Available online at www.sciencedirect.com
ScienceDirect
International Journal of Approximate Reasoning
48 (2008) 526–543
INTERNATIONAL JOURNAL OF APPROXIMATE REASONING
www.elsevier.com/locate/ijar

# Fuzzy classifier identification using decision tree and multiobjective evolutionary algorithms

Pietari Pulkkinen *, Hannu Koivisto

*Institute of Automation and Control, Tampere University of Technology, P.O. Box 692, FIN-33101 Tampere, Finland*

Received 29 November 2006; received in revised form 3 October 2007; accepted 8 October 2007
Available online 17 October 2007

**Abstract**

This paper presents a hybrid method for identification of Pareto-optimal fuzzy classifiers (FCs). In contrast to many existing methods, the initial population for multiobjective evolutionary algorithms (MOEAs) is neither created randomly nor a priori knowledge is required. Instead, it is created by the proposed two-step initialization method. First, a decision tree (DT) created by C4.5 algorithm is transformed into an FC. Therefore, relevant variables are selected and initial partition of input space is performed. Then, the rest of the population is created by randomly replacing some parameters of the initial FC, such that, the initial population is widely spread. That improves the convergence of MOEAs into the correct Pareto front. The initial population is optimized by NSGA-II algorithm and a set of Pareto-optimal FCs representing the trade-off between accuracy and interpretability is obtained. The method does not require any a priori knowledge of the number of fuzzy sets, distribution of fuzzy sets or the number of relevant variables. They are all determined by it. Performance of the obtained FCs is validated by six benchmark data sets from the literature. The obtained results are compared to a recently published paper [H. Ishibuchi, Y. Nojima, Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning, International Journal of Approximate Reasoning 44 (1) (2007) 4–31] and the benefits of our method are clearly shown.


*Keywords:* Fuzzy classifiers (FCs); Multiobjective evolutionary algorithms (MOEAs); Decision trees (DTs); Initialization

## 1. Introduction

Fuzzy classifiers (FCs) with if-then rules are related to the way human beings think and that is their main advantage over black-box models, such as neural networks. Identification of FCs involves determining the adequate structure and parameters. The structure identification consists of several tasks, such as, selecting the adequate variables, assigning the adequate number of fuzzy sets to each variable and defining the number of fuzzy rules used. In addition to that, the parameters of fuzzy sets need to be specified as well. It was illus-

* Corresponding author. Tel.: +358 3 3115 2655; fax: +358 3 3115 2340.
*E-mail addresses:* pietari.pulkkinen@tut.fi (P. Pulkkinen), hannu.koivisto@tut.fi (H. Koivisto).

0888-613X/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.ijar.2007.10.004

trated in [1], that such a task is highly complex due to its enormous search space, especially when high-dimensional problems are covered.

Grid-type partitioning is a way to reduce the complexity of the identification problem. In that approach, the number of fuzzy sets assigned to each variable is fixed to some number and also the parameters of fuzzy sets are predefined. However that approach suffers from the curse of dimensionality, that is, the number of fuzzy rules is exponentially increased when the dimensionality of the problem is increased. To overcome that problem [2] applied grid-type partitioning with "don't care" linguistic values and selected only the relevant rules out of the all possible rules. The benefit of the approach is its simple implementation, because it does not modify the parameters of fuzzy sets. Nevertheless, it was stated in [3] that fuzzy sets are the major components of FCs, since they affect the accuracy of the model, interpretability of fuzzy rules and also the performance of the system. It was stated in [2], that homogenously assigned fuzzy sets are intuitive, therefore making the FCs more interpretable. However, often they do not represent the real distribution of the data and therefore the accuracy of the obtained FCs is degraded [4,3]. Moreover, the intuitiveness of the linguistic values is also deteriorated. To tackle that problem, the fuzzy sets can also be pre-specified by domain experts. However, when dealing with high-dimensional problems, domain experts will have problems in assigning the fuzzy sets for each variable. Therefore automatic tuning of the fuzzy sets is usually required.

Recently the goal in FC identification has been in obtaining accurate and interpretable FCs. Naturally accuracy and interpretability are conflicting objectives. For example, an FC with a vast rule-base may be accurate for training patterns, however, it lacks for interpretability and may not perform well on unseen samples due to the overfitting. Usually a trade-off between the accuracy and interpretability is sought using evolutionary algorithms (EAs) and often those approaches are called genetic fuzzy systems (GFS) [5]. A single trade-off solution can be found by aggregating multiple objectives (e.g. accuracy, number of rules and number of conditions) into a single fitness function and by setting the weights for each objective [6,7]. However, that requires work in choosing the appropriate weights, which may be different for each problem at hand. Moreover, it is not guaranteed that with every run a new solution is found [8]. Since multiobjective evolutionary algorithms (MOEAs) can find several widely spread Pareto-optimal solutions in a single run without assigning weight values for each objective, they are often preferred. After a set of solutions is obtained, advances and drawbacks of them can be considered and a solution can be selected based on the preferences.

When EAs are applied, the population needs to be initialized first. That can be done randomly or manually like in [9–11]. Adequate initialization, however, can improve the convergence of EAs [12,13]. Hence, it is beneficial to use, for example, decision tree (DT) or clustering algorithms to initialize the population [14,6,15–17]. Furthermore, if variable selection is applied during the initialization and only the relevant variables are used to form the fuzzy rules, EAs need to search the appropriate rules and parameters of fuzzy sets only for the reduced set of variables. That clearly reduces the search space of EAs.

As illustrated above, many FC identification methods using EAs have been developed. They, however, have some limitations, which are listed next. Some of the methods do not tune the fuzzy sets and require a priori knowledge of the distribution of the fuzzy sets [9,1,18]. Some of the approaches initializes the population randomly [10], which deteriorates the convergence. Also the variable selection in initialization phase is neglected in many approaches [10,9,1,19,15,11,17]. Moreover some approaches use aggregated fitness functions [20,16,6,7,21].

To the best of our knowledge, a method which initializes the population adequately (i.e. selects the relevant variables, creates the relevant initial rules and partitions the input space adequately), tunes the membership functions, and identifies a set of Pareto-optimal FCs has not been developed yet. This paper aims to fill that gap.

In this paper the initial population is created in two phases. First a DT is created by C4.5 algorithm [22]. Because of the rectangular decision boundaries of crisp DTs, they can be overly complex. FCs, however, can create non-axis parallel decision boundaries [23,24]. Therefore, DT is converted into an FC [6]. Because widely distributed initial population improves the convergence of MOEAs [12,13], the rest of the population is created by randomly replacing some parameters of the initial FC by random numbers, such that, the population is widely distributed. DT initialization was previously applied, for example, in [6,7]. However, in those approaches further optimization by EAs was performed using aggregated fitness functions and therefore a set of Pareto-optimal FCs was not obtained.

NSGA-II algorithm [8] is applied to optimize the initial population and to find a set of Pareto-optimal FCs. It was successfully applied in [25,1] for the same purpose. However, in this paper NSGA-II is also applied to fine-tune the parameters of fuzzy sets, not only to find the appropriate rules and rule conditions. Furthermore, only the relevant variables, selected by C4.5 algorithm, are used to form fuzzy rules.

The rest of this paper is organized as follows. Section 2 briefly represents the theory behind multiobjective problems (MOPs) and NSGA-II algorithm. Furthermore, FCs are introduced and the criteria defining their fitness is presented. Section 3 represents the proposed FC identification method. It introduces the proposed two-step initialization method and presents the coding of the FC into a chromosome, such that, NSGA-II algorithm can be applied. In Section 4 performance of our method is studied on six benchmark data sets and the obtained results are compared to the results in the literature. The results show that by the proposed method a compact set of high quality solutions is obtained. Finally, Section 5 concludes the paper.

## 2. Preliminaries

In this section a brief introduction to the theory of multiobjective problems (MOPs) is given first. Then, NSGA-II [8], a popular multiobjective evolutionary algorithm (MOEA) applied in this paper, is briefly presented. After that, the basic theory of fuzzy classifiers (FCs) is given. Finally, the fitness function applied in this paper is defined.

### 2.1. Multiobjective problems

Let us assume a MOP with $h$ objectives $f_i$, $i = 1,\ldots,h$. Let $\mathbf{s}$ be the decision vector and $\mathbf{S}$ the feasible region of the decision vector. That MOP can be formulated as:

$$\text{Minimize } f_1(\mathbf{s}), f_2(\mathbf{s}), \ldots, f_h(\mathbf{s}) \text{ subject to } \mathbf{s} \in \mathbf{S}. \tag{1}$$

It is often impossible to find a solution which simultaneously minimizes all $h$ objectives. Hence, a set of widely spread trade-off solutions is often sought. A particular interest is on the non-dominated (Pareto optimal) decision vectors. A decision vector $\mathbf{s}_1 \in \mathbf{S}$ is Pareto optimal, if there does not exist a decision vector $\mathbf{s}_2 \in \mathbf{S}$, which fulfills the following conditions:

$$\forall i, \quad f_i(\mathbf{s}_2) \leqslant f_i(\mathbf{s}_1) \quad \text{and} \quad \exists j, \quad f_j(\mathbf{s}_2) < f_j(\mathbf{s}_1). \tag{2}$$

If $\mathbf{s}_2$ meets the conditions in (2), it dominates $\mathbf{s}_1$. The Pareto-optimal set is formed of non-dominated solutions and their image under the objective functions is the Pareto front. [26,1].

