Applied Soft Computing 50 (2017) 344–355

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Many-objective evolutionary optimization based on reference points

Yiping Liu [a], Dunwei Gong [a,b,*], Xiaoyan Sun [a], Yong Zhang [a]

[a] School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou 221116, China
[b] School of Electrical Engineering and Information Engineering, LanZhou University of Technology, Lanzhou 730050, China

ARTICLE INFO

Article history:
Received 15 December 2015
Received in revised form 1 October 2016
Accepted 7 November 2016
Available online 25 November 2016

Keywords:
Evolutionary optimization
Multi-objective optimization
Many-objective optimization
Reference point
Distance

ABSTRACT

Many-objective optimization problems are common in real-world applications, few evolutionary optimization methods, however, are suitable for solving them up to date due to their difficulties. A reference points-based evolutionary algorithm (RPEA) was proposed in this paper to solve many-objective optimization problems. The aim of this study is to exploit the potential of the reference points-based approach to strengthen the selection pressure towards the Pareto front while maintaining an extensive and uniform distribution among solutions. In RPEA, a series of reference points with good performances in convergence and distribution are continuously generated according to the current population to guide the evolution. Furthermore, superior individuals are selected based on the evaluation of each individual by calculating the distances between the reference points and the individual in the objective space. The proposed algorithm was applied to seven benchmark optimization problems and compared with ε-MOEA, HypE, MOEA/D and NSGA-III. The results empirically show that the proposed algorithm has a good adaptability to problems with irregular or degenerate Pareto fronts, whereas the other reference points-based algorithms do not. Moreover, it outperforms the other four in 8 out of 21 test instances, demonstrating that it has an advantage in obtaining a Pareto optimal set with good performances.

1. Introduction

Various optimization problems with multiple objectives commonly exist in real-world applications, e.g., water management [1], software development [2], and industrial scheduling [3]. Their common characteristics are that they contain more than one objective and there exist some conflicts among these objectives, indicating that there is no solution which is optimal for all objectives. They are termed as multi-objective optimization problems (MOP). Problems with more than three objectives are defined as many-objective optimization problems (MaOP). Without loss of generality, the MOP considered in this study is formulated as follows:

$$\begin{aligned} &\min \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_M(\boldsymbol{x})) \\ &\text{s.t.}\ \boldsymbol{x} \in S \subset \boldsymbol{R}^n \end{aligned} \tag{1}$$

where $\boldsymbol{x}$ represents an n-dimensional decision variable in space S. $f_m(\boldsymbol{x})$, $m = 1, 2, \ldots, M$ is the mth objective to be minimized, and M is the number of objectives. When $M > 3$, this problem is a MaOP.

Over the past two decades, a large number of multi-objective evolutionary algorithms (MOEAs), e.g., nondominated sorting genetic algorithm II (NSGA-II) [4], strength Pareto evolutionary algorithm 2 (SPEA2) [5] and multiobjective particle swarm optimization (MOPSO) [6], have been proposed. MOEAs usually work very well on two- or three-objective problems. However, they noticeably deteriorate their search ability when solving MaOPs, due to the following serious difficulties [7]:

(1) The Pareto dominance-based selection pressure toward the Pareto front sharply decreases as the number of objectives increases. Almost all solutions in a population are non-dominated when the number of objectives is large. This makes the Pareto-based selection fail to distinguish individuals.
(2) Exponential increase in the number of solutions is required for approximating the entire Pareto front. Since the Pareto front is a hyper-surface in the high-dimensional objective space, thousands of non-dominated solutions may be required to approximate the entire Pareto front of a MaOP.
(3) The visualization of obtained non-dominated solutions is difficult due to the large number of objectives. Consequently, it could be difficult for a decision maker to choose a final solution in many-objective optimization.

Given the above difficulties, seeking for new methods to effectively solve MaOPs is of considerable necessity. In this paper, we focus

* Corresponding author.
E-mail addresses: yiping0liu@gmail.com (Y. Liu), dwgong@vip.163.com (D. Gong), xysun78@126.com (X. Sun), yongzh401@126.com (Y. Zhang).

http://dx.doi.org/10.1016/j.asoc.2016.11.009
1568-4946/

on solving the first difficulty by establishing a novel evolutionary algorithm based on reference points.

Reference points have been employed to guide the evolution in many situations. On account of the evaluation of individuals by the distances between reference points and them, the selection pressure of superior individuals will not lose in many-objective optimization. Most preference-based MOEAs usually adopt only one reference point to search solutions in the objective sub-spaces of interest to a decision maker [8–10]. Intuitively, adopting a series of reference points to obtain the whole Pareto front has the potential in solving MaOPs. For example, both [11,12] generate multiple reference points uniformly distributed on a hyperplane to guide solutions to converge. However, the positions of these reference points are predefined in these methods, whereas the true Pareto front of a practical optimization problem is usually unknown beforehand. The mismatches between the reference points and the true Pareto front may degrade the search ability of the algorithms. Therefore, if appropriate reference points can be continuously generated during the evolution according to information provided by the current population, it will be of great possibility to achieve a solution set with good performances.

In view of this, a reference points-based evolutionary algorithm (RPEA) was proposed in this study. During the evolution, a series of reference points with good performances in convergence and distribution are adaptively generated according to the current population. Then, by minimizing the distances between the reference points and the individuals, superior individuals are selected in the environment selection process. Some preliminary results of this study were presented in [13], where a number of reference points are generated in terms of the parent individuals, and the distance between an individual and each of the reference point is evaluated by a modified Euclidean distance. In this study, we extended the methodology and proposed the following new features:

(1) The proposed algorithm is generalized to a common framework.
(2) Only the non-dominated individuals in the parent and the offspring populations are chosen to generate the reference points, thus further enhancing the performances of these reference points.
(3) An achievement function based on the Tchebychev distance is adopted instead of the modified Euclidean distance, to efficiently evaluate the candidates.
(4) The behavior of RPEA with different parameter settings is empirically investigated.
(5) RPEA is compared with three other state-of-the-art MOEAs, i.e., ε-MOEA, MOEA/D and NSGA-III, on more benchmark optimization problems.

This study has the following contributions: (1) presenting an approach to generating reference points which are adaptive to optimization problems; (2) proposing a method of selecting superior individuals based on the distances between reference points and individuals.

The remainder of this paper is organized as follows. Section 2 reviews the related work. The proposed approach, RPEA, is described in Section 3. Section 4 presents the benchmark MaOPs, performance metrics, and the algorithm settings used in performance comparison. The experiment results and analysis are given in Section 5. Finally, Section 6 concludes this paper and offers suggestions on possible opportunities for future research.

2. Related work

2.1. Multi-objective evolutionary optimization

Formula (1) describes different types of optimization problems in case of different values of M. When the value of M is larger than one, formula (1) represents a MOP. In such a scenario, the optimization objectives usually conflict with each other, i.e., improvements on one objective will cause deterioration on the other objectives, which indicates that it is impossible to find one optimal solution that meets all the objectives. Then, a set of solutions that have a good tradeoff among all objectives may be more desirable.

In multi-objective optimization, the following concepts have been well defined and widely applied.

Pareto dominance: for any two different solutions of formula (1), $\boldsymbol{x}_1, \boldsymbol{x}_2 \in S$, if $\forall m = 1, 2, \ldots, M, f_m(\boldsymbol{x}_1) \leq f_m(\boldsymbol{x}_2)$, and $\exists i = 1, 2, \ldots, M, f_i(\boldsymbol{x}_1) < f_i(\boldsymbol{x}_2)$, then $\boldsymbol{x}_1$ dominates $\boldsymbol{x}_2$, denoted as $\boldsymbol{x}_1 \succ \boldsymbol{x}_2$.

Pareto optimal set: for a solution of formula (1), $\boldsymbol{x}^* \in S$, if there is no $\boldsymbol{x}' \in S$ satisfying $\boldsymbol{x}' \succ \boldsymbol{x}^*$, then $\boldsymbol{x}^*$ is termed as the Pareto optimal solution. All such solutions form into a set which is often called Pareto optimal solution set.

Pareto front: the image of Pareto optimal solution set on the objective space is known as the Pareto front.

MOEAs are powerful for solving MOPs. Among existing popular MOEAs, NSGA-II [4] is more highlighted. In NSGA-II, different levels of non-domination, obtained by employing a fast non-dominated sorting approach based on the Pareto dominance comparison to the union set of parent and offspring populations, is used to evaluate an individual. In addition, the crowding distance is considered if more than one solution belongs to the same front. Thus, the individuals with good convergence and distribution are reserved to the next generation. Besides NSGA-II, there are other outstanding MOEAs, e.g., SPEA2 [5]. In SPEA2, the fitness of an individual is calculated based on the Pareto dominance. Then the density of the individual is obtained by the kth nearest neighbor method so as to maintain a fixed population size.

2.2. Many-objective evolutionary optimization

The Pareto-based MOEAs mentioned above are efficient for problems with no more than three objectives. When the number of objectives is larger than three, these methods may also be applied to solve them. However, their performance will be greatly impaired along with the increase of the objectives, or even cannot find a Pareto optimal set [14]. Therefore, it is of considerable necessity to explore novel methods for solving MaOPs.

