



# A decomposition-based archiving approach for multi-objective evolutionary optimization



Yong Zhang<sup>a</sup>, Dun-wei Gong<sup>a,b,\*</sup>, Jian-yong Sun<sup>c,\*</sup>, Bo-yang Qu<sup>d</sup>

<sup>a</sup> School of Information and Control Engineering, China University of Mining and Technology, 221116, China

<sup>b</sup> School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao, 266061, China

<sup>c</sup> School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, China

<sup>d</sup> School of Electrical and Information Engineering, Zhongyuan University of Technology, 450007, China

## ARTICLE INFO

### Article history:

Received 25 May 2017

Revised 21 November 2017

Accepted 26 November 2017

Available online 28 November 2017

### Keywords:

Evolutionary optimization

Multi-objective optimization

Archive

Decomposition

## ABSTRACT

External archive can be used to improve the performance of a multi-objective evolutionary algorithm. Various archiving approaches have been developed but with some drawbacks. These drawbacks such as **computation-inefficiency**, **retreating** and **shrinking**, have not yet been well addressed. In this paper, we propose an efficient **decomposition-based archiving approach (DAA)** inspired from the decomposition strategy for dealing with multi-objective optimization. In DAA, the whole objective space is uniformly divided into a number of subspaces according to a set of weight vectors. At each generation, **only one non-dominated solution** lying in a subspace is chosen to be used for **updating the external archive** in consideration of its **diversity**. A **normalized distance-based** method, incorporated with the Pareto dominance, is proposed to decide which subspace a new solution should fall into, and whether this solution should replace existing one in this subspace or not. Empirical results on a diverse set of benchmark test problems show that DAA is more efficient than a number of state-of-the-art archiving methods in terms of the diversity of the obtained non-dominated solutions; and DAA can accelerate the convergence speed of the evolutionary search for most test problems.

© 2017 Elsevier Inc. All rights reserved.

## 1. Introduction

Multi-objective optimization problems (MOPs) exist widely in real-world applications. A continuous multi-objective minimization problem can be stated as follows:

$$\begin{aligned} &\text{Minimize } F(X) = (f_1(X), f_2(X), \dots, f_M(X)) \\ &\text{subject to } X \in \Omega \end{aligned} \quad (1)$$

where  $\Omega$  is the decision space,  $f_1 - f_M$  are the  $M$  objectives to be optimized. These objectives are assumed to be conflicting with each other. For MOPs, there often exist a **set of optimal solutions** that are not comparable with each other, i.e. any solution in the set is non-dominated by the others. The set of optimal solutions is called the **Pareto front (PF)**. The two primal goals in multi-objective optimization are to find a set of approximated optimal solutions that (1) are as close as possible to the PF, and (2) are distributed as evenly as possible on the entire PF.

\* Corresponding author.

E-mail addresses: [dwgong@vip.163.com](mailto:dwgong@vip.163.com) (D.-w. Gong), [jy.sun@xjtu.edu.cn](mailto:jy.sun@xjtu.edu.cn) (J.-y. Sun).

Classical optimization methods (such as target vector approach [22]) are inefficient for solving multi-objective optimization problems because they can only find one Pareto optimal solution in a single run, meaning that they have to run several times to achieve a set of approximated Pareto optimal solutions [49]. Evolutionary algorithms (EAs) are a population-based, stochastic optimization method with the ability to search for multiple Pareto solutions in one run [4]. Multi-objective evolutionary algorithms (MOEAs) have become popular in multi-objective optimization since the first MOEA [8] developed by Schaffer. From then on, various MOEAs have been designed including the non-dominated sorting genetic algorithm (NSGA) [39], the niched-Pareto genetic algorithm (NPGA) [17], the multi-objective genetic algorithm (MOGA) [14], the multi-objective particle swarm optimization algorithm (MOPSO) [25,40], the multi-objective differential evolution (MODE) [43], the multi-method-based multi-objective evolutionary algorithm [29], the multi-objective teaching-learning optimization algorithm [34], and many others. Interested readers please refer to [51] for recent advances in the development of MOEAs.

In existing MOEAs, an external archive is usually adopted to save elite solutions found during the evolutionary search. For examples, some well-known MOEAs, such as the non-dominated sorting genetic algorithm II (NSGA-II) [11], the strength Pareto evolutionary algorithm (SPEA2) [52], the Pareto archived evolution strategy (PAES) [20], MOEA/D [46], and many others, are equipped with external archives. These studies have shown that such elitism mechanism can provide a monotonically non-degrading performance [9,21,38,53]. It has also been shown that using the external archive to create new solutions can help improve the population diversity, especially for MOPs with complicated PFs [3,33,52].

Though the use of external archive can bring certain advantages to the performance of an MOEA, it also causes some drawbacks. First, to save a candidate solution into the external archive, we need to check the dominance of this solution with respect to all the archived solutions. The time required for this process will grow exponentially along the increase of the archive size. This causes a serious computation efficiency problem. To overcome this drawback, various data structures that permit a fast search over the entire archive are proposed for efficient sortation of the elite solutions, e.g., the dominated tree [13]. However, the time complexity of these approaches is still prohibitively high as the archive size increases. Therefore, an archive with relatively small size is commonly used in existing MOEAs.

With a small-sized archive, the archive pruning (updating) process is a key for preserving its diversity and convergence to the Pareto front. A variety of pruning methods have been proposed in the archive-based MOEAs. Representative methods include the clustering approach [52,53], the adaptive grid approach [5,20], the crowding distance technique [11,45], the  $\epsilon$ -dominance-based approach [31], the preference rank-based approach [6,44], and so on. However, drawbacks such as inefficient computation, retreating and shrinking, occur in those approaches. Therefore, it is imperative to propose effective archiving methods.

In this paper, we propose a pruning method based on the decomposition strategy. Decomposition is well-known in traditional multi-objective optimization [26,42]. In 2007, Zhang and Li first employed the decomposition strategy in multi-objective optimization to propose the so-called multi-objective evolutionary algorithm with decomposition (MOEA/D) [46]. In MOEA/D, a MOP is decomposed into a number of scalar optimization sub-problems. Different solutions in the current population are linked with different sub-problems. Hence, the “diversity” among these sub-problems will naturally lead to the diversity of the population. Due to the excellent performance of MOEA/D, the idea of MOEA/D has also been incorporated with some newly proposed MOEAs, such as the reference-point-based many-objective optimization algorithm (NSGA-III) [10], the dual population paradigm [24], and the diversity-first sorting based evolutionary algorithm [19], the decomposition-based self-adaptive multi-objective evolutionary algorithm [35], the decomposition-based multi-objective evolutionary algorithm with epsilon-constraint [1], the multi-objective evolutionary algorithm with mating neighborhood sizes and reproduction operators [48], and others.

Inspired by the decomposition strategy, this paper proposes an efficient archiving approach, called the decomposition-based archiving approach (DAA), to prune the external archive. In DAA, the whole objective space is uniformly divided into a number of subspaces by a set of weight vectors. During the evolution process, only the best solution found so far can be saved into a specific subspace. The size of the archive thus depends on the number of subspaces. To save a newly created solution, we propose a normalized distance to decide which subspace it belongs to, then compare the solution with the archived solutions in the subspace. In this method, we do not need to compare new solutions with all the archived solutions. This can reduce the computational cost significantly. Experimental results on some widely-used benchmark functions show that the proposed method achieves better results in comparison with four state-of-the-art archiving approaches.

The rest of this paper is organized as follows. In Section 2, we briefly review popular archiving approaches. Section 3 formulates the proposed DAA. Experimental results are demonstrated in Section 4. Section 5 concludes the paper, and discusses further research.

## 2. Related work

Existing archive pruning methods can be classified into two categories: the additional criterion-based approach and the dominance variant-based approach.

### 2.1. The additional criterion-based approach

This kind of approach imposes some additive criteria or techniques in addition to the Pareto dominance when a non-dominated solution enters the archive [3]. Representative methods include the clustering approach (CA), the adaptive grid approach (AGA), the crowding distance approach (CDA), and others.

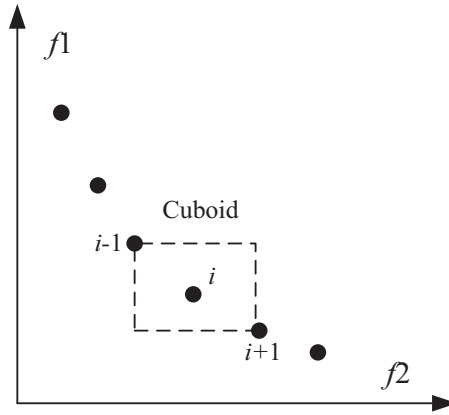


Fig. 1. Calculating the crowding distance of the  $i$ -th solution.

In the clustering approach as implemented in SPEA [52,53], non-dominated solutions in the archive are grouped into a set of clusters. If the number of clusters is more than the archive size, two clusters with minimal distance are merged into one cluster. The above step is repeated until the number of clusters is the same as the archive size. The reduced archive population is formed by selecting the solution in each cluster which has the **minimal average distance** to all other solutions within the cluster. This method can largely maintain the diversity of the archive. However, it is prone to the **retreating or oscillating phenomenon**. It has been observed that a new solution saved in the archive can be dominated by previously deleted archive solutions [8]. This phenomenon may reduce the exploration capability of the approach. Moreover, since a clustering algorithm needs to be executed repeatedly, this approach usually requires a high computational cost. In fact, the time complexity of the clustering approach is  $O(MN^3)$ , where  $N$  represents the total size of the population and archive, and  $M$  is the number of objectives.

The **crowding distance technique** presented in NSGA-II [11] is initially used for preserving the population diversity in environmental selection. It has also been widely applied to maintain the diversity of the external archives, e.g. in the hybrid multi-objective evolutionary algorithm [41], the learning-guided multi-objective evolutionary algorithm [27], the teaching-learning-based optimization algorithm [54], and others.

