A Classification and Pareto Domination based Multiobjective Evolutionary Algorithm

Jinyuan Zhang, Aimin Zhou, and Guixu Zhang
Shanghai Key Laboratory of Multidimensional Information Processing
Department of Computer Science and Technology
East China Normal University, 500 Dongchuan Road, Shanghai, China
Email: jyzhang@ecnu.cn, {amzhou,gxzhang}@cs.ecnu.edu.cn

***Abstract*—In multiobjective evolutionary algorithms, most selection operators are based on the objective values or the approximated objective values. It is arguable that the selection in evolutionary algorithms is a classification problem in nature, i.e., selection equals to classifying the selected solutions into one class and the unselected ones into another class. Following this idea, we propose a classification based preselection for multiobjective evolutionary algorithms. This approach maintains two external populations: one is a positive data set which contains a set of 'good' solutions, and the other is a negative data set contains a set of 'bad' solutions. In each generation, the two external populations are used to train a classifier firstly, then the classifier is applied to filter the newly generated candidate solutions and only the ones labeled as positive are kept as the offspring solutions. The proposed preselection is integrated into the Pareto domination based algorithm framework in this paper. A systematic empirical study on the influence of different classifiers and different reproduction operators has been done. The experimental results indicate that the classification based preselection can improve the performance of Pareto domination based multiobjective evolutionary algorithms.**

***Index Terms*—Evolutionary algorithm, multiobjective optimization, classification, preselection**

I. Introduction

In scientific and engineering areas, it is often required to optimize several objectives simultaneously. This kind of problems are called *multiobjective optimization problems (MOPs)*. MOPs can be modeled into different mathematical formulations. In this paper, we consider the following box-constrained continuous MOP:

$$\begin{array}{ll} \min & F(x) = (f_1(x), \cdots, f_m(x))^T \\ \text{s.t} & x \in \Omega \end{array} \tag{1}$$

where $x = (x_1, \cdots, x_n)^T \in R^n$ is a decision variable vector, $\Omega = \Pi_{i=1}^n [a_i, b_i] \subset R^n$ is the feasible region of the search space, $f_i : R^n \rightarrow R, i = 1, \cdots, m$, is a continuous mapping, and $F(x)$ is an objective vector.

Since the objectives in (1) are often contradicted with each other and the increase of one objective may lead to the deterioration of another, there may not exist a solution that can optimize all objectives at same time. A set of tradeoff solutions among objectives are thus of interest. Let x and y be two feasible solutions of (1), x is said to dominate y, denoted as $F(x) \prec F(y)$, if and only if $f_i(x) \leq f_i(y)$ for $i = 1, \cdots, m$ and $F(x) \neq F(y)$. A solution $x^* \in \Omega$ is called a Pareto optimal solution of (1) if there does not exist a solution $x \in \Omega$ such that $F(x) \prec F(x^*)$. The set of all the Pareto optimal solutions of (1) is called the *Pareto set (PS)* in decision space and the *Pareto front (PF)* in objective space. Usually the whole PF (PS) of (1) is not able to be obtained. Therefore, in practice the solvers aim to find an approximation to the PF (PS) for a decision maker [1].

Comparing to traditional optimization methods, *evolutionary algorithms (EAs)* [2], [3] are more suitable for dealing with multiobjective optimization problems. A major reason is that an EA usually maintains a population of candidate solutions, which makes it be able to approximate the PF (PS) in a single run. In the last few decades, a variety of *multiobjective evolutionary algorithms (MOEAs)* have been proposed and applied to real-world applications [4]. Most current existing MOEAs can be roughly classified into following categories.

1) *Domination based MOEAs:* This approach uses the Pareto domination or its variants and some other strategies to differentiate and order solutions. Some algorithms, such as PAES [5], NSGA-II [6], and SPEA2 [7], fall into this category.
2) *Indicator based MOEAs:* This approach is a set based selection strategy. The performance indicators are utilized to measure the quality of some sets of solutions, and set with the best quality value will be selected. Some indicators, such as hypervolume [8]–[10] and R2 [11], [12], have been used.
3) *Decomposition based MOEAs:* This approach decomposes an MOP into a set of subproblems and tackle these subproblems simultaneously. The offspring reproduction and environmental selection are based on the subproblems. A typical algorithm in this category is MOEA/D [13], [14].

It is clear that the major work on the above three types of MOEAs is with the environmental selection, i.e., how to select some solutions into the next generation. The reproduction operation, i.e., how to generate new trial solutions from the population, has not been emphasized as the environmental selection. Some reproduction operators that are designed for scalar-objective optimization are directly applied in MOEAs [14]–[18]. Some work has indicated that these reproduction operators may not be suitable for multiobjective optimization [19]. A major reason is that the PF (PS) of an

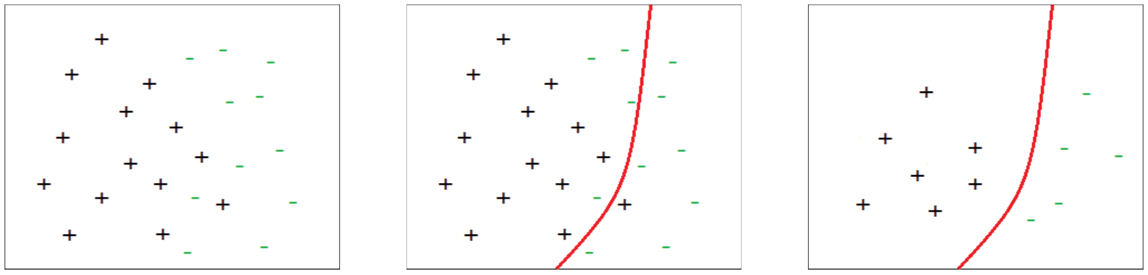

(a) data preparing (b) classifier training (c) classifying

Fig. 1. An illustration of the basic idea of CPS.

MOP is not an isolated solution but a set of solutions which show some kinds of regularity in both objective and decision spaces [20]. Therefore in order to design efficient algorithms, the properties of MOPs should be considered. Following this idea, we have recently proposed a series of MOEAs [18], [21]. The basic idea behind these algorithms is to utilize the priori and/or posterior problem information in reproduction operator designing.

This paper considers the reproduction from another angle and addresses the preselection, which is a part of reproduction. In this paper, the preselection is used to choose some offspring solutions from a large set of candidate offspring solutions that are generated by reproduction operators. A *classification based preselection (CPS)* is proposed for this purpose. In CPS, we use some recorded data points that are with positive or negative labels to train a classifier in each generation. After a set of candidate offspring solutions are generated, the classifier is applied to label the candidate offspring solutions and only those with positive labels are kept as the offspring solutions. Since CPS is based on Pareto domination, it can be naturally integrated into Pareto domination based MOEAs. Since classification algorithms can easily separate a whole data set into different classes according to their own characteristics, many researchers have applied classification into EAs to do surrogate procedures [22]–[24]. Different from these approaches, this paper considers to use classification to do preselection.

The rest of the paper is organized as follows. Section II presents the MOEA framework with CPS. The algorithm implementation details, such as classification model building, and offspring reproducing, are introduced as well. The proposed approach is systematically studied in Section III. Finally Section IV concludes this paper with some future work remarks.

