Design of Nearest Neighbor Classifiers Using an Intelligent Multi-Objective Evolutionary Algorithm

Jian-Hung Chen, Hung-Ming Chen, and Shinn-Ying Ho

Department of Information Engineering and Computer Science,
Feng Chia University, Taichung 407, Taiwan
TEL:886-4-24517250 ext. 3723, FAX: 886-4-24516101
E-mail: jh.chen@ieee.org, hmchen@ms25.url.com.tw, syho@fcu.edu.tw

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. Two 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.

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 pattern 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. Ho *et al*. [4] proposed an intelligent genetic algorithm IGA for simultaneous reference set editing and feature selection to design 1-nn classifiers, using a weighted-sum approach by combining multiple objectives into a single-objective function. However, in order to obtain good solutions using the weighted-sum approach, domain knowledge and large computational 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 [6], [7] is proposed to solve the problem of designing optimal 1-nn classifiers. 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. Two 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

The investigated problem of designing optimal 1-nn classifiers is described as follows [4], [5]:

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 R^n and let $Z = \{z_1, \ldots, z_N\}$, $z_j \in R^n$, be a data set. Associated with each z_j, j=1, …, 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. Define a real-valued function $P_{1\text{-nn}}(V, S_1, S_2)$ as the classification accuracy of a 1-nn classifier with S_1 and S_2:

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

where $P(Z)$ is the power set of Z and $P(X)$ is the power set of X. The classification accuracy $P_{1\text{-nn}}$ uses a counting estimator $h^{CE}(v_j)$ [11] measured on a given validation set $V = \{v_1, \ldots, v_m\}$, as shown in Equation (2). If v_j is correctly classified using S_1 and S_2 by the 1-nn rule, $h^{CE}(v_j)$ =1, and 0 otherwise.

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

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

Essentially, the investigated problem has a search space of C(N+n, card(S_1)+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:

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

2.2 Review of Weighted-sum Approaches

For editing a reference set, Kuncheva *et al.* [1] and Cano *et al.* [3] 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 [4], [5]:

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

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. Despite having been successfully used to solve many optimization problems, conventional GAs cannot efficiently solve large parameter optimization problems. Therefore, Ho *et al*. [4] proposed IGA using the fitness function F in Equation (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 [5] in terms of both classification accuracy and the number card(S_1)$\times$card(S_2). However, different data sets represent different classification problems with different degrees of difficulties [3]. 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.

3 IMOEA-Designed 1-nn Classifier

3.1 Chromosome Representation

The feasible solution S corresponding to the reduced reference and feature sets is encoded using 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 i-th 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 $\{X_2, X_3, X_5, X_6\}$, 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

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 [6], [7]. 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:

$$GPSIFF(Y) = p - q + c, \tag{5}$$

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.

3.3 Intelligent Crossover (IC)

In the 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) [4], [6]-[8] 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. The procedure of IC, theoretically analysis and experimental studies for illustrating the superiority of IC with the use of OED can be found in [4], [6]-[8].

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 (ω-1) columns and use the first γ columns; the other (ω-γ-1) columns are ignored. The algorithm of constructing OAs can be found in [13]. OED can reduce the number of experiments for factor analysis.

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 $k = 1, 2$:

$$S_{jk} = \sum_{t=1}^{\omega} y_t \cdot F_t \ , \tag{6}$$

where $F_t = 1$ if the level of factor j of combination t is k; otherwise, $F_t = 0$. 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 $MED = \left| S_{j1} - S_{j2} \right|$. 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 combinations.

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 jth 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 IGC. 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.

3.4 Intelligent Multi-objective Evolutionary Algorithm

IMOEA uses an elite set E with capacity N_{Emax} to maintain the non-dominated individuals 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 an empty elite set E and an empty temporary elite set E'.

Step 2: (Evaluation) Compute all objective function values of each individual in the population. Assign each individual a fitness value by using GPSIFF.

Step 3: (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. If the number N_E of non-dominated individuals in E is larger than N_{Emax}, randomly discard excess individuals.

