

Hybrid Multiobjective Differential Evolution Incorporating Preference Based Local Search *

Ning Dong¹ and Yuping Wang^{2†}

¹ School of Science, School of Computer Science and Technology,
Xidian University,
Xi'an, 710071, China

E-mail: dongning@snnu.edu.cn

² School of Computer Science and Technology,
Xidian University ,
Xi'an, 710071, China

E-mail: ywang@xidian.edu.cn

Received 31 May 2012

Accepted 29 March 2013

Abstract

The performance of Differential Evolution (DE) for multiobjective optimization problems (MOPs) can be greatly enhanced by hybridizing with other techniques. In this paper, a new hybrid DE incorporating preference based local search is proposed. In every generation, a set of nondominated solutions is generated by DE operation. Usually these solutions distribute unevenly along the obtained nondominated set. To get solutions in the sparse region of the nondominated set, a mini population and preference based local search algorithm is specifically designed, and is used to exploit the sparse region by optimizing an achievement scalarizing function (ASF) with the dynamically adjusted reference point. As a result, multiple solutions in the sparse region can be obtained. Moreover, to retain uniformly spread nondominated solutions, an improved ϵ -dominance strategy, which would not delete the extreme points found during the evolution, is proposed to update the external archive set. Finally, numerical results and comparisons demonstrate the efficiency of the proposed algorithm.

Keywords: multiobjective optimization, hybrid differential evolution, preference, sparse region, dynamical adjustment

1. Introduction

Many real world problems involve optimization of two or more objectives. Very often, the objectives contradict each other, and improvement of one objective may lead to deterioration of another. Thus, there is no single optimal solution that is the best

with respect to all objectives, but rather a set of alternative solutions. The best tradeoffs among the objectives are so called the nondominated solutions or Pareto optimal solutions, from which the decision maker (DM) may select one. For many real world multi-objective problems, there is no any information about the Pareto solutions beforehand, and DM

*This research is supported by the National Natural Science Foundation of China (No. 61272119).

†Corresponding author.

has no any preference to the Pareto solutions. Thus a method that can find as many as possible Pareto solutions scattered evenly in the entire Pareto set is considered as an efficient one.

Evolutionary algorithms (EAs) are a class of stochastic and population based optimization techniques, and can find multiple optimal solutions in a single run, they are ideally suited for MOPs. In the past of several decades, many efficient multiobjective evolutionary algorithms (MOEAs) were proposed, e.g., NSGAI¹, SPEA2², PAES³, etc. DE⁴ is a simple but powerful evolutionary algorithm, and it is very robust for solving single-objective optimizations. Later on, researchers extended it to handle MOPs⁵⁻¹¹. Among them, Abbass et al.⁵ first applied DE to MOPs and proposed Pareto-frontier Differential Evolution(PDE). In PDE, the offspring is produced using DE, and then compared with its parent using dominant relationship. The dominated solution is discarded and the nondominated one is retained to go to the next generation. A Pareto-based differential evolution for MOPs was proposed in Ref.6, where the algorithm incorporates nondominated sorting and ranking selection procedure proposed in Ref. 1 into DE. Tobič et al.⁷ presented DEMO algorithm for multiobjective optimization. This algorithm maintains only one population, and incorporates two crucial mechanisms. One is the immediate replacement of the parent with the candidate that dominates it, which enables a fast convergence towards the Pareto front. Another is the Pareto-based ranking and crowding distance sorting, which can promote the uniform spread of solutions. The algorithms in Refs. 6 and 7 used nondominated sorting and crowding distance to evaluate candidates, which have a relative high cost of computation. Laumanns et al.¹² proposed a relaxed form of dominance, named the ε -dominance, for multiobjective evolutionary algorithms. This mechanism acts as an archiving strategy to ensure properties of diversity among the nondominated solutions found. Then, several DE algorithms incorporated with ε -dominance were proposed to get a well-distributed set of solutions along the Pareto front without computing the crowding distance, such as ε -MyDE⁸ and ε -MOEA⁹. Though an algorithm with

ε -dominance can maintain good distribution of solutions in archive, it obviously has the limitation of losing both the extreme points and the points in almost horizontal or vertical segments of the Pareto front. To overcome these drawbacks and maintain the good properties of ε -dominance, a Pareto-adaptive ε -dominance (pa ε -dominance)¹⁰ was proposed, where an initial Pareto front approximation F must be generated firstly. The quality of efficient points in F is critical for the final performance. However, it is difficult to get any information of Pareto front before starting to solve a MOP. Although the pa ε -dominance can remedy some limitations of ε -dominance, it may also lose the extreme points for some MOPs¹³. pa ε -ODEMO¹³ was proposed based on pa ε -dominance¹⁰, while it took an additional storage technique in every generation to retain the extreme points found so far.

The above MOEAs belong to posteriori methods, where the opinions of the DM are not taken into consideration. Another widely-used class of multiobjective optimization methods is interactive methods. In interactive methods, the DM takes actively part in the solution search process and directs the search according to her/his preference. A common way for a DM to express her/his preference is to use a reference point, which is a vector consisting of the components that the DM wishes to achieve for all objective functions, respectively. Once a reference point is determined, the most usual way to find a solution is to optimize an ASF, and the optimal solutions of ASF are weakly, properly or Pareto optimal solutions. There are many reference point based interactive methods¹⁴⁻¹⁷ for MOPs. In Ref. 14, a reference point based preference strategy was incorporated into MOEA to find a set of solutions near the reference points. The preference strategy is performed by a biased crowding scheme defined by the reference point but do not use an ASF, which is an essential concept in the context of reference point. A variation of the concept of Pareto dominance, called g-dominance¹⁵, was proposed. This concept leads the algorithm search the approximate Pareto front around the area of the preferred point without using any ASF, and it can be easily used with any MOEAs. L. Thiele et al. proposed a preference-based evo-

lutionary approach, called PBEA¹⁶, which incorporates preference information into an indicator-based evolutionary algorithm IBEA¹⁸ and aims to produce a good, probably small set of approximate efficient solutions related to the reference point. A parallel multiple reference points approach for multiobjective optimization was proposed in Ref. 19. These multiple reference points are generated uniformly on the cutting plane which is defined by the estimated bounds of the Pareto front. Multiple reference points are used to establish multiple subproblems to be optimized by PBEA in parallel. However, the quality of the estimated bounds of the Pareto front will greatly influence the performance of the parallel approach. To improve the convergence of MOEAs, local search in multiobjective scenarios is a promising scheme^{20–23}. However, the efficiency of the local search depends heavily on the choice of the single objective function to be optimized. ASF in interactive methods is a good choice for two reasons: one is it has the minimal solution among Pareto optimal solutions of the original problem, the other is ASF can lead the local search towards the preference region of the reference point. In Ref. 23, a hybrid approach was proposed by incorporating ASF based local search to NSGA-II. But in this hybrid algorithm, the local search is performed on ASF established by the randomly selected reference point with probability, where the reference point may be in dense region, and the local search makes no contribution for adding solutions in sparse region.