II. CPS based MOEA

This section introduces a general *Pareto domination based evolutionary algorithm framework with a classification based preselection*. For simplicity, we use CPS-MOEA to denote this approach.

A. Algorithm Framework

It is arguable that the selection, including preselection, in an MOEA can be regarded as a classification problem. A set of solutions can be labeled as positive sample points, hopefully they are nondominated, while the other solutions are labeled as negative sample points. With these sample points, we can train a classifier and use it to classify the newly generate candidate offspring solutions. The *classification based preselection (CPS)* is following this idea. Fig. 1 shows the basic procedure of CPS.

To implement CPS, we introduce two external populations P_+ and P_-: P_+ contains some 'good' solutions found so far with label $+1$, hopefully they are nondominated, and P_- contains some 'bad' solutions with a label -1.

We use $Q = NDS(P, N)$ denote the selection procedure based on the nondominated sorting scheme [6]. Firstly the population P is partitioned into several clusters according to the Pareto domination, and solutions in each cluster are nondominated with each other. Secondly in each cluster the crowding distance of each solution is calculated. Thirdly according to cluster and crowding distance, all solutions in P are ordered. The procedure returns the best N solutions and stores them in Q. The details of the procedure are referred to [6].

In each generation, CPS-MOEA maintains

- a population of N solutions $P = \{x^1, x^2, \cdots, x^N\}$ and their objective values $\{F(x^1), F(x^2), \cdots, F(x^N)\}$,
- an external population P_+ which contains $\lfloor \frac{N}{2} \rfloor$ 'good' solutions found so far, and
- an external population P_- which contains $\lfloor \frac{N}{2} \rfloor$ 'bad' solutions found so far.

The CPS-MOEA framework is shown in Algorithm 1, which is a typical Pareto domination based MOEA. The population is initialized in *Line 1*, the offspring solutions are generated in *Line 5-13*, and the Pareto domination based selection is used to update the population in *Line 14*. CPS is integrated in this procedure, and we would like to make the following comments.

- *Population initialization:* In *Line 1*, N solutions are uniformly randomly sampled from Ω, the external population P_+ is initialized as the best $\lfloor \frac{N}{2} \rfloor$ solutions in P and P_- is initialized as the other half.
- *Stopping condition:* The algorithm will stop when the number of function evaluations exceeds a given maximum number of function evaluation FES in *Line 2*.

Algorithm 1: Framework of CPS-MOEA

```
1  Initialize the population $P = \{x^1, x^2, \cdots, x^N\}$, set
   $P_+ = NDS(P, \lfloor \frac{N}{2} \rfloor)$ and $P_- = P \backslash P_+$;
2  while termination condition is not satisfied do
3      Train a classifier $l = \hat{c}(x)$ with data set $P_+ \cup P_-$;
4      Set $Q = \emptyset$;
5      foreach $x \in P$ do
6          Generate $M$ candidate offspring solutions
           $Y = \{y^1, \cdots, y^M\}$;
7          Set $V = \{y \in Y | \hat{c}(y) = 1\}$;
8          if $V = \emptyset$ then
9              $V = Y$ ;
10         end
11         Randomly choose $y \in V$ as the offspring solution;
12         Set $Q = Q \cup \{y\}$;
13     end
14     Update $P = NDS(P \cup Q, N)$;
15     Set $Q_+$ be the nondominated solutions in $Q$ and $Q_-$
       be dominated solutions in $Q$;
16     Update $P_+ = NDS(P_+ \cup Q_+, \lfloor \frac{N}{2} \rfloor)$;
17     Update $P_- = NDS(P_- \cup Q_-, \lfloor \frac{N}{2} \rfloor)$;
18 end
19 return the population $P$.
```

- *Classifier training:* In *Line 3*, a classifier is built based on the training data $P_+ \cup P_-$.
- *Offspring reproduction:* In *Lines 5-13*, a set of M candidate offspring solutions are generated by using a reproduction operator for each solution, then these candidate offspring solutions will get their predicted labels through the classifier, finally a solution with a predicted label $+1$ is chosen as the offspring solution (if no solutions with label $+1$ are found, a solution will be picked up randomly).
- *Population selection:* In *Lines 14*, the selection based on nondominated sorting scheme is applied to select solutions into the next generation.
- *External population update:* In *Lines 15-17*, the newly generated nondominated and dominated solutions are used to update P_+ and P_- respectively.

In followings, we explain how to train a classifier and how to generate candidate offspring solutions.

B. Classifier Training

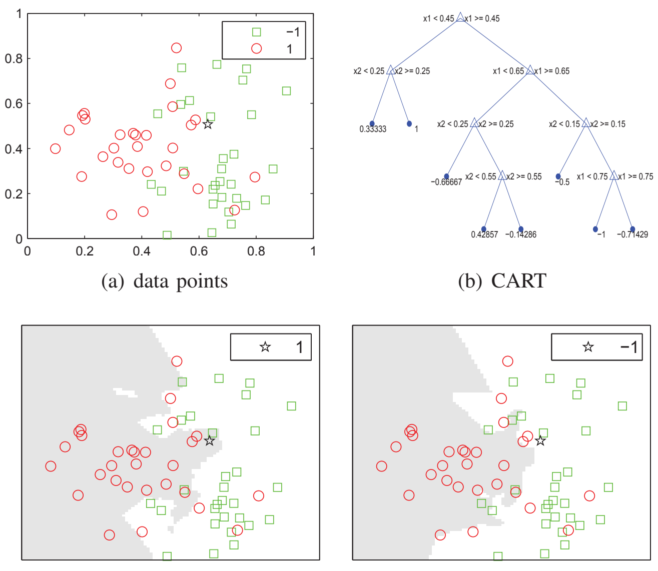
Classification aims to find a classifier based on a given set of categorized data points, and thus to predict the categories of new data points [25], [26]. Mathematically, a classifier training problem can be formulated as follows. Let $\{< x, l >\}$ be a set of training points, where x is called the feature vector of the data point, $l \in L$ is called the label of the data point and L is a limited set of integers. Suppose the relationship between feature vector and label can be denoted as $l = C(x)$. The target of a training is to find an approximated relationship $l = \hat{C}(x)$ based on the training data set.

In our case, there are two categories, i.e., $L = \{-1, +1\}$, and a solution x can be directly regarded as a feature vector. The training data set $P_+ = \{< x, +1 >\}$ contains a set of 'good' solutions found by an MOEA, while the training data set $P_- = \{< x, -1 >\}$ contains a set of 'bad' ones. In multiobjective optimization, we can naturally use the Pareto domination to judge whether a solution is 'good' or 'bad': a nondominated solution can be labeled as a 'good' one while a dominated solution can be labeled as a 'bad' one.

A variety of classification methods have been proposed in the community of statistical and machine learning [27]. In this paper, we choose the following two classification methods to test our CPS approach: *classification and regression tree (CART)* [28] and *k-nearest neighbor (KNN)* [29].

1) CART: CART is a non-parametric decision tree learning method, which can be used to do both classification and regression by using a decision tree [28]. In the case of classification, each interior node of the decision tree is an element of the feature vector, and each leaf node denotes a category label. A path from the root node to a leaf node represents a decision rule. The Gini [30] impurity is usually used to train the decision tree.