Step 4: (Selection) Select N_{pop}-N_{ps} individuals from the population using binary tournament selection and randomly select N_{ps} individuals from E to form a new population, where $N_{\mathrm{ps}}=N_{\mathrm{pop}}\cdot p_{\mathrm{s}}$. If $N_{\mathrm{ps}} > N_{\mathrm{E}}$, let $N_{\mathrm{ps}}= N_{\mathrm{E}}$.

Step 5: (Recombination) Perform the IC operations for $N_{\mathrm{pop}}\cdot p_{\mathrm{c}}$ selected parents. For each IC operation, add non-dominated individuals derived from by-products and two children to E'.

Step 6: (Mutation) Apply bit mutation with p_{m} to the population.

Step 7: (Termination test) If a stopping condition is satisfied, stop the algorithm. Otherwise, go to Step 2.

4 Experimental Results

The 11 well-known data sets with numerical attribute values are used to evaluate performance of the proposed approach. All the data sets are available from [12]. 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.

The coverage metric C(A, B) of two solution sets A and B [9] is used to compare the performance of two corresponding algorithms considering the three objectives:

$$C(A,B)=\frac{\left|\{a\in A; b\in B; a\succeq b\}\right|}{\left|B\right|}. \tag{7}$$

C(A, B) = 1 means that all individuals in B are weakly dominated by A. On the contrary, C(A, B) = 0 means that none of individuals in B are weakly dominated by A. The comparison results of two solution sets using the coverage metric are depicted using box plots. For easy understanding, the data reduction ratio *Drd* is used to measure the efficiency of editing reference sets:

$$Drd = \frac{card(S_1)}{N}. \tag{8}$$

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

$$Frd = \frac{card(S_2)}{n}. \tag{9}$$

The parameter settings of IGA are as follows: N_{pop}=30, p_s=0.4, p_c=0.6 and p_m=0.05. The fitness function of IGA is F in Equation (4). Nine different weight values of α, α=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_{Emax}=30, p_s=0.4, p_c=0.6 and p_m=0.05. The factor value of OA is 7 in both IGA and IMOEA. The stopping condition is the number of function evaluations N_{eval}=10000. Thirty independent runs were performed. Each of IGA, and IMOEA performed 30 independent runs. The solution sets of 30 runs are compared using the coverage metric.

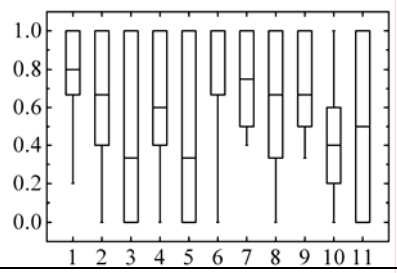
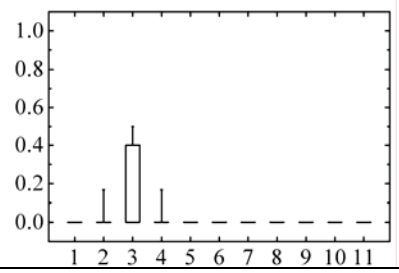
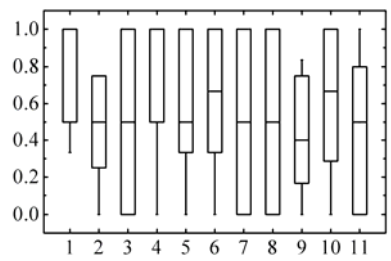
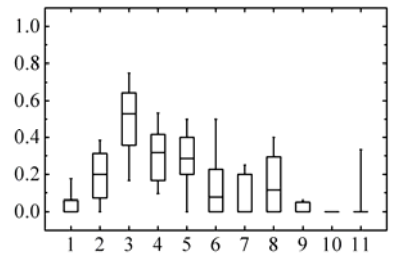
(training, test)	C(IMOEA, IGA)	C(IGA, IMOEA)
(V_1, V_2)		
(V_2, V_1)		

Fig. 1. Box plots based on the coverage metric C. The vertical axis is the value of C and the horizontal axis is the index of data sets.