In this paper, we suggest a hybrid approach which incorporates idea from interactive multiobjective optimization into DE, and intends to get a set of sufficiently wide and evenly scattered representative set of Pareto front. In DE process, we use a hybrid mutation scheme to improve the exploitation ability of DE mutation. Then, an ASF is used to perform local search, where the reference point and weighting vector in ASF are adaptively calculated by the Pareto optimal solutions obtained by DE, which make the ASF prone to search the sparse region of Pareto front the reference point prefers. In the end, to get a sufficiently wide and evenly scattered representative set of solutions of Pareto front, a minor revision on ϵ -dominance scheme is made which can overcome its

deficiency of losing the extreme points, while does not need any additional extreme points storage technique.

The rest of this paper is organized as follows. In section 2, we give some preliminaries of MOPs, and briefly introduce DE algorithms and ASF. A hybrid local search based multiobjective DE is exhibited in section 3. In section 4, the experiment results and comparisons with other algorithms are given. Finally, the conclusion is made.

2. Preliminary

2.1. MOP and some related definitions

Without loss of generality, a multiobjective optimization problem can be stated as follows:

$$\begin{cases} \min & f(x) = (f_1(x), f_2(x), \dots, f_m(x))^T \\ \text{s.t.} & x \in \Omega, \end{cases} \quad (1)$$

where Ω is the decision space, $f : \Omega \rightarrow R^m$ consists of m real-valued objective functions and R^m is called the objective space. Generally, the decision space Ω is a hyper-rectangle, and denoted by $\Omega = \{x \in R^n | a \leq x \leq b\}$, where $a, b \in R^n$.

Definition 1 (Pareto Dominance) A solution x is said to dominate y , denoted as $x \preceq y$, if $f_i(x) \leq f_i(y)$ for all $i = 1, 2, \dots, m$ and $f_j(x) < f_j(y)$ for at least one index $j \in \{1, 2, \dots, m\}$; x is said to strongly dominate y , denoted as $x \prec y$, if $f_i(x) < f_i(y)$ for all $i = 1, 2, \dots, m$.

A point x^* is called a (weakly) Pareto optimal solution of (1) if there is no point $x \in \Omega$ that (strongly) dominates x^* . The image of x^* , $z^* = f(x^*)$ in objective space, is called Pareto optimal vector. The set of all the Pareto optimal solutions is called the Pareto optimal set (PS), and correspondingly the set of Pareto optimal vectors is called the Pareto optimal front (PF). Consequently, the optimization goal in MOP is not only to search solutions as close to the PF as possible but also to find solutions as diverse as possible in the obtained nondominated front.

Because it is often useful to know the ranges of objective vectors in the PF, the ideal and nadir vectors are defined below:

Definition 3 (Ideal vector) The ideal vector $z^* = (z_1^*, z_2^*, \dots, z_m^*)$ is the vector that minimizes each objective function individually, i.e.,

$$z_i^* = \min_{x \in \Omega} f_i(x) = \min_{x \in PS} f_i(x), \quad i = 1, \dots, m.$$

Definition 4 (Nadir vector) The nadir vector $z^{nad} = (z_1^{nad}, z_2^{nad}, \dots, z_m^{nad})$ is the vector that maximizes each objective function in Pareto optimal set individually, i.e.,

$$z_i^{nad} = \max_{x \in PS} f_i(x), \quad i = 1, \dots, m.$$

Laumanns et al.¹² proposed two different schemes to implement ε -dominance: the additive and the multiplicative approaches. In this paper, we use the additive scheme defined as follows.

Definition 5(ε -dominance) Let $f, g \in R^m$. Then f is said to ε -dominate g for some $\varepsilon > 0$, denoted as $f \prec_\varepsilon g$, if and only if for all $i \in \{1, 2, \dots, m\}$, $f_i - \varepsilon \leq g_i$.

2.2. Differential evolution

Differential evolution⁴ is a parallel stochastic direction search method which follows the general procedure of an evolutionary algorithm. Initial population P with NP individuals are produced randomly in decision space. For each individual $x_i \in P (i = 1, 2, \dots, NP)$ at each generation, DE then employs the three operations below in turn.

Mutation: A trial vector v_i for each individual x_i is generated based on the current parent population via certain mutation strategy. The most frequently used mutation strategies are:

1) “DE/rand/1”⁴

$$v_i = x_{r1} + F \cdot (x_{r2} - x_{r3}), \quad (2)$$

2) “DE/best/1”²⁴:

$$v_i = x_{best} + F \cdot (x_{r1} - x_{r2}) \quad (3)$$

3) “DE/current-to-best/1”²⁴:

$$v_i = x_i + F \cdot (x_{best} - x_i) + F \cdot (x_{r1} - x_{r2}) \quad (4)$$

4) “DE/rand/2”²⁴:

$$v_i = x_{r1} + F \cdot (x_{r2} - x_{r3}) + F \cdot (x_{r4} - x_{r5}) \quad (5)$$

5) “DE/best/2”²⁴:

$$v_i = x_{best} + F \cdot (x_{r1} - x_{r2}) + F \cdot (x_{r3} - x_{r4}) \quad (6)$$

where $r1, r2, r3, r4$ and $r5$ are mutually different integers randomly selected from $\{1, 2, \dots, NP\} \setminus \{i\}$, x_{best} represents the best individual in the current population, $F \in [0, 2]$ is a scale factor which controls the amplification of the differential variation.

Crossover: To increase the diversity, the following offspring vector u_i is then generated from its parent x_i and mutation vector v_i by

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{if } \text{rand}(0,1) < Cr \text{ or } j = j_{rand} \\ x_{i,j}, & \text{else} \end{cases} \quad (7)$$

for $j = 1, 2, \dots, n$, where $u_{i,j}$, $v_{i,j}$ and $x_{i,j}$ are the j -th components of u_i , v_i and x_i , respectively, $Cr \in [0, 1]$ is crossover probability and $j_{rand} \in \{1, 2, \dots, n\}$ is a randomly chosen index, which ensures that u_i is different from its parent x_i in at least one component.

Selection: The offspring u_i is compared to its parent x_i using greedy criterion, and the one with better fitness will be selected into the next population.

2.3. Achievement scalarizing function

The achievement scalarizing function approach was first proposed by Wierzbicki²⁵, and now is frequently used to solve MOPs. The achievement scalarizing function is based on a reference point \bar{z} , no matter it is feasible or not, and the main idea is to project \bar{z} on to the Pareto optimal front along a specified direction. Given a reference point \bar{z} , one of the widely-used achievement scalarizing function is as follows:

$$\begin{cases} \min & \max_{i=1 \sim m} [\omega_i (f_i(x) - \bar{z}_i)] \\ \text{s.t.} & x \in \Omega, \end{cases} \quad (8)$$

where $\omega = (\omega_1, \omega_2, \dots, \omega_m)$ is a given weighting vector for scalarizing the objectives with $\omega \geq 0$ for all $i = 1, 2, \dots, m$ and $\sum_{i=1}^m \omega_i = 1$. The solution of (8) is a (weakly) Pareto optimal solution for any reference point²⁶. Fig. 1 illustrates the behavior

of achievement scalarizing function for two objectives to be minimized. In Fig. 1, \bar{z} is the reference point and $\omega = (\omega_1, \omega_2)$ is a given weighting vector, where $\omega_i = 1/(z_i^{nad} - z_i^*)$, then \bar{z} is projected to z' , which is the closest Pareto optimal solution (in the sense of the weighted-sum of the objectives) following the direction (ω_1, ω_2) . For further investigation, it is concluded that the reference point could be projected to different Pareto optimal solutions between A and B by altering weighting vectors²⁶. And recently, many researchers focus on the influences of different weights on the DM preferences^{17,27,28}.

