Soft Computing (2020) 24:13179–13195
https://doi.org/10.1007/s00500-020-04732-y

METHODOLOGIES AND APPLICATION

Self-adaptive weight vector adjustment strategy for decomposition-based multi-objective differential evolution algorithm

Rui Fan[1] · Lixin Wei[1] · Xin Li[2] · Jinlu Zhang[1] · Zheng Fan[3]

Published online: 4 February 2020

Abstract

In multi-objective and many-objective optimization, weight vectors are particularly crucial to the performance of decomposition-based optimization algorithms. The uniform weight vectors are not suitable for complex Pareto fronts (PFs), so it is necessary to improve the distribution of weight vectors. Besides, the balance between convergence and diversity is a difficult issue as well in multi-objective optimization, and it becomes increasingly important with the augment of the number of objectives. To address these issues, a self-adaptive weight vector adjustment strategy for decomposition-based multi-objective differential evolution algorithm (AWDMODE) is proposed. In order to ensure that the guidance of weight vectors becomes accurate and effective, the adaptive adjustment strategy is introduced. This strategy distinguishes the shapes and adjusts weight vectors dynamically, which can ensure that the guidance of weight vectors becomes accurate and effective. In addition, a self-learning strategy is adopted to produce more non-dominated solutions and balance the convergence and diversity. The experimental results indicate that AWDMODE outperforms the compared algorithms on WFG suites test instances, and shows a great potential when handling the problems whose PFs are scaled with different ranges in each objective.

Keywords Evolutionary computations · Multi-objective optimization · Decomposition · Many-objective optimization · Weight vector

1 Introduction

Various real-world optimization problems can be described as multi-objective optimization problems (MOPs), e.g., optimal power flow problem (Medina et al. 2014), economic emission unit commitment problem (Trivedi et al. 2015), vehicle routing problem with stochastic demands (Gee et al. 2016), metal rolling control (Hu et al. 2019a). A MOP contains more than one objective and they often conflict with each other. There is usually no solution can achieve the optimum for all objectives at the same time. That means, the optimal solution of a MOP is a set of trade-off solutions rather than a single one.

Without loss of generality, a MOP can be described as follows:

$$\begin{aligned} &\min\ F(x) = (f_1(x), f_2(x), \ldots, f_M(x))^T, \\ &\text{s.t.}\quad x \in \Omega^D, \end{aligned} \tag{1}$$

where $x = (x_1, x_2, \ldots x_N)^T$, N is the scale of the solution set, $x_i = (x_i^1, \ldots, x_i^D)$ for all $i = 1, \ldots, N$ is one of the solutions with D-dimensional decision variables in the decision space Ω^D, and $\Omega^D = \prod_{i=1}^{D}[a_i, b_i] \subset R^D$, M is the number of objective functions and the mapping function $F : \Omega^D \rightarrow R^M$ defines M- objective functions in the objective space R^M. When $M > 3$, the MOP is called many-objective optimization problems (MaOPs). Multi-objective evolutionary algorithms (MOEAs) are demonstrated to be suitable for handling kinds of MOPs.

Communicated by V. Loia.

✉ Lixin Wei
wlx2000@ysu.edu.cn

Rui Fan
fanfanfr@163.com

[1] Institute of Electrical Engineering, Yanshan University, Qinhuangdao 066004, Hebei, People's Republic of China

[2] Faculty of Information Technology, Beijing University of Technology, Beijing 100124, People's Republic of China

[3] Luoyang Ruize Petrochemical Engineering Co., Ltd., Luoyang 471000, Henan, People's Republic of China

Over the past two decades, most MOEAs have been proposed, such as SPEA (Zitzler and Thiele 1999), NSGAII (Deb et al. 2002), SPEA2 (Zitzler et al. 2001), ASMiGA (Nag et al. 2015). These algorithms evaluate individuals by using Pareto dominance, which evaluate the fitness together with a diversity maintenance as a secondary criterion and perform well in two or three objectives optimization. However, the performance of the algorithm drops sharply when the number of objectives increases. Because almost all individuals in the population are non-dominated solutions. The degenerated selection pressure makes the Pareto-dominance-based algorithms fail to solve MaOPs (Purshouse and Fleming 2007).

Apart from those Pareto-dominance-based MOEAs, it has been an increasing interest in decomposition-based approach to solving MOPs and MaOPs. Decomposition-based approaches decompose MOPs or MaOPs into a number of scalar subproblems and optimize them simultaneously, such as MOEA/D (Zhang and Li 2007), MOEA-TPN (Jiang and Yang 2016), NSGAIII (Deb and Jain 2014). In this type of algorithm, the setting of weight vectors of aggregation functions plays a key role in the performance (Fan et al. 2019). In other words, weight vectors with different shapes produce different optimization results (Ishibuchi et al. 2017). Some efforts have been made to adjust weight vectors dynamically, the detailed introduction will be given in Sect. 2.2.

Besides, the used aggregation method and the setting of reference points have also some effects on the performance of algorithms (Wang et al. 2017b). In study Wang et al. (2017a), the dynamic reference point was adopted through the adjustment of ϵ, which can strike a good balance between exploitation and exploration. In a recent study RPEA (Liu et al. 2017), according to the information of the current population, a series of reference points including several local ideal points and a global ideal point with good performances in convergence and distribution were continuously generated to guide the evolution. Literature (Liu et al. 2010) utilized a monotonic increasing function, in order to transform each objective function into a hyperplane in the original objective function space and ensure the non-dominance solutions are close to it. However, it cannot work well when solve complex problems with disconnected subregions, a sharp peak and a long tail.

In addition to the above two methods, indicator-based methods for multi-objective optimization have also been studied, such as hypervolume estimation algorithm for multi-objective optimization (HypE) (Bader and Zitzler 2011) and R2MOPSO (Li et al. 2015a). The optimization process is guided according to the indicators values in this category of algorithms. Although such methods have sound theoretical support, the heavy computational complexity is a fatal flaw.

Some other algorithms can be viewed as the hybrid of different mechanisms, e.g., MOEA/DD (Li et al. 2015b) used the Pareto-dominance-based fitness evaluation and the MOEA/D framework. R2HMOPSO (Wei et al. 2018) mixed three methods above. Stable matching model (STM) was introduced to coordinate the selection process (Li et al. 2014). Some studies (Ying et al. 2017; Yuan et al. 2016; Hu et al. 2017, 2019b) made efforts on the balance of convergence and diversity, which is a difficulty in multi-objective and many-objective optimization.

Given the above considerations, the evenly distributed weight vectors are not suitable for complex Pareto fronts (PFs), so a new optimizer including the adaptive adjustment of weight vectors and the balance of convergence and diversity is considered in this paper. Based on this, a self-adaptive weight vector adjustment strategy for decomposition-based multi-objective differential evolution algorithm (AWDMODE) is proposed in this work. The main properties of AWDMODE can be summarized as follows.

(1) At the initialization phase, three weight vectors with different shapes (line, concave, convex) are generated, so as to facilitate adaptive processing of optimization problems with different shapes of PFs.

(2) Not only the shapes of PFs, but also the scope of PFs are considered in AWDMODE. When the algorithm runs to a certain degree, the weight vectors are adjusted in the searched space according to the information of the current population. The purpose of this strategy is to make weight vectors more relevant to the PFs and ensure the guidance of weight vectors becomes more accurate and effective.

(3) A parameter ξ is introduced to balance the convergence and diversity, different ξ guides the algorithm to choose different environment selection strategies. In the early stage of the optimization, the main task of AWDMODE is to enhance convergence. In order to accelerate the convergence rate, a self-learning strategy is adopted. In this strategy, the solutions in lower dominance level are close to that in higher dominance level. When the algorithm reaches a certain degree of convergence, the main task of evolution is increasing diversity. The coordinated switching of the two environmental selection ensures better convergence and distribution of the final solution set.

In the remainder of this paper, Sect. 2 reviews the related work. Then, the proposed AWDMODE is outlined in detail in Sect. 3. Thereafter, the simulation results on a set of test instances are presented and the analysis is given in Sect. 4. Finally, Sect. 5 concludes this paper and provides suggestions on possible opportunities for future research.

2 Background and related work

2.1 Decomposition methods

Three scalar functions were introduced in original MOEA/D framework: (1) the weighted sum; (2) the weighted Tchebycheff function; and (3) the penalty-based boundary intersection (PBI) function. A set of uniformly spread weight vectors is generated, i.e., $W = \{w_1, w_2, \ldots, w_N\}$, where $w_i = (w_i^1, w_i^2, \ldots, w_i^M)$ for all $i = 1, \ldots, N$. The MOP in Eq. (1) is decomposed into N scalar subproblems by these weight vectors and the subproblems are optimized simultaneously.

(1) *Weighted Sum (WS) approach* In this approach, the aggregate function is written as

$$\min\ g^{ws}(x_i|w_i) = \sum_{j=1}^{M} w_i^j f_j(x_i), \tag{2}$$

where $i = 1, \ldots N$. This approach works well to solve the problems with convex PF. However, it cannot obtain the entire PF and the effect is slightly poor for nonconvex PFs.

(2) *Tchebycheff approach* The corresponding aggregate function formula is defined as:

$$\min\ g^{te}(x \mid w, z^*) = \max_{1 \leq i \leq M} \{w_i |f_i(x) - z_i^*|\}, \tag{3}$$

where $z^* = (z_1^*, z_2^*, \ldots, z_M^*)^T$ is a reference point, and $z_i^* = min\{f_i(x) \mid x \in \Omega^D\}$, $i = 1, 2, \ldots, M$, w_i corresponding to the i-th subproblem.

