Hindawi
Complexity
Volume 2021, Article ID 8870356, 15 pages
https://doi.org/10.1155/2021/8870356

Research Article

An Adaptive Reference Vector Adjustment Strategy and Improved Angle-Penalized Value Method for RVEA

Wenbo Qiu,[1] Jianghan Zhu,[1] Huangchao Yu ,[2] Mingfeng Fan,[3] and Lisu Huo[1]

[1] *College of Systems Engineering, National University of Defense Technology, Changsha 410073, Hunan, China*
[2] *Institute of Unmanned System, National University of Defense Technology, Changsha 410073, Hunan, China*
[3] *School of Traffic and Transportation Engineering, Central South University, Changsha 410073, Hunan, China*

Correspondence should be addressed to Huangchao Yu; yuhc1221@nudt.edu.cn

Received 16 September 2020; Revised 6 October 2020; Accepted 21 October 2021; Published 17 November 2021

Academic Editor: Inés P. Mariño

Decomposition-based evolutionary multiobjective algorithms (MOEAs) divide a multiobjective problem into several subproblems by using a set of predefined uniformly distributed reference vectors and can achieve good overall performance especially in maintaining population diversity. However, they encounter huge difficulties in addressing problems with irregular Pareto fronts (PFs) since many reference vectors do not work during the searching process. To cope with this problem, this paper aims to improve an existing decomposition-based algorithm called reference vector-guided evolutionary algorithm (RVEA) by designing an adaptive reference vector adjustment strategy. By adding the strategy, the predefined reference vectors will be adjusted according to the distribution of promising solutions with good overall performance and the subspaces in which the PF lies may be further divided to contribute more to the searching process. Besides, the selection pressure with respect to convergence performance posed by RVEA is mainly from the length of normalized objective vectors and the metric is poor in evaluating the convergence performance of a solution with the increase of objective size. Motivated by that, an improved angle-penalized distance (APD) method is developed to better distinguish solutions with sound convergence performance in each subspace. To investigate the performance of the proposed algorithm, extensive experiments are conducted to compare it with 5 state-of-the-art decomposition-based algorithms on 3-, 5-, 8-, and 10-objective MaF1–MaF9. The results demonstrate that the proposed algorithm obtains the best overall performance.

1. Introduction

In the real world, decision makers often encounter some problems with more than one objective to be solved simultaneously. These problems are called multiobjective optimization problems (MOPs), and if the number of objectives is larger than 3, they are termed as many-objective optimization problems (MaOPs). An MOP or MaOP can be formulated as follows:

$$\begin{aligned}&\min F(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})],\\&\text{s.t.}\,\mathbf{x} \in \Omega,\end{aligned} \tag{1}$$

where $\Omega \subset R^n$ denotes the decision space and F is a map from R^n to R^M, i.e., from the decision space to the objective space. $f_i(\mathbf{x})$ is the ith objective value of the problem and M is the number of objectives. Mostly, the objectives of an MOP or MaOP are contradictory to each other and thus there rarely exists a solution that can surpass all the other solutions on each objective. As a result, a set of solutions representing the trade-off among all the objectives, called Pareto-optimal solutions, can be achieved and pursued by researches. The Pareto-optimal solutions are known as Pareto front (PF) in the objective space and Pareto set (PS) in the decision space, respectively.

To address the multiobjective problems (MOPs) and many-objective problems (MaOPs) appearing in many real-world industries, e.g., data mining [1], airline crew roster recovery [2], and software engineering [3], multiobjective evolutionary algorithms (MOEAs) and many-objective evolutionary algorithms (MaOEAs) are seen as the major methods for approximating PF of MOPs because of its population-based nature. During the past decades, a large number of algorithms have been developed such as Nondominated Sorting Genetic Algorithm II (NSGA-II) [4], multiobjective evolutionary algorithm based on decomposition (MOEA/D) [5], and so on [6]. According to their environmental selection strategies, the existing algorithms can be roughly divided into four categories: (1) Pareto-dominance-based; (2) decomposition-based; (3) indicator-based; and (4) others. Pareto-dominance-based algorithms [4, 7, 8] often divide solutions into different nondominated levels and use a second criterion to select solutions in the last level; decomposition-based MOEAs [9–13] decompose the original MOP into multiple subproblems and solve them in a cooperative way; for indicator-based MOEAs and MaOEAs, such as hypervolume-based many-objective (HypE) [14] and indicator-based multiobjective evolutionary algorithm with reference point adaptation (AR-MOEA) [15], it tends to develop an indicator to evaluate the overall performance and sort the individuals according to their indicator values.

Among the four categories, a branch of decomposition-based MOEAs and MaOEAs divides the whole objective space into a number of subspaces by a set of predefined uniformly distributed reference vectors or reference weights, such as RVEA [16] and MOEA/D-M2M [17]. This branch shows promising performance in maintaining population diversity. However, there are still some shortcomings in this branch of algorithms. Firstly, when the MOP owns irregular PF, e.g., degenerated or disconnected, the PF is not uniformly distributed in the objective space, and thus the predefined reference vectors are not able to evenly divide the PF which may affect the final quality of the output population. The PFs of 3-objective DTLZ7 and MaF6 are shown in Figure 1 where the PF of DTLZ7 is disconnected and that of MaF6 is degenerated.

Besides, although the PFs of the above two problems are irregular, the PF of 2 or 3 objectives can still be divided to some degree of satisfaction because the objective space is not large and the predefined reference vectors can still be densely distributed in the area where the PF is located. However, with the increase of the number of objectives, the objective space expands sharply and the distribution of reference vectors in the objective space is very sparse. As a result, sometimes the PF is divided by few reference vectors. To visually describe this phenomenon, an example about the optimization of 10-objective MaF1 by RVEA is shown in Figure 2.

In RVEA, a reference vector is called active when at least one candidate solution is associated with it. Figure 2 shows the proportion of active reference vectors among all the predefined reference vectors during the searching process, and the population size is set as 275. We can see that only less than 10% of the predefined reference vectors are working.

To cope with the above issues, a new decomposition-based evolutionary algorithm is proposed in this paper. In the newly proposed algorithm, a new definition termed as angle position is developed to evaluate the position of a candidate solution in the objective space (the new algorithm is thus called AP-RVEA). Then, the reference vectors are adaptively adjusted according to the distribution of the candidate solutions on the basis of the definition of angle position. The inactive reference vectors will be deleted and some candidate solutions will be constructed into new reference vectors according to their positions relative to the active reference vectors in the objective space. In this way, the subspaces in which the true PF lies will be further divided.

Besides, the metric in APD value (the selection indicator used in RVEA) responsible for evaluating the convergence of solutions is the length of the normalized objective vector. This metric is poor at posing enough selection pressure in terms of convergence performance with the increase of the objective size. To fix the above issue, an improved APD value is proposed in AP-RVEA to keep promising solutions in problems with many objectives.

The main contributions of this paper can be summarized as follows:

(1) A novel adaptive reference vector adjustment strategy based on the positions of candidate solutions in the objective space relative to the current active reference vectors is proposed. With the strategy, the predefined reference vectors will be adjusted according to the distribution of promising solutions, which can be considered as an approximation of the true PF in some way. During each generation that runs the strategy, the subspaces in which the true PF lies may be further divided into more subspaces and contribute more to that final result.

(2) To cope with the poor selection pressure in terms of convergence performance posed by APD value, an improved APD value method is designed to better evaluate the convergence performance of solutions in each subspace.

The rest of this paper is organized as follows. Section 2 gives a survey of strategies about reference vector adjustment. Section 3 introduces some basic definitions of the proposed AP-RVEA. In Section 4, the details of the proposed AP-RVEA are described. Section 5 presents the experimental results of AP-RVEA and other 5 algorithms on MaF [18] test suite. Finally, the conclusion and future work are covered in Section 6.

2. Related Studies

Reference vector-based MOEAs or MaOEAs are efficient in maintaining population diversity when dealing with MOPs or MaOPs [19, 20]. However, since the shape of PF is unknown in advance for most MaOPs, using predefined reference vectors to acquire a set of evenly distributed solutions might not be feasible especially when the problem has an irregular PF. As a result, some research studies about the adjustment of reference vectors have been developed in recent years.

Jiang et al. [21] proposed a asymmetric Pareto-adaptive (apa) scheme for MOEA/D [5] to deal with problem with symmetric and asymmetric Pareto fronts. The apa scheme was useful when the PF of the problem satisfied that $f_1^p + f_2^p + \cdots + f_M^p = 1$, where p is a parameter to estimate

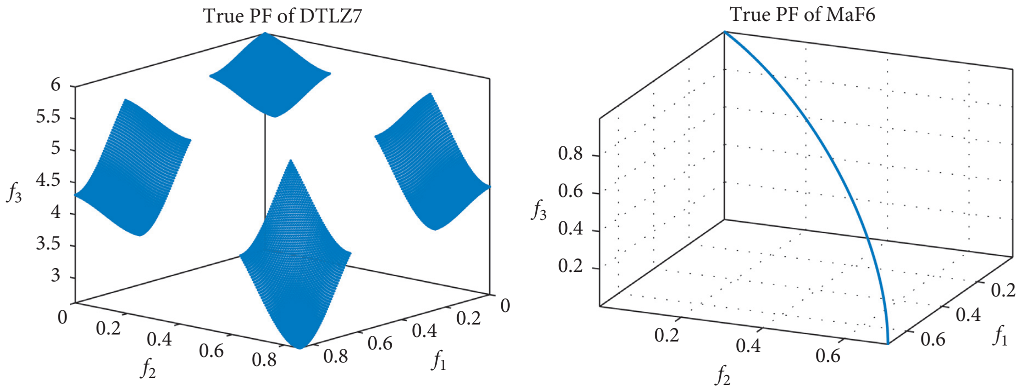


Figure 1: The true PFs of DTLZ7 and MaF6.