### 2.2. Multiobjective evolutionary algorithms

MOEAs have been widely used to solve MOPs. Some of the application areas are, for example, the stochastic multiobjective environmental/economic dispatch problems [27] and scheduling of drilling operations [28]. Like mentioned earlier, they have also been used to design the fuzzy classifiers and function estimators.

NSGA-II [8] is a popular MOEA. It is a well-applicable algorithm, because it includes, for example, an efficient constraint-handling method, a fast non-dominated sorting procedure, an elitist approach and uses parameterless crowding distance measure to maintain the diversity of population. It is applied in this paper with polynomial mutation and simulated binary cross-over (SBX) [29] as genetic operators. The details of NSGA-II are not given in this paper but they can be found from [8]. Other good MOEAs are SPEA2 [30] and $\epsilon$-MOEA [31], just to mention a few.

### 2.3. Fuzzy classifiers

Fuzzy classification rules consist of fuzzy sets in the antecedent and a class label in the consequent. Let us denote the data set with $D$ data points and $n$ variables as $\mathbf{Z} = [\mathbf{X}\ \mathbf{y}]$, where input matrix $\mathbf{X}$ and output vector $\mathbf{y}$ are given as:

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,n} \\ x_{2,1} & x_{2,2} & \dots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{D,1} & x_{D,2} & \dots & x_{D,n} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_D \end{bmatrix}. \tag{3}$$

According to [6] fuzzy classification can be performed as follows:

$$R_i: \text{ If } x_1 \text{ is } A_{i,1} \dots \text{ and } x_n \text{ is } A_{i,n} \text{ then } g_i, \quad i = 1, \dots, R, \tag{4}$$

where $R$ is the number of rules, $A_{i,j}, j = 1, \dots, n$ is a membership function, $g_i \in \{1, \dots, C\}$ is the rule consequent and $C$ is the number of different classes in data set. For each data point $\mathbf{x}_k$, the degree of fulfillment of a rule is computed as:

$$\beta_i(\mathbf{x}_k) = \prod_{j=1}^{n} A_{i,j}(x_{k,j}). \tag{5}$$

The rule with the highest degree of fulfillment is declared as the winner rule (i.e. Winner takes all strategy). The output of the classifier is the rule consequent associated to that rule. There are also other types of fuzzy rules and $t$-norms which can be applied to reasoning [32] and the properties of fuzzy classifiers are discussed in detail in [33].

### 2.4. Fitness of a fuzzy classifier

Accuracy of FCs is measured by calculating the number of misclassifications. However, there is no generic way to measure the interpretability of FCs [34]. Often the interpretability is measured by calculating the number of rules and the total number of antecedents in the rules (total rule length) [18]. It was stated in [1], that the number of rules together with the total rule length can prevent overfitting. Consequently, it is beneficial to use both of those objectives. So in this paper, the objectives to be minimized are the number of misclassifications, the number of rules and the total rule length.

## 3. Proposed hybrid fuzzy classifier identification method

This section introduces the hybrid fuzzy classifier (FC) identification method, which is based on decision tree (DT) and multiobjective evolutionary algorithms (MOEA). When any standard MOEA is applied, the first step is the creation of the initial population. In contrast to many existing methods, the initial population is not created randomly or based on a priori knowledge, but by a two-step initialization method. First, an FC is identified using C4.5 algorithm. Then, to improve the convergence of EAs, the rest of the population is created by randomly replacing some parameters of that FC such a way, that the initial population is widely distributed. Finally, NSGA-II algorithm is applied to optimize the initial population and a set of non-dominated solutions is obtained. The proposed method is summarized in Fig. 1.

The rest of this section is organized as follows. First, FC initialization by C4.5 algorithm is discussed. Then, coding of an FC into a chromosome is presented and an illustrative example is given. Finally, it is introduced how the rest of the population is created in a way that the initial population is widely spread.

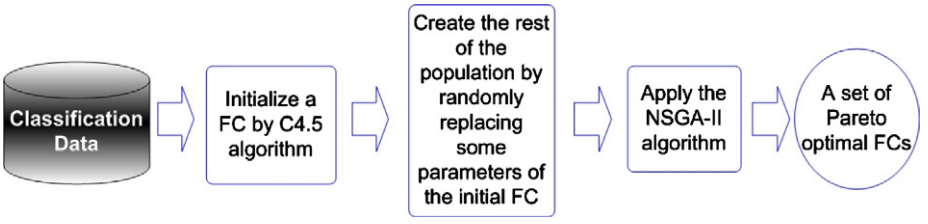


Fig. 1. Proposed hybrid fuzzy classifier identification method.

### 3.1. Initialization of FCs

First C4.5 algorithm is applied to create a decision tree (DT). C4.5 was selected as a part of initialization, because it is a top rated DT algorithm [35], it can select the relevant variables and partition the input space [6]. DT is then converted into an FC like shown in [6]. That can be done without decomposition error, if trapezoidal membership functions (MFs) are used, such that, they present the crisp decision boundaries of DT [6]. In this paper, however, the crisp decision boundaries are softened by using generalized bell (gbell) MFs:

$$f(x;a,b,c)=\frac{1}{1+\left|\frac{x-c}{a}\right|^{2b}}, \tag{6}$$

where $x$ is the data point, and $a$, $b$ and $c$ are the parameters of a gbell MF. The value of $b$ defines the fuzziness of a MF. If it is set to a high value, say more than 100, then a MF is very close to a crisp function. Therefore, in this paper $0 < b < 10$.

The value of $a$ is restricted as:

$$\max(0, a_{\text{initial}} * (1-\alpha)) < a < a_{\text{initial}} * (1+\alpha),$$

where $a_{\text{initial}}$ denotes the value of $a$, when a DT is converted into an FC. Value of $\alpha = 1/M_j$ defines how much parameters can vary around their initial values [14]. $M_j$ stands for the maximum number of fuzzy sets assigned to a variable $j$ and equals to the number of fuzzy sets in variable $j$ in initial FC.

The value of $c$, which defines the center of gbell MF, is restricted as:

$$\max(c_{\text{initial}} - \alpha * \chi, \text{lbound}) < c < \min(c_{\text{initial}} + \alpha * \chi, \text{ubound}),$$

where 'ubound' and 'lbound' are respectively the upper and lower bounds of a variable and $\chi = \text{ubound}-\text{lbound}$ denotes the range of a variable.

There are also other reasons for applying gbell MFs instead of trapezoidal MFs. Gbell MFs may have better fit to the data [36] and they have three parameters in contrast to four parameters of trapezoidal MFs. Furthermore, the parameters of gbell MFs can be optimized independently, which is not the case when trapezoidal MFs are used. Therefore, standard mutation and cross-over operators of NSGA-II algorithm can be used without the need to make sure that, for example, parameter $b$ is greater than parameter $a$. The decomposition error caused by transformation of trapezoidal MFs into gbell MFs can be usually overcome by EA optimization [7].

### 3.2. Structure of a chromosome

Each individual (chromosome) contains an FC. Their structure is coded as a real coded vector including antecedents of the rules **A** and parameters of the fuzzy sets **P**. Antecedent vector **A** is defined as:

$$\mathbf{A} = (A^*_{1,1}, A^*_{1,2}, \ldots, A^*_{1,n_s}, A^*_{2,1}, A^*_{2,2}, \ldots, A^*_{2,n_s}, \ldots, A^*_{R,1}, A^*_{R,2}, \ldots, A^*_{R,n_s}), \tag{7}$$

where $R$ denotes the number of rules in initial FC and $n_s$ stands for the number of variables selected from $n$ variables by C4.5 algorithm. Naturally $n_s \leqslant n$, but usually $n_s < n$. Since real coding of the variables is used for all parameters, the integers $A_{i,j} = \{0, 1, \ldots, M_j\}$, indicating which membership function is used for variable $j$ in rule $i$, are coded as real coded values $A^*_{i,j}$, which are rounded to the nearest integer when fitness evaluation is performed. Therefore, $A_{i,j} = \text{round}(A^*_{i,j})$, where $-0.5 < A^*_{i,j} < M_j + 0.5$.

During MOEA optimization, number of rules, rule conditions and variables can be decreased. If variable $j$ is not used in rule $i$, then $A_{i,j} = 0$. If rule $i$ is not used in an FC, then $\forall j$, $A_{i,j} = 0$. If variable $j$ is not used in an FC, then $\forall i$, $A_{i,j} = 0$. In this paper it is required that each chromosome has at least one antecedent and one rule (i.e. $\exists i, \exists j, A_{i,j} \neq 0$).

