Research Article

A Many-Objective Optimization Algorithm Based on Weight Vector Adjustment

Yanjiao Wang and Xiaonan Sun

School of Electrical Engineering, Northeast Electric Power University, Jilin 132000, Jilin, China

Correspondence should be addressed to Xiaonan Sun; 15043222562@163.com

Received 29 June 2018; Revised 29 August 2018; Accepted 13 September 2018; Published 22 October 2018

Academic Editor: Michele Migliore

Copyright © 2018 Yanjiao Wang and Xiaonan Sun. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In order to improve the convergence and distribution of a many-objective evolutionary algorithm, this paper proposes an improved NSGA-III algorithm based on weight vector adjustment (called NSGA-III-WA). First, an adaptive weight vector adjustment strategy is proposed to decompose the objective space into several subspaces. According to different subspace densities, the weight vector is sparse or densely adjusted to ensure the uniformity of the weight vector distribution on the Pareto front surface. Secondly, the evolutionary model that combines the new differential evolution strategy and genetic evolution strategy is proposed to generate new individuals and enhance the exploration ability of the weight vector in each subspace. The proposed algorithm is tested on the optimization problem of 3–15 objectives on the DTLZ standard test set and WFG test instances, and it is compared with the five algorithms with better effect. In this paper, the Whitney–Wilcoxon rank-sum test is used to test the significance of the algorithm. The experimental results show that NSGA-III-WA has a good effect in terms of convergence and distribution.

1. Introduction

Many-objective optimization problems (MAOPs) [1] refer to optimization problems whose number of objectives is over three and need to be processed simultaneously. As the number of objectives increases, the number of non-dominated solutions will grow explosively in the form of e exponent [2], most of the solutions are nondominated, the pros and cons between the solutions become more difficult to be evaluated, and namely, many-objective evolutionary algorithms (MOEAs) have poor performance in convergence. In addition, the sensitivity of the Pareto front surface [3] will increase along with the increase of the spatial dimension, which makes it difficult to maintain the distribution among individuals.

In order to ensure the convergence and distribution of MOEAs [4], scholars have proposed the following three solutions to improve the aforementioned problems:

- (1) Change the dominance relationship [5], and then increase the pressure of selecting solution to improve

the convergence of the algorithm. In 2005, Laumanns et al. proposed ϵ -domination [6] to reduce the fitness value of each individual by $(1 - \epsilon)$ times before the individual performs the Pareto dominance relationship comparison. In 2014, the ϵ -dominant mechanism proposed by Hernández-Díaz et al. [7] added an acceptable threshold to the comparison of individual fitness. In 2014, Yuan et al. proposed the θ domination [8] to maintain the balance between convergence and diversity in EMO. In 2015, Jingfeng et al. proposed a simple Pareto adaptive ϵ -domination differential evolution algorithm for multi-objective optimization [9]. In 2016, Yue et al. proposed a grid-based evolutionary algorithm [10] which modifies the dominance criterion to increase the convergence speed of evolutionary many-objective optimization (EMO). In 2017, Lin et al. proposed an evolutionary many-objective optimization based on alpha dominance, which provides strict Pareto stratification [11] to remove most of the blocked domination solutions. The methods above

can enhance the selection pressure, but this relaxed strategy is limited to handle the situation with a little number of objectives. Moreover, it is very hard that parameters need to be adjusted for different optimization problems.

- (2) Method based on decomposition [12], which means to decompose the objective space into several subspaces without changing the dimension of the objective, thus transforming MAOP into single-objective subproblem or many-objective subproblems. In 2007, Zhang and Li proposed a many-objective evolutionary algorithm based on decomposition [13] (MOEA/D) for the first time whose convergence effect is significantly better than MOGLS and NSGA-II. In 2016, Yuan et al. proposed a distance-based update strategy, MOEA/DD [14], to maintain the diversity of algorithms in the evolutionary process by exploiting the vertical distance between solutions and weight vectors. In 2017, Segura and Miranda proposed a decomposition-based MOEA/D-EVSD [15] evolutionary algorithm, steady-state form, and a reference direction to guide the search. In 2017, Xiang et al. proposed the framework of VAEA [16] algorithm based on angle decomposition. The algorithm does not require the reference point, and the convergence and diversity of many-objective space are well balanced. However, the self-adjusting characteristics of the algorithms mentioned above make them fall into local optimum more easily although the convergence speed is improved. The distribution is not unsatisfactory.
- (3) The reference point method, this kind of algorithm decomposes the MAOPs into a group of many-objective optimization subproblems with simple frontier surfaces [17]. However, unlike the decomposition method, the subproblem is solved using the many-objective optimization method. In 2013, Wang et al. [18] proposed the PICEA-w-r algorithm whose set of weight vectors evolved along with populations; the weight vector adjusts adaptively according to its own optimal solution. In 2014, Qi et al. adopted an enhanced weight vector adjustment method in the MOEA/D-AWA algorithm [19]. In 2014, Deb and Jain proposed a nondominated sorting evolution many-objective optimization algorithm based on reference points [20] (NSGA-III), and its reference point is uniformly distributed throughout the objective space; in the same year, Liu et al. proposed the MOEA/D-M2M method; the entire PF can be divided into multiple segments and solved separately by dividing the entire objective space into multiple subspaces. Each segment corresponds to a many-objective optimal subproblem [21], which improves the distribution of solution sets. In 2016, Bi and Wang proposed an improved NSGA-III many-objective optimization algorithm [22] (NSGA-III-OSD) based on objective space decomposition. The uniformly distributed weight vector was decomposed into several subspaces through a clustering approach.