(3) *Penalty-based boundary intersection (PBI) approach* In this approach, the i-th subproblem is defined in the form

$$\min\ g^{pbi}(x|w_i, z^*) = d_1 + \theta d_2, \tag{4}$$

where θ is penalty parameter and d_1, d_2 are defined as follows:

$$\begin{aligned} d_1 &= \frac{|(f(x) - z^*)^T w|}{\|w\|}, \\ d_2 &= \|f(x) - z^* - d_1 \frac{w}{\|w\|}\|. \end{aligned} \tag{5}$$

Different decomposition methods produce great influence on the performance of algorithm (Lee et al. 2014). Here, Tchebycheff decomposition method is adopted in this paper.

2.2 Related work

The weight vectors play a fundamental role in MOEA/D, while the evenly distributed weight vectors are not suitable for complex PFs, especially for the PFs with a long tail and a sharp peak. In study Ishibuchi et al. (2017), the experimental results illustrated that the distribution of weight vectors should be adjusted to the shape and the size of the PF. There are several related works on the improvement of weight vectors of MOEA/D.

In MOEA/D-AWA (Qi et al. 2014), WS-transformation was proposed as a new weight vectors initialization method. According to the current optimal solution set periodically, an adaptive weight vector adjustment (AWA) strategy was designed to adjust the distribution of the weight vectors. Besides, MOEA/D-AWA divides the whole optimization procedure into two phases to balance convergence and diversity. When dealing with MOPs with discrete PFs, the biggest challenge is how to identify sparse regions and true discrete regions. Therefore, vicinity distance is used to evaluate the sparsity of the solutions in the current non-dominated set, then new subproblems are added into the real sparse regions. This strategy demonstrates that MOEA/D-AWA is great potential for solving complex problems with disconnected subregions, a sharp peak or a long tail.

Similarly, in MOEA/D-TPN (Jiang and Yang 2016), a two-phase strategy (TP) is employed. During the first stage, convergence is as important as diversity in optimization, and at the end of the first stage, the algorithm identifies the geometric shape of the PF. Then, according to the geometric shape (convex or concave), the algorithm adjusts weight vectors and decides whether or not to allocate computational resources to unsolved subproblems. In MOEA/D-TPN, it is important to identify extreme solutions and add solutions in extreme regions.

In RVEA (Cheng et al. 2016), the angle-penalized distance (APD) is designed dynamically to balance convergence and diversity in MaOPs. The algorithm focus on convergence in the early stage and puts the main position to improve the diversity in the late stage of the optimization. During the optimization, the distribution of the reference vectors is adjusted according to the ranges of different objective functions. RVEA normalizes the reference vectors rather than the objective functions to ensure a uniform distribution of the candidate solutions in the objective space. However, a possible situation is that more than one reference vector associates with a same candidate solution, which may cause the sparse areas in objective space.

Pareto-adaptive weight vectors ($pa\lambda$) MOEA/D (Jiang et al. 2011) was proposed to adjust uniform automatically according to geometrical characteristics of PF. The initial weight vectors of $pa\lambda$-MOEA/D are generated by mixture uniform design (MUD) to obtain an arbitrary number of weight vectors. The algorithm identifies the geometrical characteristics (concave or convex) of the Pareto front by calculating the hypervolume value, and then, weight vectors are adjusted automatically to scatter or assemble weight vectors. $pa\lambda$-MOEA/D assumes that PF is symmetric, which is a challenging problem for handling asymmetric PF.

Different from the above improvement methods, weight vectors in PICEA-w (Wang et al. 2015) are adaptively modified in a coevolutionary manner with the candidate solutions during the search. In PICEA-w, candidate solutions are ranked by each of the weighted aggregate function, and a ranking matrix is created, then the fitness of candidate solutions is calculated based on this matrix. The weights are coevolved with the candidate solutions toward an optimal distribution. This strategy makes the algorithm less sensitive to the geometry of the problem. Nevertheless, the key factor affecting the performance of the algorithm is the generation mechanism of offspring.

From another perspective, a monotonic increasing function is utilized in T-MOEA/D to transform the objective functions into a new function, which makes the curve shape of the candidate solutions close to the unit hyperplane in the original objective function space (Liu et al. 2010). The algorithm can achieve good distribution with fixed weight mode. However, the algorithm may be less effective in dealing with MaOPs and the MOPs with discrete Pareto front.

To overcome the drawback of fixed weight, many improvements have been attempted on adjusting weights adaptively in decomposition- based algorithms. However, there is less research on PF with different axis ranges, such as WFG series test functions. Therefore, to address such PFs, a self-adaptive weight vector adjustment strategy for decomposition-based multi-objective differential evolution algorithm is proposed.

3 The proposed algorithm: AWDMODE

This section proposes a novel algorithm called multi-objective differential evolution algorithm by decomposition with adaptive weight vectors (AWDMODE). In the following paragraphs, the implementation details of each component in AWDMODE will be explained step-by-step.

3.1 General framework

The pseudocode of the proposed method is shown in Algorithm 1. The AWDMODE shares a common framework that is employed by many evolutionary algorithms. In the first part, an initial population P is formed by generating N individuals randomly, then their fitness values are calculated and the reference point z^* is acquired. Next, weight vectors with three different shapes (line, concave, convex) are generated and one of them is chosen as the original weight vectors. During the process of iteration and update, an adaptive weight vector is introduced to ensure a uniform distribution on PF. Then, differential evolution operators are performed to obtain an offspring population Q and neighborhood is updated similar to that of MOEA/D-DE. Finally, elitism strategy is adopted to select N best solutions based on the weight vectors for survival. It can be seen that there are three key operators in AWDMODE: generation of weight vectors (line 4), adaptive adjustment of weight vectors (line 8) and selection of elite individuals (line 12). In the following subsections, the above key operators will be described in detail.

Algorithm 1 Framework of the proposed AWDMODE

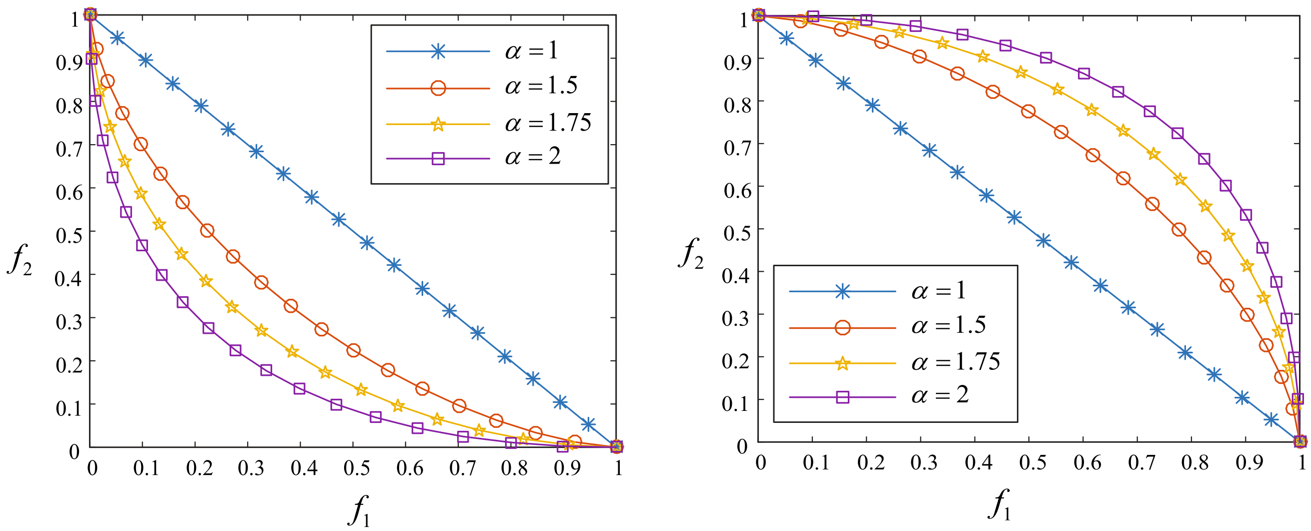
```
Input: $N, FEmax, M$
Output: $Archive$
1: Part1: Initialization
2: Initialize population $(P)$
3: Initialize the reference point $z^* = (z_1^*, z_2^*, \ldots, z_M^*)$, $z_j^* = min\ Fit_j(x)$ and $j = 1, \ldots, M$;
4: $(W, E)$ = Generatew$(N, M)$;
5: $w = W.line, neighbor = E.line, Archive = \emptyset$;
6: Part2: Iteration and Update
7: while $evaluation < FEmax$ do
8:     $w$ = Adaptivew$(W, P, t, evaluation)$;
9:     $Q$ = DEoperation$(P)$;
10:    $Q$ = Updateneighbor$(w, neighbor, P, Q)$;
11:    $S = Q \cup Archive$;
12:    $Archive$ = UpdateArchive$(S)$;
13:    $P = Archive$;
14: end while
15: Return $Archive$
```

3.2 Generate weight vector

The weight vector plays a crucial role in decomposition-based optimization algorithms, and affects the performance of the algorithm. It is worth mentioning that the evenly distributed weight vectors are not suitable for complex PFs, especially for the PFs that have a long tail and a sharp peak. Hence, three different shapes (line, concave, convex) are generated in advance at the initialization phase of the proposed algorithm, in order to solve different PFs.

Algorithm 2 (W, E) = Generatew(N, M)

```
Input: $N, M$
Output: $W, E$
Generate uniform weight vectors $weight$;
$W.line = weight$;
$W.convex = (W.line)^\alpha$;
Generate $W.concave$ based on $W.convex$;
Find $T$ nearest vectors among weights as neighborhood to each weight;
Generate three neighborhood as $E.line, E.concave, E.convex$;
Return $W, E$
```



(a) Convex weight vectors with different α

(b) The mappings of convex weight vectors

Fig. 1 The weight vectors with different α

As described in Algorithm 2, weight vectors are generated according to the method in MOEA/D, and these vectors are treated as a basic weight and called $line-weight$ in this paper (lines 1–2), i.e., $W.line = (w_1, w_2, \ldots, w_N)$. It is well-known that the weight vectors of $line-weight$ are a series of discrete points evenly distributed in $f_1 + f_2 + \cdots + f_M = 1$. Based on this, $convex-weight$ can be generated through the following equation:

$$W.convex = (W.line)^{\alpha}, \tag{6}$$

where α is a parameter to determine the curvature of the weight vectors. Then, $concave-weight$ is the symmetrical mapping of $convex-weight$ on $f_1 + f_2 + \cdots + f_M = 1$.