Parameter vector **P** is given as:

$$\mathbf{P} = (P_{1,1}, P_{1,2}, \ldots, P_{1,\beta}, P_{2,1}, P_{2,2}, \ldots, P_{2,\beta}, \ldots, P_{\gamma,1}, P_{\gamma,2}, \ldots, P_{\gamma,\beta}), \tag{8}$$

where $\gamma$ is the number of parameters used to define a membership function and $\beta = \sum_{j=1}^{n_s} M_j$ is the total number of fuzzy sets in initial FC. In this paper Gbell membership functions are used, so $\gamma = 3$.

Consequent part of the fuzzy rule $\mathbf{g} = (g_1, \ldots, g_R)$ is not included in an individual. It is static and created in initialization phase by C4.5 algorithm. So, NSGA-II is used to select rules, rule antecedents and parameters of membership functions for the pre-specified class labels. The total number of parameters $\theta$ to be optimized by NSGA-II algorithm is therefore given as:

$$\theta = R \times n_s + \gamma \times \beta. \tag{9}$$

Each parameter is restricted with lower and upper bounds defined in current and previous subsections. Therefore the number of constrains is $2 \times \theta$.

### 3.3. Coding of a chromosome: an example

Let us consider a classification problem with 5 classes and 4 variables. Let us assume that an FC with 5 rules and 5 fuzzy sets has been created by transforming a DT into an FC. C4.5 algorithm has selected 2 variables, $x_1$ and $x_2$, assigned 3 fuzzy sets to variable $x_1$ and 2 fuzzy sets to variable $x_2$. The obtained rules with total rule length of 9 are the following:

*Rule$_1$: If $x_1$ is 1 and $x_2$ is 1 then Class is 5*
*Rule$_2$: If $x_1$ is 1 and $x_2$ is 2 then Class is 4*
*Rule$_3$: If $x_1$ is 2 and $x_2$ is 1 then Class is 3*
*Rule$_4$: If $x_1$ is 2 and $x_2$ is 2 then Class is 2*
*Rule$_5$: If $x_1$ is 3 then Class is 1*

Coding of the antecedent part would be then:

$$\mathbf{A} = \left( \underbrace{1,1}_{Rule_1}, \underbrace{1,2}_{Rule_2}, \underbrace{2,1}_{Rule_3}, \underbrace{2,2}_{Rule_4}, \underbrace{3,0}_{Rule_5} \right).$$

Coding of the five membership functions would be:

$$\mathbf{P} = \left( \underbrace{P_{1,1}, P_{1,2}, P_{1,3}, P_{1,4}, P_{1,5}}_{\text{Gbell parameter } a}, \underbrace{P_{2,1}, P_{2,2}, P_{2,3}, P_{2,4}, P_{2,5}}_{\text{Gbell parameter } b}, \underbrace{P_{3,1}, P_{3,2}, P_{3,3}, P_{3,4}, P_{3,5}}_{\text{Gbell parameter } c} \right).$$

The rule consequents, which are the same for all individuals and not included in an individual are:

$$\mathbf{g} = \left( \underbrace{5}_{Rule_1}, \underbrace{4}_{Rule_2}, \underbrace{3}_{Rule_3}, \underbrace{2}_{Rule_4}, \underbrace{1}_{Rule_5} \right).$$

### 3.4. Initializing the rest of the population

The rest $N - 1$ chromosomes, where $N$ is the population size, are created by randomly replacing some parameters of the FC created by C4.5 algorithm in Section 3.1. The replacement algorithm creates a set of widely distributed chromosomes and it is given next:

**Repeat for** $I = 1, \ldots, N - 1$, where $I$ is the chromosome iterator.
*Step 1:* Calculate the number of parameters to be replaced $m$ as follows:

$$m = \text{round}\left( \frac{I}{(N-1)} \times \theta \right), \tag{10}$$

where 'round' stands for the operator rounding the result to the nearest integer.
*Step 2:* Choose randomly $m$ parameters out of $\theta$.

*Step 3:* Replace them by randomly generating $m$ parameters between their corresponding limits, defined in Sections 3.1 and 3.2.

**End for**

So the algorithm above generates widely distributed chromosomes, which all share the same structure, defined in Section 3.2. Some of the chromosomes are either very similar or very different to the chromosome generated by C4.5 algorithm, whereas some are between those extremes. That is important, because it speeds up the convergence of EAs to the correct Pareto front [13].

## 4. Experiments

In this section performance of the proposed identification framework is validated. First, the experimental setup used in this paper is described. Then, as illustrative examples, well-known Wine and Sonar classification data sets are studied. Finally, six benchmark data sets from UCI Machine Leaning Repository [37] are studied and a rigorous comparison of our results to the results presented in [1] is performed.

### 4.1. Experimental setup

Six benchmark data sets, Wisconsin breast cancer (Wisc), Pima Indians diabetes (Pima), Glass, Cleveland heart disease (Cleve), Sonar, and Wine were studied in this paper. These data sets represent problems with different number of classes, variables and data points (see Table 1) and they were also studied in [1]. Wisconsin breast cancer and Cleveland heart disease data sets contained data points with missing values. Those data points were removed.

In [1] a multiobjective fuzzy genetics-based machine learning (GBML) algorithm based on NSGA-II algorithm was applied to obtain a set of non-dominated fuzzy rule-based classifiers. The parameters of fuzzy sets were pre-specified by partitioning each input variable with 14 fuzzy sets and with a "don't care" value. So the problem was to specify the appropriate number of rules and to select the antecedents to these rules from those aforementioned 15 fuzzy sets. Variable selection was not applied before executing GBML algorithm, so for each rule $n$ antecedents need to be specified. However, GBML algorithm can remove variables by assigning "don't care" values for certain variables in all rules.

Three formulations of multiobjective optimization problems (MOPs) were applied in [1]. MOP-1 was used to maximize the accuracy and to minimize the number of rules. MOP-2 was applied to maximize the accuracy and to minimize the total number of conditions in rules. MOP-3 was used to maximize the accuracy, minimize the total number of conditions in rules and to minimize the number of rules. So MOP-3 uses the same fitness function as in this paper.

In [1], the number of generations and population size were set to 5000 and 200, respectively. Like illustrated in [1], the search space is enormous, even when the parameters of fuzzy sets are pre-specified. In our case, when the parameters of fuzzy sets need to optimized as well, it is beneficial to use larger population size [38]. However to perform a fair comparison, the number of fitness evaluations was limited to 1 000 000[1] (i.e. $5000 \times 200$). So in this paper the number of generations and population size were both set to 1000. The applied parameters for NSGA-II algorithm are shown in Table 2. The same cross-over and mutation probabilities and distribution indexes were also applied in [8]. For C4.5 algorithm, the pruning confidence value was set to 5 in order to reduce the complexity of the initial FCs. The rest of the parameters were kept as their default values defined in [22].

### 4.2. Illustrative examples

The purpose of this subsection is to shed light on the proposed method by identifying FCs for Wine and Sonar classification data sets (see also Table 1). Those data sets were selected as examples, because they

[1] It was, however, illustrated in [1], that better results can be obtained by increasing the number of fitness evaluations.

Table 1
Data sets used in this study

| Data | Variables | Data points | Classes |
|---|---|---|---|
| Wisconsin breast cancer | 9 | 683 | 2 |
| Pima Indians diabetes | 8 | 768 | 2 |
| Glass | 9 | 214 | 6 |
| Cleveland heart disease | 13 | 297 | 5 |
| Sonar | 60 | 208 | 2 |
| Wine | 13 | 178 | 3 |

Table 2
NSGA-II parameters

| | |
|---|---|
| Population size | 1000 |
| Number of generations | 1000 |
| Distribution index for mutation | 20 |
| Distribution index for cross-over | 20 |
| Cross-over probability | 0.9 |
| Mutation probability | $1/\theta$ |

represent problems with different complexity, that is, Wine data have a moderate number of variables (13 variables) and Sonar data have a high number of variables (60 variables).

*4.2.1. An illustrative example 1: wine classification*

The problem is to classify 3 types of wines based on 13 variables, labeled here as $x_1, x_2, \ldots, x_{13}$. There are total of 178 datapoints in the data set. To perform the experiments, the data set was randomly divided into training and testing set. Training set consisted of 80% of datapoints, that is, 142 datapoints and testing set the rest 20% of datapoints (36 datapoints).

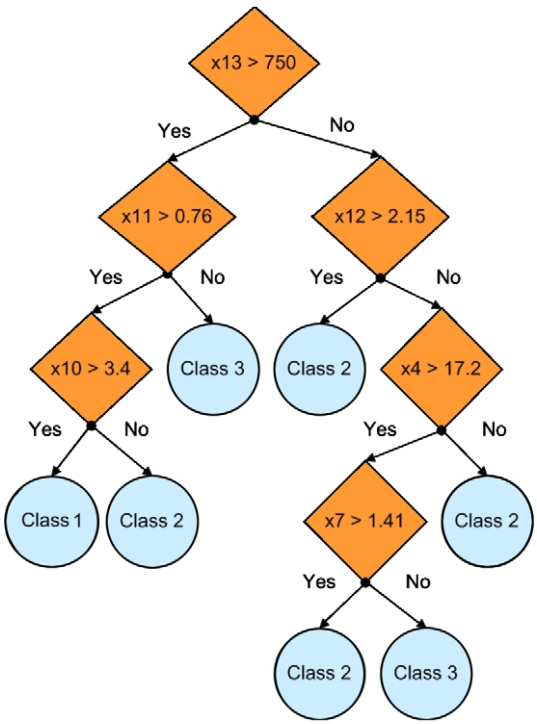


Fig. 2. Wine data: the obtained decision tree.