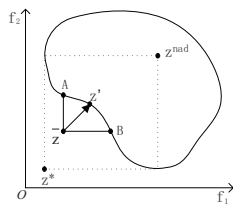


Fig. 1. Behavior of achievement scalarizing function

There are also other achievement scalarizing functions that are frequently used in literatures, for example,

$$\begin{cases} \min & \max_{i=1 \sim m} [\omega_i(f_i(x) - \bar{z}_i)] + \rho \sum_{i=1}^m \omega_i(f_i(x) - \bar{z}_i) \\ \text{s.t.} & x \in \Omega, \end{cases} \quad (9)$$

where ρ is an arbitrary small positive number ($0 < \rho \ll 1$). Problem (9) has minimal solutions at Pareto optimal solutions with bounded trade-offs between objectives (so-called properly Pareto optimal solutions), which often in practice are more useful than weakly Pareto optimal solutions²⁶. The above function is also called augmented ASF, and ρ is so called augmentation coefficient.

3. Hybrid Multiobjective DE Incorporating Preference based Local Search

In this section, we present a hybrid multiobjective DE algorithm where the ASF based local search is incorporated to improve the DE efficiency. In the hybrid algorithm, local search is performed to optimize the ASF where the reference point and weighting vector are adaptively determined, which prefers

to the solutions in the sparse region of nondominated set obtained by DE.

3.1. Hybrid mutation strategy

In DE literatures, formula (5) is a frequently used mutation scheme which has good exploration ability in the entire search space, while its exploitation ability is weak. To enhance the exploitation ability of DE mutation schemes, descent search similar to that of traditional optimization is used in offspring generating process. In every generation, solutions in population can be divided to dominated or nondominated ones by dominance relationship. For a particular solution x_j which is nondominated, we use formula (5) to generate mutation offspring. For a solution x_j which is dominated, there is at least one nondominated solution x_{non} dominating x_j , and along the direction from x_j to x_{non} , all objectives may decrease, so we can move x_j along this direction to some extent as follows:

$$v_i = x_j + F \cdot (x_{non} - x_j). \quad (10)$$

In the above hybrid mutation strategy for MOPs, offspring of the nondominated solutions are randomly generated by (5) which maintain the exploration ability, while offspring of the dominated solutions go close to a randomly selected nondominated solution which accelerates the convergence.

3.2. Local search based on achievement scalarizing function

In this section, local search based on ASF is presented to find the nondominated solutions in the sparse part of the obtained approximation of PF. After every generation of DE, the nondominated solutions of the current population and offspring can be selected. Then we can find the nondominated solutions in the sparse part of the nondominated front. Subsequently, the reference point is dynamically adjusted by these nondominated solutions such that its preference region is just the sparse part of the obtained approximation of PF. Finally, the reference point is used in the ASF (8) or (9). The local search behavior of the ASF is illustrated in Fig. 2(a). In Fig.

2(a), black dots denote the nondominated solutions obtained by DE, and the sparse part is between A and B. The reference point \bar{z} is determined by A and B with each coordinate being the smaller one of A and B (any other methods can be used to determine \bar{z} , provided that the preference region is the sparse part between A and B), then the preference region of \bar{z} on PF is between C and D, so the achievement scalarizing function using \bar{z} will induce a Pareto vector between C and D by a given weighting vector (any m -dimension vector with non-negative components whose summation is unit 1 can be used as the weighting vector, in this paper, ω is set to the normalized direction from \bar{z} to the other vertex of diagonal lines of the hyper box specified by A and B). In many complicated MOPs, the nondominated solutions obtained by DE are coarse in initial phases, and the reference point \bar{z} is far from the PF, as illustrated in Fig. 2(b). Thus the preference region of \bar{z} is almost the entire PF and the result induced by local search will greatly accelerate the convergence. In a word, the local search can not only accelerate the convergence in initial phases, but also improve the distribution of the obtained approximation of PF in later phases, so the performance of DE will be greatly enhanced by the proposed preference based local search strategy.

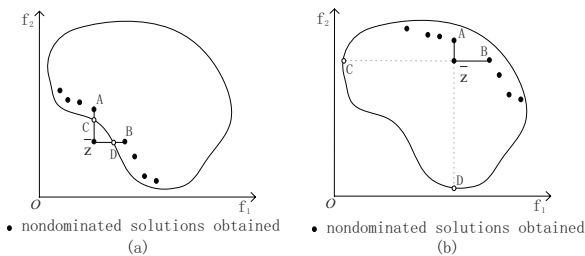


Fig. 2. Local search based on achievement scalarizing function

For the ASF with formulation (8), the following two conclusions can be conducted from Fig. 3. On one hand, even if A dominates B, it is still possible that their ASF value (8) are equal, and on the other hand, even if the ASF value of C is less than that of A, C and A are still nondominated each other. Moreover, traditional gradient based method to minimize the ASF can only get a/an (local) optimal solution D in the end, some solutions generated in the optimization process which are not optimal to ASF but

are nondominated may be deleted. Thus, the traditional local search method which only returns one optimal solution of ASF is not enough.

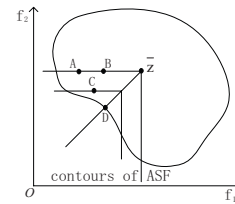


Fig. 3. Relationship between ASF and dominance

It can be seen from the above analysis that designing a mini population differential evolutionary algorithm to do the local search on ASF may be a reasonable choice. It can find multiple solutions in sparse region which are probably nondominated each other although they may not be optimal to ASF. Suppose that x_1 and x_2 are two solutions in sparse region, the detailed mini population differential evolutionary algorithm for local search is as follows.

Algorithm 1. Mini population DE for local search

Step 1. Initialization. For each solution x_i ($i=1,2$) in sparse region, generate N_0 (N_0 is a small integer) individuals around x_i which obey normal distribution with x_i as mean, and $|x_1 - x_2|$ as variance, and the $2N_0$ individuals consist of the initial population.

Step 2. Evaluation. For each individual generated in Step 1, calculating its ASF value based on (8) or (9), and the solution with optimal ASF value is denoted as x_{best} .

Step 3. Evolution. Evolve the mini population using DE with DE/current-to-best/1 mutation scheme (4) and crossover (7), in which mutation scheme (4) has good exploitation ability.