Under the framework of Pareto-based MOEAs, developing new dominance relations to compare individuals so as to maintain enough selection pressure is the best for MaOPs. Up to date, several new dominance concepts, e.g., ε-dominance [15], the electre-tri method [16], grid-based dominance [17], have been well studied and applied. These new dominance relations can finely compare different individuals. However, they may also cause the population to converge to a sub-region of the Pareto optimal front [18].

Objective reduction is an effective alternative method for solving MaOPs. In such a case, the dimensionality of objectives is first reduced, and the traditional Pareto dominance is then adopted to evaluate an individual. Popular methods include: optimization on partial objectives [19], redundant objectives deletion [20], and objectives weighting [21]. Recently, de Freitas et al. [22] proposed the aggregation tree technique, which can be used for not only objective reduction but also visualization. By using the above methods, a reduced set of objectives is the subset of the original set, which indicates that if the number of objectives is very large, the reduced set may also contain more than three objectives.

Using an indicator function to measure the quality of a solution set is a theoretically well-supported alternative to the traditional Pareto dominance. This kind of MOEA is referred to as an indicator-based evolutionary algorithm (IBEA) [23]. Since hypervolume is strictly monotonic with regard to the Pareto dominance, it has been a widely used indicator in IBEA. However, its computation complexity increases exponentially along with the number of objectives, which inhibits the full exploitation of its potential. HypE [24] uses a Monte Carlo simulation to approximate the exact hypervolume values, by which the accuracy of the estimates can be trade off against the computing resources available. This makes HypE very competitive for solving MaOPs.

The aggregation approach is another perspective alternative to the traditional Pareto dominance. In this approach, the objectives of an MOP (MaOP) are aggregated into a scalar. The diversity of a population is maintained by specifying a set of well-distributed weight vectors to guide the individuals to simultaneously search towards different directions. The most representative MOEAs based on this concept are MSOPS [25] and MOEA/D [26] which are considered as important approaches for solving MaOPs. Recently, Dai and Wang [27] combined MOEA/D with CDAS [28] to improve their performances on MaOPs. [29] proposed a new way to balancing convergence and diversity in MOEA/D.

The set-based MOEA developed for MaOPs have been focused on, and several outstanding studies have been attained. Bader et al. [30] proposed a set-based MOEA, in which the evolutionary population contains some sets of solutions with the same size, and then the hypervolume indicator is adopted to assess the performance of those sets. Under such a framework, the original MaOP is transformed into a single-objective optimization problem. The goal of the MOEA is to obtain a Pareto optimal set satisfying some indicators, e.g., hypervolume, diversity or spread. From this point, Zitzler et al. [31] advocated that the optimization process of an MOEA is essentially to find an optimal set rather than to find some solutions, and thus proposed the indicator-guided set-based optimization framework. Gong et al. [32] presented a set-based Pareto dominance relation and designed a fitness function reflecting the decision maker's preference to effectively solve MaOPs.

Very recently, more many-objective optimizer based on shift-based density estimation [33], knee point [34], and meta-objective method [35] are proposed, which are of great potentials in solving MaOPs.

2.3. Reference points-based evolutionary optimization

Existing reference points-based approaches usually adopt only one reference point to represent the decision maker's ideal solution. Wierzbicki [8] suggested a reference point approach in which the goal is to achieve a Pareto optimal solution closest to a supplied reference point of aspiration level based on solving an achievement scalarize problem. Deb et al. [9] introduced the decision maker's preference to find a preferred set of solutions near the reference point. Mohammadi et al. [10] combined decomposition strategies with reference point approaches to search for preferred regions.

Up to date, there is only a few researches on achieving the whole Pareto optimal solution set by employing multiple reference points. In [36], the reference points were generated by estimating the bounds of the Pareto front, and solutions near each reference point could be obtained in parallel. Although this priori method is very convenient, the reference points may be not appropriate at later evolution process. Wang et al. [37] proposed a co-evolution method to simultaneously optimize solutions and reference points during evolution, but the fitness value of an individual is also calculated by the traditional Pareto dominance. In the method proposed by [11], a hyperplane covering the whole objective space is obtained based on the current population, and a family of well-distributed reference points are generated on the hyperplane. In practice, the Pareto fronts of most optimization problems are not uniformly distributed in the whole objective space, and it is necessary to adopt reference points which are adaptive to various problems.

In the approach proposed in this study, with the purpose of obtaining the whole Pareto front, a series of reference points suitable for different problems are continuously generated based on the current population; moreover, the selection pressure in many-objective optimization is improved by calculating the distances between these reference points and individuals.

3. Proposed method

3.1. General framework

Algorithm 1. Reference points-based evolutionary algorithm

Input: MaOP (1);
a stopping criterion;
the population size, N;
the number of iterations, t_{grp}, to determine whether the reference points are generated or not.
Output: the final non-dominated solution set, FNS.
1: Set $t=0$;
2: Initialize a population, $P(t)=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_N\}$, by a random or problem-specific method;
3: **while** the stopping criterion is not met **do**
4: Perform genetic operators on $P(t)$ so as to generate an offspring population, $P'(t)$, with the same size as $P(t)$;
5: Let $Q(t)=P(t)\cup P'(t)$;
6: **if** $t\,\%\,t_{grp}=0$[1] **then**
7: Generate a series of reference points, $R=\{\boldsymbol{r}_1,\boldsymbol{r}_2,\ldots,\boldsymbol{r}_N\}$, based on $Q(t)$;
8: **end if**
9: Select individuals from $Q(t)$ based on R and form $P(t+1)$;
10: $t=t+1$;
11: **end while**
12: Select non-dominated solutions in $P(t)$ and form FNS;
13: Output FNS.

[1] A%B denotes the remainder when A is divided by B, where both A and B are integers.

Algorithm 1 gives a general framework of the proposed RPEA. Its basic procedure is similar to most generational EMOs. First, an initial population, $P(t)$, is formed by randomly generating N individuals. Then, genetic operators are performed to obtain an offspring population, $P'(t)$. Next, a set of reference points, R, is generated based on the combined population, $Q(t)$, every t_{grp} generations. Finally, N best solutions are selected based on the reference points for survival. It can be seen that there are two key operators in RPEA: generation of reference points (line 7) and selection of individuals (line 9). In this study, reference points with good performances in convergence and distribution are generated by making full use of information provided by the current population. In addition, superior individuals are selected based on the evaluation of each individual by calculating the distances between the reference points and the individual in the objective space. In the following two subsections, the above key operators will be illustrated in detail.

3.2. Generation of reference points

In this subsection, the concept of reference point was first presented; a method of generating reference points with good performances in convergence and distribution was then proposed.

3.2.1. Concept of reference point

Broadly speaking, the reference point is a point in the objective space that guides the evolution. There are two typical types of reference points, i.e., ideal point and nadir point, where the objective

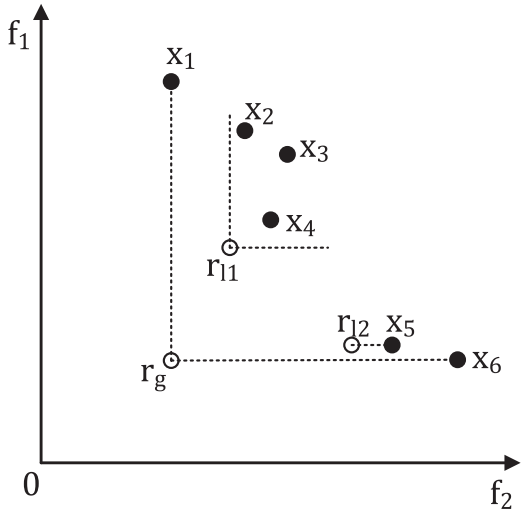


Fig. 1. Local and global ideal points.

function value(s) of an ideal point is (are) not inferior to the best objective function value(s) of the given solutions. On the contrary, the objective function value(s) of a nadir point is (are) not superior to the worst objective function value(s) of the given solutions. Clearly, the closer an individual is to an ideal point and the farther it from a nadir point, the better convergence performance the individual has.

For a point in the objective space, if its value(s) is (are) not inferior to that (those) of a part of the known solutions, it is termed as a local ideal point. For instance, when solving the problem represented as formula (1) with a MOEA, $\boldsymbol{r}_l = \left(r_l{}^1, r_l{}^2, \ldots, r_l{}^m, \ldots, r_l{}^M\right)$ is a local ideal point, when $r_l{}^m$ is

$$r_l^m = \min_{\boldsymbol{x} \in P_l} f_m(\boldsymbol{x}) - \varepsilon_m, \quad m = 1, 2, \ldots, M \tag{2}$$

where P_l is a subset of the current population. ε_m is either zero or an arbitrarily small positive. In particular, the local ideal point

$$\boldsymbol{r}_l = (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}) - \varepsilon_m, \ldots, f_M(\boldsymbol{x})) \tag{3}$$

must not be inferior to solution $\boldsymbol{x}$.

For a point in the objective space, if its value(s) is (are) not inferior to that (those) of all the known solutions, it is termed as a global ideal point. If

$$r_g^m = \min_{\boldsymbol{x} \in P_g} f_m(\boldsymbol{x}) - \varepsilon_m, \quad m = 1, 2, \ldots, M \tag{4}$$

where P_g is the current population, $\boldsymbol{r}_g = \left(r_g^1, r_g{}^2, \ldots, r_g{}^m, \ldots, r_g{}^M\right)$ is a global ideal point.

