Available online at www.sciencedirect.com

SCIENCE DIRECT®

International Journal of Approximate Reasoning
40 (2005) 3–22

INTERNATIONAL JOURNAL OF APPROXIMATE REASONING

www.elsevier.com/locate/ijar

Design of nearest neighbor classifiers: multi-objective approach

Jian-Hung Chen [a], Hung-Ming Chen [a], Shinn-Ying Ho [b,*]

[a] *Department of Information Engineering and Computer Science, Feng Chia University, Taichung, Taiwan 407, ROC*

[b] *Department of Biological Science and Technology, Institute of Bioinformatics, National Chiao Tung University, Hsin-Chu, Taiwan 300, ROC*

Received 1 September 2004; accepted 1 November 2004
Available online 6 January 2005

Abstract

The goal of designing optimal nearest neighbor classifiers is to maximize classification accuracy while minimizing the sizes of both reference and feature sets. A usual way is to adaptively weight the three objectives as an objective function and then use a single-objective optimization method for achieving this goal. This paper proposes a multi-objective approach to cope with the weight tuning problem for practitioners. A novel intelligent multi-objective evolutionary algorithm IMOEA is utilized to simultaneously edit compact reference and feature sets for nearest neighbor classification. Three comparison studies are designed to evaluate performance of the proposed approach. It is shown empirically that the IMOEA-designed classifiers have high classification accuracy and small sizes of reference and feature sets. Moreover, IMOEA can provide a set of good solutions for practitioners to choose from in a single run. The simulation results indicate that the IMOEA-based approach is an expedient method to design nearest neighbor classifiers, compared with an existing single-objective approach.

[*] Corresponding author. Address: Department of Biological Science and Technology, Institute of Bioinformatics, National Chiao Tung University, Hsin-Chu, Taiwan 300, ROC. Tel.: +886 3 5712121 56905; fax: +886 3 5729288.

E-mail address: syho@mail.nctu.edu.tw (S.-Y. Ho).

0888-613X/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.ijar.2004.11.009

Keywords: Nearest neighbor classifier; Genetic algorithm; Multi-objective optimization; Feature selection; Minimum reference set

1. Introduction

The nearest neighbor (1-nn) classifier is commonly used due to its simplicity and effectiveness [1–5]. According to 1-nn rule, an input is assigned to the class of its nearest neighbor from a labeled reference set. The goal of designing optimal 1-nn classifiers is to maximize classification accuracy while minimizing the sizes of both reference and feature sets. The design of 1-nn classifiers is related to concept formation and conception relationship identification in granular computing [6–8]. Each subset of a universe is a granule representing a certain concept. A concept consists of two parts, the extension and intension of the concept. The extension is the set of objects which are instance of concept. The intension of a concept consists of all attributes that are valid for all those objects. In an 1-nn classifier, the selected prototypes partition patterns into disjoint subsets. Each subset can be regarded as a concept, i.e., a granule of a certain class. The patterns in the subset are the extension of the concept. The patterns in the same subset have same property: they have the same nearest prototype, which is the intension of the concept. Designing optimal 1-nn classifiers is to search for a set of prototypes associated with a subset of features to optimize multiple objective functions.

Several studies [1,3–5,9,10] have found that genetic algorithms (GAs) [11] and evolutionary algorithms (EAs) [12] are suitable for editing a compact reference set (prototype selection) and selecting useful features individually, and the simulation results indicate that EA-based methods outperform some existing non-EA based methods in designing 1-nn classifiers. It has been recognized that reference and feature sets must be simultaneously edited when designing compact 1-nn classifiers with high classification power [1,3]. Ho et al. [1] proposed an intelligent genetic algorithm IGA for simultaneous editing and feature selection to design 1-nn classifiers, using a weighted-sum approach by combining multiple objectives into a single-objective function. The IGA-based method is an efficient approach, compared with methods of editing followed by feature selection, feature selection followed by editing, individual feature selection, individual editing, and Kuncheva's GA-based method [3]. However, in order to obtain good solutions using the weighted-sum approach, domain knowledge and large computation cost are required for determining a set of good weight values.

In this paper, a multi-objective approach utilizing a novel intelligent multi-objective evolutionary algorithm IMOEA [13,14] is proposed to solve the problem of designing optimal 1-nn classifiers. IMOEA is superior to conventional multi-objective evolutionary algorithms (MOEAs) in solving some large multi-objective optimization problems (MOOPs). IMOEA is a multi-objective version of IGA by making use of Pareto dominance relationship. Therefore, the proposed approach can cope with the weight tuning problem for practitioners. Furthermore, IMOEA can efficiently obtain a set of non-dominated solutions in a single run, compared with a single-objective EA using multiple runs in terms of solution quality and computation cost. Three comparison studies are designed to evaluate performance of the proposed approach. It is shown empirically that the IMOEA-designed classifiers have high classification accuracy and small sizes of reference and feature sets. The experimental results indicate that the IMOEA-based approach is an expedient method to design nearest neighbor classifiers, compared with an existing single-objective approach.

The organization of this paper is as follows. The investigated problem is described in Section 2. Section 3 presents the design of optimal 1-nn classifiers using IMOEA. Section 4 reports the experimental results and Section 5 concludes this paper.

2. The investigated problem

2.1. Designing 1-nn classifier

1-nn classifiers demand significant computation resources (time and memory). Two ways of reducing operational cost of 1-nn classifiers are data editing and feature selection. Simultaneous optimization of data editing and feature selection has been recognized to be an efficient way to achieve high classification accuracy [1,3]. The investigated problem of designing optimal 1-nn classifiers is described as follows [1,3]:

Let $X = X_1, \ldots, X_n$ be a set of features describing objects as n-dimensional vectors $x = [x_1, \ldots, x_n]^T$ in $\Re^n$ and let $Z = z_1, \ldots, z_N$, $z_j \in \Re^n$, be a data set. Associated with each z_j, $j = 1, \ldots, N$, is a class label from a set $C = 1, \ldots, c$. The criteria of data editing and feature selection are to find subsets $S_1 \subseteq Z$ and $S_2 \subseteq X$ such that the classification accuracy is maximal and the sizes of the reduced sets, card(S_1) and card(S_2), are minimal, where card(·) denotes cardinality. Fig. 1 shows an example of editing sets of

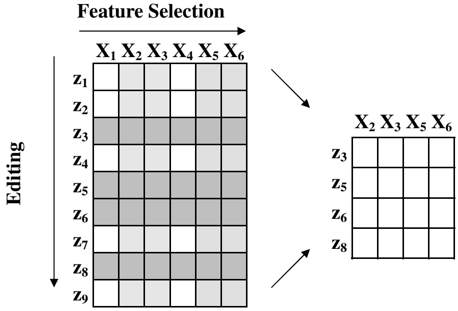

Fig. 1. Editing reference set and feature selection for the design of 1-nn classifiers. The reduced reference set z_3, z_5, z_6, z_8 and feature set X_2, X_3, X_5, X_6 correspond to the chromosome $S = [0\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 1]$.