Step 4. Selection. Once an offspring is generated, immediate comparison is performed between offspring and its parent with their ASF values. If the offspring is better than its parent, then it replaces the parent. Furthermore, if it is better than x_{best} , x_{best} is then replaced by it. Otherwise, the offspring is discarded.

Step 5. If the given number of function evaluation is used, stop and the final population is as the output of local search, else, go to Step 3.

3.3. Improved ε -dominance updating strategy

In most MOEAs, an external population, named archive, is used to retain the nondominated solutions found so far, and some of them are updated by ε -dominance strategy^{9,12}: each member $f = (f_1, f_2, \dots, f_m)$ of the archive is assigned an identification array $B = (B_1, B_2, \dots, B_m)^T$ (m is the number of objectives), which can divide objective space into hyper boxes (B is also called the coordinate of the hyper box) as follows :

$$B_j(f) = \lfloor (f_j - f_j^{\min}) / \varepsilon_j \rfloor, \quad (11)$$

where f_j^{\min} is the minimum possible value of the j -th objective and ε_j is the same as ε used in definition 5. Each offspring used to update archive is first assigned an identification array computed by (11), and then all offspring and all members in archive have identification arrays. Compare their identification arrays using the usual dominance relationship, (i.e., compare the dominance relation of identification arrays of individuals), and keep the nondominated individuals in archive. If the identification arrays of two individuals are same, then the one with the smaller distance between its objective vector and its identification array will retain in the archive. This mechanism acting as an archive strategy has been shown the good convergence towards the Pareto optimal set and good diversity among the solutions found. However, as indicated in Ref. 10, the strategy has several drawbacks such as the loss of the extreme points, and the difficulties of determining an appropriate value of ε , which controls the number of nondominated solutions. Even though the approach in Ref. 10 provided the method to select appropriate ε , it greatly depends on the initial PF approximation. Clearly, it is difficult to obtain any information of PF before starting to solve the MOP. Thus the method in Ref. 10 to choose the appropriate ε is hard to implemented in practice. To avoid the loss of extreme points, an improved ε -dominance strategy based on strongly dominance relationship is presented in this subsection, which does not need any additional extreme points storage. The presented strategy is described in details as follows.

If the identification array of offspring is nondominated with respect to any archive member or the off-

spring locates in same hyper box with any archive member, the acceptance rule remains unchanged with original ε -dominance. Otherwise, suppose that B_a is the identification array of an archive member a and B_c is the identification array of an offspring c . We consider three cases:

1) If $B_a \prec B_c$, i.e., the identification array B_a of an archive member a strongly dominates B_c of offspring c , then c is not only ε -dominated but also dominated by this archive member, and offspring c is not accepted.

2) If $B_c \prec B_a$, then the archive member a is deleted and the offspring c is accepted.

3) If $B_c \preceq B_a$ and $B_c \not\prec B_a$ or viceversa, then a and c are checked as the usual domination. If the offspring c dominates the archive member a , c is accepted and a is deleted. If c is dominated by a , then c is not accepted. If neither of the above two cases occur, i.e., a and c are nondominated, then c is accepted and a is also retained in the archive (see Fig. 4). If the original ε -dominance is used, offspring c is retained and archive member a is deleted in Fig. 4(a), while in Fig. 4(b), offspring c is deleted and archive member a is retained.

In the above update strategy, we can retain the offspring that locate in different hyper boxes and are nondominated with archive members. Obviously, the improved strategy not only retains the extreme points but also improve the uniformity of Pareto front. Moreover the archive size needs not to be pre-defined, and the archive is bounded.

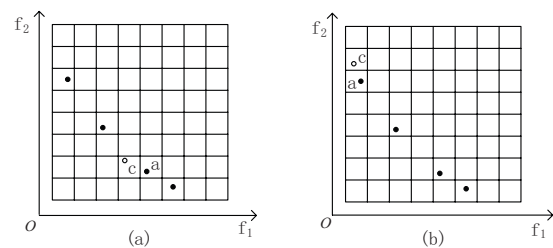


Fig. 4. Box weak dominance relationship

Suppose that c, d and e are members of archive, and a, b, f and g are offspring generated. From Fig. 5, one can see that only c, d and e are retained in the archive if the original ε -dominance strategy is used. Even with the additional extreme points storage strategy¹³, all solutions exclusive of b and f can

be retained, while the improved ϵ -dominance strategy will keep all of the seven individuals in the end. Thus the improved strategy not only avoids the loss of extreme points but also enhances the uniformity of PF obtained. Fig. 5 clearly shows the advantages of the proposed method over the original ϵ -dominance strategy.

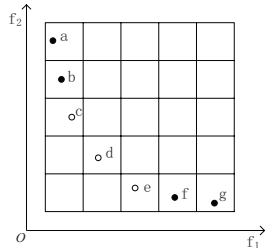


Fig. 5. Archive members updated by improved ϵ -dominance

It should be mentioned that even though the above improved strategy is similar to that in Ref. 29, the box and the number of individuals including in the box are different. The hyper boxes in Ref. 29 are changed with the evolution and every box can hold more than one individual, while the boxes are unchanged once the ϵ is given and each box can contain only one individual in the proposed algorithm.

3.4. Hybrid multiobjective DE incorporating preference based local search

For multiobjective problem (1), the proposed hybrid multiobjective DE algorithm incorporating preference based local search and improved ϵ -dominance, denoted by PLSDE, is described as follows:

Algorithm 2. PLSDE

Step 1. Initialization. Generate the initial population P using orthogonal design (see Ref. 30 in detail), i.e. generate the initial individuals uniformly in the search space Ω , select the nondominated solutions of P based on the improved ϵ -dominance and save them in external archive AR .

Step 2. Evolution. For each solution $x_i \in P$, ($i = 1, 2, \dots, NP$), generate offspring based on procedure in subsection 3.1, and all offspring constitute OP .

Step 3. Update. Use nondominated solutions of OP to update the external archive set AR via the improved ϵ -dominance strategy in subsection 3.3.

Step 4. Local Search. According to the distribution of the solutions in AR , local search is performed via Algorithm 1 in subsection 3.2, and the output of local search is denoted by LP .

Step 5. Selection. Select the next population from $P \cup OP \cup LP$ using nondominated ranking method, i.e., select the first rank individuals into the population, then the second rank individuals, and the process is repeated until some rank where the individuals are more than the vacancy in the population, then the required number of individuals in this rank are randomly selected into population without calculating the crowding distance.

Step 6. Update the external archive AR with solutions in LP by improved ϵ -dominance strategy in subsection 3.3.

Step 7. If stopping criterion is not met, go to Step 2, else, output the external archive AR .

4. Simulation Results

4.1. Test instances

To test the performance of PLSDE, we make the simulations on five 2-objective test problems: ZDT1, ZDT2, ZDT3, ZDT4, ZDT6 and five 3-objective problems: DTLZ1, DTLZ2, DTLZ3, DTLZ4 and DTLZ7. These problems have different characteristics in their PF, which are frequently used to verify the efficiency and effectiveness of different algorithms. For more details about the test instances, please refer to Refs. 9 and 31.

