Information Sciences 491 (2019) 204–222

Contents lists available at ScienceDirect

Information Sciences

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

An adaptive decomposition-based evolutionary algorithm for many-objective optimization

Dong Han[a], Wenli Du[a,*], Wei Du[a], Yaochu Jin[a,b], Chunping Wu[c]

[a] Key Laboratory of Advanced Control and Optimization for Chemical Processes, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China
[b] The Department of Computer Science, University of Surrey, Guildford GU2 7XH, U.K.
[c] School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200237, China

ARTICLE INFO

Article history:
Received 1 August 2018
Revised 31 January 2019
Accepted 24 March 2019
Available online 27 March 2019

Keywords:
Evolutionary multi-objective optimization
Many-objective optimization
Decomposition
Convergence
Diversity
Penalty boundary intersection
Adaptation

ABSTRACT

Penalty boundary intersection (PBI) is one popular method in decomposition based evolutionary multi-objective algorithms, where the penalty factor is crucial for striking a balance between convergence and diversity in a high-dimensional objective space. Meanwhile, the distribution of the obtained solutions highly depends on the setting of the weight vectors. This paper proposes an adaptive decomposition-based evolutionary algorithm for many-objective optimization, which introduces one adaptation mechanism for PBI-based decomposition and the other for adjusting the weight vector. The former assigns a specific penalty factor for each subproblem by using the distribution information of both population and the weight vectors, while the latter adjusts the weight vectors based on the objective ranges to handle problems with different scales on the objectives. We have compared the proposed algorithm with seven state-of-the-art many-objective evolutionary algorithms on a number of benchmark problems. The empirical results demonstrate the superiority of the proposed algorithm.

1. Introduction

Multi-objective optimization problems (MOPs) contain two or more conflicting objectives. Due to the conflicts between the objectives, all the objectives can not be optimized by a single solution simultaneously. Instead, the goal of optimizing MOPs is to find a set of Pareto optimal solutions that presents a tradeoff relationship between different objectives. MOPs having more than three objectives are often called many-objective optimization problems (MaOPs) [19]. MaOPs are widespread in practical applications such as filters design [11] and chemical refinery process optimization [39].

Since evolutionary algorithms and other population-based metaheuristics can find multiple solutions in a single run, they are well suited for solving MOPs [42]. In the past decades, plenty of multi-objective evolutionary algorithms (MOEAs) have been proposed and shown to be powerful in solving MOPs [48].

As previous studies [1] have shown, however, the performance of most existing MOEAs will seriously deteriorate facing with MaOPs, especially the Pareto-based MOEAs. The main reason why MOEAs become ineffective is the loss of selection

[*] Corresponding author.
E-mail addresses: handong4263@126.com (D. Han), wldu@ecust.edu.cn (W. Du), duwei0203@gmail.com (W. Du), yaochu.jin@surrey.ac.uk (Y. Jin), 13918352054@163.com (C. Wu).

https://doi.org/10.1016/j.ins.2019.03.062

pressure to push the population towards the Pareto front. With the increasing number of objectives, the proportion of non-dominated solutions in the population rapidly increases, which may be even up to 90% if the number of objectives reaches five [14]. This is the so-called dominance resistance phenomenon [22], which makes the dominance-based primary criterion fail to differentiate the candidate solutions. The environmental selection is completely under the control of the density-based secondary criterion under these circumstances. As a result, the final population may have a sufficient degree of diversity but cannot converge to the Pareto front [33].

A direct way to enhence the selection pressure is to modify the Pareto dominance relationship. A large number of modified dominance relationships have been put forward, such as grid-dominance [42], and fuzzy-based domination relation [14]. The main shortcoming of the modified strategy is that it only considers increasing the selection pressure and neglects the diversity preservation mechanism, thus making the population converge to some subregions of the Pareto front. In contrast, aiming at the secondary criterion, more advanced diversity maintaining mechanisms have been designed. In [47], a knee point driven evolutionary algorithm (KnEA) is developed, where knee points are preferred in selection. Recently, a novel dominance relation is proposed in [31], which considers both convergence and diversity via an adaptive niching technique.

Different from the Pareto-based MOEAs, indicator-based and decomposition-based MOEAs are not subject to the dominance resistance phenomenon. Of course, they also experience with their own difficulties caused by the high-dimensional objectives of MaOPs.

A variety of indicators is proposed to evaluate the quality of the approximate solution set, and then evaluate the performance of the MOEAs. The indicator-based evolutionary algorithm (IBEA) [49] is the first work to introduce an indicator into MOEAs and presents a framework of indicator-based MOEAs. Hypervolume is a widespread indicator in this kind of approach, since it is strictly monotonic to the Pareto optimality [10]. Nevertheless, suffering from the curse of dimensionality, the computational complexity for calculating the hypervolume is usually unacceptably high on MaOPs [40]. To address these issues, some cheap methods [24,25] for hypervolume estimation have been proposed. Meanwhile, several computational efficient indicators are adopted in IBEAs, e.g. R2 [13] and α indicator [32]. However, due to the different preferences on convergence or diversity, a single indicator may not be able to take into account both convergence and diversity. Meanwhile, the selection of reference points may severely impact the rationale of the indicators. In [26], a novel progressive preference articulation mechanism for decision making is integrated with hypervolume indicator so that the decision-maker can easily articulate his/her preferences to guide the search. In the recently proposed stochastic ranking based multi-indicator algorithm [20], the search preferences of different indicators are balanced by a stochastic ranking strategy.

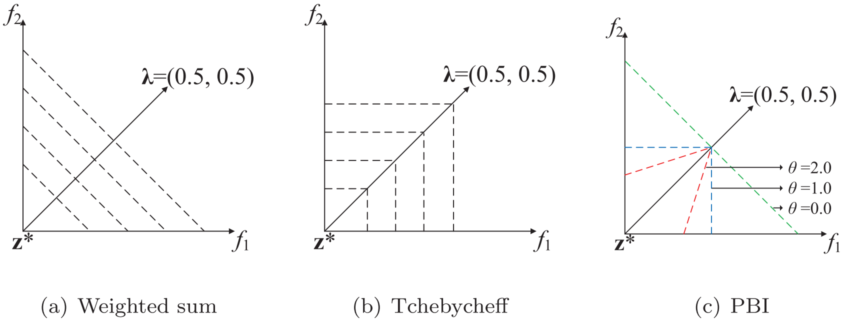
Decomposition-based MOEAs decompose an MOP into a number of single objective subproblems with a scalar function and optimize them simultaneously. A group of evenly distributed weight vectors are employed as the search direction for each subproblem in the objective space. Multiobjective evolutionary algorithm based on decomposition (MOEA/D) [45] has been an attractive decomposition-based MOEA for its simple decomposition structure and high flexibility in implementation.

Weighted sum (WS), Tchebycheff (TCH) and penalty-based boundary intersection (PBI)are three decomposition methods studied in the premier MOEA/D. The WS works well for convex Pareto front, but cannot effectively handle concave Pareto front [3], while the TCH can approximate both convex and concave Pareto front [45]. Since different decomposition approaches have different characteristic and applicability, Ishibuchi *et al.* [18] uses different approaches simultaneously in a variant of MOEA/D. In addition, several novel decomposition methods are proposed, e.g., L_p weighted approach [38] and inverted PBI [27].

The PBI is capable of obtaining a set of well-distributed solutions in the objective space and more suitable to solve MaOPs [7]. However, the property of PBI is totally determined by the value of its penalty factor. It is clear that different search stages require different search preferences [36] and an adaptive decomposition method is necessary and appropriate for such a situation [17]. Thus, several penalty strategies [16,41,43] have been suggested. However, existing work does not use the distribution information of the population or the weight vectors, which may be beneficial and reasonable for designing a penalty strategy.

Meanwhile, the performance of decomposition-based MOEAs is very sensitive to the choice of weight vectors, which directly controls the search direction for each subproblem. Unfortunately, the assumption that the uniformly distributed solutions can be specified if the weight vectors are also uniformly distributed, has been shown to be incorrect for many problems [12], especially for the MaOPs with irregular Pareto fronts or disparately scaled objectives. There are two popular ways to tackle these issues, namely, objective value normalization [44] and weight vector adjustment [2]. For irregular Pareto fronts, several effective weight vector adjustment methods [23,29] have been developed by adding and deleting weights according to the distribution of the individuals in population. In this work, we focus on the badly-scaled Pareto fronts, for which the weight adaptation strategies rely heavily on the reference point, e.g., the ideal point and nadir point. Hence, it is important to improve the estimation quality of the requisite reference point is also worthwhile. However, the estimation of the nadir point in most existing normalization methods is based only on the current solution set, although the historical solutions may help enhance the estimation quality. Additionally, the weight vector adjustment methods may deteriorate the performance on MaOPs with uniform Pareto fronts, where a set of fixed and uniformly distributed weight vectors can result in the best performance.

Based on above considerations, it is important to consider how to properly select and utilize these two components. Therefore, we mainly focus on designing the adaptation mechanisms for the PBI decomposition method and weight vector adjustment in this work, an adaptive decomposition based evolutionary algorithm (ADEA) is proposed for many-objective optimization. The contributions of this work can be summarized as follows:



(a) Weighted sum (b) Tchebycheff (c) PBI

Fig. 1. Illustration of contour lines with $\lambda = (0.5, 0.5)$, where the dashed lines are contour lines.

(1) An adaptive PBI selection strategy is developed to adjust the penalty factor for each subproblem, where the distribution information of both population and weight vectors is taken into account for a better convergence-diversity balance.
(2) According to the ideal and nadir points, a method for weight vector adaptation based on the value range of each objective value is suggested for the problems with different scales on objective values. Moreover, an archive assisted approach is adopted for the higher estimation accuracy of the nadir point and objective value range.
(3) The proposed ADEA is compared with seven state-of-the-art MOEAs on a number of test instances. The experiment results exhibit that ADEA is competitive for many-objective optimization.

The rest of this paper is organized as follows. Section 2 introduces the background knowledge briefly. We give the detailed description of the proposed ADEA in Section 3. In Section 4, the experimental results are presented and discussed. Lastly, Section 5 presents the conclusions and future work.

2. Background

In this section, several basic concepts of multi-objective optimization are first given. Then, the PBI-based decomposition and several variants are introduced.

2.1. Basic concepts

An MOP can be stated as follows:

$$\begin{array}{ll} \text{minimize} & F(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_M(\mathbf{x})) \\ \text{subject to} & \mathbf{x} \in \Omega \end{array} \tag{1}$$

where $\mathbf{x} = (x_1, \ldots, x_n)$ is an n-dimensional decision variable vector, Ω is the feasible region of the decision variables. $F: \Omega \to \mathbb{R}^M$ where M is the number of objective functions and $\mathbb{R}^M$ is the objective space.

Pareto Dominance: For two solutions $\mathbf{x}_1$, $\mathbf{x}_2 \in \Omega$, $\mathbf{x}_1$ dominates $\mathbf{x}_2$, denoted as $\mathbf{x}_1 \preceq \mathbf{x}_2$, if and only if $\forall i \in \{1, 2, \ldots, M\}$, $f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2)$ and $\exists j \in \{1, 2, \ldots, M\}$, $f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2)$.

Pareto Optimal Solution: A solution $\mathbf{x}^* \in \Omega$ is Pareto optimal, if and only if $\nexists \mathbf{x} \in \Omega$, $\mathbf{x} \preceq \mathbf{x}^*$.

Pareto Set: The set of all the Pareto optimal solutions is called Pareto set (PS): $PS = \{\mathbf{x} \in \Omega | \nexists \mathbf{y} \in \Omega, \mathbf{y} \preceq \mathbf{x}\}$.

Pareto Front: The mapping of the Pareto set in the objective space is called Pareto front (PF).

2.2. PBI Decomposition

The PBI decomposition method is defined as follows:

$$\begin{array}{ll} \text{minimize} & g^{\text{PBI}}\left(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{z}^*\right) = d_1 + \theta d_2 \\ \text{subject to} & \mathbf{x} \in \Omega \\ & d_1 = \frac{\left\| (F(\mathbf{x}) - \mathbf{z}^*)^T \cdot \boldsymbol{\lambda} \right\|}{\|\boldsymbol{\lambda}\|} \\ & d_2 = \left\| F(\mathbf{x}) - \left(\mathbf{z}^* - d_1 \frac{\boldsymbol{\lambda}}{\|\boldsymbol{\lambda}\|} \right) \right\| \end{array} \tag{2}$$

where $\boldsymbol{\lambda}$ and $\mathbf{z}^* = (z_1^*, \ldots, z_M^*)$ are the weight vector and the reference point, respectively. d_1 is the projection distance of vector $F(x) - \mathbf{z}^*$ on the weight vector $\boldsymbol{\lambda}$ and d_2 the perpendicular distance from $F(\mathbf{x})$ to $\boldsymbol{\lambda}$. θ is a predefined penalty factor which controls the bias towards convergence (minimization on d_1) or diversity (minimization on d_2). The population is accordingly pushed towards the reference point $\mathbf{z}^*$ by minimizing g^{PBI}.

Fig. 1 briefly illustrates the contour lines of three classical decomposition methods with $\boldsymbol{\lambda} = (0.5, 0.5)$. As shown in Fig. 1(a) and Fig. 1(b), the search preference of WS and TCH methods stays the same throughout the whole optimization process. On the contrary, as to PBI, Fig. 1(c) indicates that different penalty value results in different search preference. Compared with WS and TCH, the counter line of PBI shows a good comprehensive property upon the first two methods. In particular, the PBI may be considered as WS when $\theta = 0$ or TCH when $\theta = 1$.

Since the different settings of the penalty factor lead to different performances of the PBI method, several advanced penalty strategies are proposed. Ishibuchi et al. [16] propose quadratic and two-level PBI functions to strike a convergence-diversity balance. The basic idea is that by setting a threshold d^*, a smaller penalty is implemented when $d_2 < d^*$. On the contrary, a larger penalty is used when $d_2 > d^*$. Yang et al. [41] design two penalty schemes, adaptive penalty scheme (APS) and subproblem-based penalty scheme (SPS). APS linearly increases the penalty factor in a predefined interval with the iterations, whereas SPS independently specifies a penalty factor for each subproblem based on the location of its corresponding weight vector in advance. Ming et al. [43] propose an adaptive PBI method which selects the penalty values from a candidate pool Θ. Specifically, $|\Theta|$ solutions, which can achieve the best PBI function values under each candidate penalty value, are first obtained. Then, the penalty value is set according to the solution with the maximum diversity among these solutions.

