A Pareto-based many-objective evolutionary algorithm using space partitioning selection and angle-based truncation

Hui Bai [a],[*], Jinhua Zheng [a],[b],[*], Guo Yu [c], Shengxiang Yang [d], Juan Zou [a]

[a] Key Laboratory of Intelligent Computing and Information Processing, Ministry of Education, Information Engineering College of Xiangtan University, Xiangtan, Hunan Province, China
[b] Hunan Provincial Key Laboratory of Intelligent Information Processing and Application, Hengyang, 421002, China
[c] Department of Computer Sciences, University of Surrey, Guildford, U.K.
[d] School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, U.K.

ARTICLE INFO

ABSTRACT

Evolutionary algorithms (EAs) have shown to be efficient in dealing with many-objective optimization problems (MaOPs) due to their ability to obtain a set of compromising solutions which not only converge toward the Pareto front (PF), but also distribute well. The Pareto-based multi-objective evolutionary algorithms are valid for solving optimization problems with two and three objectives. Nevertheless, when they encounter many-objective problems, they lose their effectiveness due to the weakening of selection pressure based on the Pareto dominance relation. Our major purpose is to develop more effective diversity maintenance mechanisms which cover convergence besides dominance in order to enhance the Pareto-based many-objective evolutionary algorithms. In this paper, we propose a Pareto-based many-objective evolutionary algorithm using space partitioning selection and angle-based truncation, abbreviated as SPSAT. The space partitioning selection increases selection pressure and maintains diversity simultaneously, which we realize through firstly dividing the normalized objective space into many subspaces and then selecting only one individual with the best proximity estimation value in each subspace. To further enhance convergence and diversity, the angle-based truncation calculates the angle values of any pair of individuals in the critical layer and then gradually removes the individuals with the minimum angle values. From the comparative experimental results with six state-of-the-art algorithms on a series of well-defined optimization problems with up to 20 objectives, the proposed algorithm shows its competitiveness in solving many-objective optimization problems.

© 2018 Elsevier Inc. All rights reserved.

1. Introduction

In many real-world applications, there are always many optimization problems involving multiple objectives, sometimes with up to 10 or 20 objectives. For example, in environmental engineering, in order to generate alternatives for site retail and service facilities [40], five criteria need to be considered: height, geology, aspect, land use and the distance between two urban cities. In control engineering, when designing the time and frequency of linear control systems [44], nine objectives

are considered: stability, closed-loop sensitivity, disturbance rejection, plant uncertainty, actuator saturation, rise time, overshoot, settling time, and steady state error. Moreover, multi-objective optimization problems exist in many areas including engineering, science, industry and finance [9]. Recently, using evolutionary algorithms (EAs) have proven to be a powerful approach for solving multi-objective optimization problems (MOPs) due to their ability to obtain a set of well-converged and well-distributed compromising solutions after a single run.

In Pareto optimization, the purpose is to find a set of Pareto optimal solutions. It has to optimize these conflicting objectives simultaneously; thus, a set of trade-off solutions among subproblems are obtained according to Pareto dominance relation. Given two solutions x_1 and x_2 for a MOP, x_1 is said to Pareto dominate x_2 (x_1 is better than x_2), if x_1 is not worse than x_2 in all objectives, and x_1 is better than x_2 in at least one objective. The final optimal solutions in the decision space are defined as the Pareto set, and the mapping of the Pareto set in the objective space is referred to as the Pareto front. The Pareto dominance relation gives a partial order of solutions in the objective space. Taking account of a solution in the objective space, another solution compared to it may locate in three places: dominating space, dominated space, and non-dominated space. So the proportion e of non-dominated space in M-dimensional objective space is calculated as: $e = (2^M - 2)/2^M$. With an increase in the number of objectives, the non-dominated space increases exponentially. Therefore, only using the Pareto dominance relation to estimate convergence performance between solutions is not practical for many-objective evolutionary algorithms.

In evolutionary multi-objective optimization (EMO), two goals, converging to the Pareto front and gaining a high degree of diversity, are pursued in terms of the obtained solution set. The traditional Pareto-based multi-objective evolutionary algorithms (MOEAs) are very effective in solving optimization problems with two and three objectives, like NSGA-II [16], SPEA2 [52], PEAS-II [10], M-PAES [32], etc. Yet, when encountering MaOPs (MOPs with more than three objectives), these algorithms become invalid. The reason is that they fail to distinguish the mutual performance between the non-dominated solutions, which is called the dominance resistance (DR) phenomenon [41]. Additionally, the diversity-driven evolution causes the individuals hard to approximate the true Pareto front, which is known as the active diversity promotion (ADP) [35].

Generally, in order to overcome DR and ADP problems, there are two avenues to improve the performance of the Pareto-based MOEAs. One is to relax the Pareto dominance relation, and the other is to design more effective diversity maintenance mechanisms. To relax the Pareto dominance relation, two classes of approaches have been developed. The first class is to expand the dominance region of an individual. Based on this idea, a variety of powerful algorithms are representatives, like CDAS [43], ε-MOEA [15] and GrEA [46]. CDAS contracts or expands the dominance area of individuals by means of an angle parameter. ε-MOEA and GrEA exploit grids to increase the dominance area of individuals. The second class is to compare individuals by counting the number of better, worse or equal objectives, or to evaluate the numerical difference between their objectives, like $(1 - k)_F$-dominance [19], L-dominance [18], Fuzzy-Pareto Dominance relation [38], and fuzzy-based Pareto optimality [22]. In the Pareto-based MOEAs, besides the above two convergence approaches, the diversity maintenance mechanisms aim to combine convergence and diversity information, and reduce the adverse impact of diversity maintenance. Several methods, like shift-based diversity estimation (SDE) [35] and the diversity management operator (DMO) [1], have been developed for this purpose.

Different from Pareto-based MOEAs, non-Pareto-based MOEAs [33], such as aggregation-based and indicator-based algorithms, do not rely on the Pareto dominance relation, and thus do not face DR problem. The aggregation-based algorithms (also known as the decomposition-based algorithms) transform multiple objectives into a set of scalar subproblems by aggregation functions. Specifically, the decomposition-based MOEA (MOEA/D) [48] bounds individuals with weight vectors, and therefore optimizes the subproblems simultaneously toward different directions. Similarly, MSOPS [24], AR [20] and RVEA [42] are typical aggregation-based algorithms. It is worth noting that the diversity performance of aggregation-based algorithms mainly depends on the distribution of weight vectors. Thus, it could be less effective when solving MOPs with a discontinuous PF or different surface of manifold. The indicator-based algorithms exploit performance indicators to estimate the performance of two compared individuals or a set of individuals. The pioneer, IBEA [51], modifies the traditional performance indicators into binary performance indicators to compare the quality of two individuals. SMS-EMOA [6] and HypE [5] employ the hypervolume indicator to estimate the comprehensive performance of a solution set. SMS-EMOA works by deleting individuals that have the lowest hypervolume contribution from the population. HypE uses the Monte Carlo simulation-based hypervolume contribution estimation to solve the huge computational cost problem on MaOPs. Many researchers are attempting to reduce the computational cost of the hypervolume indicator according to the approximate estimation.

In addition to the Pareto-based and non-Pareto-based MOEAs, the third class of MOEAs is the combination of Pareto-criterion and non-Pareto-criterion, which is not uncommon in EMO. Al Moubayed et al. [3] proposed to incorporate Pareto dominance with decomposition to a developed multi-objective particle swarm optimizer. Ishibuchi et al. [25] implemented a hybrid algorithm by combining the Pareto-based NSGA-II with the weight sum fitness function in both the mating and environmental selection processes. The combination of Pareto dominance and decomposition has been proven to be very effective reported in some recent works [13,30,42], but this approach is still in its infancy. For instance, Deb and Jain [13] proposed the reference-point-based many-objective NSGA-II (NSGA-III) that selects non-dominated individuals closed to a set of given reference points. Li et al. [30] utilized the merits of both dominance- and decomposition-based approaches to balance convergence and diversity in the evolution process, which is MOEA/DD. Moreover, Li et al. [37] presented a bi-criterion evolution (BCE) framework consisting of two parts of Pareto criterion evolution and non-Pareto criterion evolution in order to make the most of their advantages as well as effectively compensate their disadvantages. Jiang et al. [29] used the reference vectors in the early-developed SPEA2, abbreviated as SPEA/R. In the past two decades, many achievements have been made

for dealing with MaOPs, but seeking better proximity and diversity of final solutions for increasingly real-world problems is not enough.

It is known that Pareto dominance has its own advantages of finding solutions with good proximity for MOPs with only two or three objectives. Nevertheless, it has some shortcomings, such as slow convergence to the Pareto front, no information of performance difference between two individuals [37] and inferior performance on many-objective problems [27,36,37]. Given the above, one question could arise: whether it is possible to inherit the advantages and compensate for the disadvantages of Pareto dominance relation on many-objective problems. In this paper, we make an attempt along this line and present a Pareto-based many-objective evolutionary algorithm using space partitioning selection and angle-based truncation (SPSAT). In SPSAT, we adopt the framework of the state-of-the-art NSGA-II, and the Pareto dominance is used as the first selection criterion to choose some layers of well-converged non-dominated individuals. In the critical layer, space partitioning selection and angle-based truncation are proposed to enhance the selection pressure and improve diversity at the same time. The former selects only one individual with the best proximity estimation in each subspace, which makes two incommensurable non-dominated individuals become comparable on the basis of moderate diversity. Moreover, the angle-based truncation is employed to further improve proximity and diversity.

The rest of this paper is organized as follows. Section 2 explains the motivation of the proposed approach. Section 3 describes the procedure of SPSAT, including space partitioning selection, proximity estimation space partitioning selection and angle-based truncation. Section 4 introduces the experimental design. Section 5 gives the experimental result and analysis of SPSAT in comparison with six classic algorithms as well as the parameter analysis in SPSAT. Finally, Section 6 draws the conclusions of the paper.

2. Motivation and related work

Over the past few years, traditional Pareto-based MOEAs have demonstrated their success in dealing with many challenging MOPs with two and three objectives, like an MOP with a large number of local optima, and with a complex Pareto front. They typically consider the selection criteria of Pareto dominance relation and density of candidate solutions [37]. Nevertheless, the insufficiency of the diversity maintenance mechanism has a negative effect on the search ability of the Pareto-based algorithms. For example, NSGA-II [16] introduces the density of individuals in the critical layer as fitness values to rank them and assigns infinite values to the boundary solutions in order to preserve them as candidate solutions. Moreover, SPEA2 [52] puts forward an archive truncation method that ensures the uniform distribution of solutions and guarantees the preservation of boundary solutions. However the preservation of boundary solutions in the two algorithms, which are probably dominance resistant solutions, may cause the degeneration of the solution set. The Pareto-based algorithms selecting non-dominated individuals in the critical layer solely based on the diversity is not effective on MaOPs, which lacks proximity pressure. One effective method to solve this issue is to introduce convergence information into the diversity maintenance mechanism. Some work has been done to improve the performance of Pareto-based algorithms by continuing the advantages of Pareto dominance relation and compensating for the drawbacks of the diversity maintenance mechanism in the non-dominated critical layer. For example, the knee point-driven evolutionary algorithm (KnEA) [49] adopts Pareto dominance as the first selection criterion, and the second selection criterion is to preserve neighborhood knee points with larger hypervolume indicator values in the critical layer. Additionally, modification of the second selection criterion makes the Pareto-based algorithms effective and efficient for coping with many-objective problems. For example, the shift-based density estimation (SDE) strategy [35] covers both convergence and diversity information of individuals in the selection process. The vector angle based evolutionary algorithm (VaEA) [47] firstly puts all extreme individuals and some best-converged individuals in the critical layer into the archive and then selects one individual with the biggest angle to the individuals in the archive at a time. This algorithm also employs the diversity of the population to strengthen the convergence of the Pareto-based MOEA. However, the work of this aspect is still little. We present two cooperation approaches, used as the second selection criterion, consisting of space partitioning selection and angle-based truncation; each approach both covers convergence and diversity information of individuals.

A general scheme of space partitioning in multi-objective optimization was introduced by Aguirre and Tanaka [2]. It partitions objective space into subspaces using a partitioning strategy to group all objectives. Three partitioning strategies: random strategy, shift strategy and fixed strategy are utilized to group objectives. Each group of objectives is called a subspace. Jaimes et al. [28] used conflicting information among objectives to group them. In conclusion, the early space partitioning methods divide the objective space into subspaces according to the groups of objectives. In recent years, MOEA/DD proposes to use a set of well-distributed weight vectors to divide the objective space into subspaces in view of geometry. RVEA and SPEA/R also use the weight vectors to partition the objective space. However, the weight vectors generation method for many-objective space is still a hard problem. Thus, this paper suggests a strategy by just using one vector and an angle to divide the objective space adaptively, which is not influenced by the number of objectives. By uniting with a proximity estimation method, the balance between the convergence and diversity can easily be achieved because the size of the subspace can be feasibly controlled according to different problems.

The archive truncation method in SPEA2 [52] truncates any number of individuals and finally obtains a well-distributed solution set, but merely using the distance information among individuals to delete individuals may be not beneficial to the convergence of the solution set. In order to combine the convergence information of individuals into the diversity main-

tenance mechanism, we propose an angle-based truncation method that employs the angle values between individuals to delete individuals. The details are given in Section 3.5.

3. Proposed SPSAT algorithm

SPSAT is an elitist Pareto-based MOEA. The Pareto dominance relation is used as the primary selection criterion. Space partitioning selection and angle-based truncation are employed as the secondary selection criterion to strengthen selection pressure and to balance proximity and diversity. The basic framework of the proposed SPSAT algorithm is presented in Algorithm 1. SPSAT starts with an initial parent population. For each iteration, SPSAT obtains a population by the mating

---

Algorithm 1 Framework of SPSAT.