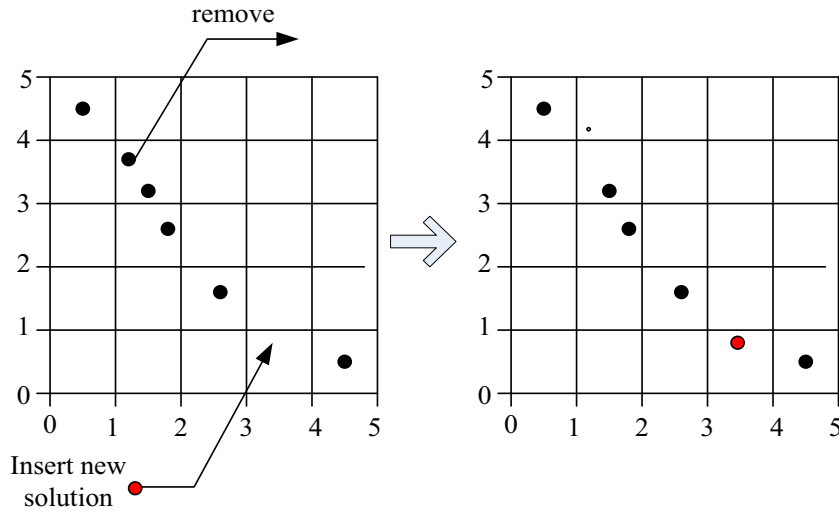
Using the crowding distance technique to save a solution to the current archive, first the solution is checked on whether it is dominated by any solution in the archive. If it is non-dominated, it will be inserted into the archive; otherwise it is discarded. To insert, all solutions that are dominated by this solution is removed from the archive. If there are more solutions than the archive size, the crowding distance of each solution in the archive is calculated. The solution(s) with the **smallest crowding distance is (are) removed from the archive**. The above procedure is repeated until the number of solutions in the archive is equal to the archive size.

Taking a bi-objective optimization problem as example, Fig. 1 depicts the method of calculating the crowding distance of a solution, where solid dots represent solutions in the archive. In the figure, the crowding distance of the  $i$ -th solution is computed as the average length of the cuboid around the solution. During the selection process, those with the smallest crowding distances will be excluded. The time complexity of this approach is  $O(MN^2)$  where  $N$  is the archive size.

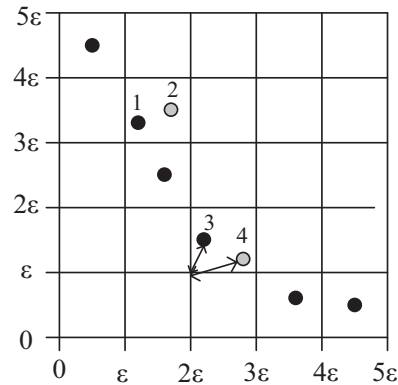
Several advanced crowding measurements have been developed exclusively for keeping the diversity of external archives. For example, Xia et al. defined two crowding distances to evaluate the diversity in **both the objective and the decision spaces**, and constructed an **aggregated fitness function** to evaluate the solutions in the archive based on the two crowding distances [45].

The **AGA approach** was proposed in PAES [20]. It aims to prune the archive to produce a well-distributed Pareto front. The main difference between AGA and the crowding distance approach lies in the pruning process when the archive reaches its maximum capacity. Instead of computing the crowding distance, AGA employs **a grid as an additional measure** to prune the archive. In AGA, the objective space is first divided into a number of regions; then the region containing the largest number of solutions is sought; a randomly-chosen solution in this region is then removed. Fig. 2 illustrates the procedure. It is repeated until the number of solutions in the archive is equal to the archive size. Note that if a new solution in the population lies beyond the bounds of these regions, these regions have to be relocated. The belonging of the solutions to these regions have to be recalculated. This will increase the computation burden. Further, since a new solution in the current archive may be dominated by the deleted solutions, AGA can also result in a retreating or oscillating phenomenon. The time complexity of AGA is  $O(MN^2)$ .

Coello et al. have applied the above AGA approach in multi-objective particle swarm optimization [5]. Cheng et al. proposed a **grid density-based measurement** combining the idea of the crowding distance and the adaptive grid. The measurement is used to quantify the crowdedness of an individual along its Pareto front, which is defined as the summation of the inversed grid distance to all other individuals in the same Pareto front [2]. Recently, Rostami and Neri proposed a new



**Fig. 2.** Illustration of the adaptive grid approach (AGA). In the figure, the black solid dots represent the non-dominated solutions in the archive, while the red solid dot refers to the new solution to be inserted in the archive (reproduced from [19]).



**Fig. 3.** Illustration of using the  $\varepsilon$ -dominance to update the archive. In the figure, the black solid dots represent the non-dominated solutions in the archive, and the gray solid dots represent the new solutions inserted in the archive.

hypervolume-sorted AGA (HAGA) to form the parent population for subsequent evolutions [36,37]. Compared with existing hypervolume-based selection approaches, HAGA calculates the value of hypervolume based only on a small number of solutions obtained by a grid logic inspired by AGA. With the application of HAGA, the solutions of a many-objective optimization problem can be achieved in an acceptable computing time.

## 2.2. The dominance variant-based approach

This kind of approach filters solutions in an archive by modifying the traditional Pareto dominance. No additional criteria are required. Hence usually this kind of approaches requires less time than the additional criterion-based approach. The most popular method in this category is the  $\varepsilon$ -dominance approach (EDA) [32]. Mostaghim and Teich [31] applied the  $\varepsilon$ -dominance to filter the archive in a multi-objective particle swarm optimization algorithm, and claimed that the computational time of this method is much less than that of the clustering technique in most cases.

When applying the  $\varepsilon$ -dominance to prune the archive, first a number of boxes with length  $\varepsilon$  are defined. At most one non-dominated solution is saved into each of these boxes at each generation. Fig. 3 shows an example of using the  $\varepsilon$ -dominance to update the archive. In the figure, solutions 1 and 2 fall in the same box. Solution 2 will be removed since solution 1 dominates it. Solutions 3 and 4 are in the same box but incomparable. Solution 4 is to be removed since it is further away from the left down corner of the  $\varepsilon$ -box.

The main operation of EDA is on finding the box to which a solution belongs. Suppose the number of boxes is  $L$ , seeking the box of an individual needs  $O(M \times L)$  basic operations. Obviously, for a population having  $Np$  individuals, seeking all the boxes needs  $O(M \times Np \times L)$  basic operations. Therefore, the time complexity of this approach is  $O(M \times Np \times L)$ .

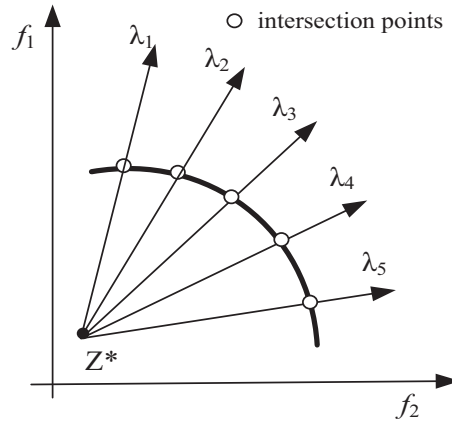


Fig. 4. Illustration of the intersection points and weight vectors.

There are some limitations with the  $\varepsilon$ -dominance approach. First, it may unexpectedly discard the extreme solutions lying on the ends of the Pareto front [2]. Second, it is worth noting that the archive size of this approach depends heavily on the value of  $\varepsilon$ . Its appropriate setting requires a priori knowledge about the optimization problem [23] which is usually not available.

Except for the EDA, approaches such as fuzzy Pareto dominance [16], preference rank [6,44], b-dominance [23] are also employed to rank the non-dominated solutions in the archive.

### 3. Decomposition-based archiving approach

In this section, we present an efficient archiving approach, called the Decomposition-based Archiving Approach (DAA). Unlike the aforementioned methods, in DAA the objective space is divided into a number of subspaces by a set of uniform weight vectors. During the evolution process, only the best solution found so far in each subspace is to be saved in the archive.

#### 3.1. Motivation

The boundary intersection (BI) approach is a frequently used decomposition technique in multi-objective optimization [7,30,46]. In this approach, the whole Pareto front is divided into a number of sub-fronts by a set of hyper-planes originating from a reference point  $Z^* = (z_1^*, z_2^*, \dots, z_M^*)$ . These hyper-planes are with different directions. Therefore, the directions can be considered as weight vectors related to the objectives.

Fig. 4 illustrates the idea of BI in 2-D objective space, where  $\lambda_1, \lambda_2, \dots, \lambda_5$  denote the weight vectors. From the figure, we see that if the lines defined by the weight vectors are distributed evenly, the intersection points between the weight vectors and the PF can provide a good approximation in terms of diversity to the whole Pareto front. Given the fact that those intersection points are unknown in advance, the following questions must be addressed when extending the idea of decomposition to updating the external archive:

**Question 1:** How to define the set of weight vectors,  $\lambda$ , so as to generate evenly distributed lines (in case  $M=2$ ) or hyper-planes (in case  $M>2$ ) ?

**Question 2:** How can we decide, from all the non-dominated solutions, the closest solution to the intersection point?

#### 3.2. The generation of evenly distributed hyper-planes

The direction  $\lambda$  influences not only the archive size, but also the distribution of solutions in the archive. In this paper, we use the method of Zhang and Li [46] to set  $\lambda$ . To be specific, let  $\lambda_1, \lambda_2, \dots, \lambda_{Na}$  represent the weight vectors. Each vector  $\lambda_i, i = 1, 2, \dots, Na$  takes a value from the following set:

$$H = \left\{ \frac{0}{h}, \frac{1}{h}, \dots, \frac{h}{h} \right\} \quad (2)$$

and satisfies  $\sum_{j=1}^M \lambda_{i,j} = 1$ . The parameter,  $h$ , determines the number of weight vectors,  $Na$ ,

$$Na = C_{h+M-1}^{M-1} \quad (3)$$

For details about the relationship among  $h$ ,  $Na$ , and  $M$ , please refer to [46]. It is expected that the directions originating from  $Z^*$  are uniformly distributed in the objective space due to the construction of the weight vectors. Since it is time-consuming to seek the exact reference point, we take the minimum value of each objective obtained during the evolution

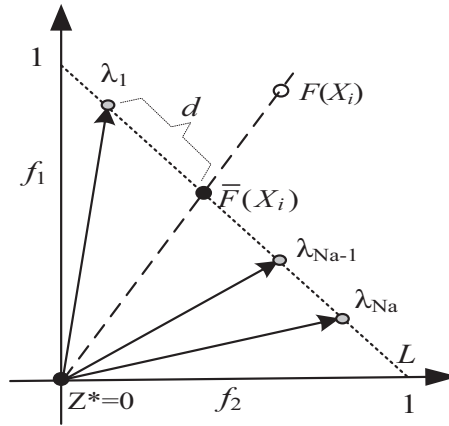


Fig. 5. Illustration of calculating the normalized distance.

process as the reference point,  $Z^*$ . That is,

$$z_i^* = \min\{f_i(X) | X \in \Omega\} \quad \text{for } i = 1, 2, \dots, M.$$

### 3.3. Updating the archive

This section aims to address the second question. Since the intersection points are unknown, an approximate method is proposed to update the archive. The proposed approximate method, called the **normalized distance method (NDM)**, consists of two steps. The first step is to determine the **affiliated weight vector** for each solution. Solutions with the same weight vector are considered to be in the same subspace. The second step is to prune the archive by deleting redundant solutions in each subspace. In the following, we present the two steps.