A illustration example of local and global ideal points is given in Fig. 1, where $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_5, \boldsymbol{x}_6\}$ is the current population. $\boldsymbol{r}_{l1}$ is a local ideal point for the subset $\{\boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4\}$, where both ε_1 and ε_2 are positive. $\boldsymbol{r}_{l2}$ is a local ideal point for the solution $\boldsymbol{x}_5$, where $\varepsilon_1 = 0$ and $\varepsilon_2 > 0$. $\boldsymbol{r}_g$ is a global ideal point, where both ε_1 and ε_2 are set to 0.

Similarly, a local nadir point and a global nadir point can also be defined. It is worth noting that a local ideal point can become a local nadir point, and vice versa. A reference point may be a local ideal point for some solutions; however, it may be a local nadir point for other more optimal solutions, and vice versa.

3.2.2. Generation of reference points

If only one reference point is adopted to guide the evolution, the whole Pareto front of a problem can be hardly obtained. Whereas it is relatively easy to obtain the whole Pareto front, when a series of reference points are adopted. In this study, a series of reference points with good performances in convergence and distribution are generated by making full use of information provided by the current population. These reference points are assumed as the ideal population to be obtained in the following iterations. Intuitively, compared to searching for a new solution superior to an obtained solution in several objectives, it is easier to find a solution better in just one objective. In view of this, formula (3) is adopted to generate reference points in this study, and ε_m is set as $\delta\left(f_m^{\max} - f_m^{\min}\right)$, where $\delta \in (0, 1)$ is a parameter, $f_m^{\max}$ and $f_m^{\min}$ are the maximal and minimal values of the mth objective, respectively, in the current population.

In the earlier work [13], based on formula (3), the reference points are generated in terms of all the parent individuals of the current generation, i.e., $P(t)$. However, there exit several dominated individuals in $P(t)$ despite the considerably low probability in the high-dimensional space. The reference points generated by these dominated individuals may degrade the search performance of the algorithm. In addition, some new regions explored by the offsprings are not assigned to reference points. Consequently, information may be lost during the evolution. In this study, only the non-dominated individuals in the combined population, $Q(t)$, are chosen to generate the reference points, thus resolving the above issues.

In terms of formula (3) and $Q(t)$, there will exist at most $2MN$ reference points with a better performance in convergence than the individuals in the current population. However, they may not have a good performance in distribution. The number of reference points generated was limited to N in this study. If all the $2MN$ possible reference points have been generated, selecting N well-distributed ones from them will demand a high computation complexity, due to the large number of objectives.

On account of this, a method with less computation complexity for generating reference points was proposed in this subsection, whose ideas are as follows. All the non-dominated individuals in $Q(t)$ are first sorted based on the crowding distances in each dimensional objective space, and $\lceil \alpha N \rceil$ $\left(\alpha \in \left[1/M, 1\right]\right)$ individuals with the largest crowding distances are then chosen to generate reference points by reducing the corresponding objective values according to formula (3). Finally, based on the crowding distances in the original high-dimensional objective space, N_R ($N_R \leq N$) non-dominated reference points with good performance in distribution are selected. The detailed procedure is given in Algorithm 2.

Algorithm 2. Generation of Reference Points

Input: $Q(t)$, δ, α.
Output: $R = \left\{\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_{N_R}\right\}$.
1: Set $R_\Sigma = \emptyset$;
2: Form the non-dominated individuals in $Q(t)$ into $Q'(t)$;
3: **for** $m = 1 \rightarrow M$ **do**
4: Sort the individuals in $Q'(t)$ based on the crowding distances in the m-th dimensional objective space;
5: Select $\lceil \alpha N \rceil$ individuals with the largest crowding distances;
6: According to formula (3), set $\varepsilon_m = \delta\left(f_m^{\max} - f_m^{\min}\right)$, then generate reference points based on these selected individuals, and form set R_m;
7: Let $R_\Sigma = R_\Sigma \cup R_m$;
8: **end for**
9: Form the non-dominated reference points in R_Σ into R;
10: **if** $|R| > N$ **then**
11: Delete $|R| - N$ reference points with the smallest crowding distances in M-dimensional objective space in R;
12: **end if**
13: Output R.

Fig. 2 shows an example of generating reference points, where the current population is $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_5, \boldsymbol{x}_6\}$. First, since $\boldsymbol{x}_3$ is dominated, only the other five solutions will be used for generating reference points. Then, for f_1, $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $\boldsymbol{x}_4$, $\boldsymbol{x}_6$ are chosen for their large crowding distances, and the reference points, $\boldsymbol{r}_2$, $\boldsymbol{r}_3$, $\boldsymbol{r}_5$, $\boldsymbol{r}_8$, are

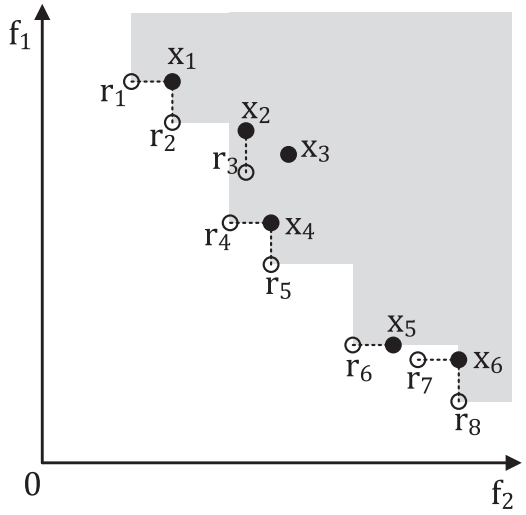


Fig. 2. Generation reference points based on the current population.

accordingly generated. Similarly, $\boldsymbol{r}_1, \boldsymbol{r}_4, \boldsymbol{r}_6, \boldsymbol{r}_7$ are generated for f_2. Next, $\boldsymbol{r}_3$ is eliminated since it is dominated by the other reference points. Finally, assuming we only need six reference points, $\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_4, \boldsymbol{r}_5, \boldsymbol{r}_6, \boldsymbol{r}_8$ are output as the reference point set, due to their large crowding distances in the bi-objective space. It can be seen from this example that the final reference set have better performance the current population.

3.3. Selection of individuals

As stated previously, the evaluation of an individual is based on the distance from reference points. In this subsection, several reference points-based approaches for evaluating individuals were first presented, and the method of selecting superior individuals was then given.

3.3.1. Evaluation of individuals

(1) Euclidean approach [9]

In this approach, the performance of an individual, $\boldsymbol{x}_i$, is evaluated by the weighted Euclidean distance measure, d_{ij}^e, which is described as follows:

$$d_{ij}^e = \sqrt{\sum_{m=1}^{M} \omega_m \left(\frac{f_m(\boldsymbol{x}_i) - r_j^m}{f_m^{\max} - f_m^{\min}} \right)^2} \tag{5}$$

where r_j^m represents the mth objective value of the jth reference point, $\boldsymbol{r}_j$, in R, $f_m^{\max}$ and $f_m^{\min}$ mean the maximal and minimal values of mth objective in the population. In addition, $\omega_m = 1/M$ refers to a weight value emphasizing each objective equally.

For the reference point, $\boldsymbol{r}_j$, a superior individual is supposed as the one with the smallest d_{ij}^e, which is appropriate when the reference point is an ideal point. However, for other individuals with better convergence, the reference point could be a nadir one, and the smaller d_{ij}^e these individuals have, the worse they are. Therefore, this Euclidean distance measure does not work well. In [13], if an individual dominates the reference point, the negative value of the Euclidean distance is used to evaluated it. However, it is not efficient enough to select superior individuals.

(2) Tchebychev approach [26]

The achievement function based on Tchebychev distance is defined as follows:

$$d_{ij}^c = \max_{m=1,2,\ldots,M} \omega_m \left(\frac{f_m(x_i) - r_j^m}{f_m^{\max} - f_m^{\min}} \right) \tag{6}$$

where d_{ij}^c represents the Tchebychev approach between $\boldsymbol{x}_i$ and the reference point, $\boldsymbol{r}_j$. If $\boldsymbol{x}_1$ dominates $\boldsymbol{x}_2$, d_{1j}^c will be smaller than d_{2j}^c. Hence, the individual with the smallest d_{ij}^c will be superior no matter whether the reference point is an ideal point or not. Moreover, a small value of d_{ij}^c also indicates that solution $\boldsymbol{x}_i$ may have a good performance in distribution since this solution is close to the reference point which is well-distributed.

It is worth noting that other reference points-based approaches for evaluating individuals in the literature can also be used in our algorithm framework. Since the main purpose of this study is to investigate the feasibility and efficiency of the algorithm framework, the Tchebychev approach is only adopted in the experiments.

