and $|S|$ for the generated strings. Update the tentative set of non-dominated solutions.

Step 3 (*Selection*): Calculate the fitness value of each string using random weight values in (25), (26). Select a pair of strings from the current population according to the selection probability $P(S)$ in (14). This procedure is repeated for selecting $\frac{1}{2}N_{\text{pop}}$ pairs of parent strings.

Step 4 (*Crossover*): For each selected pair, apply the uniform crossover operation to generate two strings in the same manner as the single-objective genetic algorithm.

Step 5 (*Mutation*): For each bit value of the generated strings by the crossover operation, apply the mutation operation in the same manner as the single-objective genetic algorithm.

Step 6 (*Elitist strategy*): Randomly remove N_{elite} strings from the generated N_{pop} strings, and add N_{elite} strings that are randomly selected from the tentative set of non-dominated solutions.

Step 7 (*Termination test*): If a pre-specified stopping condition is not satisfied, return to Step 1.

We applied the proposed multi-objective genetic algorithm to the iris data. In order to compare the multi-objective genetic algorithm with the single-objective algorithm in the last section under the same computation load, the execution of the multi-objective algorithm was repeated ten times. The number of elite solutions N_{elite} was specified as $N_{\text{elite}} = 3$. By the ten trials of the multi-objective algorithm, the following non-dominated solutions were obtained:

$$\{(NCP(S), |S|)\} = \{(0,0), (50,1), (100,2), (142,3), (146,4), (147,5), (148,6)\}. \quad (27)$$

Here we summarize the non-dominated solutions obtained by each method in the last section (see Tables 1–3):

(1) By the method based on variable weights in Subsection 3.2:

$$\{(NCP(S), |S|)\} = \{(142,3), (146,4), (147,5)\}. \quad (28)$$

(2) By the method based on the constraint condition $|S| \leqslant N_{\text{rule}}$ in Subsection 3.3:

$$\{(NCP(S), |S|)\} = \{(142,3), (146,4), (147,5)\}. \quad (29)$$

(3) By the method based on the constraint condition $NCP(S) \geqslant N_{\text{pattern}}$ in Subsection 3.4:

$$\{(NCP(S), |S|)\} = \{(142,3), (146,4), (147,6)\}. \quad (30)$$

From the comparison of the result in (27) by the multi-objective algorithm with these results in (28)–(30) by the single-objective algorithm, we can see that a slightly better set of non-dominated solutions was obtained by the multi-objective genetic algorithm. For example, a rule set that can correctly classify 148 patterns was not found by any method based on the single-objective genetic algorithm in the last section (see (28)–(30)).

5. Extension to a hybrid algorithm

As Nozaki et al. [23] demonstrated, the classification performance of a fuzzy classification system can be improved by adjusting the grade of certainty CF_j of each linguistic classification rule. In this section, we propose a hybrid algorithm by combining the learning method of CF_j in Nozaki et al. [23] with the multi-objective genetic algorithm in the last section.

5.1. Learning method

From the fuzzy reasoning procedure for classifying a pattern $x_p = (x_{p1}, \ldots, x_{pn})$ in Subsection 2.4, we can see that $\boldsymbol{x}_p$ is classified by a linguistic classification rule $R_{\hat{j}}$ that satisfies the following relation:

$$\mu_{\hat{j}}(\boldsymbol{x}_p) \cdot CF_{\hat{j}} = \max\{\mu_j(\boldsymbol{x}_p) \cdot CF_j \mid \text{Rule } R_j \in S\}. \quad (31)$$

If the consequent class $C_{\hat{j}}$ of this rule is the same as the actual class of $\boldsymbol{x}_p$, $\boldsymbol{x}_p$ is correctly classified, otherwise $\boldsymbol{x}_p$ is misclassified.

When $\boldsymbol{x}_p$ is correctly classified by the linguistic classification rule $R_{\hat{j}}$, the grade of certainty $CF_{\hat{j}}$ of this rule is increased as the reward of the correct classification [23]:

$$CF_{\hat{j}}^{\text{new}} = CF_{\hat{j}}^{\text{old}} + \eta_1 \cdot (1 - CF_{\hat{j}}^{\text{old}}), \quad (32)$$

where η_1 is a positive learning constant for increasing the grade of certainty. On the contrary, when $\boldsymbol{x}_p$ is misclassified by the linguistic classification rule $R_{\hat{j}}$, the grade of certainty $CF_{\hat{j}}$ of this rule is decreased as the punishment of the misclassification [23]:

$$CF_j^{\text{new}} = CF_j^{\text{old}} - \eta_2 \cdot CF_j^{\text{old}}, \tag{33}$$

where η_2 is a positive learning constant for decreasing the grade of certainty.