3. The proposed ADEA

3.1. General framework

The general framework of ADEA is discribed in Algorithm 1. Similar to MOEA/D, ADEA decomposes an MOP into N single objective subproblems by PBI with N unit weight vectors. Firstly, the simplex-lattice design method [28] is used to generate a set of uniformly distributed weight vectors.

Algorithm 1 Framework of ADEA.

```
Require: Population size $N$, population $P$, a set of unit weight vectors $\Lambda = \{\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_N\}$, neighborhood size $T$, archive $P^{nad}$
$\Lambda_0 = \Lambda$
$P^{nad} = \Phi$
for $i = 1$, to, $N$ do
    Find the neighborhood set $B(i) = \{i_1, \ldots, i_T\}$
end for
Initialize the population $P$
$\mathbf{z}^* = $ Estimate_Ideal_Point $(P)$
$[\mathbf{z}^{nad}, P^{nad}] = $ Estimate_Nadir_Point $(P, P^{nad})$
while $Iter < MaxGen$ do
    for $i = 1$, to, $N$ do
        $P' = $ Mating_Selection$(P, \boldsymbol{\lambda}_i, B(i))$
        $y = $ Variation$(P')$
        $\mathbf{z}^* = $ Update_Ideal_Point$(y, \mathbf{z}^*)$
        $P = $ Adaptive_PBI_Selection$(\Lambda, P, y)$
    end for
    $[\Lambda, P^{nad}] = $ Weight_Vector_Adaption$(\Lambda, \Lambda_0, \mathbf{z}^*, \mathbf{z}^{nad}, P, P^{nad})$
    $Gen = Gen + 1$
end while
return $P$
```

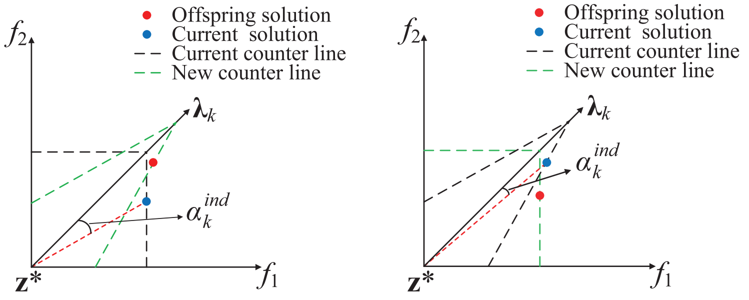
$$\begin{aligned} &\boldsymbol{\lambda}_i = \left(\lambda_i^1, \lambda_i^2, \ldots, \lambda_i^M\right), i = 1, \ldots, N \\ &\lambda_i^j \in \left\{\tfrac{0}{H}, \tfrac{1}{H}, \ldots, \tfrac{H}{H}\right\} \end{aligned} \tag{3}$$

where $\lambda_i \geq 0$ and $\sum_{i=1}^{M} \lambda_i = 1$, and H is a preset parameter in the simplex-lattice design, thus the number of weight vectors is $N = C_{H+m-1}^{m-1}$.

For each $i = 1, \ldots, N$, the set $B(i)$ contains the indices of T closest weight vectors to $\boldsymbol{\lambda}_i$ based on the Euclidean distance. The population P is initialized by randomly generating N individuals in the feasible space. $\mathbf{z}^*$ is the ideal point which contains the best value ever obtained on each objective. Different from the estimation method of the nadir point in original MOEA/D and other existing methods [8,34], we estimated the nadir point $\mathbf{z}^{nad}$ by the maximum objective value of all non-dominated individuals ever found. As the nadir point indicates the upper bound of each objective scale, an archive P^{nad} is involved to store the solutions which provide the nadir point for the next estimation to enhance the estimation accuracy.

For each subproblem, a binary tournament strategy is firstly used for mating selection. Next, the ideal point is updated by the offspring. Meanwhile, an adaptive PBI method is implemented to update the population. After a loop for all subproblems, the nadir point is re-estimated by the current population P and archive P^{nad}. Finally, the weight vectors are adjusted by an adaptive strategy. This process repeats until the stopping criterion is met.

In what follows, we introduce the adaptive PBI selection strategy and the weight vector adaptation in detail, which are the four important components in ADEA, aw well as the mating selection and variation.



(a) Increasing θ (b) Decreasing θ

Fig. 2. Illustration of the strategy for adjustment of θ in a 2-dimensional objective space.

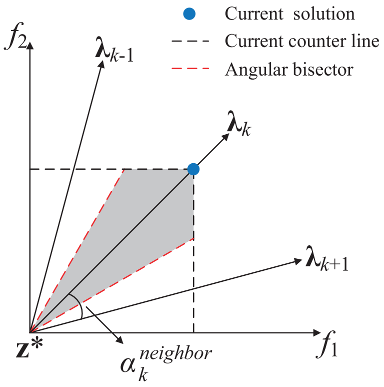


Fig. 3. Illustration of possible improvement region (shaded area) of PBI.

3.2. The adaptive PBI selection

The environmental selection is of great importance to optimize a subproblem, which aims at the convergence-diversity balance. In classical decomposition-based approaches, each newly offspring can replace the solution of the currently focused subproblem and its neighbors. However, in some cases, the offspring may be poor for the current subproblem and its neighbors, but good for other subproblems. This will waste some computational resources and slow down the search progress. To ensure that the population always consists of the best solutions ever found, we first associate the new offspring with a most suitable subproblem before the population replacement. The most suitable subproblem means that its corresponding weight vector is closest to the offspring individual based on the angle between them. The angle α_i between offspring $\mathbf{x}_{off}$ and the weight vector $\boldsymbol{\lambda}_i$ can be calculated as:

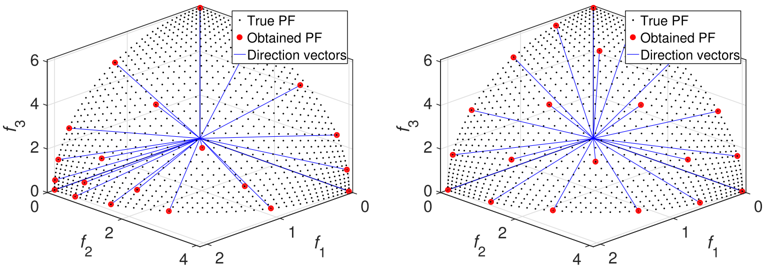
$$\alpha_i = \arccos \frac{(F(\mathbf{x}_{off}) - \mathbf{z}^*) \cdot \boldsymbol{\lambda}_i}{\|(F(\mathbf{x}_{off}) - \mathbf{z}^*)\|} \tag{4}$$

Then the index k of the associated subproblem is

$$k = \arg\min_{i \in \{1,\ldots,N\}} \alpha_j \tag{5}$$

Once the association is completed, an adaptive PBI selection is implemented for the associated subproblem k. To maintain the diversity, only the associated subproblem has the chance to update its solution. As mentioned before, the penalty factor θ in PBI is crucial for the performance of a penalty-based method. Based on the property of counter line, a larger θ encourages the diversity, whereas a smaller one enhances the convergence. For a decomposition-based approach, the quality of solutions mainly manifests in two aspects: one is the distance to $\mathbf{z}^*$, referred to as convergence, and the other is the distance to its corresponding weight vector, referred to as diversity. It is desirable to decrease these two distances as small as possible. Therefore, the value of θ is supposed to be guided by the distribution information of the population.

Fig. 2 illustrates the adjustment strategy of θ in a 2-dimensional objective space. When the current solution of a subproblem k is far from its weight vector, appropriately increasing θ can narrow the counter line to push the search direction towards the weight vector. On the contrary, it is favorable to decrease θ to give the priority to new solutions that are closer to $\mathbf{z}^*$, if the current one shows a good diversity. On the other hand, the possible improvement region of PBI is directly related to the distribution of the weight vectors. As Fig. 3 shows, the wider the angle between the focused weight vector



(a) In the case of uniformly distributed weight vectors.

(b) In the case of adaptive weight vectors.

Fig. 4. Illustration of the Pareto optimal solutions determined by different weight vectors on 3-objective WFG4.

and its neighbors is, the larger the improvement region will be. Thus, the setting of θ is determined by the distribution of both population and weight vectors. In addition, the number of objectives M should also be taken into consideration since it influences the population sparsity [2]. Consequently, an angle-based dynamic penalty factor adaptation strategy is proposed as follows:

$$\theta_k = K \cdot M \cdot \left(\alpha_k^{ind} + \alpha_k^{neighbor}\right) \tag{6}$$

where α_k^{ind} is the angle between the weight vector k and the current solution $\mathbf{x}_k$, $\alpha_k^{neighbor}$ is the angle between the weight vector k and the one which is closest to it. Here, K is a scaling parameter. Since the angle is naturally normalized into the range [0°, 90°], we use it to measure the diversity and sparsity of the weight vectors instead of the distance. Noting that at each generation, the proposed strategy can provide a particular θ for a particular subproblem.

According to the obtained penalty factor, the solution with a lower g^{PBI} is selected. Algorithm 2 gives the pseudo code of the adaptive PBI selection.

Algorithm 2 Adaptive PBI Selection.

```
Require: Weight vector $\Lambda$, population $P$, offspring $y$
/*Association*/
for $i = 1$, to, $N$ do
    $\alpha_i = \arccos \frac{(F(y)-\mathbf{z}^*)\cdot\boldsymbol{\lambda}_i}{\|(F(y)-\mathbf{z}^*)\|}$
end for
$k = \arg\min_{j\in\{1,\ldots,N\}} \alpha_j$
/*Adaptive PBI*/
$\alpha_k^{ind} = \arccos \frac{(F(\mathbf{x}_k)-\mathbf{z}^*)\cdot\boldsymbol{\lambda}_k}{\|(F(\mathbf{x}_k)-\mathbf{z}^*)\|}$
$\alpha_k^{neighbor} = \min_{i\in\{1,\ldots,N\},i\neq k} \langle\lambda_k, \lambda_i\rangle$
$\theta_k = K \cdot M \cdot \left(\alpha_k^{ind} + \alpha_k^{neighbor}\right)$
if $g^{\mathrm{PBI}}(y|\boldsymbol{\lambda}_k, \mathbf{z}^*) < g^{\mathrm{PBI}}(\mathbf{x}_k|\boldsymbol{\lambda}_k, \mathbf{z}^*)$ then
    $\mathbf{x}_k = y$
end if
return $P$
```

3.3. The weight vector adaptation

To guarantee the uniformly distribution of solutions, the weight vectors need to be chosen carefully since the approximated front consists of intersections of the weight vectors and the true PF. A popular assumption is that uniformly distributed weight vectors can produce uniformly distributed solutions [45]. In fact, it only stands if the scale of each objective stays the same. When dealing with the disparately scaled objectives existing in considerable MaOPs, e.g., the WFG test suite [15], the uniformity of the weight vectors cannot guarantee the uniformity of solutions, as Fig. 4(a) shows.

Accordingly, we adopt a weight vector adaptation strategy in ADEA for solving problems with different scales as follows:

$$\boldsymbol{\lambda}_i = \left(\mathbf{z}^{nad} - \mathbf{z}^*\right) \circ \boldsymbol{\lambda}_{0,i} \tag{7}$$

where $i = 1, \dots, N$, $\boldsymbol{\lambda}_i$ and $\boldsymbol{\lambda}_{0,i}$ are the i-th adapted weight vector and the uniformly distributed weight vector, respectively. The $\circ$ operator is the Hadamard product. The difference between $\mathbf{z}^{nad}$ and $\mathbf{z}^*$ is used to estimate the value range of each objective. Based on the above weight vector adaptation strategy, ADEA is able to handle the scaled problems, as shown in Fig. 4(b).

Specifically, as suggested in [2], the weight vector adaptation should be periodically implemented to preserve the convergence stability. The adaptation period is controlled by a parameter t, referring to line 2 in Algorithm 3. The smaller the t value is, the higher the frequency of weight vector adaptation will be.

Algorithm 3 Weight Vector Adaptation.

```
Require: Weight vector set $\Lambda = \{\boldsymbol{\lambda}_1, \dots, \boldsymbol{\lambda}_N\}$ and $\Lambda_0 = \{\boldsymbol{\lambda}_{0,1}, \dots, \boldsymbol{\lambda}_{0,N}\}$, ideal point $\mathbf{z}^*$, nadir point $\mathbf{z}^{nad}$, population $P$, archive $P^{nad}$
 1: $[\mathbf{z}^{nad}, P^{nad}] = \text{Calculate\_Nadir\_Point}\ (P, P^{nad})$
 2: if $Gen \bmod (MaxGen \cdot t) == 0$ then
 3:     for $i = 1$, to, $N$ do
 4:         $\boldsymbol{\lambda}_i = \left(\mathbf{z}^{nad} - \mathbf{z}^*\right) \circ \boldsymbol{\lambda}_{0,i}$
 5:     end for
 6:     for $i = 1$, to, $N$ do
 7:         calculate the neighborhood $B(i) = \{i_1, \dots, i_T\}$
 8:     end for
 9:     $P^{nad} = \Phi$
10: end if
11: return $\Lambda$ and $P^{nad}$
```

The procedure of weight vector adaption is similar to that in RVEA but different in estimation of the objective value range. The range estimation completely depends on the identification of the ideal and nadir points, which indicate the lower and upper limits for each objective. Here, the ideal point contains the best value ever found on each objective. As to the nadir point, in order to enhance its estimation accuracy, an archive P^{nad} is used to preserve the solutions which compose the nadir point. At each generation, an efficient non-dominated sort [46] is applied to the union of P^{nad} and population P. Then, the nadir point can be estimated as the maximum value of each objective of solutions in the first non-dominated front, and P^{nad} is updated as well, as described in Algorithm 4. Once the weight vector is adjusted, the archive P^{nad} is emptied for the later estimations of the nadir point.