3.3.2. Selection of individuals

The approach for selecting individuals is described as follows. First, the distances between the individuals and the reference points represented as formula (6) are calculated to evaluate the individuals. Then, for each reference point, the individual with the smallest distance from the reference point is selected to constitute the new population. In addition, if an individual is the best with respect to several reference points, it can be selected only once, in order to increase the diversity of the population. The above approach is similar to that in [13]. Since given the fact that the number of reference points may be smaller than the population size, all reference points are available again for selecting the rest individuals, after each reference point is assigned to an individual. The detailed procedure of selecting individuals is given in Algorithm 3. Through this method, the new population has better performances in convergence and distribution than the old. As a result, the Pareto optimal set of the problem can be obtained along with the population's evolution.

Algorithm 3. Selection of Individuals

Input: the reference points, $R = \{\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_{N_R}\}$;
the candidate solution set, $P'(t) = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{2N}\}$.
Output: the new population, $P(t+1)$.

```
Let P(t+1) = ∅, R' = ∅;
for all x_i in P'(t) do
    for all r_j in R do
        Calculate the value of d_ij^c;
    end for
end for
while |P(t+1)| < N do
    if R = ∅ then
        R = R';
        R' = ∅;
    end if
    Seek for the reference point, r_{j_min}, in R and the individual, x_{i_min}, in
    P'(t) with respond to the smallest value of d_ij^c, d_{i_min j_min}^c;
    Let P(t+1) = P(t+1) ∪ {x_{i_min}};
    Delete x_{i_min} from P'(t);
    Let R' = R' ∪ {r_{j_min}};
    Delete r_{j_min} from R;
end while;
Output P(t+1).
```

3.4. Computational complexity analysis

The main difference between RPEA and traditional EMOs is its environment selection strategy which requires to generate a series of reference points. The computational complexity analysis is provided in this subsection.

For the population size of N and an optimization problem with M objectives, when $\alpha = 1/M$, our method of generating reference points has the lowest computation complexity, represented as $O(MN \log N)$. Whereas, the complexity of selecting

Table 1
Characteristics of test problem and parameter settings in ε-MOEA, where the value of ε corresponds to the number of objectives of a problem.

Problem	M	n	Characteristics	ε in ε-MOEA
DTLZ1	6, 8, 15	M+4	Linear, multimodal	0.055, 0.056, 0.60
DTLZ2	6, 8, 15	M+9	Concave	0.23, 0.29, 0.31
DTLZ3	6, 8, 15	M+9	Concave, multimodal	0.23, 0.15, 0.92
DTLZ4	6, 8, 15	M+9	Concave, biased	0.23, 0.29, 0.31
DTLZ5	6, 8, 15	M+9	Concave, degenerate	0.11, 0.13, 0.15
DTLZ6	6, 8, 15	M+9	Concave, degenerate, biased	0.75, 1.15, 1.95
DTLZ7	6, 8, 15	M+19	Mixed, disconnected, multimodal	0.15, 0.23, 0.85

N well-distributed reference points from $2MN$ possible ones is $O\left(M^2N\log(MN)\right)$, much larger than the proposed method.

The computation of all distances between the current population and the reference points requires at most $O\left(MN^2\right)$ operations. Given the worst situation, when selecting individuals based on these distances, every individual will be checked for the smallest distance, which makes the complexity of $O\left(N^2\right)$.

When $\alpha = 1/M$, the overall complexity of RPEA is $O\left(MN^2\right)$, which is equal to that of state-of-the-art EMOs, e.g., NSGA-II [4], indicating that the algorithm is computationally efficient. However, if α is set to a larger value, RPEA will have a higher computational complexity.

3.5. Comparison with previous methods

From the above subsections, the main loop of RPEA can be summarized into the following three steps: generating new individuals, generating reference points, and selecting superior individuals. From this viewpoint, most previous reference points-based methods can be classified into this framework.

With respect to the number of reference points, most previous studies generate only one reference point, and regard it as an ideal solution that the decision maker expects to obtain. In the method proposed by [8], the decision maker is required to provide a reference point each time, based on the most preferred individual in the current population. The location of the reference point guides the algorithm to focus on a specific region of the Pareto front, and seek for a single Pareto-optimal solution tradeoff objectives. Although there are other methods that utilize two or more reference points [9,10], they focus on multiple interest regions on the Pareto front. However, in REPA, all individuals in the current population can be regarded as the ones preferred by the decision maker, then a series of reference points can be generated to guide the algorithm to focus on the whole Pareto front, and seek for a Pareto optimal solution set.

It is of necessity to generate different reference points at different stages, since the true Pareto front of an optimization problem in real-world application is often unknown in advance. By estimating the bound of each objective, Figueira et al. [36] generated reference points to depict the Pareto front. This method has a small amount of calculation since the reference points are generated only once in the beginning of the algorithm. Nevertheless, these fixed reference points may not suitable for different stages of the evolution. Deb and Jain [11] adopted reference points distributed uniformly in the whole objective space, whereas the Pareto front of a practice problem may be not. Wang et al. [37] proposed a co-evolutionary method to simultaneously optimize solutions and reference points, so that the reference points can adapt to the problem. In RPEA, the reference points are also continuously generated to provide up-to-date information of the Pareto front. However, this advantage comes at the cost of a high calculation. It can be seen from Section 3.4 that, the computation cost will be considerably high as the number of objectives and the population size increase.

In general, the proposed algorithm provides a new way to continuously generating multiple reference points suitable for various problems, and is promising to obtain the whole true Pareto front.

4. Experimental design

This section is devoted to the experimental design for investigating the performance of RPEA. The test problems and performance metrics used in the experiments are first given. Then four state-of-the-art MOEAs, ε-MOEA [15], MOEA/D [26], HypE [24], and NSGA-III [11], which are used to compare with the proposed algorithm, are briefly introduced. Finally, the commonly used parameters are set for comparative studies of these algorithms.

4.1. Test problems and performance metrics

DTLZ [38], a well-defined test problem suite, is selected in this study. DTLZ is a continuous problem suite that can be scaled to any number of objectives and decision variables, and is commonly used in many-objective optimization. The DTLZ suite is composed of several problems with various characteristics, such as linear, concave, multimodal, disconnected, biased, or degenerate Pareto fronts. These characteristics are summarized in Table 1. A detailed description of the DTLZ suite can be found in [38].

In order to measure the performance of different algorithms on these test problems, the inverted generation distance (IGD) [39], a widely used quality metrics, is chosen in this paper. IGD can measure both the convergence and the diversity of a solution set, and the smaller the value of IGD, the better the performance of the algorithm. Since IGD requires a reference set of optimal solutions, which are uniformly distributed on the Pareto optimal fronts of test problems, we set the number of reference points to around 5000, 5000, and 10,000 when M is 6, 8, and 15, respectively.

Additionally, two other metrics, generation distance (GD) and spacing (SP) [40], are adopted to measure various performances of RPEA with different parameter settings. GD evaluates the convergence of a solution set. When GD is 0, all solutions will lie in the true Pareto front. For SP, it evaluates the uniformity of a solution set. When SP is 0, the solution set will be uniformly distributed.

4.2. Four comparative algorithms

Four state-of-the-art MOEAs were chosen as comparative algorithms to evaluate the performances of the proposed RPEA:

(1) ε-MOEA [15] is a steady-state algorithm using the ε-dominance relationship, and it has been found to perform well on MaOPs. Having divided the objective space into a number of hyperboxes with the size of ε, ε-MOEA assigns each hyperbox at most one solution on the basis of ε-dominance among hyperboxes and Pareto dominance within a hyperbox. This ε-dominance implementation is very similar to our proposed method for generating reference points. Hence, it is meaningful to compare RPEA with ε-MOEA.
(2) MOEA/D [26] is an aggregation-based algorithm, which is suitable for tackling MaOPs. In fact, when using reference vectors to convert the problem into a series of scalar optimization problems, MOEA/D can also be regarded as a reference points-based algorithm. The main difference between MOEA/D and RPEA is

Table 2
Settings of α and δ in RPEA and the population size, where p_1 and p_2 are parameters controlling the number of reference points (vectors) in NSGA-III and MOEA/D

M	p_1	p_2	N	α	δ
6	4	1	132	0.4	0.05
8	3	2	156	0.4	0.1
15	2	1	135	0.2	0.15

that, MOEA/D adopts only one reference point and a number of reference vectors, whereas RPEA adopts a number of reference points and only one reference vector. Therefore, MOEA/D is also chosen as a peer algorithm.

(3) HypE [24] is an indicator-based algorithm, using the hypervolume metric to guide the search. HypE adopts the Monte Carlo simulation to approximate the exact hypervolume value, significantly reducing the time spent in calculating hypervolume, which makes it very competitive in many-objective optimization. Although HypE has no similarities with RPEA, it is helpful to evaluate the proposed method by comparing with it.

(4) NSGA-III [11] was obtained by improving its former version, NSGA-II, for handling MaOPs. NSGA-III uses a reference points-based selection criterion instead of the density-based counterpart in the original NSGA-II. These reference points are uniformly distributed on a hyperplane and applied for the maintenance of population diversity. It is quite necessary to compare RPEA with this typical reference points-based algorithm.

4.3. Parameter settings

There are three types of parameters in the experiments.