In Fig. 1a, there are several convex weight vectors generated by Eq. (6) with different α, where $N = 20$, $M = 2$. It can be observed from Fig. 1a, the curvature of the curve is changed with different α. The larger α, the greater the curvature. The curvature of $\alpha = 2$ is large, and the curvature of $\alpha = 1.5$ is small, which does not improve the guidance of the population significantly. Intuitively, $\alpha = 1.75$ is set in this paper, which is more suitable for distinguishing different PFs. The detailed analysis is described in Sect. 4.5.2.

3.3 Adaptive weight vector

In this study, not only the shape of the PFs, but also the scope of the PFs is considered. Weight vectors are adjusted adaptively by making full use of information provided by the current population. The detailed procedure is given in Algorithm 3.

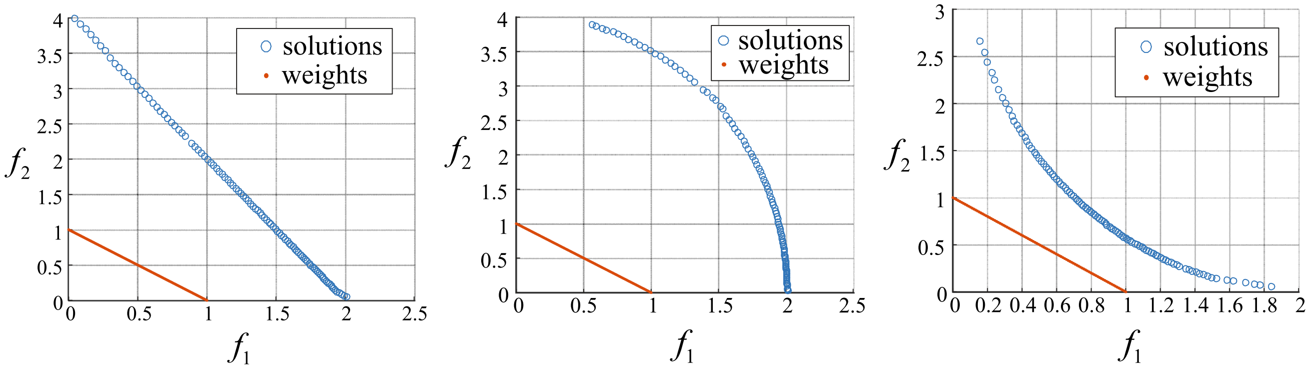
Algorithm 3 w=Adaptivew(W, P)

```
Input: $W, P, t, evaluation$
Output: $w$
1: if $(evaluation \geq 0.4 * FEmax) \wedge (t\%30 = 0)$
2:    $maxf = max(Fit)$;
3:    $Fit\_normal = normalized(Fit)$;
4:    $mean\_Fit = sum(Fit\_normal)/N$;
5:    if $mean\_Fit > \frac{1}{C} * (1 + 0.05)$
6:       $w = W.concave * maxf$;
7:    else if $\frac{1}{C} * (1 - 0.05) < mean\_Fit < \frac{1}{C} * (1 + 0.05)$
8:       $w = W.line * maxf$;
9:    else
10:      $w = W.convex * maxf$;
11:   end if
12: end if
13: Return $w$
```

When the algorithm goes to a certain degree (40% maximum function evaluation times), AWDMODE adjusts adaptively the weight vectors every 30 generations (line 1 in Algorithm 3). First of all, the shape of PF should be identified in order to choose suitable type of weight vectors. The fitness values are normalized by the following formula:

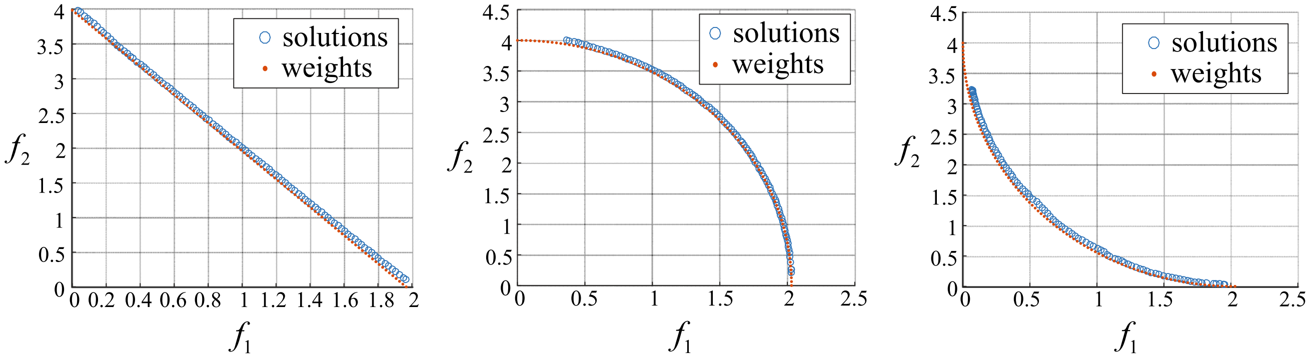
$$Fit\_normal = \frac{Fit - Fit_{min}}{Fit_{max} - Fit_{min}} \quad , \tag{7}$$

where the Fit_{max} and Fit_{min} are the maximum and minimum of the fitness values (line 3 in Algorithm 3), respectively. The mean of the Fit_normal is calculated to judge the bumpiness of the PF. It is worth noting that C is a scaling parameter and it is set to 1 in this paper because the population is normalized in the previous steps. During the optimization process, the fault tolerance is set to 5%. That means, when the average fitness value of the population (i.e., $mean\_Fit$ in Algorithm 3) is



(a) WFG3 **(b)** WFG4 **(c)** WFG10

Fig. 2 MOEA/D with original weight vectors



(a) WFG3 **(b)** WFG4 **(c)** WFG10

Fig. 3 MOEA/D with adjusted weight vectors

between $(1-0.05)$ and $(1+0.05)$, the $line-weight$ vector is adopted. If $mean\_Fit$ is greater than $(1+0.05)$, then the shape of the current population is considered as concave. If $mean\_Fit$ is less than $(1-0.05)$, then the shape of the current population is considered as convex. Then, the suitable weight vectors are chosen and adopted in the later optimization. Finally, the weight vectors are enlarged or reduced according to the maximum of the Fit.

To demonstrate the effectiveness of adjusting weights more intuitively, Figs. 2 and 3 show the results of MOEA/D on WFG3, WFG4, WFG10 with different weight vectors. In Fig. 2, the different range on each objective results in uneven distribution of the solutions. After adopting the improved weight vectors in this paper, it is obvious that the distribution is increased on different PF shapes in Fig. 3. The purpose of this study is to make weights more relevant to the PFs, so that the guidance of weight vectors becomes more accurate and effective.

3.4 Environment selection

Environmental selection mechanism is an important factor that affects the convergence or diversity of an algorithm. However, for a MOP with complex Pareto fronts, classic decomposition-based scheme cannot work well in balancing the convergence and diversity. Previous research shows that the number of non-dominated solutions in archive can reflect the progress of the algorithm to some extent (Fan et al. 2018). The completion-checking factor ξ is adopted to make up for the insufficiency of traditional decomposition-based algorithms. Algorithm 4 shows the environmental selection procedure.

In the early stages of optimization, the main task of the algorithm is to speed up the convergence so that the solution set can converge to the true PF as soon as possible. First of all, non-dominated-sorting operation is executed to measure the current process of the algorithm. When there are less non-dominated solutions in the archive, dominance-based approach is adopted to accelerate convergence. Not only the solutions in the first dominance level, but also the solutions in other dominance levels are also used to construct a new archive. The solutions at the other dominance level move to the first level until N solutions are obtained (lines 5–11 in Algorithm 4). That means the individuals at low dominance levels move one by one toward the first front (lines 2–3 in Algorithm 5). If the new solution dominates the old one,

Algorithm 4 $Archive$=UpdateArchive(S)

```
Input: $S$
Output: $archivenew$
1: $archivenew = \emptyset$;
2: $(F_1, F_2, \ldots, F_l) = Non-dominated-sorting(S)$;
3: if $| F_1 | < N * \xi$
4:    $archivenew = F_1$;
5:    for $j = 2$ to $l$
6:      if $| archivenew | < N$
7:        Execute self-learning strategy (Algorithm 5);
8:      else
9:        break
10:     end if
11:   end for
12: else
13:   $(rho, d)$=Associatetoweight($S$, $w$);
      % $rho$ is the the number of solutions associated with each weight vector, $d$ is the Euclidean distance from solutions to the weight vectors
14:   for $i = 1$ to $| w |$
15:     Execute diversity selection (Algorithm 6);
16:     $archivenew = archivenew \cup S(R)$;
17:   end for
18: end if
19: Return $archivenew$
```

then the new solution survives(lines 4–6 in Algorithm 5). Through this self-learning method between individuals, the convergence rate of the entire population can be accelerated. The detailed steps are reflected in Algorithm 5.