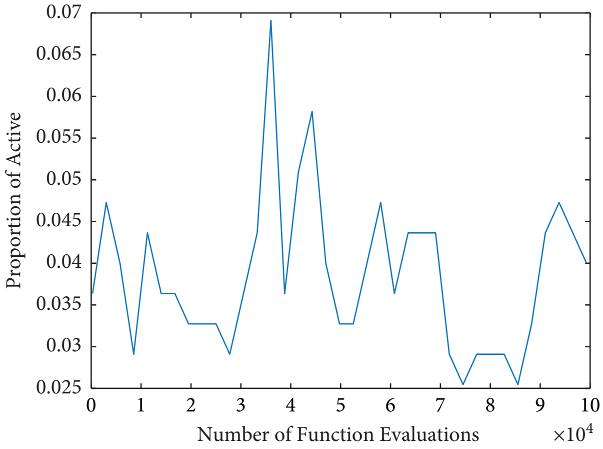


Figure 2: The proportion of active reference vectors during the optimization of 10-objective MaF1 by RVEA.

the shape of the PF and M is the number of objectives. In EMOSA [22], a set of Q evenly distributed weight vectors (reference vectors) and a set of C_{H+M-1}^{m-1} predefined candidate wight vectors P^* are maintained during the optimization process, where H denotes a positive integer and $C_{H+M-1}^{m-1} \gg Q$. For a solution x_1 (associated with λ_1), its closest nondominated neighbor x_2 (associated with λ_2) is labeled. The new weight vector λ is chosen from P^* and λ_1 is replaced if and only if (1) $\mathrm{dist}(\lambda_1, \lambda_2) < \mathrm{dist}(\lambda, \lambda_2))$ and (2) $\mathrm{dist}(\lambda, \lambda_1) \le \mathrm{dist}(\lambda, P^*)$.

In RVEA* [16], for each inactive subspace (empty subspace), the corresponding reference vector would be deleted and replaced by a unit vector that was randomly generated inside the range of the minimum and maximum value. Wang et al. [23] also proposed an algorithm called PICEA-w where weights were co-evolved with candidate solutions during the optimization process. As a result, suitable weights could be adaptively constructed and guide the candidate solution to the true Pareto front (PF) effectively.

In Qi et al. [24], an improved MOEA/D with adaptive weight vector adjustment was proposed to handle problems with irregular PFs. During the searching process, the weights were periodically adjusted to redistribute the weights of the subproblems so as to obtain better uniformity of solutions. In Jiang [25], new reference vectors were chosen from the combination of parent population and offspring according to their distances to the current reference vectors. In the first step of the algorithm, all the extreme solutions would be selected into the empty reference vector set.

Liang et al. [26] developed two reference vector adaptation strategies called scaling of reference vectors (SRV) and transformation of solutions location (TSL) for many-objective evolutionary algorithms. SRV introduced a center vector and adjusted the other reference vectors around by a scaling function; TSL transformed the promising solutions in the current generation into a set of new reference vectors. Cai et al. [27] also proposed a decomposition-based MaOEA with two types of adjustments for direction vectors where a Pareto-dominance-based mechanism was used to evaluate the effectiveness of each direction vector and the ineffective direction vectors would be adjusted to better fit the shape of the true PF.

Asafuddoula et al. [28] proposed an enhanced decomposition-based evolutionary algorithm with adaptive reference vectors called G-DBEA. In G-DBEA, two types of reference vectors called active and inactive reference vectors were maintained. During the search process, if an offspring preferred an inactive reference vector, the vector would be removed from the set of inactive reference vectors and the corresponding reference vector of the offspring would be deleted from the set of active reference vectors. At the same time, if a reference vector was not associated with any nondominated solutions in a certain period, it will be deleted from the set of active reference vectors and the solution with the smallest perpendicular distance to the reference vector would be constructed as a new reference vector.

Zhao et al. [29] proposed a modified decomposition-based many-objective ant colony optimization (ACO) algorithm and employed an adaptive reference point mechanism that chooses the ideal point or nadir point as the reference point according to the distribution of the candidate solutions. Zhou et al. [30] developed a novel entropy-based evolutionary algorithm with adaptive reference points called EARPEA where entropy computed based on reference points and a learning period are employed to control adaptation of the reference points. Wang et al. [31] gave a new preference-based MOEAs called MOEA/D-AWV in which the weight vectors are generated adaptively by the decision maker's preference and finally guide the solutions to converge to a preferred distribution.

In Deb et al. [32], reference points with no associated solution would be directly deleted and a simplex of M reference points around a remaining reference point would be added. In Cheng et al. [33], a solution that had the least similarity from active reference vectors was found and normalized, and then an inactive reference vector would be randomly deleted and replaced by the normalized solution. Liu et al. [34] used the growing neural gas network to learn the distribution of the reference vectors, thus achieving automatic yet stable adaptation.

In the proposed AP-RVEA, inactive reference vectors will be deleted after a period of generations. Besides, the newly defined angle position works to evaluate the position of solutions with promising performance against the reference vectors. Then, new reference vectors will be chosen on the basis of the information given by angle position. The core idea of the AP-RVEA is to adaptively adjust the reference vectors to further divide the subspaces where the true PF may lie. Furthermore, an improved version of APD method called F-APD based on the fractional dominance relation is developed to distinguish the convergence performance of the solutions. They cooperate with each other to achieve promising convergence and diversity and sound balance between them.

3. Background

3.1. The Objective Space Decomposition Strategy. In this section, an objective space decomposition strategy is shown to divide the objective space into a set of subspaces [17]. Each subspace owns a set of solutions and can be seen as a subproblem. In other words, a population is divided into a set of subpopulations to maintain diversity for the whole population. In each generation, all the subpopulations are optimized together.

Definition 1 (subspace). At first, N uniformly distributed unit vectors: $u^1, u^2, \ldots, u^N$, are predefined to divide the objective space into N subspaces $\Omega^1, \Omega^2, \ldots, \Omega^N$. Then, each solution will be associated with an subspace according to their acute angles between the solution and unit vectors. $\Omega^i = \{v \mid \langle v, u^i\rangle \le \langle v, u^j\rangle, \forall j \in 1, 2, \ldots, N\}$, where $\langle v, u^i\rangle$ denotes the acute angle between v and u^i. According to the definition of subspace, a vector v belongs to Ω^i, if and only if v has the smallest angle to u^i compared to other unit vectors. Each solution is to be associated with an unit vector and in a subspace during the optimization process. u^i is the corresponding reference vector of all the solutions in Ω^i and the acute angle of v to Ω^i is the acute angle of v to u^i.

Definition 2 (neighborhood subspace). Let K be the size of neighborhood spaces. For subspace Ω^i $(i = \{1, 2, \ldots, N\})$, its neighborhood subspaces are defined as $\mathrm{NS}_i = \{\Omega^{i_1}, \ldots, \Omega^{i_K}\}$, where $\forall\, \Omega^j \in \mathrm{NS}_i, \forall\, \Omega^k \notin \mathrm{NS}_i$, $\langle u^j, u^i\rangle \le \langle u^k, u^i\rangle$. In other words, the neighborhood subspaces of subspace Ω^i are the union of subspaces whose unit vectors have the first K smallest acute angles to u^i among all the unit vectors.

3.2. RVEA. On the basis of the above objective space decomposition strategy, RVEA [16] aims to maintain a sound balance between convergence and diversity for the population. It adopts the acute angles between solutions and the reference vectors and the length of the objective vectors to assess the overall performance of solutions by using the following formula called angle-penalized distance (APD):

$$d_{t,i,j} = \left(1 + P\left(\theta_{t,i,j}\right)\right) \cdot \left\| f'_{t,i} \right\|, \tag{2}$$

where $P(\theta_{t,i,j})$ and $\|f'_{t,i}\|$ measure the diversity performance and convergence performance, respectively, $\theta_{t,i,j}$ is the acute angle between the ith objective vector and u^i, and $P(\theta_{t,i,j})$ is a penalty function related to $\theta_{t,i,j}$:

$$\begin{aligned} P\left(\theta_{t,i,j}\right) &= M \cdot \left(\frac{t}{t_{\max}}\right)^{\alpha} \cdot \frac{\theta_{t,i,j}}{\gamma_{u_{t,j}}}, \\ \gamma_{u_{t,j}} &= \min\langle u^i, u^j\rangle, \quad i \in 1, 2, \ldots, N, i \ne j, \end{aligned} \tag{3}$$

where N denotes the population size and M is the objective number. $\gamma_{u_{t,j}}$ is the minimum angle between u^i and other reference vectors. The value of α controls the changing rate of $P(\theta_{t,i,j})$ and a larger α means that more emphasis will be allocated to the convergence performance compared to the diversity performance.

In the environmental selection of RVEA, a solution will be selected from each subspace, and among all the solutions in a subspace, the solution being assigned with the minimum APD method will be selected.

3.3. The Definition of Angle Position. In this paper, we try to further divide the active subspaces during the optimization of algorithms with the objective space decomposition strategy. Consequently, it is essential to know the accurate position of a solution to the active reference vectors. For this purpose, the definition of angle position is proposed.

Definition 3 (neighborhood reference vector). At first, the definition of neighborhood reference vectors of a solution is defined. Let M, the objective number, be the size of neighborhood reference vectors. For one solution x_i, its neighborhood reference vectors is defined as $\mathrm{NRV}^i = u^{i,1}, u^{i,2}, \ldots, u^{i,M}$ where for each $u^j \notin \mathrm{NRV}^i$ and each $u^k \in \mathrm{NRV}^i$, $\langle x_i, u^j\rangle < \langle x_i, u^k\rangle$. In other words, the neighborhood reference vectors of x_i are the reference vectors with the first M smallest acute angle to x_i among all the reference vectors.

Definition 4 (angle position). The definition of angle position is designed to determine the spatial position of one solution relative to the reference vectors around it. For a solution x_i, NRV^i is the set of its neighborhood reference vectors. If $M = 2$, the angle position of x_i is defined as $\mathrm{AP}_i = |\langle x_i, \mathrm{NRV}^i_1\rangle - \langle x_i, \mathrm{NRV}^i_2\rangle|$. Generally speaking, the angle position of a solution with biobjectives is the absolute value of the difference between the solution and its two neighborhood reference vectors. If $\mathrm{AP}_i = 0$, it means that the vector of the solution is in the middle of its two neighborhood reference vectors and the space between the two neighborhood reference vectors can be evenly divided by the vector of x_i. If $M > 2$, $\mathrm{AP}_i = \mathrm{Var}(\mathrm{NRV}_1^i, \mathrm{NRV}_2^i, \ldots, \mathrm{NRV}_M^i)$, the variance of the angles between X_i and its neighborhood reference vectors. Similarly, the vector of the solution with a smaller value of angle position is able to divide the space among the neighborhood reference vectors of the solution more evenly.