Require: N (population size)
 1: Create an initial parent population P;
 2: $\overline{P} := Objective\_Normalization(P)$;
 3: **while** stopping criterion not met **do**
 4:     $P' := Mating\_Selection(\overline{P})$;
 5:     $O := Crossover(P')$;
 6:     $O := Mutation(O)$;
 7:     $Q := P \cup O$;
 8:     $\overline{Q} := Objective\_Normalization(Q)$;
 9:     $P := Environmental\_Selection(\overline{Q})$;
10: **end while**
11: **return** P;

---

selection process on the basis of the preserved parent population, followed by the crossover and mutation operator to generate an offspring population. Then, based on an union of the parent and offspring populations, the objective normalization operator is applied to the union population. Afterwards, the environmental selection process is conducted on the normalized population to construct a new parent population for the next generation. In the following subsections, the implementation of each component of SPSAT will be detailed step by step.

3.1. Objective normalization

SPSAT conducts the objective normalization to make preparations for mating selection and environmental selection (line 2 in Algorithm 1 and line 8 in Algorithm 1, respectively). The normalization procedure is described in Algorithm 2. First,

---

Algorithm 2 $Objective\_Normalization(P)$

Require: population P
 1: **for** $i := 1$ to M **do**
 2:     Compute the ideal point $z_i^* := min_{q \in P}(f_i(q))$;
 3: **end for**
 4: **for** each individual $q \in P$ **do**
 5:     Compute the normalized objectives by Eq. (1);
 6:     Save the normalized q to $\overline{P}$;
 7: **end for**
 8: **return** $\overline{P}$;

---

the ideal point $z^* = (z_1^*, \ldots, z_M^*)$, where $z_i^* = min(f_i(q))$ for any q in the population, $i = 1, \ldots, M$. Then, the objectives of individual q are normalized as follows:

$$\hat{f}_i(q) = f_i(q) - z_i^*. \tag{1}$$

Where $i \in \{1, \ldots, M\}$ and $\hat{f}_i(q)$ denotes the ith normalized objective of individual q. The function of the objective normalization has three aspects: (1) to guarantee that all the normalized objective values are located in the first quadrant (2) to give the extreme point of each corresponding normalized objective coordinate axis a big chance to survive to maintain more diversity and (3) to maximize the coverage of all solutions in the normalized objective coordinate system, therefore making the later space partitioning selection much more effective. For example, all solutions flocking together in the original coordinate system can have an more extensive coverage in the normalized coordinate system.

3.2. Mating selection

Mating selection makes use of the impactful selection strategy to pick out promising solutions for information exchange and makes preparations for subsequent crossover and mutation through filling up a mating pool. SPSAT employs the binary tournament selection strategy to select an individual once from two comparative individuals, as given in Algorithm 3. For

---

Algorithm 3 *Mating_Selection(P, N).*

Require: population P, size of mating pool N
 1: Generate an empty mating pool P';
 2: **for** $i := 1$ to N **do**
 3:     Randomly select two individuals p, q from P;
 4:     **if** p Pareto dominate q **then**
 5:         $P' := P' \cup p$;
 6:     **else if** q Pareto dominate p **then**
 7:         $P' := P' \cup q$;
 8:     **else**
 9:         **if** $f_{pr}(p) < f_{pr}(q)$ **then**
10:             $P' := P' \cup p$;
11:         **else**
12:             $P' := P' \cup q$;
13:         **end if**
14:     **end if**
15: **end for**

---

two candidates, if one individual Pareto dominates another individual, and then the dominating one should be selected; if they are non-dominated, the proposed proximity estimation f_{pr} is applied to select the individual with smaller proximity estimation f_{pr} value or one is randomly selected in case of the same f_{pr} values. The definition of f_{pr} is described in Section 3.4.2, which is the objective sum of an individual.

3.3. Environmental selection

Environmental selection is intended to pick well-converged and well-distributed solutions to form a new parent population for guiding the next generation. SPSAT implements the environmental selection given in Algorithm 4 by integrating

---

Algorithm 4 *Environmental_Selection(P, N, α).*

Require: population P, N (archive size), α (angle parameter)
 1: Generate an empty archive set A;
 2: $\{L_1, L_2, \ldots, L_i, \ldots\} := Nondominated\_Sorting(P)$;
 3: **if** $|A| < N$ **then**
 4:     $Q := Space\_Partitioning\_Selection(P, L_i, \alpha)$;
 5:     $A := A \cup Q$;
 6:     **if** $|A| < N$ **then**
 7:         $R := L_i - Q$;
 8:         /* R is remained individuals in critical layer after space partationing selection, which is used for angle-based truncation*/
 9:         $R' := Angle\_Truncation(R, |R| - (N - |A|))$;
10:         /*delete $|R| - (N - |A|)$ individuals from R by angle-based truncation, R' is remained individuals*/
11:     **end if**
12:     $A := A \cup R'$;
13: **end if**
14: **return** A;

---

the Pareto dominance relation, the proposed space partitioning selection and angle-based truncation. Here, non-dominated sorting [34] is adopted as the primary selection criterion for its efficiency in low-dimensional space. By using the Pareto dominance relation, non-dominated sorting divides the population into different layers ($L_1, L_2, \ldots, L_i, \ldots$, each layer is non-dominated). It is essential to find the critical layer L_i which satisfies $0 < N - |L_1 \cup L_2 \cup \ldots \cup L_{i-1}| \leq |L_i|$, for the reason that on MaOPs, a vast majority of individuals are non-dominated and locate in the critical layer; thus, the selection strategy used in the critical layer plays a very important role in prompting the evolution of the population. Before selection of the critical layer, the former layers $L_1, L_2, \ldots, L_{i-1}$ are put into an archive set A. If $|A|$ is smaller than the preset archive size N, and then
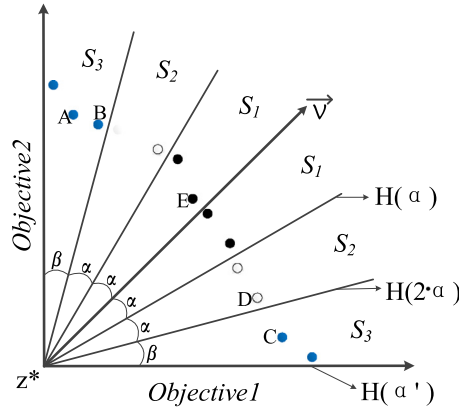
Fig. 1. Space partitioning strategy for two dimensional objective space. (Individuals depicted by black points are in the first layer of subspace, individuals painted by white points are in the second layer of subspace, individuals showed by blue points are in the last layer of subspace.). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the objective space is partitioned into different layers of subspaces by the space partitioning strategy, and an individual with the smallest proximity estimation value in each subspace is selected into A. If $|A|$ is still smaller than N, individuals are deleted from the remaining critical layer gradually through angle-based truncation and until $N - |A|$ individuals are left in the remaining critical layer set. Lastly, the rest $N - |A|$ individuals are put into the archive A.

3.4. Space partitioning selection

Space partitioning selection (denoted SPS) has two procedures including a space partitioning strategy and an individual selection process based on the former strategy.

3.4.1. Space partitioning strategy

The objective space is divided into multiple subspaces in view of geometry. To achieve this, three parameters are needed here. Firstly, a vector $\overrightarrow{v}$ whose coordinate is $(1, 1, \ldots, 1_M)$ in M-dimensional objective space is constructed. Secondly, an angle parameter α is designed to divide objective space into different layers of subspaces. Thirdly, the *sensitive upper bound* value of α is defined as α', which means when α is bigger than α', the objective space is not divided; however, α' is never bigger than $90°$. The calculation formula of α' is given in Eq. (2).

$$\alpha' = \arccos \frac{1}{\sqrt{M}}.\tag{2}$$

When $M = 2$, $\alpha' = 45°$; $M > 2$, $45° < \alpha' < 90°$.

The objective space is denoted as $Z = R^n$; N^+ represents the set of positive integers in mathematics. Using the ideal point z^*, the vector $\overrightarrow{v}$, the angles α and α', the objective space could be divided into different layers of subspaces. Each subspace S_i $(i \in N^+)$ is defined in Eq. (3).

$$S_i = \begin{cases} \{z \subseteq Z : Front((i-1) \cdot \alpha) \leq z \leq Front(i \cdot \alpha)\} & \left(1 \leq i < \frac{\alpha'}{\alpha}\right) \\ \{z \subseteq Z : Front((i-1) \cdot \alpha) < z \leq Front(\alpha')\} & \left(\frac{\alpha'}{\alpha} \leq i \leq \left\lceil \frac{\alpha'}{\alpha} \right\rceil\right) \end{cases}\tag{3}$$

Here, $Front((i-1) \cdot \alpha)$, $Front(i \cdot \alpha)$ and $Front(\alpha')$ represent the fronts whose angle with $\overrightarrow{v}$ are $(i-1) \cdot \alpha$, $i \cdot \alpha$ and α', respectively.

Starting from z^*, the first layer of subspace S_1 is the space whose angle with $\overrightarrow{v}$ is between 0 and α, and the second layer of subspace S_2 includes the space whose angle with $\overrightarrow{v}$ is between α and $2 \cdot \alpha$. Accordingly, the i-th layer of subspace S_i includes the space whose angle with $\overrightarrow{v}$ is between $(i-1) \cdot \alpha$ and $i \cdot \alpha$. The last layer of subspace is not contained by all the former layers of subspaces, but is the space surrounded by all axes and the last former layer of subspace. For example, when $M = 2$, $\alpha' = 45°$, and α is set to $15°$, the objective space is divided into three layers of subspaces S_1, S_2, S_3; when $M = 2$, $\alpha' = 45°$, and α is set to $10°$, the objective space is divided into four layers of subspaces S_1, S_2, S_3, S_4. The angles of the former three layers of subspaces are all $10°$, but the angle of the last layer of subspace is $5°$, and thus the regional size of S_4 is smaller than the former three subspaces.

Fig. 1 shows the space partitioning strategy for two dimensional objective space. The objective space is divided into three layers of subspaces, S_1, S_2, S_3, when the angle parameter $\alpha = 15°$. The last layer of subspace is the region within β. In this case, all the subspaces are the same size. Using the strategy, the individuals depicted by black points locate in the first layer of subspace; the points painted by white are included in the second layer of subspace; the blue points are contained in the last layer of subspace. In the two dimensional objective space, the region of each subspace is composed of two cones,

but this is not the case for higher dimensional objective space. The size of each layer of subspace is the same probably except for the last layer of subspace. The setting of angle parameter α affects the size of the subspaces, thus influencing the number of individuals and their distribution in each subspace. The larger α is, the more individuals can be included in every subspace in addition to the possible difference in the last layer of subspace. Meanwhile, angle α influences the size of the last layer of subspace.

A different value of angle parameter has a distinct effect on the performance of SPSAT for MOPs with various properties, and also gives a balance between the space partitioning selection and angle-based truncation. The experiments includes the sensitive of parameter α and the balance are given in Section 5.5.

3.4.2. Proximity estimation

Except for the Pareto dominance relation used for the convergence estimation in the primary selection criterion, SPSAT estimates the proximity (donated as f_{pr}) of an individual p in the critical layer by summing values of all objectives (not conducting objective normalization) according to Eq. (4):

$$f_{pr}(p) = \sum_{i=1}^{m} f^i(p). \tag{4}$$

Here, $f^i(p)$ represents the objective value of an individual p in the ith objective; m is the number of objectives. The individuals with small objective values (the minimization problems are considered in this paper) in most objectives obtain lower values in their proximity estimation.

Generally, a solution with good performance in the most objectives is likely to obtain a better (i.e., lower) f_{pr} value. However, the proximity estimation of an individual cannot be completely reflected through its f_{pr} value. Instead, accuracy of proximity estimation depends on the shape of the MOPs' Pareto front to a large extent [36]. Taking the problem with a wavy Pareto front as an example, individuals with a larger f_{pr} may be closer to the true Pareto front than the individuals having a smaller f_{pr}. To solve this issue, the objective space is divided into multiple smaller subspaces, and thus evaluating the proximity estimation of individuals separated in each subspace is more accurate and fair. Accordingly, space partitioning selection including the space partitioning strategy and proximity estimation is proposed to deal with this issue.

3.4.3. Space partitioning selection

The general procedure of the space partitioning selection is presented in Algorithm 5. The space partitioning selection

---

Algorithm 5 *Space_Partitioning_Selection*(P, L, α).

Require: population P, critical layer L, angle α
 1: Generate an empty Q;
 2: Divide the objective space into multiple subspaces by the *space partitioning strategy*;
 3: Allocate individuals in P to different subspaces;
 4: **for** each individual $p \in L$ **do**
 5:    Compute $f_{pr}(p)$ by Eq. (4);
 6: **end for**
 7: **for** each subspace in objective space **do**
 8:    select only one individual q with the smallest proximity estimation value;
 9:    $Q := Q \cup q$;
10: **end for**
11: **return** Q;

---

firstly divides the objective space into different layers of subspace by the space partitioning strategy. Then, in each subspace, only one individual with the smallest proximity estimation value f_{pr} is selected into an archive. The angle α decides the number of divided subspaces, thus ensuring the number of individuals to be selected.

Conversely, without the space partitioning strategy, three individuals are selected through the proximity estimation in the non-divided objective space. For instance, in Fig. 1, three individuals, A, B and C, are selected by the proximity estimation because of their good convergence. Whereas A and B with two smaller proximity estimation values are very close to each other. In this case, selecting individuals with good proximity and bad diversity in environmental selection can leave the population unable to explore wider objective space, and therefore the population could easily fall into local optimum. Thus, choosing to lose a small part of convergence is a feasible solution for obtaining more extensive distribution.