The first is the parameters of evolutionary operators. The evolutionary and selection operators are two key components in a MOEA. The former generates new solutions, and the latter selects good ones among them. What makes the comparative MOEAs different is in fact their different selection operators. To compare the effectiveness of these selection operators, the compared MOEAs should have the same capability in generating solutions. Therefore, the parameters of evolutionary operators are required to be same for all the comparative MOEAs. In this study, the operators for crossover and mutation are simulated binary crossover and polynomial mutation with both distribution indexes of 20. The crossover and mutation probabilities are 1.0 and $1/n$, respectively.

The second is the parameters related to the computational consumption, i.e., the population size and the number of evaluations of solutions. If a comparative MOEA is given more computational resource than the others, the comparison will be unfair. Therefore, these parameters should also be set the same values. In NSGA-III and MOEA/D, to avoid all the generated reference points (vectors) being located along the boundary of the Pareto optimal front for a problem with a large number of objectives, the strategy of two-layered reference points (vectors) is used, thus the population size of these two algorithms cannot be arbitrarily specified. As a result, we set the population size of the other algorithms to the same value as those of NSGA-III and MOEA/D. The setting of the population size, N, and the parameters for controlling the number of reference points (vectors) are listed in Table 2. In addition, since the population size is determined by the value of ε in ε-MOEA, we also adjust the value of ε to different problems (shown in Table 1). The termination criterion is that the number of evaluations reaches to the predefined one. For DTLZ1, DTLZ3 and DTLZ6, it is set $N \times 1000$, and for DTLZ2, DTLZ4, DTLZ5, and DTLZ7, $N \times 300$.

The third is the unique parameters in each MOEA. In RPEA, the smallest distance between the individual and the reference points is employed as the tournament strategy in mating selection; t_{grp} is set to 1 so that the reference points can be generated in each iteration and the proposed method can have good performances; the settings of α and δ are shown in Table 2. For the unique parameters in the other algorithms, we set them by following the guidelines provided in their original studies to achieve the best performances of these algorithms. In MOEA/D, the Tchebychev approach is selected as the scalarizing method, and the neighborhood size is set to $N \times 0.1$ as suggested in [41]. Following the practice in [24], in HypE, the number of sampling points to estimate hypervolume is set to 10,000 in order to trade off the accuracy of the estimates and the available computing resources.

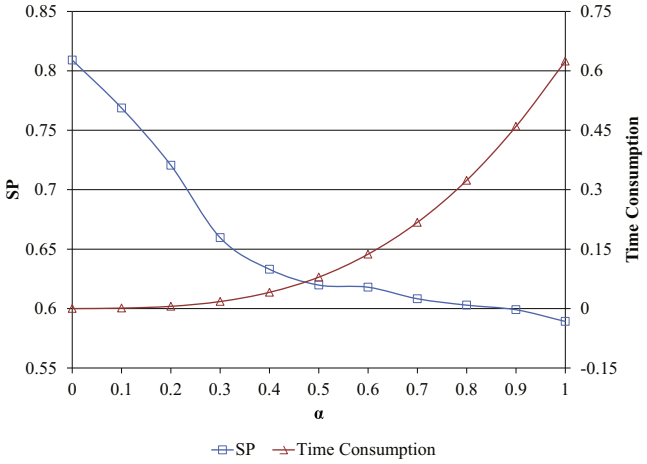


Fig. 3. Metric SP and time consumption of different methods in generating reference points.

Each algorithm runs 20 times on each optimization problem, and the mean values and variances of each metric described in Section 4.2 are calculated. In addition, the Wilcoxon's rank sum test is employed to determine whether one algorithm has a significant difference with the other on a metric, and the null hypothesis is rejected at a significant level of 0.05.

5. Experimental results and analysis

In this section, the performances of RPEA are investigated according to the experimental design described in the previous section. The experiments are divided into the following two parts. The first investigates the effects of two parameters for generating references points in the proposed algorithm, and the second compares RPEA with the other four state-of-the-art MOEAs.

5.1. Studies of different parameter settings in RPEA

5.1.1. Effect of parameter α

In this subsection, the effect of parameter α on the distribution of references points is investigated. In ten-dimensional objective space, 100 points are randomly generated and their values were in the range of [0,1]. These points are assumed to be the individuals of a population in the objective space. Along with the value of α changing from 0.1 to 1.0 with the step size of 0.1, metric SP of the reference points and the time consumption of generating them by the proposed method are investigated. In addition, metric SP and the time consumption obtained by the random method are also investigated as a contrast. Fig. 1 depicts metric SP and the time consumption of different methods in generating reference points, where $\alpha = 0$ represents the random method. The left Y-axis is metric SP, and the right Y-axis is time consumption in seconds.

Fig. 3 reports that, (1) the curve of SP is steep in the beginning, but flat when $\alpha > 0.5$, suggesting that increasing α contributes to

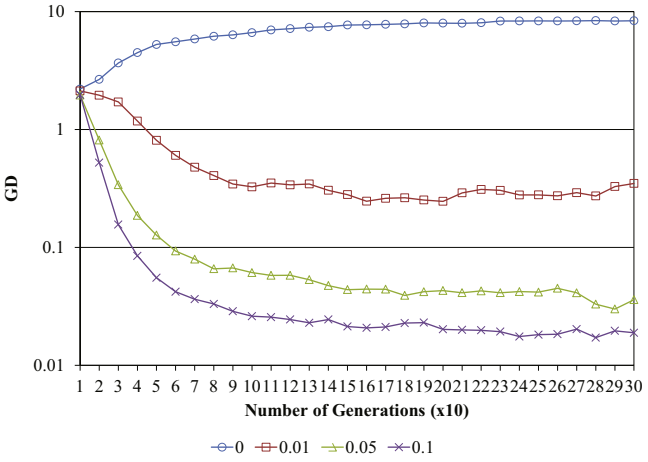


Fig. 4. Curves of GD w.r.t. number of generations for different δ.

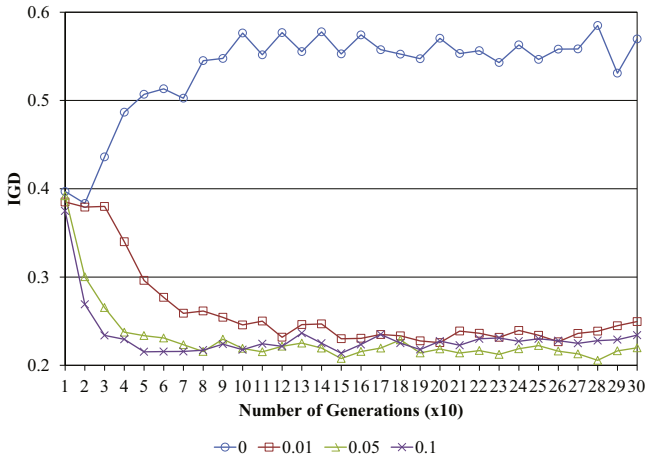


Fig. 5. Curves of IGD w.r.t. number of generations for different δ.

the distribution of the reference points, but as α becomes larger, the effect is less significant; (2) as α increases, the time consumption increases, and the speed of increasing is rapid as well, which means that a large value of α can substantially increase the time consumption in generating the reference points; (3) the time consumption of the random method is almost 0, but a large value of SP indicates a poor distribution of reference points. Therefore, to balance the distribution performance of reference points and the time consumption, we suggest to set α between 0.2 and 0.6 in RPEA.

5.1.2. Effect of parameter δ

The essential parameter, δ, determines the performance of the reference points generated based on the current population, so the performances of RPEA. In this subsection, the effect of δ is investigated by tackling 6-objective DTLZ2. It is worth noting that similar results can be obtained for the other test cases. Figs. 4 and 5 depict the curves of GD and IGD with respect to the number of generations when δ is set 0, 0.01, 0.05, and 0.1, respectively.

Figs. 4 and 5 report that, (1) the larger δ is, the more rapidly GD decreases, suggesting that a large value of δ is beneficial to improve the efficiency of the evolution; (2) In the first 50 generations, the IGD value obtained by the method with a large value of δ also declines fast, contributed by the good convergence; (3) After about 50 generations, the curves of IGD become flat. Then, the method with $\delta = 0.05$ achieves the best IGD value in the end. This indicates that a large value of δ may also count against the distribution performance and a proper value of δ is conducive to balance convergence and diversity. (4) when $\delta = 0$, the reference points are actually the individuals in the current population. In this case, the population can hardly converge towards the Pareto front driven by itself, verifying that the local ideal points are necessary to be adopted for guiding evolution. To conclude, δ should be set to a proper positive value to achieve a good overall performance.