The definition of neighborhood reference vector is designed for a solution while the definition of neighborhood subspace is for a subspace. To visually describe the two definitions and the difference between them, a simple example is given. As can be seen in Figure 3, V_1, V_2, V_3, V_4, and V_5 are 5 reference vectors; X and Y are 2 solutions. A, B, C, and D are the angles between solutions and reference vectors. It is apparent that A is smaller than B and D is smaller than C; as a result, X is associated with V_2 and Y is associated with V_4. The neighborhood spaces of the subspace of V_2 are the corresponding subspaces of V_1 and V_3. The neighborhood reference vectors of X are V_2 and V_3. The angle position value of X is $|A - B|$ and that of Y is $|C - D|$. It is obvious that $|A - B| < |C - D|$ and the space between V_2 and V_3 can be more evenly divided by X when it is compared with the space between x and V_4 divided by Y.

4. The Proposed AP-RVEA

In this section, the details of the proposed AP-RVEA will be given. The overall framework of AP-RVEA is presented at first, followed by its main components.

4.1. The Main Framework of AP-RVEA. The pseudocode of the proposed AP-RVEA is shown in Algorithm 1. As shown in Algorithm 1, 4 parameters are first initialized in lines 1–4: (1) a set of uniformly distributed reference vectors V; (2) a population with N individuals; (3) the number of function evaluations: FEs; and (4) an archive in which elite individuals will be stored and will be updated in each generation. Then, the main loop of AP-RVEA is given in lines 5–12. In lines 6–7, the offspring population is generated and the FEs will be updated. Q is combination of the parent population and offspring population. In lines 9–10, when half of the optimization process pasts, the reference vectors will be adaptively adjusted via function Reference Vector Adjustment. The adjusted reference vector and the archive of elite individuals will be used to select promising individuals from the combined population Q. In line 12, the new archive of elite solutions will be generated according to the current archive and the new parent population P. The environmental selection strategy and the reference vectors adjustment strategy are the two main contributions of this paper and will be detailed in the following section.

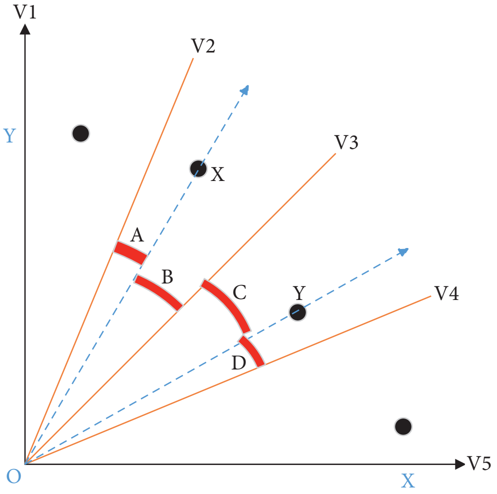


FIGURE 3: An example showing the definitions of subspace.

4.2. Reference Vector Adjustment. The pseudocode of function Reference Vector Adjustment is described in Algorithm 2. In this function, the inactive reference vectors (no solution exists in its corresponding subspace) will be deleted and new promising reference vectors will be generated from archive, the combination of elite solutions. The inputs of this function are current reference vectors V, the set of elite individuals archive, and the combine population Q. The output of this function refers to the updated reference vectors V.

At first, individuals in archive are normalized according the following equations:

$$\mathrm{archive}(i) = \frac{\mathrm{archive}(i) - Z^{\min}}{Z^{\max} - Z^{\min}}, \qquad (4)$$

where $Z^{\min} = \{Z^{\min_1}, Z^{\min_2}, \ldots, Z^{\min_M}\}$ and $Z^{\max} = \{Z^{\max_1}, Z^{\max_2}, \ldots, Z^{\max_M}\}$ and $Z^{\min_i}, i \in \{1, 2, \ldots, M\}$, and $Z^{\max_i}$, $i \in \{1, 2, \ldots, M\}$, are the minimum value and maximal value of the ith objective among archive, respectively. The normalized $\mathrm{archive}(i)$ ranges from 0 to 1. After that, new reference vector is constructed as

$$\mathrm{new}\,A(i) = \frac{\mathrm{archive}(i)}{\sum_{j=1}^{M} \mathrm{archive}(i)_j}, \qquad (5)$$

and through this transformation, $\sum_{i=1}^{M} \mathrm{new}\,A(i)_i = 1$.

Then, each individual in Q will be associated with an unit reference vector in V, i.e., each will lie in a subspace. In lines 6-7, the neighborhood reference vectors of each individual in new A are obtained and stored in NRV. In lines 8–10, the inactive reference vectors will be deleted from the current reference vectors. The addition of new reference vectors are described in lines 11–17. Since the new reference vectors are chosen from new A and the maximal number of reference vectors is N, once $|\mathrm{new}\,A|$ is 0 or $|V|$ is larger than N, the process will stop. In line 12, the AP (angle position) value of each individual in new A is initialized. In lines 13-14, the AP values of all the individuals will be calculated and stored. Among those individuals, the individual with the smallest AP value will be chosen as a new reference vector. Next, individuals

```
Input: population Size $N$; objective size $M$; the maximal function evaluations (MFEs);
Output: final population $P$;
Generate a set of unit reference vectors $V = \{v_1, v_2, \ldots, v_N\}$;
Generate a population $P \leftarrow$Initialtization$(N)$;
Initialize function evaluations FEs$\leftarrow N$;
archive$\leftarrow\varnothing$;
while FEs < MFEs do
    $P' \leftarrow$Generate Offspring;
    FEs$\leftarrow$FEs + $N$;
    $Q \leftarrow P \cap P'$;
    if (FEs/MFEs) > 50% then
    $V \leftarrow$Reference Vector Adjustment$(V, Q$, archive);
    $P \leftarrow$Environmental Selection$(V, Q, N)$;
    archive$\leftarrow$Archive Updat$(P$, archive);
```

Algorithm 1: The main procedure of AP-RVEA.

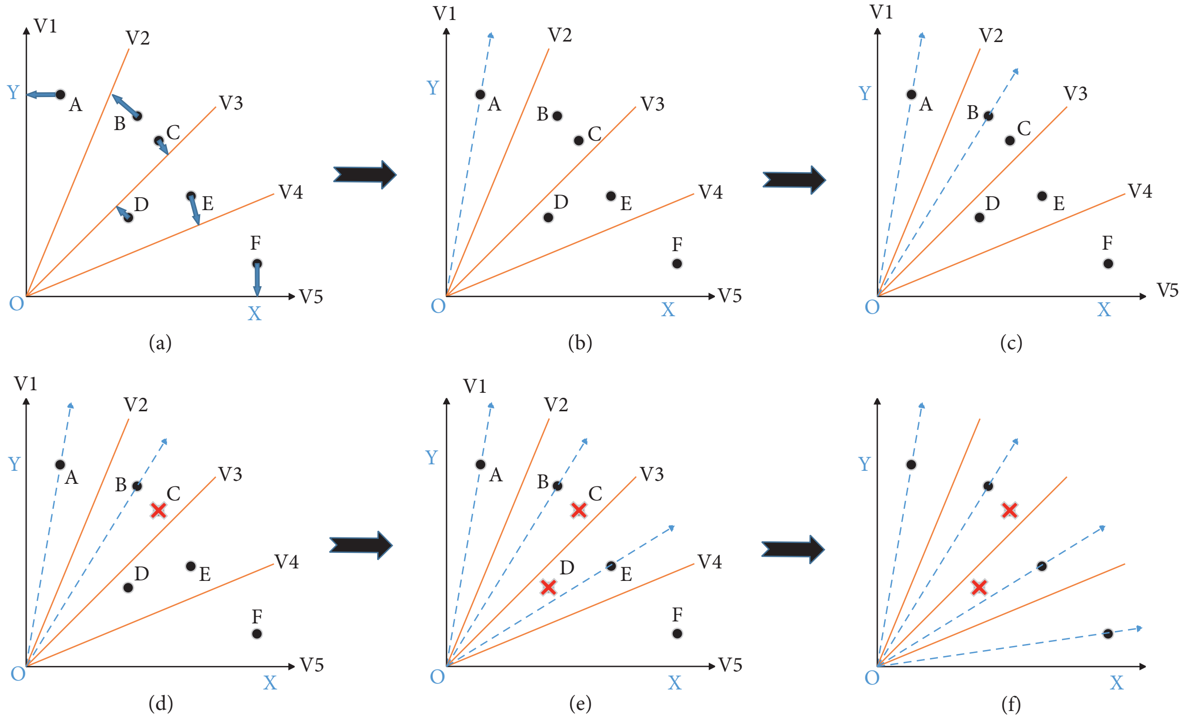
```
Input: $V$; archive; the combined population $Q$;
Output: updated $V$;
(1) for $i = 1 \longrightarrow$ |archive| do
(2)     archive$(i) \leftarrow$ (archive$(i) - Z^{\min}/(Z^{\max} - Z^{\min})$);
(3) for $i = 1 \longrightarrow$ |archive| do
(4)     new $A(i) \leftarrow$ (archive$(i)/\sum_{j=1}^{M}$ archive$(i)_j$);
(5) Associate each solution in $Q$ with an unit reference vector in $V$ according to Definition 2;
(6) for $i = 1 \longrightarrow$ |new $A$| do
(7)     Find the neighborhood reference vectors of new $A_i$ among $V$ and store as NRV$_i$;
(8) For $i = 1 \longrightarrow |V|$ do
(9)     if no solution is associated to $V_i$ then
(10)        $V \leftarrow V/V_i$;
(11) while |new $A$| > 0 and $|V| < N$ do
(12)        AP$\leftarrow 0_{1,|\text{new } A|}$;
(13)        for $i = 1 \longrightarrow$ |new $A$| do
(14)        AP$_i \leftarrow$ Calculate angle position value for new $A_i$;
(15)        $k \leftarrow \arg\min$ AP$_i$, $i \in \{1, 2, \ldots, |\text{new } A|\}$;
(16)        $V \leftarrow V \cup$ new $A_k$
(17)        Remove all the solutions that own the same neighborhood reference vectors with new $A_k$ in new $A$;
```

Algorithm 2: Reference vector adjustment.

that have the same neighborhood reference vectors will be removed from new A. That is to say, through each iteration, a new reference vector will be chosen and some individuals will be removed from new A, and once V has N reference vectors or new A is an empty set, this procedure will stop. It should be noted that removing some individuals from new A is a penalization mechanism. To visually show the detailed mechanism of this function, a simple example is plotted.