Take the data points in Fig. 2(a) as an example, which contains two classes of data points marked as circles and squares respectively. Fig. 2(b) shows the corresponding decision tree that has eight decision rules. The test point marked as pentagram in Fig. 2(a) will be labeled as $+1$ according to these rules.



(a) data points (b) CART (c) 3NN (d) 5NN

Fig. 2. An example of CART and KNN model training and prediction.

2) KNN: KNN is a type of instance-based learning method, which does not have an explicit training process as CART does [29]. Like CART, KNN can also be applied to do both regression and classification. In terms of classification, a new

TABLE I
THE MEAN AND STD. IGD VALUES OBTAINED BY CPS-MOEA WITH 3NN, 5NN, AND CART AFTER 20%, 40%, 60%, 80%, AND 100% FES OVER 30 RUNS ON F1-F10. THE NUMBERS IN THE BRACKETS DENOTE THE RANKS OF THE THREE COMPARED METHODS.

	Classifier	20%	40%	60%	80%	100%	Mean Rank
	3NN	4.06e-02$_{2.86e-02}$[1]	5.99e-03$_{6.14e-04}$[1]	4.71e-03$_{1.71e-04}$[1]	4.43e-03$_{1.10e-04}$[3]	4.36e-03$_{1.27e-04}$[2]	1.6
F1	5NN	5.87e-02$_{3.75e-02}$[2]	6.44e-03$_{8.41e-04}$[2]	4.72e-03$_{1.94e-04}$[2]	4.37e-03$_{1.01e-04}$[1]	4.31e-03$_{8.36e-05}$[1]	1.6
	CART	6.20e-02$_{3.45e-02}$[3]	6.89e-03$_{9.64e-04}$[3]	4.87e-03$_{2.59e-04}$[3]	4.41e-03$_{1.25e-04}$[2]	4.37e-03$_{9.32e-05}$[3]	2.8
	3NN	1.53e-01$_{6.89e-02}$[2]	7.58e-03$_{1.47e-03}$[2]	4.67e-03$_{3.10e-04}$[2]	4.31e-03$_{1.57e-04}$[2]	4.20e-03$_{9.31e-05}$[2]	2.0
F2	5NN	1.41e-01$_{6.92e-02}$[1]	7.35e-03$_{1.31e-03}$[1]	4.63e-03$_{2.47e-04}$[1]	4.30e-03$_{1.45e-04}$[1]	4.19e-03$_{9.95e-05}$[1]	1.0
	CART	1.84e-01$_{6.73e-02}$[3]	1.03e-02$_{8.15e-03}$[3]	4.76e-03$_{3.02e-04}$[3]	4.37e-03$_{1.90e-04}$[3]	4.25e-03$_{1.02e-04}$[3]	3.0
	3NN	3.17e+00$_{2.35e-01}$[1]	9.39e-01$_{4.79e-01}$[3]	4.45e-02$_{3.62e-02}$[1]	1.04e-02$_{2.62e-03}$[1]	5.20e-03$_{1.31e-03}$[2]	1.6
F3	5NN	3.20e+00$_{2.10e-01}$[2]	9.25e-01$_{4.53e-01}$[2]	6.25e-02$_{7.91e-02}$[2]	1.14e-02$_{8.37e-03}$[3]	4.75e-03$_{7.21e-04}$[1]	2.0
	CART	3.21e+00$_{2.40e-01}$[3]	9.09e-01$_{4.82e-01}$[1]	6.31e-02$_{8.74e-02}$[3]	1.06e-02$_{2.06e-03}$[2]	5.57e-03$_{1.18e-03}$[3]	2.4
	3NN	1.95e-01$_{2.23e-02}$[2]	6.49e-02$_{7.06e-03}$[1]	5.34e-02$_{1.54e-03}$[3]	5.07e-02$_{1.36e-03}$[3]	4.93e-02$_{1.50e-03}$[1]	2.0
F4	5NN	1.92e-01$_{2.77e-02}$[1]	6.65e-02$_{8.19e-03}$[2]	5.28e-02$_{1.90e-03}$[2]	5.05e-02$_{1.57e-03}$[2]	4.95e-02$_{1.39e-03}$[2]	1.8
	CART	2.09e-01$_{2.89e-02}$[3]	6.77e-02$_{1.04e-02}$[3]	5.27e-02$_{1.95e-03}$[1]	5.04e-02$_{1.30e-03}$[1]	4.95e-02$_{1.42e-03}$[3]	2.2
	3NN	2.41e-01$_{5.79e-02}$[3]	3.69e-02$_{2.87e-02}$[2]	7.79e-03$_{2.78e-03}$[2]	5.40e-03$_{3.16e-04}$[1]	5.05e-03$_{1.93e-04}$[1]	1.8
F5	5NN	2.09e-01$_{4.97e-02}$[1]	2.53e-02$_{2.14e-02}$[1]	7.68e-03$_{4.06e-03}$[1]	5.41e-03$_{4.89e-04}$[2]	5.05e-03$_{2.36e-04}$[2]	1.4
	CART	2.36e-01$_{6.45e-02}$[2]	4.11e-02$_{3.69e-02}$[3]	9.16e-03$_{8.50e-03}$[3]	6.41e-03$_{5.41e-03}$[3]	5.60e-03$_{3.14e-03}$[3]	2.8
	3NN	6.95e-01$_{5.75e-02}$[1]	4.85e-01$_{1.46e-01}$[1]	1.37e-01$_{1.57e-01}$[1]	1.67e-02$_{1.52e-02}$[1]	6.81e-03$_{1.25e-03}$[1]	1.0
F6	5NN	7.14e-01$_{3.33e-02}$[3]	5.66e-01$_{9.48e-02}$[3]	2.09e-01$_{1.64e-01}$[3]	3.45e-02$_{7.30e-02}$[3]	9.94e-03$_{1.17e-02}$[3]	3.0
	CART	7.03e-01$_{2.89e-02}$[2]	5.21e-01$_{1.24e-01}$[2]	1.63e-01$_{1.45e-01}$[2]	2.27e-02$_{1.98e-02}$[2]	7.99e-03$_{2.67e-03}$[2]	2.0
	3NN	1.39e+00$_{5.89e-01}$[2]	2.10e-01$_{2.31e-01}$[3]	8.88e-02$_{6.10e-02}$[2]	6.22e-02$_{2.38e-02}$[2]	4.89e-02$_{1.34e-02}$[2]	2.2
F7	5NN	1.33e+00$_{4.54e-01}$[1]	1.84e-01$_{1.74e-01}$[1]	9.41e-02$_{5.35e-02}$[3]	6.71e-02$_{2.58e-02}$[3]	5.28e-02$_{1.72e-02}$[3]	2.2
	CART	1.44e+00$_{5.90e-01}$[3]	1.88e-01$_{1.29e-01}$[2]	8.42e-02$_{3.21e-02}$[1]	6.16e-02$_{1.21e-02}$[1]	4.73e-02$_{8.01e-03}$[1]	1.6
	3NN	4.33e-01$_{3.35e-02}$[3]	1.65e-01$_{8.43e-02}$[2]	9.31e-02$_{5.73e-02}$[2]	8.27e-02$_{7.75e-02}$[3]	6.50e-02$_{1.16e-02}$[3]	2.6
F8	5NN	4.13e-01$_{5.32e-02}$[1]	1.63e-01$_{8.79e-02}$[1]	1.03e-01$_{7.38e-02}$[3]	6.88e-02$_{4.73e-03}$[1]	6.31e-02$_{3.78e-03}$[1]	1.4
	CART	4.21e-01$_{3.32e-02}$[2]	1.67e-01$_{7.64e-02}$[3]	9.01e-02$_{5.12e-02}$[1]	6.94e-02$_{4.56e-03}$[2]	6.47e-02$_{2.40e-03}$[2]	2.0
	3NN	2.53e-02$_{8.58e-03}$[1]	7.78e-03$_{6.62e-03}$[3]	6.71e-03$_{6.59e-03}$[3]	6.46e-03$_{6.59e-03}$[3]	6.40e-03$_{6.54e-03}$[3]	2.6
F9	5NN	2.53e-02$_{8.67e-03}$[2]	7.20e-03$_{3.41e-03}$[2]	6.03e-03$_{3.29e-03}$[2]	5.79e-03$_{3.27e-03}$[2]	5.69e-03$_{3.29e-03}$[2]	2.0
	CART	2.99e-02$_{8.75e-03}$[3]	6.08e-03$_{2.06e-03}$[1]	4.93e-03$_{2.10e-03}$[1]	4.66e-03$_{2.09e-03}$[1]	4.57e-03$_{2.08e-03}$[1]	1.4
	3NN	1.44e+02$_{9.99e+00}$[3]	1.38e+02$_{1.15e+01}$[1]	1.35e+02$_{1.07e+01}$[1]	1.34e+02$_{1.07e+01}$[1]	1.32e+02$_{1.21e+01}$[1]	1.4
F10	5NN	1.44e+02$_{1.05e+01}$[2]	1.39e+02$_{1.01e+01}$[2]	1.36e+02$_{9.36e+00}$[3]	1.35e+02$_{9.41e+00}$[3]	1.33e+02$_{8.44e+00}$[3]	2.6
	CART	1.43e+02$_{5.97e+00}$[1]	1.39e+02$_{7.98e+00}$[3]	1.36e+02$_{8.87e+00}$[2]	1.34e+02$_{9.35e+00}$[2]	1.32e+02$_{8.17e+00}$[2]	2.0
Mean Rank over F1-F10		3NN	1.88	5NN	1.9	CART	2.2