The space partitioning selection can overcome this problem. In Fig. 1, three individuals B, E and D, locate in different layers of subspaces respectively, are selected according to the strategy. Compared to the individuals A, B and C, the individuals B, E and D have better diversity and similar proximity. Adjusting the angle parameter α makes it possible to balance the convergence and diversity, although space partitioning selection cannot ensure the selected individuals both the best proximity and diversity. Meanwhile, the accuracy of proximity estimation would not be easily influenced by the shape of the MOPs' Pareto fronts.
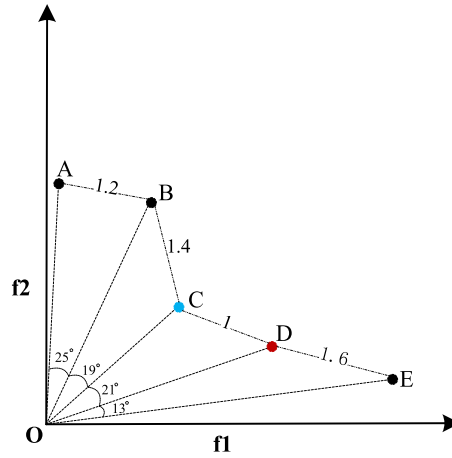
Fig. 2. An individual truncation example of the angle-based truncation and the Euclidean distance-based truncation.

3.5. Angle-based truncation

Given individuals p and q, the diversity estimation of p can be defined by Eq. (5):

$$angle_{p \to q} = arccos \frac{\overrightarrow{p} \; \overrightarrow{q}}{|\overrightarrow{p}||\overrightarrow{q}|}. \tag{5}$$

where $angle_{p \to q}$ is the angle between p and q, which stands for the diversity estimation of both p and q with regard to each other. $\overrightarrow{p}$ and $\overrightarrow{q}$ are the objective vectors of p and q, respectively. When removing an individual from the population, the angle-based truncation (denoted AT) has the following four steps:

- Calculate the angle of any two individuals in the population.
- Find the minimum angle of all the individuals, and according to this minimum angle, mark its corresponding two individuals as a, b.
- Seek the minimum angles of the individuals a, b to all the other individuals in the population respectively, which are marked as $angle_a$ and $angle_b$, respectively.
- Compare $angle_a$ and $angle_b$. If $angle_a > angle_b$, individual b should be removed, and if $angle_a < angle_b$, individual a is supposed to be removed, and else randomly remove an individual from the individuals a and b.

To reduce a population with n individuals to m individuals ($n > m$), the angle-based truncation should carry out the previous four steps iteratively. The angle-based truncation algorithm is given in Algorithm 6.

**Algorithm 6** *Angle_Truncation*(P, m).

Require: population P (size: n), m (number of individuals after truncation)
 1: Compute the angle $angle_{p \to q}$ of any two individuals p, q in the population P by Eq. (5) to get *AngleMatrix*;
 2: **while** $|P| > m$ **do**
 3:     Pair$<a$, $b>$:= FindMinAngle(*AngleMatrix*);
 4:     $angle_a$:= FindSecondMinAngle(*AngleMatrix*, a);
 5:     $angle_b$:= FindSecondMinAngle(*AngleMatrix*, b);
 6:     **if** $angle_a < angle_b$ **then**
 7:         Delete(a);
 8:     **else**
 9:         Delete(b);
10:     **end if**
11: **end while**

In order to further verify the angle-based truncation can largely enhance the performance of the traditional multi-objective optimization algorithm, the diversity maintenance mechanism in SPEA2 is compared with the angle-based truncation. In fact, the truncation processes of these two diversity mechanisms are quite similar. The diversity maintenance mechanism in SPEA2 is an Euclidean distance-based truncation, and it uses the Euclidean distance as the truncation assessment of individuals, which has been proven to be very effective in maintaining diversity of the population. Fig. 2 gives an individual truncation example of the angle-based truncation and the Euclidean distance-based truncation. There are five

(a) proximity comparison

(b) diversity comparison
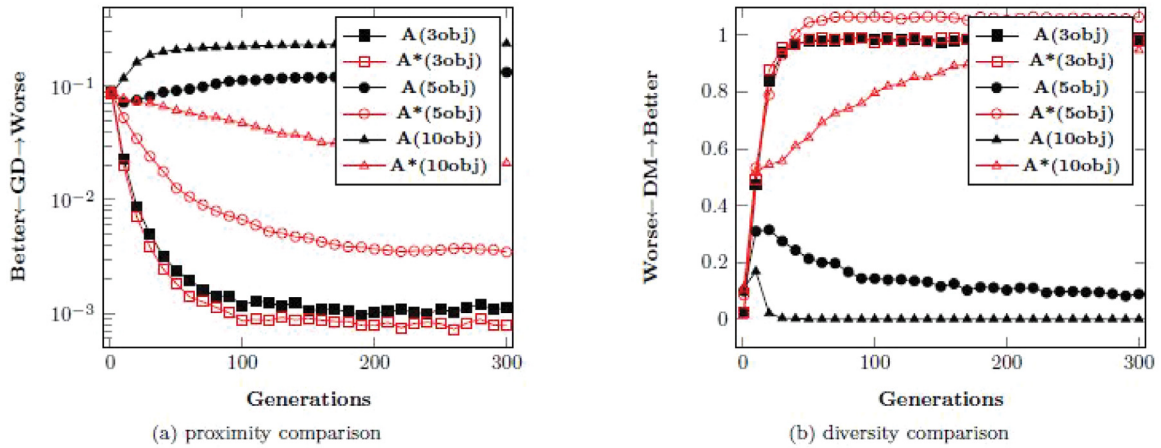
Fig. 3. The comparison trajectories of GD and DM results between the original SPEA2 (marked as A) and the modified version of SPEA2 (marked as A*) in which the diversity mechanism is replaced by angle-based truncation.

nondominated individuals in the figure, and two individuals have to be deleted from the population. Using the angle-based truncation strategy, individuals B and D are deleted. Using the Euclidean distance-based truncation strategy, individuals B and C are truncated from the population. To some extent, the convergence of individual C is better than that of individual D, however, C is deleted by using the Euclidean truncation. Therefore, we can see that an individual (like D) with a quite small angle to another individual may have worse convergence than an individual (like C) with a quite small Euclidean distance to another individual, and this phenomenon is quite common in the evolution process, while using the angle-based truncation strategy can reserve this kind of individuals like C, thus enhancing the convergence of the algorithm largely.

Because of the similarity of Euclidean distance-based truncation process and angle-based truncation process, the influences of these two methods on both convergence and diversity of the obtained population are further researched. Fig. 3 gives the comparison trajectories of convergence and distribution results between the original SPEA2 and its modified version in which the diversity maintenance mechanism is replaced by angle-based truncation on the 3-, 5- and 10-objective DTLZ2. The populations obtained by these two mechanisms are evaluated by a proximity indicator of generational distance (GD) [45] which calculates the average Euclidean distance from the solution set to the true Pareto front and a diversity indicator of diversity metric (DM) [4] that reckons uniform and extensive degree of distribution of the solution set.

As can be seen in Fig. 3(a), the interval between the GD trajectories of the two algorithms becomes more obvious with an increase in the number of objectives. For the 3-objective problem, the trajectories of the two algorithms are very close, but the modified SPEA2 with angle-based truncation is closer to the true Pareto front. For the 5-objective and 10-objective problems, the modified SPEA2 achieves better GD results than the original SPEA2 during the evolution process.

Specific reasons could be analyzed. The Euclidean distance-based truncation in SPEA2 is likely to reserve individuals with good diversity but may be far away from the Pareto front. These individuals have an unfavorable impact on the proximity of the algorithm. Moreover, in dealing with many-objective problems, the diversity maintenance mechanism sometimes plays a role in making the population gradually evolve away from the PF during the environmental selection. In view of the angle-based truncation, however, these individuals tend to have smaller angles compared with other individuals. Thus, they could easily be removed in the environmental selection precess.

According to Fig. 3(b), similar results with Fig. 3(a) can be drawn. With the increase of the number of objectives, the interval between the DM trajectories of the two algorithms becomes more evident. For the 3-objective problem, the two algorithms have coincidence of their trajectories. For the 5-objective problem, the modified SPEA2 acquires much better DM results than the original SPEA2. Furthermore, the modified SPEA2 obtains better DM results in the 5-objective problem than in the 3-objective problem. For the 10-objective problem, the modified SPEA2 is still better than the original SPEA2 in diversity.

From the observation of Fig. 3, the angle-based truncation is not only an effective diversity maintenance mechanism, but it also makes contributions to convergence of the algorithm at the same time.

The way of combining the space partitioning selection and the angle-based truncation is quite essential to enhance convergence and maintain diversity of the Pareto-based MOEAs on the MaOPs. When truncating individuals, this paper just truncates the remained nondominated solutions that apart from the individuals selected by SPS without treating them as a whole. This way has some reasons and advantages. As shown in Fig. 4, it gives two cases of truncating the remained solutions without considering the already selected individuals by SPS and taking the already selected individuals by SPS into consideration, respectively. There are ten individuals in the critical layer, and when the parameter $\alpha = 22.5°$, the objective space is divided into two subspaces, the two individuals c and e with smaller proximity estimation are selected into the archive by SPS, and then two individuals have to be deleted from the non-dominated solution set. If considering the eight
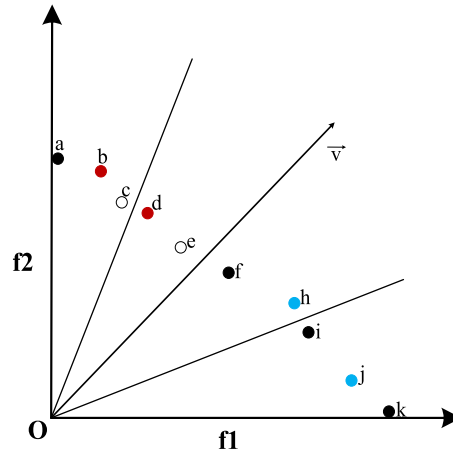
Fig. 4. Two cases of truncating the remained solutions without considering the already selected individuals by SPS and taking the already selected individuals by SPS into consideration, respectively.

remained individuals and the two already selected individuals, the angles among the ten individuals are calculated, and two individuals corresponding to the smallest angle are c and d, and then d is firstly deleted because c is already in the archive, and then b is deleted using the same method. The remained individuals a, f, h, i, j and k are put into the archive, and the total eight selected individuals are marked as set A (including a, c, e, f, h, i, j and k). However, if the already selected two individuals are not considered, the angle among the remained eight individuals are calculated, then two individuals corresponding to the smallest angle are h and i, thus h is deleted firstly, and the individual j is deleted secondly. Then the remained individuals a, b, d, f, i and k are put into the archive, and the total eight selected individuals are marked as set B (including a, b, c, d, e, f, i and k). Normally, the individuals selected by SPS have good convergence and moderate diversity, applying AT solely to the remained solutions in this paper can preserve more well-converged individuals that may near the individuals selected by SPS, and thus can further enhance the local search ability. Meanwhile, it preserves the boundary individuals that ensure the extensity of solution set. Moreover, AT also ensure the remained individuals good uniformity. As can be seen from the figure, set A has better diversity than set B, but set B also remains diverse individuals like f, i and k, and reserves more individuals near the well-converged regions that can enhance the convergence in the next evolution iteration.

Different from the recent proposed MOEAs, like NSGA-III and VaEA, the former algorithm entirely reserves the individuals with the smallest distances to their associated reference points and the latter algorithm always selects an individual with the largest angle to the individuals in the archive, and they both depend largely on the diversity of the population to guide the convergence. This paper emphasizes the importance of convergence on the basis of diversity. When deleting individuals from the nondominated set, the work does not assess the diversity of the remained solutions and the already selected individuals in the archive, conversely, by subtly combine SPS and AT, it strengthens the convergence and maintains diversity. Additionally, the balance between convergence and diversity can be achieved by adjusting the parameter α according to distinct problems.

3.6. Discussion

After introducing the technical details of SPSAT, this section discusses the similarities and some differences of SPSAT compared with MOEA/DD, RVEA and VaEA.

1. *Similarities between SPSAT and MOEA/DD:*

   Both of them divide the objective space into subspaces.
   Both of them employ Pareto dominance relation to sort individuals.

2. *Similarities between SPSAT and RVEA:*

   Both of them divide the objective space into subspaces.
   Both of them use the angle between two vectors.

3. *Similarities between SPSAT and VaEA:*

   Both of them employ the Pareto dominance relation to sort individuals.
   Both of them use the angle between two individuals to select individuals.
   Both of them use the formula of the sum of all objectives for the individuals.

4. *Differences between SPSAT and MOEA/DD:*

Table 1
Time Complexity of peer MOEAs.

Algorithm	Worst time complexity
SPSAT	$O(MN^2)$
MOEA/D	$O(MNT)$ [48]
NSGA-III	$max\{O(Nlog^{M-2}N), O(MN^2)\}$ [13]
MOEA/DD	$O(MN^2)$ [30,31]
RVEA	$O(MN^2)$ [42]
VaEA	$max\{O(Nlog^{M-2}N), O(MN^2)\}$ [47]
Two_Arch2	$max\{O(Nlog^{M-2}N), O(MN^2)\}$ [21]

SPSAT does not employ the approach of decomposition that MOEA/DD uses.

The methods of space division are different. MOEA/DD applies a set of weight vectors to divide the objective space, while SPSAT uses only one vector and an angle to divide the objective space.

The parameter α in SPSAT can be adjusted and control the size of subspace in order to adapt to different problems, but the size of subspace in MOEA/DD always remains unchanged.