Algorithm 4 Estimate Nadir Point.

```
Require: Population $P$, archive $P^{nad}$
$Q = (P \cup P^{nad})$
$F_1 = Non\_Dominated\_Sort(Q)$
$P^{nad} = \Phi$
for $i = 1$, to, $M$ do
    $z_i^{nad} = \max_{p \in F_1} f_i(p)$
    $P^{nad} = P^{nad} \cup (p : \arg\min_{p \in F_1} f_i(p))$
end for
return $\mathbf{z}^{nad}$ and $P^{nad}$
```

3.4. The mating selection and variation

The purpose of the mating selection is to select the more appropriate parent individuals, so that the generated offspring can have a better performance of convergence with a large possibility. Then the individuals possessing high promising convergence performance are preferred to be selected for variation. Here, the convergence performance for a individual is firstly evaluated based on the Pareto dominance criterion. However, most of the time, the dominance comparison cannot differentiate the convergence of individuals. Since the counter line analysis and experiment results [17] indicate that WS method results in a better convergence performance over other aggregation methods, an alternative criterion based on the WS value is used when Pareto dominance criterion is invalid.

In addition, the distance between each individual in a high-dimensional space is often very large. In this case, offspring individuals are probably far away from its parents which might deteriorate the efficiency of offspring generation [21]. Therefore, mating selection for each subproblem is implemented within its neighborhood to avoid a larger distance between two parent individuals.

In ADEA, the binary tournament selection is adopted to obtain two mating parent individuals. Algorithm 2 presents the pseudo code of binary tournament mating selection. For a subproblem i, two candidate individuals are first randomly chosen

Algorithm 5 Mating Selection.

```
Require: Weight vector $\boldsymbol{\lambda}_i$, corresponding neighborhood $B(i)$
$P^* \leftarrow \Phi$
while $|P^*| < 2$ do
    randomly choose two indices $j$, $k$ from $B(i)$
    if $\mathbf{x}_j \prec \mathbf{x}_k$ then
        $P^* = P^* \cup \{\mathbf{x}_j\}$
    else if $\mathbf{x}_k \prec \mathbf{x}_j$ then
        $P^* = P^* \cup \{\mathbf{x}_k\}$
    else if $g^{\mathrm{WS}}(\mathbf{x}_j|\boldsymbol{\lambda}_i) < g^{\mathrm{WS}}(\mathbf{x}_k|\boldsymbol{\lambda}_i)$ then
        $P^* = P^* \cup \{\mathbf{x}_j\}$
    else if $g^{\mathrm{WS}}(\mathbf{x}_j|\boldsymbol{\lambda}_i) > g^{\mathrm{WS}}(\mathbf{x}_k|\boldsymbol{\lambda}_i)$ then
        $P^* = P^* \cup \{\mathbf{x}_k\}$
    else if $rand < 0.5$ then
        $P^* = P^* \cup \{\mathbf{x}_j\}$
    else
        $P^* = P^* \cup \{\mathbf{x}_k\}$
    end if
end while
return $P^*$
```

Table 1
The parameter settings and the characteristics of the test instances.

Problem	M	D	Parameter	Characteristics
DTLZ1	3,6,8,10	$M+K-1$	$K=5$	Multi-modal, linear
DTLZ2	3,6,8,10	$M+K-1$	$K=10$	Uni-modal, concave
DTLZ3	3,6,8,10	$M+K-1$	$K=10$	Multi-modal, concave
DTLZ4	3,6,8,10	$M+K-1$	$K=10$	Uni-modal, concave, biased
WFG1	3,6,8,10	$K+L$	$K=M-1, L=10$	Uni-modal, mixed, biased, scaled
WFG4	3,6,8,10	$K+L$	$K=M-1, L=10$	Multi-modal, concave, scaled
WFG5	3,6,8,10	$K+L$	$K=M-1, L=10$	Deceptive, concave, scaled
WFG6	3,6,8,10	$K+L$	$K=M-1, L=10$	Uni-modal, concave, scaled, non-separable
WFG7	3,6,8,10	$K+L$	$K=M-1, L=10$	Uni-modal, concave, biased, scaled
WFG8	3,6,8,10	$K+L$	$K=M-1, L=10$	Uni-modal, concave, biased, scaled, non-separable
WFG9	3,6,8,10	$K+L$	$K=M-1, L=10$	Multi-modal, deceptive, concave, biased, scaled, non-separable

from the neighborhood of the corresponding weight vector $\boldsymbol{\lambda}_i$. If one dominates the other, the former is chosen. When both individuals are non-dominated with each other, we prefer the one with a lower WS value on $\boldsymbol{\lambda}_i$. The WS value can be computed as below:

$$g^{\mathrm{WS}}\left(\mathbf{x}|\boldsymbol{\lambda}_i\right) = \sum_{j=1}^{M} \lambda_i^j \cdot f_j(\mathbf{x}) \tag{8}$$

If two individuals are on equal WS value, we randomly select one of them.

When the mating selection completes, the simulated binary crossover [5] and polynomial mutation [6] are executed to produce a new offspring, which are suggested in many powerful MOEAs, such as, NSGA-III [7] and MOEA/DD [21].

4. Experimental studies

4.1. Experimental design

4.1.1. Benchmark problem

DTLZ 1–4 from the DTLZ test suite [9] and WFG1, WFG4-9 from the WFG test suite [15] are selected as the test instances. The number of objectives $M \subseteq \{3, 6, 8, 10\}$. As suggested in [9] and [15], the detail description of the problems are given in Table 1. However, for DTLZ5-7 and WFG2-3 with disconnected or degenerate Pareto optimal fronts, specific weight vector generation and adjustment methods are required and will be investigated in our future work.

4.1.2. Performance indicators

Inverted generational distance (IGD) [50] and hypervolume (HV) [10] are adopted as the performance indicators in our experimental study. A smaller IGD value and a larger HV value indicate a better performance of the algorithm.

Table 2
Settings of the population size in ADEA, MOEA/D, MOEA/DD, NSGA-III and RVEA. H_1 and H_2 are parameters for generating uniformly distributed reference vectors on the outer boundaries and the inside layers, respectively.

M	Parameter (H_1, H_2)	Population size N
3	13, 0	105
6	4, 1	132
8	3, 2	156
10	3,2	275

IGD measures the average distance from the obtained front to reference front, which is calculated as

$$IGD(P, P^*) = \frac{\sum_{\mathbf{x} \in P^*} \min d(\mathbf{x}, P)}{|P^*|} \tag{9}$$

where P is the approximated front obtained by a given MOEA, P^* is the reference point set which is uniformly sampled along the true PF, and $d(x, P)$ is the Euclidean distances between x and the points in P. If the size of P^* is large enough, IGD is able to make a comprehensive evaluation on both convergence and diversity. According to [35], 500,000 reference points are sampled from the true PF following the approach proposed in [7,30].

HV is the volume of the region which is dominated by the obtained approximated front P. Given a reference point $\mathbf{r} = r_1, \ldots, r_M$, the HV value can be calculated as

$$HV(P, \mathbf{r}) = Lebesgue\left(\underset{\mathbf{x} \in P}{\cup} [f_1(\mathbf{x}), r_1] \times \cdots \times [f_M(\mathbf{x}), r_M] \right) \tag{10}$$

In the experiments, the reference point is set as M-dimensional vector of ones. Considering the objective values have disparate scales, the objective values of all test instances are normalized to by the $1.1\mathbf{z}^{nad}$ of the true PF. For enhancing computational efficiency, the HV value is approximated by the Monte Carlo method with 1,000,000 sampling points when the objective number is more than three.

4.1.3. Algorithms under comparison

To study the performance of ADEA, seven state-of-the-art algorithms are chosen for comparison, namely, MOEA/D-PBI [45], KnEA [47], NSGA-III, RVEA [2], MOEA/DD, PICEA-g [37] and Two_Arch2 [35]. MOEA/D-PBI is a classical decomposition-based MOEA. KnEA adopts a similar framework as NSGA-II [4], and prefers the knee points in the environmental selection. NSGA-III and RVEA are two reference points (or vectors) assisted algorithms, where the reference points (or vectors) are utilized to manage the diversity and guide the searching direction, respectively. MOEA/DD integrates the superiority of both decomposition and dominance-based approaches. PICEA-g is a preference-inspired algorithm where the preferences are co-evolved together with the population. Two_Arch2 is a two-archive algorithm that relies on two archives to promote convergence and diversity separately.

4.1.4. Parameter settings

Table 2 lists the population size settings for ADEA, NSGA-III, RVEA, MOEA/D-PBI and MOEA/DD. The evenly distributed weight vectors are generated by a two-layered reference points strategy, as recommended in [7]. For fair comparison, the same population size is adopted for other rivals.

The simulated binary crossover and polynomial mutation is implemented to produce new offsprings in all algorithms. For the simulated binary crossover, the distribution index is set to $\eta_c = 30$ in ADEA, RVEA, NSGA-III and MOEA/DD, $\eta_c = 15$ in Two_Arch2, $\eta_c = 20$ in the rest two algorithms, and the crossover probability is set to $p_c = 1.0$ in all algorithms. For the polynomial mutation, the distribution index is set to $\eta_m = 20$, and the mutation probability is set to $p_m = 1/N$.

In ADEA, the scaling parameter is set to $K = 0.06$ and frequency control parameter is set to $t = 0.2$. The neighborhood size in ADEA and MOEA/D-PBI is set to $0.1N$. For other specific parameters in each algorithm, we use the same settings as suggested in the references.

For statistical comparisons, each algorithm is run 30 times independently on each test instance. The stopping criterion is the maximum number of generations, which is set to 1000 for all test instances. Wilcoxon's rank sum test at a 5% significance level is used to compare the indicator evaluation results.

4.2. Comparative results

The mean and standard deviation of indicator evaluation results of the all MOEAs on each test instance are summarised in Tables 3 and 4, where the highlight is the best results on the mean values. The symbols " $+$ ", " $=$ ", and " $-$ " indicate that ADEA performs significantly better than, similar to, and worse than the peer algorithm, respectively. The bottom of Tables 3 and 4 shows the summary of statistical results.

Table 3
Comparison of the IGD values of the eight algorithms on all test instances. The best result on each test instance is highlighted. " + ", " = ", and " − " denote that ADEA performs significantly better than, similar to, and worse than the compared algorithm, respectively.