feature vector is classified by a majority vote of its neighbors in the training data set. More specifically, K nearest neighbors to a new point x are firstly selected from the training set, then x will be assigned to the category most common among these selected training data points. If $K = 1$, x will be assigned the same label as the nearest training data point.

Let us consider the test point marked as pentagram in Fig. 2(a). Fig. 2(c)-(d) show the classification maps for $K = 3$, and 5 respectively. The test point will be classified as $+1$, and -1 for $K = 3$, and 5. It is clear that KNN is quite easy to implement, but it is sensitive to the parameter K.

C. Offspring Reproduction

Line 6 in Algorithm 1 is to generate a set of candidate offspring solutions. In this paper we choose *differential evolution (DE)* [31] and *estimation of distribution algorithm (EDA)* [32] based operators to do so.

1) DE based reproduction: For the DE based offspring reproduction, we use the same reproduction procedure introduced in MOEA/D-DE [14]. A mating pool is set to be the whole population, then a trial solution is generated by using the current solution and other two different solutions from the mating pool, the operator is from the DE [31]. Next the trial solution is repaired to be feasible. Finally the trial solution is mutated by the polynomial mutation [6]. As in [14], the control parameters are $F = 0.5$, $p_m = \frac{1}{n}$, $\eta = 20$.

2) EDA based reproduction: For the EDA based offspring reproduction, we use the same reproduction procedure as RM-MEDA [33]. Firstly the whole population is partitioned into several disjoint clusters by the Local PCA algorithm [34]. Then a probability model is built for each cluster. Finally a set of candidate solutions are sampled from the probability model. As in [33], the control parameters are: the number of clusters is 5, the maximum iterations of Local PCA is 50, and the extension rate is 0.25 in sampling.

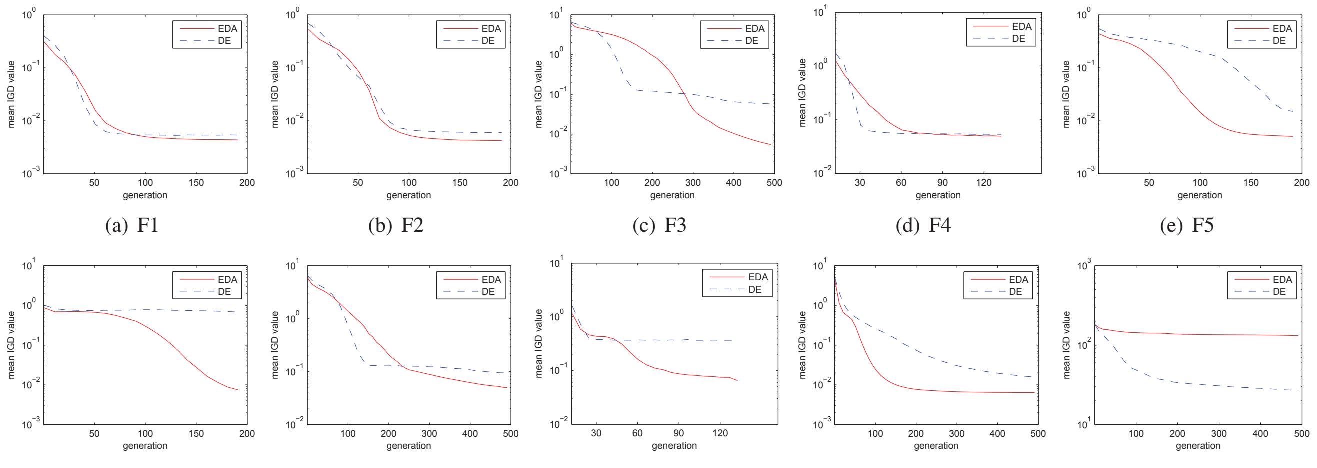
III. EXPERIMENTAL RESULTS

In this section we test CPS-MOEA on 10 benchmark functions, denoted as F1-F10, which are introduced in [33]. The dimension of the instances is $n = 30$. The maximal number of function evaluations (FES) is $20,000$ on F1, F2, F5, and F6, $40,000$ on F4 and F8, as well as $100,000$ on F3, F7, F9, and F10. The population size is set as $N = 100$ on F1, F2, F5, F6, and 200 on F3, F4 and F7-F10. And the value of generated offspring solutions for each solutions is set as $M = 3$. Each algorithm independently runs for 30 times

TABLE II

THE MEAN AND STD. IGD VALUES OBTAINED BY CPS-MOEA WITH TWO DIFFERENT REPRODUCTION OPERATORS DE AND EDA AFTER 20%, 40%, 60%, 80%, AND 100% FES OVER 30 RUNS ON F1-F10. '+', '-', AND '$\sim$' DENOTES THAT CPS-MOEA-DE PERFORMS BETTER, WORSE, OR SIMILAR TO CPS-MOEA-EDA ACCORDING TO THE WILCOXON TEST WITH THE 5% SIGNIFICANCE LEVEL.