- (1) Consider a candidate solution  $X_i$  to be inserted into the archive, if it is dominated by any solution in the archive, it is discarded directly. Otherwise, the **normalized distance, denoted  $asd(X_i, \lambda)$** , between  $\bar{F}(X_i) = (\bar{f}_1(X_i), \dots, \bar{f}_M(X_i))$  and  $\lambda$ , is calculated as follows:

$$d(X_i, \lambda) = \|\bar{F}(X_i) - \lambda\| \quad (4)$$

where

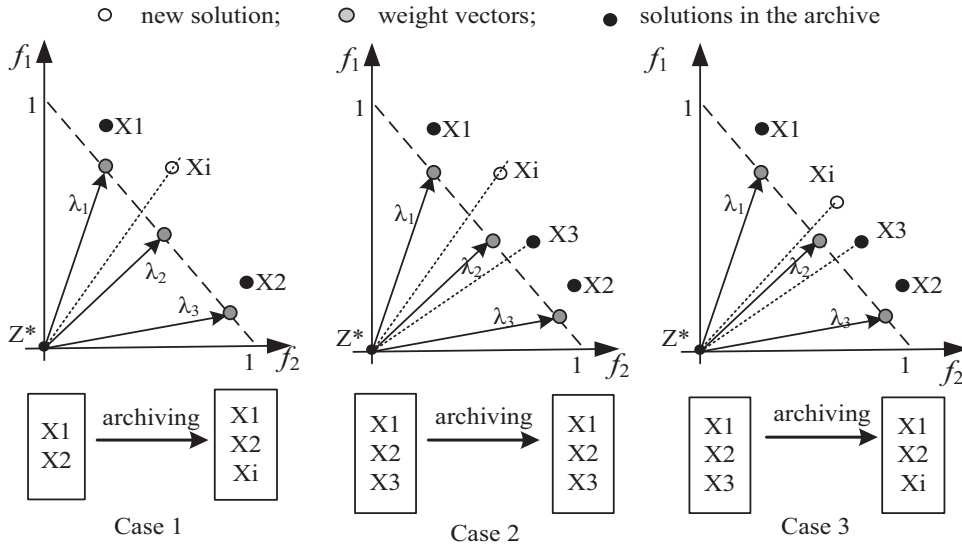
$$\bar{f}_m(X_i) = \frac{f_m(X_i) - z_m^*}{\sum_{j=1}^M (f_j(X_i) - z_j^*)}, \quad m = 1, 2, \dots, M. \quad (5)$$

Here  $\bar{F}(X_i)$  is the normalized form of  $F(X_i)$ . Let  $D = \{d(X_i, \lambda_1), d(X_i, \lambda_2), \dots, d(X_i, \lambda_{Na})\}$  be the set of distances between  $X_i$  and the  $Na$  weight vectors.  $X_i$  is then affiliated to the weight vector with the smallest normalized distance. All solutions affiliated to the same weight vector are tagged with the same label.

Fig. 5 illustrates the calculation of the normalized distance. In the figure,  $L$  represents a line segment,  $\lambda_k$  is the  $k$ -th weight vector, and  $X_i$  represents a candidate solution. For any point in  $L$ , the sum of its coordinate values is equal to 1. It is clear that all the weight vectors produced by the method in Section 3.2 will be located in this segment. Similarly, the normalized value of a solution by using Eq. (5) is also located in this segment. So it is reasonable to employ Eq. (4) to approximate the distance between  $X_i$  and the intersection points.

- (2) After assigning all non-dominated solutions into their subspaces, we then focus on deleting redundant solutions from each subspace. In our method, for solutions with the same label, **only the one closest to the affiliated weight vector can be saved in the archive.**

In Fig. 6, we give an example to show how a non-dominated solution  $X_i$  is treated in the updating procedure. First, if there is no solution in the class, then  $X_i$  is saved directly in the archive (see case 1 in Fig. 6). Otherwise, the normalized distance values of all the solutions in the class are calculated. If the distance value of  $X_i$  is not the least,  $X_i$  is discarded (see case 2 in Fig. 6). If there are solutions that have a greater distance than  $X_i$ , then such solutions are to be removed from the archive and the class. If no solution in the class has a smaller distance than  $X_i$ , then  $X_i$  is saved in the archive (see case 3 in Fig. 6).



**Fig. 6.** Illustration of all possible cases in pruning the archive, where  $Z^*$  is the reference point,  $\lambda_k$  represents the  $k$ -th weight vector related to the objectives. In case 1, there are no other solutions affiliated to  $\lambda_2$ , so  $X_i$  is kept. In case 2,  $X_i$  and  $X_3$  are all affiliated to  $\lambda_2$ , but  $X_3$  is closer to  $\lambda_2$ , so  $X_3$  is kept. In case 3,  $X_i$  is closer to  $\lambda_2$  than  $X_3$ , so  $X_i$  is saved.

### 3.4. The archiving pruning procedure

The pseudo code of the archive pruning approach is presented in [Algorithm 1](#). In the algorithm, solutions in a population  $POP$  at generation  $t$  are to be inserted into the current archive  $A(t)$ . It outputs a new archive,  $A(t+1)$ . [Algorithm 1](#) consists of two steps. First, all newly generated solutions in  $POP$  is compared with solutions in the current archive  $A(t)$  one by one (line 4). **Non-dominated solutions in  $POP$  are selected as candidate solutions** for the archive. They are saved in a temporary set, called  $TS$ . It is initialized as empty (line 1). Note that in this step, the dominated solutions in  $A(t)$  are also discarded. In the second step, solutions in the union of  $A(t)$  and  $TS$  are pruned to obtain  $A(t+1)$ . In this step, we first update the reference point  $Z^*$  based on the union of  $A(t)$  and  $TS$  (line 10). If  $Z^*$  has been changed, the affiliated weight vector for each solution in  $A(t)$  is decided by using the normalized distance (lines 11–13). For each candidate solution in  $TS$ , its affiliated weight vector is also recalculated (lines 14–16). Finally, solutions in the union of  $A(t)$  and  $TS$  are pruned to form the new external archive  $A(t+1)$  (line 17).

### 3.5. Discussions

- (1) **The upper bound of the archive.** In the proposed archiving approach, only one non-dominated solution is affiliated to each weight vector. Therefore, the upper bound of the **archive size is  $C_{H+M-1}^{M-1}$** .
- (2) **The balance of convergence and diversity.** In the first stage of our method (lines 3–8), only non-dominated solutions can be inserted into the archive. This will ensure the convergence of the archive. In the archive pruning process (lines 10–17 in [Algorithm 1](#)), since the weight vectors produced by [Section 3.2](#) are distributed evenly in the objective space, this guarantees the diversity of the non-dominated solutions among the archive.
- (3) **Time complexity.** There are **three main operations** in [Algorithm 1](#), including the **Pareto domination comparison** (lines 3–7), **the decision on the affiliated weight vectors** (lines 9–13), and **the archive pruning process** (lines 14–17). In the worst case, i.e. all solutions in the population and the archive are non-dominated, the first operation needs  $M \sum_{i=1}^{Np} (Na + i)$  Pareto comparisons. For the second operation, in the worst case, we have to recalculate the affiliated weight vectors for all the solutions in the archive (lines 11–16). In such a case, calculating the distances of all solutions in the union of  $A(t)$  and  $TS$  to the weigh vectors takes  $O(M(Na + Np) Na)$  basic operations, where  $Np$  is the population size. The decision on the affiliated weight vectors for all solutions needs  $O(M(Na + Np) Na)$  basic operations. To prune the archive, choosing the **closest solution from** a class can be done in  $O(Na + Np)$ . Therefore, the time complexity of [Algorithm 1](#) is  $O(M(Na + Np) Na)$ .

### 3.6. DAA based on the boundary intersection-based approach

Except for the DAA approach, we could also use the penalty-based boundary intersection approach to prune the archive. In the sequel, we present the method, denoted by **DAA-BI**. This approach is similar to the DAA, except that it replaces the normalized distance method by the boundary intersection method, as proposed in [\[46\]](#). In DAA-BI, the penalty-based BI is



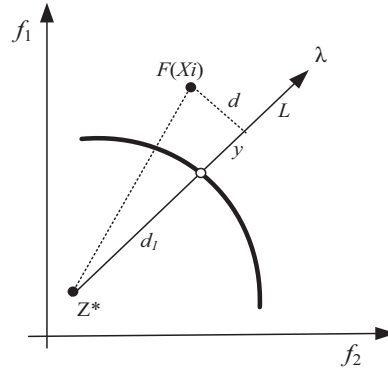


Fig. 7. Illustration of calculating the BI distance.

used to decide the affiliated weight vector for each solution. That is, the distance values between  $F(X_i)$  and the weight vector  $\lambda$  that emanates from the reference point  $Z^*$ , is computed as follows:

$$d(X_i, L) = \|F(X_i) - (Z^* + d_1 \lambda)\|$$

$$d_1 = \frac{\|(F(X_i) - Z^*)^T \lambda\|}{\|\lambda\|} \quad (6)$$

Fig. 7 shows an example of calculating the distance by DAA-BI, where  $Z^*$  is the reference point,  $\lambda$  is a weight vector,  $y$  is the projection of  $F(X_i)$  on line  $L$ , and  $d$  is the distance between  $Z^*$  and  $y$ . Similar to DAA, DAA-BI also classify a solution  $X_i$  to the weight vector with the smallest distance. All solutions affiliated to the same weight vector are tagged with the same label. After that, a method similar to DAA is used to delete redundant solutions from each subspace.

#### 4. Experimental results

In the experiments, we first justify the normalized distance method proposed in Section 3.3 by comparing DAA with DAA-BI, and discuss the effectiveness of the normalization technique to the performance of DAA. Then we compare DAA with four popular archiving approaches, namely CA, CDA, AGA, and EDA. All simulations reported in this work were conducted on a PC with a Pentium 2.6 GHz, 2.0 G CPU.