5. *Differences between SPSAT and RVEA:*

The approaches of space division are different. RVEA uses the weight vectors to assist dividing the objective space; SPSAT employs a vector and an angle to partition the objective space.

The application methods of the angle are distinct. RVEA calculates the angle between an individual and the weight vectors to decide which subpopulation the individual belongs to, and meanwhile it needs to preset an reference parameter. Additionally, the angle is used to construct the Angle-Penalized Distance. SPSAT computes angles between any two individuals to assess the diversity of population and uses the angles to truncate population, and it does not require parameter.

6. *Differences between SPSAT and VaEA:*

In SPSAT, the individuals in each suspace with the best objective sum values are selected into the archive. In VaEA, the M individuals in the critical layer with the best objective sum values are directly put into the archive.

SPSAT uses angles between any two individuals to estimate the diversity and truncate population. In VaEA, the angle is used to associate non-selected individuals with individuals in the archive, and an individual with the largest angle is put into the archive. Meanwhile, an individual in the archive with a very small angle with its associated better-converged non-selected individual is replaced by the latter individual.

3.7. Empirical computational complexity analysis

In this section, we provide an upper bound of the runtime of SPSAT. Within one generation, SPSAT mainly performs the following three operations: (1) mating selection; (2) variations; (3) environmental selection. For a population size N and an optimization problem with M objectives, the mating selection needs a runtime of $O(NM)$ to form a mating pool of size N, as the calculation of the proximity estimation is $O(M)$ for each individual. Variations, here simulated binary crossover (SBX) [12] and polynomial mutation [12], are applied to each decision variable of the parent solutions, which needs a runtime of $O(DN)$ to generate N offspring, where D is the dimension of decision variables. The environmental selection consists of three parts: non-dominated sorting, space partitioning selection and angle-based truncation. Non-dominated sorting needs a runtime of $O(MN^2)$ for the combined population of size $2N$ in the worst case. Space partitioning selection has a computation cost of $O(N)$, as it includes a runtime cost of $O(N)$ of the space partitioning strategy; a runtime of $O(N)$ of the proximity estimation. The angle-based truncation needs to calculate the angle of any pair of solutions, so the computation cost is $O(N^2)$. Overall, the runtime cost of environmental selection is $O(MN^2)$. Therefore, SPSAT takes the runtime of a generation of $O(MN^2)$. The worst time complexity of the tested peer MOEAs in the paper is shown in Table 1. T in MOEA/D is the number of the weight vectors in the neighborhood of each weight vector.

4. Experimental design

This section provides the experimental design for the purpose of investigating the performance of SPSAT in many-objective optimization. Firstly, it recommends three well-defined sets of test problems employed in the experiment. Secondly, two widely used performance indicators are presented. Then, we briefly introduce six state-of-the-art EMO algorithms including MOEA/D [48], NSGA-III [13], MOEA/DD [30], RVEA [42], VaEA [47] and Two_Arch2 [21], which are used to verify the validity of the proposed algorithm. Finally, the general experimental settings are described for the evolutionary process.

4.1. Test problems

Many efforts for designing test suites and toolkits have been made for inspecting EMO algorithms [8,23]. For a more comprehensive study of the proposed algorithm compared with six state-of-the-art algorithms, fifteen problems from three

Table 2
Parameter setting of α in SPSAT for all test problems.

Problem	Number of objectives (M)	Number of variables	Properties	angle in SPSAT			
				$M=5$	$M=10$	$M=15$	$M=20$
DTLZ1	5,10,15,20	$M-1+k(k=5)$	Linear, multimodal	22	2	2	6
DTLZ2	5,10,15,20	$M-1+k(k=10)$	Concave, unimodal	18	1	2	4
DTLZ3	5,10,15,20	$M-1+k(k=10)$	Concave, multimodal	33	4	3	3
DTLZ4	5,10,15,20	$M-1+k(k=10)$	Concave, biased	35	3	2	2
DTLZ5	5,10,15,20	$M-1+k(k=10)$	Concave	1	1	1	1
DTLZ6	5,10,15,20	$M-1+k(k=10)$	Concave, biased	1	1	1	1
DTLZ7	5,10,15,20	$M-1+k(k=20)$	Mixed, multimodal, disconnected	2	1	1	1
WFG1	5,10,15,20	$2\times(M-1)+20$	Convex, mixed, biased	6	5	5	7
WFG2	5,10,15,20	$2\times(M-1)+20$	Convex, nonseparable, disconnected	15	8	7	10
WFG3	5,10,15,20	$2\times(M-1)+20$	Linear, nonseparable	1	1	1	1
WFG8	5,10,15,20	$2\times(M-1)+20$	Concave, nonseparable, baised	9	5	3	4
WFG9	5,10,15,20	$2\times(M-1)+20$	Concave, nonseparable, deceptive, baised	5	7	5	2
MaF1	5,10,15,20	$M-1+k(k=10)$	Linear, unimodal	34	3	6	9
MaF2	5,10,15,20	$M-1+k(k=10)$	Concave, unimodal	7	10	6	13
MaF3	5,10,15,20	$M-1+k(k=10)$	Convex, multimodal	12	3	3	2

famous benchmark suites, DTLZ test suite [17], WFG toolkit [23] and MaF test problems [8], under four scenarios when the number of objectives $m = 5, 10, 15$ and 20, are considered in the experiments. The properties of all the test problems are summarized in Table 2.

DTLZ is a continuous problem suite, which can be scaled to any number of objectives. Moreover, it encompasses various problem properties of linear, concave, unimodal, multimodal, bias, degeneration and disconnection. The different properties are used for investigating diverse capabilities of underlying MOEAs. Nine problems are contained in the DTLZ test suite. Here, the first seven problems, DTLZ1 to DTLZ7, are chosen in the experiments, because the last two problems have side constraints which need special treatment in MaOPs. As suggested in [23], the parameter k in DTLZ is set to different values according to distinct problems.

Another popular continuous test suite WFG is scalable to any number of objectives and decision variables. It includes more varying problems consisting of mixed, convex, nonseparable and deceptive characteristics that are not contained in the DTLZ problem suite. According to the suggestion by Huband et al. [23], the parameters k and l are set to $2 \times (m-1)$ and 20, respectively. Here, WFG1-3, WFG8 and WFG9 are selected for the experiments, because the properties of WFG4-7 are included in the tested problems in this paper.

MaF test suite is also a continuous test suite that contains 15 problems selected or modified from existing test problems. MaF has more distinct characteristics compared with the former two problem suites, which includes more complex Pareto fronts, such as conversed and badly scaled Pareto fronts. While the shapes of Pareto front can strongly influence the performance of decomposition-based many-objective algorithms [26]. Thus, it is really worth to employ MaF to validate the seven MaOEAs. MaF1-3 are chosen for the experiments.

4.2. Performance indicators

In this experiment, two widely used performance indicators are chosen to qualify the performance of MOEAs. Inverted generational distance (IGD) [7] and hypervolume (HV) [53] are considered since they both measure the convergence and the diversity at the same time.

IGD works by calculating the averaged Euclidean distance from the true Pareto front to the optimal solution set. The smaller the IGD values, the better the comprehensive performance. For calculation of IGD, a set of reference points should be generated for every specific benchmark problem by using the point generation method in [50]. In this method, 100,000 points are randomly produced firstly according to the specific problem (the number of the distance variable in the decision variable is set to 0, and thus all initial points are already on the Pareto front), and then these points are optimized, and lastly these points are reduced to certain number of well-distributed points by means of the truncation approach in SPEA2 [52]. Thus, a more accurate IGD value can be obtained from the reference points with relatively convergence and uniform distribution in the high-dimensional space. This paper adopts IGD to evaluate the solution set obtained by the MOEAs on the DTLZ problem suite.

HV is a very popular performance indicator because it is suitable for the problems without requiring knowledge of the true Pareto front. It calculates the volume of the objective space encompassed by the optimal solution set and the reference points. For calculation of HV, two issues of the reference point setting and estimation speed are taken into consideration. For WFG test problems, the reference points are all set to $2 \cdot i + 1$ (i is the objective ordinal number). For tested MaF problems, the reference points are set to $2 \cdot u_i$ (u_i is the upper bound of the ith objective). With regard to the calculation speed, since the accuracy calculation of the volume costs enormous time with multiple objectives, the Monte Carlo sampling method in [5] is applied to approximately estimate the HV and thus the time consumption was greatly reduced. Following the suggestion in [5], the number of the sampling points was set to 10,000,000 to ensure accuracy.

Table 3

Settings of population size in MOEA/D, NSGA-III, MOEA/DD and RVEA. H_1 and H_2 are the simplex-lattice design factors for generating uniformly distributed weight vectors on the outer boundaries and the inside layers, respectively.

M	(H_1, H_2)	Population size
5	(5, 0)	126
10	(3, 2)	275
15	(2, 1)	135
20	(2, 1)	230

4.3. Selected MOEAs for comparison

The proposed algorithm is compared with the six state-of-the-art EMO algorithms, including MOEA/D [48], NSGA-III [13], MOEA/DD [30], RVEA [42], VaEA [47] and Two_Arch2 [21]. The general working principles of the six algorithms are presented in the following paragraphs. The experiments are conducted on the PlatEMO experimental platform, where the source code of the six algorithms can be found [8].

1. *MOEA/D*: it is a decomposition-based multi-objective evolutionary algorithm. It decomposes multi-objective problems into single objective problems by multiple preseted weight vectors. Through associating with weight vectors, individuals optimize simultaneously toward different directions. Three aggregation functions, weight sum approach, Tchebycheff approach and penalty-based boundary intersection (PBI) approach are proposed in the original paper. Here, PBI is used in MOEA/D in the experiment.
2. *NSGA-III*: it is a many-objective optimizer developed under the framework of NSGA-II. The weight vectors are employed to maintain diversity by selecting non-dominated individuals closed to them. NSGA-III uses the diversity of population to guide the search in high-dimensional space.
3. *MOEA/DD*: it is a combined many-objective evolutionary algorithm exploiting the advantages of decomposition-based and dominance-based approaches to balance the convergence and diversity of the evolution process. A widely spread weight vectors generation method is proposed, and each weight vector not only stands for a subproblem, but also defines a subregion in the objective space.
4. *RVEA*: it is a reference vectors guided many-objective evolutionary algorithm. The reference vectors not only define the subproblems, but also reflect the users' preference. A scalarization function, the angle-penelized distance, is used to balance the convergence and diversity. Besides, the distribution and the number of the reference vectors can be dynamically adjusted to adapted to different problems.
5. *VaEA*: it is a many-objective evolutionary algorithm using the vector angles, and it does not require any algorithm parameter in the evoltion process. A maximum-vector-angle-first principle is presented to ensure wideness and uniformity of the distribution, and a worse-elimination principle is used to replace the individuals with worse convergence.
6. *Two_Arch2*: it is a hybrid MOEA based on the indicator and the Pareto dominance, which are employed in the convergence archive (CA) and diversity archive (DA), respectively. In particular, $I_{\epsilon+}$ is adopted as the selection criterion in CA, and $L_{1/m}$-norm-based diversity maintenance is used in DA. The interaction between CA and DA prompts the convergence of DA, which is the final output solution set.

4.4. Parameter settings in experiment

The parameter settings for all conducted experiments are as follows.

1. *Number of runs and stopping criterion*: we execute 30 independent runs for each algorithm in every test instance. Since the difficulty of problems is different, the stopping criteria are also distinct. For DTLZ1, DTLZ3, DTLZ6, WFG1, MaF1 and MaF3, the maximum numbers of iterations are all set to 1000, but for DTLZ2, DTLZ4, DTLZ5, DTLZ7, WFG2, WFG3, WFG8, WFG9 and MaF2, the stopping criteria are all 300 iterations.
2. *Population size and weight vectors*: for MOEA/D, NSGA-III, MOEA/DD and RVEA, the population size is determined by the number of weight vectors. As the recommendation in [30], for the problems with more than 7 objectives, a two-layer vector generation strategy is employed to generate weight vectors. The detailed settings of the population sizes of the four algorithms are give in Table 3 with respect to different objective numbers. For VaEA, Two_Arch2 and SPSAT, the population sizes are set according to Table 3.
3. *Crossover and mutation*: for all problems, simulated binary crossover (SBX) [12] and polynomial mutation (PM) [14] are adopted to generate offspring. Their distribution index is set to 20, as recommended in [11]. The crossover probability $p_c = 1.0$ and the mutation probability $p_m = 1/n$ (where n denotes the number of decision variables) are applied.

4. *Other parameters*: parameters need to be set in some algorithms. In MOEA/D and MOEA/DD, the neighborhood size is 20, and the penalty parameter is set to 5. Particularly, in MOEA/DD, probability used to select in the neighborhood: $\sigma = 0.9$, as recommended in [30]. For RVEA, the index $\alpha = 2$ and the frequancy $f_r = 0.1$, as refered to [42]. The parameter setting of α in SPSAT for all test problems is given in Table 2.

5. Experimental results

In this section, we verify the performance of SPSAT according to the experimental design in the previous section. Table 4 gives the IGD results of the DTLZ test suite; Tables 5 and 6 provide the HV results of the WFG test suite and MaF problems, respectively. In each table, the best and the second best results for each instance are highlighted by dark gray and light gray background, respectively, and the data in each cell are mean and standard deviation (included in the bracket). Moreover, a non-parameter statistical hypothesis test, the Mann-Whitney-Wilcoxon rank-sum test, was employed to investigate whether significant differences exist at a 0.05 significance level between the results of SPSAT and its competitors. In the aforementioned three tables, symbol "†" indicates that the results of the five peer algorithms are significantly different from that of SPSAT at a level of 0.05 by the Mann-Whitney-Wilcoxon rank-sum test. Additionally, a sensitive study of parameter α in SPSAT is discussed at the end of this section.

