Hindawi
Complexity
Volume 2018, Article ID 1753071, 20 pages
https://doi.org/10.1155/2018/1753071

WILEY | Hindawi

Research Article

A Decomposition-Based Multiobjective Evolutionary Algorithm with Adaptive Weight Adjustment

Cai Dai and Xiujuan Lei

College of Computer Science, Shaanxi Normal University, Xi'an 710119, China

Correspondence should be addressed to Xiujuan Lei; xjlei@snnu.edu.cn

Received 24 April 2018; Revised 25 July 2018; Accepted 1 August 2018; Published 12 September 2018

Academic Editor: Michele Scarpiniti

Recently, decomposition-based multiobjective evolutionary algorithms have good performances in the field of multiobjective optimization problems (MOPs) and have been paid attention by many scholars. Generally, a MOP is decomposed into a number of subproblems through a set of weight vectors with good uniformly and aggregate functions. The main role of weight vectors is to ensure the diversity and convergence of obtained solutions. However, these algorithms with uniformity of weight vectors cannot obtain a set of solutions with good diversity on some MOPs with complex Pareto optimal fronts (PFs) (i.e., PFs with a sharp peak or low tail or discontinuous PFs). To deal with this problem, an improved decomposition-based multiobjective evolutionary algorithm with adaptive weight adjustment (IMOEA/DA) is proposed. Firstly, a new method based on uniform design and crowding distance is used to generate a set of weight vectors with good uniformly. Secondly, according to the distances of obtained nondominated solutions, an adaptive weight vector adjustment strategy is proposed to redistribute the weight vectors of subobjective spaces. Thirdly, a selection strategy is used to help each subobjective space to obtain a nondominated solution (if have). Comparing with six efficient state-of-the-art algorithms, for example, NSGAII, MOEA/D, MOEA/D-AWA, EMOSA, RVEA, and KnEA on some benchmark functions, the proposed algorithm is able to find a set of solutions with better diversity and convergence.

1. Introduction

In real-world applications, there are many problems needed to simultaneously optimize multiple objectives which are typically characterized by conflicting objectives. These problems are called as multiobjective optimization problems (MOPs). A continuous optimization problem can be formulated as follows [1]:

$$\begin{aligned} \min \quad & F(x) = (f_1(x), f_2(x), \dots, f_m(x)) \\ \text{s.t.} \quad & g_i(x) \le 0, \quad i = 1, 2, \dots, q, \\ & h_j(x) = 0, \quad j = 1, 2, \dots, p, \end{aligned} \tag{1}$$

where $x = (x_1, \dots \dots, x_n) \in X \subset R^n$ is a n-dimensional decision variable bounded in the decision space X, and m is the number of objective functions. $f_i(x) (i = 1, \dots, m)$ is the ith objective function to be minimized, $g_i(x) (i = 1, 2, \dots, q)$ defines the ith inequality constraint, and $h_j(x) (j = 1, 2, \dots, p)$ defines the jth equality constraint. Moreover, all the inequality and equality constraints determine a set of feasible solutions which is denoted by Ω, and $Y = \{F(x) \mid x \in \Omega\} \subset R^m$ is denoted as the objective space. Because the objectives often contradict each other, the improvement of one objective may cause to the deterioration of other objectives. So, MOPs have many optimal solutions which can be called nondominated solutions [2]. Some important definitions are introduced as follows. Let $x, z \in \Omega$, x is said to be better than z, if $F(x) \ne F(z)$ and $f_i(x) \le f_i(z)$ for $i = 1, 2 \dots, m$. If there is no other x such that x is better than x^*, x^* is called Pareto optimal solution. The set of all the Pareto optimal solutions is defined as the Pareto set (PS). The image of the PS $(\mathrm{PF} = \{F(x) \mid x \in \mathrm{PS}\})$ is called the Pareto optimal front (PF) [2].

Because the number of PF may be infinite, it is impractical to obtain all the Pareto optimal solutions. Thus, the principal goal of solving MOPs is to find out a set of solutions with good diversity and convergence. Currently, multiobjective evolutionary algorithms (MOEAs) use the strategy of the population evolution to simultaneously optimize the solutions of the population in a run. MOEAs can well deal with some complex problems which are characterized with discontinuity, multimodality, and nonlinearity [3]. Nowadays, many MOEAs [4–30] with good performance have been proposed, such as multiobjective genetic algorithms [3], multiobjective particle swarm optimization algorithms [7–10], multiobjective differential evolution algorithms [10, 11], multiobjective immune clone algorithms [12], group search optimizer [13], evolutionary algorithms based on decomposition [14–17], and hybrid algorithms [8, 22]. Moreover, many MOEAs are used to solve numerous applications [31–34].

Recently, Zhang and Li [16, 21] introduce the decomposition approaches into MOEA and developed an outstanding MOEA, MOEA/D, which has a superior performance for many problems. MOEA/D decomposes the MOP into a number of subproblems and uses the EA to optimize these subproblems simultaneously. The two main advantages of MOEA/D are that it uses the neighbor strategy to improve the search efficiency and well maintain the diversity of obtained solutions by the given weight vectors. In the last decade, MOEA/D has attracted many research interests and many related articles [17–22] have been published.

In this work, we mainly study the refinement of weight vectors in MOEA/D to enhance the diversity of obtained solutions. Zhang and Li [16] claim that the weight vectors should be selected properly to obtain the nondominated solutions evenly distributed over the true PF. The basic assumption of MOEA/D is that the set of weight vectors with good uniformity can help obtained nondominated solutions to maintain the diversity. However, recent studies have suggested that MOEA/D which uses the fixed weight vectors might not well solve MOPs with complex PFs [35].

In this paper, we develop an improved decomposition-based multiobjective evolutionary algorithm with adaptive weight vector adjustment (IMOEA/DA) to solve MOPs. The main contributions of this paper are as follows: firstly, a new method [36] based on uniform design and crowding distance [5] is used to generate a uniformity of weight vectors; secondly, some weight vectors are adaptively deleted or added according to the distances of obtained nondominated solutions to solve the problems with complex PF; thirdly, a selection strategy is used to help each subobjective space to obtain a nondominated solution (if have). The frame of decomposition-based multiobjective evolutionary algorithm with adaptive weight vector adjustment and the initialization method of weight vectors has been studied. Moreover, the research result has been presented in the conference "2017 13th International Conference on Computational Intelligence and Security (CIS)". In this conference paper, the adaptive weight adjustment [37] is used. In this new paper, a new adaptive weight adjustment is proposed.

The rest of this paper is organized as follows: Section 2 summarizes the related works of refinements of the weight vectors. Section 3 presents the proposed algorithm IMOEA/DA in detail, while the experiment results of the proposed algorithm and the related analysis are given in Section 5; finally, Section 5 provides the conclusions and proposes the future work.

2. Related Works

MOEA/D uses the predetermined uniformly distributed weight vectors. Recent studies have shown that the fixed weight vector used in MOEA/D might not be able to cover the whole PF very well [35]. Therefore, some researches have refined the weight vectors in MOEA/D. Gu and Liu [38] periodically create the new weight vectors according to the distribution of the current set of weight vectors. Li and Landa-Silva [35] suggest that according to the strategy, the solution of each subproblem should be a long way from the corresponding nearest neighbor to adjust each weight vector. Qi et al. [37] propose an adaptive weight adjustment which utilizes the obtained nondominated solutions to reinstall the weight vectors. In the adaptive weight adjustment, the intersection angle of the target vector of each nondominated solution and the corresponding weight vector of this nondominated solution is zero. Jiang et al. [39] develop an adaptive weight adjustment by sampling the regression curve of objective vectors of the solution in an external population.

Other MOEAs use reference points to solve MOPs. These algorithms guide solutions to converge to the reference points. The principle of algorithms based on reference points or weight vectors is the same. Jain and Deb [40] adjust the reference points in terms of the distribution of candidate solutions in the current population at each generation. Jain and Deb [40] delete reference points with an empty niche and randomly add new reference points inside each crowded reference point with a high niche count. Cheng et al. [41] design two sets of reference vectors, where one maintains uniformly distributed and the other one is adaptively adjusted. Asafuddoula et al. [42] also adopt two sets of reference vectors, where one is called active set which is adaptively adjusted and the other one is called inactive set which stores the discarded reference vectors. In this algorithm [42], the two sets of reference vectors are tuned dynamically over the course of evolution.

3. The Proposed Algorithm

In this paper, an improvement decomposition-based multiobjective evolutionary algorithm with adaptive weight vector adjustment (IMOEA/DA) is proposed to address the MOPs with complex PF. The proposed algorithm mainly consists of two parts: a new weight vector initialization method based on uniform design and crowding distance and adaptive weight vector adjustment strategy, which will be introduced in this section.

3.1. Motivation. The main goal of this paper is to use decomposition-based multiobjective evolutionary algorithm to obtain a set of nondominated solutions which evenly distribute on the true PF and have a good convergence. In this paper, we adaptively add or delete some weight vectors to achieve this goal. In decomposition-based multiobjective evolutionary algorithms, the main role of weight vectors is to improve the convergence of obtained solutions by guiding the search of subproblems. Thus, weight vectors should maintain relative stability to improve the convergence of obtained solutions. We adaptively delete or add some weight vectors by the distances of obtained nondominated solutions to maintain relative stability of weight vectors and solve the problems with complex PF. In addition, we consider that the current optimal solutions of some subproblems are dominated solution, but their optimal solutions are nondominated solution. Some of corresponding weight vectors of these subproblems are retained, and a selection strategy is used to make each subproblem obtain a nondominated solution (if have).

3.2. A New Weight Vector Initialization Method-Based Uniform Design. In this subsection, the new weight vector initialization-based [36] uniform design and crowding distance is presented. Firstly, a uniform design method is briefly shown. For a given bounded and closed set $\mathbf{G} \subset R^M$ (where $\mathbf{M}$ is the dimension of the set $\mathbf{G}$), the uniform design was developed to sample some points which have a small number and are uniformly scattered on $\mathbf{G}$. In this paper, we only consider a specific case of $\mathbf{G}$ and introduce the main features of uniform design. More details can be obtained by referring the literature [34].

For a given set $\mathbf{C} = \{(\theta_1, \theta_2, \ldots, \theta_{\mathbf{M}}) \mid 0 \le \theta_i \le 1, i = 1, \ldots, \mathbf{M}\}$, in general, a set of exactly uniformly scattered points on $\mathbf{C}$ is very difficult to be found. However, there are some efficient methods that can find a set of well approximately uniformly scattered points on $\mathbf{C}$. The good lattice point method (GLP) [43] is one of the simple and efficient methods and can generate a set of uniformly scattered points on $\mathbf{C}$. The details of GLP are as follows. For given integers q, $\mathbf{M}$, and μ, a $q \times \mathbf{M}$ integer matrix $\mathbf{G}(q, \mathbf{M})$ called uniform array is denoted by

$$\mathbf{G}(q,\mathbf{M}) = \left[\mathbf{G}_{ij}\right]_{q\times\mathbf{M}},$$
$$\text{where } \mathbf{G}_{ij} = \left(\bmod\left(i\mu^{j-1}, q\right)\right) + 1, i = 1\sim q, j = 1\sim\mathbf{M}, \tag{2}$$

where $2 \le \mu \le q$, and $\bmod(i\mu^{j-1}, q)$ is the remainder of $i\mu^{j-1}/q$. Thus, there are $q-1$ different integer matrices be generated by these all μ. So, for given q and $\mathbf{M}$, they can determine a number δ (Table 1 lists the vales of δ for different values of q and $\mathbf{M}$) which determines an integer matrix with the smallest discrepancy among these $q-1$ different integer matrices. In this paper, the discrepancy is denoted as $\sup_{r\in G} |q(r)/q - r_1, r_2, \ldots, r_{\mathbf{M}}|$, where $q(r)/q$ is the fraction of the points falling in the hyperrectangle $\mathbf{G}(r) = \{\theta_1\theta_2 \ldots \theta_{\mathbf{M}} \mid 0 \le \theta_i \le r_i, i = 1, \ldots, \mathbf{M}\}$. In practice, the greatest common divisor of μ and q should be 1 to reduce the amount of calculation, which is because that the integer matrix with the smallest discrepancy must be determined by these μ [43].