C4.5 algorithm was run and a decision tree (DT) was obtained (see Fig. 2). It was converted into an FC consisting of 7 rules and 21 rule conditions. The initial FC used six variables, $x_4$, $x_7$, $x_{10}$, $x_{11}$, $x_{12}$, and $x_{13}$, selected by C4.5. The fuzzy sets of the initial FC are shown in Fig. 3. Since each of the 6 variables are partitioned with 2 fuzzy sets, they can be labeled, for example, as small and large. Therefore, seven rules, generated by collecting all the conditions on the way from the root of the tree to each of the seven leaves of DT [6] can be expressed as:

*If* $x_{13}$ *is large and* $x_{11}$ *is small then Class is 3*
*If* $x_{13}$ *is large and* $x_{11}$ *is large and* $x_{10}$ *is large* then Class is 1
*If* $x_{13}$ *is large and* $x_{11}$ *is large and* $x_{10}$ *is small then Class is 2*
*If* $x_{13}$ *is small and* $x_{12}$ *is large then Class is 2*
*If* $x_{13}$ *is small and* $x_{12}$ *is small and* $x_4$ *is small then Class is 2*
*If* $x_{13}$ *is small and* $x_{12}$ *is small and* $x_4$ *is large and* $x_7$ *is large then Class is 2*
*If* $x_{13}$ *is small and* $x_{12}$ *is small and* $x_4$ *is large and* $x_7$ *is small then Class is 3*

After that, the rest of the population was created by randomly modifying some parameters of the initial FC, like illustrated in Subsection 3.4. The initial population is shown in Fig. 4.

Then, MOEA optimization was performed and a set of Pareto-optimal FCs was obtained. It can be seen from Fig. 4 that complexity of FCs was highly reduced due to MOEA optimization. That, however, did not deteriorate the accuracy. From the set of Pareto-optimal FCs, an FC can be selected based on the

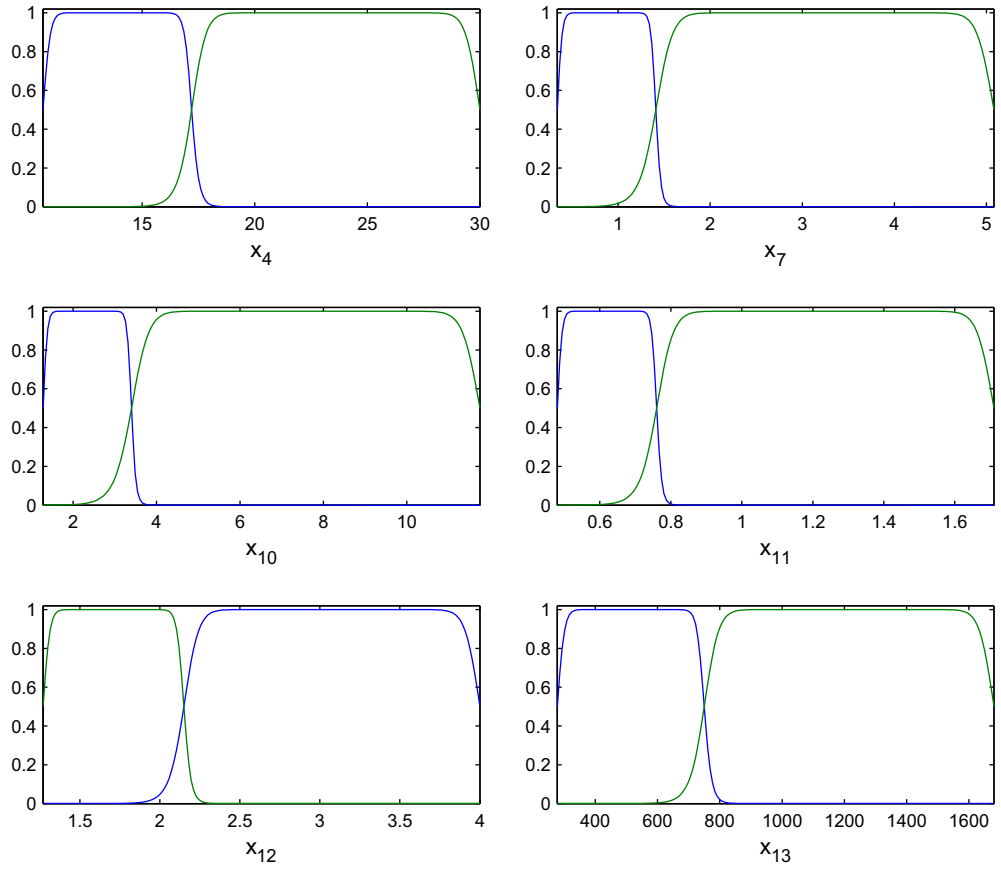


Fig. 3. Wine data: the fuzzy sets of the initial FC.

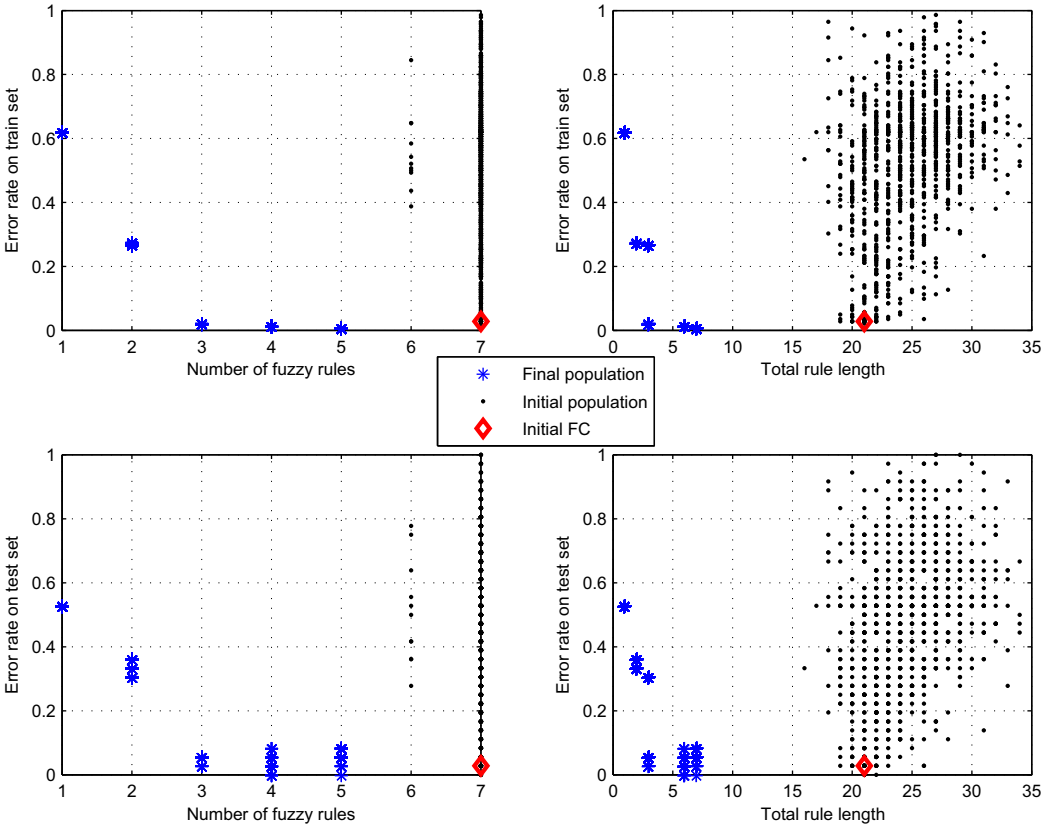


Fig. 4. Wine data: initial population and final population.

preferences. For example, an FC with 4 rules and 6 rule conditions can be selected. Its fuzzy sets are shown in Fig. 5 and its rules are the following:

*If* $x_7$ *is small then Class is 3*
*If* $x_{10}$ *is small then Class is 2*
*If* $x_7$ *is large and* $x_{13}$ *is small then Class is 2*
*If* $x_{10}$ *is large and* $x_{13}$ *is large then Class is 1*

The properties of the selected FC along with properties of DT and initial FC are shown in Table 3. It is seen that the selected FC is highly accurate, yet it is the most interpretable solution. From that Table it can be noticed that the training error of initial FC is slightly worse than training error of DT. That is due to the decomposition error caused when DT was converted into FC.

*4.2.2. An illustrative example 2: sonar classification*

Sonar data with 60 variables is studied in order to emphasize the variable selection capability of the proposed method. Sonar data consist of 111 and 97 patterns obtained by bouncing off the sonar signals from metal cylinders and rocks, respectively [39]. The problem is to distinguish between those signals based on 60 variables, named here $x_1, x_2, \ldots, x_{60}$.