##### 4.1. Performance metrics

To validate the proposed approach, three popular metrics are used to evaluate the convergence and the diversity of the found solutions. The first is the inverted generational distance (IGD) [46]. Let  $P^*$  be a set of uniformly distributed solutions along the true Pareto front, and  $A$  be a Pareto optimal set obtained by an algorithm, the average distance from  $P^*$  to  $A$  is defined as follows:

$$IGD(A, P^*) = \frac{\sum_{u \in P^*} d(u, A)}{|P^*|} \quad (7)$$

where  $d(u, A)$  is the minimal distance between  $u$  and all the solutions in  $A$ . If  $|P^*|$  is large enough to well represent the true Pareto front, this metric can be used to measure both the diversity and the convergence of  $A$ .  $IGD = 0$  indicates that all solutions in  $A$  are in the true Pareto front, and no part of the whole Pareto front is missed.

The second is the set coverage criterion [53]. Let  $A$  and  $B$  be two approximations to the Pareto front of a problem,  $C(A, B)$  is defined as the percentage of solutions in  $B$  that are dominated by at least one solution in  $A$ , i.e.,

$$C(A, B) = \frac{|\{u \in B | \exists v \in A : v \text{ dominates } u\}|}{|B|} \quad (8)$$

$C(A, B) = 1$  represents that all solutions in  $B$  are dominated by solutions in  $A$  ( $A$  is said to cover  $B$ ). In this case, the Pareto front obtained by  $A$  is closer to the true Pareto front than that obtained by  $B$ .

The last one is the hypervolume (HV) criterion [53]. Suppose that  $Y^* = (y_1^*, y_2^*, \dots, y_M^*)$  is a reference point in the objective space which are dominated by any Pareto optimal solution,  $A$  is an obtained approximation to the true Pareto front, the HV value of  $A$  is defined as the Lebesgue measure to the set  $\cup_{x \in A} \{x' | x < x' < Y^*\}$  where  $x < y$  denotes that  $x$  dominates  $y$ . Since HV can simultaneously evaluate the diversity and the convergence of a Pareto set, it has been used widely in the literature. A larger HV value indicates a better approximation to the PF.



**Table 1**

The average metric values obtained by DAA-BI and DAA for KUR, DTLZ1 and DTLZ2 (The data to the right of the forward slash represent the standard deviation).

Algorithm	Metric	KUR	DTLZ1	DTLZ2
DAA	IGD	0.0369/0	0.0203/0	0.0529/0
	HV	193.77/0	7.9747 /0	7.4215/0
	Time	5.81/0.12	19.91/0.43	33.17/1.06
DAA-BI	IGD	0.0401/0	0.0211/0	0.0556/0
	HV	193.01/0	7.9709/0	7.4115/0
	Time	11.71/0.46	30.78/0.69	51.57/1.42
C(DAA, DAA-BI)		0/0	0/0	0/0
C(DAA-BI, DAA)		0/0	0/0	0/0

**Table 2**

The average metric obtained by DAA, CA, CDA, AGA and EDA for KUR (except C metric).

Metric	DAA	CA	CDA	AGA	EDA
IGD	<b>0.0369/0</b>	0.0376/0	0.0374/0	0.0405/0.002	0.0382/0
HV	193.77/0	<b>194.05/0</b>	193.91/0	193.41/0.280	193.27/0
Time	5.81/0.12	5.94/0.09	<b>3.26/0.09</b>	4.01/ 0.14	3.59/0.10

#### 4.2. Experiment on synthetic populations

- (1) *Experimental settings.* In this experiment, we study the performance of the developed archive pruning approaches. Three benchmark problems are used to carry out the study. These problems are KUR with two objectives, DTLZ1 and DTLZ2 with three objectives [12]. Here, KUR has a discrete Pareto front and disparately scaled objectives. DTLZ1 has a convex Pareto front, and DTLZ2's Pareto front is concave.

To assess the performance of the developed approach, we first adopt SPEA2 to generate some synthetic populations in advance. Then the developed archiving method was applied on the synthetic population to obtain a final archive. Four other archiving approaches were also applied to the synthetic populations for comparison. To obtain the synthetic population above, for each test problem, SPEA2 was run for  $T_{max}$  generations. Here, we set  $T_{max}$  to 100 for KUR, and 300 for DTLZ1 and DTLZ2. In SPEA2, the archive size is set to 100. The bit-flip mutation and single-point crossover operators are employed. The crossover and mutation rates are set to  $1/D$  and 0.9, respectively, where  $D$  is the number of decision variables. We kept all the archive solutions generated in the  $T_{max}$  generations. After termination, we obtain  $100 \times 100$  solutions for KUR, and  $300 \times 100$  solutions for DTLZ1 and DTLZ2, respectively. The five archiving approaches will be applied on these solutions for 30 independent runs.

In AGA, we set the grid size on each objective to 60 for KUR, and 10 for DTLZ1 and DTLZ2, respectively. In EDA, the value of  $\epsilon$  is set to 0.16 for KUR, 0.1 for DTLZ2, and 0.05 for DTLZ1. In DAA, the value of  $h$  in Eq. (2) is set to 110 for KUR, and 13 for DTLZ1 and DTLZ2. These settings are to ensure that the compared approaches achieve a good diversity. The reference point in the HV metric sets is  $[0, 0]$  for KUR, and  $[2, 2, 2]$  for DTLZ1 and DTLZ2, respectively.

- (2) *Comparison between DAA and DAA-BI.* This experiment tests the effectiveness of the proposed normalized distance method by comparing DAA and DAA-BI. Table 1 lists the mean and standard deviation of the three performance metrics when they are employed to deal with the three groups of populations. Figs. 8–10 display the archives with the smallest IGD value found by DAA, DAA-BI, and the four compared approaches for the three problems, respectively.

It can be seen from Table 1 that the coverage criterion obtained by DAA is similar to those obtained by DAA-BI as shown by their C values. However, the IGD and HV values of DAA are better than those of DAA-BI. We may conclude that DAA performs well in maintaining the diversity of the approximated solutions to the PF. From Figs. 8–10, it is clear that DAA can obtain archives with good diversity for DTLZ1 and DTLZ2. In addition, both DAA and DAA-BI are robust as shown by their standard deviation values on IGD and HV.

Furthermore, we can see from Table 1 that the running time of DAA is obviously less than that of DAA-BI. The main reason is that when deciding the affiliated weight vector of a solution, DAA calculates the distance directly from this solution to the weight vectors, whereas DAA-BI calculates the distance from this solution to the lines generated by the weight vectors. Overall, we may conclude that DAA outperforms DAA-BI on the three problems, KUR, DTLZ1, and DTLZ2.

- (3) *Comparison between DAA and the four popular archiving approaches.* For KUR with a discrete Pareto front, Tables 2 and 3 list the mean and standard deviation of the three performance metrics obtained by the five compared approaches.

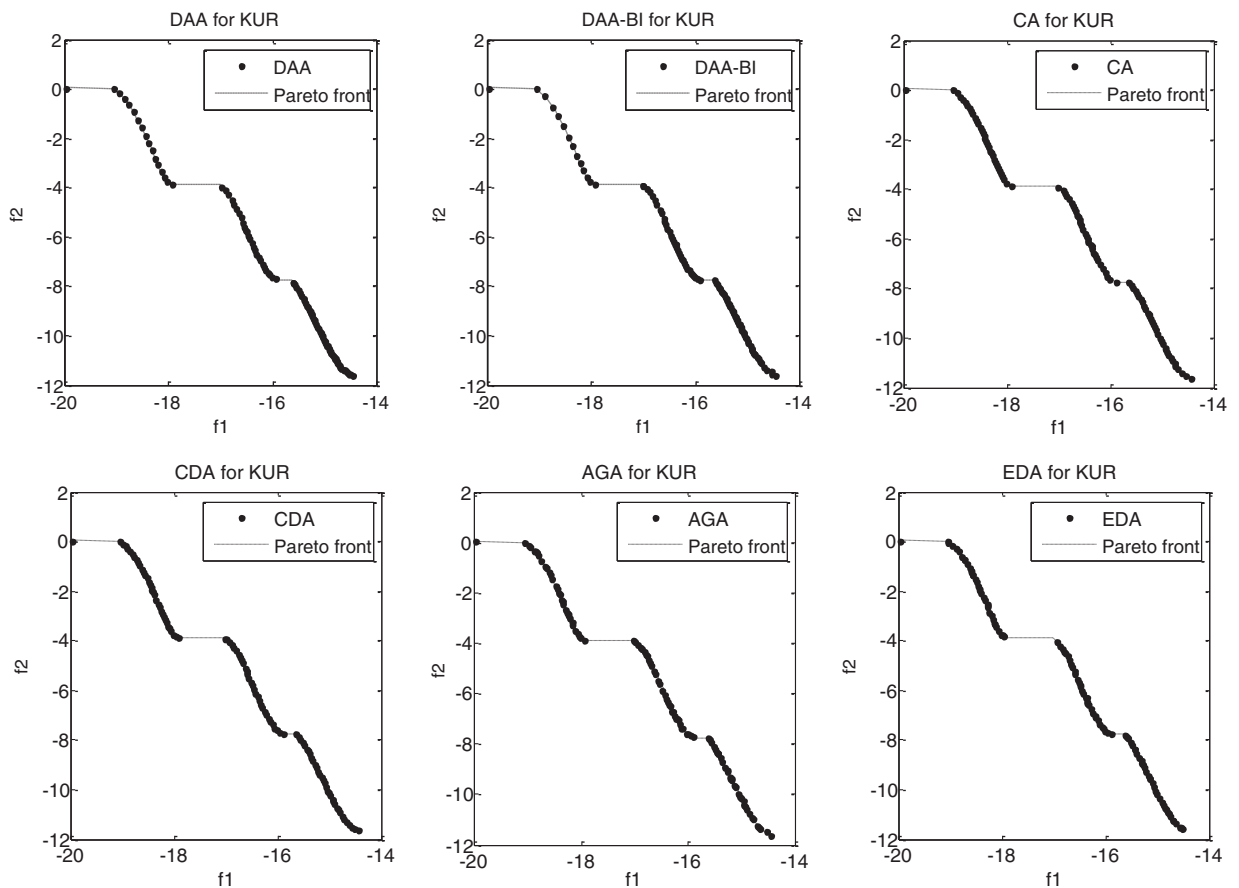


Fig. 8. Plots of the archives with the smallest IGD value obtained by the six approaches for KUR.

Table 3

The average set coverage metric values obtained by DAA, CA, CDA, AGA and EDA for KUR.