Problem	M	ADEA	MOEA/D-PBI	MOEA/DD	KnEA	NSGA-III	PICEA-g	RVEA	Two_Arch2
DTLZ1	3	**1.87e-2(4.21e-5)**	1.90e-2(1.54e-5)+	1.90e-2(3.81e-6)+	4.50e-2(2.44e-2)+	1.91e-2(1.95e-5)+	2.34e-1(1.07e-2)+	1.91e-2(6.25e-5)+	2.06e-2(3.07e-4)+
	6	8.14e-2(2.64e-4)	8.40e-2(1.53e-4)+	8.39e-2(7.33e-5)+	2.41e-1(3.55e-2)+	8.53e-2(9.21e-3)+	3.69e-1(3.20e-2)+	8.13e-2(1.58e-4)=	**7.59e-2(7.24e-4)–**
	8	**9.74e-2(1.65e-4)**	1.21e-1(2.12e-4)+	1.21e-1(3.63e-5)+	3.60e-1(5.26e-2)+	1.22e-1(4.07e-3)+	4.17e-1(2.54e-2)+	9.97e-2(1.35e-2)+	1.00e-1(1.18e-3)+
	10	1.17e-1(1.39e-4)	1.33e-1(7.00e-5)+	1.33e-1(8.06e-5)+	3.93e-1(7.11e-2)+	1.48e-1(3.62e-2)+	3.92e-1(3.00e-2)+	1.17e-1(3.62e-4)−	**1.15e-1(9.37e-4)–**
DTLZ2	3	5.03e-2(2.49e-6)	**5.03e-2(5.38e-7)–**	5.03e-2(9.82e-7)=	6.63e-2(3.14e-3)+	5.04e-2(1.70e-5)+	1.06e-1(7.85e-3)+	5.05e-2(2.09e-5)+	5.69e-2(1.34e-3)+
	6	2.57e-1(4.11e-4)	2.62e-1(3.41e-6)+	2.62e-1(3.65e-6)+	2.82e-1(6.31e-3)+	2.61e-1(2.93e-4)+	3.65e-1(4.20e-2)+	**2.57e-1(2.07e-4)=**	2.65e-1(3.10e-3)+
	8	**3.17e-1(3.30e-4)**	3.59e-1(3.02e-6)+	3.59e-1(2.48e-6)+	3.83e-1(5.18e-3)+	3.80e-1(7.07e-2)+	5.55e-1(6.26e-2)+	3.18e-1(3.52e-4)+	3.83e-1(4.15e-3)+
	10	4.24e-1(6.90e-4)	4.29e-1(7.68e-6)+	4.29e-1(7.02e-6)+	**4.03e-1(4.87e-3)–**	4.74e-1(7.78e-2)+	5.41e-1(6.83e-2)+	4.20e-1(4.14e-4)−	4.76e-1(5.38e-3)+
DTLZ3	3	**5.07e-2(4.60e-4)**	5.47e-2(3.26e-4)+	5.45e-2(7.24e-5)+	1.00e-1(5.13e-2)+	5.08e-2(4.26e-4)=	4.66e-1(1.20e-2)+	5.41e-2(5.88e-3)+	5.61e-2(1.31e-3)+
	6	**2.59e-1(8.07e-4)**	2.62e-1(3.16e-4)+	2.62e-1(3.12e-4)+	1.00e+0(1.81e-1)+	2.80e-1(9.13e-2)+	8.17e-1(5.10e-2)+	2.60e-1(2.30e-3)+	2.62e-1(6.46e-3)+
	8	**3.19e-1(3.26e-3)**	4.32e-1(2.24e-1)+	3.59e-1(2.31e-4)+	1.09e+0(1.35e-1)+	4.88e-1(2.92e-1)+	9.77e-1(5.26e-2)+	3.20e-1(1.31e-3)+	3.86e-1(1.18e-2)+
	10	4.23e-1(2.38e-3)	4.51e-1(1.19e-1)+	4.29e-1(1.37e-4)+	1.13e+0(1.13e-1)+	1.00e+0(1.50e+0)+	1.03e+0(5.28e-2)+	**4.22e-1(1.00e-3)=**	4.50e-1(9.08e-3)+
DTLZ4	3	**5.03e-2(4.40e-6)**	2.45e-1(2.68e-1)=	8.32e-2(1.25e-1)+	6.44e-2(2.84e-3)+	1.32e-1(1.86e-1)+	2.74e-1(2.47e-1)+	5.06e-2(3.78e-5)+	8.49e-2(1.63e-1)+
	6	**2.58e-1(8.39e-4)**	5.50e-1(1.77e-1)+	2.62e-1(4.46e-6)+	2.80e-1(5.73e-3)+	2.91e-1(8.01e-2)+	4.18e-1(1.06e-1)+	2.59e-1(4.62e-4)+	2.69e-1(2.44e-3)+
	8	**3.18e-1(5.81e-4)**	8.11e-1(1.07e-1)+	3.67e-1(1.89e-2)+	4.42e-1(2.56e-2)+	4.28e-1(9.95e-2)+	4.76e-1(8.12e-2)+	3.24e-1(2.22e-2)+	3.88e-1(4.92e-3)+
	10	4.24e-1(7.87e-4)	6.00e-1(8.06e-2)+	4.29e-1(4.67e-6)+	**4.04e-1(1.44e-2)–**	4.40e-1(4.47e-2)+	4.71e-1(2.49e-2)+	4.20e-1(8.75e-4)−	4.81e-1(7.59e-3)+
WFG1	3	4.02e-1(3.90e-2)	2.87e-1(1.31e-2)−	1.92e-1(2.27e-2)−	1.95e-1(2.70e-2)−	**1.39e-1(8.22e-3)–**	2.58e-1(3.45e-2)−	3.51e-1(5.52e-2)−	1.42e-1(5.71e-3)−
	6	2.40e+0(1.25e-1)	1.90e+0(6.92e-2)−	1.29e+0(9.38e-2)−	6.58e-1(3.10e-2)−	6.87e-1(2.21e-2)−	2.26e+0(2.53e-1)−	9.37e-1(1.34e-1)−	**5.80e-1(1.59e-2)–**
	8	2.72e+0(1.82e-1)	2.53e+0(1.03e-1)−	2.35e+0(3.33e-2)−	9.58e-1(1.35e-1)−	1.00e+0(8.15e-2)−	2.53e+0(4.55e-2)−	1.26e+0(1.18e-1)−	**8.69e-1(2.60e-2)–**
	10	3.22e+0(1.77e-1)	3.14e+0(8.97e-2)−	3.07e+0(2.21e-2)−	1.28e+0(1.33e-1)−	1.21e+0(9.08e-2)−	2.62e+0(3.50e-1)−	1.72e+0(1.15e-1)−	**1.21e+0(2.35e-2)–**
WFG4	3	**2.03e-1(2.66e-4)**	2.25e-1(1.25e-3)+	2.22e-1(4.69e-4)+	2.48e-1(8.20e-3)+	2.05e-1(8.31e-5)+	3.49e-1(2.33e-2)+	2.10e-1(8.73e-4)+	2.18e-1(4.84e-3)+
	6	1.74e+0(8.06e-3)	3.13e+0(2.13e-1)+	1.92e+0(6.41e-4)+	1.92e+0(2.88e-2)+	1.81e+0(6.08e-2)+	2.22e+0(5.44e-2)+	1.76e+0(2.28e-3)+	**1.73e+0(1.83e-2)–**
	8	**2.97e+0(9.62e-3)**	5.56e+0(1.67e-1)+	3.54e+0(6.56e-2)+	3.43e+0(4.79e-2)+	3.40e+0(4.89e-2)+	4.40e+0(6.71e-1)+	2.99e+0(1.28e-2)+	3.14e+0(3.26e-2)+
	10	4.23e+0(5.80e-2)	7.43e+0(1.70e-1)+	5.34e+0(8.65e-2)+	**4.15e+0(3.72e-2)–**	4.34e+0(5.87e-2)+	6.12e+0(9.50e-1)+	4.25e+0(3.40e-2)=	4.45e+0(5.71e-2)+
WFG5	3	**2.14e-1(3.38e-4)**	2.28e-1(1.31e-3)+	2.28e-1(6.09e-4)+	2.58e-1(8.89e-3)+	2.16e-1(4.88e-5)+	3.82e-1(2.62e-2)+	2.18e-1(7.14e-4)+	2.30e-1(5.12e-3)+
	6	1.73e+0(4.82e-3)	2.84e+0(1.46e-1)+	1.89e+0(4.27e-3)+	1.88e+0(3.74e-2)+	1.78e+0(3.41e-3)+	2.21e+0(7.13e-2)+	1.74e+0(1.14e-3)+	**1.72e+0(1.41e-2)–**
	8	**2.94e+0(3.66e-3)**	5.23e+0(1.15e-1)+	3.56e+0(4.25e-2)+	3.34e+0(2.57e-2)+	3.37e+0(1.23e-2)+	3.69e+0(2.12e-1)+	2.97e+0(1.49e-2)+	3.08e+0(2.90e-2)+
	10	4.24e+0(1.76e-2)	7.18e+0(7.60e-2)+	5.43e+0(7.98e-2)+	**4.16e+0(4.50e-2)–**	4.38e+0(8.80e-3)+	4.97e+0(2.75e-1)+	4.28e+0(4.56e-2)+	4.39e+0(5.15e-2)+
WFG6	3	**2.20e-1(1.11e-2)**	2.48e-1(1.25e-2)+	2.35e-1(9.48e-3)+	2.86e-1(1.03e-2)+	2.25e-1(8.55e-3)+	4.05e-1(2.93e-2)+	2.31e-1(8.50e-3)+	2.38e-1(1.11e-2)+
	6	**1.73e+0(5.62e-3)**	3.78e+0(8.49e-2)+	1.89e+0(1.18e-2)+	1.94e+0(4.21e-2)+	1.79e+0(5.53e-3)+	2.41e+0(3.39e-1)+	1.75e+0(3.22e-3)+	1.74e+0(2.02e-2)+
	8	**2.96e+0(1.08e-2)**	6.21e+0(3.33e-1)+	3.55e+0(8.72e-2)+	3.50e+0(7.46e-2)+	3.37e+0(1.18e-2)+	5.81e+0(1.00e+0)+	2.98e+0(1.02e-2)+	3.12e+0(3.30e-2)+
	10	**4.23e+0(2.92e-2)**	7.82e+0(3.72e-1)+	5.41e+0(1.32e-1)+	4.25e+0(3.37e-2)+	4.40e+0(1.19e-2)+	6.16e+0(9.29e-1)+	4.30e+0(3.84e-2)+	4.41e+0(6.53e-2)+
WFG7	3	**2.04e-1(3.02e-4)**	2.42e-1(7.64e-3)+	2.23e-1(8.64e-4)+	2.42e-1(1.01e-2)+	2.06e-1(1.07e-4)+	4.50e-1(3.94e-2)+	2.07e-1(1.70e-4)+	2.19e-1(5.25e-3)+
	6	1.74e+0(6.10e-3)	3.91e+0(5.88e-2)+	1.92e+0(6.31e-3)+	1.93e+0(3.32e-2)+	1.80e+0(3.13e-3)+	2.33e+0(7.03e-2)+	1.77e+0(3.21e-3)+	**1.71e+0(1.61e-2)–**
	8	**2.97e+0(9.74e-3)**	5.89e+0(1.40e-1)+	3.48e+0(5.74e-2)+	3.34e+0(4.49e-2)+	3.41e+0(1.34e-2)+	3.77e+0(2.19e-1)+	3.02e+0(1.88e-2)+	3.11e+00(3.73e-2)+
	10	4.27e+0(3.09e-2)	7.65e+0(7.54e-2)+	5.11e+0(1.06e-1)+	**3.89e+0(5.42e-2)–**	4.38e+0(5.78e-2)+	4.80e+0(1.47e-1)+	4.24e+0(2.41e-2)−	4.32e+00(4.74e-2)+
WFG8	3	**2.55e-1(1.38e-3)**	2.81e-1(3.90e-3)+	2.75e-1(1.85e-3)+	3.37e-1(1.15e-2)+	2.69e-1(4.20e-3)+	5.45e-1(6.34e-2)+	2.83e-1(4.73e-3)+	2.95e-1(4.56e-3)+
	6	**1.73e+0(1.01e-2)**	3.38e+0(3.12e-1)+	1.88e+0(4.45e-3)+	1.92e+0(2.12e-2)+	1.75e+0(8.38e-2)+	2.34e+0(1.02e-1)+	1.74e+0(6.64e-3)+	1.93e+0(1.86e-2)+
	8	**3.04e+0(4.09e-2)**	5.76e+0(4.60e-1)+	3.32e+0(8.30e-2)+	3.55e+0(4.84e-2)+	3.42e+0(3.56e-1)+	4.85e+0(2.29e-1)+	3.05e+0(3.43e-2)=	3.51e+0(3.90e-2)+
	10	4.34e+0(1.33e-2)	7.68e+0(5.43e-1)+	5.20e+0(4.27e-1)+	4.38e+0(3.96e-2)+	4.56e+0(5.16e-1)=	6.28e+0(3.60e-1)+	**4.31e+0(9.33e-2)=**	4.99e+0(1.28e-1)+
WFG9	3	**2.01e-1(7.77e-4)**	2.35e-1(2.54e-2)+	2.24e-1(2.25e-2)+	2.29e-1(2.26e-2)+	2.11e-1(2.38e-2)+	3.66e-1(2.89e-2)+	2.06e-1(1.26e-3)+	2.15e-1(4.01e-3)+
	6	1.71e+0(9.73e-3)	3.98e+0(1.33e-1)+	1.90e+0(2.72e-2)+	**1.69e+0(2.01e-2)–**	1.70e+0(2.39e-2)=	2.19e+00(5.11e-2)+	1.72e+0(4.95e-3)+	1.70e+0(1.42e-2)−
	8	**2.93e+0(1.16e-2)**	6.90e+0(4.34e-1)+	3.69e+0(1.43e-1)+	3.07e+0(3.86e-2)+	3.10e+0(1.52e-1)+	3.69e+00(4.06e-1)+	2.96e+0(1.16e-2)+	3.15e+0(3.21e-2)+
	10	4.25e+0(2.86e-2)	9.23e+0(1.65e-1)+	5.03e+0(1.97e-1)+	**4.12e+0(9.85e-2)–**	4.52e+0(1.46e-1)+	4.99e+00(3.90e-1)+	4.18e+0(4.26e-2)−	5.19e+0(6.68e-1)+
+/=/−		-	38/1/5	39/1/4	33/0/11	37/3/4	40/0/4	29/6/9	34/0/10

Table 4
Comparison of the HV values of the eight algorithms on all test instances. The best result on each test instance is highlighted in boldface. “ + ”, “ = ”, and “ − ” denote that ADEA performs significantly better than, similar to, and worse than the compared algorithm, respectively.