The weight vectors can specify a unique subarea. A smaller objective space helps overcome the invalidity of the many-objective Pareto dominance relationship [23]; the distribution and the uniformity of the solution surface decrease because of the sparse solution of each subspace edge caused by the fixed subspace. In 2016, Cheng et al. proposed an evolutionary algorithm based on reference vector guidance for solving MAOPs [24] (RVEA). Its principle of the adaptive strategy is to adjust the weight vectors dynamically according to the objective function form. The weight vectors generated by the methods above are uniformly distributed, while the reference point on the solution surface cannot be guaranteed to be uniform, and there is also a possibility that the convergence may be lost.

In order to further improve the convergence and distribution of many-objective algorithms, based on NSGA-III, a many-objective optimization algorithm (NSGA-III-WA) based on weight vector adjustment is proposed. First, in order to enhance the exploration ability of the solution to the weight vector, an evolutionary model in which a novel differential evolution strategy and a genetic evolution strategy are integrated is used to generate new individuals. Then, in order to ensure the uniform distribution of weight vectors in the solution surface, the objective space is divided into several subspaces by clustering the objective vectors. According to the adjustment of the weight of each subspace, the spatial distribution of the objective is improved. We will carry out simulation experiments on the DTLZ standard test set [25] and WFG standard test set [26]. We compare the proposed algorithm with the five algorithms that are currently performing better on the optimization problem of 3 to 15 objectives. The GD, IGD, and HV are compared as performance indicators. The experimental results show that NSGAWA has good effect in convergence and distribution.

The rest of the paper is organized as follows. Section 2 introduces the original algorithm. Section 3 describes the proposed many-objective evolutionary algorithm. Section 4 compares the similarities and differences between this algorithm and similar algorithms. Section 5 gives the experimental parameters of each algorithm and comprehensive experiments and analysis. Finally, Section 6 summarizes the full text and points out the issues to be studied next.

2. NSGA-III

The NSGA-III algorithm is similar to the NSGA-II algorithm in that it selects individuals based on nondominated ordering. The difference is that the individual choice after nondominated sorting is different. The NSGA-III algorithm is introduced as follows:

First, a population A of size N is set up, and population P_t is operated by genetic operators (selection, reorganization, and variation) to obtain a population Q_t of the same size, and then population P_t and population Q_t are mixed to obtain a population R_t of $2N$.

The population R_t is subjected to nondominated sorting to obtain layers of individuals with nondominated levels ($F1$, $F2$, and so on). Individuals with nondominated levels are sequentially added to the set S_t of the next generation of children until the size of the set S_t is greater than N . The nondominated level at this time is defined as the L layer. Pick K individuals from the L level so that the sum of K and all previous levels is equal to N . Prior to this, the objective value is normalized by the ideal point and the extreme point. After normalization, the ideal point of S_t is a zero vector, and the provided reference point is located exactly on the normalized hyperplane. The vertical distance between each individual in S_t and each weight vector (connect the origin to the reference point) is calculated. Each individual in S_t is then associated with a reference point having a minimum vertical distance.

Finally, the niche operation is used to select members from $F1$. A reference point may be associated with one or more objective vectors or there are also possibilities that none of the objective vectors is associated with a reference point. The purpose of the niche operation is to select the K closest reference points from the $F1$ layer into the next generation. Firstly, calculate the number of individuals associated with each reference point in the $S_t/F1$ population and use ρ_j to represent the number of individuals associated with the j th reference point. The specific operation is as follows:

When the number of individuals associated with a reference point is zero, in other words, ρ_j is equal to zero, the operation next depends on whether there are individuals related to the reference point in F_t . If one or more individuals are related to the reference vector, extract the point with the smallest distance, add it to the next generation, and set $\rho_j = \rho_j + 1$. If no individual is associated with the reference point in F_t , the reference point vector in this generation will be deleted. If $\rho_j > 0$, choose the nearest reference point until the population size is N .