X and Z. Define a real-valued function $P_{\text{1-nn}}(V, S_1, S_2)$ as the classification accuracy of a 1-nn classifier with S_1 and S_2:

$$P_{\text{1-nn}} : P(Z) \times P(X) \to [0, 1] \tag{1}$$

where $P(Z)$ is the power set of Z and $P(X)$ is the power set of X. The classification accuracy $P_{\text{1-nn}}$ uses a counting estimator [15] measured on a given set $V = v_1, \ldots, v_m$, as shown in Eq. (2). If v_j is correctly classified using S_1 and S_2 by the 1-nn rule, $h^{\text{CE}}(v_j) = 1$, and 0 otherwise.

$$P_{\text{1-nn}}(V, S_1, S_2) = \sum_{j=1}^{m} h^{\text{CE}}(v_j)/m \tag{2}$$

The problem is how to search for S_1 and S_2 in the combined space such that $P_{\text{1-nn}}$ is maximal, and card(S_1) and card(S_2) are minimal.

Essentially, the investigated problem is an MOOP having a search space of $C(N + n,\ \text{card}(S_1) + \text{card}(S_2))$ instances, i.e., the number of ways of choosing card(S_1) + card(S_2) out of $N + n$ binary decision variables with three incommensurable and competing objectives. The investigated problem can be formulated as the following multi-objective optimization problem:

$$\begin{cases} \text{Maximum} \quad f_1 = P_{\text{1-nn}} \\ \text{Minimum} \quad f_2 = \text{card}(S_1) \\ \text{Minimum} \quad f_3 = \text{card}(S_2) \end{cases} \tag{3}$$

2.2. Review of weighted-sum approaches

For editing a reference set, Kuncheva et al. [9] and Cano et al. [4] found that EAs using a weighted-sum objective function can offer high classification accuracy and a good data reduction ratio for designing 1-nn classifiers. To edit a reference set and select useful features simultaneously, Kuncheva et al. proposed a GA with a weighted-sum approach, using a fitness function F as follows:

$$F = P_{\text{1-nn}}(V, S_1, S_2) - \alpha\left(\frac{\text{card}(S_1) + \text{card}(S_2)}{N + n}\right). \tag{4}$$

The sum of card(S_1) and card(S_2) is used as a penalty term. The weight value α is used to tune the degree of penalty.

Generally, the number $N + n$ of binary decision variables is large. Large parameter optimization problems often pose a great challenge to engineers due to the large parametric space, the possibility of large infeasible and non-uniform areas, and the presence of multiple peaks [16]. Despite having been successfully used to solve many optimization problems, conventional GAs cannot efficiently solve large parameter optimization problems. Therefore, Ho et al. [1] proposed IGA using the fitness function F in Eq. (4) to solve the investigated problem with a large number of decision variables. It have been shown empirically that the IGA-designed classifiers outperform some existing methods, including Kuncheva's GA-based method [3] in terms of both classification accuracy and the number card(S_1) × card(S_2).

The weighted-sum approach is intuitively simple and is capable of obtaining a good solution in a single run. If dependencies among features are known, one can easily take the dependencies into account and uses individual weight values for different features in the weighted-sum approach [10]. However, different data sets represent different classification problems with different degrees of difficulties [4]. Without using domain knowledge, it is difficult for practitioners to determine appropriate weight values in the weighted-sum approach and the results may be sensitive to weight values. In order to obtain high performance, multiple experiments with different weight values for different data sets are necessary in the weighted-sum approach. For example, considering 10 levels of a weight value, the weighted-sum approach without using domain knowledge has to perform 10 experiments to determine a good weight value. If there are 20 different kinds of data sets, $10 \times 20 = 200$ experiments are necessary for efficiently designing 20 classifiers. As a result, it is essential to develop an efficient approach to coping with the weight tuning problem.

2.3. The proposed approach

Recently, MOEAs have been recognized to be well-suited for solving MOOPs because their abilities to exploit and explore multiple solutions in parallel and to find a widespread set of non-dominated solutions in a single run [17]. Several MOEAs based on Pareto dominance relationship are proposed to solve MOOPs directly, and present more promising results than single-objective optimization techniques theoretically and empirically [17–20]. By making use of Pareto dominance relationship, MOEAs are capable of performing fitness assignment without using a weighted linear combination of all objectives. Pareto dominance relationship and some related terminologies are introduced below. Assume the multi-objective functions are to be minimized. Mathematically, MOOPs can be represented as the following vector mathematical programming problems:

$$\text{minimize } F(Y) = \{f_1(Y), f_2(Y), \ldots, f_I(Y)\}, \tag{5}$$

where Y denotes a solution and $f_i(Y)$ is generally a nonlinear objective function. When the following inequalities hold between two solutions Y_1 and Y_2, Y_2 is a non-dominated solution and is said to dominate $Y_1(Y_2 \succ Y_1)$:

$$\forall i\colon f_i(Y_1) > f_i(Y_2) \wedge \exists j : f_j(Y_1) > f_j(Y_2). \tag{6}$$

When the following inequality hold between two solutions Y_1 and Y_2, Y_2 is said to weakly zdominate $Y_1(Y_2 \succeq Y_1)$:

$$\forall i\colon f_i Y_1 \geqslant f_i(Y_2). \tag{7}$$

A feasible solution Y^* is said to be a Pareto-optimal solution if and only if there does not exist a feasible solution Y which dominates Y^*, and the corresponding vector of Pareto-optimal solutions is called Pareto-optimal front. An example in a bi-objective space is shown in Fig. 2, where the circles represent non-dominated solutions and the black dots are dominated solutions. MOEA seems to be an alternative approach for solving the investigated problem on the assumption that no information

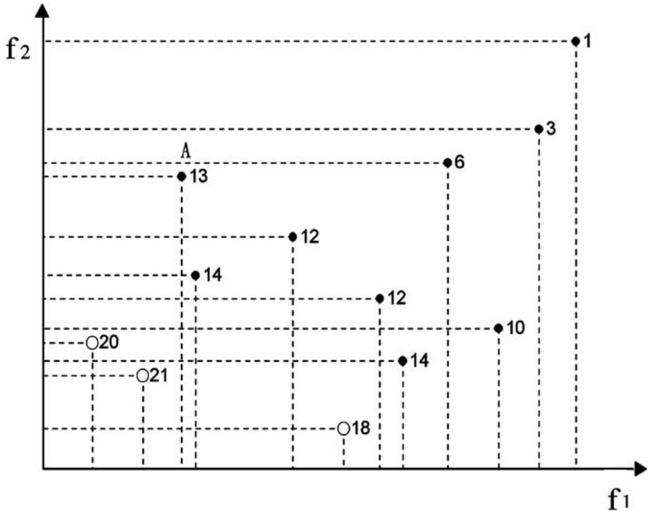

Fig. 2. Fitness values of the participant individuals with $c = 12$ in the objective space. The circles represent non-dominated solutions and the black dots are dominated solutions. The fitness value of the dominated individual A using GPSIFF is $3 - 2 + 12 = 13$.

on the preference among objectives is available. Moreover, a set of non-dominated solutions can be provided for practitioners to choose from. If a solution is not suitable, practitioners can easily choose another solution without performing another experiment. The issue now is how to develop an efficient MOEA for effectively solving the problem of designing 1-nn classifiers.