4.2. Performance metric

The inverted generational distance (IGD)³² was widely used to evaluate the performance of an algorithm. The IGD is defined as

$$IGD(P^*, P) = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|},$$

where P^* is a set of uniformly distributed points on PF, P is an approximation to PF, $d(v, P)$ is the minimum Euclidean distance between v and the points in P , and $|P^*|$ denotes the number of points in P^* .

Because P^* is a set of uniformly distributed points on PF, if we want to get a small value of $IGD(P^*, P)$, P must be of diversity, uniform distribution and good convergence. Thus, $IGD(P^*, P)$ could be as a performance measure to evaluate the diversity, distribution and convergence of P . In this paper, we use IGD as the performance measure.

4.3. Experimental setting

Experimental parameters are given below:

- 1) The number of decision variables (n): n is 30 for ZDT1, ZDT2 and ZDT3, and 10 for the other problems.
- 2) The size of population (NP): NP is 100 for ZDT test instances and 200 for DTLZ test instances.
- 3) The maximum number of function evaluations is set to 10000 for ZDT test instances except ZDT4

and 50000 for DTLZ problems, the maximum number is 20000 for ZDT4 for it has many local PFs.

4) The number of the solutions generated around each solution in a sparse region and function evaluations in local search are 5 and 50 for ZDT series problems, respectively, and 10 and 100 for DTLZ series problems, respectively.

5) The control parameter F and crossover probability C_r in DE are set to 0.5 and 0.3 for all test instances.

6) The value of ϵ in archive updating is 0.02 for ZDT1 to ZDT4, and 0.016 for ZDT6 to maintain about 100 Pareto solutions. For DTLZ test problems, the value is 0.035 for DTLZ1, and 0.09 for other DTLZ problems to retain about 300 Pareto solutions in the final archive.

7) The number of P^* used in IGD metric is 500 for ZDT series test instances, 990 for DTLZ1-4 instances, and 961 for DTLZ7 problem.

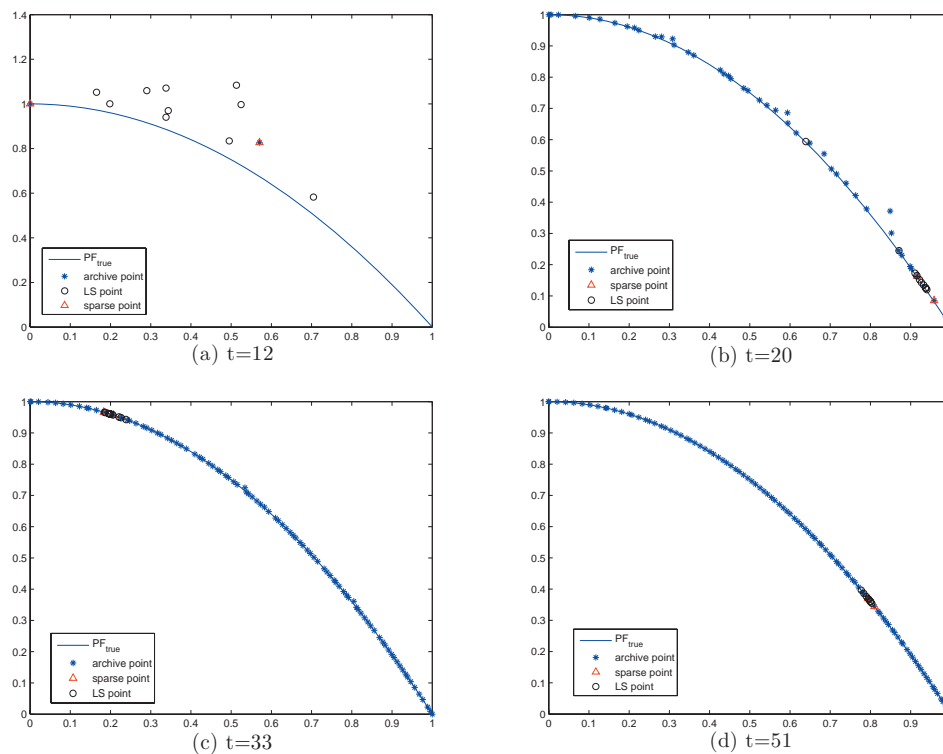


Fig. 6. Local search results by mini population DE at generation t

4.4. Effectiveness of the local search in evolution

To verify the efficiency of ASF based local search in the proposed algorithm, we exhibit several local search results in different stages of evolution for ZDT2 test instance in Fig. 6. Fig. 6(a) demonstrates the first local search results at generation $t = 12$ when there are more than one solution in the archive, which are not all Pareto optimal solutions. From Fig. 6(a), one can see that the local search not only accelerates the convergence but also increases the diversity of the archive members. With the evolution process going on, from Fig. 6(b) at generation $t = 20$, one can see that there are many nondominated solutions in the archive which are scattered unevenly. Then the local search is performed and the resulted solutions are exactly on the sparse part of PF denoted by black circle. The other two Fig. 6(c) and 6(d) at $t = 33$ and $t = 51$ are also illustrated that the preference based local search always concentrates intensively on accordingly sparse region in different evolutionary stages. These processes illustrate that the local search makes the search prefer to the sparse region all the time, and the algorithm designed for local search is efficient in obtaining multiple Pareto optimal solutions.

4.5. Experimental results

We execute the proposed algorithm PLSDE and standard DE (DE without local search) with original ϵ -dominance (SDE) for 10 independent runs, and present the best, mean, worst and standard deviation (std.) of the IGD metric values of the solutions obtained by these two algorithms in Table 1. From Table 1, one can see the best, mean, worst and std. of IGD values in 10 independent runs, obtained by PLSDE, are much better than those obtained by SDE. Therefore, we can conclude that the proposed strategies of preference based local search and improved ϵ -dominance greatly enhance the efficiency of standard DE with original ϵ -dominance.

Table 2 presents the best and standard deviation of IGD metric values of the solutions obtained

by NSGA-II¹, MOEA/D³² and PLSDE for five 2-objectives ZDT instances and two 3-objectives instances DTLZ1 and DTLZ2 (the results of NSGA-II and MOEA/D are directly taken from Refs. 1 and 32, respectively, and the results for the remaining three test problems, DTLZ3, DTLZ4 and DTLZ7, are not available for NSGA-II and MOEA/D in these references. Thus we can not make a direct comparison with NSGA-II and MOEA/D for these three test problems). From Table 2, one can see that the final solutions obtained by PLSDE are better than those of NSGA-II and MOEA/D in terms of IGD metric for all ZDT test instances and two DTLZ test instances DTLZ1 and DTLZ2. Furthermore, the number of function evaluations of PLSDE is fewer than that of NSGA-II and MOEA/D.