As can be seen in Figure 4(a), V_1, V_2, V_3, V_4, and V_5 denote the predefined unit reference vectors, and A, B, C, D, E, and F represent 6 solutions. The arrows in Figure 4(a) point from the solutions to their associated reference vectors, e.g., A is associated with V_1 and C is associated with V_2. At first, the AP values of the six solutions are calculated and the solution with the least AP value will be selected as the first new reference vector. In Figure 4(b), the solution A is chosen as the first new reference vector and there is no solution with the same neighborhood reference vectors as A that need to be removed (line 17 in Algorithm 2). Then, in Figure 4(c), the solution B is selected as the second new reference vector. In Figure 4(d), as the solution B and the solution C own the same neighborhood reference vectors (V_2 and V_3), solution C is removed from the candidate solution set. In Figure 4(e), the solutions E and D are selected or removed in the same way. Figure 4(f) shows the final result of the reference vectors after being modified by the reference vector adjustment strategy. It is obvious that the active subspace is further divided by the new reference vectors and may contribute more for the diversity performance of the population in the following environmental selection method. Besides, for a solution, its position in the objective space instead of corresponding reference vector matters whether it can be constructed as a new reference vector.

4.3. Environmental Selection. The environmental selection strategy of RVEA has two obvious weaknesses. (1) In $d_{t,i,j} = (1 + P(\theta_{t,i,j})) \cdot \|f'_{t,i}\|$, $|f'_{t,i}|$ works to distinguish the



(a) (b) (c) (d) (e) (f)

Figure 4: An example of the reference vector adjustment strategy.

convergence performance of a solution. However, when the number of objectives increases, the length of the objective vector is poor at posing enough selection pressure to the candidate population and selecting solutions with promising convergence performance. (2) After the APD method of each solution is calculated, RVEA selects one solution from each subspace to maintain sound diversity for the new population, while it is hard to guarantee that each reference vector can be associated with one or more solutions, especially with the increase of the number of objectives. To cope with the above two issues, the proposed AP-RVEA offers two methods.

We choose to construct a new parameter to replace $f_{t,i}'$ to evaluate the convergence performance in $d_{t,i,j}$. As can be seen in Algorithm 3, all the solutions in Q are first normalized in lines 1–4. Then, each solution is associated with a subspace in lines 5–7. P_i means the solutions in the ith subspace. Inspired by fractional dominance relation [35], if x_i lies in P_j, the new parameter can be constructed as follows:

$$\begin{aligned} fv(i) &= \sum_{k=1}^{|P_j|} \sum_{q=1}^{M} |\{q\}\, |x_i^q < P_{j,k}^q|, \\ fv'(i) &= \frac{fv(i)}{\sum_{g=1}^{|P_j|} fv\left(P_{j,g}\right)}, \end{aligned} \tag{6}$$

where $P_{j,k}$ means the kth solution in P_j and $P_{j,k}^q$ means the qth objective of $P_{j,k}$. $fv(i)$ denotes the sum of the number of objectives on which x_i surpasses each solution that lies in the same subspace with x_i. $fv'(i)$ is a normalized value of $fv(i)$ and ranges from 0 to 1. For two solutions x_i and x_j in the same subspace, if $fv'(i) > fv'(j)$, x_i surpasses x_j in terms of convergence performance. It is worth noting that the calculation of fv' in different subspaces is independent of each other and thus offers a good way to guarantee that solutions with good convergence performance in each subspace can be found and selected. The improved APD can be calculated as $d_{t,i,j}' = (1 + P(\theta_{t,i,j})) \cdot (1/fv'(t,i))$, where $|f_{t,j}'|$, the length of the normalized solution, is replaced by the aforementioned new parameter. With the improved APD method, the shortage of the original APD method that it is poor at posing enough selection pressure can be filled.

To make the definition of $fv(i)$ more clear, an example is shown to make the readers easily understand $fv(i)$. x_1, x_2, x_3, x_4, and x_5 are 5 objective vectors. x_1, x_2, and x_3 lie in the first subspace, and x_4 and x_5 are in the second subspace. x_1 surpasses x_2, x_3, x_4, and x_5 on 3, 4, 2, and 6 objectives, respectively. Then, the value $fv(x_1)$ is 7, the sum of the objective on which x_1 surpasses x_2 and x_3. The number of the objectives on which x_1 surpasses x_4 and x_5 is not used in the calculation of $fv(x_1)$ because x_4 and x_5 lie in different subspaces with x_1. The same way can be used to calculate $fv(x_2)$, $fv(x_3)$, $fv(x_4)$, and $fv(x_5)$.

After the improved APD methods of population Q are calculated in line 8 in Algorithm 3, the method tailored for addressing the second issue mentioned above is described in lines 9–13. The loop will continue till the population size increases to N: in line 12, one solution will with the least improved APD method will be selected into P in each subspace and these solutions will be removed from Q in line 13; then, if the size of P still less than N, the remaining solutions in Q will be selected into P in the same way as the selected solutions until $|P| = N$.

```
Input: $V$; population size $N$; the combined population $Q$;
Output: new population $P$;
$P \leftarrow \varnothing$
Calculate the ideal point $Z^{\min}$ for $Q$
for $i = 1 \longrightarrow |Q|$ do
    $f_i \leftarrow f_i - Z^{\min}$
for $i = 1 \longrightarrow |Q|$ do
    $k \leftarrow \mathrm{argmin}_{j \in \{1,2,\ldots,|V|\}} \langle Q_i, V_j \rangle$
    $P_k \leftarrow P_k \cup Q_i$
Calculate the improved APD method for each solution in $P_k, k = \{1, 2, \ldots, |V|\}$
while $|P| < N$ do
    for $i = 1 \longrightarrow |V|$ do
        if $|P_i| > 0$ then
            Selected the solution with the minimum improved APD method into $P$;
            Remove the solution from $P_i$;
```

Algorithm 3: Environmental selection.

4.4. Archive Update. The pseudocode of function ArchiveUpdate is shown in Algorithm 4. The inputs of this function are the current archive, population size N, and the new parent population P. In line 1, the dominated solutions in P will be filtered out and the remaining solutions will be stored in P_{ND}. If $|\mathrm{archive}| + |P_{\mathrm{ND}}|$ is larger than N, some solutions in the current archive will be removed until the size of combination of archive and P_{ND} is equal to the population size, the predefined size of archive. If the size of the combination of P_{ND} and the current archive is less than N, all solutions in P_{ND} will be absorbed into archive. The updated criterion is that nondominated solutions in current generation are superior to the nondominated solutions in the last generation, so the excess nondominated solutions in last generation will be randomly removed. In each generation, archive will be updated.

5. Experimental Studies and Discussion

To test the performance of the proposed AP-RVEA, we compare it with 5 state-of-the-art algorithms: A-NSGA-III [32], MOEA/D [5], RVEA* [16], MaOEA-IT [36], and MOEA/D-PaS [37]. The proposed AP-RVEA and the comparative algorithms are all encoded in MATLAB and embedded in PlatEMO [38], which is free to public. All the experiments are run on MATLAB R2018a.

5.1. Experimental Settings

5.1.1. Benchmark Problems. The performance of the six algorithms is compared in the context of MaF1 to MaF9 taken from MaF test suite [18], with 3, 5, 8, and 10 objectives. These 9 benchmarks contain various properties, e.g., disconnected, multimodal, irregular PF, deceptiveness, etc. In this section, a benchmark with a specific number of objective is referred to as a test instance.

5.1.2. Performance Indicator. The hypervolume [39] (HV) and inverted generational distance [40] are chosen to evaluate the final population output by the six algorithms.

(1) The HV is the volume of the space consisting of a reference point and a set of solutions and is widely used to reflect the performance of a population in terms of both convergence and diversity. A population with a larger HV value tends to own better overall performance. In this paper, the reference point used to calculate the HV value of a population on each test instance is embedded in PlatEMO. Besides, in this paper, all the HV values are normalized to $[0, 1]$.

(2) IGD evaluates the overall performance of a population by calculating the distance between the population and the true Pareto front. During the process, the true Pareto front is represented by a set of solutions lying in it and the number in PlatEMO is set to 10,000. The IGD can be calculated as follows:

$$\mathrm{IGD}\left(P^*, P\right) = \frac{\sum_{v \in P*} D_{\min}(v, P)}{\left|P^*\right|}, \tag{7}$$

where P^* is a set of reference points and P denotes the objective vectors of a population. $D_{\min}(v, P)$ represents the minimum Euclidean distance from the point v to all the points in P.

5.1.3. Termination Condition. The maximal number of function evaluations (MFEs) is adopted as the termination condition for the six algorithms. For MaF1–MaF9 with 3, 5, 8, and 10 objectives, the MFEs is set to 100,000.

5.1.4. Population Size. For MOEA/D [5] and other decomposition-based algorithms, the population size is largely determined by the total number of reference points in an M-objective problem. For problems with $M > 8$, a two-layer vector generation strategy can be employed to generate reference (or weight) vectors not only on the outer boundaries but also on the inside layers of the Pareto fronts [7]. Therefore, the population sizes of the six algorithms on

```
Input: archive; population size $N$; selected population $P$;
Output: updated archive;
Filter out the dominated solutions in $P$ and store the remaining solutions into $P_{ND}$
if $|archive| + |P_{ND}| > N$ then
    $k \leftarrow |archive| + |P_{ND}| - N$;
    Randomly delete $k$ solutions from archive; $archive \leftarrow archive \cup P_{ND}$
else
    $archive \leftarrow archive \cup P_{ND}$
```

Algorithm 4: Archive update.