5.1. DTLZ problem suite

Table 4 gives the comparative results of the seven algorithms on the DTLZ test suite with 5, 10, 15 and 20 objectives. As can be seen from the table, SPSAT generally defeats the six algorithms with obvious superiority. Specially, SPSAT achieves the best and the second best IGD values on 16 and 6 out of the 28 instances, respectively. SPSAT has the best and the second IGD values on all test instances of DTLZ2, DTLZ3, DTLZ5, DTLZ6 and DTLZ7. Additionally, it performs the best on DTLZ4 with 10 and 15 objectives. MOEA/D with 6 best and 6 second best IGD values and MOEA/DD with 4 best and 7 second best IGD values perform similarly and outperform the other four algorithms. They both have their own strong points. MOEA/D works better on DTLZ5 and DTLZ6, but MOEA/DD has better performance on DTLZ3 and DTLZ4. They both have the best and the second IGD values on DTLZ1 and DTLZ2, and MOEA/D performs the best on the 10- and 15-dimensional DTLZ1; MOEA/DD performs the best on the 20-dimensional DTLZ1 and DTLZ2. RVEA performs the best on the 20-dimensional DTLZ4, and the second on the 5-dimensional DTLZ1 and DTLZ4 and on the 10-, 15- and 20-dimensional DTLZ7. The dynamic adjustment strategy of the reference vectors in RVEA makes the reference vector based algorithms more competitive in solving problems with discontinuous Pareto fronts. Similarly, Two_Arch2 works well on some instances of DTLZ1, DTLZ4 and DTLZ7. Specifically, it works the best on DTLZ1 with 5 objectives, DTLZ4 with 10 and 15 objectives, and DTLZ7 with 5 objectives. Obviously, NSGA-III and Two_Arch2 do not converge on DTLZ3 with 10, 15 and 20 objectives, and VaEA performs badly on DTLZ1 and DTLZ3 because this algorithm preserves the extreme individuals. The three algorithms can not approximate the Pareto front of the high-dimensional DTLZ6. Apparently, all the seven algorithms can not approximate closely to the Pareto front of DTLZ7 with 10,15 and 10 objectives, except for SPSAT on the 10-dimensional DTLZ7.

In order to visualize the final solutions of the seven MOEAs, Fig. 5 plots the final solutions of one run according to the 10-objective DTLZ4 by parallel coordinates [39]. Specially, the paper Li et al. [39] systematically explains how to read many-objective solution sets in the parallel coordinates, and it also introduces that the quality of the solution sets in the parallel coordinates can be reflected through four measures: convergence, coverage, uniformity and extensity. Here, we analyze the performance of the seven algorithms from the four measures. As can be seen from the parallel coordinates straightforwardly, the extensity of the seven algorithms is good, with their objective values ranging from 0 to 1. MOEA/D, RVEA, MOEA/DD and SPSAT converge well on the 10-dimensional DTLZ4; a few individuals of NSGA-III can not converge, but they have very small distances to the Pareto front; there are more individuals of VaEA do not converge compared with NSGA-III; very few individuals of Two_Arch2 can not reach the Pareto front, but they are all quite away from the Pareto front. The coverage of MOEA/D is not quite good, since it fails to cover the 5, 8, and 9 objectives. NSGA-III, RVEA and MOEA/DD fail to cover the region between 0.2 to 0.4 on all 10 objectives. However, VaEA, Two_Arch2 and SPSAT cover well on all 10 objectives. From the explanation in [39], the uniform distribution in the parallel coordinates always imply the uniform solution sets, and a bad uniformity in the parallel coordinates does not mean bad-distributed solution sets. Thus, we can see that the uniformity of MOEA/D, NAGS-III, RVEA and MOEA/DD is good; VaEA, Two_Arch2 and SPSAT may have better uniformity than the former four algorithms.

Fig. 6 gives the final solution sets of the seven MOEAs on the 5-dimensional DTLZ7 in order to visualize their distribution on the disconnected problem. For the 5-dimensional DTLZ7, the boundaries of 1 to 4 dimensions are the same closed regions [0,1], and the boundary of the 5th dimension is the closed region [3.49,10] according to the formula of DTLZ7. From the figure, we can see that the six algorithms converge to the Pareto front except for Two_Arch2 where only a few individuals do not converge. SPSAT, VaEA, Two_Arch2 and RVEA have good extensity with individuals on the boundaries. The solution set of NSGA-III fails to cover the lower boundary below 4 of the last dimension and the last dimension of MOEA/D does not extend to the region 8 to 10. Besides, all solutions of MOEA/DD converge to a point. The coverage of SPSAT is the best with its solutions covering the whole Pareto front. The coverage of NSGA-III is slightly inferior to that of SPSAT. VaEA, Two_Arch2, MOEA/D and RVEA have not solutions covering some regions of the Pareto front. Finally, the uniformity can not be exactly observed from the parallel coordinates of DTLZ7 with disconnected irregular Pareto fronts.

Table 4
IGD comparison of the six EMO algorithms on DTLZ problem suite.

Problem	Obj.	MOEA/D	NSGA-III	RVEA	MOEA/DD	VaEA	Two_Arch2	SPSAT
DTLZ1	5	6.3365e-2 (5.19e-5)	6.3383e-2 (7.89e-5)	6.3331e-2 (5.76e-5)	6.3361e-2 (6.45e-5)	1.0564e-1 (2.94e-2)†	6.1012e-2 (4.33e-4)†	6.3389e-2 (1.00e-3)
	10	1.0613e-1 (3.79e-4)†	1.8892e-1 (1.02e-1)†	1.0855e-1 (2.07e-3)†	1.0677e-1 (3.03e-4)†	3.1022e-1 (1.44e-1)†	1.1049e-1 (1.93e-3)	1.1477e-1 (1.82e-3)
	15	1.4304e-1 (2.80e-3)†	2.6192e-1 (9.96e-2)†	1.7396e-1 (4.98e-3)	1.5025e-1 (7.43e-3)	2.9073e-1 (9.19e-2)†	1.4677e-1 (4.70e-3)	1.5906e-1 (5.88e-3)
	20	1.6696e-1 (2.09e-3)	2.6042e-1 (7.82e-2)†	1.7295e-1 (6.40e-3)	1.6557e-1 (4.61e-3)	4.8622e-1 (2.59e-1)†	2.1066e-1 (4.44e-3)†	1.7614e-1 (1.65e-3)
DTLZ2	5	1.9499e-1 (4.32e-5)†	1.9514e-1 (1.24e-4)†	1.9498e-1 (6.03e-5)†	1.9496e-1 (5.46e-5)†	1.9572e-1 (1.44e-3)†	1.9566e-1 (1.77e-3)†	1.7040e-1 (8.26e-3)
	10	4.1727e-1 (2.35e-3)	4.9470e-1 (7.52e-2)†	4.2442e-1 (1.79e-3)†	4.2292e-1 (1.72e-3)†	5.1311e-1 (1.07e-2)†	4.4247e-1 (8.56e-3)†	4.0594e-1 (2.33e-3)
	15	6.2788e-1 (2.18e-2)†	7.5778e-1 (5.07e-2)†	6.2578e-1 (2.29e-3)†	6.1761e-1 (3.51e-3)†	6.4800e-1 (1.36e-2)†	6.7158e-1 (2.71e-2)†	5.7951e-1 (5.01e-3)
	20	7.0205e-1 (4.31e-2)†	7.9583e-1 (6.01e-2)†	6.5565e-1 (5.33e-2)	6.2734e-1 (1.54e-2)	8.0232e-1 (3.01e-2)†	8.6588e-1 (3.98e-2)†	6.4156e-1 (4.23e-3)
DTLZ3	5	1.9605e-1 (1.40e-3)†	2.8325e-1 (2.75e-1)†	1.9551e-1 (8.51e-4)	1.9411e-1 (3.79e-4)	3.5270e-1 (8.23e-2)†	2.0054e-1 (8.25e-4)†	1.7147e-1 (9.83e-3)
	10	6.6271e-1 (3.30e-1)†	5.2969e+0 (4.89e+0)†	4.7231e-1 (4.51e-2)†	4.2548e-1 (3.79e-3)	2.2284e+1 (1.12e+1)†	1.2038e+0 (3.97e-1)†	4.0849e-1 (2.37e-3)
	15	1.0579e+0 (2.99e-1)†	6.1426e+1 (3.24e+1)†	6.3243e-1 (6.09e-3)†	6.2621e-1 (1.67e-3)†	1.4776e+1 (7.74e+0)†	1.4010e+1 (6.89e+0)†	5.9472e-1 (5.74e-3)
	20	9.7417e-1 (3.39e-1)†	5.6958e+1 (2.74e+1)†	1.1442e+0 (1.63e+0)†	6.2549e-1 (1.02e-2)	1.8942e+1 (8.87e+0)†	1.5049e+1 (7.85e+0)†	6.4356e-1 (4.37e-3)
DTLZ4	5	5.7714e-1 (2.12e-1)†	2.4249e-1 (9.97e-2)	1.9495e-1 (6.44e-5)	1.9494e-1 (7.30e-5)	1.9680e-1 (1.74e-3)	1.9953e-1 (1.58e-3)	2.2433e-1 (1.31e-1)
	10	6.9960e-1 (9.26e-2)†	4.5977e-1 (3.01e-3)†	4.4240e-1 (1.62e-3)	4.3238e-1 (4.35e-3)	5.3227e-1 (2.70e-2)†	4.2771e-1 (7.10e-3)	4.2211e-1 (2.25e-3)
	15	8.7280e-1 (7.56e-2)†	8.8964e-1 (4.75e-2)†	6.3238e-1 (6.77e-3)†	6.3005e-1 (5.77e-3)†	6.6235e-1 (2.11e-2)†	6.1963e-1 (3.88e-3)	6.0600e-1 (3.26e-3)
	20	8.6573e-1 (6.13e-2)†	7.1208e-1 (6.41e-2)†	6.2294e-1 (2.27e-4)†	6.2432e-1 (5.22e-3)	8.1239e-1 (5.00e-2)†	7.1843e-1 (1.57e-2)†	6.5808e-1 (3.20e-3)
DTLZ5	5	3.2394e-2 (2.79e-4)†	3.2738e-1 (1.19e-1)†	1.7865e-1 (2.40e-2)†	1.0635e-1 (7.78e-3)†	1.5034e-1 (4.81e-2)†	8.3327e-2 (2.62e-2)†	1.6149e-2 (1.78e-3)
	10	1.8702e-2 (8.12e-5)†	3.7518e-1 (1.16e-1)†	3.3197e-1 (8.56e-2)†	1.3371e-1 (2.21e-2)†	3.7846e-1 (8.08e-2)†	1.7161e-1 (3.85e-2)†	2.9419e-2 (7.35e-3)
	15	9.5704e-2 (4.67e-5)†	4.8765e-1 (1.12e-1)†	3.9697e-1 (2.43e-1)†	1.5532e-1 (3.30e-3)†	4.9938e-1 (1.83e-1)†	4.2556e-1 (1.26e-1)†	3.4919e-2 (5.53e-3)
	20	1.0199e-1 (2.27e-4)†	5.5657e-1 (1.36e-1)†	3.7109e-1 (1.04e-1)†	1.6047e-1 (4.48e-3)†	4.5800e-1 (1.51e-1)†	5.7624e-1 (6.27e-2)†	4.0485e-2 (7.28e-3)
DTLZ6	5	3.2452e-2 (3.66e-4)†	2.9034e-1 (9.37e-2)†	1.0875e-1 (1.80e-2)†	1.1298e-1 (9.08e-3)†	2.6034e-1 (4.63e-2)†	1.2403e-1 (9.27e-3)†	4.8471e-2 (1.18e-2)
	10	1.9068e-2 (5.52e-4)†	4.6927e-1 (1.35e+0)†	2.1703e-1 (5.50e-2)†	1.4826e-1 (3.19e-2)†	2.7687e+0 (3.50e-1)†	3.9074e-1 (8.25e-2)†	1.0220e-1 (1.46e-2)
	15	9.5726e-2 (7.33e-6)	6.6036e+0 (9.84e-1)†	2.0107e-1 (4.22e-2)†	1.6076e-1 (3.19e-2)†	8.9003e-1 (4.11e-1)†	7.2406e-1 (3.22e-2)†	9.0524e-2 (1.50e-2)
	20	1.0200e-1 (3.09e-5)	7.9336e+0 (5.58e-1)†	1.9143e-1 (2.44e-2)†	1.5991e-1 (8.88e-3)†	1.5751e+0 (5.85e-1)†	7.0436e-1 (7.43e-2)†	1.2875e-1 (2.03e-2)
DTLZ7	5	8.4449e-1 (3.11e-1)†	3.3590e-1 (1.09e-2)†	5.9077e-1 (2.76e-2)†	3.0005e+0 (5.40e-7)†	3.4832e-1 (3.54e-2)†	3.0507e-1 (1.69e-2)	3.0474e-1 (7.17e-3)
	10	2.4732e+0 (5.32e-1)†	3.8278e-1 (7.94e-1)†	1.6573e+0 (3.17e-1)†	2.3903e+0 (8.85e-2)†	1.7272e+0 (1.21e-1)†	2.5983e+0 (3.94e-1)†	8.3931e-1 (1.10e-2)
	15	2.9748e+0 (4.57e-1)†	7.1634e+0 (1.68e+0)†	2.4006e+0 (2.23e-1)†	3.4008e+0 (8.59e-2)†	2.6023e+0 (3.49e-1)†	9.7420e-1 (1.10e+0)†	2.0002e+0 (1.91e-1)
	20	3.7781e-1 (5.88e-1)†	1.6880e-1 (1.23e+0)†	3.1299e+0 (7.18e-1)†	4.2476e+0 (7.45e-2)†	6.0294e+0 (1.99e+0)†	3.2048e-1 (1.81e+0)†	2.4055e+0 (2.69e-1)

Table 5
HV comparison of the six EMO algorithms on WFG problems.