	Operator	20%	40%	60%	80%	100%	+/-/($\sim$)
F1	EDA	4.06e-02$_{2.86e-02}$	5.99e-03$_{6.14e-04}$	**4.71e-03**$_{1.71e-04}$	**4.43e-03**$_{1.10e-04}$	**4.36e-03**$_{1.27e-04}$	
	DE	**2.08e-02**$_{1.69e-02}$[+]	**5.58e-03**$_{2.41e-04}$[+]	5.38e-03$_{1.74e-04}$[−]	5.40e-03$_{2.06e-04}$[−]	5.35e-03$_{1.54e-04}$[−]	2/3/0
F2	EDA	1.53e-01$_{6.89e-02}$	**7.58e-03**$_{1.47e-03}$	**4.67e-03**$_{3.10e-04}$	**4.31e-03**$_{1.57e-04}$	**4.20e-03**$_{9.31e-05}$	
	DE	**1.03e-01**$_{1.46e-01}$[+]	9.98e-03$_{7.83e-03}$[−]	6.33e-03$_{5.22e-04}$[−]	6.04e-03$_{2.62e-04}$[−]	5.95e-03$_{2.19e-04}$[−]	1/4/0
F3	EDA	3.17e+00$_{2.35e-01}$	9.39e-01$_{4.79e-01}$	**4.45e-02**$_{3.62e-02}$	**1.04e-02**$_{2.62e-03}$	**5.20e-03**$_{1.31e-03}$	
	DE	**1.40e+00**$_{7.69e-01}$[+]	**1.21e-01**$_{5.00e-03}$[+]	9.88e-02$_{4.57e-03}$[−]	6.50e-02$_{4.19e-03}$[−]	5.62e-02$_{3.97e-03}$[−]	2/3/0
F4	EDA	1.95e-01$_{2.23e-02}$	6.49e-02$_{7.06e-03}$	**5.34e-02**$_{1.54e-03}$	**5.07e-02**$_{1.36e-03}$	**4.93e-02**$_{1.50e-03}$	
	DE	**6.28e-02**$_{4.49e-03}$[+]	**5.57e-02**$_{1.48e-03}$[+]	5.43e-02$_{1.87e-03}$[−]	5.35e-02$_{1.35e-03}$[−]	5.36e-02$_{1.48e-03}$[−]	2/3/0
F5	EDA	**2.41e-01**$_{5.79e-02}$	**3.69e-02**$_{2.87e-02}$	**7.79e-03**$_{2.78e-03}$	**5.40e-03**$_{3.16e-04}$	5.05e-03$_{1.93e-04}$	
	DE	3.61e-01$_{6.32e-02}$[−]	2.67e-01$_{8.02e-02}$[−]	1.57e-01$_{1.01e-01}$[−]	3.86e-02$_{5.93e-02}$[−]	1.34e-02$_{2.26e-02}$[−]	0/5/0
F6	EDA	**6.95e-01**$_{5.75e-02}$	**4.85e-01**$_{1.46e-01}$	**1.37e-01**$_{1.57e-01}$	**1.67e-02**$_{1.52e-02}$	**6.81e-03**$_{1.25e-03}$	
	DE	7.44e-01$_{2.52e-02}$[−]	7.57e-01$_{5.17e-02}$[−]	7.61e-01$_{1.56e-01}$[−]	7.30e-01$_{2.29e-01}$[−]	6.59e-01$_{3.22e-01}$[−]	0/5/0
F7	EDA	1.39e+00$_{5.89e-01}$	2.10e-01$_{2.31e-01}$	**8.88e-02**$_{6.10e-02}$	**6.22e-02**$_{2.38e-02}$	**4.89e-02**$_{1.34e-02}$	
	DE	**7.40e-01**$_{7.33e-01}$[+]	**1.32e-01**$_{1.75e-03}$[+]	1.24e-01$_{2.26e-03}$[−]	1.08e-01$_{6.26e-03}$[−]	9.33e-02$_{5.07e-03}$[−]	2/3/0
F8	EDA	4.33e-01$_{3.35e-02}$	**1.65e-01**$_{8.43e-02}$	**9.31e-02**$_{5.73e-02}$	**8.27e-02**$_{7.75e-02}$	**6.50e-02**$_{1.16e-02}$	
	DE	**3.69e-01**$_{6.77e-03}$[+]	3.66e-01$_{2.00e-03}$[−]	3.66e-01$_{1.05e-03}$[−]	3.69e-01$_{2.33e-02}$[−]	3.68e-01$_{4.39e-02}$[−]	1/4/0
F9	EDA	**2.53e-02**$_{8.58e-03}$	**7.78e-03**$_{6.62e-03}$	**6.71e-03**$_{6.59e-03}$	**6.46e-03**$_{6.59e-03}$	**6.40e-03**$_{6.54e-03}$	
	DE	2.61e-01$_{5.35e-02}$[−]	7.48e-02$_{3.74e-02}$[−]	3.06e-02$_{1.67e-02}$[−]	1.97e-02$_{9.05e-03}$[−]	1.55e-02$_{7.89e-03}$[−]	0/5/0
F10	EDA	1.44e+02$_{9.99e+00}$	1.38e+02$_{1.15e+01}$	1.35e+02$_{1.07e+01}$	1.34e+02$_{1.07e+01}$	1.32e+02$_{1.21e+01}$	
	DE	**4.88e+01**$_{1.32e+01}$[+]	**3.37e+01**$_{9.23e+00}$[+]	**3.07e+01**$_{7.96e+00}$[+]	**2.86e+01**$_{7.47e+00}$[+]	**2.66e+01**$_{7.10e+00}$[+]	5/0/0
sum							15/35/0



(a) F1 (b) F2 (c) F3 (d) F4 (e) F5

(f) F6 (g) F7 (h) F8 (i) F9 (j) F10

Fig. 3. The mean IGD values versus generations obtained by CPS-MOEA with DE and EDA reproduction operators over 30 runs.

on each test instance. We use *inverted generational distance (IGD)* [35] as the performance metric, and 10000 distributed points [20] are generated as the uniformly distributed Pareto optimal points in PF.

A. Influence of Classifier

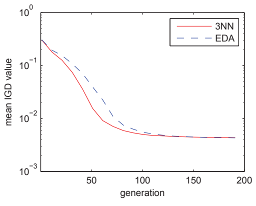
In this experiment, we use the EDA based reproduction operator, and test CPS-MOEA with CART, 3NN, and 5NN. For simplicity, we directly use CART, 3NN, and 5NN to denote CPS-MOEA with CART, CPS-MOEA with 3NN, and CPS-MOEA with 5NN.

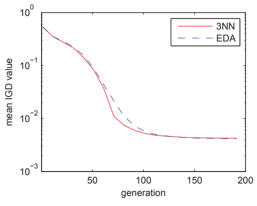
Table I shows the mean(std) IGD value of these three CPS-MOEA variants after 20%, 40%, 60%, 80%, and 100% FES over 30 runs.