Metric	DAA	CA	CDA	AGA	EDA
C(DAA,*)	–	8.0/0	6.0/0	8.58/3.13	4.0/0
C(CA,*)	0/0	–	4.0/0	5.90/3.04	1.0/0
C(CDA,*)	0/0	5.0/0	–	5.24/1.55	2.0/0
C(AGA,*)	0/0	2.55/1.50	1.25/0.50	–	1.25/0.5
C(EDA,*)	0/0	5.0/0	6.0/0	8.30/2.90	–

It can be seen from Table 2 that for the archives found by the five compared approaches, DAA obtains the best IGD values. This implies that DAA is able to improve the convergence to the Pareto front. Furthermore, we see from the C values shown in Table 3 that 8.0% solutions of CA, 6.0% solutions of CDA, 8.58% solutions of AGA, and 4.0% solutions of EDA are dominated by those of DAA, respectively. In contrast, the C values obtained between DAA and the compared approaches are all 0. Since the scale of  $f_1$  of KUR is larger than  $f_2$ , most solutions of DAA are concentrated in the bottom right of the Pareto front, as shown in Fig. 8. Those sparse solutions on the top left of the Pareto front degrade the HV value of DAA.

Tables 4 and 5 list the experimental results obtained by the five archiving approaches for DTLZ1. We can see that among the archives found by the five approaches, DAA has the best performance in terms of IGD and HV, CA is the second best, but its running time is longer than DAA and the other approaches. EDA ranks the third in terms of IGD and HV, but it spends the shortest time. Moreover, as shown in Table 5, DAA achieves the best coverage rate. In the archives obtained by the rest approaches, there are still some solutions that are dominated by DAA. In contrast, the C values between DAA and the compared approaches are all 0. From Fig. 9, we can see that there is always one isolated point in the approximated Pareto sets obtained by CA, CDA, and AGA. It is also seen that the distributions of the solutions obtained by CDA and AGA are worse than the other three compared approaches.

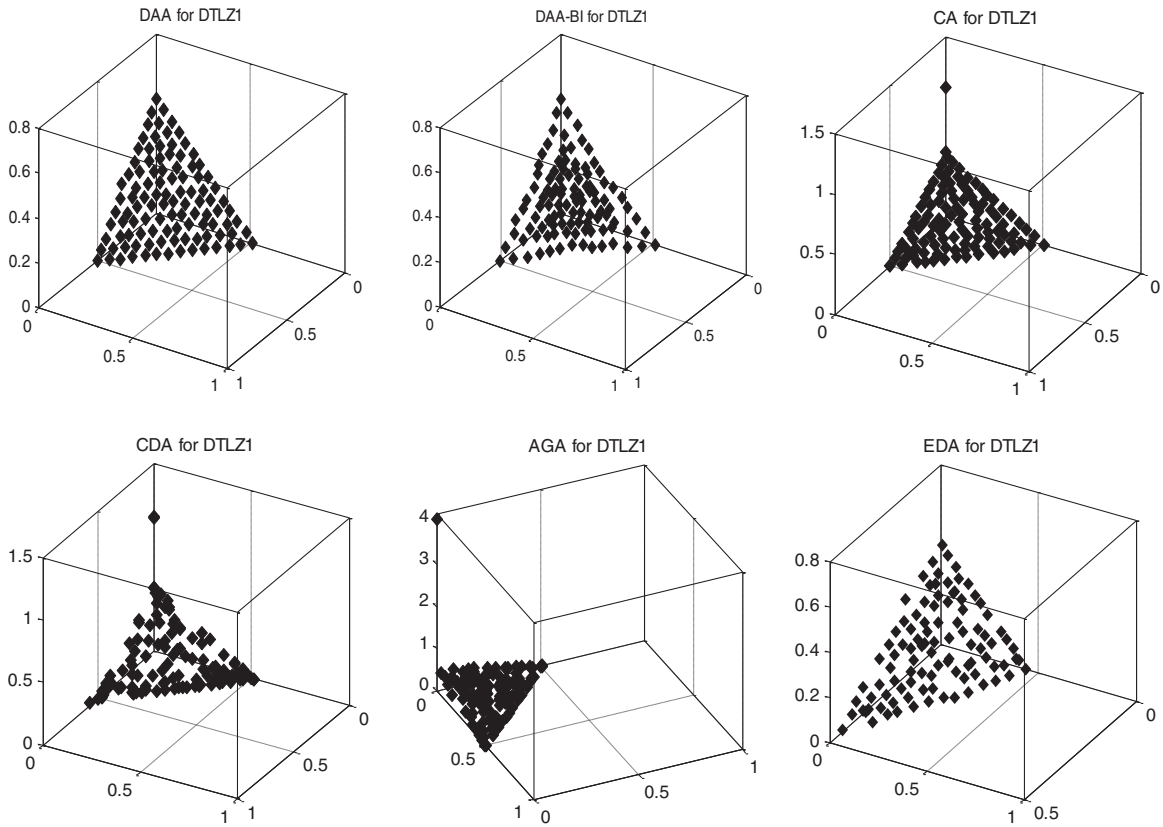


Fig. 9. Plots of the archives with the smallest IGD value obtained by the six approaches for DTLZ1.

Table 4

The average metric values obtained by DAA, CA, CDA, AGA and EDA for DTLZ1 (except C metric).

Metric	DAA	CA	CDA	AGA	EDA
IGD	<b>0.0203/0</b>	0.0212/0	0.0249/0	0.0241/0.0013	0.0231
HV	<b>7.9747 /0</b>	7.9734/0	7.9725/0	7.9645/0.0017	7.9730/0
Time	19.9/0.43	21.9/0.61	16.5/0.11	15.3/0.42	<b>14.5/0.47</b>

Table 5

The average set coverage metric values obtained by DAA, CA, CDA, AGA and EDA for DTLZ1.

Metric	DAA	CA	CDA	AGA	EDA
C(DAA,*)	–	1.0/0	2.0/0	1.9/ 0.43	4.0/0
C(CA,*)	0/0	–	2.0/0	2.08/0.86	2.0/0
C(CDA,*)	0/0	1.0/0	–	0.94/0.07	2.0/0
C(AGA,*)	0/0	0/0	0.54/0.5	–	0.18/0.40
C(EDA,*)	0/0	0/0	0/0	1.29/0.44	–

Tables 6 and 7 summarize the experimental results of the five archiving approaches for DTLZ2. Comparing DAA with the other four approaches, we see that DAA achieves the best performance in terms of IGD and HV. According to the C values shown in Table 7, we see that there are some solutions dominated by DAA in the archives obtained by CA, CDA, AGA and EDA. In contrast, the proportions by which they dominate DAA are all 0. CA has obtained good IGD and HV values, but its running time is much longer than that of DAA and the other approaches. Furthermore, Fig. 10 shows that the diversities of the archives obtained by both DAA and CA are good. However, among the archives obtained by EDA and CDA, some solutions are still far away from the true Pareto front.

Overall, although the running time of DAA is relatively longer than those of CDA, AGA, and EDA, DAA has achieved the best convergence for all the three test problems and the best diversity for DTLZ1 and DTLZ2. For KUR with disparately scaled objectives, the diversity achieved by DAA is worse than the four compared approaches. However, according to

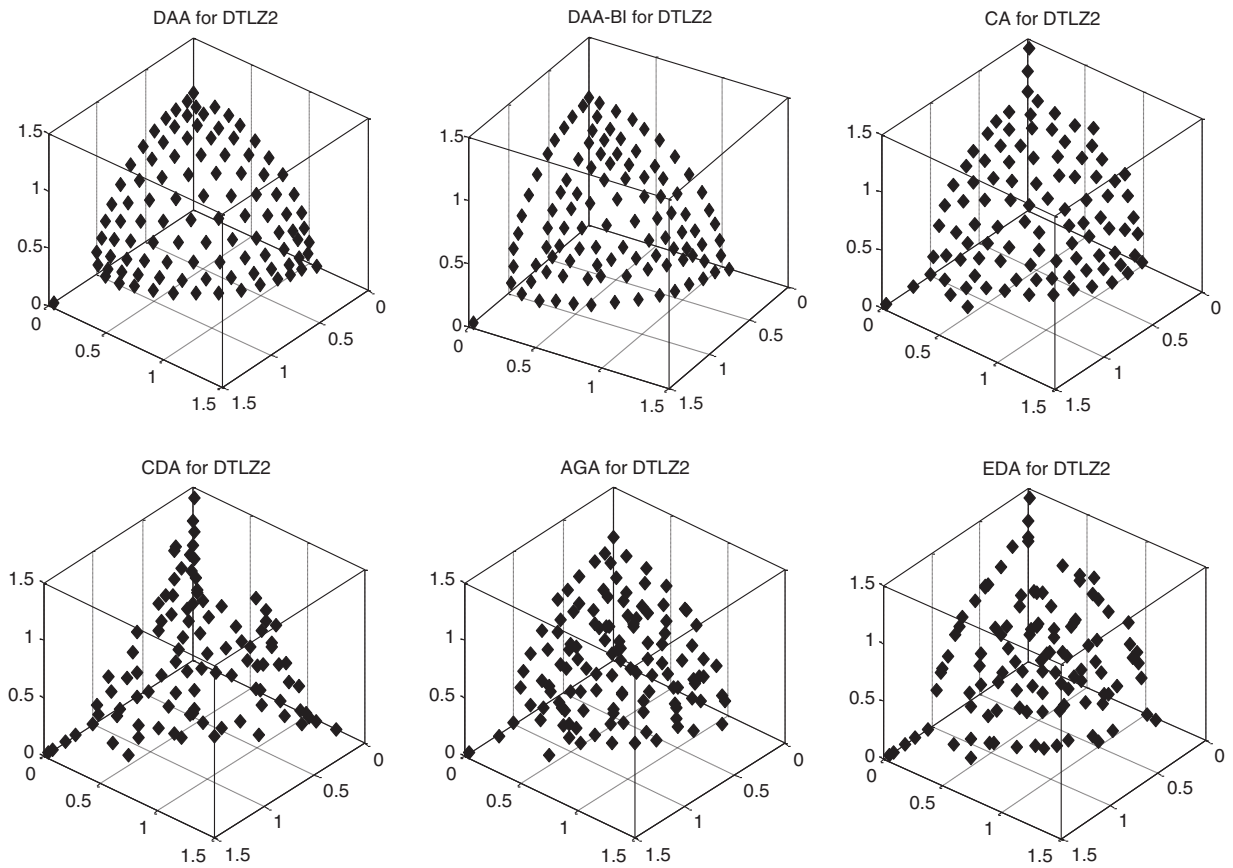


Fig. 10. Plots of the archives with the smallest IGD value obtained by the six approaches for DTLZ2.