5.2. Hybrid algorithm

The learning method of the grade of certainty CF_j is combined with our multi-objective genetic algorithm. Since the learning method is applicable to any rule set S, we apply it to all the rule sets (i.e., all the strings) generated by the crossover and mutation operations in the multi-objective genetic algorithm. That is, the following procedure is inserted between Step 6 and Step 7 of the multi-objective genetic algorithm described in Section 5:

Step 6.5 (*Learning*): Apply the learning method to each rule set S generated by the crossover and mutation operations. The learning for each rule set S is iterated N_{learning} times for all the training patterns.

5.3. Simulation result

The proposed hybrid algorithm was applied to the iris data using the same parameter specifications as the multi-objective genetic algorithm in Section 4. The learning rates η_1 and η_2 were specified as $\eta_1 = 0.001$ and $\eta_2 = 0.1$. We examined four specifications of N_{learning}, i.e., $N_{\text{learning}} = 0, 1, 2, 10$. Table 4 shows non-dominated solutions by ten trials of the hybrid algorithm with each specification of N_{learning}. For example, we can see from Table 4 that the following non-dominated solutions were obtained by specifying N_{learning} as $N_{\text{learning}} = 10$:

$$\{(NCP(S), |S|)\} = \{(0,0), (50,1), (100,2), (145,3), (147,4), (148,5)\}. \tag{34}$$

From Table 4, we can see that the classification performance of the selected linguistic rules was improved by combining the learning method into the multi-objective genetic algorithm. For example, three linguistic classification rules selected by the non-hybrid algorithm with no learning (i.e., $N_{\text{learning}} = 0$) correctly classified 142 patterns while 145 patterns were correctly classified by three rules selected by the hybrid algorithm with $N_{\text{learning}} = 2$ and $N_{\text{learning}} = 10$. In Fig. 11, we show rule sets with three linguistic classification rules obtained by the non-hybrid algorithm. The five rule sets in Fig. 11, which has the same classification performance (i.e., which can correctly classify 142 patterns), were obtained by the ten trials of the non-hybrid algorithm. On the other hand, three rule sets with three linguistic classification rules that can correctly classify 145 patterns were obtained by the ten trials of

Table 4
Solutions obtained by the hybrid algorithm with various specifications of the number of iterations of the learning method (i.e., N_{learning}). The non-hybrid multi-objective genetic algorithm corresponds to the case of $N_{\text{learning}} = 0$. "*" denotes that a non-dominated solution with the corresponding number of selected rules was not obtained

The number of selected rules: $\|S\|$	The number of correctly classified training patterns: $NCP(S)$			
	$N_{\text{learning}} = 0$	$N_{\text{learning}} = 1$	$N_{\text{learning}} = 2$	$N_{\text{learning}} = 10$
0	0	0	0	0
1	50	50	50	50
2	100	100	100	100
3	142	143	145	145
4	146	147	147	147
5	147	*	148	148
6	148	148	149	*
7	*	149	*	*

No.	x_1 x_2 x_3 x_4	Class	CF	# of patterns
1		1	1.00	50
2		2	0.95	43
3		3	0.57	49

No.	x_1 x_2 x_3 x_4	Class	CF	# of patterns
1		1	1.00	50
2		2	0.79	47
3		3	0.70	45

No.	x_1 x_2 x_3 x_4	Class	CF	# of patterns
1		1	1.00	50
2		2	0.83	47
3		3	0.59	45

No.	x_1 x_2 x_3 x_4	Class	CF	# of patterns
1		1	1.00	50
2		2	0.79	44
3		3	0.59	48

No.	x_1 x_2 x_3 x_4	Class	CF	# of patterns
1		1	1.00	50
2		2	0.79	47
3		3	0.70	45

Fig. 11. Rule sets obtained by the non-hybrid algorithm.