Table 1: The corresponding values of parameter δ for different values of q and M.

q	M	δ
5	2–4	2
7	2–6	3
11	2–10	7
13	2	5
	3	4
	4–12	6
17	2–16	10
19	2-3	8
	4–18	14
23	2, 13-14, 20–22	7
	8–12	15
	3–7, 15–19	17
29	2	12
	3	9
	4–7	16
	8–12, 16–24	8
	13–15	14
	25–28	18
31	2, 5–12, 20–30	12
	3-4, 13–19	22

Each row of matrix $\mathbf{G}(q, \mathbf{M})$ determiners a point $\mathbf{C}_i = (c_{i,1}, c_{i,2}, \ldots, c_{i,\mathbf{M}})$ of $\mathbf{C}(q, \mathbf{M})$ by

$$c_{ij,} = \frac{2G_{i,j} - 1}{2q}, \quad i = 1\sim q, j = 1\sim\mathbf{M}. \tag{3}$$

$\mathbf{C}(q, \mathbf{M})$ is given by $\mathbf{C}(q, \mathbf{M}) = \{C_i \mid i = 1\sim q\}$. Then each row of matrix $\mathbf{C}(q, \mathbf{M})$ defines a point $\mathbf{D}_i = (d_{i,1}, d_{i,2}, \ldots \ldots, d_{i,\mathbf{M}+1})$ of $\mathbf{D}(q, \mathbf{M}+1)$ by

$$d_{i,j} = \begin{cases} \sin(0.5c_{i,s}\pi), & \text{if } j = \mathbf{M}+1, \\ \prod_{s=1}^{\mathbf{M}} \cos(0.5c_{i,s}\pi), & \text{if } j = 1, \\ \sin\left(0.5c_{i,\mathbf{M}-j+2}\pi\right) \prod_{s=1}^{\mathbf{M}-j+1} \cos(0.5c_{i,s}\pi), & \text{if } 2 \le j \le \mathbf{M}. \end{cases} \tag{4}$$

$\mathbf{D}(q, \mathbf{M}+1) = \{\mathbf{D}_i = (d_{i,1}, d_{i,2}, \ldots, d_{i,\mathbf{M}+1}), i = 1\sim q\}$ can be considered as a set of q uniformly distributed weight vectors [44]. According to (4), we can obtain that many values of the first dimension of $\mathbf{D}(q, \mathbf{M}+1)$ are closed to zeros (see an illustration in Figure 1(a)), which can reduce the diversity of $\mathbf{D}(q, \mathbf{M}+1)$. We use the following method to address this problem. We use $\mathbf{D}(q, \mathbf{M}+1)$ to define a set $\mathbf{W}(q * (\mathbf{M}+1), \mathbf{M}+1) = \{\mathbf{W}_i = (\mathbf{W}_{i,1}, \mathbf{W}_{i,2}, \ldots, \mathbf{W}_{i,\mathbf{M}+1}), i = 1\sim q * (\mathbf{M}+1)\}$, where $\mathbf{W}_j = \mathbf{D}_j, \mathbf{W}_{j+(b-1)*q,t} = \mathbf{D}_{j,t+b-1}, \mathbf{W}_{j+(b-1)*q,k+\mathbf{M}+2-b} =$

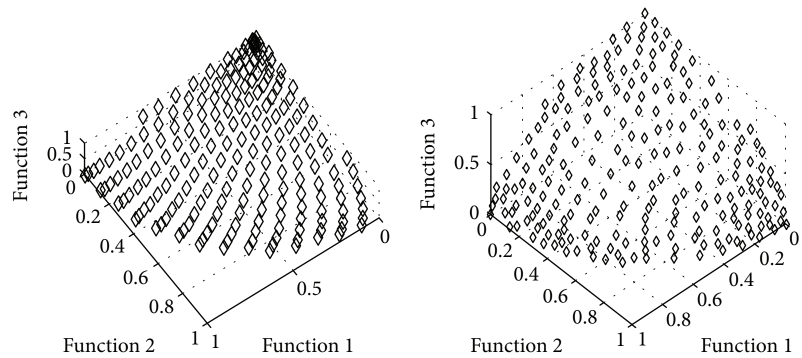


FIGURE 1: Uniformly distributed weight vectors generated by two methods.

$\mathbf{D}_{j,k}$, $j=1\sim q$, $b=2\sim\mathbf{M}+1$, $t=1\sim\mathbf{M}+2-b$, $k=1\sim b-1$. Then we use the crowding distance to select q points from $\mathbf{W}(q*(\mathbf{M}+1),\mathbf{M}+1)$ as the weight vectors (see an illustration in Figure 1(b)).

3.3. Adaptive Weight Vector Adjustment. In this subsection, the adaptive weight vector adjustment is presented. The main idea of this adjustment is that, if the distance of two adjacent nondominated solutions is large, some weight vectors are added between corresponding weight vectors of these two nondominated solutions and, if the distance of two adjacent nondominated solutions is small, one or two weight vectors of these weight vectors should be deleted. This adjustment strategy uses the distances of obtained nondominated solutions to delete or add some weight vectors to solve the problems with complex PF and maintain relative stability of weight vectors. The main difference of this adaptive weight vector adjustment and the method [37] is that the adaptive weight adjustment [37] utilizes the obtained nondominated solutions to reinstall the weight vectors, and the intersection angle of the target vector of each nondominated solution and the corresponding weight vector of this nondominated solution is zero; our adjustment strategy uses the distances of obtained nondominated solutions to delete or add some weight vectors. An illustration of our adjustment strategy is shown in Figure 2.

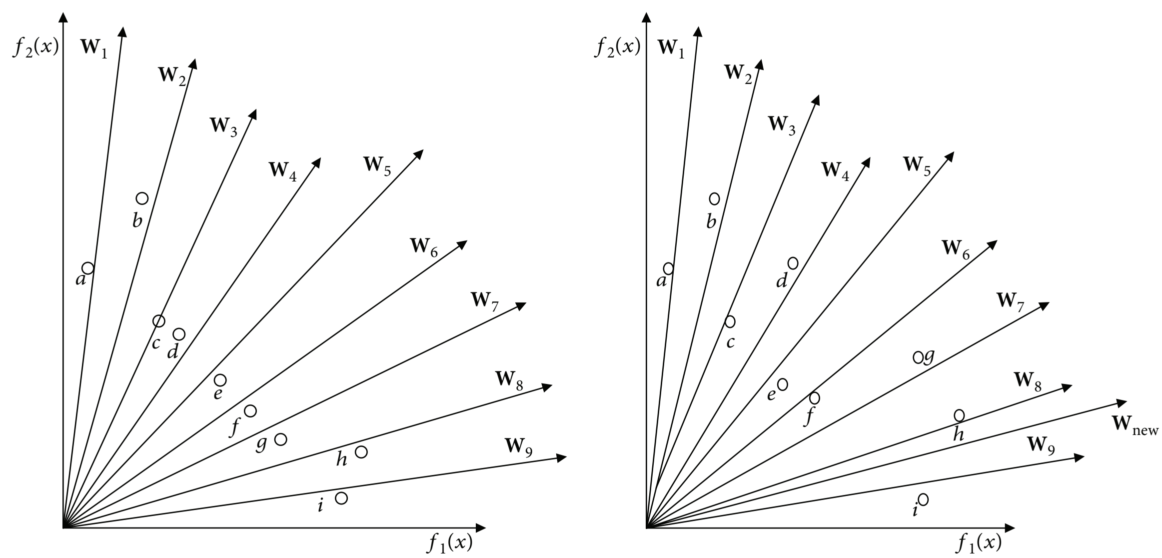
The detail of the adaptive weight vector adjustment strategy is as follows. For the current weight vectors $\mathbf{W}=(\mathbf{W}_1,\mathbf{W}_2,\dots,\mathbf{W}_H)$ and current population $\mathrm{POP}=(x^1,x^2,\dots,x^H)$, where H is the number of solutions or weight vectors and x^i $(i=1\sim H)$ is the current optimal solution of the corresponding subproblem of the weigh vector $\mathbf{W}_i$, we find the nondominated solutions of POP. For convenience, we suggest that $(x^1,x^2,\dots,x^{\mathbf{K}})$ $(\mathbf{K}\le H)$ are the nondominated solutions of POP and denote as $\mathbf{WW}=(\mathbf{W}_{1+\mathbf{K}},\mathbf{W}_{2+\mathbf{K}},\dots,\mathbf{W}_H)$. The distances $\mathbf{ND}_i$ of obtained nondominated solutions of $\mathbf{W}_i$ $(i=1\sim H)$ is calculated as $\mathbf{ND}_i=\max\{|f_j(x^{j_1})-f_j(x^i)|, |f_j(x^i)-f_j(x^{j_2})|, j=1\sim m\}$, where $j_1=\operatorname{argmin}\{s\mid \mathbf{W}_{i,j}>\mathbf{W}_{s,j}, s=1\sim\mathbf{K}\}$ and $j_2=\operatorname{argmax}\{s\mid \mathbf{W}_{i,j}<\mathbf{W}_{s,j}, s=1\sim\mathbf{K}\}$. The values of $\mathbf{ND}_i$ are mainly used to delete the weight vector. In addition, all $|f_j(x^{j_1})-f_j(x^i)|$ and $|f_j(x^i)-f_j(x^{j_2})|$ are sorted to add the weight vectors. For convenience, we use $\mathrm{PD}_{i,u_i}=\max\{|f_j(x^s)-f_j(x^i)|, j=1\sim m, s=1\sim i\}$ to denote the distance of obtained nondominated solutions of $\mathbf{W}_{u_i}$ and $\mathbf{W}_i$, where $u_i=\operatorname{argmax}\{s\mid \max\{|f_j(x^s)-f_j(x^i)|, j=1\sim m\}, s=1\sim\mathbf{K}\}$.

The deleting strategy is as follows. If $\mathbf{K}>\mathbf{N}$ (where $\mathbf{N}$ is the size of the initial population), $\mathbf{N}-\mathbf{K}$ weight vectors with the minimum $\mathbf{ND}_i$ are deleted from $\mathbf{W}$. Then, if $\max\{\mathbf{ND}_i, i=1\sim\mathbf{N}\}/\min\{\mathbf{ND}_i, i=1\sim\mathbf{N}\}>2$, the corresponding weight vector with the minimum $\mathbf{ND}_i$ is deleted from $\mathbf{W}$. After some weight vectors whose corresponding solutions are the dominated solutions are deleted from $\mathbf{W}$, the adding strategy is that, if the size of the current $\mathbf{W}$ is smaller than $\mathbf{N}$, find the $\mathbf{N}-\mathbf{K}$ maximum distances $\mathrm{PD}_{j,u_j}(j=1\sim\mathbf{K})$ of obtained nondominated solutions and $\mathbf{N}-\mathbf{K}$ new weight vectors are generated as follows:

$$\mathbf{W}_{\text{new}}=\begin{cases}\dfrac{0.25^*\mathbf{W}_{u_i}+0.75^*\mathbf{W}_i}{\mathrm{yy}}, & \text{if } \exists\mathbf{W}_k\in\mathbf{WW}, \mathbf{W}_i*\mathrm{tt}'<\mathbf{W}_k tt',\\ \mathrm{tt}, & \text{else},\end{cases} \quad (5)$$

where $\mathrm{yy}=\|0.25*\mathbf{W}_{u_i}+0.75*\mathbf{W}_i\|_2$, $\mathrm{tt}=(0.5*\mathbf{W}_{u_i}+0.5*\mathbf{W}_i)/\|0.5*\mathbf{W}_{u_i}+0.5*\mathbf{W}_i\|_2$, $|\mathbf{W}|$ is the size of $\mathbf{W}$, and the distance PD_{i,u_i} of $\mathbf{W}_{u_i}$ and $\mathbf{W}_{\mathbf{i}}$ is one of the $\mathbf{N}-\mathbf{K}$ maximum distances $\mathrm{PD}_{j,u_j}(j=1\sim\mathbf{K})$ of obtained nondominated solutions. The condition $\exists\mathbf{W}_k\in\mathbf{WW}, \mathbf{W}_i*\mathrm{tt}'<\mathbf{W}_k*\mathrm{tt}'$ makes the optimal solution of the new subproblem generated by the weigh vector $\mathbf{W}_{\text{new}}$ to be nondominated solution. In other word, we do not want that the generated weight vectors locate these spaces which have no nondominated solution. The adaptive weight vector adjustment is summarized in Algorithm 1.

In Step 4, some weight vectors of $\mathbf{WW}$ are kept, which is to record these regions with no nondominated solution and make these subproblems to quickly find nondominated solutions (if have).