3. IMOEA-designed 1-nn classifier

A novel intelligent multi-objective evolutionary algorithm IMOEA is utilized to solve the problem of designing optimal 1-nn classifiers. The chromosome representation is presented in Section 3.1. The fitness assignment strategy of IMOEA is described in Section 3.2. An intelligent crossover operation which plays an important role in IMOEA is described in Section 3.3. The used IMOEA for designing 1-nn classifiers is provided in Section 3.4.

3.1. Chromosome representation

The feasible solution S corresponding to the reduced reference and feature sets is encoded as a binary string consisting of $N + n$ bits. The first N bits are used for $S_1 \subseteq Z$ and the last n bits for $S_2 \subseteq X$. The ith bit has a value 1 when the respective element of $Z(X)$ is included in $S_1(S_2)$, and 0 otherwise. The search space consists of 2^{N+n} points. For example, considering the reduced reference set z_3, z_5, z_6, z_8 and feature set X_2, X_3, X_5, X_6 in Fig. 1, the corresponding chromosome is $S =$ [0 0 1 0 1 1 0 1 0 0 1 1 0 1 1] with $N = 9$ and $n = 6$.

3.2. Fitness assignment

Fitness assignment strategy is known as an important issue in solving MOOPs. The fitness assignment strategy of IMOEA uses a generalized Pareto-based scale-independent fitness function GPSIFF considering the quantitative fitness values in the Pareto space for both dominated and non-dominated individuals. GPSIFF makes the use of Pareto dominance relationship to evaluate individuals using a single measure of performance.

Let the fitness value of an individual Y be a tournament-like score obtained from all participant individuals by the following function:

$$\text{GPSIFF}(X) = p - q + c, \tag{8}$$

where p is the number of individuals which can be dominated by Y, and q is the number of individuals which can dominate Y in the objective space. Generally, a constant c can be optionally added in the fitness function to make fitness values positive. In this study, c is the number of all participant individuals. Note that GPSIFF is to be maximized in IMOEA.

GPSIFF uses a pure Pareto-ranking fitness assignment strategy, which differs from the traditional Pareto-ranking methods, such as non-dominated sorting [11,21] and Zitzler and Thiele's method [19]. GPSIFF can assign discriminative fitness values to not only non-dominated individuals but also dominated ones. Fig. 2 shows an example for illustrating the fitness value using GPSIFF for a bi-objective minimization problem. For example, three individuals are dominated by A ($p = 3$) and two individuals dominate A ($q = 2$). Therefore, the fitness value of A is $3 - 2 + 12 = 13$. It can be found that one individual has a larger fitness value if it dominates more individuals. On the contrary, one individual has a smaller fitness value if more individuals dominate it.

3.3. Intelligent crossover (IC)

In conventional crossover operations of GAs, two parents generate two children with a combination of their chromosomes using randomly selected cut points. The merit of IC is that, the systematic reasoning ability of orthogonal experimental design (OED) [22–24] is incorporated in the crossover operator to economically estimate the contribution of individual genes to a fitness function, and then the better genes are intelligently picked up to form the chromosomes of children. Theoretically analysis and experimental studies for illustrating the superiority of IC with the use of OED can be found in [1,14,25,26].

3.3.1. Orthogonal array and factor analysis

Orthogonal array (OA) is a factional factorial matrix, which assures a balanced comparison of levels of any factor or interaction of factors. It is a matrix of numbers arranged in rows and columns where each row represents the levels of factors in each experiment, and each column represents a specific factor that can be changed from each experiment. The array is called orthogonal because all columns can be evaluated independently of one another, and the main effect of one factor does not bother the estimation of the *main effect* of another factor. A two-level OA used in IC is described as follows.

Let there be γ factors with two levels for each factor. The total number of experiments is 2^γ for the popular "one-factor-at-a-time" study. The columns of two factors are orthogonal when the four pairs, (1,1), (1,2), (2,1), and (2,2), occur equally frequently over all experiments. Generally, levels 1 and 2 of a factor represent selected genes from parents 1 and 2, respectively. To establish an OA of γ factors with two levels, first we obtain an integer $\omega = 2^{\lceil \log_2(\gamma+1) \rceil}$, where the bracket represents a ceiling operator. Then, build an orthogonal array $L_\omega(2^{\omega-1})$ with ω rows and $(\omega-1)$ columns and use the first γ columns; the other $(\omega - \gamma - 1)$ columns are ignored. Table 1 illustrates an example of OA $L_8(2^7)$. The algorithm of constructing OAs can be found in [24]. OED can reduce the number of experiments for factor analysis. The number of OA combinations required to analyze all individual factors is only ω or $O(\gamma)$, where $\gamma + 1 \leqslant \omega \leqslant 2\gamma$.

After proper tabulation of experimental results, we can further proceed factor analysis to determine the relative effects of various factors. Let Y_t denote a function value of the combination t, where $t = 1,\ldots,\omega$. Define the main effect of factor j with level k as S_{jk} where $j = 1,\ldots,\gamma$ and $k = 1,2$

$$S_{jk} = \sum_{t=1}^{\omega} Y_t \cdot O_t. \tag{9}$$

where $O_t = 1$ if the level of factor j of combination t is k; otherwise, $O_t = 0$. Since GPSIFF is to be maximized, the level 1 of factor j makes a better contribution to the function than level 2 of factor j does when $S_{j1} > S_{j2}$. If $S_{j1} < S_{j2}$, level 2 is better. If $S_{j1} = S_{j2}$, levels 1 and 2 have the same contribution. The main effect reveals the individual effect of a factor. The most effective factor j has the largest main effect difference MED = $|S_{j1} - S_{j2}|$. Note that the main effect holds only when no or weak interaction exists, and that makes the OED-based IC efficient.

After the better one of two levels of each factor is determined, a reasoned combination consisting of γ factors with better levels can be easily derived. The reasoned combination is a potentially good approximation to the best one of the 2^γ combina-

Table 1
Orthogonal array $L_8(2^7)$

Experiment no. t	Factor							Y_t
	1	2	3	4	5	6	7	
1	1	1	1	1	1	1	1	Y_1
2	1	1	1	2	2	2	2	Y_2
3	1	2	2	1	1	2	2	Y_3
4	1	2	2	2	2	1	1	Y_4
5	2	1	2	1	2	1	2	Y_5
6	2	1	2	2	1	2	1	Y_6
7	2	2	1	1	2	1	1	Y_7
8	2	2	1	2	1	1	2	Y_8

tions. OED uses well-planned procedures in which certain factors are systematically set and modified, and then main effects of factors on the response variables can be observed. Therefore, OED using OA and factor analysis is regarded as a systematic reasoning method.