No.	x_1 x_2 x_3 x_4	Class	CF	# of patterns
1		1	1.00	50
2		2	0.42	47
3		3	0.14	48

No.	x_1 x_2 x_3 x_4	Class	CF	# of patterns
1		1	1.00	50
2		2	0.42	47
3		3	0.14	48

No.	x_1 x_2 x_3 x_4	Class	CF	# of patterns
1		1	1.00	50
2		2	0.42	47
3		3	0.14	48

Fig. 12. Rule sets obtained by the hybrid algorithm.

the hybrid algorithm (Fig. 12) with $N_{\text{learning}} = 10$. The first rule set in each figure consists of the same three linguistic classification rules except for their grades of certainty (i.e., CF in each figure).

6. Conclusions

In this paper, we proposed genetic-algorithm-based methods for constructing a compact fuzzy classification system with a small number of linguistic classification rules. We first formulated a rule selection problem of linguistic classification rules with two objectives: to maximize the number of correctly classified training patterns and to minimize the number of selected rules. Then we proposed three methods based on a single-objective genetic algorithm for finding a set of non-dominated solutions of the rule selection problem. We also proposed a method based on a multi-objective genetic algorithm. By computer simulations on the iris data, we showed that a slightly better set of non-dominated solutions was obtained by the multi-objective genetic algorithm than the single-objective genetic algorithm. Finally we proposed a hybrid algorithm by combining a learning method of linguistic classification rules with the multi-objective genetic algorithm. By computer simulations on the iris data, we showed that the combination of the learning algorithm had an effect on improving the classification performance of selected rules.

When we construct a compact fuzzy classification system with high classification performance for a specific pattern classification problem, first a set of non-dominated solution of the rule selection problem are found by the proposed methods, then one of the non-dominated solutions is chosen depending on the preference of human users. The proposed methods can be also viewed as knowledge acquisition tools because classification knowledge is automatically extracted from numerical data as a small number of linguistic classification rules. Because the number of selected rules by the proposed methods is small, human users can carefully examine all the selected rules. If hundreds of rules are selected, it is practically impossible for human users to examine all the selected rules carefully. In the proposed methods, human users can also easily understand each rule because the selected rules are linguistic rules. This clarity of the selected rules is the main advantage of the proposed rule selection methods.

Acknowledgement

This research was partially supported by the Grant-in-Aid for International Scientific Research (No. 06045047) from the Ministry of Education, Science and Culture in Japan.

References

[1] D.S. Feldman, Fuzzy network synthesis with genetic algorithms, *Proc. 5th ICGA* (University of Illinois at Urbana-Champaign, 1993) 312–317.

[2] R.A. Fisher, The use of multiple measurements in taxonomic problems, *Ann. Eugenics* **7** (1936) 179–188.

[3] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning* (Addison-Wesley, Reading, MA, 1989).

[4] F. Herrera, M. Lozano and J.L. Verdegay, Tuning fuzzy logic controllers by genetic algorithms, *Internat. J. Approx. Reasoning* **12** (1995) 299–315.

[5] J.H. Holland, *Adaptation in Natural and Artificial Systems* (University of Michigan Press, Ann Arbor, Michigan, 1975).

[6] A. Homaifar and E. McCormick, Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms, *IEEE Trans. on Fuzzy Systems* **3** (1995) 129–139.

[7] H. Ishibuchi, T. Murata and H. Tanaka, Construction of fuzzy classification systems with linguistic if-then rules using genetic algorithms, in: S.K. Pal and P.P. Wang, Eds., *Genetic Algorithms for Pattern Recognition* (CRC Press, Boca Raton, FL, USA, 1996) 227–251.

[8] H. Ishibuchi, T. Murata and I.B. Türkşen, A genetic-algorithm-based approach to the selection of linguistic classification rules, *Proc. EUFIT'95* (Aachen, Germany, 1995) 1414–1419.

[9] H. Ishibuchi, T. Murata and I.B. Türkşen, Selecting linguistic classification rules by two-objective genetic algorithms, *Proc. IEEE-SMC'95* (Vancouver, Canada, 1995) 1410–1415.

[10] H. Ishibuchi, T. Nakashima and T. Murata, A fuzzy classifier system that generates fuzzy if-then rules for pattern classification problems, *Proc. IEEE-ICEC'95* (Perth, Australia, 1995) 759–764.

[11] H. Ishibuchi, K. Nozaki and H. Tanaka, Distributed representation of fuzzy rules and its application to pattern classification, *Fuzzy Sets and Systems* **52** (1992) 21–32.