3. The Proposed Algorithm

3.1. The Proposed Algorithm Framework. In order to further improve the convergence speed and distribution of NSGA-III algorithm, a multiobjective optimization algorithm based on weight vector adjustment (NSGA-III-WA) is proposed. Algorithm 1 is the framework of the NSGA-III-WA algorithm. The algorithm is mainly improved in two aspects: evolution strategy and weight vector. This paper also adds the discriminating condition for enabling weight vector adjustment, which speeds up the running of the algorithm without affecting the performance of the algorithm. First, we initialize population P_t with population size N and weight vector W_unit . Secondly, we enter the algorithm iteration process, generate the population Q_t by the operating population P_t using the differential operator, and then obtain population R_t sized $2N$ using the combination of P_t and Q_t . R_t should be updated through the environmental selection strategy. The next generation of population P_{t+1} is obtained. Lastly, adjust the weight vector and determine if the termination condition is satisfied. If so, output the current

result and terminate the algorithm; otherwise, continue iterating.

3.2. Initialization. The initial population is randomly generated whose size is the same as the number of weight vectors in its space. This article uses Das and Dennis's systematic method [27] to set weight vectors $W_unit = \{w^1, w^2, \dots, w^N\}$. The total number of weight vectors is equal to $N = C_{H+M-1}^{M-1}$, where H represents the dimension of the solution vector and M is the number of objective functions. The initialized weight vectors (reference points) are uniformly distributed in the objective space, and each weight vector generation method is as follows: for $w_i \geq 0, i = 1, \dots, H, \sum_{i=1}^H w_i = 1$.

3.3. Evolutionary Strategy. The evolutionary strategy is essential to the convergence speed and accuracy of the solutions because it will determine the quality of new solutions to subquestions directly during evolutionary process. In order to improve the convergence speed, this paper proposes a new differential evolution strategy to replace the original strategy. The pseudocode is shown in Algorithm 2. Every individual performs the same operation as follows.

3.3.1. Variation. It is mainly divided into two parts:

- (a) Select three individuals x_{r1}, x_{r2}, x_{r3} randomly from the population. A new individual x_v will be obtained using (1) for parental vector variation to maintain population diversity. At the beginning of the algorithm, the mutation rate should be relatively large to make the individuals different from each other. This not only improves the search ability of the algorithm, but also prevents the individual from falling into local optimum. The mutation rate should decrease as the number of iterations increases and the solution approaches the Pareto optimal front surface (PFs) [28]. At this time, the mutation rate should decrease to accelerate the convergence of the algorithm to the optimal. This not only improves the convergence speed, but also reduces the complexity of the algorithm. Based on the above analysis of the needs of the algorithm, this paper proposes an adaptive mutated factor $F = 0.5 + 0.5 \cos(\pi \times \text{gen}/\text{gen_max})$. It can be seen that as the number of iterations increases, the mutation rate decreases in size. At the beginning of the algorithm, it can enhance the individual's ability to jump out of local optimum and find superior individuals. As the mutation rate is smaller, the algorithm tends to be stable. To maintain the diversity of populations, select individuals x_{r1}, x_{r2} to generate a new individual (line 3 in Algorithm 2) by simulating the binary recombination operator (2) and (3). In (4), u is a random number between $[0, 1]$ and η is a constant with the fixed value of 20.

Input: N structured reference points W_unit , parent population P_t
Output: P_{t+1}

- (1) Initialization (P_t , W_unit)
- (2) $Gen = 1$
- (3) **While** $Gen \leq Gen_max$ **do**
- (4) $Q_t = \text{Evolutionary_strategy}(P_t)$
- (5) $R_t = P_t \cup Q_t$
- (6) $P_t + 1 = \text{Environmental_selection}(R_t)$
- (7) $W_unit = \text{Weight_Adjustment}(W_unit)$
- (8) $Gen++$
- (9) **End While**
- (10) **Return** P_{t+1}

ALGORITHM 1: Generation framework of the proposed NSGA-III-WA.