Algorithm 5 Self-learning strategy

```
Input: $F, S, archivenew, j$
Output: $archivenew$
1: for each $x_k \in F_j$
2:   $model = rand(F_1)$;
3:   $y = x_k + rand * (model - x_k)$;
4:   if $y$ dominates $x_k$
5:     $archivenew = archivenew \cup \{y\}$;
6:   end if
7: end for
8: Return $archivenew$
```

In the second phase of the algorithm, the main task is improving distribution. Every solution is associated with weight vectors according to the distance just like that in NSGAIII (Deb and Jain 2014) (line 13 in Algorithm 4). For each weight vector, there is one solution which associates with it to ensure the distribution of the algorithm (lines 14–17 in Algorithm 4). For an arbitrary weight vector w_i, there are three scenarios. The first scenario is that only one solution is associated with w_i. In this case, this solution is selected as a candidate solution and added to the archive (lines 1–2 in Algorithm 6). In the second case, there are more than one solution associated with w_i. At this time, among the solutions associated with w_i, the solution with the smallest aggregation function value (g^{te}) is selected as a candidate solution and added to the archive (lines 3–4 in Algorithm 6). If there is more than one solution with the smallest aggregation function value, one of the solutions is chosen at random. The last case is that there is no solution associated with w_i. At this time, the solution closest to w_i is selected as a candidate solution and added to the archive (lines 5–6 in Algorithm 6). It should be noted that at the beginning of the second phase, some dominating solutions may be retained in the archive.

Algorithm 6 Diversity selection

```
Input: $rho, d, i$
Output: $R$
1: if $rho(i) = 1$
2:   $R = rho(i)$;
     % R is the index of the solution who is associated with $w_i$
3: else if $rho(i) > 1$
4:   $R = argmin_{R \in rho(i)} g^{te}(x_R \mid w_i, z^*)$;
     % R is the index of the solution that associate with $w_i$ and with min $g^{te}$
5: else if $rho(i) = 0$
6:   $R = argmin\ d(i)$;
     % R is the index of the solution with min(d(i))
7: end if
8: Return $R$
```

Actually, dominance-based approach can work well when the number of objectives is small. As the number of objectives increases, the performance of the algorithm plummets. Therefore, ξ is set by Eq. (8):

$$\xi = \frac{1}{(M-1)^2}, \tag{8}$$

where M is the number of objectives. More details of the sensitivity of parameter of ξ can be found in Sect. 4.5.1.

4 Experimental studies

To benchmark the performance of AWDMODE, four competitor MOEAs are considered: NSGAII (Deb et al. 2002), R2MOPSO (Li et al. 2015a), RVEA (Cheng et al. 2016) and MOEA/D-STM (Li et al. 2014). Besides, in order to reflect the effectiveness of adaptive weighting strategies, AWDMODE with the original weight (the weight vectors adopted in MOEA/D (Zhang and Li 2007) is also compared in the following experiment, which is denoted as AWDMODE0.

4.1 Test instance

WFG1-WFG10 (Huband et al. 2006) are used in this study. The PFs of WFG test problems are irregular, being discontinued or mixed, and are scaled with different ranges in each objective. The WFG parameter k (position parameter) is set to 18 and the number of decision variables D is set to 32, i.e., the WFG parameter l (distance parameter) is 14, where $l = D - k$ for each test instance. In this work, 2-, 3- and 4-objective of these problems are focused.

4.2 Performance metric

In this paper, the widely used performance metric hypervolume (HV) (Zitzler and Thiele 1999) is adopted to evaluate the performance of all compared algorithms, which can assess convergence and diversity of obtained solutions. Pareto solution set with higher HV value has better convergence and diversity. Hypervolume metric is computed as follows:

$$HV(S) = Leb(\cup_{x \in S}[f_1(x), R_1] \times \cdots \times [f_M(x), R_M]), \tag{9}$$

where set S is the obtained non-dominated solutions, Leb(S) is the Lebesgue measure of S. $R = (R_1, \ldots, R_M)^T$ is a reference point which is dominated by any point in the set S. In this experiment, R is set to $(3, \ldots, 2M + 1)^T$ for M-objective test instances.

4.3 Parameter settings

Each algorithm is performed for 20 runs independently on each test instance, the detailed parameter settings are summarized as follows.

Reproduction operators: The mutation probability $pm = 1/D$, distribution index $\eta_m = 20$. For the DE operator, probability is used to select in the neighborhood: $\delta = 0.9$, $CR = 1.0$, and $F = 0.5$ as recommended in MOEA/D-DE (Li and Zhang 2009).

Population size and stopping condition: $N = 100$ for bi-objective test instances, 210 for the three-objective ones and 286 for the four-objective ones. Function evaluations, i.e., $FEs = 100000, 210000, 286000$ for 2-, 3-,4-objective instances respectively.

Neighborhood size: T = 20.

4.4 Experimental results

Tables 1, 2 and 3 show the mean and standard deviation (SD) of HV performance metric for six algorithms, where AW refers to AWDMODE, AW0 represents the algorithm with the original weight vectors AWDMODE0 and STM represents MOEA/D-STM. In these tables, the best results of the mean or standard deviation for each test function are marked as bold face. In addition, the Wilcoxon rank sum test is adopted to compare the results obtained by AWDMODE and those by five compared algorithms at a significance level of 0.05, which examined that whether the HV mean values obtained by AWDMODE are different from that obtained by the other algorithms statistically. Namely, when p value (p) is bigger than 0.05, it means that the compared HV values are similar in statistics. The bold italics in tables show that there is no significant difference between AWDMODE and the contrastive algorithms. In these tables, the $better/similar/worse$ $(+/=/-)$ denote that the performance of AWDMODE is significantly better than, equivalent to, or worse than that of the compared algorithms on all test functions.

AWDMODE0 employs a set of evenly distributed weights, which faces difficulties on the problems with complex PF whose geometry is not similar to a hyperplane. The distribution of solutions for those problems is poor because the performance of AWDMODE0 is guided by the evenly distributed weights.

For bi-objective optimization problems, as shown in Table 1, AWDMODE is the most effective algorithm which obtains the most of the best results. RVEA performs very competitively to AWDMODE and it performs best on WFG1, WFG4 and WFG10. NSGAII performs well when the number of objectives is small. The performance of R2MOPSO is poor than that of its competitors. It may result from the update mechanism of the particle swarm algorithm is not suitable for dealing with complex PF like WFG test instances. The proposed AWDMODE obtains significantly better performance than the other algorithms on WFG2, WFG3 and WFG7-9, and shows clear improvements over AWDMODE0, R2MOPSO and MOEA/D-STM on almost all of 10 test instances.

Table 2 gives results of all the WFG test instances with three-objective optimization problems. It is clear that AWDMODE performs best, presenting a clear advantage over the other five algorithms on the majority of the test instances. AWDMODE performs best on WFG3, WFG5-9 comparing with other algorithms, while RVEA outperforms other algorithms on WFG1, WFG2 and WFG10. It is demonstrated that AW0, R2MOPSO and MOEA/D-STM obtain relatively poor performance compared with their counterparts. This phenomenon may be due to the particularity of the WFG series test problems. Uniform distribution of weight vectors cannot guarantee a good distribution of the obtained solutions on true PF. Obviously, performance of NSGAII in three-objective optimization problems is worse than that in the bi-objective. The reason for this can be the reduction in selection pressure mentioned in the introduction. On WFG5, WFG6 and WFG10, the final solutions by four algorithms are plotted in Fig. 4. It is evident that AWDMODE obtains the best distribution and is able to approximate the boundary parts well.

As can be seen from Table 3, NSGAII obtains weaker performance than AWDMODE and RVEA. This verifies the fact to some extent that the performance of the dominance-

Table 1 *HV* values on WFG1-10 ($M = 2$)

	AW	AW0	NSGAII	R2MOPSO	RVEA	STM
WFG1						
Mean	3.381E−01	3.267E−01	4.530E−01	3.483E−01	**6.317E−01**	1.901E−01
SD	1.459E−02	1.681E−02	1.352E−02	**1.616E−03**	4.479E−02	1.927E−02
p		***6.992E−02***	6.796E−08	1.348E−03	6.796E−08	6.796E−08
WFG2						
Mean	**6.235E−01**	5.905E−01	6.155E−01	5.367E−01	6.006E−01	7.132E−02
SD	3.510E−02	9.394E−03	2.838E−03	1.015E−02	3.240E−02	**6.674E−04**
p		1.065E−07	1.625E−03	6.796E−08	6.796E−08	6.786E−08
WFG3						
Mean	**7.098E−01**	6.028E−01	6.962E−01	5.096E−01	6.969E−01	9.792E−02
SD	7.968E−03	1.248E−02	3.320E−03	7.171E−03	3.548E−02	**2.055E−03**
p		6.786E−08	3.983E−06	6.786E−08	6.786E−08	6.786E−08
WFG4						
Mean	5.030E−01	4.605E−01	5.233E−01	4.142E−01	**5.568E−01**	7.967E−02
SD	6.021E−03	9.961E−03	1.760E−02	7.901E−03	5.229E−02	**2.197E−03**
p		6.786E−08	7.825E−05	6.786E−08	2.353E−06	6.776E−08
WFG5						
Mean	4.879E−01	4.692E−01	**5.283E−01**	4.126E−01	5.196E−01	2.015E−01
SD	1.244E−02	1.426E−02	5.806E−03	1.092E−02	4.193E−02	**3.965E−03**
p		3.382E−04	6.796E−08	6.796E−08	1.576E−06	6.796E−08
WFG6						
Mean	4.967E−01	4.595E−01	**5.292E−01**	5.034E−01	5.242E−01	1.797E−01
SD	7.293E−03	9.295E−03	7.706E−03	5.298E−03	3.427E−02	**2.434E−03**
p		6.796E−08	6.796E−08	3.336E−03	6.796E−08	6.796E−08
WFG7						
Mean	**5.176E−01**	4.282E−01	4.763E−01	4.069E−01	4.333E−01	4.306E−02
SD	**5.883E−03**	8.700E−03	1.980E−02	6.456E−03	2.276E−02	8.340E−03
p		6.796E−08	1.657E−07	6.796E−08	6.796E−08	6.796E−08
WFG8						
Mean	**4.684E−01**	3.505E−01	4.147E−01	2.752E−01	4.175E−01	1.289E−01
SD	1.396E−02	2.448E−02	2.424E−02	6.903E−03	3.305E−02	**4.164E−03**
p		6.796E−08	3.939E−07	6.796E−08	6.796E−08	6.796E−08
WFG9						
Mean	**5.026E−01**	4.650E−01	4.960E−01	4.678E−01	4.858E−01	1.686E−01
SD	1.292E−02	1.583E−02	2.099E−02	**3.371E−03**	2.920E−02	8.805E−03
p		1.376E−06	2.853E−02	9.173E−08	2.563E−07	6.796E−08
WFG10						
Mean	8.288E−01	8.062E−01	8.700E−01	7.548E−01	**8.744E−01**	3.023E−01
SD	4.885E−03	4.957E−03	**3.835E−03**	6.413E−03	1.604E−02	9.950E−03
p		6.796E−08	6.796E−08	6.796E−08	6.796E−08	6.786E−08
$+/=/-$		9/1/0	5/0/5	8/0/2	5/0/5	10/0/0

based algorithms declines in solving MaOPs. Figure 5 depicts the parallel coordinate plots, where the horizontal axis represents the number of objective functions and the vertical axis denotes the value of the objective functions. The different colors in these figures represent different solutions. It can be observed from Fig. 5 that AWDMODE gains good distribution along the Pareto front on WFG1, WFG6 and WFG8.