The experiments were carried out exactly the same manner as in the previous example, but for the sake of brevity, DT and the fuzzy sets of initial and final FCs are not shown here. C4.5 selected 11 variables, namely $x_8$, $x_{11}$, $x_{13}$, $x_{36}$, $x_{45}$, $x_{46}$, $x_{51}$, $x_{53}$, $x_{54}$, $x_{59}$, $x_{60}$. The initial population and the final population after MOEA optimization are shown in Fig. 6. It is seen that even the dimensionality of the problem is high, the complexity

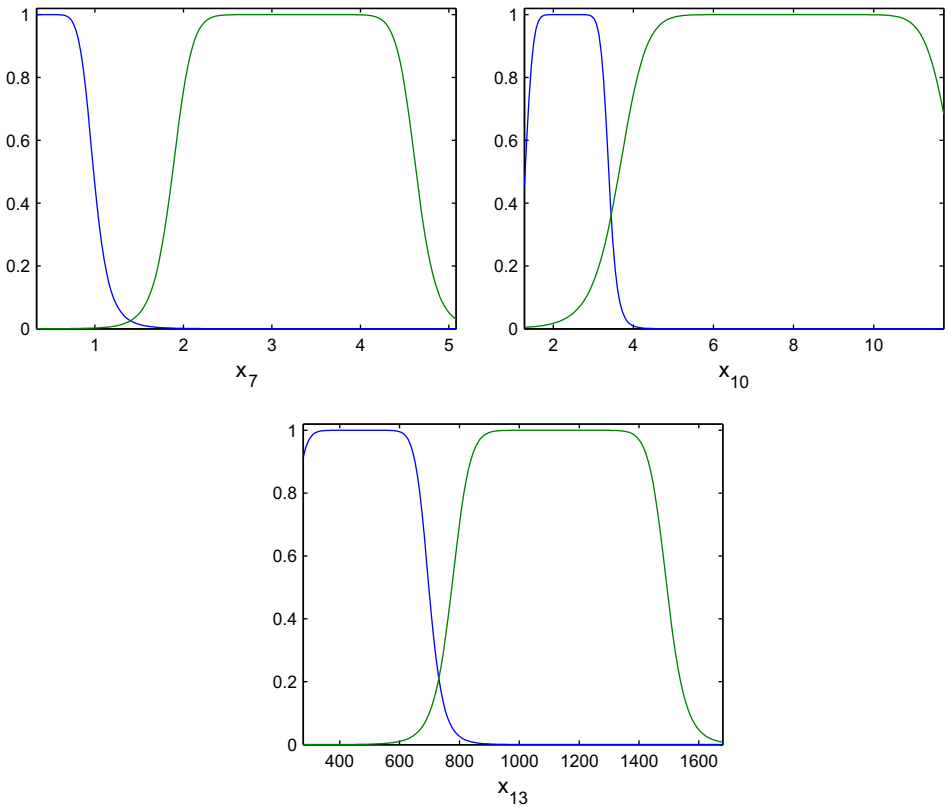


Fig. 5. Wine data: obtained fuzzy sets for the selected FC after MOEA optimization.

Table 3
Wine data: result comparison for decision tree, initial FC and selected FC

| Method | Train error rate | Test error rate | Rules | Total rule length | Variables |
|---|---|---|---|---|---|
| C4.5 | 0.0211 | 0.0278 | 7 | 21 | 6 |
| Initial FC | 0.0282 | 0.0278 | 7 | 21 | 6 |
| Selected FC | 0.0141 | 0 | 4 | 6 | 3 |

of the initial FC is still moderate, consisting of 12 rules and 49 rule conditions. In Table 4 properties of four FCs of final population are shown along with the initial FC and DT. Selected FC 1 is the most complex FC of the final population and it consists of 10 rules and 26 rule conditions in contrast to 12 rules and 49 rule conditions of the initial FC. Yet, selected FC 1 is more accurate than more complex initial FC. It is also noticed from Table 4 that the best testing accuracy is obtained when the number of rules and rule conditions are 3 and 4, respectively.

### 4.3. Results comparison

For each of the six data sets, described in Table 1, a 10-fold cross-validation (10-CV) [40,35] was performed 10 times (i.e. $10 \times 10$-CV). So the total number of runs for each data set was 100. A different random seed was used for each of the ten 10-CV runs. Since NSGA-II algorithm was applied, a set of non-dominated solutions was obtained for each run. Usually those sets are not identical. They may contain solutions with different structures (i.e. the number of rules and number of rule conditions) and the number of different solutions in

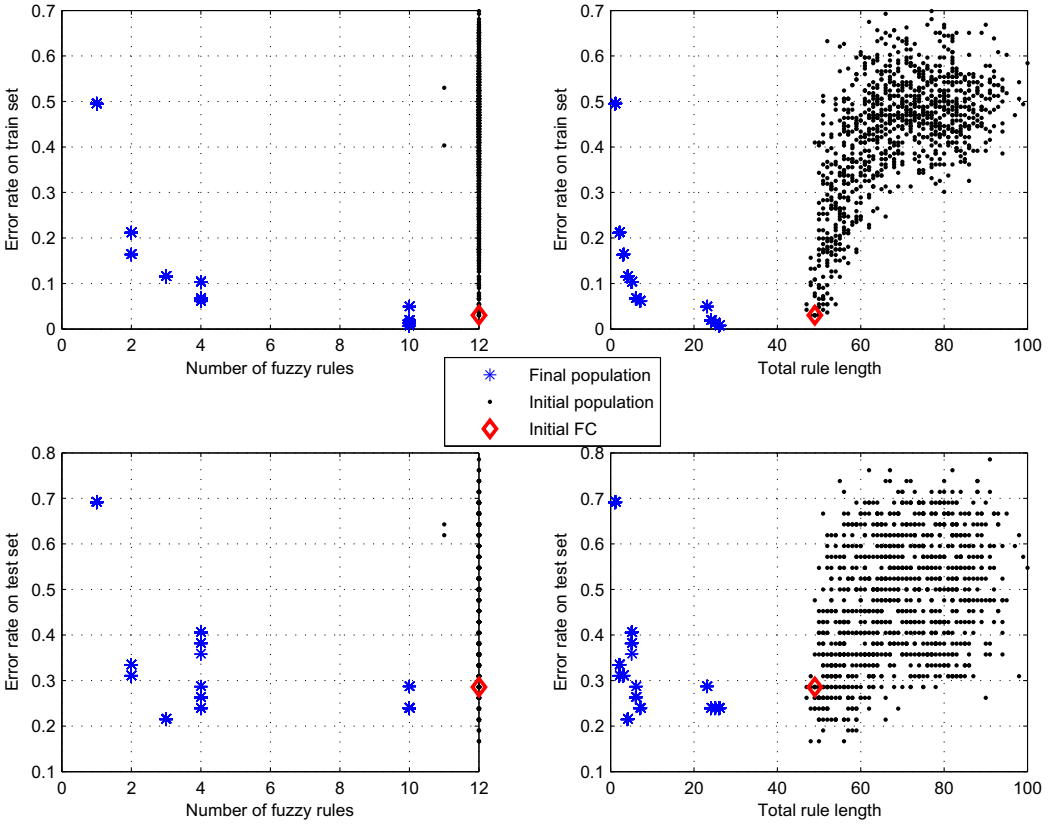


Fig. 6. Sonar data: initial population and final population.

Table 4
Sonar data: result comparison for decision tree, initial FC and some selected FCs

| Method | Train error rate | Test error rate | Rules | Total rule length | Variables |
|---|---|---|---|---|---|
| C4.5 | 0.0181 | 0.2857 | 12 | 49 | 11 |
| Initial FC | 0.0301 | 0.2857 | 12 | 49 | 11 |
| Selected FC 1 | 0.0060 | 0.2381 | 10 | 26 | 11 |
| Selected FC 2 | 0.0602 | 0.2381 | 4 | 7 | 6 |
| Selected FC 3 | 0.1145 | 0.2143 | 3 | 4 | 4 |
| Selected FC 4 | 0.1627 | 0.3095 | 2 | 3 | 3 |

a set may vary as well. In [1] only the solutions which were present in at least in 51 out of 100 runs were represented to have reliable results and we did that as well in order to perform a fair comparison. So the average error rates on test set over the number of runs the solution was present were calculated. In this paper, there may exist several solutions with the same structure and training error, but with different test error. That is possible because the parameters of fuzzy sets are not fixed. In those cases, the average of those test errors was selected to represent the solution for that run.

The solutions are presented in Figs. 7–12 along with the results of [1].[2] It is noted that all of our solutions for Wisconsin breast cancer, Glass and Wine data sets are not dominated by MOP-1, MOP-2 or MOP-3.

[2] The exact values to reconstruct the figures were kindly provided by Hisao Ishibuchi and Yusuke Nojima, the authors of that paper.