(a) Deleting strategy (b) Adding strategy

Figure 2: The adaptive weight vector adjustment. Assume, $\mathbf{N} = 6$, $a, b, \ldots, i$ are the current optimal solutions of the corresponding subproblem of the weigh vectors $\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_9$, respectively. (a) Deleting strategy: b and g are the dominated solutions, and $\mathbf{W}_2$ and $\mathbf{W}_8$ are firstly deleted; $\mathbf{K} = 7 > \mathbf{N}$, d is the most crowded solution, and $\mathbf{W}_4$ is deleted. (b) Adding strategy: b, d, g, and h are dominated solutions, and $\mathbf{W}_2$, $\mathbf{W}_4$, $\mathbf{W}_7$, and $\mathbf{W}_8$ are firstly deleted; $\mathbf{K} = 5 < \mathbf{N}$, i is the thinnest solution, $\mathbf{W}_{new}$ is added.

Require: the size of the initial population N, the current weight vectors $W = (W_1, W_2, \cdots, W_H)$ and current population $POP = (x^1, x^2, \cdots, x^H)$

Ensure: the weight vectors W

Step 1: Find the non-dominated solutions $(x^1, x^2, \cdots, x^K)$ of POP and denote $WW = (W_{1+K}, W_{2+K}, \cdots, W_H)$. Calculate the ND_i and $W_{u_i,j}$.

Step 2: Deleting weight vectors:

If $K > N$, then $N - K$ weight vectors with the minimum ND_i are deleted from W.

While $\max\{ND_i, i = 1 \sim N\}/\min\{ND_i, i = 1 \sim N\} > 2$ **do**

The corresponding weight vector with the minimum ND_i is deleted from W and recalculate the ND_i and $W_{u_i,j}$.

Step 3: Adding weight vectors:

If $N > |W|$ **then**

Find the $N - K$ maximum distances PD_{i,u_i} of obtained non-dominated solutions, and use Eq. (5) to generate the new weight vectors.

Step 4: Deleting some weight vectors of WW from W

If $|WW| > 0.5\,N$ **then**

Use the crowding distance to delete $|WW|$-$0.5\,N$ weight vectors of WW from W.

Algorithm 1: Adaptive weight vector adjustment.

3.4. The Proposed Algorithm IMOEA/DA. IMOEA/DA uses the evolutionary framework of MOEA/D and adopts the strategy [45] of allocating different amount of computational resources to different subproblems. The major roles of this strategy are to distribute computational efforts to each subproblem and optimize these subproblems with nondominated solution to generate nondominated solutions. The major differences between MOEA/D and IMOEA/DA are that the weight vector initialization method is different and IMOEA/DA updates the weight vectors during the course of evolution. The algorithm IMOEA/DA is shown as Pseudocode 1.

In this work, the aggregation function is the variant of Tchebycheff approach which formulation is as follows:

$$\min_{x \in \Omega} \quad g^{TE}(x \mid \mathbf{W}_i, Z^*) = \max_{1 \le j \le m} \left\{ \frac{\left| f_j(x) - z_j^* \right|}{\mathbf{W}_{i,j}} \right\}, \tag{6}$$

where Z^* is the reference point of the MOP. The optimal solution x_i^* of (6) must be the Pareto optimal solution of (1). If the optimal solution x_i^* of (6) is not the Pareto optimal solution of (1), there is a solution y which is better than x_i^*, so

Input:
MOP (1)
A stopping criterion
N: the number of the initial weight vectors (the sub-problems)
T: the number of weight vectors in the neighborhood of each weight vector, $0 < T < N$
$W_1, W_2, \ldots, W_N$: a set of N uniformly distributed weight vectors
Output: Approximation to the PF: $\{F(x^1), F(x^2), \cdots, F(x^N)\}$
Initialization: Generate an initial population $POP = (x^1, x^2, \cdots x^N)$ randomly or by a problem-specific method; determine $Z = (z_1, \cdots, z_m)$ by a problem-specific method; determine $B(i) = \{i_1, \cdots, i_T\}$, $(i = 1, \cdots, N)$, where $W_{i_1}, \ldots W_{i_T}$ are the T closet weight vectors to W_i; set $gen = 0$ and $EP = \{x^1, x^2, \cdots x^N\}$.
While *the stopping criterion is not met* **do**
Step 1: Update the utility function $\varphi_1, \varphi_2, \ldots, \varphi_H$ as follows:

$$\varphi_i = \begin{cases} 0.01 & \text{if } W_i \text{ is } \textit{firstly used} \\ \frac{\textit{old aggregation funtion} - \textit{new aggregation funtion}}{\textit{old aggregation funtion}}, & \textit{else} \end{cases}$$

According to φ_i, use the 10-tournament selection method [46] to select N indices which are denoted as I
Step 2: For $i \in I$, **do**
Select mating scope: uniformly create a random rand (0, 1) from (0, 1) and set

$$P = \begin{cases} B(i) & \text{rand } (0,1) < 1 \\ \{1, 2, \cdots, H\} & \textit{else} \end{cases}$$

Generate offspring $x_{new} = (x_{new,1}, \cdots, x_{new,n})$: set $r_1 = i$ and randomly choose one index r_2 from P, use the SBX operator [47] to generate a solution y from x^{r_1} and x^{r_2} and apply the polynomial mutation operator [47] on y to generate a new offspring x_{new}. Set $EP = EP \cup x_{new}$
Update of Z: For $k = 1, \ldots, m$, if $z_k < f_k(x_{new})$, then set $z_k = f_k(x_{new})$
Update the population by the updated strategy of the literature [44].
end for
Step 3: If *gen* is a multiple of 50, then, use Algorithm 1 to modify the weight vectors W, re-determine $B(i) = \{i_1, \cdots, i_T\}$, $(i = 1, \cdots, H)$ (where H is the size of W), and randomly select solutions from POP to allocate the new sub-problem as their current solution.
Step 4: Set $gen = gen + 1$
end while

Pseudocode 1: The pseudocode of the algorithm IMOEA/DA.

$|f_j(y) - z_j^*| \le |f_j(x_i^*) - z_j^*|, j = 1, \ldots, m$, $\max_{1\le j\le m}\{|f_j(y) - z_j^*|/\mathbf{W}_{i,j}\} \le \max_{1\le j\le m}\{|f_j(x_i^*) - z_j^*|/\mathbf{W}_{i,j}\}$. Thus, x_i^* is not the optimal solution of (6), which is a contradiction.

In Step 2, we use the strategy of the literature [44] which can well maintain the diversity of obtained solutions to update the solutions.

4. Experimental Results and Discussion

In this section, some experiments are conducted to demonstrate the effectiveness of the proposed algorithm IMOEA/DA. Firstly, IMOEA/DA compares with two other algorithms: MOEA/D [16] and NSGAII [5]. Secondly, we compare IMOEA/DA with other MOEA/D with weight vector adjustment strategy: MOEA/D-AWA [37] and EMOSA [35]. Thirdly, IMOEA/DA compares with RVEA [41] and KnEA [48]. Fourthly, we study the effectiveness of IMOEA/DA on MOPs with complex PF. Fifthly, we study the effectiveness of the initialization method of weight vectors and the adaptive weight vector adjustment strategy. Moreover, the effectiveness of IMOEA/DA on many-objective test problems is studied.

4.1. Test Problems. A wide range of well-known and very challenging test problems is selected to test the performance of the proposed algorithm IMOEA/DA in the experiments. These test problems include five biobjective ZDT test problems [2], seven triobjective DTLZ problems [49], ten problems CF of CEC09 competition [50], two problems F1 and F2 [37] with a sharp peak and low tail, nine biobjective WFG test problems [51], and fifteen three-objective MaF problems [52]. To investigate the ability of IMOEA/DA on many-objective problems, two problems DTLZ5(I,m) and DTLZ4(I,m) [53] are selected as the test instances. The IMOEA/DA is implemented by using MATLAB language on a PC with Intel Xeon CPU E3-1226 (3.30 GHz for a single core and the Windows 10 operating system.

4.2. Performance Metrics. In this paper, the true Pareto optimal fronts of the selected test problems are well known. In our experiment, there are four performance metrics are used to quantificationally compare with the performances of algorithms. These three metrics are generational distance (GD) [54], inverted generational distance (IGD) [54], and hypervolume indicator (HV) [55]. Wilcoxon rank-sum test [56] is used in the sense of statistics to compare the mean

IGD, GD, and HV of the compared algorithms. It tests whether the performance of IMOEA/DA on each test problem is better ("+"), same ("="), or worse ("−") than/as that of the compared algorithms at a significance level of 0.05 by a two-tailed test.

4.3. Parameter Setting. All algorithms are coded as real vectors. Distribution index is 20 in the SBX operator; distribution index is 20 and polynomial mutation probability is $1/n$ in the mutation operator; the initial population sizes of all algorithms are set to 105 on all test problems and 105 initial weight vectors are generated; each algorithm is run 30 times with the maximal number of function evaluations 50,000 on all test problems. MOEA/D uses the method of the literature [16] to generate the weight vectors and uses the Tchebycheff approach as the aggregate function. For IMOEA/DA and MOEA/D, the size of neighborhood list T is set to $0.1N$, the probability of choosing mate subproblem from its neighborhood J is set to 0.9.

4.4. Comparisons of IMOEA/DA with Other Multiobjective Algorithms. In this subsection, metric values obtained by IMOEA/DA and other multiobjective algorithms on test problems are shown in Tables 2–6, and some comparisons are made to demonstrate the performance of IMOEA/DA. The best results obtained are highlighted bold in these tables.

4.4.1. Comparisons of IMOEA/DA with NSGAII and MOEA/D. In this subsection, IMOEA/DA is compared with NSGAII and MOEA/D on these twenty-six test problems which include nine WFG problems, ten UF problems, and seven DTLZ problems, and the results obtained three algorithms which are presented in Table 2. Table 2 presents the mean and standard deviation of the HV, GD, and IGD metric values of the final solutions obtained by three algorithms on twenty-six 10-dimensional problems. We can obtain from Table 2 that the results of Wilcoxon rank-sum test show that IMOEA/DA outperforms MOEA/D and NSGAII on all twenty-six test problems and twenty-four test problems in the form of the HV and IGD metrics, respectively; these indicate that the diversity of solutions obtained by IMOEA/DA is better than NSGAII and MOEA/D, and the solutions obtained by IMEOA/DA have a good convergence; in the form of the GD metrics, IMOEA/DA outperforms NSGAII on twenty-three test problems and outperforms MOEA/D on ten test problems, and these imply that the solutions obtained by IMEOA/DA have a good convergence. Moreover, it can be seen from Table 2 that, in the form of the HV and IGD metrics, the results obtained by IMOEA/DA are better than those obtained by NSGAII and MOEA/D on all twenty-six test problems, which indicates that the final solutions obtained by IMOEA/DA have a better diversity than those obtained by NSGAII and MOEA/D and have a good convergence. We also can obtain from Table 2 that the mean values of GD metric contained by IMOEA/DA are bigger than those obtained by MOEA/D on twelve test problems, which include WFG2, WFG8, UF2, UF3, UF6, UF8, UF9, DTLZ2–DTLZ5, and DTLZ7, and are smaller than those obtained by MOEA/D on other fourteen test problem; the mean values of GD metric contained by IMOEA/DA are smaller than those obtained by NSGAII on all these problems except for problems UF6, DTLZ4, and DTLZ7, and these imply that IMOEA/DAA can obtain a set of solutions with better convergence than MOEA/D and NSGAII on most test problems. Moreover, the mean values of IGD obtained by IMOEA/DA are smaller at least 15% than those obtained by MOEA/D and NSGAII on most test problems of these twenty-six test problems. To visually show the performance of the proposed algorithm, according to the median values of IGD metric, the nondominated solutions obtained by IMEOA/DA are plotted in Figures 3 and 4. It is evident that the found approximated PF is distributed uniformly on the true PF on these twenty-six test problems except for problem UF5. These comparisons indicate that IMOEA/DA is better at maintaining the diversity of obtained solutions than MOEA/D and NSGAII, and IMOEA/DA can also obtain a set of solutions with good convergence.

MOEA/D decomposes a MOP into a number of subproblems and optimizes them simultaneously. Each subproblem is optimized by using information from its several neighboring subproblems. In MOEA/D, the current solution of each subproblem is updated by these offspring which are generated by the neighbors of this subproblem, which can improve the convergence, and which cannot be conducted to maintain the diversity. In our algorithm, each offspring is firstly classified, the current solutions of the same class as the offspring may be updated [44], and our adaptive weight vector adjustment strategy can solve MOPs with complex PFs, which help obtained solutions to maintain the diversity. Moreover, the above comparisons indicate that IMOEA/DA is better at maintaining the diversity of obtained solutions than MOEA/D. NSGAII uses the nondominated sorting and the crowding distance to rank the solutions. During selection, NSGAII uses a crowded comparison operator that takes into consideration both the nondomination rank of an individual in the population and its crowding distance (i.e., nondominated solutions are preferred over dominated solutions, but between the two solutions with the same nondomination rank, the one that resides in the less crowded region is preferred). Our algorithm uses the neighboring solutions to generate the offspring, which can generate better offspring than NSGAII. And the above comparisons declare that IMOEA/DA is better at maintaining the diversity of obtained solutions and improving the convergence than NSGAII.