To compare the convergence speed of the algorithms, the evolution of the mean HV-metric values obtained by five algorithms is plotted in Figs. 6, 7 and 8. Obviously, all the algorithms obtain a fast convergence performance

Table 2 *HV* values on WFG1-10 ($M = 3$)

	AW	AW0	NSGAII	R2MOPSO	RVEA	STM
WFG1						
Mean	5.258E−01	4.980E−01	4.682E−01	4.710E−01	**5.835E−01**	3.121E−01
SD	1.262E−02	4.127E−03	7.865E−03	**3.511E−03**	1.055E−02	3.084E−02
p		6.796E−08	6.796E−08	6.796E−08	6.796E−08	6.796E−08
WFG2						
Mean	7.709E−01	7.156E−01	6.397E−01	7.363E−01	**8.081E−01**	5.268E−02
SD	6.185E−02	7.205E−02	1.157E−01	1.140E−02	**2.811E−03**	4.242E−03
p		2.594E−05	2.594E−05	1.227E−03	1.915E−07	6.786E−08
WFG3						
Mean	**8.058E−01**	7.630E−01	7.560E−01	6.052E−01	7.972E−01	1.064E−01
SD	6.366E−03	6.013E−03	4.303E−03	9.396E−03	4.880E−03	**2.574E−03**
p		6.796E−08	6.796E−08	6.796E−08	5.896E−05	6.796E−08
WFG4						
Mean	7.116E−01	6.660E−01	6.468E−01	5.485E−01	**7.725E−01**	1.010E−01
SD	**2.817E−03**	5.960E−03	9.732E−03	8.026E−03	8.902E−03	3.536E−03
p		6.796E−08	6.796E−08	6.796E−08	6.796E−08	6.796E−08
WFG5						
Mean	**7.416E−01**	6.720E−01	6.442E−01	5.423E−01	7.088E−01	2.066E−01
SD	**3.369E−03**	4.034E−03	8.307E−03	8.497E−03	4.963E−03	4.115E−03
p		6.796E−08	6.796E−08	6.796E−08	6.796E−08	6.796E−08
WFG6						
Mean	**7.356E−01**	6.620E−01	6.340E−01	7.216E−01	7.036E−01	1.667E−01
SD	2.166E−02	1.121E−02	1.779E−02	4.261E−03	1.754E−02	**3.215E−03**
p		6.796E−08	6.796E−08	***9.620E−02***	1.807E−05	6.796E−08
WFG7						
Mean	**7.465E−01**	6.608E−01	6.591E−01	5.904E−01	7.406E−01	7.163E−02
SD	2.286E−02	1.100E−02	3.073E−02	1.243E−02	**5.002E−03**	1.041E−02
p		6.796E−08	6.796E−08	6.796E−08	4.205E−02	6.796E−08
WFG8						
Mean	**7.023E−01**	6.124E−01	5.930E−01	4.900E−01	6.129E−01	9.886E−02
SD	9.936E−03	1.836E−02	2.310E−02	1.470E−02	1.813E−02	**3.738E−03**
p		6.796E−08	6.796E−08	6.796E−08	6.796E−08	6.796E−08
WFG9						
Mean	**6.795E−01**	6.578E−01	5.721E−01	6.348E−01	6.724E−01	1.549E−01
SD	1.060E−02	6.404E−03	7.513E−03	5.981E−03	**3.942E−03**	7.890E−03
p		1.918E−07	6.796E−08	6.796E−08	***1.556E−01***	6.786E−08
WFG10						
Mean	9.431E−01	9.229E−01	9.657E−01	8.538E−01	**9.704E−01**	4.041E−01
SD	**1.985E−03**	3.013E−03	2.812E−03	6.285E−03	3.127E−03	6.398E−03
p		6.786E−08	6.786E−08	6.786E−08	6.786E−08	6.786E−08
+/ = /−		10/0/0	9/0/1	9/1/0	5/1/4	10/0/0

on the test functions. It can be observed that AWDMODE performs remarkably better than other algorithms on the most of test problems. These results show clearly that AWDMODE is capable to solve the problems with complicated PF.

4.5 Parameter analysis

4.5.1 Sensitivity of performance to the parameter ξ

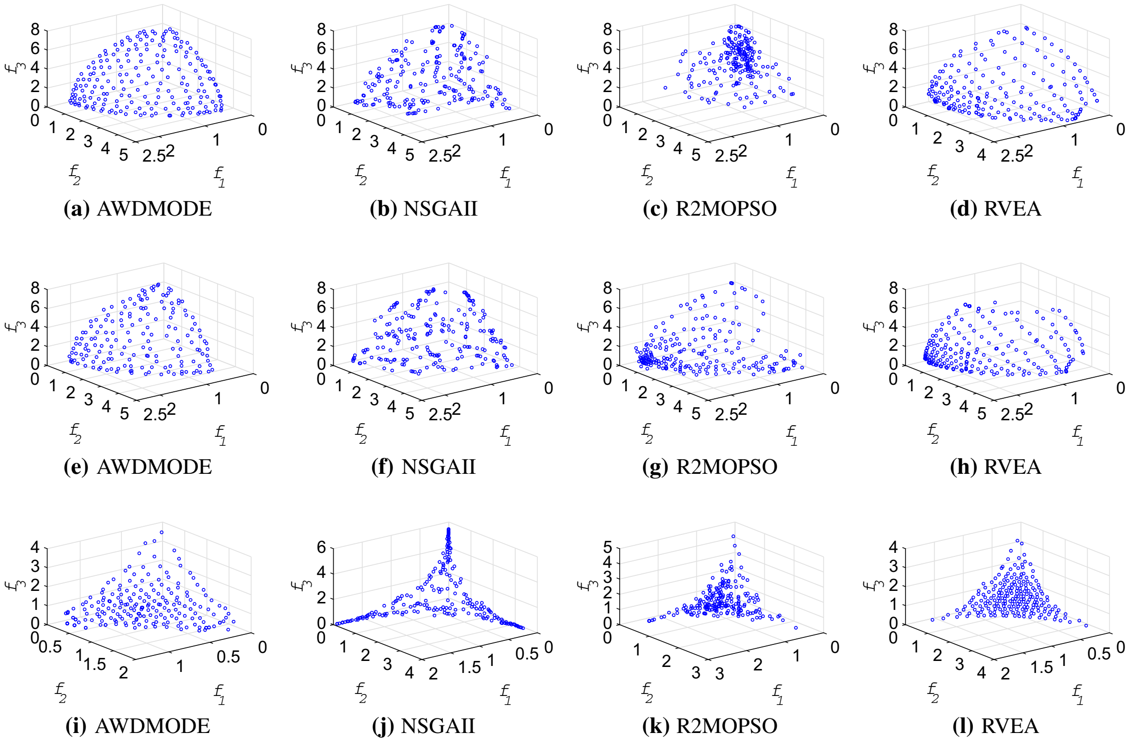
As mentioned above, the performance of dominance-based algorithms plummets with the number of objectives increases.

Table 3 HV values on WFG1-10 ($M = 4$)