MaF1–MaF9 are set according to the number of objectives, that is, 100, 212, 156, and 275 for 3, 5, 8, and 10 objectives, respectively.

5.2. Environmental Results and Analyses. All the test instances are run 30 times, and the Wilcoxon rank-sum test with $\alpha = 0.5$ is applied to test the significant differences between the HV indicators of populations output by the six algorithms. The mean and standard deviation of HV values and IGD values are shown in Tables 1 and 2, respectively. The symbols −, +, and ≈ in the two tables mean that the corresponding HV values or IGD values are worse than, better than, and equal to that of AP-RVEA. In the bottom of Tables 1 and 2, the number of test instances on which each comparative algorithm behaves worse than, better than, and equal to AP-RVEA is counted.

As can be seen in Table 1, the proposed AP-RVEA surpasses all the comparative algorithms on 17 out of 36 test instances, while the numbers of that of A-NSGA-III, MOEA/D, RVEA*, MaOEA-IT, and MOEA/D-PaS are 5, 3, 7, 0, and 4. Besides, AP-RVEA outperforms the five comparative algorithms on 23, 23, 20, 36, and 23 test instances in terms of HV metric, respectively. In summary, the proposed AP-RVEA shows the best overall performance among the six algorithms. With respect to the results measured by IGD metric shown in Table 2, although the specific statistic are different from that of Table 1, it is also obvious that the proposed AP-RVEA shows promising overall performance. In more detail, AP-RVEA performs better than A-NSGA-III, MOEA/D, RVEA*, MaOEA-IT, and MOEA/D-Pas on 23, 22, 21, 32, and 27 test instances in all, respectively.

The reason why there exist some differences between Tables 1 and 2 may lie in the way the two metrics are calculated. HV metric evaluates the overall performance of a population according to the volume of the space decided by a predefined reference point and the nondominated solutions in the population. Consequently, the HV value is affected by the convergence of each solution in the population on each objective. Generally, it is a concrete value rather than a relative value, while IGD metric assesses the overall performance of a population by calculating the distance between the population and the true Pareto front. During the process, a set of uniformly distributed reference points is used to work as an approximation of the true PF during the calculation of IGD results. When the number of objectives is small, the reference points can be seen as a good approximation of the true PF with respect to uniformity and density; however, when the number of objectives increases, the objective space grows rapidly and the number of the reference points (in PlatEMO, the number is 10,000) is not enough to represent the whole PF of the problems. As a result, some errors may appear during the calculation process of IGD values. For example, if a few solutions in a population happen to lie very close to some reference points, then their contribution to the population IGD value will also be greater than some solutions with good convergence but far away from the reference point. After all, the objective space of MaOPs is very large. It is difficult or even impossible to guarantee that 10,000 reference points can be densely and evenly distributed on the entire PF to fairly evaluate the overall performance of a population by calculating the distances between it and the reference points. Furthermore, how many points can be said to be densely distributed on the real PF of a many-objective problem is still a problem.

Considering the above issues, the IGD and HV metrics may show contradictions in some test instances. However, the overall results shown by the two metrics are consistent, that is, the proposed AP-RVEA obtains the most competitive performance.

Such superior performance of AP-RVEA can be attributed to the following facts. Firstly, the reference vectors are adaptively adjusted according to the obtained solutions with promising performance, i.e., the approximated PF of the MaOP, and thus sound diversity of the obtained population can be achieved. Besides, after the reference vectors are adjusted, the convergence performance of the solutions in each subspace can be more accurately evaluated by an improved APD method compared with the original APD method. Generally, the reference vector adjustment strategy and the improved APD method work together to get the whole performance of AP-RVEA.

In order to visually show the comparison among the six algorithms, their output populations with the largest HV values among the 30 runs on 10-objective MaF1 and 8-objective MaF5 are plotted in the parallel coordinates. Figure 5 gives the comparison results of 10-objective MaF1, and Figure 6 shows the results of 8-objective MaF5.

MaF1 is a linear problem where no single optimal solution lies in any subset of objectives [18]. For 10-objective MaF1, the value of each objective ranges in $[0, 1]$. As can be seen in Figure 5(c), we can find that only a few solutions are output by RVEA*. For A-NSGA-III and MOEA/D-Pas, they fail to converge to the true PF on some objectives. In the

Table 1: HV results of the 6 algorithms on benchmarks MaF–MaF9.

	M	A-NSGA-III [32]	MOEA/D [5]	RVEA*[16]	MaOEA-IT [36]	MOEA/D-PaS [37]	AP-RVEA
MaF1	3	2.0353e − 1 (1.24e − 3) −	1.9823e − 1 (1.76e − 6) +	2.1216e − 1 (1.81e − 3) −	8.0588e − 2 (4.47e − 2) −	1.8407e − 1 (1.07e − 3) −	2.1699e − 1 (5.89e − 4)
	5	8.3089e − 3 (1.82e − 4) +	1.1127e − 2 (3.73e − 4) +	4.0294e − 3 (4.33e − 4) −	2.6586e − 3 (2.46e − 3) −	5.4484e − 3 (1.01e − 4) −	6.1155e − 3 (5.10e − 4)
	8	1.1271e − 5 (3.71e − 6) −	3.1161e − 6 (9.84e − 7) −	7.0858e − 7 (4.00e − 7) −	7.5799e − 7 (1.56e − 6) −	2.6431e − 5 (1.73e − 6) −	2.6546e − 6 (1.19e − 6)
	10	7.9566e − 9 (4.94e − 9) −	1.3977e − 8 (1.82e − 9) −	5.7247e − 9 (3.03e − 9) −	2.6723e − 9 (5.03e − 9) −	2.3472e − 8 (3.55e − 8) −	6.4159e − 8 (1.93e − 8)
MaF2	3	2.2882e − 1 (5.62e − 4) −	2.3886e − 1 (7.14e − 4) +	2.4321e − 1 (9.30e − 4) +	1.3382e − 1 (1.83e − 2) −	2.3105e − 1 (1.28e − 3) +	2.2968e − 1 (1.52e − 3)
	5	1.8268e − 1 (1.23e − 3) −	1.8704e − 1 (4.46e − 4) −	1.7520e − 1 (1.92e − 3) −	1.3101e − 1 (3.22e − 2) −	9.2288e − 2 (6.42e − 3) −	1.9060e − 1 (1.74e − 3)
	8	2.0533e − 1 (3.14e − 3) +	2.0719e − 1 (6.02e − 4) +	1.7835e − 1 (5.28e − 3) ≈	1.4836e − 1 (3.23e − 2) −	7.4638e − 2 (1.84e − 2) −	1.7799e − 1 (4.91e − 3)
	10	1.0651e − 1 (9.50e − 3) −	1.7083e − 1 (7.94e − 4) −	1.0762e − 1 (9.34e − 3) −	9.2650e − 2 (1.41e − 2) −	4.3372e − 2 (3.33e − 4) −	1.7975e − 1 (7.45e − 3)
MaF3	3	9.5716e − 1 (6.89e − 4) −	9.5491e − 1 (1.13e − 3) −	9.6108e − 1 (8.13e − 4) +	8.4815e − 2 (2.60e − 1) −	7.1020e − 1 (3.76e − 1) −	9.5871e − 1 (1.76e − 3)
	5	9.4447e − 1 (2.12e − 1) −	9.8725e − 1 (1.44e − 3) −	8.2530e − 1 (3.35e − 1) −	0.0000e + 0 (0.00e + 0) −	2.1056e − 1 (3.80e − 1) −	9.9929e − 1 (3.38e − 4)
	8	7.8110e − 1 (3.94e − 1) −	9.6767e − 1 (2.78e − 3) −	9.9682e − 1 (3.04e − 4) −	0.0000e + 0 (0.00e + 0) −	9.5248e − 3 (2.85e − 2) −	9.9939e − 1 (2.13e − 2)
	10	9.99454e − 1 (1.28e − 2) +	9.4948e − 1 (2.78e − 3) +	9.9851e − 1 (4.61e − 4) +	0.0000e + 0 (0.00e + 0) −	2.8415e − 3 (1.56e − 2) −	9.1280e − 1 (2.72e − 1)
MaF4	3	5.2149e − 1 (6.16e − 3) +	5.0055e − 1 (3.40e − 3) ≈	5.2054e − 1 (2.85e − 3) +	4.6992e − 2 (1.37e − 1) −	4.2732e − 1 (1.80e − 1) ≈	4.5632e − 1 (8.95e − 2)
	5	5.8544e − 2 (5.49e − 3) ≈	7.4840e − 3 (2.94e − 3) −	3.4932e − 2 (1.10e − 2) −	0.0000e + 0 (0.00e + 0) −	4.2368e − 2 (2.20e − 2) −	6.0373e − 2 (1.06e − 2)
	8	3.8552e − 5 (1.55e − 5) +	2.5638e − 6 (3.90e − 7) −	3.9599e − 5 (3.54e − 5) +	0.0000e + 0 (0.00e + 0) −	2.8057e − 3 (1.47e − 4) +	1.7934e − 5 (1.69e − 5)
	10	3.0500e − 8 (2.50e − 8) −	1.4597e − 8 (1.86e − 9) −	2.8142e − 7 (2.42e − 7) +	1.1961e − 7 (6.55e − 7) −	2.0046e − 4 (4.62e − 5) +	1.8383e − 7 (1.28e − 7)
MaF5	3	4.9687e − 1 (9.09e − 2) −	4.7843e − 1 (1.35e − 1) −	4.5833e − 1 (1.72e − 1) −	1.0907e − 1 (3.90e − 2) −	4.5198e − 1 (1.56e − 1) −	5.5958e − 1 (3.09e − 5)
	5	7.3778e − 1 (2.33e − 2) −	4.4689e − 1 (1.05e − 1) −	8.1174e − 1 (4.40e − 4) +	1.2033e − 1 (9.19e − 2) −	7.4462e − 1 (3.40e − 2) −	8.0819e − 1 (2.53e − 2)
	8	8.2333e − 1 (1.43e − 2) −	3.5794e − 1 (8.63e − 2) −	8.9065e − 1 (4.49e − 2) −	2.5990e − 2 (4.18e − 2) −	8.9836e − 1 (6.10e − 3) ≈	8.9915e − 1 (6.73e − 3)
	10	8.7108e − 1 (2.20e − 2) +	2.5125e − 1 (7.94e − 2) −	8.2530e − 1 (9.06e − 2) ≈	6.7036e − 2 (4.50e − 2) −	8.7214e − 1 (1.21e − 1) +	8.0365e − 1 (2.30e − 2)
MaF6	3	1.9257e − 1 (1.19e − 3) +	1.8185e − 1 (5.24e − 6) +	1.8865e − 1 (1.50e − 3) +	4.6163e − 2 (1.07e − 2) −	1.8937e − 1 (8.73e − 4) +	1.7995e − 1 (1.09e − 3)
	5	1.2878e − 1 (4.32e − 4) −	1.1532e − 1 (3.49e − 2) −	1.0692e − 1 (1.46e − 2) −	1.0711e − 1 (4.43e − 2) −	1.0778e − 1 (8.34e − 4) −	1.3141e − 1 (9.79e − 3)
	8	7.5412e − 2 (4.26e − 2) −	9.5638e − 2 (2.45e − 2) −	9.4549e − 2 (3.40e − 3) −	8.3754e − 2 (3.75e − 2) −	6.3906e − 2 (3.70e − 2) −	9.5782e − 2 (1.33e − 3)
	10	8.4470e − 2 (5.98e − 3) −	7.1992e − 2 (4.28e − 2) ≈	9.4688e − 2 (1.14e − 3) ≈	9.0951e − 3 (2.77e − 2) −	8.1006e − 2 (2.76e − 2) ≈	8.5933e − 2 (1.23e − 2)
MaF7	3	2.5256e − 1 (5.32e − 3) −	2.5331e − 1 (1.38e − 2) ≈	2.6855e − 1 (1.22e − 2) +	1.2596e − 1 (3.78e − 2) −	2.1428e − 1 (9.79e − 2) ≈	2.5443e − 1 (5.81e − 3)
	5	2.1329e − 1 (8.04e − 3) −	1.4613e − 1 (1.31e − 2) −	2.2129e − 1 (3.85e − 3) −	4.9265e − 3 (9.10e − 3) −	2.1578e − 1 (6.01e − 2) −	2.4446e − 1 (2.11e − 3)
	8	9.2868e − 2 (1.71e − 2) −	3.9645e − 3 (1.39e − 2) −	1.8451e − 1 (2.99e − 3) +	0.0000e + 0 (0.00e + 0) −	1.8405e − 1 (7.74e − 3) +	1.2609e − 1 (1.95e − 2)
	10	1.3202e − 1 (1.55e − 2) +	5.3985e − 5 (1.47e − 4) −	1.0158e − 1 (4.76e − 2) +	0.0000e + 0 (0.00e + 0) −	1.5212e − 3 (8.33e − 3) −	6.6357e − 2 (1.80e − 2)