Problem	M	ADEA	MOEA/D-PBI	MOEA/DD	KnEA	NSGA-III	PICEA-g	RVEA	Two_Arch2
DTLZ1	3	8.43e-1(3.20e-4)	8.44e-1(2.05e-4)−	**8.44e-1(5.72e-5)−**	7.60e-1(5.42e-2)+	8.43e-1(2.56e-4)=	2.96e-1(3.77e-2)+	8.43e-1(6.36e-4)=	8.42e-1(4.37e-4)+
	6	**9.89e-1(1.53e-4)**	9.88e-1(1.38e-4)+	9.88e-1(8.90e-5)+	5.75e-1(9.14e-2)+	9.86e-1(3.91e-3)+	3.56e-1(9.62e-2)+	9.88e-1(1.04e-4)+	9.81e-1(7.57e-4)+
	8	**9.98e-1(6.89e-5)**	9.97e-1(5.69e-5)+	9.97e-1(4.27e-5)+	3.83e-1(1.12e-1)+	9.96e-1(1.07e-3)+	3.08e-1(8.28e-2)+	9.96e-1(1.83e-3)+	9.91e-1(7.36e-4)+
	10	**1.00e+0(2.27e-5)**	1.00e+00(2.33e-5)+	1.00e+00(1.71e-5)+	4.06e-1(2.00e-1)+	9.88e-1(3.42e-2)+	4.79e-1(1.18e-1)+	9.99e-1(1.94e-5)+	9.96e-1(2.41e-4)+
DTLZ2	3	5.62e-1(8.41e-5)	5.63e-1(7.99e-6)−	**5.63e-1(5.60e-6)−**	5.44e-1(3.56e-3)+	5.62e-1(2.20e-5)+	5.33e-1(6.14e-3)+	5.61e-1(2.45e-4)+	5.61e-1(9.33e-4)+
	6	8.52e-1(4.14e-4)	8.54e-1(2.82e-4)−	**8.54e-1(2.96e-4)−**	8.26e-1(7.81e-3)+	8.54e-1(5.03e-4)−	8.00e-1(2.09e-2)+	8.52e-1(4.77e-4)+	7.65e-1(7.98e-3)+
	8	9.20e-1(5.45e-4)	**9.36e-1(2.51e-4)−**	9.36e-1(1.96e-4)−	8.83e-1(5.89e-3)+	9.20e-1(4.65e-2)−	8.26e-1(3.88e-2)+	9.21e-1(3.97e-4)−	7.49e-1(1.70e-2)+
	10	9.66e-1(4.83e-4)	9.75e-1(1.56e-4)−	**9.75e-1(1.39e-4)−**	9.52e-1(6.48e-3)+	9.47e-1(4.62e-2)+	9.22e-1(3.24e-2)+	9.68e-1(2.81e-4)−	7.50e-1(1.30e-2)+
DTLZ3	3	5.57e-1(3.16e-3)	5.56e-1(2.98e-3)+	5.58e-1(1.33e-3)=	5.08e-1(4.17e-2)+	5.57e-1(3.38e-3)=	2.48e-1(2.71e-3)+	5.42e-1(1.49e-2)+	**5.60e-1(2.81e-3)−**
	6	8.48e-1(3.47e-3)	**8.53e-1(1.73e-3)−**	8.52e-1(1.84e-3)−	1.56e-1(8.36e-2)+	8.27e-1(1.07e-1)=	2.79e-1(4.11e-2)+	8.42e-1(5.67e-3)+	7.95e-1(8.25e-3)+
	8	9.14e-1(5.05e-3)	8.53e-1(2.52e-1)+	**9.35e-1(6.38e-4)−**	1.40e-1(6.07e-2)+	7.92e-1(3.03e-1)+	2.34e-1(4.36e-2)+	9.15e-1(3.59e-3)=	8.18e-1(1.49e-2)+
	10	9.65e-1(2.35e-3)	9.48e-1(1.48e-1)+	**9.75e-1(1.84e-4)−**	1.53e-1(6.31e-2)+	7.66e-1(3.66e-1)=	2.48e-1(4.40e-2)+	9.65e-1(1.01e-3)=	8.64e-1(1.21e-2)+
DTLZ4	3	**5.62e-1(1.27e-4)**	4.72e-1(1.26e-1)=	5.47e-1(6.03e-2)+	5.46e-1(2.79e-3)+	5.25e-1(8.42e-2)+	4.55e-1(1.18e-1)+	5.60e-1(3.99e-4)+	5.46e-1(8.60e-2)=
	6	8.52e-1(5.58e-4)	6.88e-1(1.51e-1)+	**8.54e-1(3.49e-4)−**	8.36e-1(4.67e-3)+	8.35e-1(4.84e-2)+	7.77e-1(6.55e-2)+	8.51e-1(6.58e-4)+	7.28e-1(1.10e-2)+
	8	9.19e-1(8.54e-4)	5.97e-1(1.24e-1)+	**9.22e-1(7.03e-3)−**	8.60e-1(2.83e-2)+	8.91e-1(6.44e-2)=	8.74e-1(4.68e-2)+	9.19e-1(5.41e-3)−	7.33e-1(1.66e-2)+
	10	9.66e-1(4.56e-4)	9.04e-1(4.90e-2)+	**9.75e-1(1.37e-4)−**	9.57e-1(2.28e-3)+	9.67e-1(2.73e-2)−	9.52e-1(8.28e-3)+	9.68e-1(4.00e-4)−	7.22e-1(1.75e-2)+
WFG1	3	8.67e-1(3.38e-2)	9.07e-1(8.63e-3)−	9.36e-1(3.68e-3)−	9.30e-1(3.72e-3)−	9.46e-1(1.16e-3)−	9.32e-1(3.31e-3)−	8.33e-1(2.95e-2)+	**9.48e-1(8.72e-4)−**
	6	2.74e-1(2.68e-2)	9.11e-1(5.71e-2)−	9.58e-1(1.53e-2)−	9.93e-1(1.50e-3)−	**9.99e-1(2.24e-4)−**	9.98e-1(4.59e-4)−	7.32e-1(6.72e-2)−	9.96e-1(6.10e-4)−
	8	2.64e-1(3.07e-2)	7.14e-1(8.76e-2)−	9.47e-1(1.00e-2)−	9.95e-1(1.49e-3)−	**1.00e+00(1.58e-4)−**	9.99e-1(1.63e-4)−	7.76e-1(8.19e-2)−	9.97e-1(6.98e-4)−
	10	2.43e-1(4.04e-2)	6.40e-1(1.08e-1)−	9.49e-1(4.35e-3)−	9.97e-1(9.18e-4)−	1.00e+00(1.76e-4)−	**1.00e+00(8.09e-6)−**	8.80e-1(1.08e-1)−	9.98e-1(5.08e-4)−
WFG4	3	5.59e-1(6.78e-4)	5.48e-1(1.57e-3)+	5.54e-1(5.13e-4)+	5.43e-1(2.22e-3)+	**5.60e-1(1.56e-4)−**	5.44e-1(3.02e-3)+	5.49e-1(1.90e-3)+	5.59e-1(1.24e-3)=
	6	8.37e-1(3.35e-3)	6.55e-1(3.05e-2)+	8.20e-1(6.59e-4)+	8.27e-1(4.59e-3)+	**8.45e-1(2.05e-2)−**	8.29e-1(4.44e-3)+	8.29e-1(4.00e-3)+	7.68e-1(4.33e-3)+
	8	8.97e-1(4.13e-3)	5.78e-1(5.73e-2)+	8.79e-1(6.01e-3)+	9.08e-1(2.43e-3)−	**9.26e-1(1.76e-2)−**	8.11e-1(6.93e-2)+	8.96e-1(4.39e-3)=	8.12e-1(6.96e-3)+
	10	9.40e-1(5.29e-3)	5.92e-1(6.76e-2)+	8.80e-1(7.32e-3)+	9.61e-1(6.97e-4)−	**9.67e-1(6.51e-3)−**	8.55e-1(4.40e-2)+	9.42e-1(4.20e-3)=	8.69e-1(5.08e-3)+
WFG5	3	**5.21e-1(5.28e-4)**	5.05e-1(4.61e-3)+	5.14e-1(3.06e-4)+	5.06e-1(2.53e-3)+	5.20e-1(4.52e-5)+	4.98e-1(4.99e-3)+	5.19e-1(6.45e-4)+	5.19e-1(1.22e-3)+
	6	7.96e-1(1.14e-3)	6.47e-1(1.84e-2)+	7.67e-1(8.92e-4)+	7.81e-1(4.86e-3)+	**7.99e-1(4.15e-4)−**	7.70e-1(5.32e-3)+	7.95e-1(8.18e-4)+	7.22e-1(4.29e-3)+
	8	8.57e-1(1.29e-3)	6.16e-1(1.72e-2)+	8.18e-1(3.78e-3)+	8.39e-1(6.05e-3)+	**8.74e-1(2.45e-4)−**	8.31e-1(2.70e-2)+	8.57e-1(1.06e-3)=	7.59e-1(7.20e-3)+
	10	8.98e-1(1.09e-3)	6.28e-1(1.55e-2)+	8.17e-1(5.47e-3)+	8.98e-1(9.88e-4)=	**9.09e-1(2.31e-4)−**	8.69e-1(2.93e-2)+	8.98e-1(9.81e-4)=	8.02e-1(5.95e-3)+
WFG6	3	5.15e-1(1.49e-2)	4.96e-1(1.62e-2)+	5.07e-1(1.66e-2)+	4.89e-1(8.86e-3)+	5.07e-1(1.12e-2)+	4.89e-1(1.36e-2)+	5.04e-1(1.04e-2)+	**5.17e-1(1.31e-2)=**
	6	**7.80e-1(1.71e-2)**	4.26e-1(5.25e-2)+	7.43e-1(2.30e-2)+	7.48e-1(1.91e-2)+	7.72e-1(1.70e-2)=	7.54e-1(2.37e-2)+	7.68e-1(1.70e-2)+	7.03e-1(1.67e-2)+
	8	8.38e-1(1.40e-2)	3.59e-1(4.02e-2)+	7.91e-1(1.93e-2)+	8.03e-1(2.30e-2)+	**8.46e-1(1.88e-2)=**	7.69e-1(2.36e-2)+	8.25e-1(2.10e-2)+	7.45e-1(2.19e-2)+
	10	8.80e-1(1.68e-2)	3.81e-1(5.28e-2)+	7.82e-1(2.09e-2)+	8.71e-1(1.58e-2)+	**8.86e-1(1.66e-2)=**	8.51e-1(3.06e-2)+	8.64e-1(1.97e-2)+	7.92e-1(1.54e-2)+
WFG7	3	**5.62e-1(2.44e-4)**	5.37e-1(3.15e-3)+	5.53e-1(8.34e-4)+	5.47e-1(3.20e-3)+	5.58e-1(2.24e-4)+	5.27e-1(5.17e-3)+	5.58e-1(3.85e-4)+	5.60e-1(1.14e-3)+
	6	8.48e-1(1.54e-3)	4.66e-1(2.85e-2)+	8.17e-1(2.18e-3)+	8.31e-1(3.83e-3)+	**8.48e-1(8.88e-4)=**	8.18e-1(5.80e-3)+	8.48e-1(9.00e-4)=	7.80e-1(4.30e-3)+
	8	9.14e-1(2.29e-3)	4.58e-1(4.19e-2)+	8.86e-1(4.24e-3)+	8.90e-1(5.75e-3)+	**9.32e-1(4.92e-4)−**	8.89e-1(2.39e-2)+	9.02e-1(4.75e-3)+	8.15e-1(6.31e-3)+
	10	9.60e-1(1.27e-3)	5.15e-1(4.07e-2)+	9.00e-1(5.96e-3)+	9.56e-1(5.99e-3)+	**9.69e-1(8.67e-3)−**	9.50e-1(1.45e-2)+	9.55e-1(1.77e-3)+	8.73e-1(5.05e-3)+
WFG8	3	**4.84e-1(9.06e-4)**	4.68e-1(2.02e-3)+	4.74e-1(1.17e-3)+	4.51e-1(4.08e-3)+	4.78e-1(2.04e-3)+	4.21e-1(1.25e-2)+	4.66e-1(2.47e-3)+	4.79e-1(1.95e-3)+
	6	**7.51e-1(2.52e-2)**	2.58e-1(1.26e-1)+	7.21e-1(1.87e-2)+	6.66e-1(6.96e-3)+	7.35e-1(1.46e-2)=	6.86e-1(8.50e-3)+	7.06e-1(2.64e-2)+	6.17e-1(8.67e-3)+
	8	8.11e-1(2.82e-2)	1.80e-1(1.16e-1)+	**8.28e-1(2.09e-2)−**	7.54e-1(1.81e-2)+	8.22e-1(5.27e-2)=	7.55e-1(1.33e-2)+	6.69e-1(8.94e-2)+	6.23e-1(1.04e-2)+
	10	**9.00e-1(2.62e-2)**	1.97e-1(1.30e-1)+	8.24e-1(4.61e-2)+	8.37e-1(5.15e-2)+	8.96e-1(4.77e-2)=	8.44e-1(1.58e-2)+	7.13e-1(1.09e-1)+	6.77e-1(2.59e-2)+
WFG9	3	**5.45e-1(2.06e-3)**	5.05e-1(2.28e-2)+	5.29e-1(2.37e-2)+	5.27e-1(2.29e-2)+	5.36e-1(2.37e-2)+	5.25e-1(2.44e-2)+	5.38e-1(2.47e-3)+	5.42e-1(2.98e-3)+
	6	7.92e-1(9.18e-3)	3.54e-1(7.27e-2)+	5.93e-1(3.21e-2)+	7.26e-1(4.16e-2)+	6.39e-1(3.45e-2)+	**8.10e-1(8.70e-3)−**	7.94e-1(8.05e-3)=	7.22e-1(9.83e-3)+
	8	8.14e-1(2.69e-2)	2.52e-1(1.20e-1)+	5.66e-1(5.00e-2)+	7.49e-1(3.93e-2)+	6.71e-1(4.64e-2)+	**8.53e-1(4.49e-2)−**	8.43e-1(3.03e-2)−	7.35e-1(1.90e-2)+
	10	8.53e-1(2.92e-2)	1.56e-1(9.86e-2)+	5.56e-1(4.52e-2)+	7.04e-1(3.76e-2)+	6.56e-1(3.91e-2)+	8.93e-1(2.46e-2)−	**8.94e-1(1.18e-2)−**	5.78e-1(1.71e-1)+
+/=/−		-	33/1/10	27/1/16	37/1/6	16/12/16	37/0/7	26/9/9	36/3/5

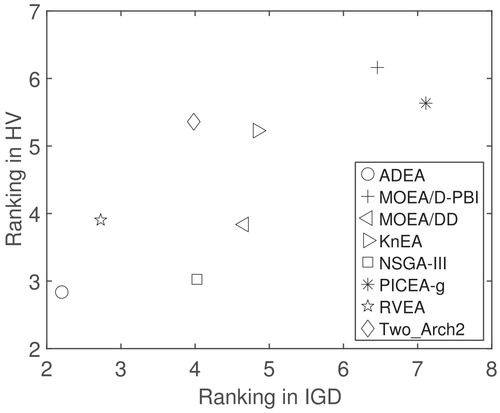


Fig. 5. Illustration of the rankings of each algorithm over all test instances.

From Tables 3 and 4, we can see that ADEA shows the most competitive IGD performance by obtaining 24 best mean IGD values of all 44 test instances, while NSGA-III and MOEA/DD perform better on DTLZ and WFG instances in terms of HV, respectively. In the following, we give a detailed discussion on the results for different problems.

4.2.1. Comparison on DTLZ1-4

For the DTLZ1 test instances, ADEA shows the best IGD value on 3- and 8-objective instances, while Two_Arch2 outperforms ADEA on 6-objective and 10-objective instances. In terms of the HV value, ADEA is only worse than MOEA/DD and MOEA/D-PBI over 3-objective instance. Fig. 6 shows that ADEA, MOEA/D-PBI, and MOEA/DD not only perform well on convergence but also maintain a good population diversity. On the contrary, both KnEA and PICEA-g have difficulties in achieving the true PF on DTLZ1 with 10 objectives.

For the DTLZ2 test instances, ADEA achieved the best mean IGD values in the case of 8 objectives and the second best in the case of 3 and 6 objectives. MOEA/DD and MOEA/D-PBI outperform the others in terms of the HV value. All decomposition-based algorithms can obtain high quality solutions on DTLZ2 with 10 objectives. From Fig. 7, only PICEA-g cannot converge in all objectives.

DTLZ3 has a large number of local optima, while DTLZ4 has a strong bias in the density of solutions. Similar results are obtained on DTLZ3 and DTLZ4, where ADEA is only worse than RVEA and KnEA on 10-objective instances in terms of IGD. On the contrary, MOEA/DD achieved the best HV performance on 6-, 8- and 10-objective instances. KnEA is outperformed by all other algorithms on DTLZ3. From Figs. 8 and 9, we can see that NSGA-III, PICEA-g, and Two_Arch2 are not stable in convergence on several objectives with the 10-objective DTLZ3 instance. Moreover, Two_Arch2 fails to obtain the boundary solutions on the 10-objective DTLZ4 instance.

4.2.2. Comparison on WFG1 and WFG4-9

The WFG test suite are much more difficult than DTLZ, since it involves several complex characteristics, e.g., a mixed shape of PF, different scales on different objective values, deceptive and non-separable in the objective space.

NSGA-III and Two_Arch2 show the best performance on WFG1. ADEA achieved higher HV values than REVA on 3-objective instance, but is outperformed by other competitors.