	AW	AW0	NSGAII	R2MOPSO	RVEA	STM
WFG1						
Mean	4.994E−01	4.963E−01	5.178E−01	4.772E−01	**5.462E−01**	3.539E−01
SD	3.702E−03	3.469E−03	6.959E−03	**2.103E−03**	7.094E−03	2.286E−02
p		1.143E−02	1.429E−07	6.786E−08	6.786E−08	6.786E−08
WFG2						
Mean	8.103E−01	8.009E−01	8.013E−01	7.835E−01	**8.532E−01**	1.122E−01
SD	**1.152E−03**	2.771E−03	5.014E−03	1.504E−02	6.963E−02	8.434E−03
p		6.786E−08	6.908E−07	1.200E−06	6.786E−08	6.786E−08
WFG3						
Mean	8.437E−01	8.340E−01	**8.706E−01**	6.454E−01	8.315E−01	1.223E−01
SD	2.335E−03	**2.210E−03**	3.680E−03	1.378E−02	9.982E−03	3.389E−03
p		6.796E−08	6.796E−08	6.796E−08	2.443E−05	6.796E−08
WFG4						
Mean	7.808E−01	7.511E−01	7.713E−01	6.363E−01	**8.330E−01**	1.159E−01
SD	4.397E−03	6.673E−03	1.067E−02	9.368E−03	1.070E−02	**3.553E−03**
p		6.796E−08	3.966E−03	6.796E−08	6.796E−08	6.796E−08
WFG5						
Mean	**8.020E−01**	7.491E−01	7.698E−01	5.997E−01	7.808E−01	2.319E−01
SD	7.845E−03	**3.949E−03**	8.152E−03	1.162E−02	9.172E−03	5.687E−03
p		6.786E−08	6.796E−08	6.786E−08	6.917E−07	6.786E−08
WFG6						
Mean	**8.323E−01**	7.786E−01	7.719E−01	8.100E−01	8.066E−01	1.645E−01
SD	8.043E−03	9.889E−03	2.041E−02	**2.585E−03**	9.017E−03	4.375E−03
p		6.796E−08	6.796E−08	7.898E−08	2.218E−07	6.796E−08
WFG7						
Mean	**8.287E−01**	7.642E−01	8.000E−01	6.879E−01	8.185E−01	9.879E−02
SD	1.193E−02	6.780E−03	**7.197E−03**	1.055E−02	3.211E−02	1.266E−02
p		6.796E−08	6.015E−07	6.796E−08	5.075E−01	6.796E−08
WFG8						
Mean	**7.416E−01**	6.963E−01	7.030E−01	5.810E−01	6.831E−01	8.688E−02
SD	1.206E−02	8.348E−03	1.220E−02	1.011E−02	1.734E−02	**3.491E−03**
p		6.796E−08	6.796E−08	6.796E−08	6.796E−08	6.796E−08
WFG9						
Mean	**7.354E−01**	7.261E−01	7.187E−01	6.940E−01	7.349E−01	1.595E−01
SD	5.415E−03	**4.737E−03**	5.942E−03	6.372E−03	8.358E−03	1.262E−02
p		1.807E−05	7.887E−08	6.796E−08	9.892E−01	6.796E−08
WFG10						
Mean	9.448E−01	9.380E−01	9.727E−01	8.759E−01	**9.865E−01**	4.442E−01
SD	**1.747E−03**	2.284E−03	4.039E−03	5.651E−03	2.708E−03	5.215E−03
p		6.796E−08	6.796E−08	6.796E−08	6.796E−08	6.796E−08
$+/=/-$		10/0/0	7/0/3	10/0/0	6/0/4	10/0/0

Actually, they can work well when the number of objectives is small. Therefore, the completion-checking factor ξ is adopted in environmental selection mechanism, i.e., Sect. 3.4. To verify the effectiveness of this strategy, different ξ values are examined on all the WFG test problems with 2-, 3-, 4-objectives. For simplicity, function evaluations are set to $N * 700$, and the same settings for other parameters in Sect. 4.3 are used.

Tables 4, 5 and 6 collect all the HV comparison results for AWDMODE with the different ξ values. In Table 4, the algorithm performance can be seen of 2-objective test functions. At first, when the value of ξ increases, the algorithm perfor-