Table 6

The average metric values obtained by DAA, CA, CDA, AGA and EDA for DTLZ2.

Metric	DAA	CA	CDA	AGA	EDA
IGD	<b>0.0529/0</b>	0.0564/0	0.0673/0	0.0594/0.0007	0.0727/0
HV	<b>7.4215/0</b>	7.3964/0	7.3859/0	7.3867/0.0098	7.3961/0
Time	33.1/1.06	60.2/2.10	<b>26.6/0.98</b>	38.5/0.97	31.0/0.75

Table 7

The average set coverage metric values obtained by DAA, CA, CDA, AGA and EDA for DTLZ2.

Metric	DAA	CA	CDA	AGA	EDA
C(DAA,*)	–	4.0/0	5.0	1.9/1.17	3.0/0
C(CA,*)	0/0	–	1.0	1.64/1.65	1.0/0
C(CDA,*)	0/0	1.0/0	–	0.9/0.74	0/0
C(AGA,*)	0/0	2.0/1.41	1.5/0.58	–	1.6/0.46
C(EDA,*)	0/0	2.0/0	1.0	1.37/1.18	–

the suggestion in [45], incorporating a simple objective normalization technique can improve the performance of DAA in the case of disparately scaled objectives.

#### (4) More discussion on the normalization technique

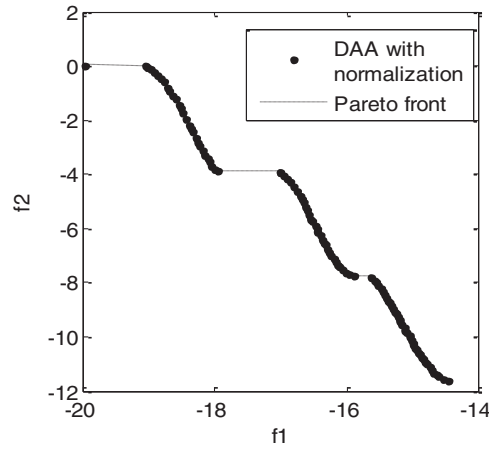
In the following, the normalization technique is incorporated into DAA to investigate whether the performance can be improved for disparately scaled objectives. In the technique, each objective is normalized as follows:

$$\tilde{f}_i = \frac{f_i - z_i^*}{z_i^{\max} - z_i^*} \quad (9)$$

**Table 8**

The average metric values obtained by DAA with/without the normalization technique for KUR.

Algorithm	IGD	HV	Time	C(DAA, DAA-N)	C(DAA-N, DAA)
DAA with normalization (DAA-N)	0.0353/0	194.01	5.94/0.13	0/0	0/0

**Fig. 11.** Plot of the archives obtained by DAA with the normalization technique for KUR.**Table 9**

The parameter settings for DAA, AGA and EDA on optimizing MOPs with different scale functions.

No. of objectives	DAA	AGA	EDA
2-objectives	$h = 100$	The grid size on each objective space $G_s = 60$	$\varepsilon = 0.01$
3-objectives	$h = 18$	$G_s = 12$	$\varepsilon = 0.06$
4-objectives	$h = 10$	$G_s = 8$	$\varepsilon = 0.12$
5-objectives	$h = 8$	$G_s = 5$	$\varepsilon = 0.25$

where  $Z^{\max} = (z_1^{\max}, z_2^{\max}, \dots, z_M^{\max})$  is the upper bound of the found Pareto front,  $Z^* = (z_1^*, z_2^*, \dots, z_M^*)$  is the reference point as stated in Section 3.2. The objective normalization technique is implemented in line 10 of Algorithm 1.

Taking KUR as an example, Table 8 lists the comparison results of DAA with and without the normalization technique. Fig. 11 plots the archive produced by DAA with the normalization technique. Comparing the IGD and HV values shown in Tables 8 and 1, we can see that the performance of DAA is improved significantly. As shown in Fig. 11, the number of solutions at the top left of the Pareto front is much more than that of the plot in Fig. 8. In addition, the normalization technique does not degrade the convergence of DAA as indicated by the set coverage values.

#### 4.3. Experiments on DAA embedded in SPEA2

To further assess the performance of DAA, we compare it with CA, CDA, AGA, and EDA when these approaches are embedded within SPEA2. That is, SPEA2 is considered as the backbone where these pruning approaches are used to manage the archives. Twelve widely used problems are selected in this experiment, including ZDT1-3, DTLZ1-2, DTLZ7 [12], WFG1-WFG5 [18], and R2\_DTLZ2\_M5 [47]. These problems cover a diverse set of multi-objective optimization problems with convex/concave, connected/disconnected Pareto fronts, disparately scaled objectives, and shifted/rotated Pareto optimal solutions.

In order to validate the capability of DAA on tackling problems with more than three objectives, the number of objectives of DTLZ is set to four. Here, the population size, the archive size and the maximal number of generations are set to 100, 200, 300, and 500, respectively. For the other parameters, please see Table 9. The reference point in HV metric are set to [1,1] for ZDT1-ZDT3, [2,2,2] and [2,2,2,2] for DTLZ1 and DTLZ2 with three and four objectives, [1,1,10] and [1,1,1,10] for DTLZ7 with three and four objectives, [3,6,9] for WFGs with three objectives, [3,3,3,3,3] for R2\_DTLZ2\_M5 with five objectives, respectively.

In this experiment, the normalization technique is incorporated into DAA, together with EDA, CDA, CA, and AGA. Tables 10 and 11 list the means and standard deviations of the two metrics obtained by the compared algorithms. The best values among the five algorithms are typeset in bold. Furthermore, the Wilcoxon rank sum test (r-test) at the significant level of 5% is employed to test the statistical significance of experimental results in terms of IGD and HV. Here “N” indicates

**Table 10**

The average IGD values obtained by SPEA2 with CA, DAA, CDA, EDA, and AGA respectively.

Algorithms	SPEA2 + DAA		r-test	SPEA2 + CA		r-test	SPEA2 + CDA		r-test	SPEA2 + EDA		r-test	SPEA2 + AGA		r-test
	Mean/std.			Mean/std.			Mean/std.			Mean/std.			Mean/std.		
ZDT1-2	<b>0.0039/5.0e-5</b>	<b>0.0039/5.0e-5</b>	N	<b>0.0039/5.0e-5</b>		N	<b>0.0039/5.0e-5</b>		N	0.0041/8.0e-5		N	0.0043/0.0002		Y+
ZDT2-2	<b>0.0040/0.0001</b>	0.0041/0.0007	N			N	0.0041/0.0001		N	0.0042/0.0001		Y+	0.0046/0.0002		Y+
ZDT3-2	<b>0.0084/0.0094</b>	0.0085/0.0106	N			N	0.0176/0.0149		Y+	0.0108/0.0111		N	0.0090/0.0091		N
DTLZ1-3	<b>0.0151/0.0014</b>	0.0159/0.0006	N			N	0.0516/0.0101		Y+	0.0414/0.0096		Y+	0.0483/0.0149		Y+
DTLZ2-3	<b>0.0381/ 0.0003</b>	0.0389/0.0003	N			N	0.0427/0.0011		Y+	0.0392/0.0005		Y+	0.0401/0.0009		Y+
DTLZ7-3	<b>0.0369/0.0005</b>	0.0389/0.0007	N			N	0.0448/0.0023		Y+	0.0443/0.0014		Y+	0.0447/0.0015		Y+
DTLZ1-4	<b>0.0376/0.0066</b>	0.0540/0.0208	Y+			Y+	0.0516/0.0101		Y+	0.0414/0.0096		Y+	0.0483/0.0149		Y+
DTLZ2-4	<b>0.0945/0.0005</b>	0.0967/0.0014	Y+			Y+	0.1051/0.0030		Y+	0.0974/0.0014		Y+	0.1034/0.0041		Y+
DTLZ7-4	<b>0.1022/0.0014</b>	0.1084/0.0013	Y+			Y+	0.1203/0.0021		Y+	0.1052/0.0012		Y+	0.1113/0.0021		Y+
WFG1-3	<b>0.5248/0.0509</b>	0.5405/0.0462	Y+			N	0.5266/0.0317		N	0.5694/0.0329		Y+	0.5278/0.0696		N
WFG2-3	<b>0.1050/0.0037</b>	0.1065/0.0061	N			Y+	0.1201/0.0043		Y+	0.1164/0.0058		Y+	0.1356/0.0054		Y+
WFG3-3	0.0792/0.0041	0.0792/0.0064	N			N	<b>0.0789/0.0049</b>		N	0.1110/0.0054		Y+	0.1058/0.0067		Y+
WFG4-3	<b>0.1478/0.0001</b>	0.1541/0.0019	Y+			Y+	0.1803/0.0042		Y+	0.1599/0.0011		Y+	0.1663/0.0034		Y+
WFG5-3	<b>0.1606/0.0007</b>	0.1622/0.0011	Y+			Y+	0.1794/0.0010		Y+	0.1657/0.0006		Y+	0.1673/0.0013		Y+
R2_DTLZ2_M5	<b>0.4318/0.0039</b>	0.4592/0.0025	Y+			Y+	0.4471/0.0138		Y+	0.4627/0.0175		Y+	0.4418/0.0188		Y+

**Table 11**

The average HV values obtained by SPEA2 with CA, DAA, CDA, EDA, and AGA, respectively.