3.3.2. Procedures of intelligent crossover

Two parents breed two children using IC at a time. How to use OA and factor analysis to perform the IC operation with γ factors is described as the following steps:

Step 1. Randomly divide the parent chromosomes into γ pairs of gene segments where each gene segment is treated as a factor.
Step 2. Use the first γ columns of OA $L_\omega(2^{\omega-1})$ where $\omega = 2^{\lceil \log_2(\gamma+1) \rceil}$.
Step 3. Let levels 1 and 2 of factor j represent the j^{th} gene segment of a chromosome coming from parents, respectively.
Step 4. Simultaneously evaluate the fitness values Y_t of the ω combinations corresponding to the experiments t, where $t = 1, \ldots, \omega$.
Step 5. Compute the main effect S_{jk} where $j = 1, \ldots, \gamma$ and $k = 1,2$.
Step 6. Determine the better one of two levels for each gene segment. Select level 1 for the jth factor if $S_{j1} > S_{j2}$. Otherwise, select level 2.
Step 7. The chromosome of the first child is formed using the combination of the better gene segments from the derived corresponding parents.
Step 8. Rank the most effective factors from rank 1 to rank γ. The factor with a large MED has a high rank.
Step 9. The chromosome of the second child is formed similarly as the first child except that the factor with the lowest rank adopts the other level.

For one IC operation, the two children are more promising to be new non-dominated individuals. The individuals corresponding to OA combinations are called *by-products* of IC. The by-products are well planned and systematically sampled within the hypercube formed by parents, so some of them are promising to be non-dominated individuals. Therefore, the non-dominated by-products will be added to the elite set in IMOEA.

IC attempts to identify good gene segments according to the main effect of factors (gene segments), and seeks the best combination consisting of a set of good gene segments. It is desirable to evolve these good gene segments based on the evolution ability of EA such that a set of optimal gene segments can exist in a population. Consequently, all these optimal gene segments can be collected to form an optimal solution through the combination phase. IC also takes advantage of GPSIFF to accurately estimate the main effect of factors and consequently can achieve an efficient recombination using IC. It is less efficient for IC to use Zitzler and Thiele's method [19] where the fitness values of dominated individuals in a cluster are always identical. The decision of number γ depends on problem difficulties and stopping conditions. If function evaluations are expensive, one may use a small value of γ. One can also use an adaptive value of γ in IC.

3.4. Intelligent multi-objective evolutionary algorithm

Since it has been recognized that the incorporation of elitism may be useful in maintaining diversity and improving the performance of multi-objective EAs [17,19,20], IMOEA selects a number of elitists from an elite set E in the selection step. The elite set E with capacity N_E maintains the best non-dominated solutions generated so far. In addition, an external set $\overline{E}$ without capacity restriction is used to store all the non-dominated solutions ever generated so far. The used IMOEA in the investigated problem is as follows:

Step 1. (Initialization) Randomly generate an initial population of N_{pop} individuals and create two empty elite sets E, $\overline{E}$ and an empty temporary elite set E'.

Step 2. (Evaluation) Compute all objective function values of each individual in the population.

Step 3. (Fitness assignment) Assign each individual a fitness value by using GPSIFF.

Step 4. (Update elite sets) Add the non-dominated individuals in both the population and E' to E, and empty E'. Considering all individuals in E, remove the dominated ones in E. Add E to $\overline{E}$, remove the dominated ones in $\overline{E}$. If the number of non-dominated individuals in E is larger than N_E, randomly discard excess individuals.

Step 5. (Selection) Select $N_{pop} - N_{ps}$ individuals from the population using the binary tournament selection and randomly select N_{ps} individuals from E to form a new population, where $N_{ps} = N_{pop} \times p_s$ and p_s is a selection proportion. If N_{ps} is greater than the number N_E of individuals in E, let $N_{ps} = N_E$.

Step 6. (Recombination) Perform the IC operations with a recombination probability p_c. For each IC operation, add non-dominated individuals derived from by-products and two children to E'.

Step 7. (Mutation) Apply the mutation operator to each gene in the individuals with a mutation probability p_m.

Step 8. (Termination test) If a stopping condition is satisfied, stop the algorithm and output $\overline{E}$. Otherwise, go to Step 2.

4. Experimental results

Three comparison studies are designed to evaluate the performance of the IMOEA-designed classifiers. First, the IGA-designed classifiers are compared with the IMOEA-designed classifiers for revealing the merits of the proposed multi-objective approach. Second, a representative multi-objective algorithm SPEA [19] which outperforms many existing MOEAs is selected to compare with IMOEA for evaluating the efficiency of IMOEA. Third, the results of a decision-tree classifier C4.5 [27] and DROP5 [28] using the same data sets are reported for further understanding the effectiveness of the IMOEA-designed classifiers.

4.1. Data sets

The 11 well-known data sets with numerical attribute values, shown in Table 2, are used to evaluate performance of the proposed approach. All the data sets are available from [29]. The set of test classification problems is composed of problems with various dimensions from 3 to 60 and various degrees of overlapping that the general test accuracy ranges from 50% to 100%. All the feature values are normalized to real numbers in the unit interval [0, 1].

To assure fair performance comparisons by avoiding the dependence on the training and test data, the following data partition is used. First, the patterns with the same class label are put together without changing their order in the original data file. Subsequently, the patterns with odd index values are assigned to the set V_1 and the other patterns are assigned to the set V_2. When $V_1(V_2)$ is used as a training set, $V_2(V_1)$ is a test set. In the training phase, the training set is used to select the reduced sets S_1 and S_2, and calculate the classification accuracy $P_{1\text{-nn}}$. The test classification accuracy is measured using the test set.

4.2. Performance measurement

The coverage metric $C(A,B)$ of two solution sets A and B [19] used to compare the performance of two corresponding algorithms considering the four objectives:

$$C(A,B) = \frac{|\{a \in A, b \in B, a \succeq b\}|}{|B|}, \tag{10}$$

where $\succeq$ stands for weakly dominate in Pareto dominance relationship. The value $C(A,B) = 1$ means that all individuals in B are weakly dominated by A. On the contrary, $C(A,B) = 0$ denotes that none of individuals in B is weakly dominated by A. Because the C measure considers the weakly dominance relationship between two sets A and B, $C(A,B)$ is not necessarily equal to $1 - C(B,A)$. The comparison results

Table 2
The number of classes, features, patterns, V_1 and V_2 of various data set

Data set	Index	No. of classes	No. of features	No. of patterns	No. of V_1	No. of V_2
cmc	1	3	9	1473	738	735
glass	2	6	9	214	109	105
haberman	3	2	3	306	154	152
heartc[a]	4	5	13	297	150	147
iris	5	3	4	150	75	75
liver-disorder	6	2	6	345	173	172
new-thyroid	7	3	5	215	108	107
pima	8	2	8	768	384	384
sonar	9	2	60	208	105	103
wdbc	10	2	30	569	285	284
wine	11	3	13	178	90	88

[a] Six patterns with missing attribute values are excluded.

of two solution sets using the coverage metric are depicted using box plots. A box plot provides an excellent visual result of a distribution. The box stretches from the lower hinge (defined as the 25th percentile) to the upper hinge (the 75th percentile) and therefore contains the middle half of the scores in the distribution. The median is shown as a line across the box. The whisker stretches 10% to 90%. The median is shown as a line across the box.