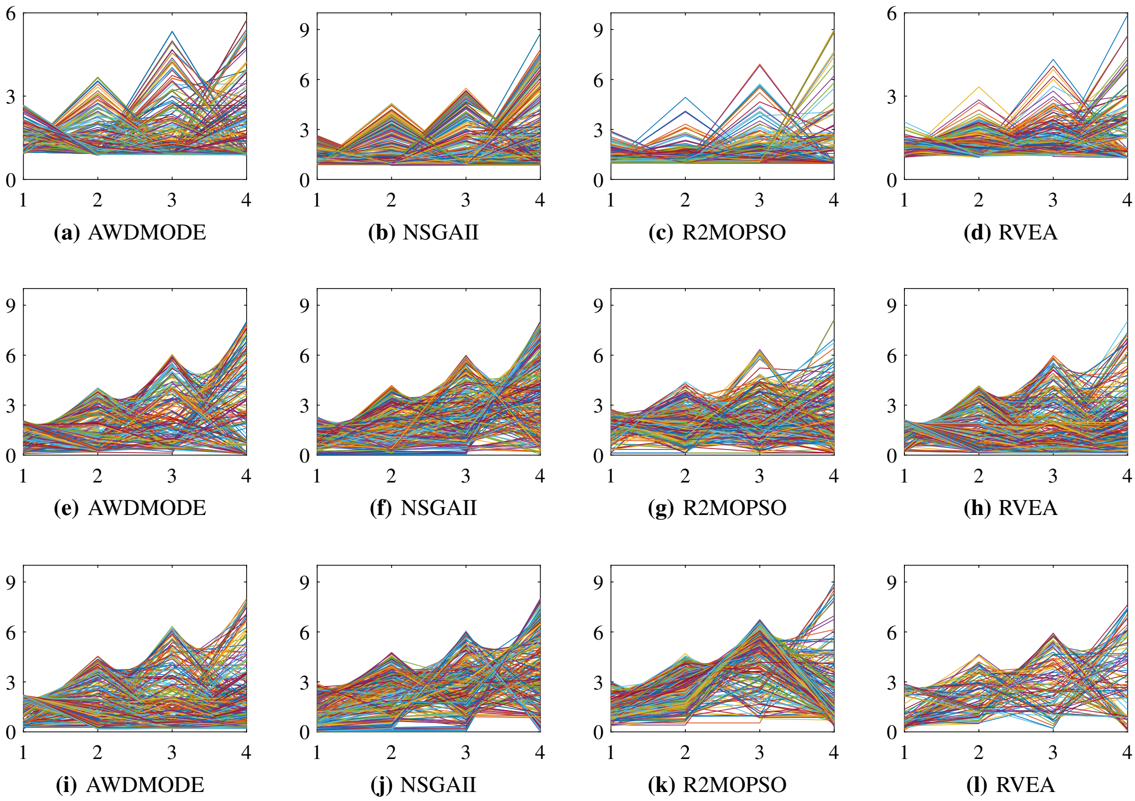


(a) AWDMODE **(b)** NSGAII **(c)** R2MOPSO **(d)** RVEA

(e) AWDMODE **(f)** NSGAII **(g)** R2MOPSO **(h)** RVEA

(i) AWDMODE **(j)** NSGAII **(k)** R2MOPSO **(l)** RVEA

Fig. 4 Pareto front on WFG5, WFG6, WFG10 with three-objective

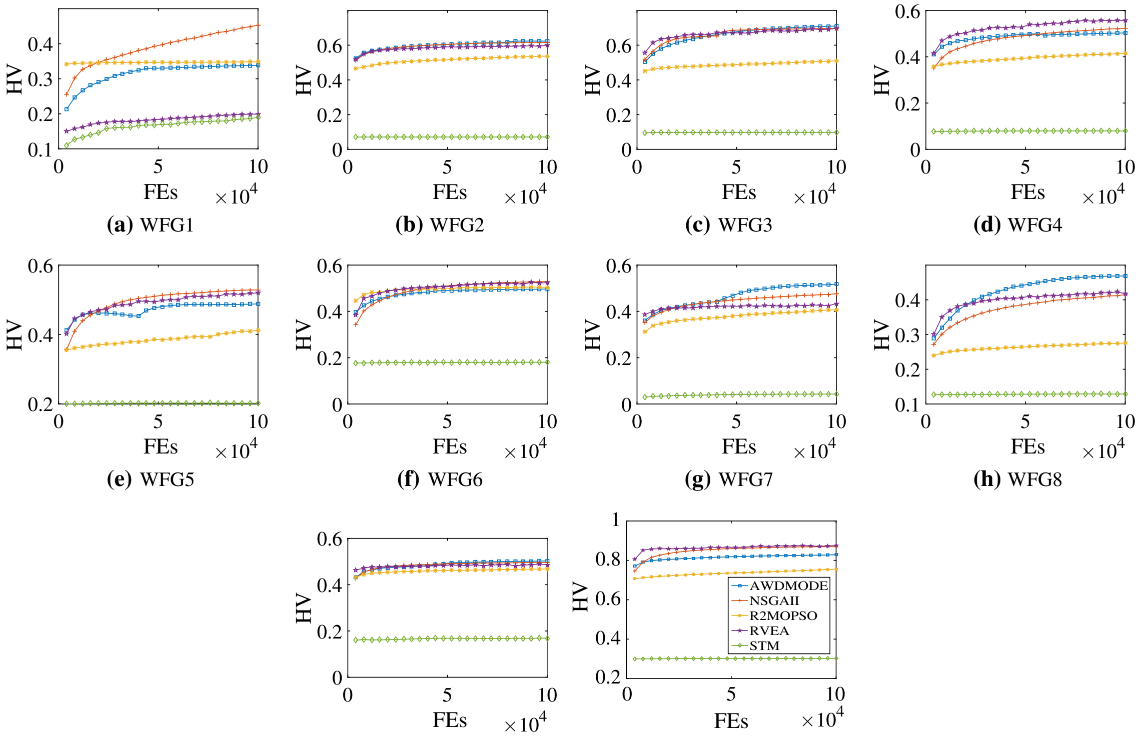


(a) AWDMODE **(b)** NSGAII **(c)** R2MOPSO **(d)** RVEA

(e) AWDMODE **(f)** NSGAII **(g)** R2MOPSO **(h)** RVEA

(i) AWDMODE **(j)** NSGAII **(k)** R2MOPSO **(l)** RVEA

Fig. 5 Pareto front on WFG1, WFG6, WFG8 with 4-objective

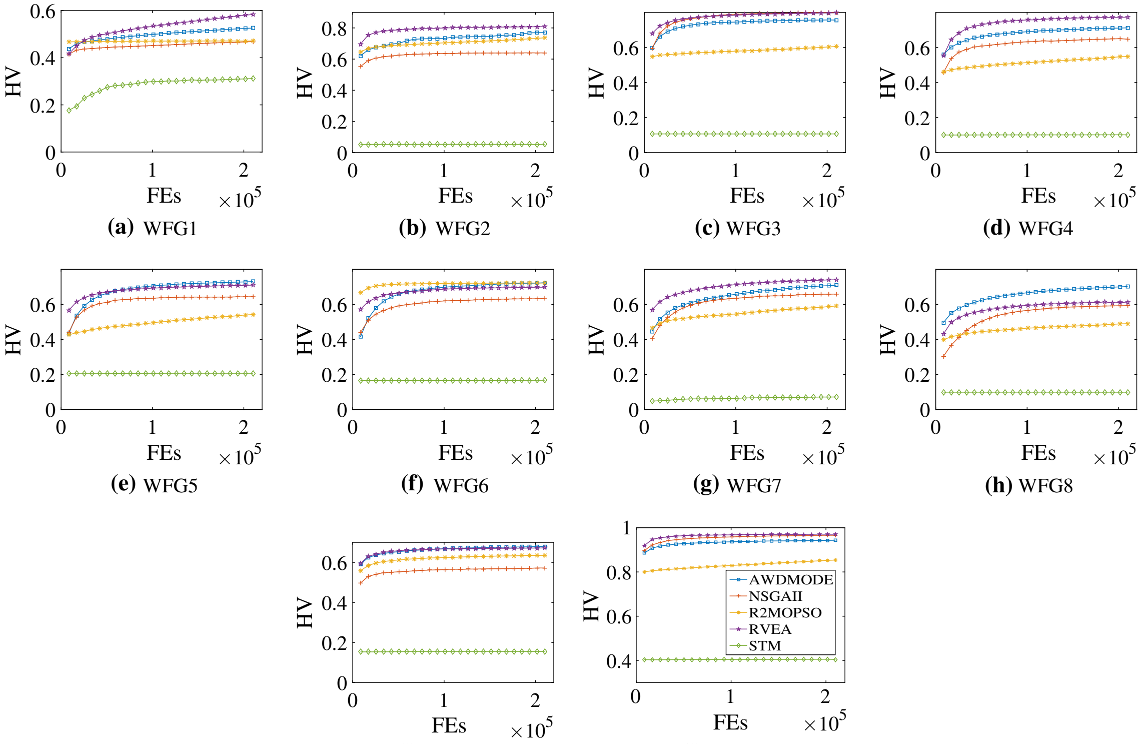


(a) WFG1 **(b)** WFG2 **(c)** WFG3 **(d)** WFG4

(e) WFG5 **(f)** WFG6 **(g)** WFG7 **(h)** WFG8

(i) WFG9 **(j)** WFG10

Fig. 6 Variation of HV-metric value on WFG1-10 with two-objective

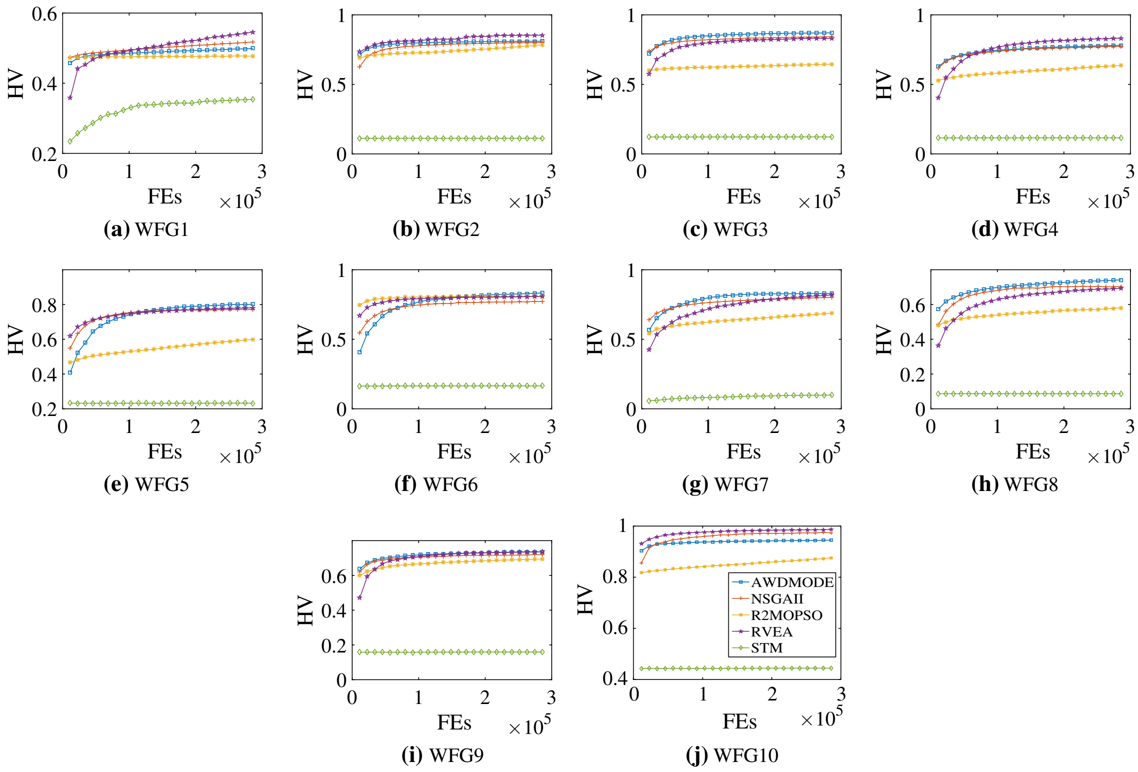


(a) WFG1 **(b)** WFG2 **(c)** WFG3 **(d)** WFG4

(e) WFG5 **(f)** WFG6 **(g)** WFG7 **(h)** WFG8

(i) WFG9 **(j)** WFG10

Fig. 7 Variation of HV-metric value on WFG1-10 with three-objective



(a) WFG1 **(b)** WFG2 **(c)** WFG3 **(d)** WFG4

(e) WFG5 **(f)** WFG6 **(g)** WFG7 **(h)** WFG8

(i) WFG9 **(j)** WFG10

Fig. 8 Variation of HV-metric value on WFG1-10 with 4-objective

Table 4 HV values on WFG1-10 with different ξ ($M = 2$)

	0.1	0.5	1	1.5
WFG1	0.3310	0.3280	**0.3386**	0.2729
WFG2	0.5776	0.5991	**0.6240**	0.5851
WFG3	0.6397	0.6450	**0.6617**	0.5964
WFG4	0.4975	0.5036	**0.5151**	0.3869
WFG5	0.4644	0.4829	**0.4917**	0.4835
WFG6	0.4530	0.4742	**0.4814**	0.4552
WFG7	0.4755	0.4989	**0.5020**	0.3803
WFG8	0.3811	0.4101	**0.4634**	0.3139
WFG9	0.4645	0.4804	**0.4840**	0.4555
WFG10	0.8235	0.8267	**0.8296**	0.7978

Best results of the mean or standard deviation for each test function are shown in bold

Table 5 HV values on WFG1-10 with different ξ ($M = 3$)

	0.05	0.25	0.6	1
WFG1	0.5052	0.5031	**0.5182**	0.5001
WFG2	0.7815	**0.7881**	0.7859	0.7823
WFG3	**0.7783**	0.7757	0.7755	0.7761
WFG4	0.6728	**0.6771**	0.6704	0.6765
WFG5	0.6741	**0.6834**	0.6791	0.6776
WFG6	0.6617	0.6776	0.6826	**0.6888**
WFG7	0.6789	**0.6871**	0.6821	0.6848
WFG8	0.6344	0.6297	0.6294	**0.6508**
WFG9	0.6699	**0.6758**	0.6666	0.6729
WFG10	0.9260	0.9267	0.9263	**0.9311**

Best results of the mean or standard deviation for each test function are shown in bold

mance shows a good trend, that is, the HV value increases with the increase of ξ. However, the algorithm performance degrades as ξ continues to increase. In other words, $\xi = 1$ is the best choice for 2-objective test functions. In Table 5, the HV value for $\xi = 0.25$ is clearly better than that with the other ξ values. Than means, $\xi = 0.25$ is the best choice for 3-objective test functions. In Table 6, it can be found that the performance with small value is significantly better than that of large ξ value. Thus, $\xi = 0.11$ is the most promising choice for 4-objective test functions.

As a result, it is unreasonable to set ξ to a fixed value for solving optimization problems with different objective numbers. Instead, the value of ξ should be adaptively adjusted when handling different problems. Therefore, for most of the test problems adopted in this paper, the setting of ξ can be summarized as Eq. (8).

4.5.2 Sensitivity of performance to the parameter α

As mentioned in Sect. 3.2, α is a parameter that controls the curvature. The larger α value, the greater the curvature. To

Table 6 HV values on WFG1-10 with different ξ ($M = 4$)

	0	0.11	0.5	1
WFG1	**0.4306**	0.4282	0.4275	0.4295
WFG2	0.7821	**0.7913**	0.7908	0.7906
WFG3	0.8014	**0.8029**	0.8027	0.8026
WFG4	0.6415	0.6378	**0.6423**	0.6418
WFG5	0.6428	0.6439	0.6441	**0.6483**
WFG6	0.6512	0.6609	**0.6647**	0.6615
WFG7	0.6532	0.6560	0.6563	**0.6587**
WFG8	**0.5529**	0.5520	0.5512	0.5500
WFG9	0.6193	**0.6221**	0.6171	0.6172
WFG10	0.9235	**0.9244**	0.9227	0.9234

Best results of the mean or standard deviation for each test function are shown in bold

obtain a more appropriate value, different α values on WFG test suites are executed with 2-, 3-, 4-objective. The same settings for other parameters in Sect. 4.3 are used. Due to the small difference in values, the further statistical analysis is conducted to facilitate more intuitive observation of the results.

Figure 9 plots the optimal results of different α values on WFG functions for HV values. For example, $\alpha = 2$ gains the best result on 2-objective WFG1, $\alpha = 1$ obtains the best performance on 3- and 4-objective WFG1. It can be clearly observed that the performance of $\alpha = 1$ and $\alpha = 1.75$ is outstanding on most issues.

Table 7 records the comparison summary of AWDMODE on the WFG test problems with different α values. The best performance is outperformed when $\alpha = 1.75$, which indicates 1.75 is an appropriate value to adopt in this paper.