Input: Parent population P_t , Mutation rate F , Crossover rate CR
Output: The new population Q_t

- (1) **For** $i = 1 : N$
- (2) Random selection of particles $x_{r1}, x_{r2}, x_{r3} \in P_t$
- (3) $x_v, x_{c1}, x_{c2} \leftarrow x_{r1}, x_{r2}, x_{r3}$
- (4) **If** $\text{rand}() < p$ % Mutation operation
- (5) **If** $\text{rand}() < 0.5$
- (6) $x_t = x_{c2}$
- (7) **else**
- (8) $x_t = x_{c1}$
- (9) **End If**
- (10) **else**
- (11) $x_t = x_v$
- (12) **End If**
- (13) **for** $j = 1 : V$ % Crossover operation
- (14) $k = \text{Random}(1 : V)$
- (15) **If** $\text{rand}() \leq CR \parallel j = k$
- (16) $u(i, j) = t(i, j)$
- (17) **else**
- (18) $u(i, j) = x(i, j)$
- (19) **End If**
- (20) **End for**
- (21) $x_i = x_u$ ($f_1(x_i), f_2(x_i), \dots, f_M(x_i) = (f_1(x_u), f_2(x_u), \dots, f_M(x_u))$)
- (22) **End for**
- (23) $Q_t = \{x_1, x_2, \dots, x_N\}$

ALGORITHM 2: Evolutionary strategy.

$$x_v = x_{r1} + \left(0.5 + 0.5 \cos \left(\pi \times \frac{\text{gen}}{\text{gen_max}} \right) \right) (x_{r2} - x_{r3}), \quad (1)$$

$$x_{c1,j} = 0.5 \left[(1 + \gamma_j) x_{r1,j} + (1 - \gamma_j) x_{r2,j} \right], \quad (2)$$

$$x_{c2,j} = 0.5 \left[(1 - \gamma_j) x_{r1,j} + (1 + \gamma_j) x_{r2,j} \right], \quad (3)$$

$$\gamma_j = \begin{cases} (2u_j)^{1/(\eta+1)} & \text{if } u_j \leq 0.5, \\ \left[\frac{0.5}{(1-u_j)} \right]^{1/(\eta+1)} & \text{otherwise.} \end{cases} \quad (4)$$

(b) Select an individual through probability selection from x_v, x_{c1}, x_{c2} (which are generated in step a) to execute the crossover operation (lines 4 to 12 in Algorithm 2). The specific operation is as follows: Generate a random number between $[0, 1]$ and compare it with p . If the number is smaller than p , execute lines 5–9 of the pseudocode in Algorithm 2; otherwise, select x_v to enter the crossover operation (the detailed operations of lines 5–9 in Algorithm 2 are as follows: Generate a random number between $[0, 1]$ and compare it with 0.5. If it is smaller than 0.5, then select x_{c2} to enter the crossover operation. Otherwise, select x_{c1} to enter the crossover operation). Here, p is 0.5.

3.3.2. Crossover. This article selects the single-point crossing method, which is located in lines 13–20 of the pseudocode in Algorithm 2. In this way, individual selectivity is enhanced. Generate a random number between $[0, 1]$. If the number is less than or equal to the crossover operator CR, then select a certain dimension of the individual randomly and execute the crossover operations at the selected point. A large number of experiments have confirmed that when the value of CR is 0.4, the effect is better. Then use the generated individuals and their fitness values to replace the original ones.

3.4. Environmental Selection. The purpose of the environmental selection operation is to select the next generation of individuals. The framework of Algorithm 3 includes the following steps: (1) Nondominated sorting is conducted on the population R_t , and individuals with a rank of 1, 2, 3, ... after nondominating sorting are added to the offspring collection S_t in order. (2) When the size of S_t is greater than or equal to N , note the nondominant level F_l at this time and determine when to terminate the operation ($|S_t| = N$) or enter the next step ($|S_t| > N$). (3) Select individuals in S_t/F_l and enter P_{t+1} until its size is N . The specific operation is discussed below.

3.4.1. Normalize Objective. Since the magnitude of the respective objective values is different, it is necessary to normalize objective values for the sake of fairness. First, calculate the minimum value z_i of each dimension for every objective function. The sets of z_i constitute the ideal points. All individuals are then normalized according to (5), where a_i is the intercept of each dimension that can be calculated according to the achievement scalarizing function (ASF) shown in (6).

$$f_i^n(x) = \frac{f_i^l(x)}{a_i - z_i^{\min}} = \frac{f_i(x) - z_i^{\min}}{a_i - z_i^{\min}}, \quad \text{for } i = 1, 2, \dots, M, \quad (5)$$

$$\text{ASF}(x, w) = \max_{i=1}^M \frac{(f_i(x) - z_i^{\min})}{w_i}, \quad \text{for } x \in S_t. \quad (6)$$

3.4.2. Associate Each Member of S_t with a Reference Point. In order to associate each individual in S_t with the reference points after normalization, a reference line is defined for each reference point on the hypersurface. In the normalized objective space, the reference point with the shortest distance is considered to be related to population members.

3.4.3. Compute the Niche Count of the Reference Point. Traversing every individual in the population, calculate the distances between itself and all reference points and record the number of individuals associated with each reference point using ρ_j , which represents the number of individuals associated with the j th reference point.