For easy understanding, the data reduction ratio Drd is used to measure the efficiency of editing reference sets:

$$\mathrm{Drd} = \frac{\mathrm{card}(S_1)}{N} \tag{11}$$

The feature reduction ratio Frd is used to measure the efficiency of editing feature sets:

$$\mathrm{Frd} = \frac{\mathrm{card}(S_2)}{n} \tag{12}$$

4.3. IMOEA vs. IGA

The parameter settings of IGA are as follows: $N_{pop} = 30$, $p_s = 0.2$, $p_c = 0.6$ and $p_m = 0.05$. The fitness function of IGA is F in Eq. (4). Nine different weight values of α, $\alpha = 0.1$, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 and 0.9 are used. In order to make comparisons with multi-objective solutions, the nine experiments using nine different weight values ranged from 0.1 to 0.9 are regarded as an IGA run. The parameter settings of IMOEA are as follows: $N_{pop} = 30$, $N_{E\max} = 30$, $p_s = 0.4$, $p_c = 0.6$ and $p_m = 0.05$. The factor value of OA is $\gamma = 7$ in both IGA and IMOEA. The stopping condition is the number of function evaluations $N_{eval} = 10000$. Thirty independent runs of IGA and IMOEA were performed. The solution set of an IGA run is compared with the solutions set of an IMOEA run using the coverage metric.

Fig. 3 shows $C(IGA,IMOEA)$ and $C(IMOEA,IGA)$ from 30 runs, for the (training, test) data sets (V_1, V_2) and (V_2, V_1). Observing the median in the box plots, the results shows that the solutions of IMOEA weakly dominate 40–80% solutions of IGA, and the solutions of IGA weakly dominate 5–40% solutions of IMOEA. The results reveal that IMOEA can evolve a set of non-dominated solutions that cover the solutions of IGA. Table 3 listed the average numbers of non-dominated solutions of IGA that are not dominated by IMOEA. Recalled that an IGA run is composed of the nine experiments using nine different weight values, $C(IMOEA, IGA)$ in Fig. 3 and Table 3 indicate that only several good weight values used in IGA can derive non-dominated solutions. This reveals that the performance of the weighted-sum approach is sensitive to the weight value.

Theoretically, the weighted-sum approach does not attempt to optimize the sizes of reference and feature sets, but only penalizes the individuals with large values of $\mathrm{card}(S_1)$ and $\mathrm{card}(S_2)$. On the contrary, the multi-objective approach tries to optimize all the three objectives. A typical Fig. 4 depicted the non-dominated solutions obtained from 30 runs of IGA and IMOEA in solving the wdbc data set. Figs. 5–7

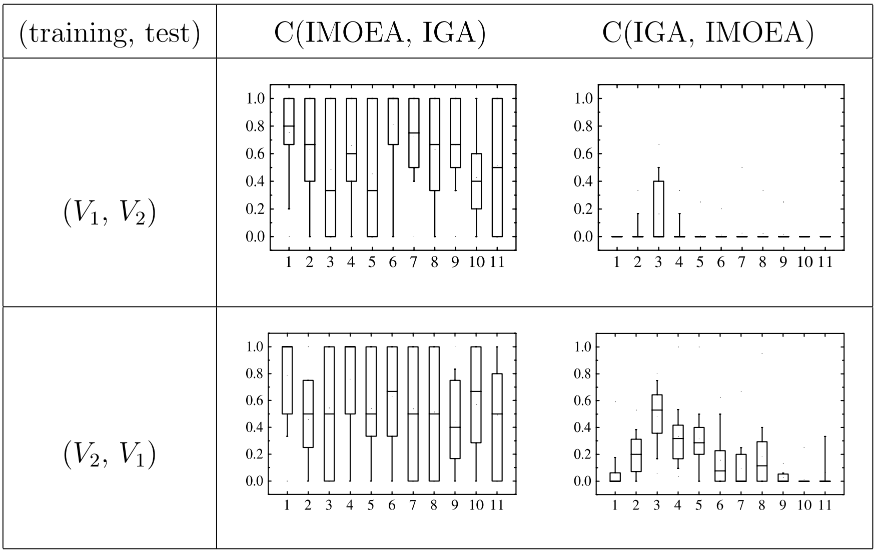

Fig. 3. Performance comparisons of IMOEA and IGA based on box plots. The vertical axis is the value of C and the horizontal axis is the index of data sets.

Table 3

The average number of non-dominated solutions of IGA that are not dominated by IMOEA, averaged from 30 IGA runs

Data set	IGA
cmc	0.87
glass	1.27
haberman	0.93
heartc	1.13
iris	1.00
liver-disorder	0.53
new-thyroid	0.80
pima	1.27
sonar	1.17
wdbc	1.90
wine	1.03

depicted the distribution of all the solutions obtained by IGA and IMOEA in each objective. Table 4 shows the best results of each objective that obtained by IMOEA. Fig. 5 shows both IGA and IMOEA can obtain high quality classification accuracy. Due to the goodness and badness of solutions are determined using Pareto dominance relationship, IMOEA may obtain non-dominated solutions with low classification accuracy, but with small values of card(S_1) and card(S_2). Figs. 6 and 7 show that IMOEA can obtain smaller reduction ratios and smaller numbers of features than IGA. The ability to optimize the three objectives simultaneously enables

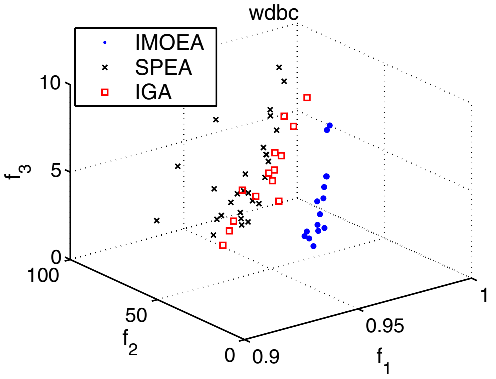

Fig. 4. The non-dominated solutions obtained by IGA, IMOEA and SPEA, the training and test sets are (V_2, V_1).