Table 1: Continued.

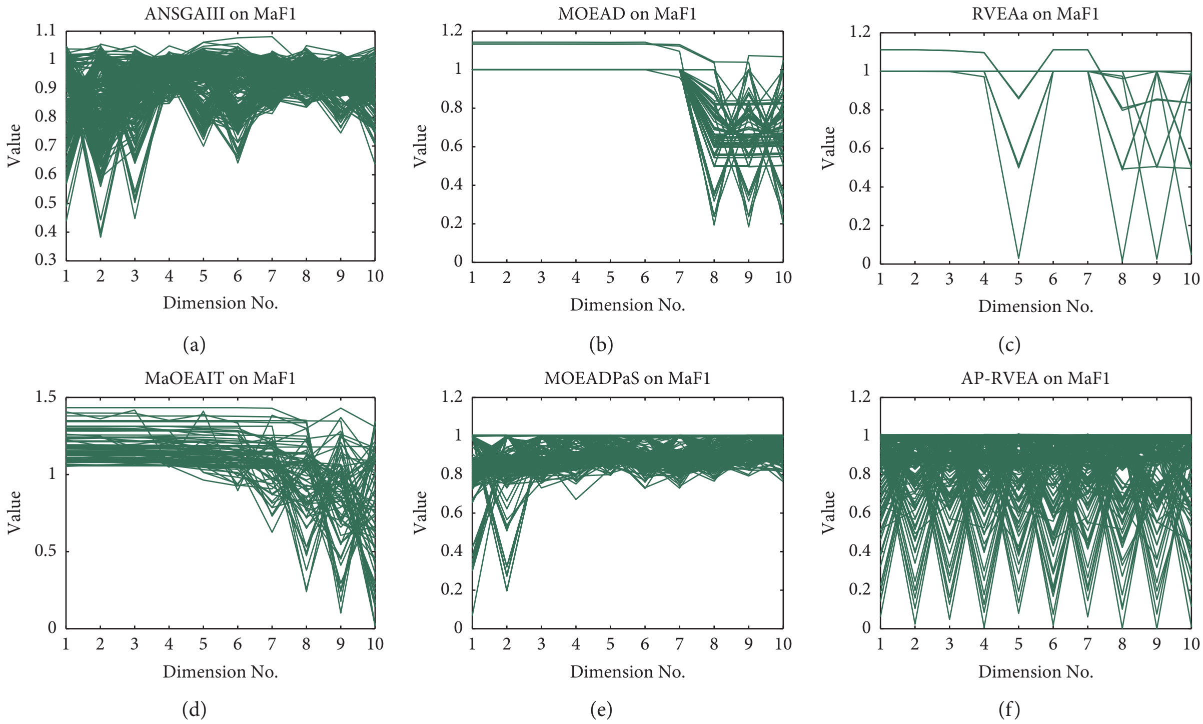
	M	A-NSGA-III [32]	MOEA/D [5]	RVEA*[16]	MaOEA-IT [36]	MOEA/D-PaS [37]	AP-RVEA
MaF8	3	2.5863e−1 (2.33e−3) +	2.5984e−1 (6.33e−3) +	2.6481e−1 (2.65e−3) +	2.4228e−2 (3.38e−2) −	0.0000e+0 (0.00e+0) −	2.4098e−1 (3.60e−3)
	5	1.0867e−1 (1.69e−3) −	1.1426e−1 (2.34e−3) −	8.8484e−2 (5.55e−3) −	6.8305e−3 (1.08e−2) −	0.0000e+0 (0.00e+0) −	1.1673e−2 (5.91e−3)
	8	2.4678e−2 (6.97e−4) +	2.2108e−2 (4.52e−4) +	1.1092e−2 (3.72e−3) −	8.5991e−4 (1.66e−3) −	0.0000e+0 (0.00e+0) −	1.5505e−2 (1.89e−3)
	10	3.8703e−3 (7.50e−4) −	6.2901e−3 (2.08e−4) −	2.6006e−3 (8.94e−4) −	4.9137e−5 (1.52e−4) −	0.0000e+0 (0.00e+0) −	7.9081e−3 (2.71e−4)
MaF9	3	8.3695e−1 (1.26e−3) −	8.3183e−1 (1.52e−2) −	8.4344e−1 (9.81e−4) −	9.9304e−3 (5.44e−2) −	7.7608e−1 (1.48e−1) −	8.5828e−1 (4.12e−2)
	5	2.8542e−1 (3.37e−3) +	3.1093e−1 (2.91e−3) +	2.3391e−1 (1.09e−2) −	0.0000e+0 (0.00e+0) −	2.3636e−1 (5.28e−2) ≈	2.5743e−1 (1.39e−2)
	8	3.3482e−2 (5.60e−3) +	4.1322e−2 (5.03e−4) +	2.2144e−2 (2.52e−3) +	0.0000e+0 (0.00e+0) −	4.4560e−2 (3.99e−4) +	1.9959e−2 (2.68e−3)
	10	6.2283e−3 (1.26e−3) −	1.1816e−2 (2.24e−3) −	4.6773e−3 (1.18e−3) −	0.0000e+0 (0.00e+0) −	1.2047e−2 (2.47e−3) ≈	1.2064e−2 (4.20e−4)
−/+/ ≈		23/12/1	23/10/3	20/13/3	36/0/0	23/7/6	

Table 2: IGD results of the 6 algorithms on benchmarks MaF–MaF9.