4.4.2. Comparisons of IMOEA/DA with MOEA/D-AWA and EMOSA. MOEA/D-AWA [37] and EMOSA [35] are an improved MOEA/D with adaptive weight vector adjustment. It has a good performance of solving the MOPs with complex PFs. IMOEA/DA is compared with MOEA/D-AWA and EMOSA on fourteen test problems which include five ZDT problems, five DTLZ problems, two constructed problems F1-F2, and two many-objective problems DTLZ4(3,6) and DTLZ5(3,6). In this experiment, EMOSA does not test many-objective problems DTLZ4(3,6) and DTLZ5(3,6). The experimental results of MOEA/D-AWA and EMOSA are directly obtained from the original literatures to make a fair comparison, and the population sizes

Table 2: HV, GD, and IGD obtained by IMOEA/DA, MOEA/D, and NSGAII.

Problems	HV		GD		IGD	
	Mean	Std	Mean	Std	Mean	Std
WFG1						
IMOEA/DA	**6.7512**	0.0008	**0.0014**	0.0001	**0.0125**	0.0002
MOEA/D	6.7205(+)	0.0012	0.0015(+)	0.0001	0.0535(+)	0.0029
NSGAII	6.7428(+)	0.0023	0.0015(+)	0.0002	0.0151(+)	0.0007
WFG2						
IMOEA/DA	**6.1411**	0.0002	0.0065	0.0006	**0.0109**	0.0003
MOEA/D	6.0341(+)	0.0971	**0.0053**(−)	0.0006	0.0773(+)	0.0319
NSGAII	6.0381(+)	0.1065	0.0059(−)	0.0004	0.0504(+)	0.0407
WFG3						
IMOEA/DA	**5.6356**	0.0006	**0.0023**	0.0003	**0.0134**	0.0004
MOEA/D	5.6080(+)	0.0036	**0.0023**(=)	0.0005	0.0146(+)	0.0005
NSGAII	5.6245(+)	0.0035	0.0025(+)	0.0005	0.0148(+)	0.0006
WFG4						
IMOEA/DA	**3.3611**	0.0017	**0.0025**	0.0002	**0.0131**	0.0003
MOEA/D	3.3404(+)	0.0014	0.0026(=)	0.0003	0.0137(+)	0.0003
NSGAII	3.3583(+)	0.0015	0.0027(+)	0.0003	0.0154(+)	0.0007
WFG5						
IMOEA/DA	**3.0364**	0.0614	**0.0533**	0.0145	**0.0575**	0.0133
MOEA/D	2.9598(+)	0.0016	0.0605(+)	0.0001	0.0708(+)	0.0007
NSGAII	3.0083(+)	0.0100	0.0636(+)	0.0005	0.0667(+)	0.0005
WFG6						
IMOEA/DA	**3.3640**	0.0010	**0.0025**	0.0002	**0.0133**	0.0006
MOEA/D	3.3383(+)	0.0028	0.0028(+)	0.0002	0.0139(+)	0.0002
NSGAII	3.3530(+)	0.0051	0.0029(+)	0.0006	0.0157(+)	0.0006
WFG7						
IMOEA/DA	**3.3634**	0.0009	**0.0025**	0.0003	0.0139	0.0002
MOEA/D	3.3415(+)	0.0004	**0.0025**(=)	0.0003	**0.0138**(=)	0.0003
NSGAII	3.3562(+)	0.0019	0.0026(=)	0.0001	0.0158(+)	0.0007
WFG8						
IMOEA/DA	**3.3504**	0.0016	0.0048	0.0003	0.0147	0.0003
MOEA/D	3.3351(+)	0.0040	**0.0032**(−)	0.0003	**0.0143**(=)	0.0004
NSGAII	3.3418(+)	0.0018	0.0058(+)	0.0005	0.0155(+)	0.0005
WFG9						
IMOEA/DA	**3.3101**	0.0005	**0.0025**	0.0002	**0.0145**	0.0006
MOEA/D	3.2826(+)	0.0097	0.0026(+)	0.0003	0.0164(+)	0.0006
NSGAII	3.2991(+)	0.0095	0.0028(+)	0.0005	0.0164(+)	0.0012
UF1						
IMOEA/DA	**0.8652**	0.0038	0.0068	0.0004	**0.0070**	0.0027
MOEA/D	0.7868(+)	0.0547	**0.0044**(−)	0.0005	0.0653(+)	0.0553
NSGAII	0.8307(+)	0.0355	0.0120(+)	0.0033	0.0324(+)	0.0276
UF2						
IMOEA/DA	**0.8695**	0.0002	0.0067	0.0023	**0.0050**	0.0002
MOEA/D	0.8631(+)	0.0017	**0.0045**(−)	0.0002	0.0083(+)	0.0013
NSGAII	0.8643(+)	0.0006	0.0080(+)	0.0005	0.0076(+)	0.0004
UF3						
IMOEA/DA	**0.8620**	0.0112	0.0077	0.0009	**0.0106**	0.0105
MOEA/D	0.7372(+)	0.0615	**0.0040**(−)	0.0004	0.1220(+)	0.0636
NSGAII	0.8559(+)	0.0126	0.0111(+)	0.0060	0.0151(+)	0.0163

Table 2: Continued.

Problems	HV Mean	HV Std	GD Mean	GD Std	IGD Mean	IGD Std
UF4						
IMOEA/DA	**0.5386**	0.0001	**0.0036**	0.0001	**0.0035**	0.0001
MOEA/D	0.5373(+)	0.0003	0.0037(=)	0.0001	0.0036(+)	0.0001
NSGAII	0.5359(+)	0.0003	0.0046(+)	0.0002	0.0053(+)	0.0001
UF5						
IMOEA/DA	**0.3538**	0.0939	**0.0427**	0.0487	**0.2350**	0.0567
MOEA/D	0.2148(+)	0.1103	0.1603(+)	0.1113	0.4141(+)	0.1319
NSGAII	0.2441(+)	0.1092	4.1854(+)	3.0226	0.3328(+)	0.1395
UF6						
IMOEA/DA	**0.6357**	0.0114	0.0195	0.0103	**0.0070**	0.0024
MOEA/D	0.5675(+)	0.0224	**0.0109**(−)	0.0031	0.0405(+)	0.0187
NSGAII	0.5984(+)	0.0066	0.0155(+)	0.0073	0.0169(+)	0.0026
UF7						
IMOEA/DA	**0.7026**	0.0003	**0.0043**	0.0002	0.**0046**	0.0002
MOEA/D	0.6946(+)	0.0069	0.0049(+)	0.0004	0.0072(+)	0.0037
NSGAII	0.6949(+)	0.0016	0.0065(+)	0.0008	0.0083(+)	0.0009
UF8						
IMOEA/DA	**0.6861**	0.0194	0.0549	0.0226	**0.0953**	0.0119
MOEA/D	0.6388(+)	0.0153	**0.0273**(−)	0.0296	0.1319(+)	0.0262
NSGAII	0.5568(+)	0.0367	0.6345(+)	0.5476	0.1487(+)	0.0263
UF9						
IMOEA/DA	**0.9420**	0.0827	0.2133	0.1745	**0.0959**	0.0541
MOEA/D	0.8996(+)	0.0765	**0.0915**(−)	0.0311	0.1082(+)	0.0537
NSGAII	0.6614(+)	0.2035	1.7880(+)	1.7009	0.2194(+)	0.0998
UF10						
IMOEA/DA	**0.5102**	0.0691	**0.0549**	0.0891	**0.1513**	0.0424
MOEA/D	0.3718(+)	0.0962	0.0897(+)	0.0402	0.2245(+)	0.0443
NSGAII	0.3967(+)	0.0590	1.3988(+)	0.9461	0.2235(+)	0.0494
DTLZ1						
IMOEA/DA	**0.0962**	0.0007	**0.0072**	0.0001	**0.0199**	0.0009
MOEA/D	0.0934(+)	0.0003	0.0074(+)	0.0001	0.0244(+)	0.0002
NSGAII	0.0947(+)	0.0019	0.0247(+)	0.0415	0.0296(+)	0.0068
DTLZ2						
IMOEA/DA	**0.7138**	0.0051	0.0214	0.0005	**0.0550**	0.0030
MOEA/D	0.6989(+)	0.0023	**0.0193**(−)	0.0002	0.0632(+)	0.0006
NSGAII	0.6973(+)	0.0037	0.0241(+)	0.0010	0.0715(+)	0.0028
DTLZ3						
IMOEA/DA	**0.7196**	0.0051	0.0204	0.0010	**0.0514**	0.0030
MOEA/D	0.6983(+)	0.0032	**0.0197**(−)	0.0007	0.0633(+)	0.0004
NSGAII	0.7104(+)	0.0054	0.0204(+)	0.0013	0.0703(+)	0.0026
DTLZ4						
IMOEA/DA	**0.7234**	0.0062	0.0246	0.0040	**0.0544**	0.0014
MOEA/D	0.6997(+)	0.0038	**0.0196**(−)	0.0006	0.0632(+)	0.0005
NSGAII	0.6128(+)	0.1256	0.0192(+)	0.0055	0.2442(+)	0.2432
DTLZ5						
IMOEA/DA	**0.4394**	0.0003	0.0009	0.0003	**0.0047**	0.0002
MOEA/D	0.4366(+)	0.0001	**0.0004**(−)	0.0002	0.0071(+)	0.0002
NSGAII	0.4376(+)	0.0003	0.0012(+)	0.0002	0.0054(+)	0.0004

Table 2: Continued.

Problems	HV		GD		IGD	
	Mean	Std	Mean	Std	Mean	Std
DTLZ6						
IMOEA/DA	**0.4412**	0.0001	**0.0004**	0.0001	**0.0045**	0.0001
MOEA/D	0.3461(+)	0.0300	0.1026(+)	0.0298	0.0820(+)	0.0289
NSGAII	0.3566(+)	0.0377	0.0763(+)	0.0333	0.0710(+)	0.0332
DTLZ7						
IMOEA/DA	**1.6576**	0.0074	0.0143	0.0028	**0.0596**	0.0046
MOEA/D	1.5107(+)	0.0617	**0.0076**(−)	0.0037	0.1790(+)	0.1156
NSGAII	1.5623(+)	0.0863	0.0129(=)	0.0016	0.1451(+)	0.1245

"+" means that IMOEA/DA outperforms its competitor algorithm, "−" means that IMOEA/DA is worse than its competitor algorithm, and "=" means that the competitor algorithm has the same performance as IMOEA/DA.

Table 3: IGD obtained by IMOEA/DA and MOEA/D-AWA.

Problems	MOEA/D-AWA		EMOSA		IMOEA/D	
	Mean	Std	Mean	Std	Mean	Std
ZDT1	$4.470e-3$	$2.239e-4$	$3.674e-3$	$5.923e-5$	$\mathbf{3.300e-3}$	$1.57e-4$
ZDT2	$4.482e-3$	$1.837e-3$	$3.900e-3$	$2.735e-4$	$\mathbf{3.300e-3}$	$1.84e-4$
ZDT3	$6.703e-3$	$4.538e-4$	$9.737e-3$	$6.636e-4$	$\mathbf{3.702e-3}$	$2.44e-4$
ZDT4	$4.238e-3$	$3.102e-4$	$5.174e-3$	$7.339e-4$	$\mathbf{3.95e-3}$	$1.47e-4$
ZDT6	$4.323e-3$	$2.819e-4$	$3.601e-3$	$4.250e-4$	$\mathbf{1.910e-3}$	$3.08e-4$
DTLZ1	$1.237e-2$	$1.617e-3$	$1.632e-2$	$2.106e-3$	$\mathbf{1.07e-2}$	$1.64e-4$
DTLZ2	$3.065e-2$	$1.183e-4$	$3.232e-2$	$9.275e-4$	$\mathbf{2.96e-2}$	$4.67e-4$
DTLZ3	$3.196e-2$	$8.036e-4$	$5.723e-2$	$9.761e-2$	$\mathbf{2.86e-2}$	$3.92e-4$
DTLZ4	$3.068e-2$	$1.351e-4$	$3.443e-2$	$9.476e-3$	$\mathbf{2.97e-2}$	$3.54e-4$
DTLZ7	$3.610e-2$	$5.054e-3$	$6.980e-2$	$2.539e-3$	$\mathbf{3.00e-2}$	$7.81e-4$
F1	$5.204e-3$	$7.7975e-5$	$6.11e-3$	$3.608e-4$	$\mathbf{4.800e-3}$	$1.45e-4$
F2	$\mathbf{1.637e-2}$	$3.104e-4$	$1.663e-2$	$3.233e-4$	$1.81e-2$	$1.000e-3$
DTLZ4(3,6)	0.0379	0.0005	NA	NA	**0.0271**	0.0017
DTLZ5(3,6)	0.0382	0.0006	NA	NA	**0.0265**	0.0014

of IMOEA/DA are set to 100, 300, and 252 for two-objective problems, three-objective problems, and six-objective problems, respectively; the maximal number of function evaluations is set to 50,000, 75,000, and 200,000 for two-objective problems, three-objective problems, and six-objective problems, respectively, and other parameter settings are the same as Subsection 4.3.