Problem	Obj.	MOEA/D	NSGA-III	RVEA	MOEA/DD	VaEA	Two_Arch2	SPSAT
WFG1	5	9.9630e+03 (1.34e+02)†	4.4531e+03 (2.12e+02)†	4.7707e+03 (5.05e+02)†	4.1588e+03 (1.09e+02)†	9.3893e+03 (4.12e+02)†	1.0350e+04 (5.21e+00)†	8.2315e+03 (7.30e+02)†
	10	8.7989e+09 (2.01e+09)†	4.1472e+09 (8.64e+07)†	4.2570e+09 (2.79e+08)†	3.8831e+09 (3.15e+07)†	5.9813e+09 (4.76e+08)†	1.1880e+10 (1.26e+08)	1.3327e+10 (1.11e+08)
	15	8.4928e+16 (1.38e+16)†	5.5691e+16 (5.47e+15)†	5.9488e+16 (5.24e+15)†	4.8360e+16 (3.64e+15)†	1.2800e+17 (1.58e+16)†	1.9107e+17 (1.63e+14)†	1.6577e+17 (1.83e+16)
	20	3.7246e+24 (3.99e+23)†	2.9136e+24 (1.46e+22)†	3.6812e+24 (3.60e+23)†	2.7832e+24 (0.00e+00)†	4.4435e+24 (0.00e+00)†	1.2042e+25 (1.34e+24)	1.2409e+25 (6.03e+23)
WFG2	5	9.6575e+03 (2.26e+02)†	1.0250e−04 (3.13e−01)	1.0180e+04 (4.36e−01)	1.0060e+04 (5.71e−01)	1.0233e+04 (2.68e−01)	1.0267e+04 (3.69e−01)	1.0002e+04 (6.45e+02)
	10	1.2019e−10 (5.97e−08)†	1.3535e−10 (7.29e−07)†	1.3097e+10 (6.24e−07)†	1.2901e+10 (5.15e−08)†	1.3413e+10 (8.12e−07)	1.3375e+10 (7.95e−06)	1.3689e+10 (2.50e−07)
	15	1.6674e+17 (1.01e+16)†	1.8865e+17 (1.11e+15)	1.8113e+17 (6.36e−15)†	1.8421e+17 (2.56e−17)†	1.8857e+17 (1.61e+15)	1.8802e+17 (1.78e−15)†	1.8483e+17 (1.33e−16)
	20	1.1112e+25 (1.03e+24)†	1.2580e+25 (1.76e+23)†	1.2596e+25 (1.33e−23)†	1.2581e+25 (2.20e+23)†	1.2961e+25 (2.74e+22)	1.2720e+25 (8.47e−22)†	1.3015e+25 (2.13e+22)
WFG3	5	6.1705e+03 (2.96e+02)†	6.8498e+03 (7.97e−01)†	6.5192e+03 (2.32e+02)†	6.6420e+03 (9.26e−01)†	6.6505e+03 (1.08e+02)†	7.0391e+03 (1.50e−01)†	7.2985e+03 (4.45e+01)
	10	3.3313e+09 (1.19e+08)†	8.6614e+09 (1.39e+08)†	4.9014e+09 (8.11e+08)†	6.7301e+09 (2.19e+08)†	8.5218e+09 (1.82e+08)†	8.6691e+09 (1.28e+08)†	9.3890e+09 (9.58e+07)
	15	3.0083e+16 (1.45e+15)†	9.5746e+16 (7.16e−15)†	3.9016e+16 (4.79e−15)†	4.5386e+16 (7.61e−15)†	1.1463e+17 (3.47e−15)†	1.1180e+17 (5.37e−15)†	1.2491e+17 (2.33e−15)
	20	1.1792e+24 (6.20e+22)†	6.2938e+24 (3.62e−23)†	2.5531e+24 (1.63e−23)†	3.0350e+24 (4.69e−23)†	8.7297e+24 (2.26e−23)†	7.4305e+24 (1.79e−23)†	8.3805e+24 (1.54e−23)
WFG8	5	6.0420e+03 (9.15e−02)†	7.8426e+03 (4.94e−01)†	7.7626e+03 (6.83e−01)†	7.7310e+03 (7.88e−01)†	7.4524e+03 (8.41e+01)†	7.1381e+03 (0.00e+00)	7.2156e+03 (4.17e−02)
	10	2.7417e+09 (8.55e−08)†	9.9496e+09 (3.84e−08)†	9.1193e+09 (1.38e−09)	8.9057e+09 (3.98e−08)	9.2461e+09 (1.99e−08)†	6.3291e+09 (1.97e−08)†	9.0359e+09 (4.97e−08)
	15	5.2143e+16 (5.37e−16)†	1.1427e−17 (1.79e−16)	1.0519e+17 (2.58e−16)	1.5565e+17 (7.25e−15)†	1.4223e−17 (5.45e−15)†	5.4270e+16 (9.46e−15)†	1.1361e+17 (1.61e−16)
	20	4.3601e+24 (3.89e+24)†	3.8352e+24 (9.67e−23)†	9.5695e+24 (2.06e−24)	1.0316e+25 (4.92e−23)†	1.0353e+25 (2.65e−23)†	3.2005e+24 (3.20e−23)†	8.4809e+24 (1.09e−24)
WFG9	5	7.3429e+03 (4.96e−02)†	8.2637e+03 (2.40e−02)†	8.3728e+03 (1.38e−02)†	8.0511e+03 (1.80e+02)†	7.9895e+03 (3.50e+02)	8.2173e+03 (0.00e+00)†	7.8439e+03 (9.04e+01)
	10	6.2469e+09 (9.53e−08)†	1.1220e−10 (3.49e−08)†	1.0765e+10 (1.31e−08)	9.1982e+09 (7.56e−08)†	1.0354e−10 (4.69e−08)	9.0804e+09 (2.95e−08)†	1.0552e−10 (1.10e−08)
	15	5.5853e+16 (2.52e−16)†	1.4314e−17 (1.07e−16)†	1.3490e+17 (8.87e−15)	1.1987e+17 (1.65e−16)†	1.2860e−17 (9.60e−15)†	9.0172e+16 (1.19e−15)†	1.3330e−17 (4.17e−15)
	20	4.2571e+24 (2.34e−24)†	8.4168e+24 (7.72e−23)†	8.3675e+24 (5.82e−23)†	7.9706e+24 (9.64e−23)†	9.0206e+24 (1.34e−23)†	5.6865e+24 (3.11e−23)†	9.2961e+24 (1.75e−23)

Table 6
HV comparison of the six EMO algorithms on MaF problems.

| Problem	Obj.	MOEA/D	NSGA-III	RVEA	MOEA/DD	VaEA	Two_Arch2	SPSAT
MaF1	5	4.7599e+4 (7.21e-3)†	6.4843e+4 (5.46e+2)†	5.7883e+4 (5.94e-3)†	4.2109e+4 (9.00e+3)†	6.7871e+4 (1.81e-2)†	9.7601e+0 (5.55e-2)†	1.0371E+1 (4.56E-2)
	10	4.7599e+4 (7.21e+3)†	1.0905e+1 (7.78e-1)†	5.2074e+0 (7.41e-1)†	4.5942e+0 (3.04e-1)†	2.8292e+1 (2.93e-1)	1.7013e+1 (3.59e-1)†	2.8192E+1 (1.06E+0)
	15	4.6750e+0 (5.54e-2)†	7.6179e+0 (2.24e-1)†	3.6485e+0 (2.80e-1)†	3.4178e+0 (3.75e-1)†	4.0246e+1 (1.51e+0)†	1.7315e+1 (7.92e-1)†	2.8819E+1 (1.20E+1)
	20	4.7882e+0 (9.00e-2)†	8.6552e+0 (5.25e-1)†	4.0544e+0 (6.13e-1)†	3.4680e+0 (2.18e-1)†	6.3671e+1 (1.02e-1)†	1.6858e+1 (1.09e+0)†	4.9669E+1 (5.29E+0)
MaF2	5	7.1691e+4 (4.50e+2)†	7.1642e+4 (4.91e+2)†	7.1514e+4 (4.44e+2)†	7.1381e+4 (2.90e+2)†	7.2488e+4 (1.59e-2)†	2.1181e+1 (4.30e-2)†	4.9669E+1 (2.05E-1)
	10	6.5910e+2 (4.98e+0)	6.6081e+2 (6.41e+0)	5.8420e+2 (3.20e+1)†	6.2965e+2 (3.59e+0)†	6.5699e+2 (4.45e+0)	6.6085e+2 (2.05e+0)	6.5365E+2 (5.98E+0)
	15	2.0364e+4 (2.06e+2)	1.9912e+4 (3.74e+2)†	1.4864e+4 (1.24e-3)†	1.8755e+4 (1.74e+2)†	2.0867e+4 (9.27e+1)	2.0205e+4 (2.23e+2)	2.0665E+4 (1.02E+2)
	20	6.5104e+5 (9.55e+3)†	6.2906e+5 (5.90e-3)†	4.9175e+5 (4.58e+4)†	6.1030e+5 (4.77e+3)†	6.6428e+5 (4.04e+3)	6.3493e+5 (6.78e+3)†	6.6158E+5 (5.02E+3)
MaF3	5	3.5197e+4 (3.28e+4)†	0.0000e+0 (0.00e+0)†	0.0000e+0 (0.00e+0)†	0.0000e+0 (0.00e+0)†	0.0000e+0 (0.00e+0)†	3.1997e+1 (2.57e-4)	3.1992E+1 (1.74E-2)
	10	1.0058e+3 (1.42e+0)†	6.8265e+1 (2.64e+2)†	6.8120e+2 (4.64e+2)†	2.4078e+2 (4.07e+2)†	0.0000e+0 (0.00e+0)†	0.0000e+0 (0.00e+0)†	1.0240E+3 (1.00E-1)
	15	3.2112e+4 (1.70e+1)†	4.6787e+3 (1.14e+4)†	3.2758e+4 (8.27e+0)†	2.7979e+4 (1.23e+4)†	0.0000e+0 (0.00e+0)†	5.0123e+3 (1.12e+4)†	3.2766E+4 (1.60E+3)
	20	1.0324e+6 (8.65e+2)	2.9847e+4 (9.44e+4)†	9.6376e+5 (2.67e+5)†	5.1647e+5 (5.38e+5)†	0.0000e+0 (0.00e+0)†	0.0000e+0 (0.00e+0)†	1.0486E+6 (2.06E+3)
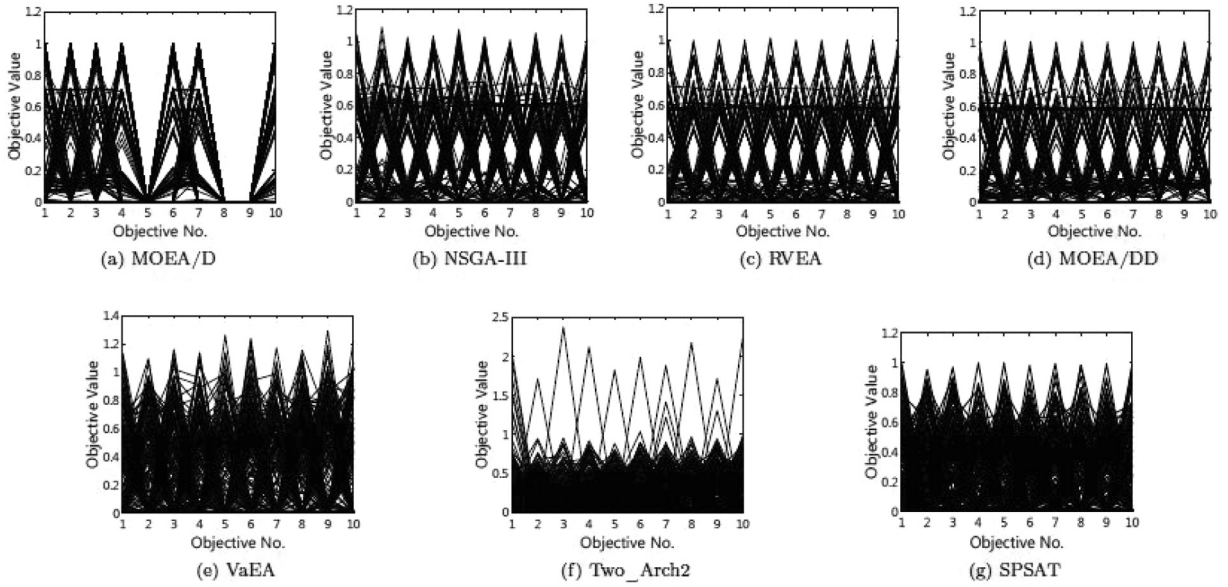
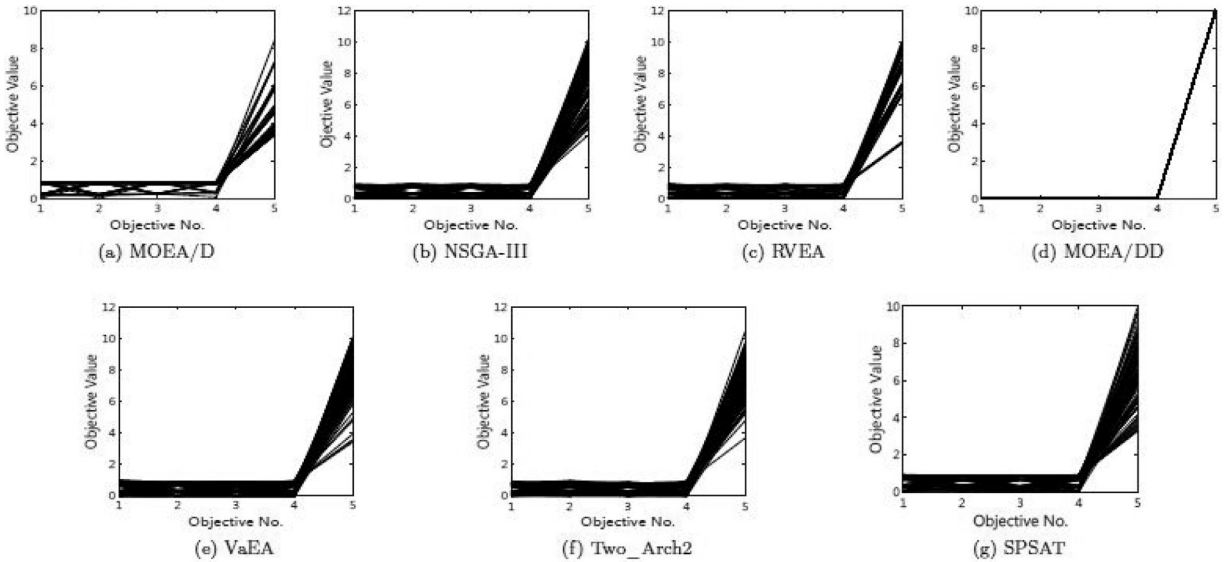**Fig. 5.** The final solution set of the six algorithms on the 10-objective DTLZ4.