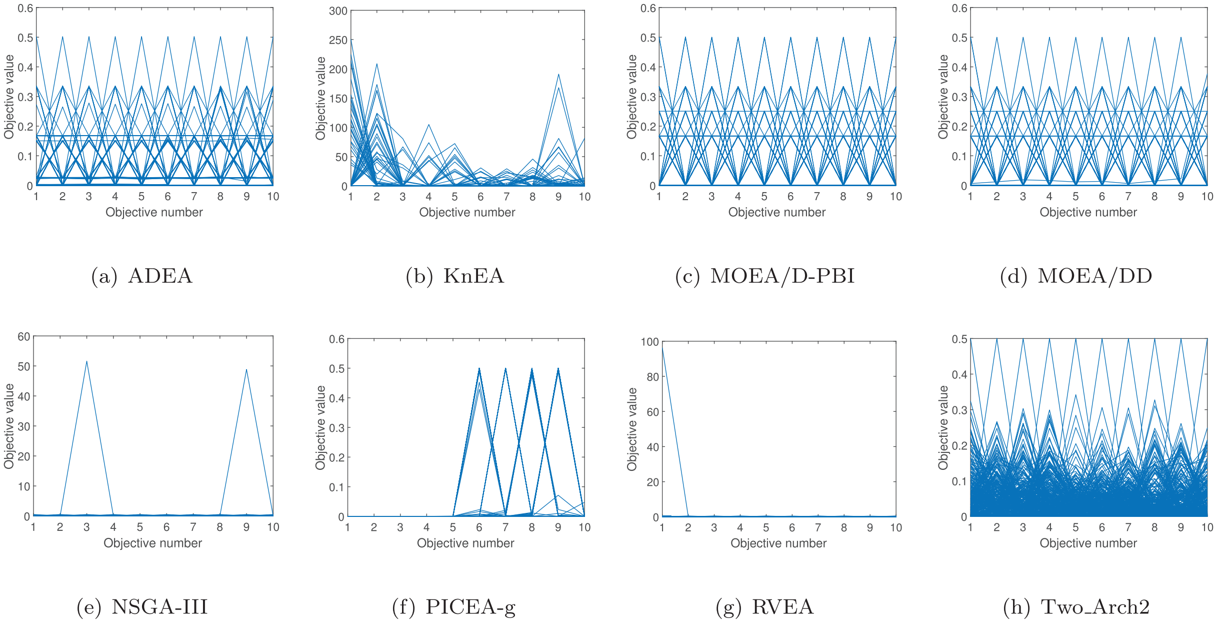
For the WFG4-9 test instances, ADEA shows better performance in terms of IGD over 3- and 8-objective instances, while ADEA is slightly worse than KnEA and Two_Arch2 over 6- and 10-objective instances. Specially, ADEA obtained the smallest IGD values on DTLZ6. ADEA is superior on WFG8 in terms of HV, and performs the second best on WFG4-7. It is surprising that PICEA-g works quite well on WFG9.

To make further study on the overall performance of different algorithms, the Friedman test is adopted to calculate the rankings of each algorithm over all test instances. As can be observed in Fig. 5, ADEA presents the most competitive overall performance on both IGD and HV indicators. NSGA-III and RVEA perform the second on IGD and HV, respectively. PICEA-g and MOEA/D-PBI show the worst overall performance. Therefore, we can conclude that the performance of ADEA is very promising among the rivals.

4.3. Effectiveness of the adaptive PBI strategy

To verify the effectiveness of the proposed adaptive PBI strategy, ADEA is compared with two existing penalty schemes [41].

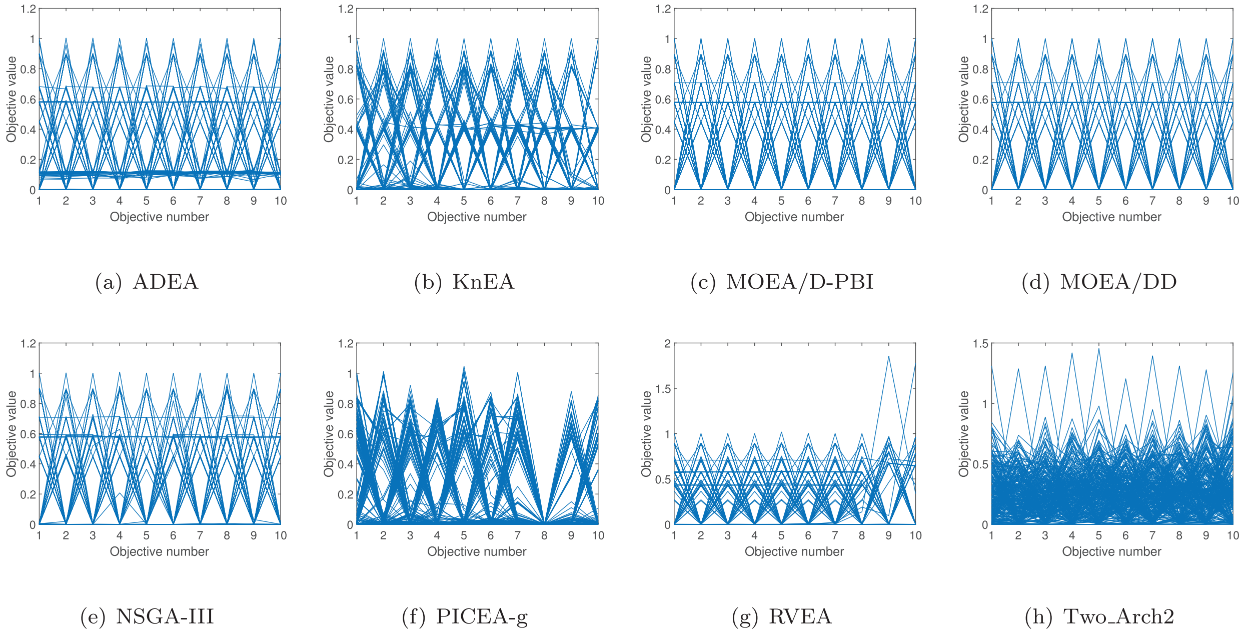
(1) APS: In this penalty scheme, the value of θ increases generation by generation in a fixed interval, which can speed up the convergence speed of the population in the early stage of the search and promote the population diversity in the later stage.



(a) ADEA (b) KnEA (c) MOEA/D-PBI (d) MOEA/DD

(e) NSGA-III (f) PICEA-g (g) RVEA (h) Two_Arch2

Fig. 6. The parallel coordinates of the obtained non-dominated front on 10-objective DTLZ1 with the median IGD value.



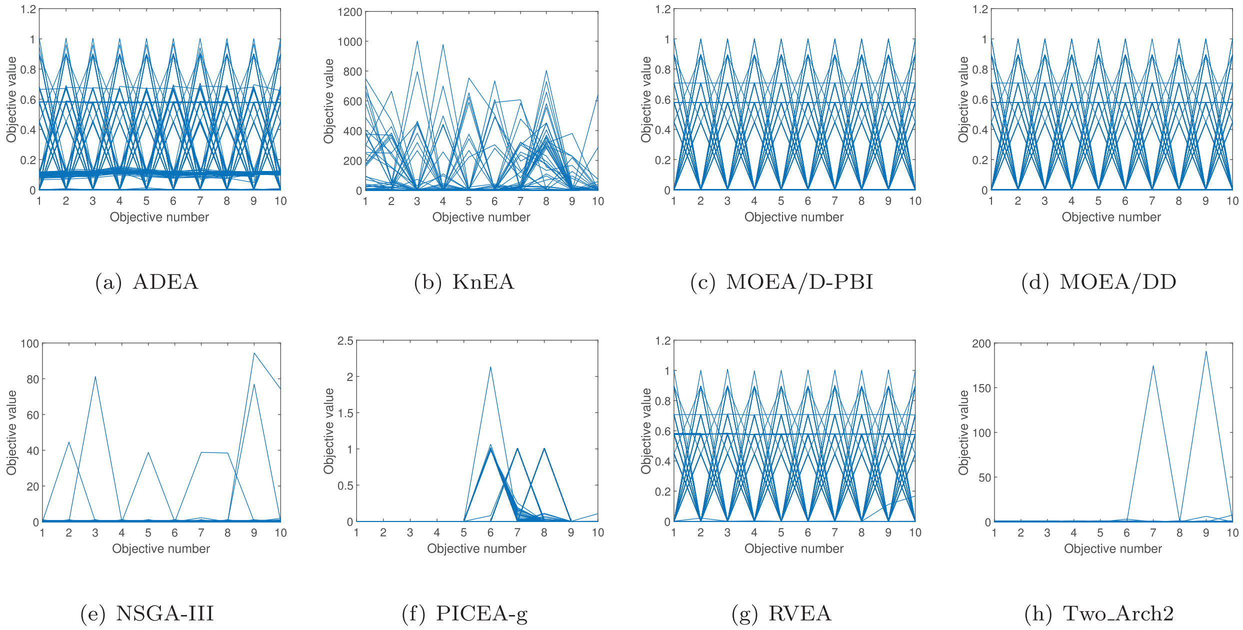
(a) ADEA (b) KnEA (c) MOEA/D-PBI (d) MOEA/DD

(e) NSGA-III (f) PICEA-g (g) RVEA (h) Two_Arch2

Fig. 7. The parallel coordinates of the obtained non-dominated front on 10-objective DTLZ2 with the median IGD value.

(2) SPS: Different from APS, SPS assigns a specific penalty value for each subproblem based on the position of its associated weight vector. The penalty value for boundary subproblems is relatively larger than that for the intermediate ones since preservation of boundary solutions is helpful to maintain diversity.

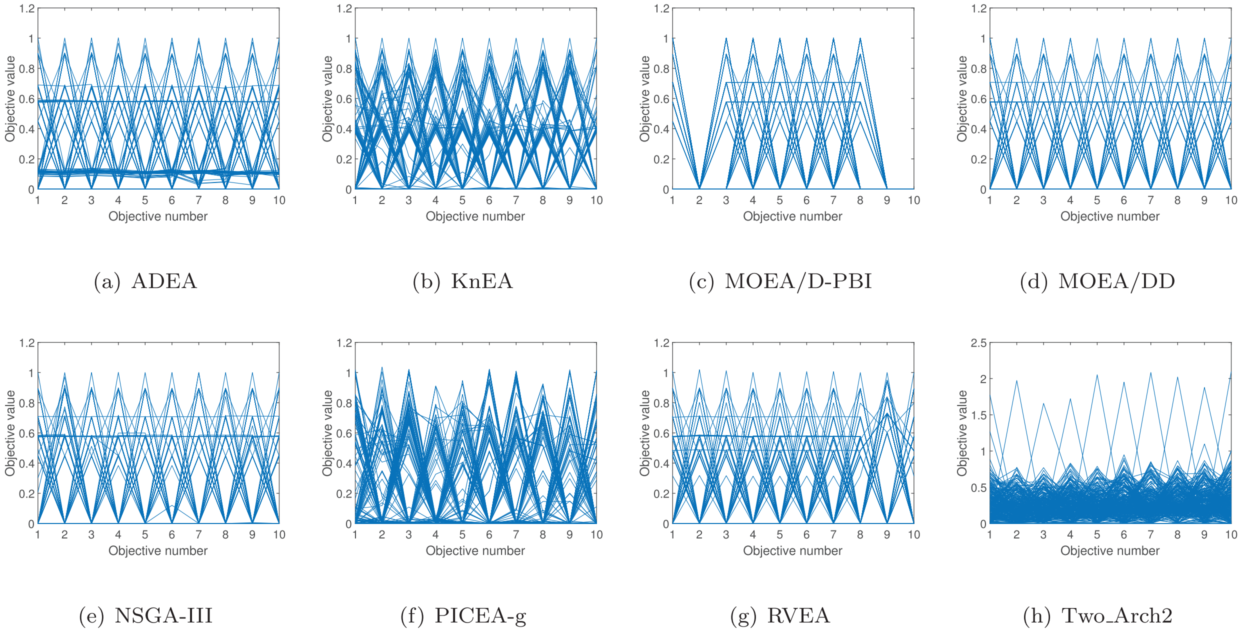
The fixed interval in APS is set as [1,10] and the control factor is set as $\alpha = 4$ in SPS, which are the same as those in [41]. We select the same test instances as mentioned in Section IV-A. Table 5 lists the mean IGD and HV results obtained by three penalty schemes over 30 independent runs. From the table, we can see that ADEA has achieved comparable IGD values to APS, but higher HV values than both APS and SPS on ten instances for the normalized DTLZ1-4. In terms of WFG4-9, the performance of ADEA is superior to that of the other two compared algorithms. As a result, the proposed ADEA is effective to strike a convergence-diversity balance.



(a) ADEA (b) KnEA (c) MOEA/D-PBI (d) MOEA/DD

(e) NSGA-III (f) PICEA-g (g) RVEA (h) Two_Arch2

Fig. 8. The parallel coordinates of the obtained non-dominated front on 10-objective DTLZ3 with the median IGD value.



(a) ADEA (b) KnEA (c) MOEA/D-PBI (d) MOEA/DD

(e) NSGA-III (f) PICEA-g (g) RVEA (h) Two_Arch2

Fig. 9. The parallel coordinates of the obtained non-dominated front on 10-objective DTLZ4 with the median IGD value.

4.4. *Effectiveness of the weight vector adaptation strategy*

To verify the effectiveness of the proposed weight vector adaptation strategy, ADEA is compared with two existing adaptation strategies proposed in RVEA and MOEA/D-AWA [23].

(1) The strategy in RVEA: In RVEA, the range of the objective value is firstly estimated by the minimum and maximum values on each objective from the current population. Then, the weight vectors are adjusted based on the estimated range periodically during the whole optimization process.

(2) The strategy in MOEA/D-AWA: In MOEA/D-AWA, an external archive that stores all achieved non-dominated solutions is used to guide the weight adding and removing for better population diversity. According to the stored solution in

Table 5
The mean IGD and HV values of ADEA, APS, and SPS on all test instances. The best result on each test instance is highlighted. " + ", " = ", and " − " denote that ADEA performs significantly better than, similar to, and worse than the compared scheme, respectively.