3.4.4. Niche Preservation Operation. If the number of populations associated with this reference point is zero (but there is an individual associated with the reference point vector in F_l), then find the point with the smallest distance and extract it from F_l to join the selected next generation population. In the setting, the number of associated populations is increased by one. If each individual is not referenced to the reference point in the F_b , the reference point vector is deleted. If the number of associated populations is not zero, then the nearest reference point is selected until the population size is N .

3.5. Weight Adjustment. The uniformity of the solution surface cannot be achieved when the algorithm reaches a certain stable state, although the weight vectors distribute uniformly in the space. This is because of the complexity caused by the irregular shape of the PFs of the objective functions. The distribution of weight vectors is particularly important when all individuals are indistinguishable from each other and locate on the first level of the dominance level. Therefore, in order to improve the distribution of many-objective algorithms, a weight vector adjustment strategy whose framework is shown in Algorithm 4 is proposed. The distribution of weight vectors is appropriately adjusted according to the shape of the nondominated frontier. In order to prevent the weight vector adjustment in the high-dimensional space from being concentrated to a certain objective, the K -means clustering method is used to divide the weight vector into different subspaces. The specific operations are described below.

First, each weight vector is associated with the population member (line 1 in Algorithm 4). Secondly, the solution space is decomposed into many subspaces using the K -means clustering method (lines 2–5 in Algorithm 4), as shown in Figure 1. To prevent errors caused by excessive differences in the solution set in space, the subspace should not be too large or small, and it can be divided into C spaces according to the size of the population. A large number of experiments confirmed that when $C = 13$, better results can usually be obtained. The solution set is decomposed into $[N/C]$ cluster spaces, and the weight vectors are adjusted by comparing the density of the entire objective space and subspaces (lines 6–17 in Algorithm 4). As shown in Figure 2, w_2 should be away from w_1 and approach w_3 . Finally, the number of weight vectors is adjusted to ensure that it can match the original number. If the number is greater than N , then the weight vector is deleted at the densest position in the entire objective space. If the number is less than N , then a weight vector is added in sparse position (lines 18–20 in Algorithm 4).

The definition of spatial density is obtained by averaging the distances of similar individuals in the population. The minimum spatial density is defined as $h_1\rho_o$. The maximum spatial density is defined as $h_2\rho_o$. Under normal circumstances, when the value of h_1 is 0.2 and h_2 is 1.3, relatively good results can be obtained. Note ρ_o is the density of the overall objective space and ρ_i is density of the subspace. The adjustment process is divided into two situations described below:

- (1) When the subspace density is less than the objective space density, determine whether the subspace

Input: Combined population $R_t = P_t \cup Q_t$
Output: Offspring population P_{t+1}

- (1) $(F_1, F_2, \dots) = \text{Nondominated sort } (R_t)$
- (2) **repeat**
- (3) $S_t = S_t \cup F_i$ and $i = i + 1$
- (4) **until** $|S_t| \geq N$
- (5) Last front to be included: $F_l = F_i$
- (6) **If** $|S_t| = N$ then
- (7) $P_{t+1} = S_t$, break,
- (8) **else**
- (9) $P_{t+1} = \cup_{j=1}^{l-1} F_j$ and $K = N - |P_{t+1}|$
- (10) Normalize objectives: $P_t = \text{Normalize } (S_t, W_unit)$
- (11) Associate each member s of S_t with a reference point: $[\pi(s), d(s)] = \text{Associate } (S_t, W_unit)$
- (12) Compute niche count of reference point $j \in W_unit$ $\rho_j = \sum_{s \in S_t / F_l} (\pi(s) = j ? 1 : 0)$
- (13) Choose K members one: Niching $(K, \rho_j, \pi, d, W_unit, F_l, P_{t+1})$
- (14) **End If**

ALGORITHM 3: Environmental_selection.

Input: Parent weight W_unit $(w_1, w_2, \dots, w_k)$
Output: Offspring weight W_unit $(w_1, w_2, \dots, w_k)$