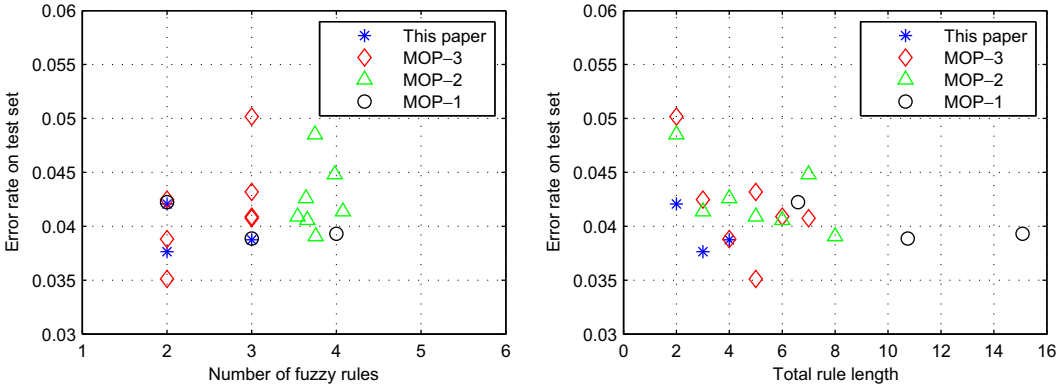


Fig. 7. Wisconsin breast cancer data.

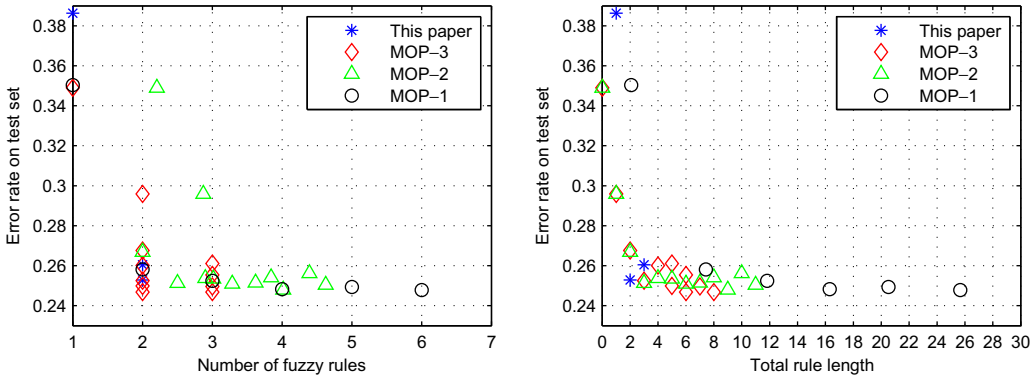


Fig. 8. Pima Indians diabetes data.

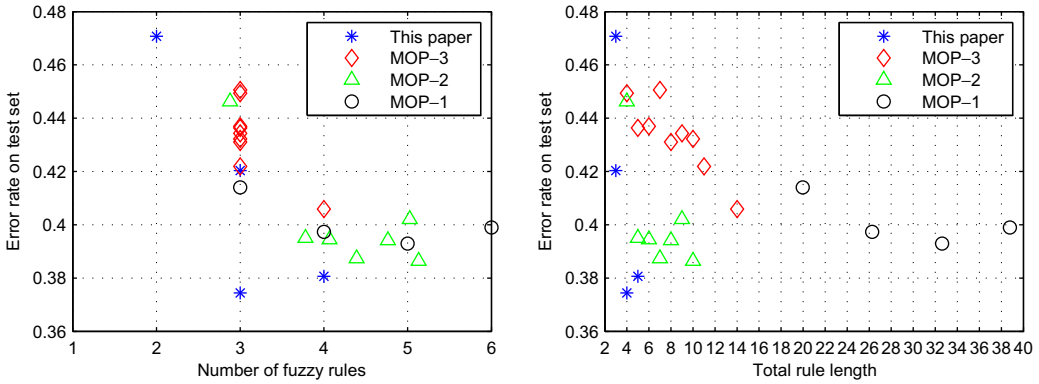


Fig. 9. Glass data.

Moreover, most of our solutions dominate some solutions of MOP-1, MOP-2 and MOP-3. For Pima Indians diabetes data, some of our solutions are dominated by MOP-3, but one of our solutions also dominates the solutions of MOP-1, MOP-2 and MOP-3. Our results for sonar data are quite similar to MOP-2 and MOP-3. Some of our solutions for Cleveland heart disease data clearly dominate the solutions of MOP-1, MOP-2 and MOP-3. However, one of our solutions with 1 rule and 1 rule condition is dominated by MOP-3.

Average best error rates for test and training sets over the 100 runs were calculated and presented in Tables 5 and 6. For comparison, the results of SOP-1, SOP-2 and SOP-3 [1], the single-objective versions of MOP-1,

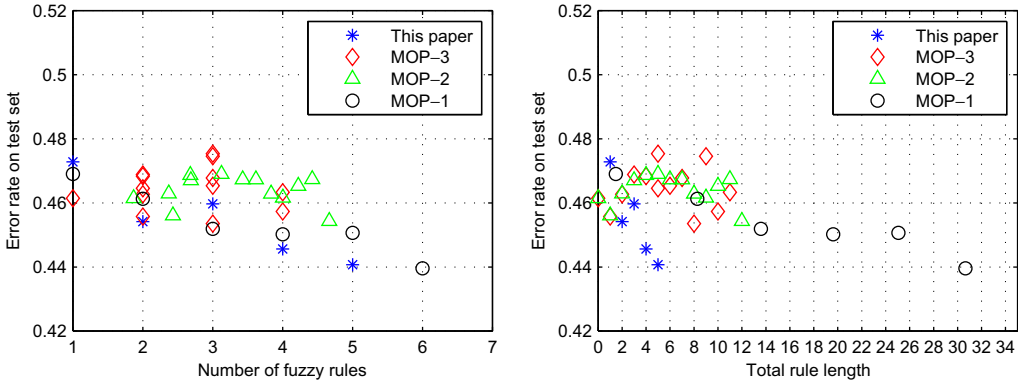


Fig. 10. Cleveland heart disease data.

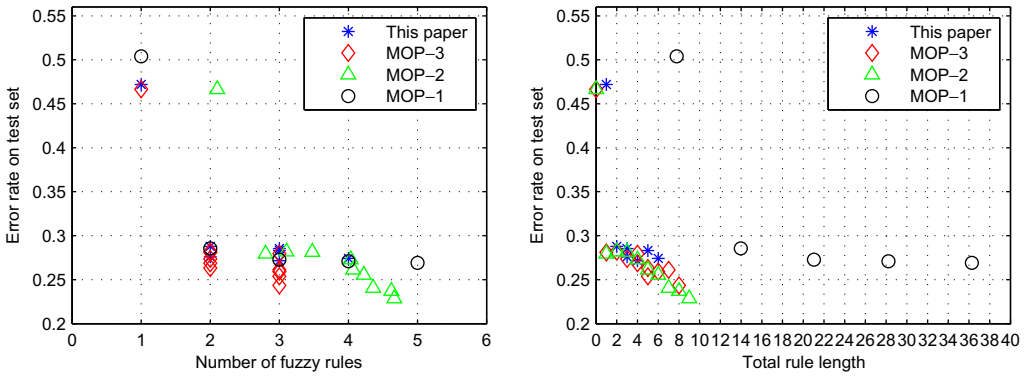


Fig. 11. Sonar data.

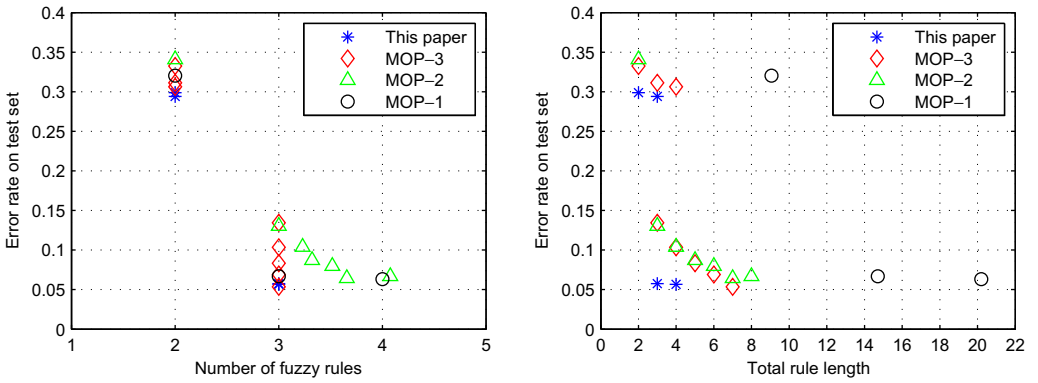


Fig. 12. Wine data.

MOP-2 and MOP-3, respectively, are included in those Tables. Moreover, the best results of six C4.5 variants, representing different splitting strategies and evaluation functions [41], are included in Table 6. Their performances were evaluated by $10 \times 10$-CV, which is the same experimental setup as in this paper.