Table 1. IGD metric values of the solutions obtained by PLSDE and standard DE with original ϵ -dominance

Function	Algorithm	best	mean	worst	std
ZDT1	PLSDE	0.0039	0.0040	0.0046	0.0002
	SDE	0.0401	0.0407	0.0418	0.0005
ZDT2	PLSDE	0.0041	0.0042	0.0044	0.0001
	SDE	0.0407	0.0409	0.0411	0.0001
ZDT3	PLSDE	0.0048	0.0051	0.0054	0.0002
	SDE	0.0414	0.0422	0.0428	0.0005
ZDT4	PLSDE	0.0037	0.0039	0.0040	0.0001
	SDE	0.0403	0.0426	0.0446	0.0016
ZDT6	PLSDE	0.0031	0.0034	0.0037	0.0002
	SDE	0.0372	0.0377	0.0386	0.0005
DTLZ1	PLSDE	0.0067	0.0101	0.0107	0.0012
	SDE	0.0205	0.0206	0.0207	0.0001
DTLZ2	PLSDE	0.0234	0.0237	0.0239	0.0001
	SDE	0.0317	0.0322	0.0326	0.0003
DTLZ3	PLSDE	0.0227	0.0228	0.0230	0.0001
	SDE	0.0319	0.0320	0.0322	0.0001
DTLZ4	PLSDE	0.0232	0.0233	0.0234	0.0001
	SDE	0.0323	0.0324	0.0325	0.0001
DTLZ7	PLSDE	0.0306	0.0315	0.0330	0.0008
	SDE	0.0331	0.0333	0.0334	0.0001

Table 2. IGD metric values of the solutions obtained by NSGA-II, MOEA/D and PLSDE

Function	NSGA-II	MOEA/D	PLSDE
ZDT1	0.0050(0.0002)	0.0055(0.0039)	0.0039(0.0002)
ZDT2	0.0049(0.0002)	0.0079(0.0109)	0.0041(0.0001)
ZDT3	0.0065(0.0054)	0.0143(0.0091)	0.0049(0.0002)
ZDT4	0.0182(0.0237)	0.0076(0.0023)	0.0037(0.0001)
ZDT6	0.0169(0.0028)	0.0042(0.0003)	0.0031(0.0002)
DTLZ1	0.0648(0.1015)	0.0317(0.0005)	0.0067(0.0012)
DTLZ2	0.0417(0.0013)	0.0389(0.0001)	0.0234(0.0001)

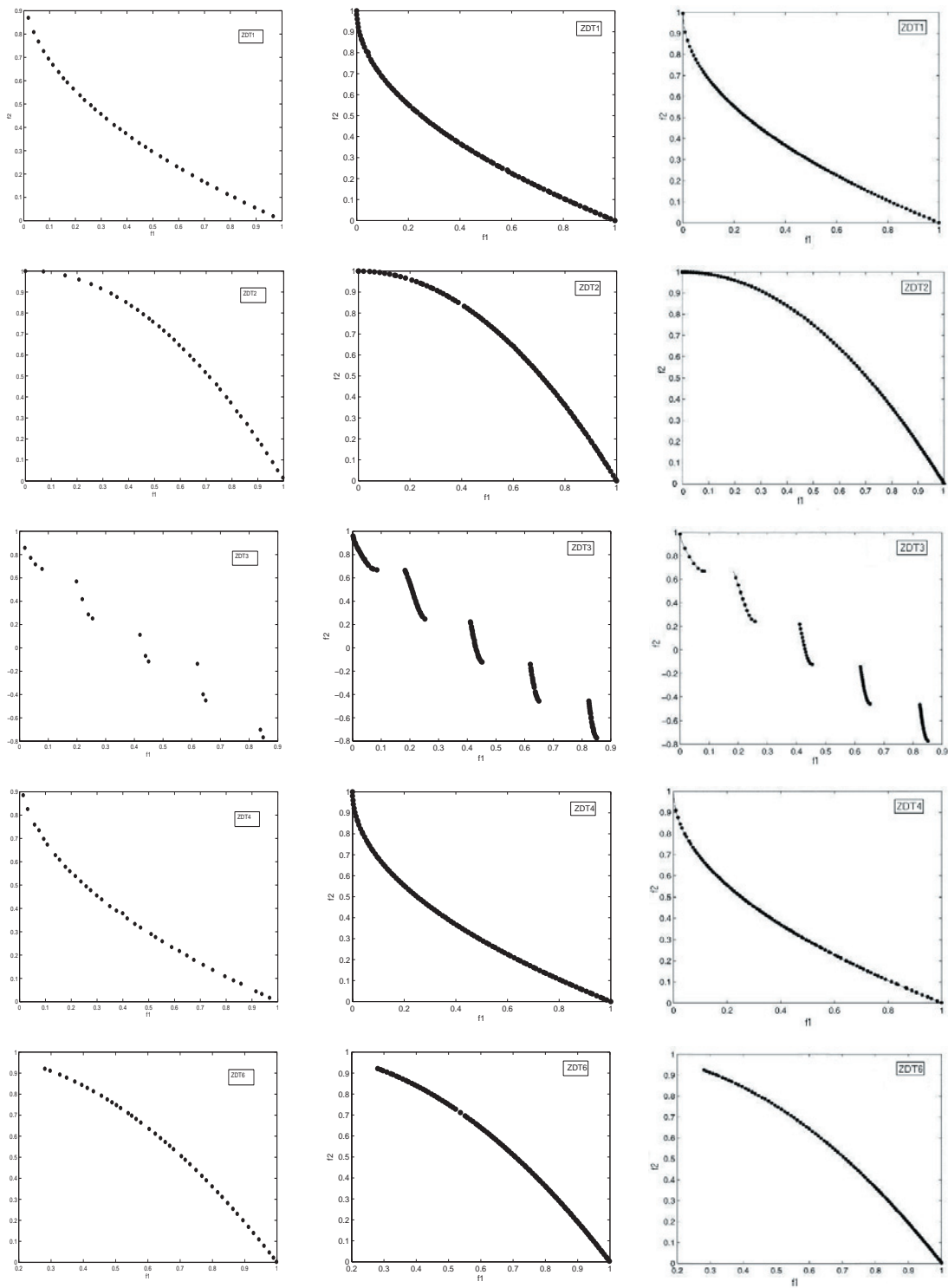


Fig. 7. PFs obtained by SDE, PLSDE and MOEA/D for five 2-objective test instances. The left column is for SDE, the middle one for PLSDE and the right one for MOEA/D