Problem	M	IGD			HV		
		ADEA	APS	SPS	ADEA	APS	SPS
DTLZ1	3	**1.87e-2**	1.90e-2+	1.92e-2+	8.43e-1	**8.43e-1=**	8.42e-1+
	6	8.14e-2	**8.14e-2=**	8.18e-2+	**9.89e-1**	9.88e-1+	9.88e-1+
	8	9.74e-2	**9.71e-2−**	9.75e-2+	**9.98e-1**	9.97e-1+	9.96e-1+
	10	1.17e-1	**1.17e-1−**	1.17e-1+	**1.00e+0**	9.99e-1+	9.99e-1+
DTLZ2	3	**5.03e-2**	5.04e-2+	5.07e-2+	**5.62e-1**	5.61e-1+	5.57e-1+
	6	**2.57e-1**	2.57e-1=	2.58e-1+	**8.52e-1**	8.52e-1+	8.46e-1+
	8	3.17e-1	**3.16e-1−**	3.18e-1+	9.20e-1	**9.21e-1−**	9.17e-1+
	10	4.24e-1	**4.22e-1−**	4.23e-1−	9.66e-1	**9.67e-1−**	9.66e-1+
DTLZ3	3	**5.07e-2**	5.19e-2+	5.40e-2+	**5.57e-1**	5.50e-1+	5.43e-1+
	6	**2.59e-1**	2.60e-1+	2.63e-1+	**8.48e-1**	8.44e-1+	8.32e-1+
	8	3.19e-1	**3.18e-1=**	3.50e-1+	**9.14e-1**	9.14e-1=	8.78e-1+
	10	4.23e-1	4.24e-1=	**4.22e-1=**	9.65e-1	**9.66e-1=**	9.64e-1+
DTLZ4	3	**5.03e-2**	5.04e-2+	5.09e-2+	**5.62e-1**	5.60e-1+	5.57e-1+
	6	2.58e-1	**2.58e-1=**	2.60e-1+	**8.52e-1**	8.51e-1+	8.44e-1+
	8	3.18e-1	**3.17e-1−**	3.19e-1+	9.19e-1	**9.20e-1−**	9.16e-1+
	10	4.24e-1	**4.22e-1−**	4.23e-1−	9.66e-1	**9.68e-1−**	9.66e-1+
WFG1	3	**4.02e-1**	4.42e-1+	8.04e-1+	**8.67e-1**	8.29e-1+	6.26e-1+
	6	2.40e+0	**9.62e-1−**	2.43e+0=	2.74e-1	**8.27e-1−**	2.70e-1=
	8	2.72e+0	**1.71e+0−**	2.71e+0=	2.64e-1	**8.50e-1−**	2.66e-1=
	10	3.22e+0	**2.44e+0−**	3.25e+0=	2.43e-1	**9.54e-1−**	2.30e-1=
WFG4	3	**2.03e-1**	2.08e-1+	2.23e-1+	**5.59e-1**	5.51e-1+	5.42e-1+
	6	1.74e+0	**1.74e+0=**	1.75e+0=	**8.37e-1**	8.36e-1=	8.33e-1+
	8	**2.97e+0**	3.27e+0+	2.97e+0=	**8.97e-1**	8.72e-1+	8.95e-1=
	10	**4.23e+0**	5.03e+0+	4.29e+0+	**9.40e-1**	8.94e-1+	9.40e-1=
WFG5	3	**2.14e-1**	2.15e-1+	2.26e-1+	**5.21e-1**	5.20e-1+	5.15e-1+
	6	**1.73e+0**	1.73e+0=	1.73e+0+	7.96e-1	**7.96e-1=**	7.96e-1=
	8	**2.94e+0**	3.13e+0+	2.95e+0+	8.57e-1	8.41e-1+	**8.57e-1=**
	10	**4.24e+0**	4.85e+0+	4.31e+0+	8.98e-1	8.70e-1+	**8.98e-1=**
WFG6	3	**2.20e-1**	2.30e-1+	2.70e-1+	**5.15e-1**	5.06e-1+	4.93e-1+
	6	**1.73e+0**	1.74e+0=	1.73e+0+	**7.80e-1**	7.76e-1=	7.76e-1=
	8	**2.96e+0**	3.85e+0+	2.97e+0+	8.38e-1	7.63e-1+	**8.46e-1=**
	10	**4.23e+0**	5.65e+0+	4.31e+0+	**8.80e-1**	7.81e-1+	8.75e-1=
WFG7	3	**2.04e-1**	2.08e-1+	2.49e-1+	**5.62e-1**	5.55e-1=	5.38e-1+
	6	**1.74e+0**	1.75e+0+	1.76e+0+	**8.48e-1**	8.42e-1+	8.42e-1+
	8	**2.97e+0**	3.44e+0+	2.98e+0+	**9.14e-1**	8.72e-1+	9.11e-1=
	10	**4.27e+0**	5.01e+0+	4.32e+0+	**9.60e-1**	9.25e-1+	9.60e-1=
WFG8	3	**2.55e-1**	2.65e-1+	2.99e-1+	**4.84e-1**	4.76e-1+	4.64e-1+
	6	**1.73e+0**	1.74e+0+	1.74e+0+	**7.51e-1**	7.44e-1=	7.44e-1=
	8	**3.04e+0**	3.87e+0+	3.05e+0+	**8.11e-1**	6.97e-1+	8.02e-1=
	10	**4.34e+0**	5.77e+0+	4.35e+0+	9.00e-1	7.14e-1+	**9.07e-1=**
WFG9	3	**2.01e-1**	2.06e-1+	2.23e-1+	**5.45e-1**	5.36e-1+	5.23e-1+
	6	1.71e+0	**1.70e+0=**	1.71e+0+	7.92e-1	**7.92e-1=**	7.75e-1+
	8	**2.93e+0**	3.18e+0+	2.95e+0+	**8.14e-1**	8.12e-1=	8.10e-1=
	10	**4.25e+0**	4.78e+0+	4.29e+0+	8.53e-1	**8.56e-1=**	8.42e-1=
+/=/−		-	26/9/9	36/6/2	-	26/11/7	26/18/0

the archive, the overcrowded weight vectors are deleted and new weight vectors are added in the sparse regions. This adaptive adjustment strategy is performed after the population has converged to some extent.

For comparison, these weight vector adaptation strategies are embedded into the proposed ADEA, denoted by ADEA-R and ADEA-AWA, respectively. The parameter in the compared strategies are set as suggested in the corresponding literature. The same benchmarks described in Section IV-A are used for testing. All the tests are run for 30 times independently, and the mean of IGD and HV are presented in Table 6. From the table, it can be clearly seen that ADEA is much better than the two compared methods in terms of HV, while ADEA and ADEA-R perform comparably in terms of IGD. It is worth noting that ADEA-AWA obtains the best results on WFG1. This is mainly due to the fact AWA strategy is designed for the irregular Pareto fronts of WFG1. Based on the above empirical results, we can conclude that the proposed adaptation strategy still performs well when dealing with both uniform-scaled and disparately-scaled problems.

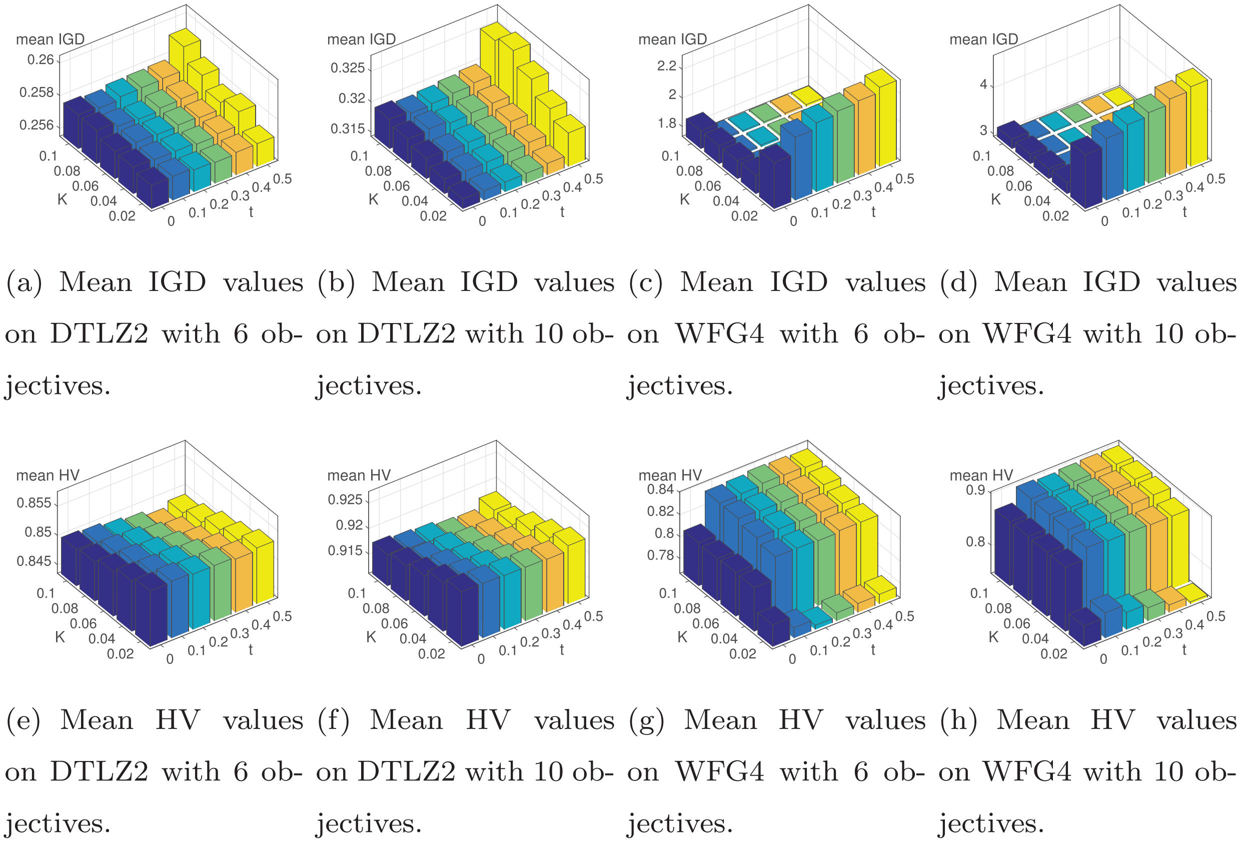
Table 6
The mean IGD and HV values of ADEA, ADEA-R, and ADEA-AWA on all test instances. The best result on each test instance is highlighted. " + ", " = ", and " − " denote that ADEA performs significantly better than, similar to, and worse than the compared scheme, respectively.

Problem		IGD			HV		
	M	ADEA	ADEA-RVEA	ADEA-AWA	ADEA	ADEA-RVEA	ADEA-AWA
	3	1.87e-2	**1.86e-2−**	2.35e-2+	**8.43e-1**	7.92e-1+	7.81e-1+
	6	**8.14e-2**	**8.15e-2+**	**1.46e-1+**	**9.89e-1**	9.80e-1+	8.60e-1+
	8	**9.74e-2**	**9.95e-2=**	**1.61e-1+**	**9.98e-1**	9.94e-1+	8.84e-1+
DTLZ1	10	**1.17e-1**	**1.18e-1+**	**1.78e-1+**	**1.00e+0**	9.99e-1+	9.10e-1+
	3	5.03e-2	**5.03e-2−**	5.95e-2+	**5.62e-1**	4.18e-1+	4.05e-1+
	6	**2.57e-1**	**2.57e-1=**	**3.83e-1+**	**8.52e-1**	7.37e-1+	5.15e-1+
	8	**3.17e-1**	**3.17e-1=**	**4.67e-1+**	**9.20e-1**	8.27e-1+	6.23e-1+
DTLZ2	10	4.24e-1	**4.24e-1=**	5.21e-1+	**9.66e-1**	9.14e-1+	7.22e-1+
	3	5.07e-2	**5.06e-2=**	5.84e-2+	**5.57e-1**	4.12e-1+	4.05e-1+
	6	**2.59e-1**	**2.59e-1+**	**5.30e-1+**	**8.48e-1**	7.23e-1+	2.51e-1+
	8	**3.19e-1**	**3.20e-1=**	**6.72e-1+**	**9.14e-1**	8.12e-1+	3.72e-1+
DTLZ3	10	**4.23e-1**	**4.24e-1+**	**6.85e-1+**	**9.65e-1**	9.10e-1+	5.12e-1+
	3	5.03e-2	**5.03e-2=**	5.88e-2+	**5.62e-1**	4.17e-1+	4.07e-1+
	6	**2.58e-1**	**2.59e-1+**	**3.51e-1+**	**8.52e-1**	7.35e-1+	5.68e-1+
	8	**3.18e-1**	**3.18e-1=**	**4.36e-1+**	**9.19e-1**	8.27e-1+	6.72e-1+
DTLZ4	10	4.24e-1	**4.24e-1=**	4.74e-1+	**9.66e-1**	9.14e-1+	7.88e-1+
	3	4.02e-1	4.05e-1=	**2.63e-1−**	8.67e-1	8.63e-1=	**8.97e-1−**
	6	2.40e+0	2.44e+0=	**1.52e+0−**	2.74e-1	2.08e-1+	**5.02e-1−**
	8	2.72e+0	2.79e+0=	**2.06e+0−**	2.64e-1	2.08e-1+	**4.30e-1−**
WFG1	10	3.22e+0	3.26e+0=	**2.27e+0−**	2.43e-1	1.93e-1+	**4.65e-1−**
	3	2.03e-1	**2.03e-1−**	2.22e-1+	**5.59e-1**	4.14e-1+	3.98e-1+
	6	1.74e+0	**1.74e+0=**	1.92e+0+	**8.37e-1**	7.19e-1+	6.43e-1+
	8	**2.97e+0**	**2.97e+0=**	**3.29e+0+**	**8.97e-1**	8.00e-1+	7.22e-1+
WFG4	10	4.23e+0	**4.23e+0=**	4.25e+0=	**9.40e-1**	8.76e-1+	8.13e-1+
	3	2.14e-1	**2.13e-1−**	2.28e-1+	**5.21e-1**	3.74e-1+	3.67e-1+
	6	1.73e+0	**1.73e+0=**	1.88e+0+	**7.96e-1**	6.78e-1+	6.07e-1+
	8	2.94e+0	**2.94e+0=**	3.24e+0+	**8.57e-1**	7.61e-1+	6.85e-1+
WFG5	10	4.24e+0	4.24e+0=	**4.22e+0−**	**8.98e-1**	8.38e-1+	7.76e-1+
	3	**2.20e-1**	**2.21e-1=**	**2.44e-1+**	**5.15e-1**	3.65e-1+	3.56e-1+
	6	1.73e+0	**1.73e+0−**	1.95e+0+	**7.80e-1**	6.55e-1+	5.76e-1+
	8	2.96e+0	**2.95e+0=**	3.36e+0+	**8.38e-1**	7.39e-1+	6.52e-1+
WFG6	10	4.23e+0	**4.22e+0=**	4.26e+0+	**8.80e-1**	8.10e-1+	7.52e-1+
	3	**2.04e-1**	**2.04e-1=**	**2.19e-1+**	**5.62e-1**	4.18e-1+	4.08e-1+
	6	**1.74e+0**	**1.75e+0+**	**1.93e+0+**	**8.48e-1**	7.29e-1+	6.53e-1+
	8	**2.97e+0**	**2.97e+0+**	**3.32e+0+**	**9.14e-1**	8.19e-1+	7.50e-1+
WFG7	10	4.27e+0	4.26e+0=	**4.21e+0−**	**9.60e-1**	9.05e-1+	8.50e-1+
	3	2.55e-1	**2.55e-1=**	2.86e-1+	**4.84e-1**	3.35e-1+	3.15e-1+
	6	1.73e+0	**1.73e+0=**	1.99e+0+	**7.51e-1**	6.05e-1+	5.02e-1+
	8	**3.04e+0**	**3.05e+0=**	**3.49e+0+**	**8.11e-1**	6.68e-1+	5.96e-1+
WFG8	10	4.34e+0	**4.31e+0−**	4.49e+0+	**9.00e-1**	8.38e-1+	7.27e-1+
	3	2.01e-1	**2.10e-1=**	2.15e-1+	**5.45e-1**	4.01e-1+	3.86e-1+
	6	1.71e+0	**1.71e+0=**	1.89e+0+	**7.92e-1**	6.56e-1+	5.80e-1+
	8	2.93e+0	**2.93e+0=**	3.30e+0+	**8.14e-1**	7.09e-1+	6.14e-1+
WFG9	10	4.25e+0	**4.24e+0=**	4.32e+0+	**8.53e-1**	7.78e-1+	7.04e-1+
+/=/−		-	7/31/6	37/1/6	-	43/1/0	40/0/4