It is noted from Table 5 that by means of our method the lowest training error rates were obtained for four out of six data sets. Furthermore, it can be seen from Table 6 that the lowest testing error rates for five

Table 5
Average best error rates on train set

| Data | This paper | MOP-1 | MOP-2 | MOP-3 | SOP-1 | SOP-2 | SOP-3 |
|---|---|---|---|---|---|---|---|
| Glass | **9.50** | 25.11 | 27.08 | 25.94 | 17.81 | 21.92 | 22.36 |
| Sonar | **1.95** | 8.55 | 8.69 | 8.42 | 3.55 | 5.89 | 5.82 |
| Wine | 0.6 | 0.01 | 0.10 | 0.03 | **0.00** | **0.00** | **0.00** |
| Cleve | **23.18** | 33.43 | 35.05 | 34.59 | 25.72 | 29.65 | 29.98 |
| Wisc | 1.94 | 1.59 | 1.71 | 1.74 | **1.08** | 1.44 | 1.51 |
| Pima | **16.17** | 19.48 | 19.79 | 19.59 | 17.74 | 18.37 | 18.41 |

Table 6
Average best error rates on test set

| Data | This paper | MOP-1 | MOP-2 | MOP-3 | SOP-1 | SOP-2 | SOP-3 | C4.5 in [41] |
|---|---|---|---|---|---|---|---|---|
| Glass | **24.05** | 35.55 | 33.93 | 34.05 | 35.76 | 39.21 | 38.36 | 27.3 |
| Sonar | **16.73** | 23.18 | 17.32 | 17.51 | 24.04 | 23.47 | 24.29 | 24.6 |
| Wine | **2.98** | 3.99 | 3.65 | 3.04 | 7.30 | 6.49 | 6.52 | 5.6 |
| Cleve | **38.22** | 42.57 | 42.85 | 42.64 | 44.83 | 45.80 | 45.44 | 46.3 |
| Wisc | 2.95 | 2.93 | 2.74 | **2.66** | 3.88 | 3.69 | 3.56 | 5.1 |
| Pima | **21.78** | 23.27 | 22.32 | 21.80 | 25.26 | 25.00 | 24.20 | 25.0 |

Table 7
Average number of obtained non-dominated FCs.

| Data | This paper | MOP-1 | MOP-2 | MOP-3 |
|---|---|---|---|---|
| Glass | 21.48 | 13.98 | 16.49 | **27.09** |
| Sonar | 16.84 | 10.01 | **20.47** | 17.66 |
| Wine | 6.03 | 11.45 | 9.96 | **11.81** |
| Cleveland | **23.35** | 11.56 | 22.17 | 18.59 |
| Wisconsin | 5.64 | 12.09 | **13.32** | 12.25 |
| Pima | 13.52 | 9.71 | 15.80 | **17.06** |

out of six data sets were also obtained by our method. That indicates good generalization capabilities of our method.

In Table 7 the average number of the obtained non-dominated solutions is presented. It is noted that generally our method obtained less non-dominated solutions than MOP-3, which has the same fitness function as our method. Only for one out of six data sets our method obtained more non-dominated solutions than MOP-3. That was due to the initialization algorithm used in this paper. Because C4.5 algorithm selected the relevant variables and created moderate number of rules and fuzzy sets, the number of possible solutions was reduced. However, as Tables 5 and 6 and Figs. 7–12 indicated, the quality of the obtained solutions was high.

A reader may wonder, that the number of obtained non-dominated FCs in Table 7 is higher than in Figs. 7–12. That is due to the experimental setup, which requires that an FC with a certain structure (i.e. certain number of rules and certain total rule length) must be present at least in 51 out of 100 runs. FCs which consist of few rules have less conceivable structures and therefore they are more likely to be presented in those aforementioned figures. To illustrate the effect of the experimental setup, Fig. 10 is constructed again, such that, one of the hundred runs is selected and the non-dominated solutions for that run are shown in Fig. 13. By comparing Figs. 10 and 13, it is seen that FCs with more than 5 rules are present in Fig. 13, but not in Fig. 10. A clear trade-off structure for train set is seen in Fig. 13. That is, however, not the case when test set is considered. Some of the more complex solutions show poor generalization capabilities due to the overfitting. That confirms again, that the number of rules together with total rule length can prevent overfitting, like illustrated in [1].

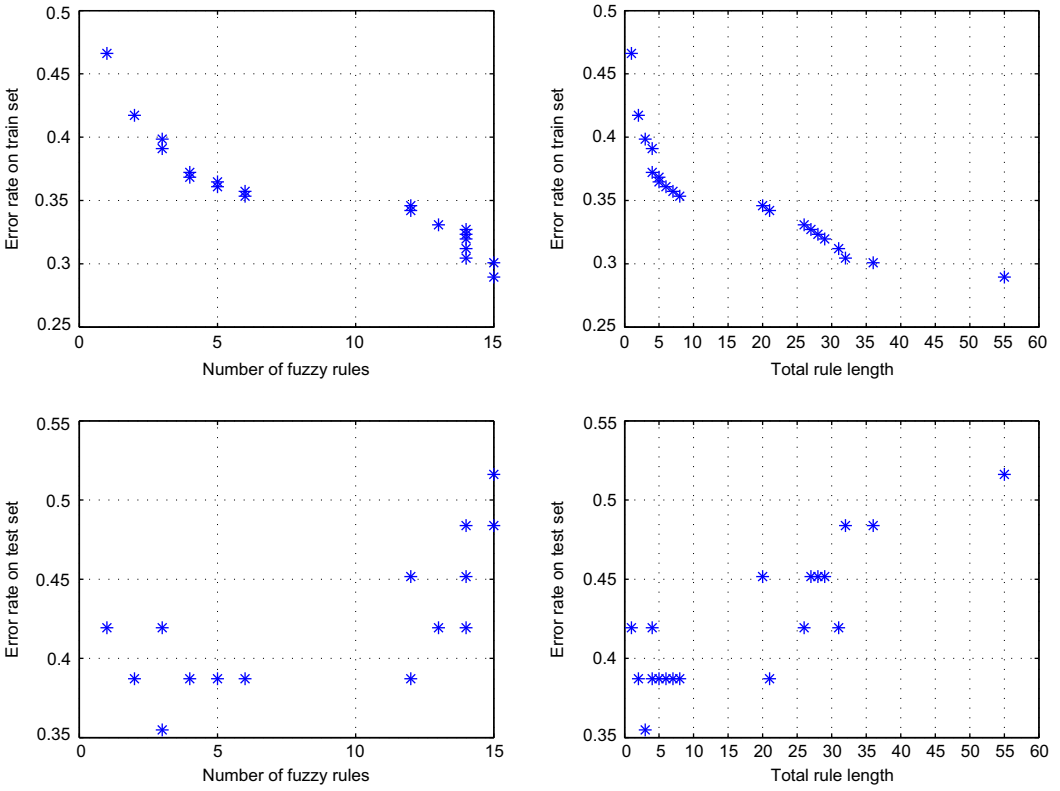


Fig. 13. Cleveland heart disease data: the results of one run, selected out of 100 runs. A clear trade-off structure is seen for train set, but that is not so clear for test set due to the overfitting.

## 5. Conclusions

This paper proposed a hybrid method for identification of Pareto-optimal fuzzy classifiers (FCs). In contrast to many existing methods, the initial population for the multiobjective evolutionary algorithms (MOEAs) was neither created randomly nor a priori knowledge was required. Instead of those techniques, a two-step initialization method was applied. First, an FC was obtained by transformation of a decision tree (DT) into an FC. Therefore, no a priori knowledge of the relevant variables, number of fuzzy sets or distribution of fuzzy sets was required. Then, the rest of the population was created by randomly replacing some parameters of that FC, in a way that the population was widely spread. That improved the convergence of MOEAs into the correct Pareto front.

FCs were coded in a way that a popular MOEA, named NSGA-II, could be used to select rules, rule antecedents and parameters of membership functions for the class labels specified by DT algorithm in initialization phase. Because the parameters of fuzzy sets were not static, it enabled us to approximate the distribution of data more accurately.

Number of misclassifications, number of rules and total rule length were used as objectives to be optimized. In the future, it can be considered, whether it is beneficial to use different objectives. For example, number of misclassifications could be replaced with the area under the receiver operating characteristic curve (AUC), which is useful when class distributions and misclassification costs are unknown [10]. Also the number of membership functions could be a object to be minimized. Those modifications to the fitness function can be easily done without affecting any other part of the proposed method. In the future it can also be considered whether MOEA should be used to modify the class labels of consequents as well. That may be useful in cases when genetic operators significantly modify the antecedents of the rules and therefore the class labels specified in initialization phase may not be adequate anymore.