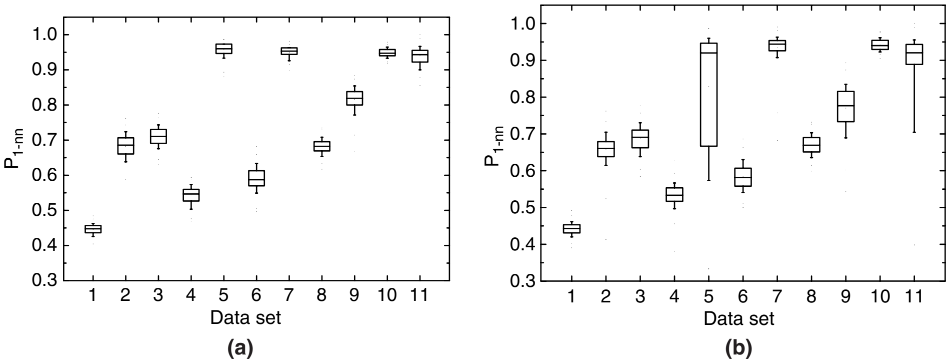

Fig. 5. The distribution of solutions on the classification accuracy: (a) IGA and (b) IMOEA.

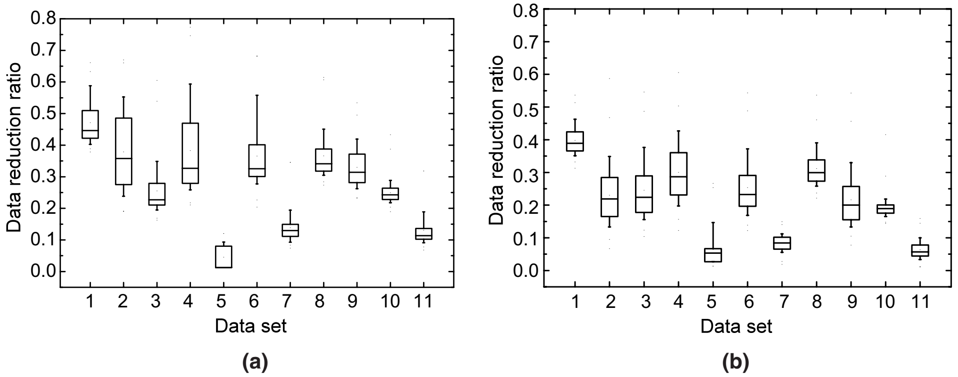

Fig. 6. The distribution of solutions on the data reduction ratio Drd: (a) IGA and (b) IMOEA.

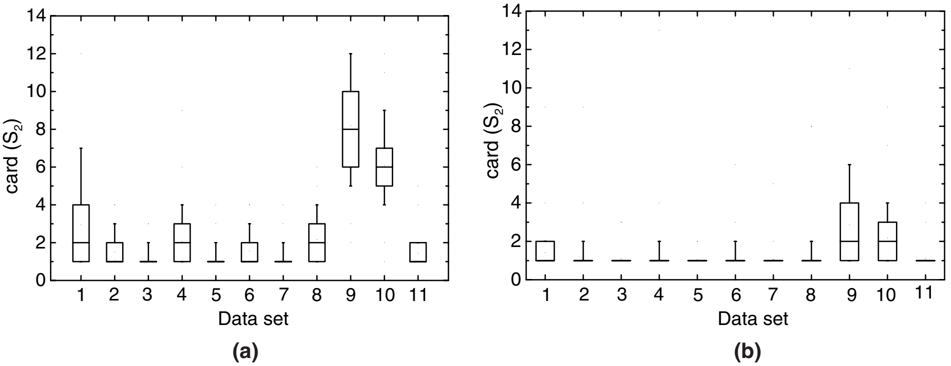

Fig. 7. The distribution of solutions on the card(S_2): (a) IGA; (b) IMOEA.

Table 4
The best classification accuracy, data reduction ratio and feature reduction ratio obtained by IMOEA

Data set	IMOEA		
	$P_{1\text{-nn}}$	Drd (%)	Frd (%)
cmc	0.4881	31.77	5.26
glass	0.7384	7.92	11.11
haberman	0.7647	10.46	50.00
heartc	0.6131	12.79	7.69
iris	0.9867	1.33	25.00
liver	0.6812	12.75	16.67
new-thyroid	0.9861	2.80	20.00
pima	0.7201	22.27	12.50
sonar	0.8750	8.17	1.67
wdbc	0.9684	14.41	3.33
wine	0.9775	1.12	7.69
Average	0.7999	11.44	14.63

IMOEA to search for representative patterns and relevant features. In consequence, IMOEA can cope with the weight tuning problems, and IMOEA can obtain widespread non-dominated solutions considering multiple objectives.

4.4. IMOEA vs. SPEA

The parameter settings of SPEA are $P_c = 0.6$ and $P_m = 0.05$. The population size and the external population size of SPEA are 75 and 25, respectively. Thirty independent runs were performed. The stopping condition is the number of function evaluations $N_{eval} = 10{,}000$.

Fig. 8 shows $C(IMOEA, SPEA)$ and $C(SPEA, IMOEA)$ from 30 runs. Observing the median in the box plots, Fig. 8 shows that the solutions of IMOEA weakly dominate 50–90% solutions of SPEA, and the solutions of SPEA weakly dominate 5–50%

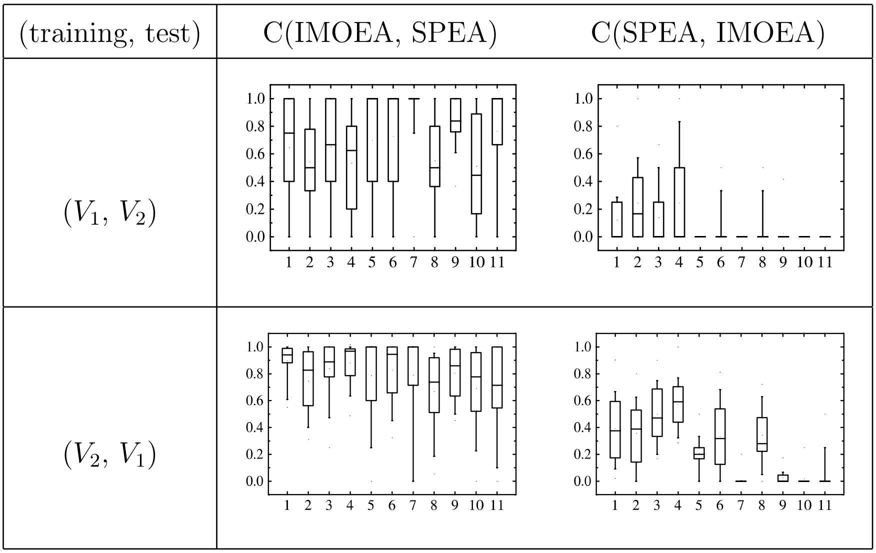

Fig. 8. Performance comparisons of IMOEA and SPEA based on box plots. The vertical axis is the value of C and the horizontal axis is the index of data sets.

solutions of IMOEA. Fig. 4 depicted the non-dominated solutions obtained from 30 runs of SPEA, IGA and IMOEA in solving the wdbc data set. The results are similar in the other data sets. The results indicate that IMOEA is an efficient MOEA and can converge to well-distributed and high-quality solutions, compared with SPEA. The reason is due to that the large number of decision variables poses difficulties for SPEA to converge to Pareto-optimal solutions in a limited time. On the contrary, IMOEA utilized IC, GPSIFF and elitism to cope with the large parameter optimization problem efficiently.