	M	A-NSGA-III [32]	MOEA/D [5]	RVEA* [16]	MaOEA-IT [36]	MOEA/D-PaS [37]	AP-RVEA
MaF1	3	6.2920e−2 (1.58e−3) +	7.0475e−2 (7.51e−7) +	5.3794e−2 (2.18e−3) +	2.8372e−1 (1.02e−1) −	9.0179e−2 (1.35e−3) −	8.0285e−2 (1.44e−3)
	5	2.5753e−1 (1.67e−2) −	1.2684e−1 (5.62e−3) +	1.8403e−1 (1.53e−2) −	3.1034e−1 (9.67e−2) −	2.2529e−1 (1.48e−3) −	1.6647e−1 (3.70e−3)
	8	3.5534e−1 (3.65e−2) −	4.6348e−1 (2.69e−2) −	5.9242e−1 (6.12e−2) −	5.5330e−1 (1.21e−1) −	5.0319e−1 (6.46e−2) −	2.7481e−1 (8.18e−3)
	10	6.1235e−1 (7.22e−2) −	5.4023e−1 (1.77e−2) −	6.3868e−1 (7.02e−2) −	6.0878e−1 (1.39e−1) −	3.3410e−1 (1.03e−2) +	4.3303e−1 (2.73e−2)
MaF2	3	3.6496e−2 (3.86e−4) +	3.8937e−2 (5.45e−4) +	2.8213e−2 (6.58e−4) +	1.4382e−1 (3.30e−2) −	5.4623e−2 (1.56e−3) −	4.2871e−2 (1.24e−3)
	5	1.1526e−1 (8.77e−4) −	1.1107e−1 (2.19e−4) −	8.7126e−2 (1.47e−3) −	1.3455e−1 (5.03e−2) ≈	2.2794e−1 (1.03e−2) −	6.1168e−2 (1.19e−3)
	8	1.9101e−1 (4.89e−3) +	2.1724e−1 (3.90e−4) +	1.7075e−1 (4.24e−3) +	3.1336e−1 (5.46e−2) +	2.1058e−1 (2.56e−3) +	7.8364e−1 (3.26e−2)
	10	3.9322e−1 (7.91e−2) −	2.6599e−1 (1.11e−3) −	2.1699e−1 (5.71e−2) −	4.4354e−1 (5.77e−2) −	8.6610e−1 (1.65e−3) −	2.1087e−1 (5.37e−3)
MaF3	3	4.7122e−2 (5.65e−4) −	5.1895e−2 (1.84e−3) −	4.1828e−2 (1.24e−3) ≈	3.8136e+2 (1.48e+3) −	1.5467e+2 (8.31e+2) −	4.2030e−2 (2.13e−3)
	5	8.7541e−1 (2.03e+0) −	1.0687e−1 (4.15e−3) −	6.9057e−1 (1.97e−3) −	1.5492e+4 (7.55e+4) −	2.0454e+3 (5.63e+3) −	1.3820e−1 (2.61e−1)
	8	9.3781e−1 (2.79e+0) −	1.6283e−1 (1.88e−3) −	1.0497e−1 (1.47e−2) −	2.0367e+7 (8.77e+7) −	1.2781e−1 (4.10e−2) −	1.0014e−1 (1.65e+4)
	10	1.0131e−1 (2.33e−2) +	1.3934e−1 (1.33e−3) +	9.6908e−2 (4.41e−3) +	3.0285e+8 (1.10e+9) −	1.5765e+8 (5.03e+8) −	2.2075e−1 (3.61e−1)
MaF4	3	4.0702e−1 (1.50e−2) ≈	6.7751e−1 (2.65e−2) −	3.3615e−1 (1.11e−2) +	5.0300e+1 (1.15e+2) −	9.9331e−1 (1.74e+0) ≈	5.5332e−1 (3.32e−1)
	5	3.4214e+0 (1.07e+0) −	1.0289e+1 (7.60e−1) −	2.8889e+0 (4.54e−1) −	4.3480e+2 (8.85e+2) −	1.4022e+1 (3.01e+1) −	2.4721e+0 (9.11e−2)
	8	3.0978e+1 (3.13e+0) ≈	1.1789e+2 (6.00e+0) −	3.4092e+1 (1.45e+0) −	2.0335e+3 (3.12e+3) −	5.1337e+1 (1.53e+1) −	2.9338e+1 (1.04e+0)
	10	2.5694e+2 (6.10e+1) −	5.3290e+2 (4.20e+1) −	1.0044e+2 (9.25e+0) +	8.8107e+3 (1.43e+4) −	1.2975e+2 (2.65e+1) +	1.9903e+2 (3.41e+1)
MaF5	3	6.4826e−1 (6.61e−1) −	8.5931e−1 (1.24e+0) −	1.1243e+0 (1.62e+0) −	4.6662e+0 (3.11e−1) −	9.8315e−1 (1.17e+0) −	2.5976e−1 (2.71e−5)
	5	1.9720e+0 (4.73e−3) +	8.1263e+0 (1.43e+0) −	1.7716e+0 (1.22e−1) +	1.0212e+1 (2.46e+0) −	2.5855e+0 (2.42e−1) ≈	2.4960e+0 (6.21e−2)
	8	3.0063e+1 (1.81e+0) −	8.3568e+1 (1.46e+0) −	2.2528e+1 (2.63e+0) −	8.9764e+1 (5.22e+1) −	2.6183e+1 (3.39e+0) −	2.1817e+1 (9.14e−1)
	10	1.3002e+2 (3.12e+1) −	3.0356e+2 (1.66e+0) −	1.1822e+2 (2.80e+1) −	2.3986e+2 (4.90e+1) −	1.4810e+2 (2.51e+1) −	1.0322e+2 (1.15e+1)

Table 2: Continued.

	M	A-NSGA-III [32]	MOEA/D [5]	RVEA* [16]	MaOEA-IT [36]	MOEA/D-PaS [37]	AP-RVEA
MaF6	3	1.4787e − 2 (1.49e − 3) +	3.3929e − 2 (4.12e − 6) ≈	2.2360e − 2 (2.67e − 3) +	3.2842e − 1 (6.47e − 2) −	2.3682e − 2 (1.53e − 3) +	3.4808e − 2 (2.06e − 3)
	5	7.5945e − 2 (8.31e − 3) −	5.4923e − 2 (9.94e − 2) −	2.2218e − 1 (2.66e − 1) −	7.6317e − 2 (1.65e − 1) −	9.6494e − 2 (1.53e − 2) −	9.1551e − 3 (3.10e − 3)
	8	1.4803e − 1 (1.71e − 1) +	1.8737e − 1 (2.34e − 1) +	3.7356e − 1 (2.84e − 1) +	7.2548e − 2 (1.60e − 1) +	1.1124e − 1 (2.02e − 2) +	8.3171e − 1 (1.91e − 1)
	10	1.5499e − 1 (4.34e − 2) ≈	2.6580e − 1 (2.54e − 1) ≈	2.3930e − 1 (2.27e − 1) ≈	4.4674e − 1 (6.27e − 2) −	2.4357e + 1 (7.21e + 1) −	1.7272e − 1 (1.26e − 1)
MaF7	3	1.0095e − 1 (4.88e − 2) +	1.9739e − 1 (1.64e − 1) −	1.2263e − 1 (1.04e − 1) −	5.2536e − 1 (1.64e − 1) −	1.1855e + 0 (2.31e + 0) −	1.1628e − 1 (7.04e − 3)
	5	4.9241e − 1 (9.87e − 3) −	5.6292e − 1 (1.78e − 1) −	3.7725e − 1 (1.17e − 2) −	3.5548e + 0 (2.52e + 0) −	9.6303e − 1 (1.04e + 0) −	3.7503e − 1 (1.42e − 2)
	8	1.2508e + 0 (7.13e − 2) +	1.7909e + 0 (3.49e − 1) +	1.6928e + 0 (1.78e − 1) +	2.2385e + 1 (2.24e + 0) −	1.8109e + 0 (1.24e − 1) +	2.6202e + 0 (5.54e − 1)
	10	2.5671e + 0 (3.68e − 1) −	2.7215e + 0 (4.95e − 1) −	1.1068e + 0 (1.01e − 1) +	3.6904e + 1 (2.06e + 0) −	1.9033e + 1 (6.97e + 0) −	1.7752e + 0 (1.58e − 1)
MaF8	3	1.1078e − 1 (4.88e − 3) +	1.1047e − 1 (1.59e − 2) +	9.2711e − 2 (5.08e − 3) +	1.6084e + 2 (2.85e + 2) −	4.0405e + 1 (2.44e + 1) −	1.3668e − 1 (8.50e − 3)
	5	2.6739e − 1 (2.24e − 2) −	2.0045e − 1 (3.08e − 2) −	2.9639e − 1 (7.17e − 2) −	6.5221e + 1 (9.04e + 1) −	2.8168e + 1 (2.21e + 1) −	1.6703e − 1 (1.08e − 2)
	8	4.6062e − 1 (5.26e − 2) +	6.8716e − 1 (1.97e − 2) +	8.8528e − 1 (1.77e − 1) +	1.5075e + 2 (1.87e + 2) ≈	6.8703e − 1 (7.28e − 2) +	4.8632e + 1 (3.46e + 1)
	10	9.4959e − 1 (1.21e − 1) −	9.6884e − 1 (2.09e − 2) −	1.1676e + 0 (2.13e − 1) −	2.1211e + 4 (1.34e + 4) −	3.1499e + 1 (2.91e + 1) −	6.3354e − 1 (5.75e − 2)
MaF9	3	6.2530e − 2 (1.31e − 3) −	6.4715e − 2 (5.85e − 3) −	5.4752e − 2 (6.20e − 4) −	6.0017e + 1 (7.30e + 1) −	9.9649e − 1 (4.90e + 0) −	5.0301e − 2 (7.12e − 2)
	5	2.7053e − 1 (2.81e − 2) −	9.9632e − 2 (3.22e − 3) +	2.0823e − 1 (3.35e − 2) −	8.5660e + 1 (1.37e + 2) −	4.7588e − 1 (1.09e + 0) −	1.6044e − 1 (8.21e − 3)
	8	3.1743e − 1 (1.55e − 1) −	2.7830e − 1 (1.79e − 2) −	6.1440e − 1 (9.20e − 2) −	1.5605e + 2 (1.05e + 2) −	6.3322e − 1 (8.37e − 2) −	1.9853e − 1 (2.97e − 3)
	10	8.3765e − 1 (1.55e − 1) −	6.5459e − 1 (5.98e − 1) −	9.8642e − 1 (1.79e − 1) −	3.9581e + 2 (4.97e + 2) −	8.1119e − 1 (1.95e + 0) −	3.8017e − 1 (2.27e − 2)
−/+/≈		23/10/3	22/12/2	21/13/2	32/2/2	27/7/2	



(a) (b) (c)

(d) (e) (f)

Figure 5: The best populations obtained by the six algorithms on 10-objective MaF1, shown by parallel coordinates.