- (1) Normalize (P_t, W_unit)
- (2) Cluster objective vectors set $\{Re^1, Re^2, \dots, Re^k\}$
- (3) $\{Re^1, Re^2, \dots, Re^k\} \leftarrow \text{Weight Space Decomposition}()$
- (4) Calculate objective space density $\rho_o(\text{gen})$
- (5) Calculate the subspace density $\{\rho_1(\text{gen}), \rho_2(\text{gen}), \dots, \rho_k(\text{gen})\}$
- (6) **If** $\rho_i(\text{gen}) < \rho_o(\text{gen})$
- (7) **If** $\rho_i(\text{gen}) < h1\rho_o(\text{gen})$
- (8) Delete a weight vector
- (9) **else**
- (10) Adjustment weight vectors by formulas (6) and (7)
- (11) **End If**
- (12) **else if** $\rho_i(\text{gen}) > h2 \times \rho_o(\text{gen})$
- (13) Adjustment weight vectors by formulas (8) and (9)
- (14) **else**
- (15) Add a weight vector
- (16) **End If**
- (17) **End If**
- (18) **If** $i = k \ \&\& \text{length}(W_unit) \neq N$
- (19) Adjustment weight vectors of whole weight space
- (20) **End If**

ALGORITHM 4: Weight_Adjustment.

density is too small. If the density of the subspace is less than the minimum space density $h1\rho_o$, then the subspace density is considered too small. In this case, the weight vector should be evacuated. At this point, using the two nearest neighbor weight vectors and adding their sum vectors to the set of weight vectors, the parent vectors are deleted. Otherwise, the weight vector should be fine-tuned to achieve uniformity across the objective plane. At this time, according to the density difference, the nearest two weight vectors in the subspace are adjusted according to (7) and (8). Among them, the vectors $W_unit(k)$ and $W_unit(l)$ are the closest weight vectors. Let mt be the minimum

distance. Let ρ_i be the density value. The vectors $W_unit(gwk)$ and $W_unit(gwl)$ are neighbor weights of the respective weights.

$$W_unit(k) = W_unit(k) + (\rho_i - mt) \times W_unit(gwk), \quad (7)$$

$$W_unit(l) = W_unit(l) + (\rho_i - mt) \times W_unit(gwl). \quad (8)$$

- (2) When the subspace density is greater than the objective space density, determine whether the subspace density is too large. If the density of the subspace is

FIGURE 1: Many subspaces by using the K-means clustering.

FIGURE 2: Weight adjustment.

TABLE 1: The population size (N) for different numbers of objectives.

Number of objectives M	Segment parameter p	Population size N
3	12	92
5	6	212
8	$p_1 = 3, p_2 = 2$	156
10	$p_1 = 2, p_2 = 2$	112
15	$p_1 = 2, p_2 = 1$	136

TABLE 2: MFE times for different numbers of objectives.

Test instance	$M = 3$	$M = 5$	$M = 8$	$M = 10$	$M = 15$
DTLZ1	36,800	127,200	117,000	112,000	204,000
DTLZ2	23,000	74,200	78,000	84,000	136,000
DTLZ3	92,000	212,000	156,000	168,000	272,000
DTLZ4	55,200	212,000	195,000	224,000	408,000
DTLZ5	55,200	212,000	187,200	168,000	272,000
DTLZ6	36,800	74,200	117,000	224,000	272,000
WFG1~4	36,800	127,200	195,000	224,000	408,000

greater than the maximum space density $h_1\rho_o$, then the subspace density is too large. In this case, the weight vector should be aggregated. At this point, take the two furthest neighboring weight vectors and add

TABLE 3: Parameter values used in NSGA-III and MOEA/D.

Parameters	NSGA-III	MOEA/D
Crossover probability p_c	1	1
Variation probability p_m	$1/V$	$1/V$
Cross-distribution index η_c	30	20
Variance distribution index η_m	20	20

their sum vectors to the set of weight vectors. Otherwise, at this time, according to the density difference, adjust the weight vectors according to (9) and (10). Among them, the vectors $W_{\text{unit}}(k)$ and $W_{\text{unit}}(l)$ are the furthest weight vectors. Note mx is the maximum distance and ρ_i is the density value.

$$W_{\text{unit}}(k) = W_{\text{unit}}(k) + \frac{(mx - \rho_i)}{2} \times W_{\text{unit}}(l), \quad (9)$$

$$W_{\text{unit}}(l) = W_{\text{unit}}(l) + \frac{(mx - \rho_i)}{2} \times W_{\text{unit}}(k). \quad (10)$$

It is worth emphasizing that the edge vector is immovable; otherwise, the search range of the algorithm will be affected. Half of the maximum number of iterations was selected as an enabling condition for the weight vector

TABLE 4: The GD average and standard deviation of NSGA-III-WA and other five algorithms on DTLZ1-6 testing problems.