4.5. Parameter sensitivity analysis

In ADEA, two major parameters are introduced, K controlling the scaling of penalty factor and t controlling the frequency to implement weight vector adaptation. To investigate the sensitivity of ADEA to these two parameters, we compare 30 different parameter configurations, where K varies from 0.02 to 0.1 with a step size of 0.02 and t varies from 0.0 to 0.5 with a step size of 0.1. We implement each configuration on DTLZ2 and WFG4 with 6 and 10 objectives for 30 independent runs.

The mean IGD and HV values obtained by all parameter configurations is plotted in Fig. 10. The following three observations can be made. First, for the normalized and uni-modal DTLZ2, the performance of ADEA is insensitive to the parameter t. It is because the uniformly distributed weight vector is sufficient to ensure a good population diversity. Since DTLZ2 is uni-modal, a small K, which can result in a loose penalty, is beneficial for improving the convergence. In contrast to DTLZ2, WFG4 scales differently on different objectives. It is evident that weight vector adaptation is necessary for maintaining a set of evenly distributed solutions (e.g., $t > 0.0$). Meanwhile, the frequency of weight vector adaptation does not much affect the



(a) Mean IGD values on DTLZ2 with 6 objectives.

(b) Mean IGD values on DTLZ2 with 10 objectives.

(c) Mean IGD values on WFG4 with 6 objectives.

(d) Mean IGD values on WFG4 with 10 objectives.

(e) Mean HV values on DTLZ2 with 6 objectives.

(f) Mean HV values on DTLZ2 with 10 objectives.

(g) Mean HV values on WFG4 with 6 objectives.

(h) Mean HV values on WFG4 with 10 objectives.

Fig. 10. Illustration of the mean IGD and HV values obtained by all parameter configurations on DTLZ2 and WFG4 with 6 and 10 objectives.

performance of ADEA. In terms of K, the setting $K = 0.02$ is always the worst whereas the performance is almost the same in the cases of $K > 0.02$. The reason for this is that a larger objective value of WFG4 requires a stronger penalty on the loss of diversity. Based on the above sensitivity analysis, it is proper to choose K between 0.04 and 0.08, and t between 0.1 and 0.5.

5. Conclusion

This paper has proposed an adaptive decomposition-based many-objective evolutionary algorithm, named ADEA. In ADEA, two main adaptation mechanisms are designed for the PBI decomposition method and weight vectors, respectively. The former assigns a specific penalty factor of PBI for different subproblems by using the distribution information of both population and weight vectors, while the latter adjusts the weight vectors in the light of the value range of each objective to solve problems with various scales on different objectives.

For the investigation of the algorithm performance, we compare ADEA with seven state-of-the-art MOEAs on a number of widely used test instances. Our experimental results demonstrate that ADEA is competitive among all compared algorithms.

In the future, we are going to extend the algorithm to solving a wider range of problems, e.g., MaOPs with a complex PF shape, constrained problems, and large-scale optimization problems. In addition, integrating ADEA with the preference provided by decision makers to search a subspace of the objective space will also be considered.

Acknowledgements

This work was supported by National Natural Science Foundation of China (Major Program: 61590923), National Science Fund for Distinguished Young Scholars (61725301), the Fundamental Research Funds for the Central Universities and the Programme of Introducing Talents of Discipline to Universities (the 111 Project) under Grant B17017.

References

[1] L.S. Batista, F. Campelo, F.G. Guimarães, J.A. Ramírez, A comparison of dominance criteria in many-objective optimization problems, in: Evolutionary Computation (CEC), 2011 IEEE Congress on, IEEE, 2011, pp. 2359–2366.

[2] R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, A reference vector guided evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 20 (5) (2016) 773–791.
[3] I. Das, J.E. Dennis, A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems, Struct. Optim. 14 (1) (1997) 63–69.
[4] K. Deb, A fast elitist multi-objective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
[5] K. Deb, R.B. Agrawal, Simulated binary crossover for continuous search space, Complex Syst. 9 (3) (1994) 115–148.
[6] K. Deb, M. Goyal, A combined genetic adaptive search (geneas) for engineering design, Comput. Sci. Inform. 26 (1996) 30–45.
[7] 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.
[8] K. Deb, K. Miettinen, S. Chaudhuri, Toward an estimation of nadir objective vector using a hybrid of evolutionary and local search approaches, IEEE Trans. Evol. Comput. 14 (6) (2010) 821–841.
[9] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable multi-objective optimization test problems, in: Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on, 1, 2002, pp. 825–830.
[10] M. Fleischer, The measure of pareto optima applications to multi-objective metaheuristics, in: C.M. Fonseca, P.J. Fleming, E. Zitzler, L. Thiele, K. Deb (Eds.), Evolutionary Multi-Criterion Optimization, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 519–533.
[11] J. Garcia, A. Berlanga, J.M.M. Lpez, Effective evolutionary algorithms for many-specifications attainment: application to air traffic control tracking filters, IEEE Trans. Evol. Comput. 13 (1) (2009) 151–168.
[12] I. Giagkiozis, R. Purshouse, P. Fleming, Generalized decomposition and cross entropy methods for many-objective optimization, Inf. Sci. (Ny) 282 (2014) 363–387.
[13] R.H. Gmez, C.A.C. Coello, Mombi: A new metaheuristic for many-objective optimization based on the r2 indicator, in: 2013 IEEE Congress on Evolutionary Computation, 2013, pp. 2488–2495.
[14] Z. He, G.G. Yen, J. Zhang, Fuzzy-based pareto optimality for many-objective evolutionary algorithms, IEEE Trans. Evol. Comput. 18 (2) (2014) 269–285.
[15] 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.
[16] H. Ishibuchi, K. Doi, Y. Nojima, Use of piecewise linear and nonlinear scalarizing functions in moea/d, in: J. Handl, E. Hart, P.R. Lewis, M. López-Ibáñez, G. Ochoa, B. Paechter (Eds.), Parallel Problem Solving from Nature – PPSN XIV, Springer International Publishing, Cham, 2016, pp. 503–513.
[17] H. Ishibuchi, Y. Sakane, N. Tsukamoto, Y. Nojima, Adaptation of scalarizing functions in moea/d: An adaptive scalarizing function-based multiobjective evolutionary algorithm, in: M. Ehrgott, C.M. Fonseca, X. Gandibleux, J.-K. Hao, M. Sevaux (Eds.), Evolutionary Multi-Criterion Optimization, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 438–452.
[18] H. Ishibuchi, Y. Sakane, N. Tsukamoto, Y. Nojima, Simultaneous use of different scalarizing functions in MOEA/D, in: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, in: GECCO '10, ACM, New York, NY, USA, 2010, pp. 519–526.
[19] H. Ishibuchi, N. Tsukamoto, Y. Nojima, Evolutionary many-objective optimization: A short review, in: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), 2008, pp. 2419–2426.
[20] B. Li, K. Tang, J. Li, X. Yao, Stochastic ranking algorithm for many-objective optimization based on multiple indicators, IEEE Trans. Evol. Comput. 20 (6) (2016) 924–938.
[21] K. Li, K. Deb, Q. Zhang, S. Kwong, An evolutionary many-objective optimization algorithm based on dominance and decomposition, IEEE Trans. Evol. Comput. 19 (5) (2015) 694–716.
[22] R.C. Purshouse, P.J. Fleming, On the evolutionary optimization of many conflicting objectives, IEEE Trans. Evol. Comput. 11 (6) (2007) 770–784.
[23] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, J. Wu, Moea/d with adaptive weight adjustment, Evol. Comput. 22 (2) (2014) 231–264.
[24] S. Rostami, F. Neri, Covariance matrix adaptation pareto archived evolution strategy with hypervolume-sorted adaptive grid algorithm, Integr. Comput. Aided Eng. 23 (4) (2016) 313–329.
[25] S. Rostami, F. Neri, A fast hypervolume driven selection mechanism for many-objective optimisation problems, Swarm Evol. Comput. 34 (2017) 50–67.
[26] S. Rostami, F. Neri, M. Epitropakis, Progressive preference articulation for decision making in multi-objective optimisation problems, Integr. Comput. Aided Eng. 24 (4) (2017) 315–335.
[27] H. Sato, Inverted PBI in MOEA/D and its impact on the search performance on multi and many-objective optimization, in: Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation, in: GECCO '14, ACM, New York, NY, USA, 2014, pp. 645–652.
[28] H. Scheff, Experiments with mixtures, J. R. Stat. Soc. 20 (2) (1958) 344–360.
[29] Y. Tian, R. Cheng, X. Zhang, F. Cheng, Y. Jin, An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility, IEEE Trans. Evol. Comput. 22 (4) (2018) 609–622.
[30] Y. Tian, R. Cheng, X. Zhang, Y. Jin, PlatEMO: A MATLAB platform for evolutionary multi-objective optimization, IEEE Comput. Intell. Mag. 12 (4) (2017) 73–87.
[31] Y. Tian, R. Cheng, X. Zhang, Y. Su, Y. Jin, A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization, IEEE Trans. Evol. Comput. (2018). 1–1
[32] M. Wagner, F. Neumann, A fast approximation-guided evolutionary multi-objective algorithm, in: Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation, in: GECCO '13, ACM, New York, NY, USA, 2013, pp. 687–694.
[33] T. Wagner, N. Beume, B. Naujoks, Pareto-, aggregation-, and indicator-based methods in many-objective optimization, in: Evolutionary multi-criterion optimization, Springer, 2007, pp. 742–756.
[34] H. Wang, S. He, X. Yao, Nadir point estimation for many-objective optimization problems based on emphasized critical regions, Soft Comput. 21 (9) (2017) 2283–2295.
[35] H. Wang, L. Jiao, X. Yao, Two_arch2: an improved two-archive algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 19 (4) (2015) 524–541.
[36] H. Wang, M. Olhofer, Y. Jin, A mini-review on preference modeling and articulation in multi-objective optimization: current status and challenges, Compl. Intell. Syst. 3 (4) (2017) 233–245.
[37] R. Wang, R.C. Purshouse, P.J. Fleming, Preference-inspired coevolutionary algorithms for many-objective optimization, IEEE Trans. Evol. Comput. 17 (4) (2013) 474–494.
[38] R. Wang, Q. Zhang, T. Zhang, Decomposition-based algorithms using pareto adaptive scalarizing methods, IEEE Trans. Evol. Comput. 20 (6) (2016) 821–837.
[39] M. Wei, M. Yang, F. Qian, W. Du, W. Zhong, Integrated dual-production mode modeling and multiobjective optimization of an industrial continuous catalytic naphtha reforming process, Industr. Eng. Chem. Res. 55 (19) (2016) 5714–5725.
[40] L. While, P. Hingston, L. Barone, S. Huband, A faster algorithm for calculating hypervolume, IEEE Trans. Evol. Comput. 10 (1) (2006) 29–38.
[41] S. Yang, S. Jiang, Y. Jiang, Improving the multiobjective evolutionary algorithm based on decomposition with new penalty schemes, Soft Comput. 21 (16) (2016) 1–15.
[42] 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.
[43] S. Ying, L. Li, Z. Wang, W. Li, W. Wang, An improved decomposition-based multiobjective evolutionary algorithm with a better balance of convergence and diversity, Appl. Soft. Comput. 57 (2017) 627–641.
[44] Y. Yuan, H. Xu, B. Wang, X. Yao, A new dominance relation-based evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 20 (1) (2016) 16–37.
[45] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731.

[46] X. Zhang, Y. Tian, R. Cheng, Y. Jin, An efficient approach to nondominated sorting for evolutionary multiobjective optimization, IEEE Trans. Evol. Comput. 19 (2) (2015) 201–213.

[47] 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.

[48] A. Zhou, B.Y. Qu, H. Li, S.Z. Zhao, P.N. Suganthan, Q. Zhang, Multiobjective evolutionary algorithms: a survey of the state of the art, Swarm Evol. Comput. 1 (1) (2011) 32–49.

[49] E. Zitzler, S. Künzli, Indicator-based selection in multiobjective search, in: X. Yao, E.K. Burke, J.A. Lozano, J. Smith, J.J. Merelo-Guervós, J.A. Bullinaria, J.E. Rowe, P. Tiňo, A. Kabán, H.-P. Schwefel (Eds.), Parallel Problem Solving from Nature - PPSN VIII, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 832–842.

[50] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, V.G. da Fonseca, Performance assessment of multiobjective optimizers: an analysis and review, IEEE Trans. Evol. Comput. 7 (2) (2003) 117–132.