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

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

[14] H. Ishigami, T. Fukuda, T. Shibata and F. Arai, Structure optimization of fuzzy neural network by genetic algorithm, *Fuzzy Sets and Systems* **71** (1995) 257–264.

[15] C.L. Karr, Design of an adaptive fuzzy logic controller using a genetic algorithm, *Proc. 4th ICGA* (San Diego, CA, USA, 1991) 450–457.

[16] C.L. Karr and E.J. Gentry, Fuzzy control of pH using genetic algorithms, *IEEE Trans. Fuzzy Systems* **1** (1993) 46–53.

[17] J. Kinzel, F. Klawonn and R. Kruse, Modifications of genetic algorithms for designing and optimizing fuzzy controllers, *Proc. IEEE-ICEC'94* (Orlando, FL, USA, 1994) 28–33.

[18] K. Kropp and U.G. Baitinger, Optimization of fuzzy logic controller inference rules using a genetic algorithm, *Proc. EUFIT'93* (Aachen, Germany, 1993) 1090–1096.

[19] C.C. Lee, Fuzzy logic in control systems: Fuzzy logic controller, *IEEE Trans. Systems Man Cybernet.* **20** (1990) 404–435.

[20] M.A. Lee and H. Takagi, Integrated design stages of fuzzy systems using genetic algorithms, *Proc. FUZZ-IEEE'93* (San Francisco, CA, USA, 1993) 612–617.

[21] J. Liska and S.S. Melsheimer, Complete design of fuzzy logic systems using genetic algorithms, *Proc. FUZZ-IEEE'94* (Orlando, FL, USA, 1994) 1377–1382.

[22] H. Nomura, I. Hayashi and N. Wakami, A self-tuning method of fuzzy reasoning by genetic algorithm, *Proc. 1992 Internat. Fuzzy Systems and Intelligent Control Conference* (Louisville, KY, USA, 1992) 236–245.

[23] K. Nozaki, H. Ishibuchi and H. Tanaka, Adaptive fuzzy-rule-based classification systems, *IEEE Trans. on Fuzzy Systems*, to appear.

[24] D. Park, A. Kandel and G. Langholz, Genetic-based new fuzzy reasoning models with application to fuzzy control, *IEEE Trans. on Systems Man Cybernet.* **24** (1994) 39–47.

[25] A. Parodi and P. Bonelli, A new approach to fuzzy classifier systems, *Proc. 5th ICGA* (University of Illinois at Urbana-Champaign, IL, USA, 1993) 223–230.

[26] A. Satyadas and K. Krishnakumar, GA-optimized fuzzy controller for spacecraft attitude control, *Proc. FUZZ-IEEE'94* (Orlando, FL, USA, 1994) 1979–1984.

[27] K. Shimojima, T. Fukuda and Y. Hasegawa, Self-tuning fuzzy modeling with adaptive membership function, rules, and hierarchical structure based on genetic algorithm, *Fuzzy Sets and Systems* **71** (1995) 295–309.

[28] M. Sugeno, An introductory survey of fuzzy control, *Inform. Sci.* **36** (1985) 59–83.

[29] M. Sugeno and T. Yasukawa, A fuzzy-logic-based approach to qualitative modeling, *IEEE Trans. on Fuzzy Systems* **1** (1993) 7–31.

[30] H. Surmann, A. Kanstein and K. Goser, Self-organizing and genetic algorithms for an automatic design of fuzzy control and decision systems, *Proc. EUFIT'93* (Aachen, Germany, 1993) 1097–1104.

[31] G. Syswerda, Uniform crossover in genetic algorithms, *Proc. 3rd ICGA* (George Mason University, U.S.A., 1989) 2–9.

[32] T. Takagi and M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, *IEEE Trans. Systems Man Cybernet.* **15** (1985) 116–132.

[33] P. Thrift, Fuzzy logic synthesis with genetic algorithms, *Proc. 4th ICGA* (San Diego, CA, USA, 1991) 509–513.

[34] M. Valenzuela-Rendon, The fuzzy classifier system: a classifier system for continuously varying variables, *Proc. 4th ICGA* (San Diego, CA, USA, 1991) 346–353.

[35] L.X. Wang and J.M. Mendel, Generating fuzzy rules by learning from examples, *IEEE Trans. Systems Man Cybernet.* **22** (1992) 1414–1427.