Table 3 shows the mean and standard deviation values of IGD metric obtained by IMOEA/DA, MOEA/D-AWA, and EMOSA on these fourteen test problems. We can obtain from this table that the mean values of IGD metric obtained by IMOEA/DA are smaller than those obtained by MOEA/D-AWA on all fourteen test problems except for problem F2 and are smaller than those obtained by MOEAS on all twelve test problems except for problem F2, which indicate that the quality of the final solutions obtained by IMOEA/DA is better than those obtained by MOEA/D-AWA and EMOSA on thirteen problems and eleven problems, respectively. These comparisons imply that IMOEA/DA is better at solving MOPs with complex PFs than MOEA/D-AWA and EMOSA on most problems of these problems. According to the median values of IGD metric, the nondominated solutions obtained by IMEOA/DA on the five ZDT test problems, F1 and F2, are plotted in Figures 3 and 4, which can visually show the good performance of IMOEA/DA. These indicate that IMOEA/DA can effectively approach the true PFs.

The adaptive weight vector adjustment strategy of MOEA/D-AWA uses the obtained nondominated solutions and crowding distance to adaptively set the weight vectors; however, the crowding distance based on nondominated solutions is not a good method to maintain the uniformity of weight vectors, especially in problems with three or more objective problems. In MOEAS, each weight vector is periodically adjusted to make its solution of subproblem far from the corresponding nearest neighbor. This can well maintain the uniformity of weight vectors for two-objective problems; however, it may lose efficacy for three or more objectives problems. In our algorithm, according to the distance of neighboring nondominated solutions, the weight vectors

Table 4: HV, GD, and IGD obtained by IMOEA/DA, RVEA, and KnEA.

Problems	HV		GD		IGD	
	Mean	Std	Mean	Std	Mean	Std
MaF1						
IMOEA/DA	**2.874** $e-$ **1**	1.94 $e-3$	**4.695** $e-$ **4**	5.36 $e-4$	**4.613** $e-$ **2**	1.12 $e-3$
RVEA	2.451 $e-1$(+)	5.36 $e-3$	3.429 $e-3$(+)	9.00 $e-4$	7.328 $e-2$(+)	2.49 $e-3$
KnEA	2.864 $e-1$(=)	4.01 $e-3$	4.868 $e-4$(=)	2.17 $e-4$	4.617 $e-2$(=)	3.53 $e-3$
MaF2						
IMOEA/DA	**2.223** $e-$ **1**	9.23 $e-4$	1.047 $e-3$	1.37 $e-4$	**3.298** $e-$ **2**	6.49 $e-4$
RVEA	2.058 $e-1$(+)	1.64 $e-3$	5.648 $e-3$(+)	4.28 $e-4$	3.945 $e-2$(+)	1.91 $e-3$
KnEA	2.166 $e-1$(=)	7.83 $e-4$	**7.401** $e-$ **4(−)**	3.68 $e-5$	3.429 $e-2$(=)	2.21 $e-3$
MaF3						
IMOEA/DA	**1.278** $e+$ **0**	2.47 $e-3$	6.967 $e-4$	5.11 $e-4$	4.000 $e-2$	2.80 $e-3$
RVEA	1.277 $e+0$(=)	1.63 $e-3$	5.434 $e-4$(−)	1.84 $e-4$	**3.945** $e-$ **2(=)**	9.79 $e-4$
KnEA	1.170 $e+0$(+)	1.58 $e-1$	**4.292** $e-$ **4(−)**	1.16 $e-4$	1.432 $e-1$(+)	1.11 $e-1$
MaF4						
IMOEA/DA	**4.472** $e+$ **1**	4.06 $e-1$	3.874 $e-2$	8.35 $e-2$	**3.744** $e-$ **1**	7.14 $e-2$
RVEA	4.178 $e+1$(+)	2.63 $e+0$	3.371 $e-3$(−)	3.26 $e-4$	4.337 $e-1$(+)	1.12 $e-1$
KnEA	4.340 $e+1$(+)	1.56 $e+0$	**3.027** $e-$ **3(−)**	1.74 $e-4$	5.478 $e-1$(+)	1.46 $e-1$
MaF5						
IMOEA/DA	**4.808** $e+$ **1**	2.69 $e-1$	**1.899** $e-$ **3**	2.75 $e-3$	**2.308** $e-$ **1**	9.46 $e-3$
RVEA	4.795 $e+1$(=)	2.72 $e-3$	2.318 $e-3$(+)	1.55 $e-5$	2.398 $e-1$(+)	3.79 $e-5$
KnEA	4.632 $e+1$(+)	3.28 $e-1$	2.309 $e-3$(+)	1.87 $e-4$	3.078 $e-1$(+)	2.05 $e-2$
MaF6						
IMOEA/DA	**1.311** $e-$ **1**	4.50 $e-4$	**4.634** $e-$ **5**	2.55 $e-5$	**8.499** $e-$ **3**	1.01 $e-3$
RVEA	1.178 $e-1$(+)	2.73 $e-3$	4.181 $e-2$(+)	6.17 $e-2$	3.530 $e-2$(+)	4.65 $e-3$
KnEA	1.216 $e-1$(+)	7.01 $e-3$	1.799 $e-4$(+)	3.28 $e-4$	2.355 $e-2$(+)	1.35 $e-2$
MaF7						
IMOEA/DA	**1.650** $e+$ **0**	1.39 $e-2$	3.356 $e-3$	2.70 $e-2$	**6.351** $e-$ **2**	4.95 $e-3$
RVEA	1.541 $e+0$(+)	1.30 $e-2$	7.870 $e-3$(+)	4.85 $e-4$	1.047 $e-1$(+)	2.10 $e-3$
KnEA	1.637 $e+0$(=)	1.81 $e-2$	**1.155** $e-$ **3(−)**	2.39 $e-4$	6.755 $e-2$(+)	4.59 $e-3$
MaF8						
IMOEA/DA	**1.783** $e+$ **0**	6.29 $e-2$	**3.871** $e-$ **3**	8.64 $e-4$	**1.024** $e-$ **1**	2.52 $e-2$
RVEA	1.681 $e+0$(+)	2.29 $e-2$	2.528 $e-2$(+)	2.03 $e-3$	1.209 $e-1$(+)	8.03 $e-3$
KnEA	1.282 $e+0$(+)	1.51 $e-1$	4.238 $e-3$(+)	2.85 $e-3$	3.212 $e-1$(+)	6.96 $e-2$
MaF9						
IMOEA/DA	**3.683** $e+$ **0**	3.70 $e-2$	1.468 $e-1$	3.82 $e-1$	7.534 $e-2$	6.12 $e-3$
RVEA	3.673 $e+0$(=)	7.29 $e-3$	**1.400** $e-$ **3(−)**	4.66 $e-4$	**5.830** $e-$ **2(−)**	2.09 $e-3$
KnEA	2.213 $e+0$(+)	6.06 $e-1$	3.288 $e+0$(+)	4.90 $e+0$	4.688 $e-1$(+)	1.73 $e-1$
MaF10						
IMOEA/DA	**5.973** $e+$ **1**	3.02 $e+0$	**4.245** $e-$ **3**	5.54 $e-4$	**8.402** $e-$ **2**	1.47 $e-3$
RVEA	5.946 $e+1$(=)	3.28 $e-1$	8.418 $e-3$(+)	9.19 $e-4$	1.789 $e-1$(+)	1.60 $e-2$
KnEA	5.909 $e+1$(=)	2.46 $e-1$	4.796 $e-3$(=)	3.02 $e-4$	1.934 $e-1$(+)	9.82 $e-3$
MaF11						
IMOEA/DA	5.941 $e+1$	1.25 $e-1$	**5.267** $e-$ **3**	3.28 $e-4$	5.309 $e-1$	5.03 $e-2$
RVEA	5.940 $e+1$(=)	4.21 $e-2$	1.009 $e-2$(+)	2.77 $e-3$	**1.802** $e-$ **1(−)**	5.70 $e-3$
KnEA	**5.943** $e+$ **1(=)**	2.24 $e-1$	6.080 $e-3$(+)	6.23 $e-4$	1.944 $e-1$(−)	2.32 $e-2$
MaF12						
IMOEA/DA	**3.482** $e+$ **1**	5.74 $e-1$	4.798 $e-3$	1.48 $e-3$	2.079 $e-1$	1.18 $e-3$
RVEA	3.464 $e+1$(=)	1.06 $e-1$	**4.227** $e-$ **3(=)**	2.47 $e-4$	**2.080** $e-$ **1(=)**	1.21 $e-3$
KnEA	3.394 $e+1$(+)	9.19 $e-2$	4.331 $e-3$(=)	3.34 $e-4$	2.250 $e-1$(+)	4.76 $e-3$

Table 4: Continued.

Problems	HV Mean	HV Std	GD Mean	GD Std	IGD Mean	IGD Std
MaF13						
IMOEA/DA	**6.915e − 1**	1.11e − 2	5.644e − 2	7.32e − 2	**7.322e − 2**	7.99e − 3
RVEA	6.846e − 1(+)	1.89e − 2	**5.526e − 3(−)**	1.76e − 3	7.418e − 2(=)	8.83e − 3
KnEA	6.078e − 1(+)	1.06e − 2	7.340e − 3(−)	2.66e − 3	1.347e − 1(+)	1.99e − 2
MaF14						
IMOEA/DA	**6.032e − 1**	1.30e − 1	4.765e + 1	9.12e + 1	**2.780e − 1**	8.56e − 2
RVEA	1.645e − 2(+)	3.68e − 2	5.541e + 2(+)	7.74e + 2	1.154e + 0(+)	2.53e − 1
KnEA	3.082e − 1(+)	7.70e − 2	**4.478e − 1(−)**	3.24e − 1	5.489e − 1(+)	8.49e − 2
MaF15						
IMOEA/DA	**4.363e − 1**	3.85e − 2	1.014e + 0	2.57e − 1	**3.547e − 1**	2.66e − 2
RVEA	0.000e + 0(+)	0.00e + 0	4.784e + 0(+)	3.50e − 1	1.013e + 0(+)	2.22e − 1
KnEA	1.010e − 1(+)	6.87e − 2	**2.813e − 1(−)**	1.98e − 1	5.712e − 1(+)	2.19e − 1

"+" means that IMOEA/DA outperforms its competitor algorithm, "−" means that IMOEA/DA is worse than its competitor algorithm, and "=" means that the competitor algorithm has the same performance as IMOEA/DA.

are adaptively set, which can well maintain the uniformity of weight vectors. And the above comparisons declare that IMOEA/DA is better at maintaining the diversity of obtained solutions than MOEA/D-AWA.

4.4.3. Comparisons of IMOEA/DA with RVEA and KnEA on MaF Problems. To demonstrate the effectiveness of the proposed algorithm MaOEAIR2, the benchmark suit for CEC2018 MaOP competition [52] is chosen, which have diverse characteristics and can test the strengths and weaknesses of MOEAs. There are fifteen many-objective benchmark functions (MaF) with box constrains in the solution space in this benchmark suit. In this work, for each test problem, the number of objectives is set to 3. RVEA [41] and KnEA [48] are used to compare with our algorithm. KnEA is a knee point-driven EA to enhance the convergence performance in many-objective optimization. RVEA uses the reference vectors to decompose the original multiobjective optimization problem into a number of single-objective subproblems and elucidate user preferences to target a preferred subset of the whole Pareto front. The codes of RVEA and KnEA are obtained from PlatEMO [57]. The initial population and initial weight vectors sizes are set to 105 for these fifteen problems; each algorithm is run 30 times with the maximal number of function evaluations 100,000 on all test problems.