4.5. IMOEA-designed 1-nn classifier vs. DROP5 and C4.5

Due to different aims and merits of classifiers, the performance of the proposed approach cannot be directly compared with those of non-1-nn classifiers in justice. However, some performance comparisons with a decision-tree classifier C4.5 release 8 [27], DROP5 [28] are given to demonstrate the merits of the proposed approach. In this section, C4.5 release 8 algorithm with pruned tree and default parameters is used. For each data set, the training set is used for training, and then the classification accuracy is measured by the test set. Two trails using (V_1, V_2) and (V_2, V_1) are (training, test) data sets are performed. The average classification accuracy, data and feature reduction ratios of DROP5 and the C4.5 are reported in Table 5. The classification accuracy of DROP5 and C4.5 are used as the baseline classification accuracy. The data reduction ratio of DROP5 is used as the baseline data reduction ratio.

Due to MOEAs' nature, IMOEA tries to optimize the three objectives and tends to obtain widespread solutions on all the three objectives. Considering only $P_{1\text{-nn}}$, it is not fair to perform *t*-test on all the classification accuracy of the IMOEA-designed classifiers to the baseline classification accuracy. Therefore, Eq. (4) is adopted as a simple decision making model to select a solution from a set of non-dominated solutions. First, for each run of IMOEA, all the non-dominated solutions are measured using the training set by Eq. (4). Then, $P_{1\text{-nn}}$ of the best solution is measured using the test set.

Table 5 reports the results of DROP5 and C4.5. Table 6 reports the results of the *t*-test on the classification accuracy of the selected IMOEA-designed classifiers using $\alpha = 0.5$ with DROP5 and the C4.5 classifiers. Table 6 shows that the classification

Table 5
Results of average classification accuracy, data reduction ratio and feature reduction ratio on DROP5 and C4.5

Data set	DROP5			C4.5		
	$P_{1\text{-nn}}$	Drd (%)	Frd (%)	$P_{1\text{-nn}}$	Drd (%)	Frd (%)
cmc	0.4888	28.31	100	0.5050	100	100.00
glass	0.6692	30.29	100	0.6730	100	77.78
haberman	0.7256	13.72	100	0.7160	100	66.67
heartc	0.5418	19.86	100	0.5420	100	96.15
iris	0.9200	20.67	100	0.9265	100	37.50
liver-disorder	0.5883	30.14	100	0.6580	100	100.00
new-thyroid	0.9210	12.56	100	0.9255	100	80.00
pima	0.7227	20.18	100	0.7055	100	87.50
sonar	0.7694	27.36	100	0.7405	100	16.67
wdbc	0.9367	8.97	100	0.9170	100	21.67
wine	0.9439	12.35	100	0.9320	100	26.92
Average	0.7480	20.40	100	0.7492	100	64.62

Table 6
Results of *t*-test on the classification accuracy of the selected IMOEA-designed classifiers, the C4.5 classifiers and DROP5, with 29 degrees of freedom at the 0.05 the significance level

Data set	IMOEA ($\alpha = 0.5$)		*t*-test	
	$P_{1\text{-nn}}$	Deviation	DROP5	C4.5
cmc	0.4461	0.0103	Lose	Lose
glass	0.6698	0.0183	Equal	Equal
haberman	0.6891	0.0176	Lose	Lose
heartc	0.5340	0.0159	Lose	Lose
iris	0.9400	0.0174	Win	Win
liver-disorder	0.5872	0.0237	Equal	Lose
new-thyroid	0.9464	0.0153	Win	Win
pima	0.6711	0.0155	Lose	Lose
sonar	0.8001	0.0199	Win	Win
wdbc	0.9426	0.0073	Win	Win
wine	0.9306	0.0158	Lose	Equal

The solutions of IMOEA are selected using Eq. (4) with $\alpha = 0.5$.

Table 7
Results of average data and feature reduction ratio on the IGA-designed classifiers, and the selected IMOEA-designed classifiers

Data set	IGA		IMOEA ($\alpha = 0.5$)	
	Drd (%)	Frd (%)	Drd (%)	Frd (%)
cmc	47.15	32.67	41.48	14.67
glass	37.88	16.67	27.95	11.11
haberman	25.54	39.67	22.50	35.67
heartc	38.30	15.77	33.52	8.69
iris	4.52	33.00	6.47	25.00
liver-disorder	36.56	25.83	26.25	17.83
new-thyroid	13.88	24.00	9.52	20.00
pima	36.60	27.25	30.75	15.63
sonar	33.00	13.68	23.69	2.45
wdbc	24.96	20.27	18.58	5.27
wine	12.99	11.15	7.19	7.69
Average	28.31	23.63	22.54	14.91

The solutions of IMOEA are selected using Eq. (4) with $\alpha = 0.5$.

accuracy of the selected IMOEA-designed classifiers are good in four data sets, but are inferior to the baseline classification accuracy in five data sets. Table 7 reports the data and the feature reduction ratios of IMOEA-designed classifiers and IGA-designed classifiers. It shows that the selected IMOEA-designed classifiers offer smaller data and feature reduction ratios than those of the IGA-designed classifiers. Compare the data reduction ratios of the selected IMOEA-designed classifiers with those of DROP5 in Table 5, it shows that the selected IMOEA-designed classifiers offer smaller data reduction ratios than those of DROP5 in small data sets, but bigger data reduction ratios than those of DROP5 in large data sets. Compare the feature reduction ratios of the selected IMOEA-designed classifiers in Table 7 with those of the C4.5 classifiers in Table 5, it shows that the selected IMOEA-designed classifiers offer smaller feature reduction ratios than those of the C4.5 classifiers. The simulation results indicate that the proposed approach can achieve better data and feature reduction ratios without losses in generalization accuracy.

If the first preference of practitioners is classification accuracy, fine tuning of can select solutions with better classification accuracy than those of the selected solutions using $\alpha = 0.5$. Other multi-criteria decision making techniques [18], such as fuzzy multi-criteria decision making, can be used to select a suitable solution for practitioners, instead of using the simple decision making model. If the first preference is the data reduction ratio, a large value N_{eval} should be given for a data set with a large number of instances.