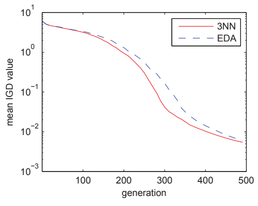
Table I clearly shows that on average, CPS-MOEA with KNN works better than CPS-MOEA with CART. The reason might be that the dimension of the feature vector, i.e., decision variable vector, is relatively high, while the number of training data points, i.e., the population size, is relatively small. In such cases, CART is sensitive to the training data points and may fail to build a high quality model. KNN, on the contrary,

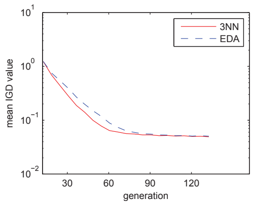
TABLE III
THE MEAN AND STD. IGD VALUES OBTAINED BY CPS-MOEA AND RM-MEDA AFTER 20%, 40%, 60%, 80%, AND 100% FES OVER 30 RUNS ON F1-F10. '+', '-', AND '∼' DENOTES THAT RM-MEDA PERFORMS BETTER, WORSE, OR SIMILAR TO CPS-MOEA ACCORDING TO THE WILCOXON TEST WITH THE 5% SIGNIFICANCE LEVEL.

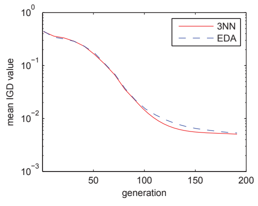
	Algorithm	20%	40%	60%	80%	100%	+/-/(∼)
F1	CPS-MOEA	**4.06e-02**$_{2.86e-02}$	**5.99e-03**$_{6.14e-04}$	**4.71e-03**$_{1.71e-04}$	4.43e-03$_{1.10e-04}$	4.36e-03$_{1.27e-04}$	
	RM-MEDA	7.46e-02$_{4.52e-02}$[−]	7.74e-03$_{2.33e-03}$[−]	4.86e-03$_{3.31e-04}$[−]	**4.39e-03**$_{1.37e-04}$[+]	**4.28e-03**$_{1.12e-04}$[+]	2/3/0
F2	CPS-MOEA	**1.53e-01**$_{6.89e-02}$	**7.58e-03**$_{1.47e-03}$	**4.67e-03**$_{3.10e-04}$	4.31e-03$_{1.57e-04}$	4.20e-03$_{9.31e-05}$	
	RM-MEDA	1.65e-01$_{7.01e-02}$[−]	1.20e-02$_{8.16e-03}$[−]	4.73e-03$_{2.87e-04}$[−]	**4.24e-03**$_{8.53e-05}$[+]	**4.14e-03**$_{7.74e-05}$[+]	2/3/0
F3	CPS-MOEA	**3.17e+00**$_{2.35e-01}$	**9.39e-01**$_{4.79e-01}$	**4.45e-02**$_{3.62e-02}$	**1.04e-02**$_{2.62e-03}$	**5.20e-03**$_{1.31e-03}$	
	RM-MEDA	3.34e+00$_{2.16e-01}$[−]	1.31e+00$_{4.72e-01}$[−]	1.68e-01$_{2.08e-01}$[−]	1.43e-02$_{5.54e-03}$[−]	5.73e-03$_{1.51e-03}$[−]	0/5/0
F4	CPS-MOEA	**1.95e-01**$_{2.23e-02}$	**6.49e-02**$_{7.06e-03}$	**5.34e-02**$_{1.54e-03}$	**5.07e-02**$_{1.36e-03}$	**4.93e-02**$_{1.50e-03}$	
	RM-MEDA	2.78e-01$_{4.58e-02}$[−]	9.08e-02$_{1.68e-02}$[−]	5.63e-02$_{2.80e-03}$[−]	5.16e-02$_{1.84e-03}$[−]	5.04e-02$_{1.64e-03}$[−]	0/5/0
F5	CPS-MOEA	2.41e-01$_{5.79e-02}$	**3.69e-02**$_{2.87e-02}$	**7.79e-03**$_{2.78e-03}$	**5.40e-03**$_{3.16e-04}$	**5.05e-03**$_{1.93e-04}$	
	RM-MEDA	**2.35e-01**$_{4.83e-02}$[+]	4.02e-02$_{2.64e-02}$[−]	9.58e-03$_{5.50e-03}$[−]	6.13e-03$_{1.99e-03}$[−]	5.16e-03$_{5.02e-04}$[−]	1/4/0
F6	CPS-MOEA	6.95e-01$_{5.75e-02}$	**4.85e-01**$_{1.46e-01}$	**1.37e-01**$_{1.57e-01}$	**1.67e-02**$_{1.52e-02}$	**6.81e-03**$_{1.25e-03}$	
	RM-MEDA	**6.95e-01**$_{2.73e-02}$[+]	5.14e-01$_{1.23e-01}$[−]	1.61e-01$_{1.17e-01}$[−]	2.79e-02$_{2.12e-02}$[−]	1.06e-02$_{5.32e-03}$[−]	1/4/0
F7	CPS-MOEA	**1.39e+00**$_{5.89e-01}$	2.10e-01$_{2.31e-01}$	**8.88e-02**$_{6.10e-02}$	**6.22e-02**$_{2.38e-02}$	**4.89e-02**$_{1.34e-02}$	
	RM-MEDA	1.55e+00$_{4.49e-01}$[−]	**2.03e-01**$_{1.47e-01}$[+]	9.19e-02$_{4.95e-02}$[−]	6.92e-02$_{2.39e-02}$[−]	5.38e-02$_{1.64e-02}$[−]	1/4/0
F8	CPS-MOEA	4.33e-01$_{3.35e-02}$	**1.65e-01**$_{8.43e-02}$	**9.31e-02**$_{5.73e-02}$	8.27e-02$_{7.75e-02}$	**6.50e-02**$_{1.16e-02}$	
	RM-MEDA	**4.12e-01**$_{3.03e-02}$[+]	2.63e-01$_{7.79e-02}$[−]	1.06e-01$_{2.37e-02}$[−]	**7.56e-02**$_{6.70e-03}$[+]	6.65e-02$_{4.58e-03}$[−]	2/3/0
F9	CPS-MOEA	**2.53e-02**$_{8.58e-03}$	7.78e-03$_{6.62e-03}$	6.71e-03$_{6.59e-03}$	6.46e-03$_{6.59e-03}$	6.40e-03$_{6.54e-03}$	
	RM-MEDA	4.94e-02$_{1.35e-02}$[−]	**7.03e-03**$_{2.86e-03}$[+]	**5.30e-03**$_{2.83e-03}$[+]	**4.91e-03**$_{2.81e-03}$[+]	**4.77e-03**$_{2.82e-03}$[+]	4/1/0
F10	CPS-MOEA	**1.44e+02**$_{9.99e+00}$	**1.38e+02**$_{1.15e+01}$	**1.35e+02**$_{1.07e+01}$	**1.34e+02**$_{1.07e+01}$	**1.32e+02**$_{1.21e+01}$	
	RM-MEDA	1.48e+02$_{7.29e+00}$[−]	1.43e+02$_{7.76e+00}$[−]	1.37e+02$_{8.00e+00}$[−]	1.35e+02$_{6.83e+00}$[−]	1.32e+02$_{7.01e+00}$[−]	0/5/0
sum							12/38/0