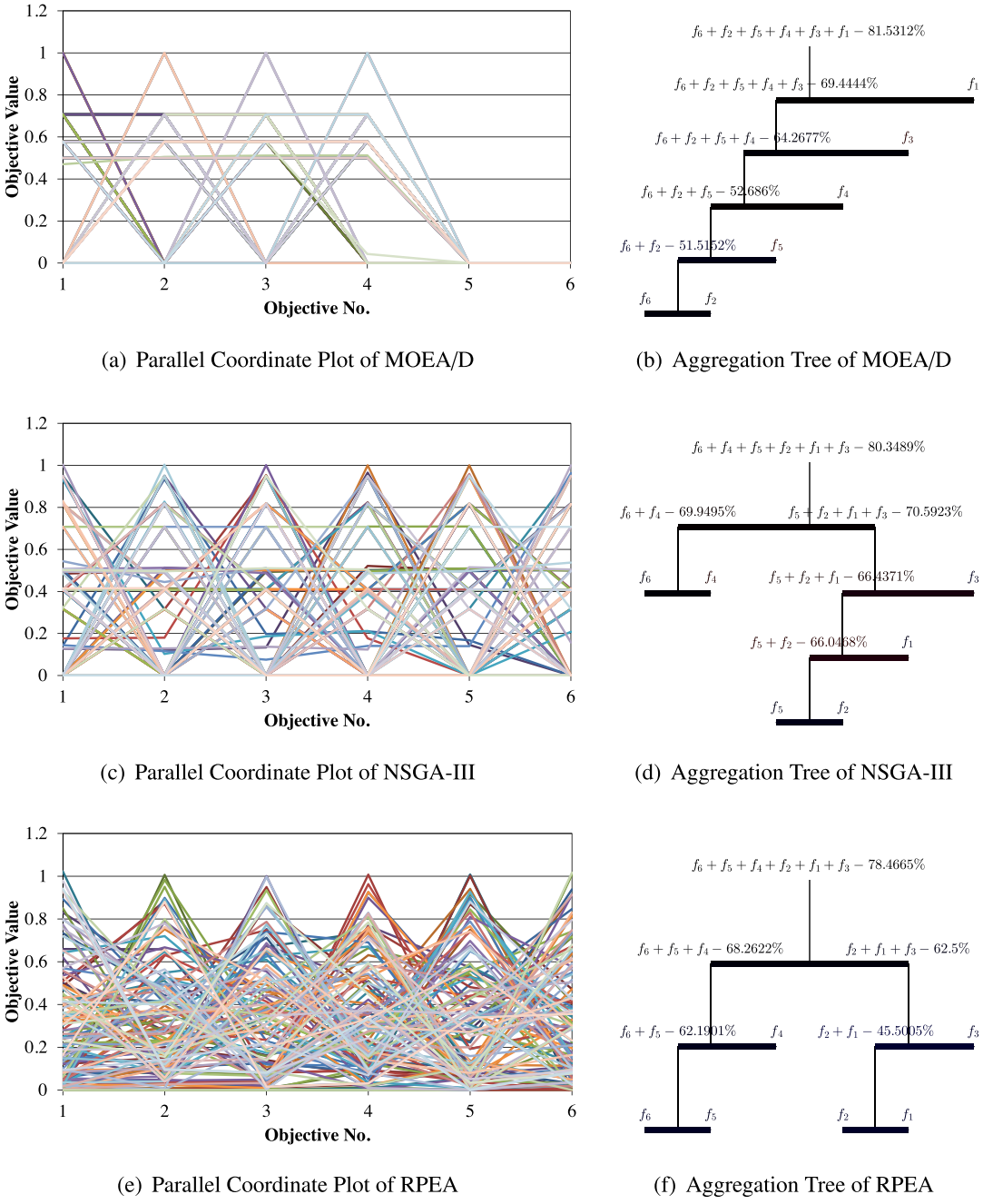
5.2. Comparison to other methods

In this section, the proposed method, RPEA, was compared with ε-MOEA, MOEA/D, HypE, and NSGA-III. Table 3 shows metric IGD in terms of the mean and standard deviation obtained by different algorithms on the DTLZ problem suite, where DTLZA-B refers to DTLZA with B objectives, the boldface data are the best among these methods, and those labeled by '†' mean data significantly different from RPEA's.

Although DTLZ1 has a linear and simple Pareto front, it is a multimodal problem with a great number of local optima. Table 3 reports that, on DTLZ1-6, NSGA-III and ε-MOEA perform well, followed by MOEA/D, RPEA, and HypE. For this problem with 8 and 15 objectives, NSGA-III remains first, and RPEA and MOEA/D also obtain

Table 3
Metric IGD of different methods.

Problem	ε-MOEA	MOEA/D	HypE	NSGA-III	RPEA
DTLZ1-6	8.809E−2(1.2E−3)†	1.434E−1(7.0E−3)†	2.774E−1(8.0E−2)†	**7.669E−2(3.1E−4)**†	1.794E−1(1.8E−2)
DTLZ1-8	1.266E−1(3.7E−3)†	1.783E−1(6.0E−3)†	3.241E−1(3.8E−1)†	**9.293E−2(1.2E−2)**†	2.294E−1(1.6E−2)
DTLZ1-15	7.578E−1(2.1E−1)†	2.259E−1(1.1E−2)	4.061E−1(1.6E−1)†	**1.612E−1(5.3E−2)**†	2.413E−1(2.0E−2)
DTLZ2-6	2.925E−1(3.1E−3)†	4.215E−1(1.8E−2)†	4.734E−1(9.3E−3)†	3.066E−1(1.2E−3)†	**2.697E−1(8.0E−3)**
DTLZ2-8	4.200E−1(4.4E−3)†	5.060E−1(3.3E−2)†	6.327E−1(1.7E−2)†	3.797E−1(2.5E−2)†	**3.620E−1(1.2E−3)**
DTLZ2-15	5.321E−1(2.6E−2)	7.831E−1(7.9E−2)†	8.712E−1(2.8E−2)†	7.197E−1(3.0E−2)†	**5.305E−1(2.2E−2)**
DTLZ3-6	3.873E−1(1.7E−1)†	4.255E−1(1.8E−2)†	5.616E+1(7.8E+0)†	**3.070E−1(1.4E−3)**†	6.700E−1(6.3E−2)
DTLZ3-8	7.181E−1(8.8E−1)	**5.251E−1(5.1E−2)**†	7.342E+1(1.2E+1)†	7.259E−1(7.3E−1)†	7.371E−1(1.0E−1)
DTLZ3-15	6.777E+1(1.7E+1)†	**7.883E−1(5.7E−2)**†	2.938E+1(1.1E+1)†	4.518E+0(2.0E+0)†	9.776E−1(2.2E−1)
DTLZ4-6	3.509E−1(7.2E−2)†	6.445E−1(1.4E−1)†	4.629E−1(1.1E−2)†	3.567E−1(9.3E−2)†	**2.678E−1(1.6E−3)**
DTLZ4-8	4.328E−1(8.7E−3)†	7.174E−1(1.1E−1)†	6.200E−1(1.4E−2)†	4.296E−1(8.3E−2)†	**3.524E−1(4.6E−3)**
DTLZ4-15	6.045E−1(4.5E−2)†	8.769E−1(4.8E−2)†	8.166E−1(3.8E−3)†	6.954E−1(3.8E−2)†	**5.367E−1(1.4E−2)**
DTLZ5-6	1.187E−1(1.5E−2)†	4.059E−2(2.5E−3)†	6.546E−2(1.1E−2)†	2.696E−1(8.3E−2)†	**3.884E−2(1.7E−2)**
DTLZ5-8	1.619E−1(9.2E−3)†	4.622E−2(5.2E−3)†	1.077E−1(1.7E−2)†	3.331E−1(9.5E−2)†	**4.446E−2(2.1E−2)**
DTLZ5-15	1.745E−1(3.0E−2)†	**4.317E−2(8.5E−3)**†	6.790E−1(4.4E−2)†	2.729E−1(3.2E−2)†	5.436E−2(2.5E−2)
DTLZ6-6	7.197E−1(2.3E−1)†	**1.251E−1(2.8E−2)**†	5.699E−1(8.9E−2)†	4.451E−1(1.4E−1)†	2.252E−1(2.4E−1)
DTLZ6-8	5.995E−1(5.0E−2)†	**9.983E−2(2.3E−2)**†	1.344E+0(6.4E−2)†	1.144E+0(7.0E−1)†	3.616E−1(1.7E−1)
DTLZ6-15	6.472E−1(1.2E−1)	**1.103E−1(2.3E−2)**†	2.942E+0(3.2E−1)†	1.031E+0(7.6E−1)†	6.576E−1(2.2E−1)
DTLZ7-6	**4.715E−1(1.8E−1)**†	7.530E−1(5.6E−2)†	4.968E−1(6.0E−3)†	6.371E−1(4.3E−2)†	5.815E−1(2.9E−1)
DTLZ7-8	**4.500E−1(1.9E−1)**†	1.202E+0(4.1E−1)†	6.896E−1(9.8E−3)†	1.110E+0(1.2E−1)†	1.399E+0(3.9E−1)
DTLZ7-15	9.393E+0(1.6E+0)†	**1.832E+0(3.1E−1)**†	2.075E+0(2.5E−1)†	6.234E+0(4.0E−2)†	4.185E+0(8.2E−1)



(a) Parallel Coordinate Plot of MOEA/D

(b) Aggregation Tree of MOEA/D

(c) Parallel Coordinate Plot of NSGA-III

(d) Aggregation Tree of NSGA-III

(e) Parallel Coordinate Plot of RPEA

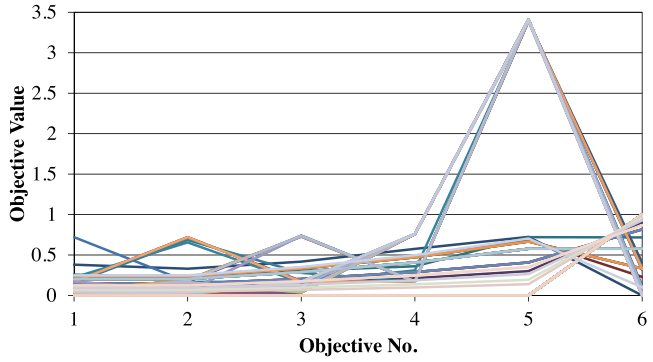
(f) Aggregation Tree of RPEA

Fig. 6. The Pareto solutions obtained by MOEA/D, NSGA-III, and RPEA on 6-objective DTLZ4, shown by the parallel coordinates slots and aggregation trees.