The validity of the proposed method was confirmed through six well-known benchmark data sets from the literature. We compared our results to another FC identification method by Ishibuchi and Nojima [1], which also utilized NSGA-II algorithm. The number of obtained Pareto-optimal solutions by our method was usually lower than in the comparative study. That was due to the initialization algorithm used in this paper. Because C4.5 algorithm selected the relevant variables and created moderate number of rules and fuzzy sets, the number of possible Pareto-optimal solutions was reduced. However, the variable selection also reduced the computational costs significantly. Furthermore, the quality of the obtained solutions was higher than in the comparative study; in five out of six data sets, we obtained more accurate solutions. Moreover, in three data sets, none of our solutions were dominated by the solutions of the comparative study and some of our solutions dominated the solutions of that study.

## Acknowledgement

The authors want to express their gratitude to Hisao Ishibuchi and Yusuke Nojima for providing their results for results comparison. In addition, the comments of anonymous reviewers are greatly appreciated.

## References

[1] H. Ishibuchi, Y. Nojima, Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning, International Journal of Approximate Reasoning 44 (1) (2007) 4–31.
[2] H. Ishibuchi, T. Nakashima, T. Murata, Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems, IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics 29 (5) (1999) 601–618.
[3] H. Huang, M. Pasquier, C. Quek, Optimally evolving irregular-shaped membership function for fuzzy systems, in: IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada, 2006, pp. 11078–11085.
[4] Y. Chen, J. Wang, Kernel machines and additive fuzzy systems: classification and function approximation, in: The 12th IEEE International Conference on Fuzzy Systems, vol. 2, 2003, pp. 789–795.
[5] O. Cordón, F. Gomide, F. Herrera, F. Hoffmann, L. Magdalena, Ten years of genetic fuzzy systems: current framework and new trends, Fuzzy Sets and Systems 141 (1) (2004) 5–31.
[6] J. Abonyi, J.A. Roubos, F. Szeifert, Data-driven generation of compact, accurate, and linguistically-sound fuzzy classifiers based on a decision-tree initialization, International Journal of Approximate Reasoning 32 (1) (2003) 1–21.
[7] P. Pulkkinen, H. Koivisto, Identification of interpretable and accurate fuzzy classifiers and function estimators with hybrid methods, Applied Soft Computing 7 (2) (2007) 520–533.
[8] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197.
[9] H. Ishibuchi, Y. Nojima, I. Kuwajima, Fuzzy data mining by heuristic rule extraction and multiobjective genetic rule selection, in: 2006 IEEE International Conference on Fuzzy Systems, Vancouver BC, Canada, 2006, pp. 7824–7831.
[10] C. Setzkorn, R. Paton, On the use of multi-objective evolutionary algorithms for the induction of fuzzy classification rule systems, BioSystems 81 (2) (2005) 101–112.
[11] A.F. Gómez-Skarmeta, F. Jiménez, J. Ibáñez, Pareto-optimality in fuzzy modeling, in: 6th European Congress on Intelligent Techniques and Soft Computing EUFIT'98, Aachen, Germany, 1998, pp. 694–700.
[12] C. Haubelt, J. Gamenik, J. Teich, Initial population construction for convergence improvement of moeas., in: C.A.C. Coello, A.H. Aguirre, E. Zitzler (Eds.), EMO 2005, Lecture Notes in Computer Science, vol. 3410, 2005, pp. 191–205.
[13] S. Poles, Y. Fu, E. Rigoni, The effect of initial population sampling on the convergence of multi-objective genetic algorithms, in: MOPGP'06: 7th Int. Conf. on Multi-Objective Programming and Goal Programming, Loire Valley (City of Tours), France, 2006.
[14] H. Roubos, M. Setnes, Compact and transparent fuzzy models and classifiers through iterative complexity reduction, IEEE Transactions on Fuzzy Systems 9 (4) (2001) 516–522.
[15] H. Wang, S. Kwong, Y. Jin, W. Wei, K. Man, Multi-objective hierarchical genetic algorithm for interpretable fuzzy rule-based knowledge extraction, Fuzzy Sets and Systems 149 (1) (2005) 149–186.
[16] Z.-Y. Xing, Y.-L. Hou, Y. Zhang, L.-M. Jia, Y. Hou, A multi-objective cooperative coevolutionary algorithm for constructing accurate and interpretable fuzzy systems, in: 2006 IEEE International Conference on Fuzzy Systems, Vancouver BC, Canada, 2006, pp. 6964–6970.
[17] Z.-Y. Xing, Y. Zhang, Y.-L. Hou, L.-M. Jia, On generating fuzzy systems based on pareto multi-objective cooperative coevolutionary algorithm, International Journal of Control, Automation, and Systems 5 (4) (2007) 444–455.
[18] H. Ishibuchi, T. Nakashima, T. Murata, Three-objective genetics-based machine learning for linguistic rule extraction, Information Sciences 136 (1–4) (2001) 109–133.
[19] J. Casillas, O. Cordón, M. del Jesus, F. Herrera, Genetic tuning of fuzzy rule deep structures preserving interpretability and its interaction with fuzzy rule set reduction, IEEE Transactions on Fuzzy Systems 13 (1) (2005) 13–29.

[20] A.G.D. Nuovo, V. Catania, An efficient approach for the design of transparent fuzzy rule-based classifiers, in: 2006 IEEE International Conference on Fuzzy Systems BC, Canada, 2006, pp. 6941–6947.
[21] M.-S. Kim, C.-H. Kim, J.-J. Lee, Evolving compact and interpretable takagi-sugeno fuzzy models with a new encoding scheme, IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics 36 (5) (2006) 1006–1023.
[22] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufman Publishers, 2929 Campus Drive, Suite 260 San Mateo, CA 94403, 1993.
[23] F. Hoppner, F. Klawonn, R. Kruse, T. Runkler, Fuzzy Cluster Analysis – Methods for Classification, Data Analysis and Image Recognition, John Wiley and Sons, 1999.
[24] H. Ishibuchi, T. Nakashima, Effect of rule weights in fuzzy rule-based classification systems, IEEE Transactions on Fuzzy Systems 9 (4) (2001) 506–515.
[25] H. Ishibuchi, T. Yamamoto, Effects of three-objective genetic rule selection on the generalization ability of fuzzy rule-based systems, in: C.M. Fonseca, P.J. Fleming, E. Zitzler, K. Deb, L. Thiele (Eds.), EMO 2003, Lecture Notes in Computer Science, vol. 2632, Springer, 2003, pp. 608–622.
[26] C.C. Coello, 20 years of evolutionary multiobjective optimization: What has been done and what remains to be done, in: G.Y. Yen, D.B. Fogel (Eds.), Computational Intelligence: Principles and Practice, IEEE Computational Intelligence Society, 2006, pp. 73–88.
[27] R.T.F.A. King, H.C.S. Rughooputh, K. Deb, Stochastic evolutionary multiobjective environmental/economic dispatch, in: IEEE Congress on Evolutionary Computation, Vancouver BC, Canada, 2006, pp. 946–953.
[28] P.-C. Chang, J.-C. Hsieh, C.-Y. Wang, Adaptive multi-objective genetic algorithms for scheduling of drilling operation in printed circuit board industry, Applied Soft Computing 7 (3) (2007) 800–806.
[29] K. Deb, R.B. Agrawal, Simulated binary crossover for continuous search space, Complex Systems 9 (2) (1995) 115–148.
[30] E. Zitzler, M. Laumanns, L. Thiele, Spea2: Improving the strength pareto evolutionary algorithm, in: Proceedings of the EUROGEN 2001 – Evolutionary Methods for Design, Optimisation and Control with Applications to Industrial Problems, 2001, pp. 19–26.
[31] K. Deb, M. Mohan, S. Mishra, A fast multi-objective evolutionary algorithm for finding well-spread pareto-optimal solutions, Tech. Rep. 2003002, Indian Institute of Technology Kanpur (February 2003).
[32] O. Cordón, 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.
[33] A. Klose, A. Nürnberger, On the properties of prototype-based fuzzy classifiers, IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics 37 (4) (2007) 817–835.
[34] O. Cordón, F. Herrera, Author's reply, IEEE Transactions on Fuzzy Systems 11 (6) (2003) 866–869.
[35] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, John Wiley and Sons, Inc, 605 Third Avenue, New York, NY 10158-0012, 2001.
[36] M. Setnes, H. Roubos, Ga-fuzzy modeling and classification: Complexity and performance, IEEE Transactions on Fuzzy Systems 8 (5) (2000) 509–522.
[37] D. Newman, S. Hettich, C. Blake, C. Merz, UCI repository of machine learning databases. URL: <http://www.ics.uci.edu/~mlearn/MLRepository.html>, 1998.
[38] R.A. Sarker, M.F.A. Kazi, Population size, search space and quality of solution: An experimental study, in: The 2003 Congress on Evolutionary Computation, vol. 3, 2003, pp. 2011–2018.
[39] R.P. Gorman, T.J. Sejnowski, Analysis of hidden units in a layered network trained to classify sonar targets, Neural Networks 1 (1988) 75–89.
[40] M. Berthold, D.J. Hand (Eds.), Intelligent Data Analysis: An Introduction, Springer-Verlag, Berlin, Heidelberg, 1999.
[41] T. Elomaa, J. Rousu, General and efficient multisplitting of numerical attributes, Machine Learning 36 (3) (1999) 201–244.