Fig. 1 shows the coverage metric of 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 1. 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*	$P_{1\text{-nn}}$	*Frd*
(1) cmc	0.4888	28.31%	0.5050	100.00%
(2) glass	0.6692	30.29%	0.6730	77.78%
(3) haberman	0.7256	13.72%	0.7160	66.67%
(4) heartc†	0.5418	19.86%	0.5420	96.15%
(5) iris	0.9200	20.67%	0.9265	37.50%
(6) liver-disorder	0.5883	30.14%	0.6580	100.00%
(7) new-thyroid	0.9210	12.56%	0.9255	80.00%
(8) pima	0.7227	20.18%	0.7055	87.50%
(9) sonar	0.7694	27.36%	0.7405	16.67%
(10) wdbc	0.9367	8.97%	0.9170	21.67%
(11) wine	0.9439	12.35%	0.9320	26.92%
Average	0.7480	20.40%	0.7492	64.62%

†Six patterns with missing attribute values are excluded.

Table 2. 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. The solutions of IMOEA are selected using Equation (4) with α =0.5

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

Table 3. Results of average data and feature reduction ratio on the IGA-designed classifiers, the selected IMOEA-designed classifiers and the C4.5 classifiers. The solutions of IMOEA are selected using Equation (4) with α=0.5

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

Due to its 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, Equation (4) is adopted as a simple decision making model to select a solution from a set of non-dominated solutions. Table 1 reports the results of C4.5[10] and DROP5[11]. Table 2 reports the results of the *t*-test on the classification accuracy of the selected IMOEA-designed classifiers using α=0.5 with the C4.5 and the DROP5 classifiers. Table 3 reports the data and the feature reduction ratios of the selected IMOEA-designed classifiers. It shows that the selected IMOEA-designed classifiers offer smaller data and feature reduction ratios than those of the IGA-designed classifiers. From Tables 1-3, the simulation results indicate that the proposed approach can achieve better data and feature reduction ratios without losses in generalization accuracy.

5 Conclusions

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 copes 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 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 the existing single-objective approach.

References

1. L. I. Kuncheva, J. C. Bezdek, Nearest prototype classification clustering, genetic algorithms, or random search?, IEEE Trans. on Systems, Man and Cybernetics-Part C, 28 (1) (1998) 160-164.
2. M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn, A. K. Jain, Dimensionality reduction using genetic algorithms, IEEE Trans. on Evolutionary Computation, 4 (2) 2000 164-171.
3. J. R. Cano, F. Herrera and M. Lozano, Using evolutionary algorithms as instance selection for data reduction in KDD: an experimental study, IEEE Trans. on Evolutionary Computation, 7 (6) 2003 561-575.
4. 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
5. L. I. Kuncheva, L. C. Jain, Nearest neighbor classifier: Simultaneous editing and feature selection, Pattern Recognition Letters, 20 (1999) 1149-1156.
6. S.-Y. Ho, X.-I Chang, An efficient generalized multiobjective evolutionary algorithm, in: Proc. of the Genetic and Evolutionary Computation Conference GECCO-99, Orlando, July 1999, pp. 871-878.
7. J.-H. Chen, S.-Y. Ho, Evolutionary multi-objective optimization of flexible manufacturing systems, in: Proc. of the Genetic and Evolutionary Computation Conference GECCO-2001, San Francisco, July 2001, pp. 1260-1267.
8. S.-Y. Ho, Y.-C. Chen, An efficient evolutionary algorithm for accurate polygonal approximation, Pattern Recognition, 34 (2001), 2305-2317.
9. E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: A comparative case study and the strengthen Pareto approach, IEEE Trans. on Evolutionary Computation, 3 (4) (1999), 257-271.
10. J. R. Quinlan, C4.5: Programs for Machine Learning, San Mateo, Morgan Kauffman, Los Altos, CA, 1993.
11. R.D. Wilson and T.R. Martinez, Reduction Techniques for Instance-based Learning Algorithms, Machine Learning 38 (2000), 257–286.
12. C. L. Blake, C. J. Merz, UCI Repository of machine learning databases, 1998. Available from <http://www.ics.uci.edu/~mlearn/MLRepository.html>.