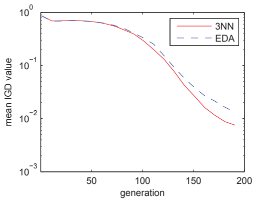

(a) F1

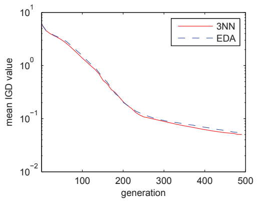

(b) F2

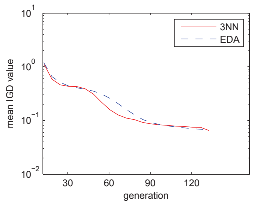

(c) F3

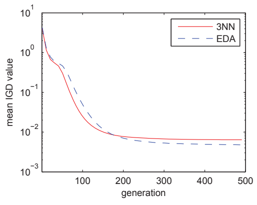

(d) F4


(e) F5


(f) F6


(g) F7


(h) F8


(i) F9

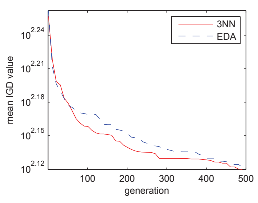

(j) F10

Fig. 4. The mean IGD values versus generations obtained by CPS-MOEA with EDA and RM-MEDA reproduction operators over 30 runs.

is more stable than CART on the given training data points. For KNN, CPS-MOEA with 3NN outperforms CPS-MOEA with 5NN. This experiment suggests that in our case, a simple classifier may work more stable than a complicated classifier.

B. Influence of Reproduction Operator

This section considers the influence of the reproduction operator. CPS-MOEA with 3NN is utilized with the DE and EDA based reproduction operators. For simplicity, we directly use DE, and EDA to denote CPS-MOEA with DE and EDA reproduction operators respectively.

Table II shows the mean(std) IGD value of these two CPS-MOEA variants after 20%, 40%, 60%, 80%, and 100% FES over 30 runs. Fig. 3 plots the run time performance in terms of the IGD values obtained by the two variants on the 10 instances.

From Table II, we can see that on all the 50 comparisons, CPS-MOEA-DE performs better than CPS-MOEA-EDA on 15 comparisons, and worse than CPS-MOEA-EDA on the other 35 comparisons according to the Wilcoxon test. CPS-MOEA-DE outperforms CPS-MOEA-EDA on F10 throughout the run-

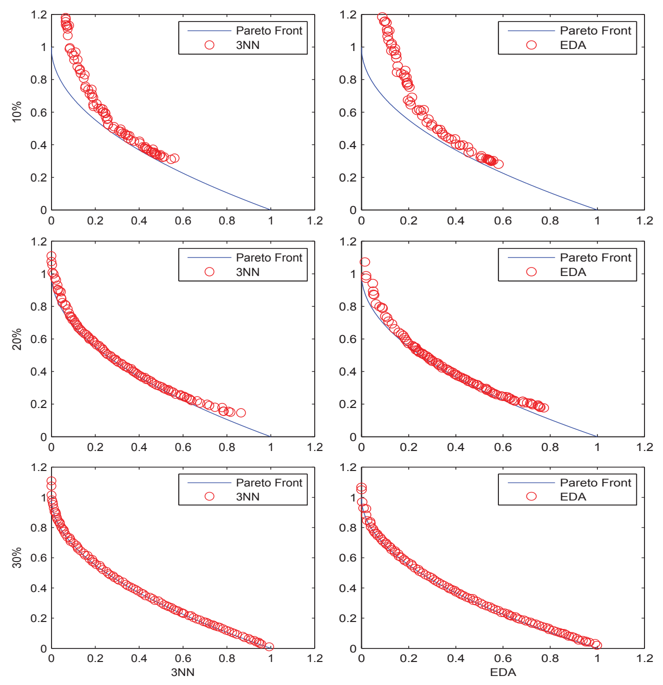


Fig. 5. The median (according to the IGD metric values) approximations obtained by 3NN and EDA after 10%, 20%, 30% FES on F1.

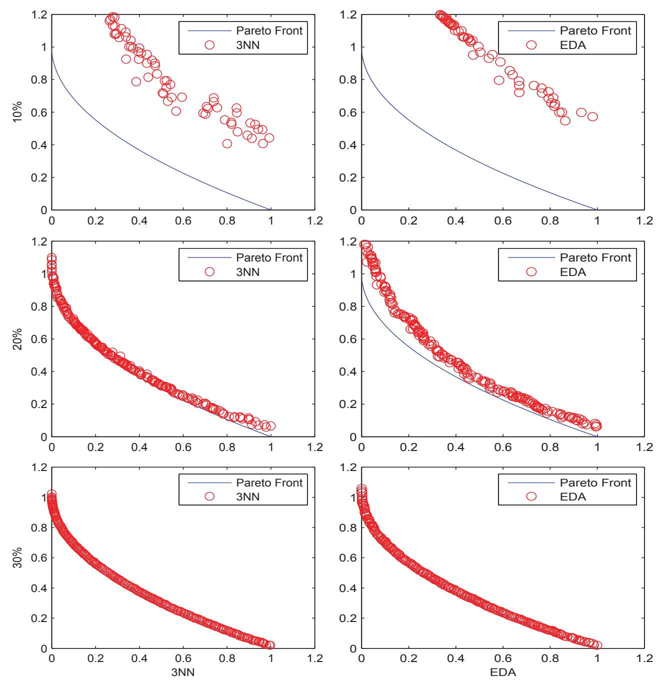


Fig. 6. The median (according to the IGD metric values) approximations obtained by 3NN and EDA after 10%, 20%, 30% FES on F9.

s. On F1, F2, F3, F4, F7, and F8, CPS-MOEA-DE works better in the early stages. This can also be confirmed from Fig. 3. The reason might be that in the early stages, the populations do not converge and the EDA based reproduction operator is not able to build high quality models to approximate the population in the decision space. In the later stages when the populations show some structures, the EDA based reproduction operator can successfully detect these structures and guide the search more efficiently. This is consistent with our previous study.

C. Comparison with Other Algorithms

The above sections study the influences of the classifier and the reproduction operator, and the results suggest that CPS-MOEA with 3NN and EDA based reproduction operator performs the best. In this section, we compare our proposed approach with a state-of-the-art method, RM-MEDA [33].

Table III shows the mean(std) IGD value obtained by CPS-MOEA and RM-MEDA after 20%, 40%, 60%, 80%, and 100% FES over 30 runs. Fig. 4 plots the run time performance in terms of the IGD values obtained by the two variants on the 10 instances. Figs. 5 and 6 plot the final obtained populations of the median runs according to the IGD values after 10%, 20%, and 30% FES on the F1 and F2 instances.