Fig. 6. The final solution set of the six algorithms on the 5-objective DTLZ7.

5.2. WFG problem suite

The HV results of the seven algorithms on the WFG problems are shown in Table 5. It is obvious that SPSAT outperforms the other six algorithms, for it has the best and the second best HV results on 8 and 2 out of 20 instances, respectively. Particularly, SPSAT behaves well on the WFG1, WFG2, WFG3 and WFG9. Therefore, a conclusion can be drawn that SPSAT holds absolutely dominant position of solving the complex problems with many objectives, combining its performance on the DTLZ test suite and WFG test problems. NSGA-III performs well on WFG2, WFG8 and WFG9. Because the Pareto fronts of WFG1 and WFG3 are mixed and disconnected respectively, it can be verified again that the reference vector based algorithms are not suitable for dealing with these kinds of problems. Similar results can be seen from the performance of MOEA/D, RVEA and MOEA/DD, and they all can not perform competitively compared with the other classes of tested algorithms. However, among them, RVEA behaves well on WFG8 and WFG9; MOEA/DD performs well on WFG8. VaEA has 2 best and 6 second best HV values, and Two_Arch2 has 3 best and 4 second best HV values. They are both good at treating
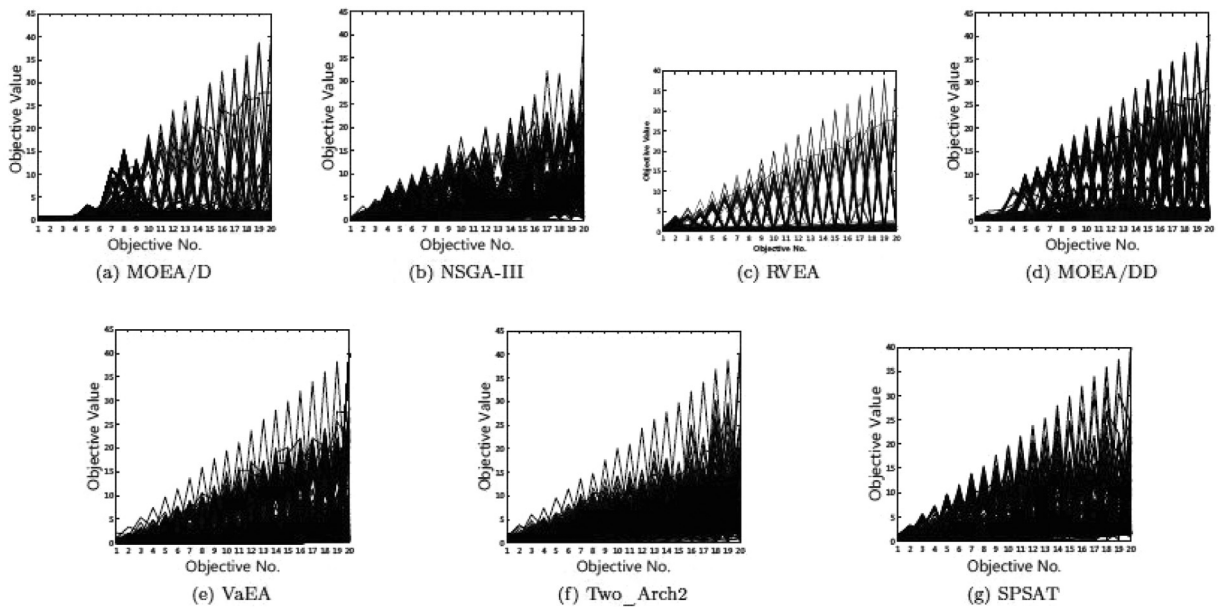
Fig. 7. The final solution set of the six algorithms on the 20-objective WFG9.

WFG2 and WFG3, but VaEA performs better in the higher dimension and Two_Arch2 behaves better in the lower dimension. Additionally, VaEA has better performance on WFG8 and WFG9.

For visualization comparison, Fig. 7 plots the final solutions of the seven algorithms on the 20-dimensional WFG9. It can be observed that all the seven algorithms converge to the Pareto front, but their distribution is distinct. Except for NSGA-III, the other six algorithms have good extensity. As for the coverage, SPSAT covers the Pareto best, MOEA/DD and RVEA have a little bit worse coverage. From the parallel coordinates of VaEA and Two_Arch2, it can be seen that some small regions fail to be covered. NSGA-III fails to cover the boundary and MOEA/D covers badly on the 1 to 6 objectives. It is hard to observe the uniformity from the parallel coordinates of WFG9 with irregular Pareto front. We can only explain that the seven algorithms all may have not bad uniformity despite their some drawbacks on the other measures.

5.3. MaF problems

Table 6 gives the HV results of the seven MOEAs on three MaF problems. MaF1 is a modified inverted DTLZ1, which is used to test if the MaOEAs are able to cope with inverted Pareto fronts. From the table, it can be seen that VaEA performs the best, and SPSAT behaves the second best. When the dimension is high, MOEA/D and NSGA-III perform not well, and the other two reference vector based algorithms, RVEA and MOEA/DD, perform the worst. Thus, the shape of the Pareto fronts can largely influence the performance of the reference vector based MOEAs, and similar conclusions can be obtained in the paper [26]. MaF2 is a modified DTLZ2, which increases the convergence difficulty and is used to test the MOEAs' abilities of concurrent convergence on different objectives. VaEA still performs the best on the MaF2, and MOEA/D performs the second best. SPSAT has a better performance on the higher dimension, while NSGA-III behaves better on the lower dimension, and both of them perform slightly worse than the former two algorithms. RVEA and MOEA/DD perform the worst on the MaF2. MaF3 is a modified convex DTLZ2, and this problem is used to assess whether MOEAs can deal with the convex Pareto fronts. On MaF3, SPSAT has the best performance. MOEA/D performs the second best. The other three reference vector based algorithms, RVEA, MOEA/DD and NSGA-III, behave the third, the fourth and the fifth best, respectively. The final solution sets of VaEA are quite away from the Pareto fronts of MaF3, which can be reflected from the 0 HV values. Two_Arch2 performs in the middle and lower level among the seven MOEAs on all tested three MaF problems.

5.4. Summary

From the experimental analysis, SPSAT outperforms the other six state-of-the-art EMO algorithms, with the best and second best results in 27 and 13 out of the 60 test instances, respectively. A conclusion can be made from the results that SPSAT has good problem-solving abilities on many-objective problems with many distinct characters. Moreover, compared with the other six algorithms, SPSAT can obtain competitive solution sets no matter what the shapes of the Pareto fronts are. The other six peer algorithms perform differently on the tested problems. Specially, MOEA/D, NSGA-III, RVEA and MOEA/DD behave well on the problems with hard convergence difficulty but encounter big challenges on MOPs with irregular and inverted Pareto fronts. The performance of VaEA is not largely influenced by the dimension of problems and the shape of
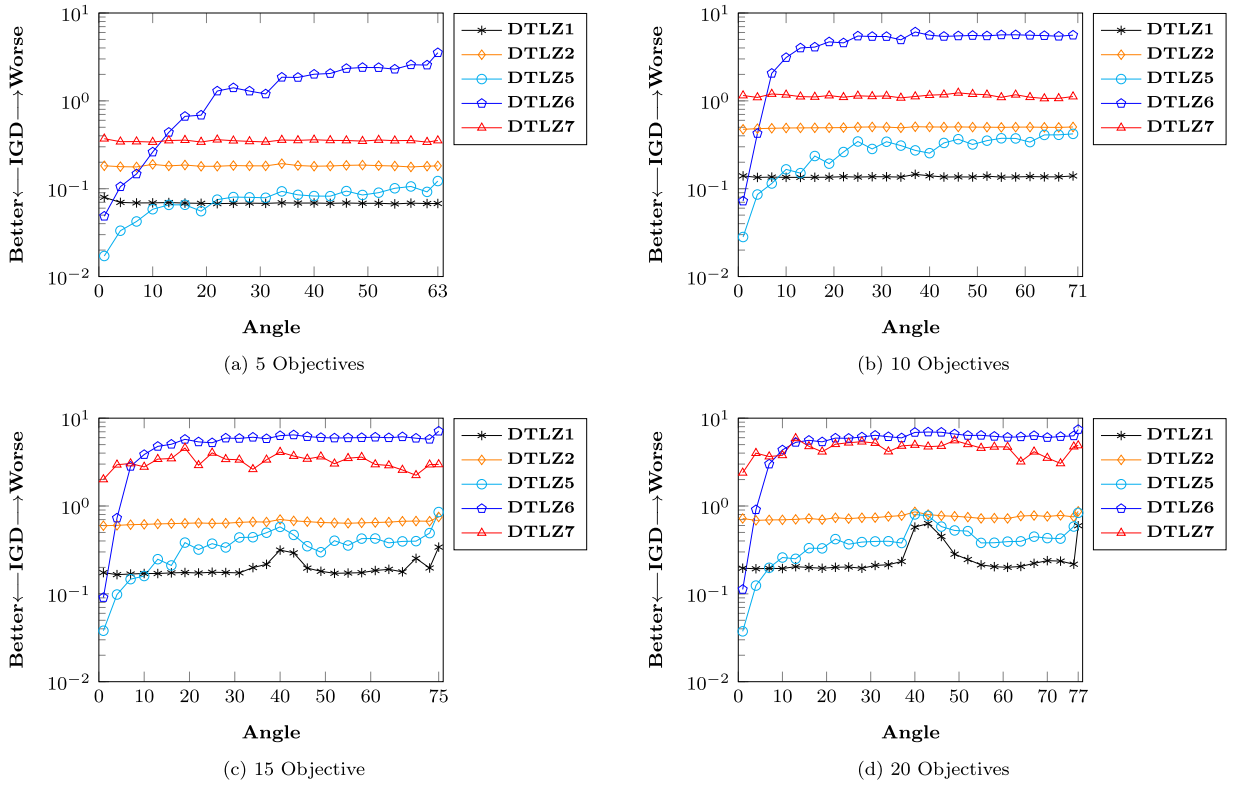
Fig. 8. IGD results for different settings of parameter α on 5-, 10-, 15- and 20-dimensional DTLZ1, DTLZ2, DTLZ5, DTLZ6, DTLZ7 test problems.

the Pareto fronts, and it is competitive in many-objective problems. Two_Arch2 can perform well on the lower dimensional problems, but is not competitive on the higher dimensional problems.

5.5. Sensitive of parameter α in SPSAT & the balance between SPS and AT

SPSAT has a specific parameter α, and it is used to control the size of subspace when partitioning the objective space, which could indirectly control the number of individuals in each subspace in order to better balance proximity and diversity. An investigation was conducted to explore the influence of α on the performance of SPSAT. The value of α varies from $0°$ to $90°$ ($0° \leq \alpha \leq 90°$), but for different numbers of objectives, the sensitive upper bound of α varies according to Eq. (2). Specially, when the number of objectives $m = 5$, the sensitive upper bound $\alpha' = 63°$; $m = 10$, $\alpha' = 71°$; $m = 15$, $\alpha' = 75°$; $m = 20$, $\alpha' = 77°$. When the setting of α is bigger than its sensitive upper bound, the objective space is not divided.

From the parameter setting in the previous experiments, α is set to different values in SPSAT depending on the number of objectives and properties of problems. For high-dimensional problems, a relatively small α is appropriate. The main reason is that a smaller α could provide bigger selection pressure among the individuals in difficult high-dimensional problems. Additionally, problems with a large number of local Pareto optimal fronts also require a relatively small α because it is helpful for SPSAT to escape from local Pareto fronts. Moreover, the shape of Pareto fronts will not influence the setting of α.