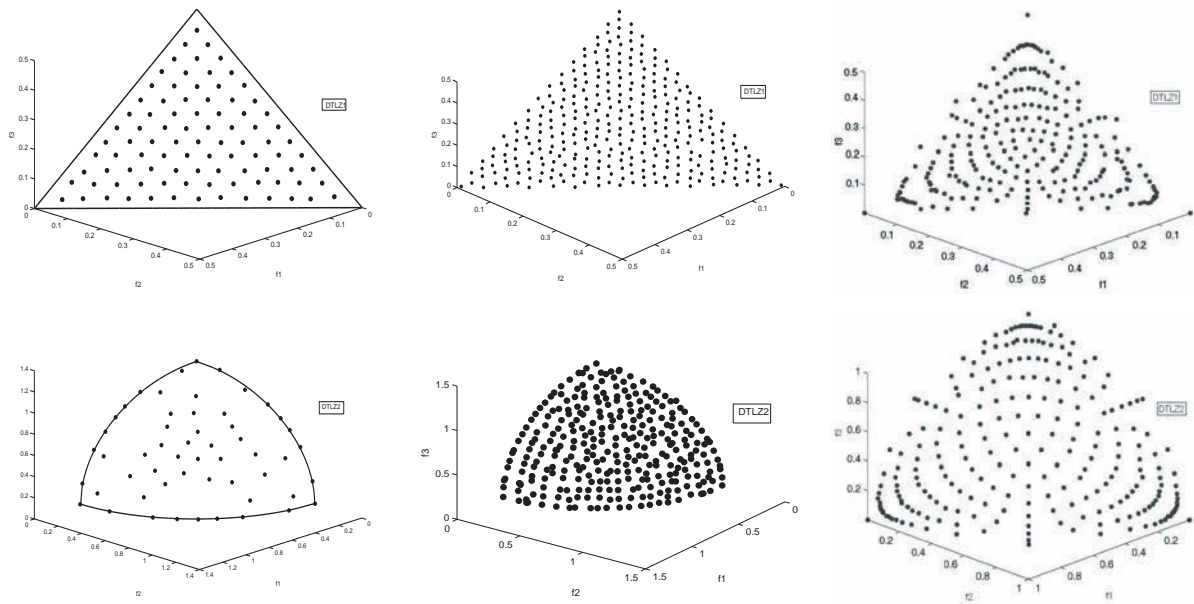


Fig. 8. PFs obtained by SDE, PLSDE and MOEA/D for 3-objective instances DTLZ1 and DTLZ2. The left column is for SDE, the middle one for PLSDE and the right one for MOEA/D.

Fig. 7 illustrates the PFs of five ZDT test instances obtained by SDE, PLSDE and MOEA/D. The results in Figure 7 from the left to the right are those of SDE, PLSDE and MOEA/D, respectively. It can be seen from this figure that the PFs got by SDE in the left column are not only worse in uniformity but also have less solutions than those of PLSDE in middle column. This is because SDE did not use the preference based local search to pay more attention to the sparse regions. Furthermore, SDE loses the extreme points of PFs, as well as points located in horizontal or vertical part of the segments of PFs due to the shortcoming of original ϵ -dominance. Compared with PFs of MOEA/D in right column, PLSDE outperforms MOEA/D on ZDT1, ZDT3 and ZDT4, especially in the extreme points and the segments that are almost horizontal or vertical, which also illustrates the preference based local search strategy and the improved ϵ -dominance update procedure used in PLSDE are helpful to find a widely spread and evenly distributed approximation of Pareto front. For the remaining problems,

PLSDE and MOEA/D perform almost the same.

Fig. 8 shows the PFs obtained by SDE, PLSDE and MOEA/D for two 3-objectives instances, DTLZ1 and DTLZ2. From the sub-figures in the left column, one can see that the PFs obtained by SDE have a small number of optimal solutions and there are a few or no solutions on the boundaries due to the shortcoming of original ϵ -dominance of losing the boundary points (for clarification, the boundaries of PF are plotted on the two figures). From the figures in the middle column, it is concluded that PLSDE found a large number of optimal solutions, which are almost uniformly distributed on the Pareto front. Moreover, PLSDE found the uniformly distributed solutions on PFs inclusive on the boundaries. The figures in the right column is for MOEA/D, which exhibits that MOEA/D also got a large number of optimal solutions on PFs, while there are only a few on the boundaries. SDE can not find the solutions on boundary for DTLZ1 and performs worst.

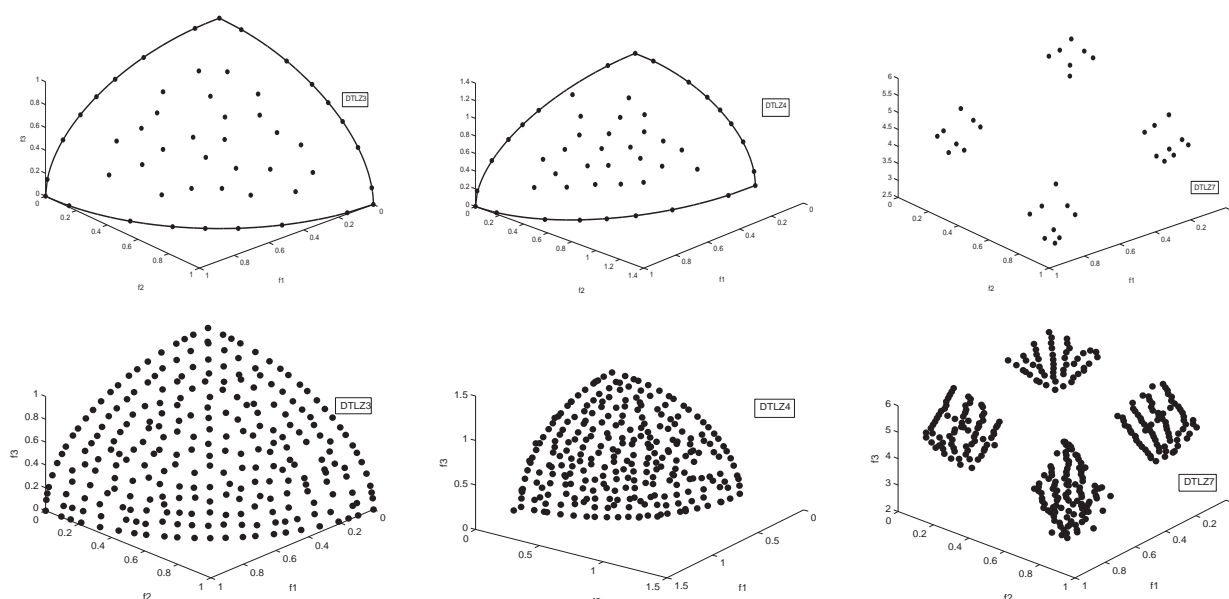


Fig. 9. PFs obtained by SDE and PLSDE for DTLZ3, DTLZ4 and DTLZ7. The top row is for SDE and the bottom one for PLSDE

Fig. 9 shows the PFs obtained by SDE in the top row and PLSDE in the bottom on the other three DTLZ test instances: DTLZ3, DTLZ4 and DTLZ7. From Fig. 9, one can see that the performance of PLSDE is much better than SDE both in the number of solutions obtained and the uniformity of solutions distribution. Moreover, for instance with four discontinuous PF, DTLZ7, PLSDE also found well distributed Pareto solutions on different segments.

From the results in the above tables and figures, it can be concluded that the proposed algorithm PLSDE can find well distributed and high quality solutions for problems with different characteristics, and is more efficient than the compared algorithms.