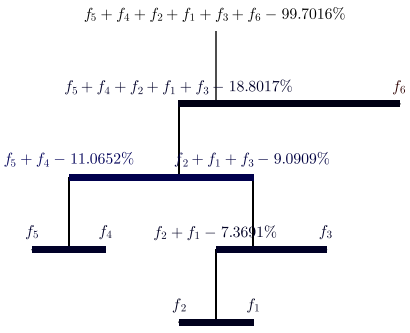
acceptable results, whereas ε-MOEA achieves poor performance. The reason that RPEA does not perform as well as NSGA-III and MOEA/D on this problem is that RPEA has a normalization procedure. For DTLZ1, the ranges of its objective values are much larger than those of its Pareto front, and the normalization procedure is prone to transforming the original objectives into their wrong scales. Similar results are also observed in [42].

With the same spherical Pareto front, problems DTLZ2, DTLZ3, and DTLZ4 are designed to test different capabilities of an algorithm, where DTLZ2 is relatively easier to solve. Based on DTLZ2, a huge number of local optima are introduced to DTLZ3, and it is a big challenge for an algorithm to achieve its Pareto front. The non-uniform solutions of DTLZ4 in distribution make an algorithm difficult to maintain diversity in the objective space. From the IGD results shown in Table 3, for DTLZ2, RPEA significantly outperforms the other in most situations. For the same reason when solving DTLZ1, the performance of RPEA decreases on DTLZ3, whereas NSGA-III shows a competitive performance on DTLZ3-6, and MOEA/D on DTLZ3-8 and DTLZ3-15. It is worth noting that NSGA-III performs poorly on DTLZ3-15, since the non-dominated sorting employed in NSGA-III has a low efficiency in selecting solutions with a good convergence performance when the number of objective is very large. For DTLZ4, RPEA achieves the best performance, indicating its strong ability in solving problems with irregular Pareto fronts.

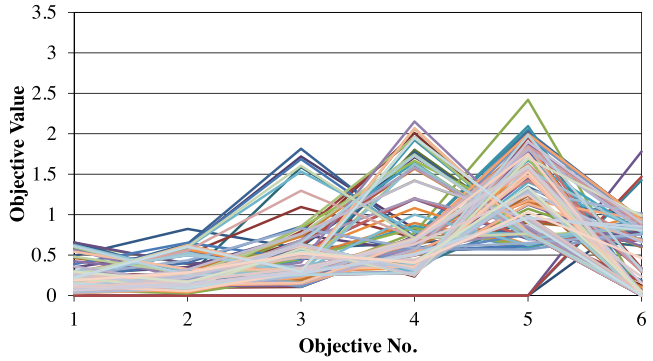
The Pareto fronts of problems DTLZ5 and DTLZ6 are degenerate curves, which are designed to test the ability of an algorithm in seeking for a lower dimensional Pareto front while working in a higher dimensional objective space. DTLZ6 is much harder to solve than DTLZ5 because of bias in its objectives. From Table 3, for



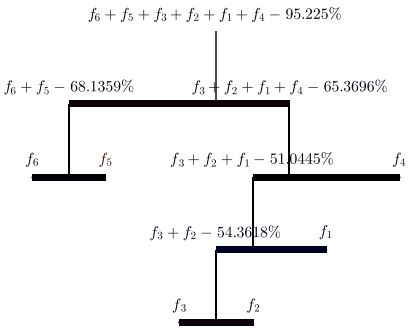
(a) Parallel Coordinate Plot of MOEA/D



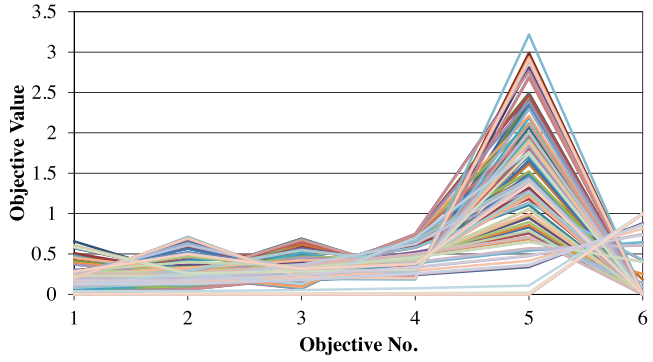
(b) Aggregation Tree of MOEA/D



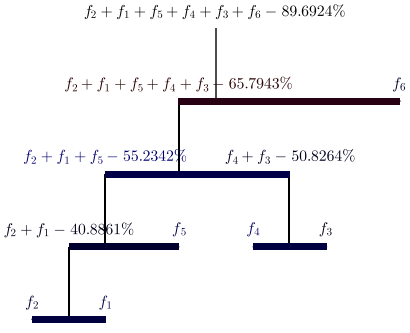
(c) Parallel Coordinate Plot of NSGA-III



(d) Aggregation Tree of NSGA-III



(e) Parallel Coordinate Plot of RPEA



(f) Aggregation Tree of RPEA

Fig. 7. The Pareto solutions obtained by MOEA/D, NSGA-III, and RPEA on 6-objective DTLZ5, shown by the parallel coordinates slots and aggregation trees.

DTLZ5, both RPEA and MOEA/D generally perform better than the other three. More specifically, RPEA obtains lower IGD values on 6- and 8-objective DTLZ5, while MOEA/D outperforms on 15-objective DTLZ5. NSGA-III is ineffective on DTLZ5 since its reference points cover the whole Pareto front. RPEA does not show advantages on DLTZ6 for its normalization procedure.

Problem DTLZ7 has a number of disconnected Pareto front regions in order to test the ability of an algorithm to maintain subpopulations in disconnected portions of the objective space. For this problem, ε-MOEA shows advantages on the low-dimensional instances, and MOEA/D performs better when a larger number of objectives are involved. The results of RPEA are acceptable.

In addition, to intuitively investigate the performances of reference points(vectors)-based algorithms, i.e., MOEA/D, NSGA-III, and RPEA, the Pareto solution set obtained by these algorithms on DTLZ4-6 and DTLZ5-6 in a single run are visualized by the parallel coordinates slots and aggregation trees [22] in Figs. 6 and 7. This particular run is associated with the result closest to the mean IGD value. From the aggregation trees in Fig. 6, we can see that all the three algorithms can treat the objectives conflicting with each other. However, from the parallel coordinate slots in Fig. 6, MOEA/D struggles to maintain the diversity of its solutions. NSGA-III performs better than MOEA/D, but not as well as RPEA. From the aggregation trees in Fig. 7, both MOEA/D and RPEA can correctly distinguish f_6 as the most conflicting one from the other. However, RPEA also regards the rest objectives as conflicting with each other rather than harmonious. NSGA-III fails to distinguish the relationship among objectives, since the reference points compel solutions to widely spread in the objective space. From the parallel coordinate slots in Fig. 7, MOEA/D actually performs poorly,

since there are a number of solutions overlapping with each other. NSGA-III struggles to converge to the true Pareto front due to its failure in distinguishing the relationship among objectives. RPEA achieves relatively good performances, although a small number of solutions cannot converge to the true Pareto front. These results show that RPEA has a good adaptability to problems with irregular or degenerate Pareto fronts, whereas the other reference points-based algorithms have not. That is because their reference points (vectors) are predefined and uniformly distributed in the whole objective space.

Overall, based on the studies on problems with various characteristics, RPEA is very competitive in many-objective optimization, especially when a problem has a irregular or degenerate Pareto front, and it outperforms the other four state-of-the-art algorithms in 8 out of 21 test instances.

6. Conclusions

This paper exploited the potential of the reference points in handling MaOPs. The proposed method, RPEA, can mainly be characterized as: (1) adaptively generating a series of reference points with good convergence and distribution based on the evolution of a population; (2) greatly increasing the selection pressure toward the Pareto front by calculating the distances between the reference points and the individuals in the environment selection process.

The propose method was applied to seven benchmark MaOPs, and compared with the other four state-of-the-art method to evaluate its performance. The results reveal that RPEA is very competitive to the others in terms of seeking for a solution set with good approximation and distribution in many-objective optimization. Furthermore, the effects of two key parameters, α and δ, are experimentally investigated. The results show that the reference points with good performances can be obtained in short time; and RPEA can achieve a good tradeoff among the convergence and the diversity

It is worth mentioning that RPEA has been applied only to optimization problems with numerical objectives. Its effectiveness in other optimization problems, especially in engineering optimization, should further be confirmed. In addition, RPEA adopts the key parameter, δ, which will affect its performance, as mentioned in the experiments above. If appropriate methods are employed to adaptively adjust the value of δ during the evolution, the performance of RPEA will further be improved, which is the focus of our future work. Moreover, the Tchebychev approach was only adopted in this study and it may not be appropriate to all kinds of problems. Hence, adopting other reference points-based approaches, e.g., normal-boundary intersection [43] and g-dominance [44], to evaluate individuals, so as to have a deeper insight into the behavior of RPEA is also our further research work.