Problem	M	NSGA-III-WA	NSGA-III	VAEA	RVEA	MOEA/D	MOEA/D-M2M
DTLZ1	3	2.003e-06 (1.017e-05)	9.210e-04 + (2.322e-04)	1.776e-04 + (8.957e-05)	1.091e-03 + (2.154e-03)	1.799e-04 + (1.533e-04)	6.851e-03 + (8.316e-03)
	5	7.400e-08 (3.413e-08)	1.543e-04 + (2.730e-04)	1.077e-04 + (1.656e-04)	4.839e-04 + (3.785e-05)	4.745e-05 + (3.232e-05)	3.885e-03 + (3.274e-03)
	8	7.121e-06 (4.504e-06)	2.045e-03 + (1.882e-03)	1.722e-03 + (2.746e-03)	1.225e-04 + (5.246e-05)	1.524e-04 + (9.587e-05)	7.884e-03 + (6.720e-03)
	10	3.065e-06 (4.042e-06)	4.618e-03 + (3.393e-03)	2.259e-03 + (9.641e-04)	2.637e-03 + (1.233e-04)	2.187e-04 + (2.477e-04)	2.199e-02 + (6.495e-03)
	15	1.639e-07 (2.239e-07)	7.531e-02 + (8.504e-03)	3.695e-03 + (1.214e-03)	1.920e-04 + (1.682e-04)	1.386e-04 + (9.843e-05)	4.632e-02 + (1.382e-03)
DTLZ2	3	1.715e-06 (6.988e-07)	1.269e-04 + (1.679e-05)	2.657e-04 + (6.309e-05)	4.557e-04 + (7.907e-05)	3.918e-03 + (1.464e-03)	2.926e-04 + (3.825e-05)
	5	6.131e-05 (1.146e-05)	2.524e-04 + (2.555e-05)	4.729e-04 + (5.229e-05)	3.822e-04 + (4.425e-05)	9.614e-02 + (1.702e-03)	2.736e-03 + (3.057e-03)
	8	2.972e-04 (3.302e-05)	6.529e-04 + (8.548e-05)	6.463e-04 + (1.351e-04)	5.385e-04 + (9.497e-05)	3.303e-03 + (5.187e-05)	1.778e-03 + (3.743e-04)
	10	6.169e-04 (1.346e-04)	1.139e-03 + (2.172e-04)	7.390e-04 + (2.642e-04)	9.629e-04 + (1.761e-04)	6.532e-03 + (8.470e-04)	2.556e-03 + (4.978e-04)
	15	2.192e-04 (1.070e-04)	5.657e-04 + (9.716e-05)	7.725e-04 + (1.826e-04)	4.202e-04 + (2.084e-04)	9.719e-02 + (2.696e-03)	1.201e-02 + (3.749e-04)
DTLZ3	3	2.224e-10 (2.411e-06)	5.619e-04 + (1.986e-04)	1.827e-04 + (5.711e-04)	3.788e-03 + (1.353e-03)	3.971e-03 + (2.979e-05)	1.157e-03 + (6.428e-04)
	5	2.942e-06 (3.373e-06)	8.990e-04 + (4.012e-04)	2.232e-03 + (1.227e-03)	1.262e-03 + (2.190e-04)	7.459e-03 + (1.364e-04)	7.998e-03 + (3.265e-04)
	8	1.153e-04 (3.104e-05)	4.541e-03 + (3.786e-04)	3.961e-03 + (9.325e-04)	1.901e-03 + (7.506e-04)	3.127e-02 + (9.883e-04)	3.481e-02 + (2.819e-03)
	10	2.368e-04 (9.563e-05)	5.926e-03 + (2.259e-03)	4.713e-03 + (2.284e-03)	9.694e-03 + (2.543e-03)	5.910e-02 + (2.794e-03)	3.009e-02 + (2.295e-03)
	15	2.508e-04 (2.785e-04)	4.179e-03 + (2.756e-03)	7.510e-03 + (4.256e-04)	8.655e-04 + (1.074e-03)	9.834e-02 + (3.003e-03)	5.392e-02 + (2.949e-03)
DTLZ4	3	4.345e-09 (4.389e-09)	4.611e-04 + (2.188e-04)	2.790e-04 + (1.033e-04)	2.852e-04 + (1.749e-04)	4.547e-03 + (1.739e-03)	1.061e-04 + (3.147e-05)
	5	7.963e-15 (3.060e-14)	1.905e-04 + (5.277e-05)	5.008e-04 + (1.074e-04)	2.794e-04 + (4.251e-05)	3.596e-04 + (1.854e-05)	5.092e-04 + (2.427e-04)
	8	3.605e-12 (1.607e-11)	4.234e-04 + (7.418e-05)	7.629e-04 + (1.203e-04)	6.762e-04 + (1.645e-04)	7.233e-03 + (2.509e-04)	3.038e-03 + (3.791e-04)
	10	3.272e-16 (9.275e-16)	5.561e-04 + (9.294e-05)	8.086e-04 + (2.006e-04)	1.084e-03 + (2.514e-04)	1.060e-03 + (2.939e-05)	2.981e-03 + (5.370e-04)
	15	4.437e-15 (2.404e-14)	3.722e-04 + (8.070e-05)	7.542e-04 + (1.518e-04)	2.257e-04 + (1.222e-04)	1.319e-03 + (4.983e-05)	3.659e-03 + (2.773e-04)
DTLZ5	3	1.400e-01 (7.900e-03)	1.344e-01 + (9.360e-03)	1.962e-01 + (1.695e-02)	2.065e-01 + (1.245e-02)	5.394e-01 + (1.339e-03)	5.097e-02 - (2.371e-03)
	5	7.633e-02 (4.638e-03)	1.659e-01 + (1.039e-03)	1.559e-01 + (8.369e-04)	3.419e-01 + (5.779e-03)	4.709e-02 = (3.682e-03)	4.543e-02 - (5.821e-04)
	8	7.580e-02 (6.965e-03)	1.885e-01 + (4.210e-03)	2.219e-01 + (1.306e-03)	5.373e-01 + (2.058e-02)	8.841e-02 + (1.342e-03)	8.630e-02 = (3.746e-03)
	10	7.731e-02 (6.006e-03)	2.259e-01 + (4.640e-03)	2.725e-01 + (8.818e-04)	3.896e-01 + (3.558e-03)	1.241e-01 + (1.379e-03)	9.947e-02 + (1.385e-03)
	15	7.714e-02 (7.846e-03)	2.352e-01 + (2.731e-03)	2.503e-01 + (7.531e-04)	1.196e-01 + (4.752e-02)	1.468e-01 + (8.034e-04)	9.665e-02 + (4.543e-03)
DTLZ6	3	9.107e-02 (1.067e-02)	1.605e-01 + (1.225e-02)	1.647e-01 + (1.689e-02)	2.299e-01 + (4.912e-03)	3.661e-01 + (3.201e-03)	2.010e-01 + (3.569e-03)
	5	1.714e-02 (2.046e-03)	1.279e-02 - (2.679e-03)	3.713e-02 + (2.098e-03)	3.492e-02 + (2.748e-03)	1.346e-01 + (1.267e-04)	6.235e-02 + (8.714e-04)
	8	1.482e-02 (3.358e-03)	4.119e-03 - (4.968e-04)	7.671e-02 + (6.004e-03)	1.448e-02 = (8.183e-03)	1.034e-01 + (6.880e-03)	2.821e-02 + (3.728e-03)
	10	1.868e-02 (4.510e-03)	1.331e-03 - (3.501e-04)	7.597e-03 - (2.829e-04)	1.891e-02 = (8.185e-03)	5.831e-02 + (2.794e-03)	1.495e-02 - (3.746e-03)
	15	5.735e-03 (4.757e-04)	3.154e-03 - (1.438e-04)	4.743e-03 - (1.742e-04)	1.701e-02 + (9.023e-03)	1.361e-02 + (1.011e-04)	3.109e-02 + (7.158e-02)
# +/=-	—	—	26/0/4	28/0/2	28/2/0	29/1/0	26/1/3