4.6. Summary

From the comparison studies, it reveals that the merits of the IMOEA-designed 1-nn classifiers are:

(1) *Generality*. For practitioners, tuning weight values for high performance of the weighted-sum approaches in solving a classification problem is not required. Weight tuning for different classification problems is not necessary, too.
(2) *Effectiveness*. High-quality and widespread solutions can be obtained, compared with some existing methods in terms of classification accuracy, the size of reference set and the size of feature set.
(3) *Economy*. The training computation cost is less than the weighted-sum approaches with multiple experiments.
(4) *Flexibility*. A set of non-dominated solutions can be generated in a single run of IMOEA. A satisfactory solutions can be fast obtained by given preferences from practitioners, without performing another run of EAs.

5. Conclusion

In this paper, we have proposed an approach to designing optimal 1-nn classifiers using a novel intelligent multi-objective evolutionary algorithm IMOEA with intelligent crossover based on orthogonal experimental design. The proposed approach cope with the weight tuning problem for practitioners. It has been shown empirically that the IMOEA-designed classifiers have high performance, compared with the IGA-based and SPEA-based classifiers in terms of classification accuracy, the size of reference set and the size of feature set. Moreover, IMOEA provides a set of solutions for practitioners to choose from. IMOEA can be easily applied without using domain knowledge to efficiently design 1-nn classifiers with high-dimensional patterns with overlapping. The simulation results indicate that the IMOEA-based approach is a good alternative method to design nearest neighbor classifiers, compared with some existing approaches.

References

[1] S.-Y. Ho, C.-C. Liu, S. Liu, Design of an optimal nearest neighbor classifier using an intelligent genetic algorithm, Pattern Recognition Letters 23 (13) (2002) 1495–1503.
[2] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed., Wiley, New York, 2001.
[3] L.I. Kuncheva, L.C. Jain, Nearest neighbor classifier: simultaneous editing and feature selection, Pattern Recognition Letters 20 (1999) 1149–1156.
[4] J.R. Cano, F. Herrera, M. Lozano, Using evolutionary algorithms as instance selection for data reduction in kdd: an experimental study, IEEE Transactions on Evolutionary Computation 7 (6) (2003) 561–575.
[5] X. Llora, J.M. Garell, Prototype induction and attribute selection via evolutionary algorithms, Journal of Intelligent Data Analysis 7 (3) (2003) 193–208.
[6] L. Zadeh, Fuzzy graph, rough sets and information granularity, in: Proceedings of the Third International Workshop on Rough Sets and Soft Computings, San Jose, 1994, pp. 10–12.
[7] T.Y. Lin, Granular computing on binary relations I: Data mining and neighborhood systems, in: L. Skoworn, A. Polkowski (Eds.), Granular Computing: An Emerging Paradigm, Rough Sets In Knowledge Discovery, 1998, pp. 107–121.

[8] T.Y. Lin, Granulation and nearest neighborhood: rough set approach, in: W. Pedrycs (Ed.), Granular Computing: An Emerging Paradigm, Physica-Verlag, Wurzburg, 2001, pp. 125–142.

[9] L.I. Kuncheva, J.C. Bezdek, Nearest prototype classification: Clustering, genetic algorithms, or random search? IEEE Transactions on Systems Man and Cybernetics Part C-Applications and Reviews 28 (1) (1998) 160–164.

[10] M.L. Raymer, W.E. Punch, E.D. Goodman, L.A. Kuhn, A.K. Jain, Dimensionality reduction using genetic algorithms, IEEE Transactions on Evolutionary Computation 4 (2) (2000) 164–171.

[11] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Co., Reading, MA, 1989, ISBN 0-201-15767-5.

[12] T. Bàck, D.B. Fogel, Z. Michalewics, Handbook of Evolutionary Computation, Institute of Physics Publishing, 1998.

[13] S.-Y. Ho, X.-I. Chang, An efficient generalized multiobjective evolutionary algorithm, in: W. Banzhaf, J. Daida, A.E. Eiben, M.H. Garzon, V. Honavar, M. Jakiela, R.E. Smith (Eds.), Proceedings of the Genetic and Evolutionary Computation Conference 1999, Vol. 1, Morgan Kaufmann Publishers, Los Altos, CA, 1999, pp. 871–878.

[14] S.-Y. Ho, L.-S. Shu, J.-H. Chen, Intelligent evolutionary algorithms for large parameter optimization problems, IEEE Transaction on Evolutionary Computation 8 (6) (2004) 522–541.

[15] S.J. Raudys, A.K. Jain, Small sample size effects in statistical pattern recognition: recommendations for practitioners and open problems, in: Proceeding of 10th International Conference on Pattern Recognition, vol. 23, 1990, pp. 417–423.

[16] K. Krishna Kumar, S. Narayanaswamy, S. Garg, Solving large parameter optimization problems using a genetic algorithm with stochastic coding, in: G. Winter, J. Periaux, M. Galan, P. Cuesta (Eds.), Genetic Algorithms in Engineering and Computer Science, John Wiley Sons, New York, 1995.

[17] K. Deb, Multi-objective optimization using evolutionary algorithmsWiley-Interscience Series in Systems and Optimization, John Wiley and Sons, New York, 2001.

[18] T. Gal, T.J. Stewart, T. Hanne, Multicriteria Decision Making: Advances in MCDM Models, Algorithms, Theory, and Applications, Kluwer Academic Publishers, Dordrecht, 1999.

[19] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strengthen Pareto approach, IEEE Transaction on Evolutionary Computation 4 (3) (1999) 257–271.

[20] C.A.C. Coello, A comprehensive survey of evolutionary-based multiobjective optimization techniques, International Journal of Knowledge and Information System 1 (3) (1999) 269–308.

[21] N. Srinivas, K. Deb, Multiobjective optimization using non-dominated sorting in genetic algorithms, Evolutionary Computation 2 (3) (1994) 221–248.

[22] A. Dey, Orthogonal fractional factorial designs, Wiley, New York, 1985.

[23] C.R. Hicks, K.V. Turner, Fundamental concepts in the design of experiments, fifth ed., Oxford University Press, New York, 1999.

[24] Y.W. Leung, Y.P. Wang, An orthogonal genetic algorithm with quantization for global numerical optimization, IEEE Transactions on Evolutionary Computation 5 (1) (2001) 41–53.

[25] S.-Y. Ho, Y.-C. Chen, An efficient evolutionary algorithm for accurate polygonal approximation, Pattern Recognition 34 (12) (2001) 2305–2317.

[26] J.-H. Chen, S.-Y. Ho, Evolutionary multi-objective optimization of flexible manufacturing systems, in: L. Spector, E.D. Goodman (Eds.), Proceeding of the Genetic and Evolutionary Computation Conference GECCO-2001, Morgan Kaufmann Publishers, Los Altos, CA, 2001, pp. 1260–1267.

[27] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kauffman, San Mateo, Los Altos, CA, 1993.

[28] R. Wilson, T. Martinez, Reduction techniques for instance-based learning algorithms, Machine Learning 38 (2000) 257–286.

[29] C.L. Blake, C.J. Merz, Uci repository of machine learning databases. Available from <http://www.ics.uci.edu/mlearn/MLRepository.html> (1998).