Acknowledgements

This work was jointly supported by National Natural Science Foundation of China with grant No. 61375067, 61673404, 61105063, and 61075061, National Basic Research Program of China (973 Program) with grant No. 2014CB046306-2, Natural Science Foundation of Jiangsu Province with grant No. BK2012566, Innovation Project for College Graduates of Jiangsu Province with grant No. KYLX15_1422.

References

[1] P.M. Reed, D. Hadka, Evolving many-objective water management to exploit exascale computing, Water Resour. Res. 50 (10) (2014) 8367–8373.
[2] A.S. Sayyad, T. Menzies, H. Ammar, On the value of user preferences in search-based software engineering: a case study in software product lines, in: 2013 35th International Conference on Software Engineering (ICSE), IEEE, 2013, pp. 492–501.
[3] Y.-Y. Han, D.-w. Gong, X.-Y. Sun, Q.-K. Pan, An improved NSGA-II algorithm for multi-objective lot-streaming flow shop scheduling problem, Int. J. Prod. Res. 52 (8) (2014) 2211–2231.
[4] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
[5] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the strength Pareto evolutionary algorithm, Tech. Rep., Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische Informatik und Kommunikationsnetze (TIK), 2001.
[6] C.C. Coello, M.S. Lechuga, MOPSO: a proposal for multiple objective particle swarm optimization, in: Proceedings of the 2002 Congress on Evolutionary Computation, 2002. CEC'02, vol. 2, IEEE, 2002, pp. 1051–1056.
[7] H. Ishibuchi, N. Tsukamoto, Y. Nojima, Evolutionary many-objective optimization: a short review, in: IEEE Congress on Evolutionary Computation, 2008, pp. 2419–2426.
[8] A.P. Wierzbicki, The use of reference objectives in multiobjective optimization, in: Multiple Criteria Decision Making Theory and Application, Springer, 1980, pp. 468–486.
[9] K. Deb, J. Sundar, N. Udaya Bhaskara Rao, S. Chaudhuri, Reference point based multi-objective optimization using evolutionary algorithms, Int. J. Comput. Intell. Res. 2 (3) (2006) 273–286.
[10] A. Mohammadi, M.N. Omidvar, X. Li, Reference point based multi-objective optimization through decomposition, in: IEEE Congress on Evolutionary Computation, IEEE, 2012, pp. 1–8.
[11] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach. Part I: Solving problems with box constraints, IEEE Trans. Evol. Comput. 18 (4) (2013) 577–601.
[12] A. Britto, A. Pozo, Using reference points to update the archive of MOPSO algorithms in Many-Objective Optimization, Neurocomputing 127 (2014) 78–87.
[13] Y. Liu, D. Gong, X. Sun, Y. Zhang, A reference points-based evolutionary algorithm for many-objective optimization, in: Proceedings of the 2014 Conference Companion on Genetic and Evolutionary Computation Companion, ACM, 2014, pp. 1053–1056.
[14] H. Ishibuchi, N. Tsukamoto, Y. Nojima, Evolutionary many-objective optimization: a short review, in: IEEE Congress on Evolutionary Computation, 2008, pp. 2419–2426.
[15] K. Deb, M. Mohan, S. Mishra, Evaluating the ϵ-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions, Evol. Comput. 13 (4) (2005) 501–525.
[16] B. Malekmohammadi, B. Zahraie, R. Kerachian, Ranking solutions of multi-objective reservoir operation optimization models using multi-criteria decision analysis, Expert Syst. Appl. 38 (6) (2011) 7851–7863.
[17] S. Yang, M. Li, X. Liu, J. Zheng, A grid-based evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 17 (5) (2013) 721–736.
[18] A. López Jaimes, C.A. Coello Coello, Study of preference relations in many-objective optimization, in: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, ACM, 2009, pp. 611–618.
[19] D. Brockhoff, E. Zitzler, Improving hypervolume-based multiobjective evolutionary algorithms by using objective reduction methods, in: IEEE Congress on Evolutionary Computation, IEEE, 2007, pp. 2086–2093.
[20] C. Zhou, J. Zheng, K. Li, H. Lv, Objective reduction based on the least square method for large-dimensional multi-objective optimization problem, in: International Conference on Natural Computation, vol. 4, IEEE, 2009, pp. 350–354.
[21] T. Murata, A. Taki, Examination of the performance of objective reduction using correlation-based weighted-sum for many objective knapsack problems, in: International Conference on Hybrid Intelligent Systems, IEEE, 2010, pp. 175–180.
[22] A.R. de Freitas, P.J. Fleming, F.G. Guimar aes, Aggregation trees for visualization and dimension reduction in many-objective optimization, Inf. Sci. 298 (2015) 288–314.
[23] E. Zitzler, S. Künzli, Indicator-based selection in multiobjective search, in: Parallel Problem Solving from Nature-PPSN VIII, Springer, 2004, pp. 832–842.
[24] J. Bader, E. Zitzler, HypE: An algorithm for fast hypervolume-based many-objective optimization, Evol. Comput. 19 (1) (2011) 45–76.
[25] E.J. Hughes, Multiple single objective Pareto sampling, in: IEEE Congress on Evolutionary Computation, vol. 4, IEEE, 2003, pp. 2678–2684.
[26] Q. Zhang, H. Li, MOEA/D: a multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731.
[27] C. Dai, Y. Wang, A new uniform evolutionary algorithm based on decomposition and CDAS for many-objective optimization, in: Knowledge-Based Systems, 2015.
[28] H. Sato, H.E. Aguirre, K. Tanaka, Controlling dominance area of solutions and its impact on the performance of MOEAs, in: Evolutionary Multi-Criterion Optimization, Springer, 2007, pp. 5–20.
[29] Y. Yuan, H. Xu, B. Wang, B. Zhang, X. Yao, Balancing convergence and diversity in decomposition-based many-objective optimizers, IEEE Trans. Evol. Comput. 20 (2) (2016) 180–198.
[30] J. Bader, D. Brockhoff, S. Welten, E. Zitzler, On using populations of sets in multiobjective optimization, in: Evolutionary Multi-Criterion Optimization, Springer, 2009, pp. 140–154.

[31] E. Zitzler, L. Thiele, J. Bader, On set-based multiobjective optimization, IEEE Trans. Evol. Comput. 14 (1) (2010) 58–79.
[32] D. Gong, X. Ji, X. Sun, Solving many-objective optimization problems using set-based evolutionary algorithms, Acta Electr. Sin. 42 (1) (2014) 77–83.
[33] M. Li, S. Yang, X. Liu, Shift-based density estimation for Pareto-based algorithms in many-objective optimization, IEEE Trans. Evol. Comput. 18 (3) (2014) 348–365.
[34] X. Zhang, Y. Tian, Y. Jin, A knee point-driven evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 19 (6) (2015) 761–776.
[35] M. Li, S. Yang, X. Liu, Bi-goal evolution for many-objective optimization problems, Artif. Intell. 228 (2015) 45–65.
[36] J.R. Figueira, A. Liefooghe, E.-G. Talbi, A.P. Wierzbicki, A parallel multiple reference point approach for multi-objective optimization, Eur. J. Oper. Res. 205 (2) (2010) 390–400.
[37] R. Wang, R.C. Purshouse, P.J. Fleming, Preference-inspired coevolutionary algorithms for many-objective optimization, IEEE Trans. Evol. Comput. 17 (4) (2013) 474–494.
[38] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable test problems for evolutionary multiobjective optimization, in: Evolutionary Multiobjective Optimization, Springer, London, 2005, pp. 105–145.
[39] Q. Zhang, A. Zhou, S. Zhao, P.N. Suganthan, W. Liu, S. Tiwari, Multiobjective optimization test instances for the CEC 2009 special session and competition, University of Essex, Colchester, UK and Nanyang Technological University, Singapore, Special Session on Performance Assessment of Multi-Objective Optimization Algorithms, Technical Report.
[40] D.A. Van Veldhuizen, G.B. Lamont, On measuring multiobjective evolutionary algorithm performance, in: IEEE Congress on Evolutionary Computation, vol. 1, IEEE, 2000, pp. 204–211.
[41] H. Ishibuchi, Y. Hitotsuyanagi, N. Tsukamoto, Y. Nojima, Many-objective test problems to visually examine the behavior of multiobjective evolution in a decision space, in: Parallel Problem Solving from Nature, PPSN XI, Springer, 2010, pp. 91–100.
[42] Y. Yuan, H. Xu, B. Wang, X. Yao, A new dominance relation based evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 0 (0) (2015), 0-0.
[43] I. Das, J.E. Dennis, Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems, SIAM J. Optim. 8 (3) (1998) 631–657.
[44] J. Molina, L.V. Santana, A.G. Hernández-Díaz, C.A. Coello Coello, R. Caballero, G-dominance: reference point based dominance for multiobjective metaheuristics, Eur. J. Oper. Res. 197 (2) (2009) 685–692.