Table 4 shows the mean and standard deviation values of IGD metric obtained by IMOEA/DA, RVEA, and KnEA on these fifteen test problems. We can get from Table 4 that IMOEA/DA is worse than RVEA and KnEA on none test problem in the form of the HV metric; these indicate that the diversity of solutions obtained by IMOEA/DA is better than RVEA and KnEA; in the form of the GD metrics, IMOEA/DA outperforms RVEA on ten test problems and outperforms KnEA on five test problems; these imply that the solutions obtained by IMEOA/DA have a good convergence; the mean values of IGD metric obtained by IMOEA/DA are smaller than those obtained by KnEA on all fifteen test problems and are smaller than those obtained by RVEA on all fifteen test problems except for problems MaF3, MaF9, MaF11, and MaF12, which indicate that the quality of the final solutions obtained by IMOEA/DA is better than those obtained by KnEA and RVEA on fifteen problems and eleven problems, respectively. These comparisons imply that IMOEA/DA is better at solving MOPs with complex PFs than RVEA and KnEA on most problems of these problems.

4.5. Performances on MOPS with Complex PFs and Many-Objective Problems. For the problems WFG2, UF5, UF6, ZDT3, and DTLZ7 which have discontinuous PF, the above experimental results show that, though IMOEA/DA cannot well solve problem UF5, IMOEA/DA can well solve other four problems; IMOEA/DA can obtain better performances than NSGAII and MOEA/D on problems WFG2, UF5, UF6, and DTLZ7; IMOEA/D can obtain better performances than MOEA/D-AWA on problems ZDT3 and DTLZ7. These show that IMOEA/DA can obtain good performances on most problems of the MOPs with discontinuous PF.

To study the performance of IMOEA/DA on problems with sharp peak or low tail PFs, we test the problems F1 and F2. The ideal PFs of F1 and F2 are $\{f_1 f_2 \mid (1-f_1)^{2.8} + (1-f_2)^{2.8} = 1, f_1, f_2 \in [0,1]\}$, $\{f_1 f_2 f_3 \mid \sqrt{f_1} + \sqrt{f_2} + f_3 = 1, f_1, f_2, f_3 \in [0,1]\}$, respectively. It can be seen from Table 3 that the performance of IMOEA/DA is worse than that of MOEA/D-AWA on problem F2 and is better than that of MOEA/D-AWA on problem F1. Figure 5 shows the distribution of the final nondominated fronts obtained by IMOEA/DA on problems F1 and F2. As shown in Figure 5, IMOEA/DA can obtain a set of nondominated solutions with good uniformity. These results suggest that the weight adjustment of IMOEA/DA does improve MOEA/D significantly in the terms of uniformity for the MOPs with complex PFs. The

Table 5: HV, GD, and IGD obtained by MOEA/D with three the initialization methods.

Problems	HV Mean	HV Std	IGD Mean	IGD Std
UF8				
MOEA/D1	0.6094(+)	0.0426	0.1246(+)	0.0395
MOEA/D2	0.6388(+)	0.0153	0.1319(+)	0.0262
MOEA/D3	**0.6421**	0.0315	**0.1165**	0.0241
UF9				
MOEA/D1	0.8036(+)	0.0954	0.1549(+)	0.1036
MOEA/D2	0.8996(=)	0.0765	0.1082(=)	0.0537
MOEA/D3	**0.9064**	0.0961	**0.1127**	0.0886
UF10				
MOEA/D1	0.3716(+)	0.1214	0.3145(+)	0.0614
MOEA/D2	0.3718(+)	0.0962	0.2245(+)	0.0443
MOEA/D3	**0.4881**	0.0634	**0.1665**	0.0476
DTLZ1				
MOEA/D1	0.0904(+)	0.0004	0.0260(+)	0.0005
MOEA/D2	0.0934(=)	0.0003	0.0244(+)	0.0002
MOEA/D3	**0.0950**	0.0009	**0.0201**	0.0004
DTLZ2				
MOEA/D1	0.6716(+)	0.0031	0.0680(+)	0.0004
MOEA/D2	0.6989(+)	0.0023	0.0632(+)	0.0006
MOEA/D3	**0.7041**	0.0061	**0.0612**	0.0009
DTLZ3				
MOEA/D1	0.6813(+)	0.0046	0.0624(+)	0.0009
MOEA/D2	0.6983(+)	0.0032	0.0633(+)	0.0004
MOEA/D3	**0.7105**	0.0068	**0.0580**	0.0008
DTLZ4				
MOEA/D1	0.6954(+)	0.0058	0.0651(+)	0.0008
MOEA/D2	0.6997(=)	0.0038	0.0632(+)	0.0005
MOEA/D3	**0.7012**	0.0045	**0.0604**	0.0012
DTLZ5				
MOEA/D1	0.4214(+)	0.0003	0.0091(+)	0.0002
MOEA/D2	0.4366(=)	0.0001	0.0071(=)	0.0002
MOEA/D3	**0.4374**	0.0003	**0.0061**	0.0004
DTLZ6				
MOEA/D1	0.3410(+)	0.0001	0.0830(+)	0.0001
MOEA/D2	0.3461(+)	0.0300	0.0820(+)	0.0289
MOEA/D3	**0.3815**	0.0020	**0.0101**	0.0042
DTLZ7				
MOEA/D1	1.4263(+)	0.0716	0.1925(+)	0.0861
MOEA/D2	1.5107(+)	0.0617	0.1790(+)	0.1156
MOEA/D3	**1.5846**	0.0604	**0.0912**	0.0754

"+" means that MOEA/D3 outperforms its competitor algorithm, "–" means that MOEA/D3 is worse than its competitor algorithm, and "=" means that the competitor algorithm has the same performance as MOEA/D3.

possible reasons for the success of IMOEA/DA are that it increases the number of subproblems of those regions with the sharp peak or low tail PFs and it treats each subproblem equally.

To test the ability of the IMOEA/DA on many-objective problems, two test problems DTLZ5(3,6) and its variation DTLZ4(3,6) are used. They are many-objective problems with low-dimensional PF in the objective space; thus, their PFs are convenient for the visual display of the distribution of solutions. The ideal PFs of DTLZ5(3,6) and DTLZ4(3,6) are described as follows: $\{(f_1, f_2, f_3, f_4, f_5) \mid f_4 = \sqrt{2} f_3 = 2 f_1 = 2f_2, 2 * f_4^2 + f_5^2 + f_6^2 = 1, f_1, f_5, f_6 \in [0, 1]\}$. It can be seen from Table 5 that the performance of IMOEA/DA is better than that of MOEA/D-AWA on these two problems. Figure 5 shows the distribution of the final nondominated fronts obtained by IMOEA/DA on these two problems. As shown in Figure 5, IMOEA/DA can obtain a set of nondominated solutions with good uniformity and convergence. The possible reason for the success of IMOEA/DA on many-objective problems is that IMOEA/DA with the help of the proposed weight adjustment can obtain a good diversity. The computing efforts in IMOEA/DA are evenly distributed and have no preference to the boundary solutions and nondominated solutions.

4.6. Major Contributions of the Proposed Algorithm. This paper has two major contributions; one is the initialization method of weight vectors, and the other is the adaptive weight vector adjustment strategy. In this subsection, their effects are discussed. The initialization method of weight vectors has a great effect on three or many objective problems. So, in the experiments, the test problems are three-objective problems of Subsection 4.4.1 and the parameters are the same as Subsection 4.4.1. To study the effect of the initialization method of weight vectors, we compare the performances of MOEA/D with initialization method [44], initialization method [16], and the proposed method. For convenience, these three MOEA/D algorithms are denoted as MOEA/D1, MOEA/D2, and MOEA/D3, respectively. Table 5 displays the mean and standard deviation values of HV and IGD metrics obtained by three MOEA/D algorithms on ten three-objective problems. We can obtain from Table 5 that, in the form of the HV and GD metrics, MOEA/D3 outperforms MOEA/D1 and is not worse than MOEA/D2 on all ten test problems, and the mean values of HV and IGD metrics obtained by MOEA/D3 are better than MOEA/D1 and MOEA/D2, which can indicate that the diversity of solutions obtained by MOEA/D3 is better than those obtained MOEA/D1 and MOEA/D2. These comparisons imply that the uniformity of weight vectors generated by the proposed method is preferable than the methods [16, 44].

The major role of the adaptive weight vector adjustment strategy is to maintain the diversity of obtained solutions. To identify this, IMOEA/DA is compared with IMOEA/DA without t the adaptive weight vector adjustment strategy which is denoted as IMOEA/D. In this experiment, the test problem and parameters are the same as those found in Subsection 4.4.1. Table 6 shows the mean and standard deviation values of HV, GD, and IGD metrics obtained by IMOEA/DA and IMOEA/D on these twenty-six problems. It can be seen from Table 6 that IMOEA/DA obtains the best results on all these problems in the form of the HV and IGD metrics

Table 6: HV, GD and IGD obtained by IMOEA/DA and IMOEA/D.

Problems	HV		GD		IGD	
	Mean	Std	Mean	Std	Mean	Std
WFG1						
IMOEA/DA	**6.7512**	0.0008	0.0014	0.0001	**0.0125**	0.0002
IMOEA/D	6.7325(+)	0.0009	0.0014(=)	0.0001	0.0214(+)	0.0009
WFG2						
IMOEA/DA	**6.1411**	0.0002	**0.0065**	0.0006	**0.0109**	0.0003
IMOEA/D	6.0824(+)	0.0010	0.0070(+)	0.0006	0.0153(+)	0.0014
WFG3						
IMOEA/DA	**5.6356**	0.0006	0.0023	0.0003	**0.0134**	0.0004
IMOEA/D	5.6179(+)	0.0009	**0.0022**(=)	0.0005	0.0256(+)	0.0005
WFG4						
IMOEA/DA	**3.3611**	0.0017	0.0025	0.0002	**0.0131**	0.0003
IMOEA/D	3.3456(+)	0.0015	**0.0022**(−)	0.0003	0.0161(+)	0.0003
WFG5						
IMOEA/DA	**3.0364**	0.0614	0.0533	0.0145	**0.0575**	0.0133
IMOEA/D	2.9432(+)	0.0712	**0.0504**(−)	0.0104	0.0724(+)	0.0152
WFG6						
IMOEA/DA	**3.3640**	0.0010	**0.0025**	0.0002	**0.0133**	0.0006
IMOEA/D	3.3512(+)	0.0028	0.0026(=)	0.0002	0.0138(+)	0.0002
WFG7						
IMOEA/DA	**3.3634**	0.0009	0.0025	0.0003	**0.0139**	0.0002
IMOEA/D	3.3571(+)	0.0004	0.0025(=)	0.0003	0.0142(+)	0.0003
WFG8						
IMOEA/DA	**3.3504**	0.0016	0.0048	0.0003	**0.0147**	0.0003
IMOEA/D	3.3314(+)	0.0040	**0.0041**(−)	0.0003	0.0151(+)	0.0004
WFG9						
IMOEA/DA	**3.3101**	0.0005	0.0025	0.0002	**0.0145**	0.0006
IMOEA/D	3.2914(+)	0.0097	0.0025(=)	0.0003	0.0161(+)	0.0006
UF1						
IMOEA/DA	**0.8652**	0.0038	0.0068	0.0004	**0.0070**	0.0027
IMOEA/D	0.8312(+)	0.0057	**0.0050**(−)	0.0005	0.0079(+)	0.0034
UF2						
IMOEA/DA	**0.8695**	0.0002	0.0067	0.0023	**0.0050**	0.0002
IMOEA/D	0.8645(+)	0.0017	**0.0045**(+)	0.0002	0.0075(+)	0.0013
UF3						
IMOEA/DA	**0.8620**	0.0112	**0.0077**	0.0009	**0.0106**	0.0105
IMOEA/D	0.8315(+)	0.0215	0.0092(+)	0.0004	0.0124(+)	0.0217
UF4						
IMOEA/DA	**0.5386**	0.0001	0.0036	0.0001	**0.0035**	0.0001
IMOEA/D	0.5371(+)	0.0003	0.0036(=)	0.0001	0.0036(+)	0.0001
UF5						
IMOEA/DA	**0.3538**	0.0939	**0.0427**	0.0487	**0.2350**	0.0567
IMOEA/D	0.3102(+)	0.1124	0.0451(+)	0.0348	0.3147(+)	0.0937
UF6						
IMOEA/DA	**0.6357**	0.0114	**0.0195**	0.0103	**0.0070**	0.0024
IMOEA/D	0.6143(+)	0.0201	0.0224(+)	0.0031	0.0084(+)	0.0012
UF7						
IMOEA/DA	**0.7026**	0.0003	0.0043	0.0002	**0.0046**	0.0002
IMOEA/D	0.6912(+)	0.0002	**0.0041**(−)	0.0002	0.0061(+)	0.0003

Table 6: Continued.