Algorithms	SPEA2 + DA		r-test	SPEA2 + CA		r-test	SPEA2 + CDA		r-test	SPEA2 + EDA		r-test	SPEA2 + AGA		r-test
	Mean/std.			Mean/std.			Mean/std.			Mean/std.			Mean/std.		
ZDT1-2	0.6614/0.0015	0.6613/0.0013	N			N	<b>0.6615/0.0014</b>		N	0.6604/0.0025		N	0.6609/0.0001		N
ZDT2-2	<b>0.3274/0.0012</b>	0.3270/0.0021	N			N	0.3271/ 0.0024		N	0.3257/ 0.0011		Y+	0.3271/ 0.0015		N
ZDT3-2	1.0348/0.0161	<b>1.0361/0.0195</b>	N			N	1.0173/ 0.0242		Y+	1.0294/ 0.0240		N	1.0339/ 0.0157		N
DTLZ1-3	<b>7.9740/0.0018</b>	7.9735/0.0019	N			N	7.9729/0.0013		N	7.9616/ 0.0062		Y+	7.9712/0.0032		N
DTLZ2-3	<b>7.4303/0.0072</b>	7.4184/ 0.0080	Y+			Y+	7.3987/0.0109		Y+	7.3930/0.008		Y+	7.4096/0.0148		Y+
DTLZ7-3	<b>5.4008/0.0112</b>	5.3972/0.0101	N			N	5.3718/0.0107		Y+	5.3262/0.0230		Y+	5.3196/0.0325		Y+
DTLZ1-4	<b>15.9933/0.0013</b>	15.9645/0.0766	Y+			Y+	15.9841/0.0160		Y+	15.9848/0.0184		Y+	15.9863/0.0199		Y+
DTLZ2-4	<b>15.5871/0.0087</b>	15.5465/ 0.0085	Y+			Y+	15.5028/0.0128		Y+	15.5259/0.0075		Y+	15.5168/0.0142		Y+
DTLZ7-4	<b>3.9222/0.0040</b>	3.9039/0.0049	Y+			Y+	3.8501/0.0212		Y+	3.8685/0.0195		Y+	3.8692/0.0234		Y+
WFG1-3	<b>133.59/4.7630</b>	132.59/4.6915	Y+			Y+	133.21/4.6292		N	129.83/2.9529		Y+	132.42/4.5652		Y+
WFG2-3	<b>157.72/0.1880</b>	157.69/0.1124	N			N	157.40/0.1265		Y+	157.33/0.1303		Y+	157.18/0.2119		Y+
WFG3-3	128.38/0.3168	128.33/0.3313	N			N	<b>128.72/0.2734</b>		N	124.53/1.2337		Y+	126.57/0.8557		Y+
WFG4-3	<b>133.63/0.3424</b>	132.88/0.2548	Y+			Y+	131.66/0.5341		Y+	130.87/0.3719		Y+	130.74/0.6231		Y+
WFG5-3	<b>129.90/0.1548</b>	129.35/0.2181	Y+			Y+	128.61/0.2981		Y+	127.87/0.2145		Y+	127.83/0.8001		Y+
R2_DTLZ2_M5	<b>27.9683/0.0596</b>	27.8736/0.0673	N			N	27.5172/0.0528		Y+	26.4548/0.1388		Y+	27.3813/0.0862		Y+

**Algorithm 1**

The archive pruning procedure.

**Input:** A population,  $POP = \{P_1, P_2, \dots, P_N\}$ ; an archive,  $A(t)$ ; the set of weight vectors  $\lambda$  and the reference point  $Z^*$ .**Output:** The new archive,  $A(t+1)$  and  $Z^*$ .1: Set  $TS = \emptyset$ ;

2: % find all non-dominated solutions in POP

3: **For**  $k = 1$  to  $N$ 4:     Compare  $P_k$  with solutions in  $A(t) \cup TS$ ;5:     **If**  $P_k$  is dominated by some solution in  $A(t) \cup TS$ , **then** discard  $P_k$ ;6:     **Else**  $TS = TS \cup P_k$ ; and remove all solutions that are dominated by  $P_k$  from  $A(t) \cup TS$ .7:     **EndIf**8: **EndFor**

9: % update the archive

10: Update the reference point:  $z_i^* = \min\{f_i(X) | X \in A(t) \cup TS\}$  for  $i = 1, 2, \dots, M$ .11: **If**  $Z^*$  is changed,12: **then** re-calculate the affiliated weight vector for each solution in  $A(t)$  by the method in Section 3.3;13: **EndIf**14: **For**  $k = 1$  to  $|TS|$  %  $|TS|$  means the size of set  $TS$ 15:     Calculate the distance between the  $k$ -th solution in  $TS$  with the weight vectors by Eq. (5), and decide its affiliated weight vector by the method in Section 3.3;16: **EndFor**17: Apply the pruning method in Section 3.3 to prune  $A(t) \cup TS$  to form  $A(t+1)$ ;18: Return  $A(t+1)$  and  $Z^*$ .

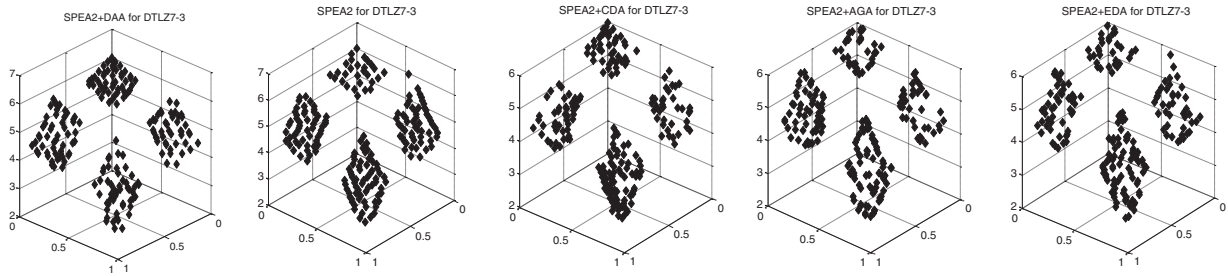


Fig. 12. Plots of the archives with the smallest IGD value obtained by the five algorithms for DTLZ7-3.

that the **null hypothesis** (i.e., the **median difference between two compared algorithms is zero** in terms of IGD or HV) cannot be rejected at the 5% level. “Y +” indicates that the alternative hypothesis holds, which suggests that SPEA2 + DAA is **significantly better than the compared one**.

For ZDT1, SPEA2 + DAA, SPEA2 + CA, SPEA2 + CDA and SPEA2 + EDA all have a good IGD value and a similar HV value. SPEA2 + AGA is significantly worse than SPEA2 + DAA in terms of the IGD value, as suggested by the r-test results. For ZDT2, SPEA2 + DAA also shows a good performance. Its IGD and HV values are **slightly better** than those of SPEA2 + CA, SPEA2 + CDA, and SPEA2 + AGA, but **significantly better than** those of SPEA2 + EDA. For ZDT3 with five disjoint Pareto fronts and disparately scaled objectives, SPEA2 + DAA achieves the best performance in terms of IGD, but its HV value is slightly worse than that of SPEA2 + CA. Possible reason is that the disjoint Pareto fronts of ZDT3 reduce the number of feasible weight vectors in DAA.

For DTLZs with more **than two objectives**, it should be noted that among the five algorithms, SPEA2 + DAA achieves the best IGD and HV values. It outperforms SPEA2 + CDA, SPEA2 + EDA, and SPEA2 + AGA significantly in most cases, as suggested by the r-test results. **Although SPEA2 + CA performs well for some DTLZ problems, such as DTLZ7-3 in terms of IGD and HV, its running time is more than that of SPEA2 + DAA**, as mentioned in Section 4.2. For illustration, the archives of DTLZ7 with three objectives obtained by the five algorithms are shown in Fig. 12. It can be seen that the Pareto fronts obtained by both SPEA2 + DAA and SPEA2 + CA show a good diversity.

For WFG1, SPEA2 + DAA achieves the best IGD and HV values. **The r-test values indicate that it significantly outperforms** SPEA2 + CA and SPEA2 + EDA. SPEA2 + CDA is competitive with SPEA2 + DAA in terms of the IGD and HV values. For WFG2, both SPEA2 + DAA and SPEA2 + CA show a good performance. The r-test values show that they significantly outperform SPEA2 + EDA, SPEA2 + CDA and SPEA2 + AGA. For WFG3, SPEA2 + DAA obtains good IGD and HV results, which is competitive with SPEA2 + DAA and SPEA2 + CA. SPEA2 + CDA achieves the best performance for WFG3. For WFG4 and WFG5, SPEA2 + DAA significantly performs the best among the five algorithms. It outperforms SPEA2 + CA, SPEA2 + CDA, SPEA2 + EDA, and SPEA2 + AGA. For R2\_DTLZ2\_M5, SPEA2 + DAA also shows the best performance in terms of IGD and HV. SPEA2 + CA achieves competitive results in comparison with SPEA2 + DAA in terms of HV.

Overall, the experimental results on these shifted/rotated test problems further validate that the proposed method is a viable alternative for pruning the archive.

## 5. Conclusion

In this paper, we proposed a novel archive pruning approach, called DAA, based on the idea of decomposition. In this approach, **the size of the archive is defined by the number of weight vectors to decompose the objective** space. At each generation, only non-dominated solutions that are close to the weight vectors are to be saved in the archive. Experimental results show that SPEA2 with DAA (SPEA2 + DAA) outperform SPEA2 with CA (SPEA2 + CA), SPEA2 with CDA (SPEA2 + CDA), and SPEA2 with EDA (SPEA2 + EDA) on most of the test problems, especially on problems with three or four objectives. Moreover, additional experiments also demonstrate that DAA with the objective normalization technique can improve the performance of MOEAs on tackling problems with disparately scaled objectives.

In the aforementioned method for generating the weight vectors, the archive size will become very large as the number of objectives increases. An important topic for further research is to seek an advanced method for generating appropriate weight vectors. Another venue of research is to apply the developed algorithm to various multi-objective optimization problems presented in cloud computing [15], image processing [50], social networking [28], and other practical application areas.

## Acknowledgements

This work was jointly supported by the **National Natural Science Foundation of China** (No. 61473299, 61573279, 61573326, 11301494), Jiangsu Six Talents Peaks Project of Province (No. DZXX-053), and Outstanding Innovation Team of China University of Mining and Technology (No. 2015QN003).



The authors would like to thank Prof. W. Pedrycz, and all the anonymous reviewers for their helpful and constructive comments.