5. Conclusion

A hybrid multiobjective differential evolution algorithm incorporating preference based local search is proposed. To obtain multiple solutions in sparse region of the obtained nondominated front, a local search which prefers to solutions in sparse region is designed based on ASF in which reference point and weighting vector are dynamically adjusted with evolution. To retain uniformly spread nondom-

inated solutions, an improved ϵ -dominance strategy, which would not delete the extreme points found in evolution, is proposed. The improved strategy can keep the extreme points without using additional extreme points storage technique. The simulation results on ten standard test instances and comparisons with other two famous MOEAs show the high performance and effectiveness of the proposed algorithm.

Acknowledgment

The authors are very grateful to the Associate Editor and anonymous reviewers for their valuable comments and constructive suggestions on the previous version of this paper.

References

1. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evolutionary Computation*, **6**(2), 182-197 (2002).
2. E. Zitzler, L. Thiele and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization," *Evolutionary Methods for Design Optimisation and Control*, 95-100 (2002).

3. J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the Pareto archive evolution strategy," *Evolutionary Computation*, **8(2)**, 149-172 (2000).
4. R. Storn and K. Price, "Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces," *Technical Report TR-95-12, International Computer Science, Berkeley, California*, (1995).
5. H. A. Abbass, R. Sarker and C. Newton, "PDE: A Pareto-frontier differential evolution approach for multiobjective optimization problems," *In Proceedings of the Congress on Evolutionary Computation 2001(CEC'2001)*, 831-836 (2001).
6. N. K. Madavan, "Multiobjective optimization using a Pareto Differential evolution approach," *In congress of Evolutionary Computation*, **2**, 1145-1150 (2002).
7. T. Tobič, and B. Filipič, "DEMO: Differential evolution for multiobjective optimization," *C.A.C. Coello et al. (Eds.): EMO 2005, LNCS 3410*, 520-533 (2005).
8. L. V. Santana-Quintero and C. A. C. Coello, "An Algorithm Based on Differential Evolution for Multi-Objective Problems," *International Journal of Computational Intelligence Research*, **1(2)**, 151-169 (2005).
9. K. Deb, M. Mohan and S. Mishra, "Evaluating the ϵ -domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions," *Evolutionary Computation*, **13(4)**, 501-525 (2005).
10. A. G. Hernández-Díaz, L. V. Santana-Quintero, C. A. C. Coello and J. Molina, "Pareto-adaptive ϵ -dominance," *Evolutionary Computation*, **15(4)**, 493-517 (2007).
11. N. Dong and Y. P. Wang, "A Hybrid Multiobjective Differential Evolution Algorithm Based on Improved ϵ -Dominance," *International Conference on Computational Intelligence and Security (CIS)*, 24-28 (2011).
12. M. Laumanns, L. Thiele, K. Deb and E. Zitzler, "Combining Convergence and Diversity in Evolutionary Multi-objective Optimization," *Evolutionary Computation*, **10(3)**, 263-282 (2002).
13. W. Y. Gong and Z. H. Cai, "An improved multiobjective differential evolution based on Pareto-adaptive ϵ -dominance and orthogonal design," *European Journal of Operational Research*, **198(2)**, 576-601 (2009).
14. K. Deb, J. Sundar, N. Udaya Bhaskara Roa, S. Chaudhuri, "Reference point based multi-objective optimization using evolutionary algorithms," *International Journal of Computational Intelligence Research*, **2(3)**, 273-286 (2006).
15. Molina J., Santana L.V., Hernández-Díaz A.G., Coello Coello C.A., Caballero, R., "g-dominance: Reference point based dominance for multiobjective metaheuristics," *European Journal of Operational Research* **197(2)**, 685-692 (2009).
16. L. Thiele, K. Miettinen, P.J. Korhonen, J. Molina, "A preference-based evolutionary algorithm for multi-objective optimization," *Evolutionary Computation*, **17(3)**, 411-436 (2009).
17. M. Luque, K. Miettinen, A. B. Ruiz, F. Ruiz, "A two-slope achievement scalarizing function for interactive multiobjective optimization," *Computers and Operations Research*, **39**, 1673-1681 (2012).
18. E. Zitzler and S. Knzli, "Indicator-based selection in multiobjective search," *In: Proceedings of the Eighth International Conference on Parallel Problem Solving from Nature (PPSN 2004)*, Springer, Berlin, 832-842 (2004).
19. J. F. Figueira, A. Liefooghe, E.-G. Talbi, A. P. Wierzbicki, "A parallel multiple reference point approach for multi-objective optimization," *European Journal of Operational Research*, **205(2)**, 390-400 (2010).
20. H. Ishibuchi and T. Murata, "A multi-objective genetic local search algorithm and its application to flowshop scheduling," *IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews*, **28(3)**, 392-403 (1998).
21. A. Jaskiewicz, "Genetic local search for multiple objective combinatorial optimization," *European Journal of Operational research*, **137(1)**, 50-71 (2002).
22. T. Goel and K. Deb, "Hybrid methods for multi-objective evolutionary algorithms," *In Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning (SEAL02)*, L. Wang, K. C. Tan, T. Furuhashi, J. H. Kim, and X. Yao, Eds., **1**, 188-192 (2002).
23. K. Sindhya, A. Sinha, K. Deb and K. Miettinen, "Local Search Based Evolutionary Multi-Objective Optimization Algorithm for Constrained and Unconstrained Problem," *In Congress on Evolutionary Computation (CEC 2009)*, IEEE Press, 2919-2926 (2009).
24. R. Storn and K. Price, "Home page of differential evolution," 2003. <<http://www.ICSI.Berkeley.edu/storn/code.html>>
25. Wierzbicki A. P., "The use of reference objectives in multiobjective optimization," *In: FandelG, GalT, editors. Multiple criteria decision making, theory and applications. Berlin: Springer-Verlag*, 468-486 (1980).
26. Miettinen K., "Nonlinear multiobjective optimization," *Boston: Kluwer Academic Publishers*, 1999.
27. Ruiz F., Luque M. and Cabello J. M., "A classification of the weighting schemes in reference point procedures for multiobjective programming," *Journal of the Operational Research Society*, **60(4)**, 544-553 (2009).
28. Luque M., Miettinen K., Eskelinen P., Ruiz F., "Incorporating preference information in interactive reference point methods for multiobjective optimization," *Omega*, **37(2)**, 450-462 (2009).

29. J. D. Knowles and D. W. Corne, "Properties of an Adaptive Archiving Algorithm for Storing Nondominated Vectors," *IEEE Trans. Evolutionary Computation*, **7(2)**, 100-116 (2003).
30. Y. W. Leung and Y. Wang, "An orthogonal genetic algorithm with quantization for global numerical optimization," *IEEE Trans. Evolutionary Computation*, **5(1)**, 41-53 (2003).
31. E. Zitzler, K. Deb and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical Results," *Evolutionary Computation*, **8(2)**, 173-195 (2000).
32. Qingfu Zhang and Hui Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," *IEEE Trans. Evolutionary Computation*, **11(6)**, 712-731 (2007).