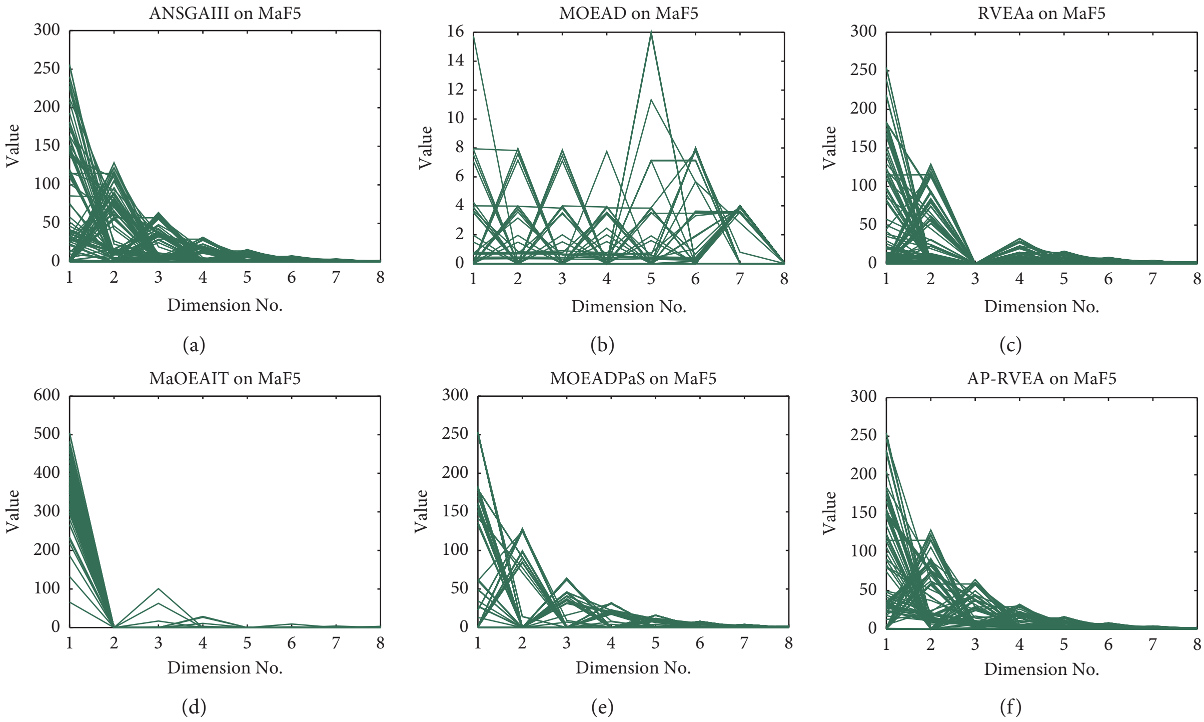


Figure 6: The best populations obtained by the six algorithms on 8-objective MaF5, shown by parallel coordinates.

populations output by MaOEA-IT and MOEA/D, it is obvious that some objectives of some solutions do not fall into $[0, 1]$. The proposed AP-RVEA gains good balance between convergence and diversity according to the figure of its population. In summary, AP-RVEA possesses the best overall performance on 10-objective MaF1.

The output populations obtained by the six algorithms on 8-objective MaF5 are plotted in Figure 6. MaF5 is a concave and biased problem and has a badly scaled PF. Each objective function of MaF5 is scaled to a substantially different range [18]. In Figures 6(a), 6(e), and 6(f), A-NSGA-III, MOEA/D-Pas, and AP-RVEA gain good approximations of the true PF of 8-objective MaF5, while RVEA* fails to converge on the third objective. For MaOEA-IT, we can see that the range of its output population is $[0, 500]$ instead of $[0, 250]$. Besides, the population obtained by MOEA/D seems to have a bad diversity as well as convergence according to the messy line in Figure 6. The results in Table 1 demonstrate that the proposed AP-RVEA shows the most promising overall performance.

6. Conclusion and Future Works

In this paper, a novel adaptive reference vector adjustment strategy is developed to adjust the predefined reference vectors according to the distribution of current promising solutions. Then, an improved APD method is shown to pose more selection pressure in terms of convergence performance to the candidate population. On the basis of the above two methods, an improved version of RVEA called AP-RVEA is proposed to maintain sound population diversity and converge efficiently to the true PF. The proposed AP-RVEA is compared with five state-of-the-art decomposition-based algorithms on 36 test instances taken from MaF test suite. The HV results and IGD results give the conclusion that AP-RVEA achieves the most competitive performance among the six algorithms.

In the future, we hope that AP-RVEA can be used to solve some real-world problems with irregular problems instead of being limited to test problems. Besides, it may also be meaningful to extend AP-RVEA to solve some constraint problems with irregular PFs [41].

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, and C. A. C. Coello, "Survey of multiobjective evolutionary algorithms for data mining: part II," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 20–35, 2014.

[2] C.-H. Chen and J.-H. Chou, "Multiobjective optimization of airline crew roster recovery problems under disruption conditions," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 1, pp. 133–144, 2017.

[3] A. S. Sayyad, T. Menzies, and H. Ammar, "On the value of user preferences in search-based software engineering: a case study in software product lines," in *Proceedings of the 2013*

35th International Conference on Software Engineering (ICSE), pp. 492–501, San Francisco, CA, USA, May 2013.

[4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[5] Q. Hui Li and H. Li, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[6] W. Qiu, J. Zhu, G. Wu, H. Chen, W. Pedrycz, and P. N. Suganthan, "Ensemble many-objective optimization algorithm based on voting mechanism," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 99, pp. 1–15, 2020.

[7] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, Part I: solving problems with Box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[8] X. Cai, H. Sun, Q. Zhang, and Y. Huang, "A grid weighted sum pareto local search for combinatorial multi and many-objective optimization," *IEEE Transactions on Cybernetics*, vol. 49, no. 9, pp. 3586–3598, 2019.

[9] Z. Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao, "Adaptive replacement strategies for moea/d," *IEEE Transactions on Cybernetics*, vol. 46, no. 2, pp. 474–486, 2016.

[10] X. Cai, Z. Mei, Z. Fan, and Q. Zhang, "A constrained decomposition approach with grids for evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 4, pp. 564–577, 2018.

[11] X. Cai, C. Xia, Q. Zhang et al., "The collaborative local search based on dynamic-constrained decomposition with grids for combinatorial multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 99, pp. 1–12, 2019.

[12] X. Xinye Cai, Y. Yexing Li, Z. Zhun Fan, and Q. Qingfu Zhang, "An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 508–523, 2015.

[13] S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 3, pp. 442–446, 2012.

[14] J. Bader and E. Zitzler, "HypE: an algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.

[15] Y. Tian, R. Cheng, X. Zhang, F. Cheng, and Y. Jin, "An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 4, pp. 609–622, 2018.

[16] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.

[17] H.-L. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 450–455, 2014.

[18] R. Cheng, M. Li, Y. Tian et al., "A benchmark test suite for evolutionary many-objective optimization," *Complex and Intelligent Systems*, vol. 3, no. 1, pp. 67–81, 2017.

[19] H. Chen, G. Wu, W. Pedrycz, P. Suganthan, L. Xing, and X. Zhu, "An adaptive resource allocation strategy for objective space partition-based multiobjective optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 3, pp. 1–16, 2019.

[20] F. Wang, Y. Li, F. Liao, and H. Yan, "An ensemble learning based prediction strategy for dynamic multi-objective optimization," *Applied Soft Computing*, vol. 96, Article ID 106592, 2020.

[21] S. Jiang, J. Zhang, and Y. S. Ong, "Asymmetric pareto-adaptive scheme for multiobjective optimization," in *Advances in Artificial Intelligence*, D. Wang and M. Reynolds, Eds., vol. 7106, Springer, Berlin, Germany, 2011.

[22] H. Li and D. Landa-Silva, "An adaptive evolutionary multi-objective approach based on simulated annealing," *Evolutionary Computation*, vol. 19, no. 4, pp. 561–595, 2011.

[23] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired co-evolutionary algorithms using weight vectors," *European Journal of Operational Research*, vol. 243, no. 2, pp. 423–441, 2015.

[24] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with adaptive weight adjustment," *Evolutionary Computation*, vol. 22, 2013.

[25] S. Jiang, X. He, and Y. Zhou, "Many-objective evolutionary algorithm based on adaptive weighted decomposition," *Applied Soft Computing*, vol. 84, Article ID 105731, 2019.

[26] Z. Liang, W. Hou, X. Huang, and Z. Zhu, "Two new reference vector adaptation strategies for many-objective evolutionary algorithms," *Information Sciences*, vol. 483, pp. 332–349, 2019.

[27] X. Cai, Z. Mei, and Z. Fan, "A decomposition-based many-objective evolutionary algorithm with two types of adjustments for direction vectors," *IEEE Transactions on Cybernetics*, vol. 48, no. 8, pp. 2335–2348, 2018.

[28] M. Asafuddoula, H. K. Singh, and T. Ray, "An enhanced decomposition-based evolutionary algorithm with adaptive reference vectors," *IEEE Transactions on Cybernetics*, vol. 48, no. 8, pp. 2321–2334, 2018.

[29] H. Zhao, C. Zhang, and B. Zhang, "A decomposition-based many-objective ant colony optimization algorithm with adaptive reference points," *Information Sciences*, vol. 540, pp. 435–448, 2020.

[30] C. Zhou, G. Dai, C. Zhang, X. Li, and K. Ma, "Entropy based evolutionary algorithm with adaptive reference points for many-objective optimization problems," *Information Sciences*, vol. 465, pp. 232–247, 2018.

[31] F. Wang, Y. Li, H. Zhang, T. Hu, and X.-L. Shen, "An adaptive weight vector guided evolutionary algorithm for preference-based multi-objective optimization," *Swarm and Evolutionary Computation*, vol. 49, pp. 220–233, 2019.

[32] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part ii: handling constraints and extending to an adaptive approach," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2014.

[33] Q. Liu, Y. Jin, M. Heiderich, and T. Rodemann, "Adaptation of reference vectors for evolutionary many-objective optimization of problems with irregular pareto fronts," in *Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1726–1733, Wellington, New Zealand, June 2019.

[34] Q. Liu, Y. Jin, M. Heiderich, T. Rodemann, and G. Yu, "An adaptive reference vector-guided evolutionary algorithm using growing neural gas for many-objective optimization of irregular problems," *IEEE Transactions on Cybernetics*, pp. 1–14, 2020.

[35] W. Qiu, J. Zhu, G. Wu, M. Fan, and P. N. Suganthan, "Evolutionary many-objective algorithm based on fractional dominance relation and improved objective space decomposition strategy," *Swarm and Evolutionary Computation*, vol. 60, Article ID 100776, 2021.

[36] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "A new two-stage evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 748–761, 2019.

[37] R. Wang, Q. Zhang, and T. Zhang, "Decomposition-based algorithms using pareto adaptive scalarizing methods," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 821–837, 2016.

[38] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: a MATLAB platform for evolutionary multi-objective optimization [educational forum]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.

[39] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[40] P. A. N. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 174–188, 2003.

[41] Z. Fan, W. Li, X. Cai et al., "Push and pull search for solving constrained multi-objective optimization problems," *Swarm and Evolutionary Computation*, vol. 44, pp. 665–679, 2019.