Problems	HV Mean	HV Std	GD Mean	GD Std	IGD Mean	IGD Std
UF8						
IMOEA/DA	**0.6861**	0.0194	0.0549	0.0226	**0.0953**	0.0119
IMOEA/D	0.6432(+)	0.0184	**0.0415**(−)	0.0224	0.1117(+)	0.0215
UF9						
IMOEA/DA	**0.9420**	0.0827	0.2133	0.1745	**0.0959**	0.0541
IMOEA/D	0.9010(+)	0.0715	**0.1421**(−)	0.1532	0.1091(+)	0.0540
UF10						
IMOEA/DA	**0.5102**	0.0691	0.0549	0.0891	**0.1513**	0.0424
IMOEA/D	0.3718(+)	0.0962	0.0694(+)	0.0407	0.1674(+)	0.0469
DTLZ1						
IMOEA/DA	**0.0962**	0.0007	**0.0072**	0.0001	**0.0199**	0.0009
IMOEA/D	0.0952(+)	0.0003	0.0084(+)	0.0001	0.0200(+)	0.0002
DTLZ2						
IMOEA/DA	**0.7138**	0.0051	0.0214	0.0005	**0.0550**	0.0030
IMOEA/D	0.7004(+)	0.0042	**0.0190**(−)	0.0002	0.0570(+)	0.0024
DTLZ3						
IMOEA/DA	**0.7196**	0.0051	0.0204	0.0010	**0.0514**	0.0030
IMOEA/D	0.7069(+)	0.0020	**0.0194**(−)	0.0007	0.0597(+)	0.0004
DTLZ4						
IMOEA/DA	**0.7234**	0.0062	0.0246	0.0040	**0.0544**	0.0014
IMOEA/D	0.7043(+)	0.0038	**0.0187**(−)	0.0006	0.0561(+)	0.0015
DTLZ5						
IMOEA/DA	**0.4394**	0.0003	0.0009	0.0003	**0.0047**	0.0002
IMOEA/D	0.4300(+)	0.0001	**0.0009**(=)	0.0002	0.0061(+)	0.0002
DTLZ6						
IMOEA/DA	**0.4412**	0.0001	0.0004	0.0001	**0.0045**	0.0001
IMOEA/D	0.3947(+)	0.0007	**0.0003**(−)	0.0001	0.0057(+)	0.0004
DTLZ7						
IMOEA/DA	**1.6576**	0.0074	0.0143	0.0028	**0.0596**	0.0046
IMOEA/D	1.5915(+)	0.0091	**0.0097**(−)	0.0038	0.0642(+)	0.0054

"+" means that IMOEA/DA outperforms IMOEA/D, "−" means that IMOEA/DA is worse than IMOEA/D, and "=" means that IMOEA/D has the same performance as IMOEA/DA.

and the results of Wilcoxon rank-sum test show that IMOEA/DA outperforms IMOEA/D on all test problems; these indicate that the diversity of solutions obtained by IMOEA/DA is better than those obtained by IMOEA/D and the proposed adaptive weight vector adjustment strategy can help obtained solutions to maintain the diversity; in the form of the GD metric, the convergence of solutions obtained by IMOEA/DA is worse than those obtained by IMOEA/D, which is because that instability in search direction of IMOEA/DA leads to a decrease in convergence speed.

5. Conclusions

In this paper, an improved decomposition-based evolutionary algorithm with adaptive weight adjustment is designed to solve multiobjective problems with complex PFs. The goal of the proposed algorithm is to enhance the diversity of obtained solutions. In this work, a new method based on uniform design and crowding distance is designed to generate a uniformity of weight vectors; an adaptive weight adjustment strategy which some weight vectors are adaptively deleted or added according to the distances of obtained nondominated solutions is proposed to adaptively change the weight vectors; a selection strategy is used to help each subobjective space to obtain a nondominated solution (if have). Moreover, the proposed algorithm tests thirty-five test problems and compares with six well-known algorithms NSGAII, MOEA/D, MOEA/D-AWA, EMOSA, RVEA, and KnEA. Simulation results show that IMOEA/DA has competitive performances on most test problems against six comparison MOEAs (i.e., NSGAII, MOEA/D, MOEA/D-AWA, EMOSA, RVEA, and KnEA. These results also imply that the proposed weight adjustment can help the proposed algorithm to obtain a good

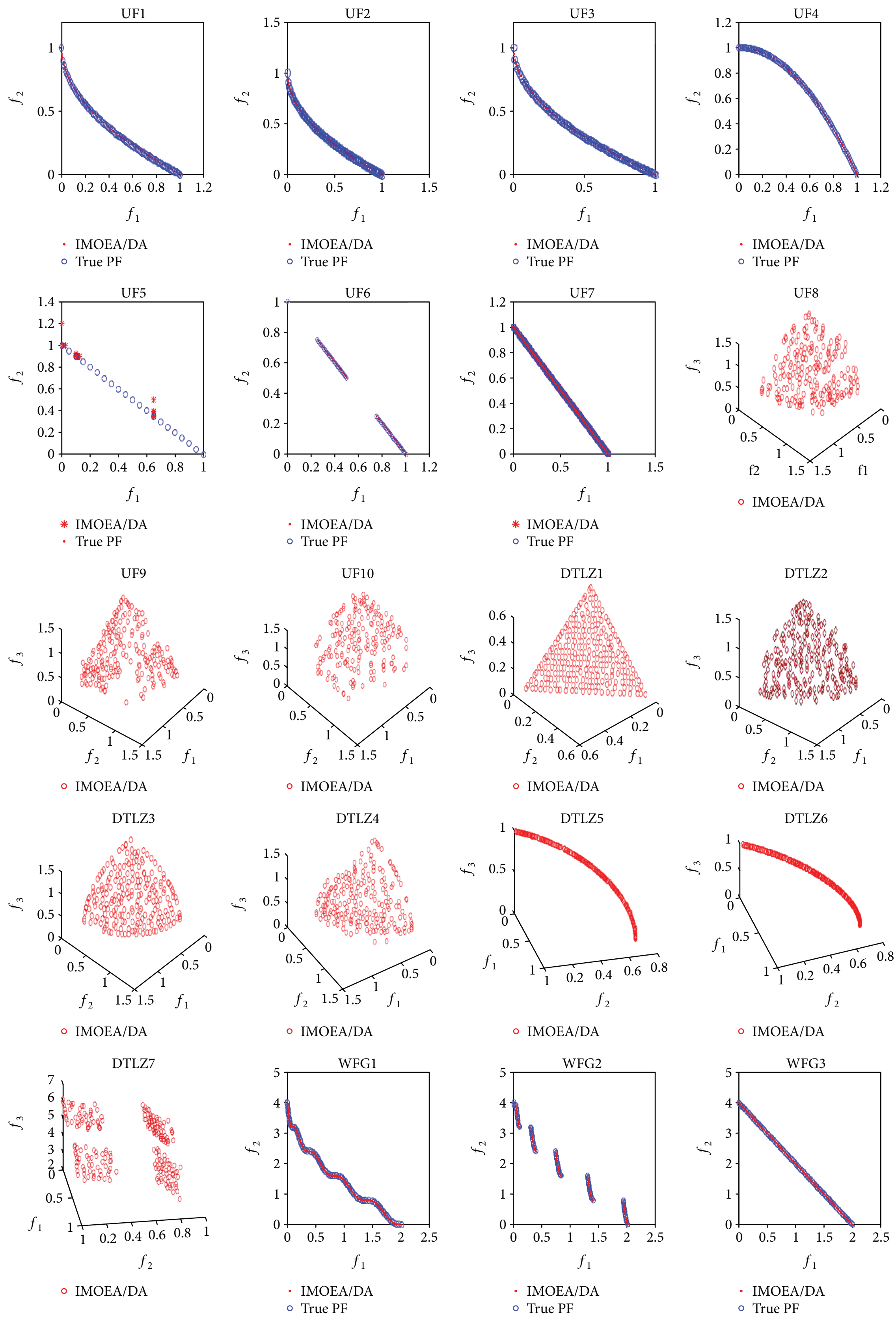


FIGURE 3: According to the median values of IGD metric, nondominated solutions obtained by IMOEA/DA on ten UF problems, seven DTLZ problems, and WFG1–WFG3.

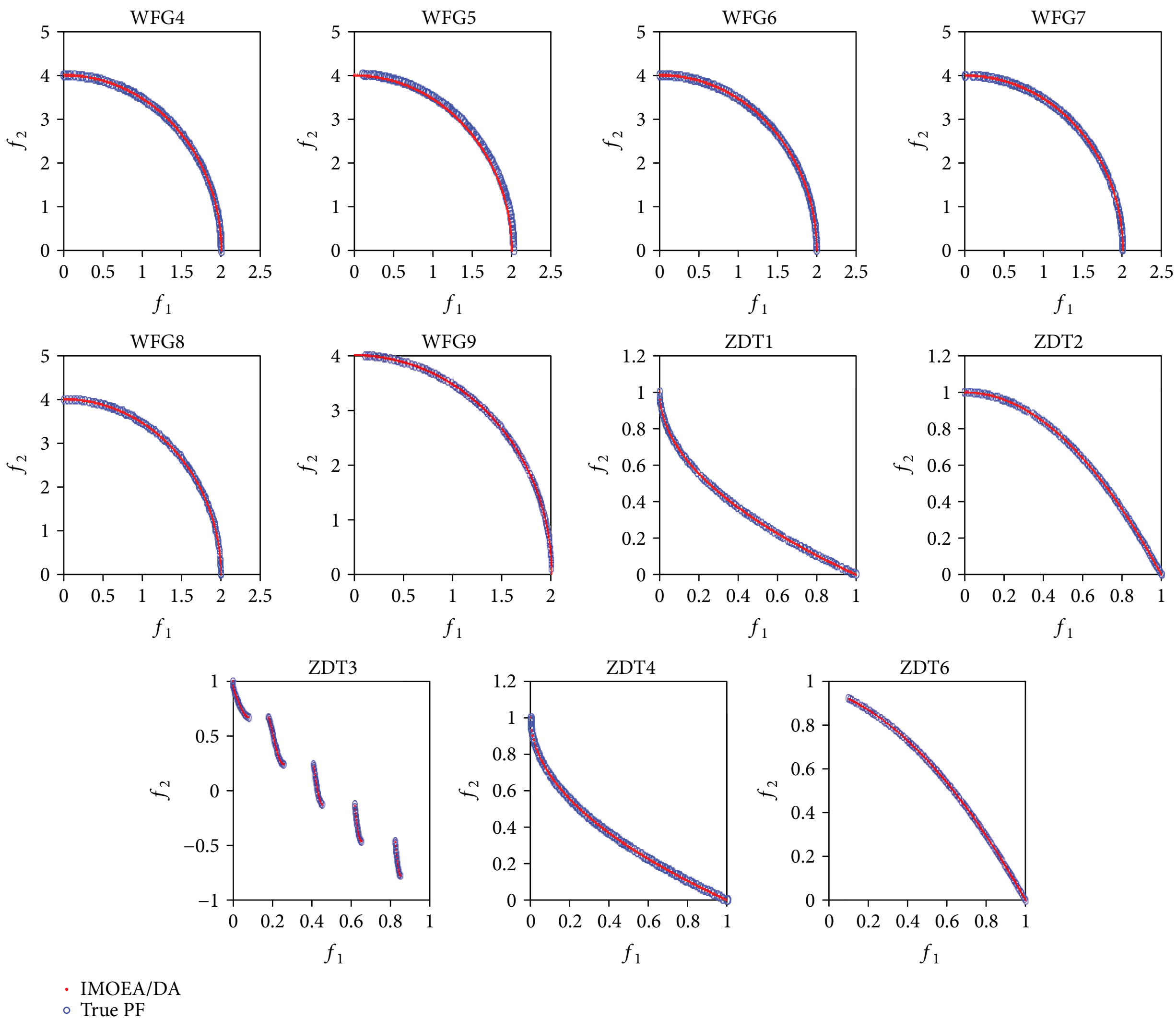


Figure 4: According to the median values of IGD metric, nondominated solutions obtained by IMOEA/DA on WFG4–WFG9 and five ZDT problems.