## References

- [1] J. Chen, J. Li, B. Xin, DMOEA-epsilon C: decomposition-based multiobjective evolutionary algorithm with the epsilon-constraint framework, *IEEE Trans. Evol. Comput.* 21 (5) (2017) 714–730.
- [2] J.X. Cheng, G.G. Yen, G.X. Zhang, A grid-based adaptive multi-objective differential evolution algorithm, *Inf. Sci.* 367–368 (2016) 890–908.
- [3] C.A. Coello Coello, Evolutionary multi-objective optimization: a historical view of the field, *IEEE Comput. Intell. Mag.* 1 (1) (2006) 28–36.
- [4] C.A. Coello Coello, A.D. Christiansen, Two new GA-based methods for multiobjective optimization, *Civil Eng. Syst.* 15 (3) (1998) 207–243.
- [5] C.A. Coello Coello, G.T. Pulido, M.S. Lechuga, Handling multiple objectives with particle swarm optimization, *IEEE Trans. Evol. Comput.* 8 (3) (2004) 256–279.
- [6] I. Das, A preference ordering among various Pareto optimal alternatives, *Struct. Multidiscip. Optim.* 18 (1) (1999) 30–35.
- [7] I. Das, J.E. Dennis, Normal-boundary intersection: a new method for generating Pareto optimal points in multicriteria optimization problems, *SIAM J. Optim.* 8 (3) (1998) 631–657.
- [8] J. David Schaffer, Multiple objective optimization with vector evaluated genetic algorithms, in: *Proceedings of the First International Conference on Genetic Algorithms, Genetic Algorithms and their Applications*, 1985, pp. 93–100.
- [9] K. Deb, Current trends in evolutionary multi-objective optimization, *Int. J. Simul. Multi. Design Optim.* 1 (1) (2007) 1–8.
- [10] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach (Part I: solving problems with box constraints), *IEEE Trans. Evol. Comput.* 18 (4) (2014) 577–601.
- [11] 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.
- [12] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable test problems for evolutionary multiobjective optimization, in: A. Abraham, L. Jain, R. Goldberg (Eds.), *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, Springer, Berlin, Germany, 2005, pp. 105–145.
- [13] J.E. Fieldsend, R.M. Everson, S. Singh, Using unconstrained elite archives for multi-objective optimization, *IEEE Trans. Evol. Comput.* 7 (3) (2003) 305–323.
- [14] C.M. Fonseca, P.J. Fleming, Genetic algorithms for multiobjective optimization: formulation, discussion and generalization, in: *Proceedings of the Fifth International Conference on Genetic Algorithms*, 1993, pp. 416–423.
- [15] Z. Fu, X. Sun, Q. Liu, L. Zhou, J. Shu, Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing, *IEICE Trans. Inf. Syst.* E98-B (1) (2015) 190–200.
- [16] S. Ganguly, N.C. Sahoo, D. Das, Multi-objective particle swarm optimization based on fuzzy-Pareto- dominance for possibilistic planning of electrical distribution systems incorporating distributed generation, *Fuzzy Sets Syst.* 213 (2013) 47–73.
- [17] J. Horn, N. Nafpliotis, D.E. Goldberg, A niched Pareto genetic algorithm for multiobjective optimization, in: *Proceedings of the 1994 IEEE Conference on Evolutionary Computation*, 1, 1994, pp. 82–87.
- [18] 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.
- [19] S. Jiang, S. Yang, Convergence versus diversity in multiobjective optimization, in: *Proceedings of the 14th International Conference on Parallel Problems Solving from Nature*, 2016, Lecture Notes in Computer Science, 9921, 2016, pp. 984–993. [18].
- [20] J.D. Knowles, D.W. Corne, Approximating the nondominated front using the Pareto archived evolution strategy, *Evol. Comput.* 8 (2) (2000) 149–172.
- [21] M. Kumar, C. Guria, The elitist non-dominated sorting genetic algorithm with inheritance (i-NSGA-II) and its jumping gene adaptations for multi-objective optimization, *Inf. Sci.* 382–383 (2017) 15–37.
- [22] S. Lalwani, S. Singhal, R. Kumar, N. Gupta, A comprehensive survey: applications of multi-objective particle swarm optimization (MOPSO) algorithm, *Trans. Combinatorics* 2 (1) (2013) 39–101.
- [23] M. Laumanns, L. Thiele, K. Deb, E. Zitzler, Combining convergence and diversity in evolutionary multi-objective optimization, *Evol. Comput.* 10 (3) (2002) 263–282.
- [24] K. Li, S. Kwong, K. Deb, A dual-population paradigm for evolutionary multiobjective optimization, *Inf. Sci.* 309 (2015) 50–72.
- [25] L. Li, W.L. Wang, X.L. Xu, Multi-objective particle swarm optimization based on global margin ranking, *Inf. Sci.* 375 (2017) 30–47.
- [26] Q.Z. Lin, Z.W. Liu, Q. Yan, Z.H. Du, C.A. Coello Coello, Z.P. Liang, W.J. Wang, J.Y. Chen, Adaptive composite operator selection and parameter control for multiobjective evolutionary algorithm, *Inf. Sci.* 339 (2016) 332–352.
- [27] K. Lwin, R. Qu, G. Kendall, A learning-guided multi-objective evolutionary algorithm for constrained portfolio optimization, *Appl. Soft Comput.* 24 (2014) 757–772.
- [28] T. Ma, J. Zhou, M. Tang, Y. Tian, A. Al-Dhelaan, M. Al-Rodhaan, S. Lee, Social network and tag sources based augmenting collaborative recommender system, *IEICE Trans. Inf. Syst.* E98-D (4) (2015) 902–910.
- [29] W.K. Mashwani, A. Salhi, Multiobjective evolutionary algorithm based on multimethod with dynamic resources allocation, *Appl. Soft Comput.* 39 (2016) 292–309.
- [30] A. Messac, A. Ismail-Yahaya, C. Mattson, The normalized normal constraint method for generating the Pareto frontier, *Struct. Multidiscip. Optim.* 25 (2) (2003) 86–98.
- [31] S. Mostaghim, J. Teich, The role of  $\epsilon$ -dominance in multi-objective particle swarm optimization methods, in: *Proceedings of the 2003 IEEE Congress on Evolutionary Computation*, 3, 2003, pp. 1764–1771.
- [32] C.H. Papadimitriou, M. Yannakakis, On the approximability of trade-offs and optimal access of web sources, in: *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, 2000, p. 86.
- [33] G.T. Parks, I. Miller, Selective breeding in a multiobjective genetic algorithm, in: *Parallel Problem Solving from Nature, PPSN V. Lecture Notes in Computer Science*, 1498, 1998, pp. 250–259.
- [34] V.K. Patel, V.J. Savsani, A multi-objective improved teaching-learning based optimization algorithm (MO-ITLBO), *Inf. Sci.* 357 (2016) 182–200.
- [35] Y.T. Qi, L. Bao, X.L. Ma, Q. Miao, X.D. Li, Self-adaptive multi-objective evolutionary algorithm based on decomposition for large-scale problems: A case study on reservoir flood control operation, *Inf. Sci.* 367–368 (2016) 529–549.
- [36] S. Rostami, F. Neri, A fast hypervolume driven selection mechanism for many-objective optimisation problems, *Swarm Evol. Comput.* 34 (2017) 50–67.
- [37] S. Rostami, F. Neri, Covariance matrix adaptation Pareto archived evolution strategy with hyper volume-sorted adaptive grid algorithm, *Integr. Comput. Aided Eng.* 23 (4) (2016) 313–329.
- [38] G. Rudolph, A. Agapie, Convergence properties of some multi-objective evolutionary algorithms, in: *Proceedings of the 2000 IEEE Conference on Evolutionary Computation*, 2, 2002, pp. 1010–1016.
- [39] N. Srinivas, K. Deb, Multiobjective optimization using nondominated sorting in genetic algorithms, *Evol. Comput.* 2 (3) (1994) 221–248.
- [40] B. Tang, Zhu Z.X., H.S. Shin, A. Tsourdos, J.J. Luo, A framework for multi-objective optimisation based on a new self-adaptive particle swarm optimisation algorithm, *Inf. Sci.* 420 (2017) 364–385.
- [41] L.X. Tang, X.P. Wang, A hybrid multiobjective evolutionary algorithm for multiobjective optimization problems, *IEEE Trans. Evol. Comput.* 17 (1) (2013) 20–45.
- [42] A. Trivedi, D. Srinivasan, K. Sanyal, A. Ghosh, A Survey of multiobjective evolutionary algorithms based on decomposition, *IEEE Trans. Evol. Comput.* 21 (3) (2017) 440–462.
- [43] X.P. Wang, L.X. Tang, An adaptive multi-population differential evolution algorithm for continuous multi-objective optimization, *Inf. Sci.* 348 (2016) 124–141.

- [44] Y.J. Wang, Y.P. Yang, Particle swarm optimization with preference order ranking for multi-objective optimization, *Inf. Sci.* 179 (12) (2009) 1944–1959.
- [45] H. Xia, J. Zhuang, D. Yu, Combining crowding estimation in objective and decision space with multiple selection and search strategies for multi-objective evolutionary optimization, *IEEE Trans. Cybern.* 44 (3) (2014) 378–393.
- [46] Q.F. Zhang, H. Li, MOEA/D: a multiobjective evolutionary algorithm based on decomposition, *IEEE Trans. Evol. Comput.* 11 (6) (2007) 712–731.
- [47] Q.F. Zhang, A.M. Zhou, S.Z. Zhao, P.N. Suganthan, W.D. Liu, S. Tiwari, Multiobjective optimization Test Instances for the CEC 2009 Special Session and Competition, Working Report, CES-887, School of Computer Science and Electrical Engineering, University of Essex, 2008.
- [48] S.X. Zhang, L.M. Zheng, L. Liu, S.Y. Zheng, Y.M. Pan, Decomposition-based multi-objective evolutionary algorithm with mating neighborhood sizes and reproduction operators adaptation, *Soft Comput.* 21 (21) (2017) 6381–6392.
- [49] X.Y. Zhang, Y. Tian, R. Cheng, Y.C. Jin, An efficient approach to non-dominated sorting for evolutionary multi-objective optimization, *IEEE Trans. Evol. Comput.* 19 (2) (2015) 201–213.
- [50] Y. Zheng, J. Byeungwoo, D. Xu, Q.M. Wu, H. Zhang, Image segmentation by generalized hierarchical fuzzy C-means algorithm, *J. Intell. Fuzzy Syst.* 28 (2) (2015) 961–973.
- [51] A.M. Zhou, B.Y. Qu, H. Li, S.Z. Zhao, P.N. Suganthan, Q.F. Zhang, Multiobjective evolutionary algorithms: a survey of the state of the art, *Swarm Evol. Comput.* 1 (1) (2011) 32–49.
- [52] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the strength Pareto evolutionary algorithm, in: K.C. Giannakoglou, D.T. Tsahalis, J. Periaux, K.D. Papailiou, T. Fogarty (Eds.), *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, International Center for Numerical Methods in Engineering (Cmine), 2001, 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.
- [54] F. Zou, L. Wang, X. Hei, D. Chen, B. Wang, Multi-objective optimization using teaching-learning-based optimization algorithm, *Eng. Appl. Artif. Intell.* 26 (4) (2013) 1291–1300.