We considered the setting of α on 5-, 10-, 15- and 20-dimensional DTLZ1, DTLZ2, DTLZ5, DTLZ6, DTLZ7 test problems, which cover all properties of unimodal, multimodal, linear, concave, mix, disconnection, bias and degeneration. Fig. 8 shows the results of IGD values for different settings of parameter α on the above instances, averaging over 20 independent runs. It can be seen that as α varies from $0°$ to α', the IGD values of SPSAT follow a fluctuating ascending trend on DTLZ5 and DTLZ6 with all tested objectives, and generally maintain a stable trend on DTLZ2 with all tested objectives. Although there are light fluctuations of the trajectories on DTLZ1 and DTLZ7 with 15 and 20 objectives, the general trend remains unchanged. For DTLZ5 and DTLZ6 with all tested objectives, the best performance has been achieved when $\alpha = 1°$ because a smaller α could ensure that a narrower area could be searched. While for DTLZ2 with a unimodal property, α can take any values between $0°$ and α' without affecting the performance of SPSAT. On the 5-, and 10- dimensional DTLZ1 and DTLZ7 with special multimodal property, any value of α between $0°$ and α' has no influence on the performance of SPSAT, but a smaller α is preferred on the 15-, and 20-dimensional DTLZ1 and DTLZ7. To summarize the above results, we can conclude that although the performance of SPSAT varies with the value of parameter α, there is a pattern that can be followed to guide the setting of parameter α. For unimodal MOPs, the setting of α has no effect on the performance of SPSAT whatever

the number of objectives is, and for multimodal and degenerated MOPs, a smaller α is sufficient for all requirements of obtaining good comprehensive performance. To sum up, a smaller α can meet all needs in high-dimensional space.

The balance between the convergence and diversity is actually the balance between SPS and AT. This can be easily achieved by selecting only one individual in each subspace and then adjusting parameter α. According to the difficulty and the dimension of problems, adjusting α can adaptively balance the convergence and diversity. SPS have the leading position in SPSAT, which we care most with regard to the convergence and diversity of problems, while AT is used to select the rest and still needed individuals that can enhance the convergence further and meanwhile maintain diversity.

6. Conclusions

In this paper, a novel MaOEA, called SPSAT, is proposed to solve MaOPs. The main idea is to incorporate the convergence information of individuals into the diversity maintenance mechanism to enhance selection pressure in solving MaOPs, thereby improving the comprehensive performance of Pareto-based MOEAs.

In SPSAT, two cooperation approaches consisting of space partitioning selection and angle-based truncation, both covering convergence and distribution information, are proposed as the second selection criterion in the Pareto-based algorithms. The former approach firstly divides the objective space into subspaces from geometry perspective just using a vector and an angle, and then selects only one individual with the best proximity estimation value in each subspace. The latter takes the angle value between two individuals as their density, and gradually delete individuals with the smallest angle values. Thus, the convergence and diversity could be enhanced simultaneously.

Systematic experiments are implemented by providing comparative studies between SPSAT and six state-of-the-art algorithms on three groups of well-defined continuous benchmark suites with 5, 10, 15 and 20 objectives. Unlike some peer algorithms, which work well on only a fraction of the test problems, SPSAT can achieve a good balance between the solution set's proximity and diversity on the tested problems with different properties. Additionally, the effect of parameter α on the performance of SPSAT is investigated, and the results demonstrate that the different settings of α have a regular pattern that can be followed, and a smaller α value could satisfy the requirements covering problems of distinct properties.

This paper indicates that the idea of incorporating convergence information into the diversity maintenance mechanism to enhance selection pressure and maintain diversity simultaneously for MaOPs is very promising. Future work on developing more effective and computationally more efficient diversity maintenance mechanisms aligned with convergence information is highly desirable for the Pareto-based MOEAs. The space partitioning strategy could also be replaced by a more effective and simpler one without any parameters. Despite strong competitiveness of SPSAT shown in our first attempt, more work is desired to further investigate its benefits and limitations in the future. In this respect, the performance of SPSAT must be verified on more real-world problems.

Acknowledgements

References

[1] S.F. Adra, P.J. Fleming, A diversity management operator for evolutionary many-objective optimisation, in: Evolutionary Multi-Criterion Optimization, Springer, Nantes, France, 2009, pp. 81–94, doi:10.1007/978-3-642-01020-0_11.

[2] H.E. Aguirre, K. Tanaka, Space partitioning with adaptive ϵ-ranking and substitute distance assignments: a comparative study on many-objective MNK-Landscapes, in: Genetic and Evolutionary Computation Conference, No. 547–554, ACM Press, Montreal, Canada, 2009, doi:10.1145/1569901.1569978.

[3] N. Al Moubayed, A. Petrovski, J. Mccall, d^2mopso: Mopso based on decomposition and dominance with archiving using crowding distance in objective and solution spaces, Evol. Comput. 22 (1) (2014) 47–77.

[4] J. Bader, K. Deb, E. Zitzler, Faster hypervolume-based search using monte carlo sampling, in: Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems, 634, Springer, Auckland, New Zealand, 2008, pp. 313–326.

[5] J. Bader, E. Zitzler, HypE: an algorithm for fast hypervolume-based many-objective optimization, Evol. Comput. 19 (1) (2011) 45–76, doi:10.1162/EVCO_a_00009.

[6] N. Beume, B. Naujoks, M. Emmerich, SMS-EMOA: multiobjective selection based on dominated hypervolume, Eur. J. Oper. Res. 181 (3) (2007) 1653–1669, doi:10.1016/j.ejor.2006.08.008.

[7] P.A. Bosman, D. ThierensIGD, The balance between proximity and diversity in multi-objective evolutionary algorithms, IEEE Trans. Evol. Comput. 7 (2) (2003) 174–188, doi:10.1109/TEVC.2003.810761.

[8] R. Cheng, M. Li, Y. Tian, X. Zhang, S. Yang, Y.J. andXin Yao, A benchmarktest suite for evolutionarym any-objective optimization, Complex Intell. Syst. 3 (1) (2017) 67–81.

[9] C.A. Coello Coello, G.B. Lamont, Applications of Multi-Objective Evolutionary Algorithms, World Scientific Publisher, Singapore, 2004.

[10] D.W. Corne, N.R. Jerram, J.D. Knowles, M.J. Oates, PESA-II: Region-based selection in evolutionary multiobjective optimization, in: Proceedings of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann, San Francisco, California, 2001, pp. 283–290.

[11] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, Wiley-Interscience Series in Systems and Optimization, first ed., John Wiley & Sons, Chichester, New York, 2001.

[12] K. Deb, R.B. Agrawal, Simulated binary crossover for continuous search space, Complex Syst. 9 (2) (1994) 115–148.

[13] 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. (2013). in press 10.1109/TEVC.2013.2281535.

[14] K. Deb, A. Kumar, Real-coded genetic algorithms with simulated binary crossover: studies on multimodal and multiobjective problems, Complex Syst. 9 (6) (1995) 431–454.

[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, doi:10.1162/106365605774666895.

[16] 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, doi:10.1109/4235.996017.

[17] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable test problems for evolutionary multi-objective optimization, in: Evolutionary Multiobjective Optimization, Advanced Information and Knowledge Processing, Springer, 2005, pp. 105–145, doi:10.1007/1-84628-137-7_6.

[18] M.G. Fabre, G.T. Pulido, C.A.C. Coello, Alternative fitness assignment methods for many-objective optimization problems, in: International Conference on Artificial Evolution, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 146–157.

[19] M. Farina, P. Amato, On the optimal solution definition for many-criteria optimization problems, in: Proceedings of the NAFIPS-FLINT International Conference, IEEE Service Center, 2002, pp. 233–238, doi:10.1109/NAFIPS.2002.1018061.

[20] M. Garza-Fabre, G.T. Pulido, C.A.C. Coello, Ranking methods for many-objective optimization, in: Advances in Artificial Intelligence, 2009, pp. 633–645.

[21] H. Wang, L. Jiao, X. Yao, Two_arch2: an improved two-archive algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 19 (4) (2015).

[22] Z. He, G.G. Yen, J. Zhang, Fuzzy-based Pareto optimality for many-objective evolutionary algorithms, IEEE Trans. Evol. Comput. (2013), doi:10.1109/TEVC.2013.2258025.

[23] S. Huband, P. Hingston, L. Barone, L. While, A review of multiobjective test problems and a scalable test problem toolkit, IEEE Trans. Evol. Comput. 10 (5) (2006) 477–506, doi:10.1109/TEVC.2005.861417.

[24] E.J. Hughes, Multiple single objective Pareto sampling, in: IEEE Congress on Evolutionary Computation, vol. 4, IEEE, Canberra, Australia, 2003, pp. 2678–2684, doi:10.1109/CEC.2003.1299427.

[25] H. Ishibuchi, T. Doi, Y. Nojima, Incorporation of scalarizing fitness functions into evolutionary multiobjective optimization algorithms, in: Parallel Problem Solving from Nature, Springer, 2006, pp. 493–502, doi:10.1007/11844297_50.

[26] H. Ishibuchi, Y. Setoguchi, H. Masuda, Y. Nojima, Performance of decomposition-based many-objective algorithms strongly depends on Pareto front shapes, IEEE Trans. Evol. Comput. 21 (2) (2016) 169–190.

[27] H. Ishibuchi, N. Tsukamoto, Y. Nojima, Evolutionary many-objective optimization: a short review, in: IEEE Congress on Evolutionary Computation, 2008, pp. 2424–2431, doi:10.1109/CEC.2008.4631121.

[28] A.L. Jaimes, H. Aguirre, K. Tanaka, C.A.C. Coello, Objective space partitioning using conflict information for many-objective optimization, in: Parallel Problem Solving from Nature, PPSN XI, vol. 6238, 2010, pp. 657–666.

[29] S. Jiang, S. Yang, A strength Pareto evolutionary algorithm based on reference direction for multi-objective and many-objective optimization, IEEE Trans. Evol. Comput. (2016).

[30] Q. Zhang, K. Li, K. Deb, S. Kwong, An evolutionary many-objective optimization algorithm based on dominance and decomposition, IEEE Trans. Evol. Comput. 19 (5) (2015).

[31] Q. Zhang, K. Li, K. Deb, Q. Zhang, Efficient nondomination level update method for steady-state evolutionary multiobjective optimization, IEEE Trans. Cybern. 47 (9) (2016) 2838–2849.

[32] J. Knowles, D. Corne, M-PAES: a memetic algorithm for multiobjective optimization, in: IEEE Congress on Evolutionary Computation, vol. 1, IEEE Service Center, Piscataway, New Jersey, 2000, pp. 325–332, doi:10.1109/CEC.2000.870313.

[33] B. Li, J. Li, K. Tang, X. Yao, Many-objective evolutionary algorithms:a survey, ACM Comput. Surv. 48 (1) (2015) 1–35.

[34] K. Li, K. Deb, Q. Zhang, Q. Zhang, Efficient nondomination level update method for steady-state evolutionary multiobjective optimization, IEEE Trans. Cybern. 47 (9) (2017) 2838–2849, doi:10.1109/TCYB.2016.2621008.

[35] M. Li, S. Yang, X. Liu, Shift-based density estimation for Pareto-based algorithms in many-objective optimization, IEEE Trans. Evol. Comput. (2013). in press doi: 10.1109/TEVC.2013.2262178.

[36] M. Li, S. Yang, X. Liu, Bi-goal evolution for many-objective optimization problems, Artif. Intell. 228 (2015) 45–65. doi: 10.1016.

[37] S.Y. Li M., X. Liu, Pareto or non-Pareto: bi-criterion evolution in multi-objective optimization, IEEE Trans. Evol. Comput. (2015).

[38] M. Koppen, V.-G. Raul, B. Nickolay, Fuzzy-Pareto-dominance and its application in evolutionary multi-objective optimization, Evolutionary Multi-Criterion Optimization, 2005.

[39] M. Li, L. Zhen, Y. Yao, How to read many-objective solution sets in parallel coordinates, IEEE Comput. Intell. Mag. 12 (4) (2017) 88–100.

[40] A. Pereira, G. Munda, M. Paruccini, Generating alternatives for siting retail and service facilities using genetic algorithms and multiple criteria decision techniques, J. Retail. Consum. Serv. (1994).

[41] R.C. Purshouse, P.J. Fleming, On the evolutionary optimization of many conflicting objectives, IEEE Trans. Evol. Comput. 11 (6) (2007) 770–784, doi:10.1109/TEVC.2007.910138.

[42] M.O. Ran Cheng Yaochu Jin, B. Sendhoff, A reference vector guided evolutionary algorithm for many-objective optimization, Trans. Evol. Comput. 20 (5) (2016).

[43] 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, doi:10.1007/978-3-540-70928-2_5.

[44] K. Tan, Y. Li, Multi-objective genetic algorithm based time and frequency domain design unification of linear control systems, Proc. Artif. Intell. Real (1997).

[45] D.A.V. Veldhuizen, G.B. Lamont, Evolutionary computation and convergence to a Pareto front, in: Late Breaking Papers at the Genetic Programming 1998 Conference, Stanford University Bookstore, University of Wisconsin, Madison, Wisconsin, USA, 1998, pp. 221–228.

[46] 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, doi:10.1109/TEVC.2012.2227145.

[47] M.L. Yi Xiang Yuren Zhou, Z. Chen, A vector angle based evolutionary algorithm for unconstrained many-objective optimization, IEEE Trans. Evol. Comput. (2016).

[48] Q. Zhang, H. Li, MOEA/D: a multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731, doi:10.1109/TEVC.2007.892759.

[49] 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, doi:10.1109/TEVC.2014.2378512.

[50] J. Zheng, H. Bai, R. Shen, M. Li, A comparative study use of OTL for many-objective optimization, in: Genetic and Evolutionary Computation Conference, ACM, Madrid, Spain, 2015. in press.

[51] E. Zitzler, S. Künzli, Indicator-based selection in multiobjective search, in: Parallel Problem Solving from Nature, Springer, 2004, pp. 832–842, doi:10.1007/978-3-540-30217-9_84.

[52] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization, in: Evolutionary Methods for Design, Optimisation, and Control, CIMNE, Barcelona, Spain, 2002, pp. 95–100.

[53] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach, IEEE Trans. Evol. Comput. 3 (4) (1999) 257–271, doi:10.1109/4235.797969.