The only difference between CPS-MOEA and RM-MEDA is that the CPS strategy is utilized to filter 'bad' candidate offspring solutions in CPS-MOEA. The statistical results in Table III indicate that the CPS strategy can significantly improve the performance of RM-MEDA. Actually, on all the 50 comparisons, CPS-MOEA outperforms RM-MEDA on 38 ones.

The populations obtained by the two algorithms in Fig. 4, also shows that on most test instances CPS-MOEA converges faster than RM-MEDA. In order to visualize the final solutions' obtained, we choose two test instances F1 and F9 to draw their approximations in Figs. 5 and 6.

IV. Conclusions

This paper has proposed a classification and Pareto Domination based multiobjective optimization evolutionary algorithm named as CPS-MOEA. In this algorithm, a *classification based preselection (CPS)* is introduced to filter possible 'bad' candidate offspring solutions. The CPS strategy is implemented into a Pareto domination based multiobjective evolutionary algorithm framework. The influences of the classifiers and the reproduction operators are empirically studied, and the proposed algorithm with the best classifier and reproduction operator is also compared with a state-of-the-art algorithm. The statistical results suggest that the CPS strategy is able to improve the algorithm performance significantly on the given test instances.

It should be noted that the work reported in this paper is very preliminary and there are some problems left. For example, how to design a more efficient CPS strategy, and how to apply the CPS strategy to the indicator based approaches and the decomposition based approaches, are worth for further investigation.

Acknowledgment

This work is supported by China National Basic Research Program of China (2011CB707104), China National Instrumentation Program (2012YQ180132), the National Natural

Science Foundation of China (61273313, 61175063), the Science and Technology Commission of Shanghai Municipality (14DZ2260800), the National Science Foundation of Guangdong Province (S2013010014108) and the Shenzhen Scientific Research Project (JCYJ20130401095559825).

REFERENCES

[1] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, 2001.

[2] B. Thomas, *Evolutionary algorithms in theory and practice : evolution strategies, evolutionary programming, genetic algorithms*. Oxford Univ. Press, 1996.

[3] D. Simon, *Evolutionary Optimization Algorithms*. John Wiley & Sons, 2013.

[4] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthanb, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, pp. 32–49, 2011.

[5] J. Knowles and D. Corne, "Approximating the nondominated front using the pareto archived evolution strategy," *Evolutionary Computation*, vol. 8, no. 2, pp. 149–172, 2000.

[6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[7] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," *Evolutionary Methods for Design Optimisation and Control*, pp. 95–100, 2001.

[8] E. Zitzler and S. Knzli, "Indicator-based selection in multiobjective search," in *Proc. 8th International Conference on Parallel Problem Solving from Nature*, vol. 3242, 2004, pp. 832–842.

[9] M. Basseur and E. Zitzler, "Handling uncertainty in indicator-based multiobjective optimization," *International Journal of Computational Intelligence Research*, vol. 2, no. 3, pp. 255–272, 2006.

[10] J. Bader and E. Zitzler, "HypE: an algorithm for fast hypervolume-based manyobjective optimization," *Tech. Rep. TIK 286, Computer Engineering and Networks Laboratory, ETH Zurich*, vol. 19, no. 1, pp. 45–76, 2010.

[11] H. Trautmann, T. Wagner, and D. Brockhoff, "R2-EMOA: Focused multiobjective search using R2-indicator-based selection," *Learning and Intelligent Optimization*, pp. 70–74, 2013.

[12] D. H. Phan and J. Suzuki, "R2-IBEA: R2 indicator based evolutionary algorithm for multiobjective optimization," in *2013 IEEE Congress on Evolutionary Computation (CEC)*, 2013, pp. 1836–1845.

[13] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[14] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.

[15] B. Alatas, E. Akin, and A. Karci, "MODENAR: multi-objective differential evolution algorithm for mining numeric association rules," *Applied Soft Computing*, vol. 8, no. 1, 2008.

[16] Y. Li, A. Zhou, and G. Zhang, "An MOEA/D with multiple differential evolution mutation operators," in *2014 IEEE Congress on Evolutionary Computation (CEC)*, 2014, pp. 397–404.

[17] V. Huang, P. Suganthan, and J. Liang, "Comprehensive learning particle swarm optimizer for solving multiobjective optimization problems,," *International Journal of Intelligent Systems*, vol. 21, no. 2, pp. 209–226, 2006.

[18] A. Zhou, Q. Zhang, and G. Zhang, "A multiobjective evolutionary algorithm based on decomposition and probability model," in *2012 IEEE Congress on Evolutionary Computation (CEC).*, 2012, pp. 1–8.

[19] A. W. Iorio and X. Li, "Rotated problems and rotationally invariant crossover in evolutionary multi-objective optimization," *International Journal of Computational Intelligence and Applications*, vol. 7, no. 2, pp. 149–186, 2008.

[20] A. Zhou, Q. Zhang, and Y. Jin, "Approximating the set of Pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1167 – 1189, 2009.

[21] H. Zhang, S. Song, A. Zhou, and X. Gao, "A clustering based multiobjective evolutionary algorithm," in *2014 IEEE Congress on Evolutionary Computation (CEC)*, 2014, pp. 723–730.

[22] S. Bandaru, A. H. Ng, and K. Deb, "On the performance of classification algorithms for learning pareto-dominance relations," in *2014 IEEE Congress on Evolutionary Computation (CEC)*, 2014, pp. 1139–1146.

[23] X.-F. Lu and K. Tang, "Classication- and regression-assisted differential evolution for computationally expensive problems," *Journal of Computer Science and Technology*, vol. 27, no. 5, pp. 1024–1034, 2012.

[24] X. Lu, K. Tang, and X. Yao, "Classification-assisted differential evolution for computationally expensive problems," in *2011 IEEE Congress on Evolutionary Computation (CEC)*, 2011, pp. 1986 – 1993.

[25] S. M. Weiss and I. Kapouleas, "An empirical comparison of pattern recognition, neural nets, and machine learning classification methods," in *In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, 1989, pp. 781–787.

[26] D. Michie, D. J. Spiegelhalter, and C. Taylor, "Machine learning, neural and statistical classification," *Ellis Horwood*, 1994.

[27] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[28] L. Breiman, J. Friedman, C. J. Stone, and R. Olshen, *Classification and regression trees*. New York:Chapman and Hall, 1984.

[29] D. Coomans and D. Massart, "Alternative k-nearest neighbour rules in supervised pattern recognition : Part 1. k-Nearest neighbour classification by using alternative voting rules," *Analytica Chimica Acta*, vol. 136, pp. 15–27, 1982.

[30] F. David and O. Richard, "Gini means," *Mathematical Association of America*, vol. 93, no. 8, pp. 603–607, 1986.

[31] R. Storn and K. Price, "Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[32] P. Larraaga and J. A. Lozano, *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, 2002.

[33] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 41–63, 2008.

[34] N. Kambhatla and T. K. Leen, "Dimension reduction by local principal component analysis," *Neural computation*, vol. 9, no. 7, pp. 1493–1516, 1997.

[35] A. Zhou, Q. Zhang, Y. Jin, E. Tsang, and T. Okabe, "A model-based evolutionary algorithm for bi-objective optimization," in *2005 IEEE Congress on Evolutionary Computation*, vol. 3, 2005, pp. 2568 – 2575.