Table 7 The comparison summary of AWDMODE on the WFG test problems with different α values

α	1	1.5	1.75	2
$M = 2$	2/10	3/10	3/10	2/10
$M = 3$	4/10	1/10	5/10	0/10
$M = 4$	4/10	1/10	5/10	0/10
Best/all	10/30	5/30	13/30	2/30

5 Conclusions and future work

In this paper, a self-adaptive weight vector adjustment strategy for decomposition-based multi-objective differential evolution algorithm is proposed, termed as AWDMODE. In AWDMODE, weight vectors with different shapes (line, concave, convex) are predefined. Then, the weight vector is adaptively adjusted according to the shape of the PFs. Besides, a parameter ξ is introduced to balance the convergence and diversity of the algorithm, and self-learning strategy is adopted to accelerate the convergence rate. Four state-of-the-art algorithms are compared with AWDMODE, namely NSGAII (dominance-based), R2MOPSO (indicator-based), RVEA and MOEA/D-STM. Experimental studies are performed on WFG1-WFG10 with 2-, 3-, 4-objective problems, respectively. In order to verify the effectiveness of adaptive weight adjustment strategy, the algorithm with the original weight vectors also participates in the experiment, termed AWDMODE0. Experimental results indicate that AWDMODE can obtain well-converged and evenly distributed approximated PFs for MOPs with complex PFs, especially with irregular PFs or the PFs with different ranges in each objective.

Although AWDMODE shows great potential in handling complicated test instances, there are still some limitations. In our future work, we may concentrate on dealing with the PFs with discontinuous regions and reducing the computational cost of adjusting weight vectors in MaOPs.

Acknowledgements The authors would like to thank the editor and reviewers for their helpful comments and suggestions to improve the quality of this paper. This work was supported by the Natural Science Foundation of Hebei Province (No. F2016203249).

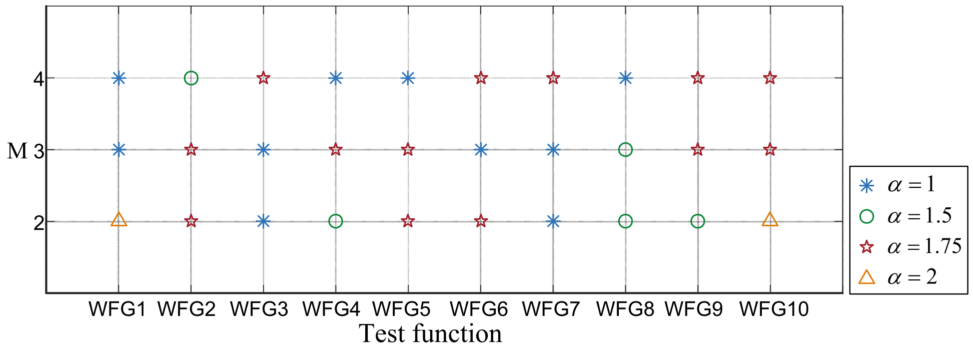


Fig. 9 The optimal results of different α values on all WFG functions for HV values

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

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

Cheng R, Jin Y, Olhofer M, Sendhoff B (2016) A reference vector guided evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 20(5):773–791. https://doi.org/10.1109/TEVC.2016.2519378

Deb K, Jain H (2014) 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):577–601. https://doi.org/10.1109/TEVC.2013.2281535

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197. https://doi.org/10.1109/4235.996017

Fan R, Wei L, Li X, Hu Z (2018) A novel multi-objective PSO algorithm based on completion-checking. J Intell Fuzzy Syst 34(1):321–333. https://doi.org/10.3233/JIFS-171291

Fan R, Wei L, Sun H, Hu Z (2019) An enhanced reference vectors-based multi-objective evolutionary algorithm with neighborhood-based adaptive adjustment. Neural Comput Appl. https://doi.org/10.1007/s00521-019-04660-5

Gee SB, Arokiasami WA, Jiang J, Tan KC (2016) Decomposition-based multi-objective evolutionary algorithm for vehicle routing problem with stochastic demands. Soft Comput 20(9):3443–3453. https://doi.org/10.1007/s00500-015-1830-2

Hu Z, Yang J, Sun H, Wei L, Zhao Z (2017) An improved multi-objective evolutionary algorithm based on environmental and history information. Neurocomputing 222(C):170–182. https://doi.org/10.1016/j.neucom.2016.10.014

Hu Z, Wei Z, Sun H, Yang J, Wei L (2019a) Optimization of metal rolling control using soft computing approaches: a review. Arch Comput Methods Eng 10:1–17. https://doi.org/10.1007/s11831-019-09380-6

Hu Z, Yang J, Cui H, Wei L, Fan R (2019b) MOEA3D: a MOEA based on dominance and decomposition with probability distribution model. Soft Comput 23(4):1219–1237. https://doi.org/10.1007/s00500-017-2840-z

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

Ishibuchi H, Setoguchi Y, Masuda H, Nojima Y (2017) Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes. IEEE Trans Evol Comput 21(2):169–190. https://doi.org/10.1109/TEVC.2016.2587749

Jiang S, Yang S (2016) An improved multiobjective optimization evolutionary algorithm based on decomposition for complex pareto fronts. IEEE Trans Cybern 46(2):421–437. https://doi.org/10.1109/TCYB.2015.2403131

Jiang S, Cai Z, Zhang J, Ong YS (2011) Multiobjective optimization by decomposition with pareto-adaptive weight vectors. In: 2011 seventh international conference on natural computation (ICNC). IEEE, vol 3, pp 1260–1264. https://doi.org/10.1109/ICNC.2011.6022367

Lee LH, Chew EP, Yu Q, Li H, Liu Y (2014) A study on multi-objective particle swarm optimization with weighted scalarizing functions. In: Proceedings of the 2014 winter simulation conference. IEEE Press, pp 3718–3729.https://doi.org/10.1109/WSC.2014.7020200

Li H, Zhang Q (2009) Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. IEEE Trans Evol Comput 13(2):284–302. https://doi.org/10.1109/TEVC.2008.925798

Li K, Zhang Q, Kwong S, Li M, Wang R (2014) Stable matching-based selection in evolutionary multiobjective optimization. IEEE Trans Evol Comput 18(6):909–923. https://doi.org/10.1109/TEVC.2013.2293776

Li F, Liu J, Tan S, Yu X (2015a) R2-MOPSO: a multi-objective particle swarm optimizer based on R2-indicator and decomposition. In: 2015 IEEE congress on evolutionary computation (CEC), pp 3148–3155. https://doi.org/10.1109/CEC.2015.7257282

Li K, Deb K, Zhang Q, Kwong S (2015b) An evolutionary many-objective optimization algorithm based on dominance and decomposition. IEEE Trans Evol Comput 19(5):694–716. https://doi.org/10.1109/TEVC.2014.2373386

Liu HL, Gu F, Cheung Y (2010) T-MOEA/D: MOEA/D with objective transform in multi-objective problems. In: 2010 international conference of information science and management engineering, vol 2, pp 282–285. https://doi.org/10.1109/ISME.2010.274

Liu Y, Gong D, Sun X, Zhang Y (2017) Many-objective evolutionary optimization based on reference points. Appl Soft Comput 50:344–355. https://doi.org/10.1016/j.asoc.2016.11.009

Medina MA, Das S, Coello CAC, Ramírez JM (2014) Decomposition-based modern metaheuristic algorithms for multi-objective optimal power flow—a comparative study. Eng Appl Artif Intell 32:10–20. https://doi.org/10.1016/j.engappai.2014.01.016

Nag K, Pal T, Pal NR (2015) ASMiGA: an archive-based steady-state micro genetic algorithm. IEEE Trans Cybern 45(1):40–52. https://doi.org/10.1109/TCYB.2014.2317693

Purshouse RC, Fleming PJ (2007) On the evolutionary optimization of many conflicting objectives. IEEE Trans Evol Comput 11(6):770–784. https://doi.org/10.1109/TEVC.2007.910138

Qi Y, Ma X, Liu F, Jiao L, Sun J, Wu J (2014) Moea/d with adaptive weight adjustment. Evol Comput 22(2):231–264

Trivedi A, Srinivasan D, Pal K, Saha C, Reindl T (2015) Enhanced multiobjective evolutionary algorithm based on decomposition for solving the unit commitment problem. IEEE Trans Ind Inf 11(6):1346–1357. https://doi.org/10.1109/TII.2015.2485520

Wang R, Purshouse RC, Fleming PJ (2015) Preference-inspired co-evolutionary algorithms using weight vectors. Eur J Oper Res 243(2):423–441. https://doi.org/10.1016/j.ejor.2014.05.019

Wang R, Xiong J, Ishibuchi H, Wu G, Zhang T (2017a) On the effect of reference point in MOEA/D for multi-objective optimization. Appl Soft Comput 58:25–34. https://doi.org/10.1016/j.asoc.2017.04.002

Wang Z, Zhang Q, Li H, Ishibuchi H, Jiao L (2017b) On the use of two reference points in decomposition based multiobjective evolutionary algorithms. Swarm Evol Comput 34:89–102. https://doi.org/10.1016/j.swevo.2017.01.002

Wei LX, Li X, Fan R, Sun H, Hu ZY (2018) A hybrid multiobjective particle swarm optimization algorithm based on R2 indicator. IEEE Access 6:14710–14721. https://doi.org/10.1109/ACCESS.2018.2812701

Ying S, Li L, Wang Z, Li W, Wang W (2017) An improved decomposition-based multiobjective evolutionary algorithm with a better balance of convergence and diversity. Appl Soft Comput 57:627–641. https://doi.org/10.1016/j.asoc.2017.03.041

Yuan Y, Xu H, Wang B, Zhang B, Yao X (2016) Balancing convergence and diversity in decomposition-based many-objective optimizers. IEEE Trans Evol Comput 20(2):180–198. https://doi.org/10.1109/TEVC.2015.2443001

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

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

Zitzler E, Laumanns M, Thiele L (2001) SPEA2: improving the strength pareto evolutionary algorithm. TIK-report 103

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.