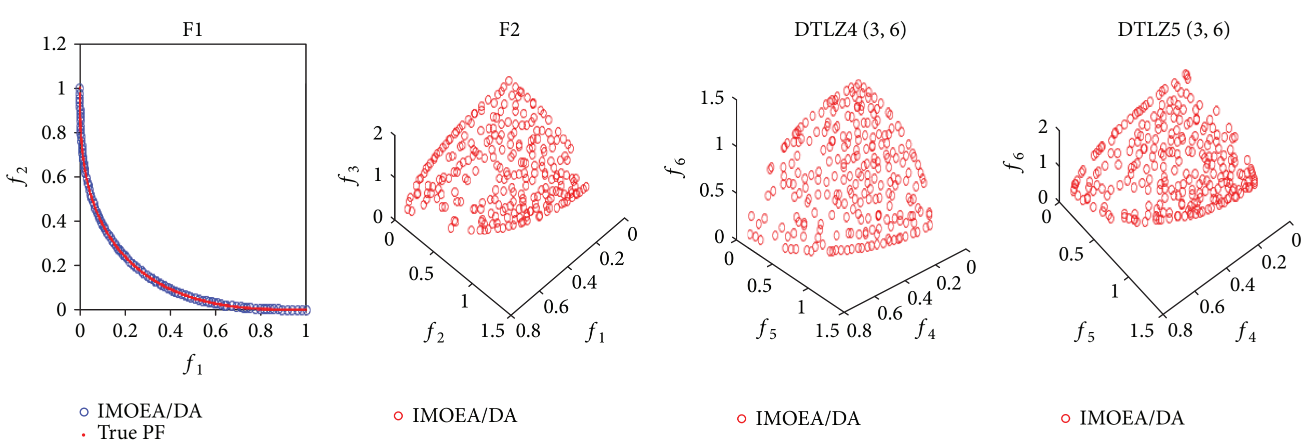


Figure 5: According to the median values of IGD metric, nondominated solutions obtained by IMOEA/DA on F1, F2, and two six-objective problems.

diversity for MOPs with complex PFs, and the decomposition-based multiobjective evolutionary algorithm with adaptive weight adjustment can obtain a set of solutions with good diversity on some MOPs with complex PFs (i.e., PFs with a sharp peak or low tail or discontinuous PFs). A further study of the proposed method needs to be developed for its application in optimization with complicated PFs and high-dimensional optimization problems.

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.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 61502290, 61401263, and 61672334), China Postdoctoral Science Foundation (no. 2015M582606), Industrial Research Project of Science and Technology in Shaanxi Province (no. 2015GY016), Fundamental Research Funds for the Central Universities (nos. GK201603094 and GK201603002), Natural Science Basic Research Plan in Shaanxi Province of China (nos. 2016JQ6045 and 2015JQ6228), and Special Scientific Research Project Fund of the Education Department of Shaanxi Province (no. 16JK1381).

References

[1] D. A. Van Veldhuizen, *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*, Air Force Institute of Technology Wright Patterson AFB, Wright-Patterson AFB, OH, USA, 1999.

[2] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.

[3] D. F. Jones, S. K. Mirrazavi, and M. Tamiz, "Multi-objective meta-heuristics: an overview of the current state-of-the-art," *European Journal of Operational Research*, vol. 137, no. 1, pp. 1–9, 2002.

[4] A. M. Zhou, B. Y. Qu, H. Li, S. Z. Zhao, P. N. Suganthan, and Q. F. Zhang, "Multiobjective evolutionary algorithms: a survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.

[5] K. Deb, A. Pratap, S. Agrawal, 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.

[6] N. Chen, W. N. Chen, Y. J. Gong et al., "An evolutionary algorithm with double-levelarchives for multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 45, no. 9, pp. 1851–1863, 2015.

[7] M. Kiani and S. H. Pourtakdoust, "State estimation of nonlinear dynamic systems using weighted variance-based adaptive particle swarm optimization," *Applied Soft Computing*, vol. 34, pp. 1–17, 2015.

[8] H.-L. Liu, L. Chen, Q. Zhang, and K. Deb, "Adaptively allocating search effort in challenging many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 3, pp. 433–448, 2018.

[9] B. Xue, M. J. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: a multi-objective approach," *IEEE transactions on cybernetics*, vol. 43, no. 6, pp. 1656–1671, 2013.

[10] Q. H. Lin, Q. F. Zhu, P. Huang, J. Chen, Z. Ming, and J. Yu, "A novel hybrid multi-objective immune algorithm with adaptive differential evolution," *Computer & Operations Research*, vol. 62, pp. 95–111, 2015.

[11] X. P. Wang and L. X. Tang, "An adaptive multi-population differential evolution algorithm for continuous multi-objective optimization," *Information Sciences*, vol. 348, pp. 124–141, 2016.

[12] R. H. Shang, L. C. Jiao, F. Liu, and W. P. Ma, "A novel immune clonal algorithm for MO problems," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, pp. 35–50, 2012.

[13] Z. H. Zhan, J. J. Li, J. N. Cao, J. Zhang, H. H. Chung, and Y. H. Shi, "Multiple populations for multiple objectives: a coevolutionary technique for solving multiobjective optimization problems," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 445–463, 2013.

[14] H. Zhang, X. J. Zhang, X. Z. Gao, and S. M. Song, "Self-organizing multiobjective optimization based on decomposition with neighborhood ensemble," *Neurocomputing*, vol. 173, pp. 1868–1884, 2016.

[15] S. Jiang and S. Yang, "An improved multiobjective optimization evolutionary algorithm based on decomposition for complex Pareto fronts," *IEEE Transactions on Cybernetics*, vol. 46, no. 2, pp. 421–437, 2016.

[16] Q. F. Zhang 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.

[17] M. Elarbi, S. Bechikh, A. Gupta, L. B. Said, and Y.-S. Ong, "A new decomposition-based NSGA-II for many-objective optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 7, pp. 1191–1210, 2018.

[18] L. Wang, Q. Zhang, and A. Zhou, "Constrained subproblems in a decomposition-based multiobjective evolutionary algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 475–480, 2016.

[19] H. Zhu, Z. He, and Y. Jia, "A novel approach to multiple sequence alignment using multiobjective evolutionary algorithm based on decomposition," *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 2, pp. 717–727, 2016.

[20] A. Zhou and Q. Zhang, "Are all the subproblems equally important? Resource allocation in decomposition-sased multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 52–64, 2016.

[21] H. Li and Q. F. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.

[22] N. Al Mpubayed, A. Petrovski, and J. McCall, "D^2*MOPSO*: MOPSO based on decomposition and dominance with archiving using crowding distance in objective and solution spaces," *Evolutionary Computation*, vol. 22, no. 1, pp. 47–77, 2014.

[23] G. Zeng, J. Chen, L. Li et al., "An improved multi-objective population-based extremal optimization algorithm with polynomial mutation," *Information Sciences*, vol. 330, pp. 49–73, 2016.

[24] A. Menchaca-Mendez and C. A. Coello Coello, "Selection mechanisms based on the maximin fitness function to solve multi-objective optimization problems," *Information Sciences*, vol. 332, pp. 131–152, 2016.

[25] C. Zhu, L. Xu, and E. K. Goodman, "Generalization of Pareto-optimality for many-objective evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 2, pp. 299–315, 2016.

[26] S. Salcedo-Sanz, A. Pastor-Sánchez, J. A. Portilla-Figueras, and L. Prieto, "Effective multi-objective optimization with the coral reefs optimization algorithm," *Engineering Optimization*, vol. 48, no. 6, pp. 966–984, 2016.

[27] S. Bechikh, A. Chaabani, and L. Ben Said, "An efficient chemical reaction optimization algorithm for multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 45, no. 10, pp. 2051–2064, 2015.

[28] H. Xia, J. Zhuang, and D. H. Yu, "Combining crowding estimation in objective and decision space with multiple selection and search strategies for multi-objective evolutionary optimization," *IEEE Transactions on Cybernetics*, vol. 44, no. 3, pp. 378–393, 2014.

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

[30] H. L. Liao and Q. H. Wu, "Multi-objective optimization by learning automata," *Journal of Global Optimization*, vol. 55, no. 2, pp. 459–487, 2013.

[31] R. Saborido, A. B. Ruiz, J. D. Bermudez, E. Vercher, and M. Luque, "Evolutionary multi-objective optimization algorithms for fuzzy portfolio selection," *Applied Soft Computing*, vol. 39, pp. 48–63, 2016.

[32] M. Li, C. Grosan, S. Yang, X. Liu, and X. Yao, "Multiline distance minimization: a visualized many-objective test problem suite," *Multiline Distance Minimization: A Visualized Many-Objective Test Problem Suite*, vol. 22, no. 1, pp. 61–78, 2018.

[33] C. Lu and M. Mandal, "Automated analysis and diagnosis of skin melanoma on whole slide histopathological images," *Pattern Recognition*, vol. 48, no. 8, pp. 2738–2750, 2015.

[34] L. H. Liao, Q. H. Wu, Y. Z. Li, and L. Jiang, "Economic emission dispatching with variations of wind power and loads using multi-objective optimization by learning automata," *Energy Conversion and Management*, vol. 87, pp. 990–999, 2014.

[35] 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.

[36] D. Cai and L. Xiujuan, "An improvement based evolutionary algorithm with adaptive weight adjustment for many-objective optimization," in *2017 13th International Conference on Computational Intelligence and Security (CIS)*, Hong Kong, December 2017.

[37] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with adaptive weight adjustment," *Evolutionary Computation*, vol. 22, no. 2, pp. 231–264, 2014.

[38] F. Gu and H. Liu, "A novel weight design in multi-objective evolutionary algorithm," in *2010 International Conference on Computational Intelligence and Security*, pp. 137–141, Nanning, China, December 2010.

[39] S. Jiang, Z. Cai, J. Zhang, and Y. S. Ong, "Multiobjective optimization by decomposition with Pareto-adaptive weight vectors," in *2011 Seventh International Conference on Natural Computation*, vol. 3, pp. 1260–1264, Shanghai, China, July 2011.

[40] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated 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.

[41] 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.

[42] 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, 2017.

[43] K. T. Fang and Y. Wang, *Number-Theoretic Method in Statistics*, Chapman and Hall, London, 1994.

[44] C. Dai and Y. Wang, "A new uniform evolutionary algorithm based on decomposition and CDAS for many-objective optimization," *Knowledge-Based Systems*, vol. 85, pp. 131–142, 2015.

[45] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," in *2009 IEEE Congress on Evolutionary Computation*, pp. 203–208, Trondheim, Norway, May 2009.

[46] B. L. Miller and D. E. Goldberg, "Genetic algorithms, tournament selection, and the effects of nose," *Complex Systems*, vol. 9, pp. 193–212, 1995.

[47] K. Deb, *Multiobjective Optimization Using Evolutionary Algorithms*, Wiley, New York, NY, USA, 2001.

[48] X. Zhang, Y. Tian, and Y. Jin, "A knee point-driven evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 761–776, 2015.

[49] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, pp. 825–830, Honolulu, HI, USA, May 2002.

[50] Q. F. Zhang and P. N. Suganthan, "Final report on CEC'09 MOEA competition," Tech. Rep. the School of CS and EE, University of Essex, UK and School of EEE, Nangyang Technological University, Singapore, 2009, http://web.mysites.ntu.edu.sg/epnsugan/PublicSite/Shared%20Documents/CEC2009-MOEA/PDF-Tech-Report.pdf.

[51] 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.

[52] R. Cheng, M. Li, Y. Tian et al., "Benchmark functions for CEC'2018 competition on many-objective optimization," Tech. Rep. CERCIA, School of Computer Science, University of Birmingham Edgbaston, Birmingham, U. K., 2017.

[53] K. Deb and D. K. Saxena, "On finding Pareto-optimal solutions through dimensionality reduction for certain large-

dimensional multi-objective optimization problems," Tech. Rep. 1005011, KanGAL, 2005.

[54] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.

[55] K. Deb, A. Sinha, and S. Kukkonen, "Multi-objective test problems, linkages, and evolutionary methodologies," in *GECCO '06 Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pp. 1141–1148, Seattle, WA, USA, July 2006.

[56] S. Robert, J. Torrie, and D. Dickey, *Principles and Procedures of Statistics: A Biometrical Approach*, McGraw-Hill, New York, NY, USA, 1997.

[57] 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.