TABLE 5: The IGD average and standard deviation of NSGA-III-WA and other five algorithms on DTLZ1-6 testing problems.

Problem	M	NSGA-III-WA	NSGA-III	VAEA	RVEA	MOEA/D	MOEA/D-M2M
DTLZ1	3	3.148e-02 (6.732e-04)	2.096e-02 = (6.245e-04)	7.776e-02 + (8.086e-04)	6.202e-02 + (2.796e-03)	4.086e-02 + (7.159e-03)	4.315e-02 + (5.569e-03)
	5	4.781e-02 (1.445e-04)	6.547e-02 + (1.645e-04)	5.203e-02 + (2.858e-04)	4.840e-02 + (2.853e-04)	7.737e-02 + (3.165e-04)	1.086e-01 + (1.434e-03)
	8	8.196e-02 (3.003e-03)	9.294e-02 + (1.489e-03)	9.351e-01 + (4.178e-03)	7.720e-02 = (5.637e-03)	1.149e-01 + (9.790e-03)	1.489e-01 + (5.276e-03)
	10	9.134e-02 (6.039e-03)	1.309 e-01 + (4.313e-03)	1.119 e-01 + (1.114e-03)	1.142e-01 + (2.156e-03)	1.022e-01 + (1.435e-04)	2.464e-01 + (3.592e-03)
	15	9.923e-02 (1.501e-03)	1.324e-01 + (2.857e-03)	1.136e-01 + (4.151e-03)	1.188e-01 + (2.306e-03)	1.132e-01 + (4.539e-03)	1.382e-01 + (1.736e-03)
DTLZ2	3	5.474e-02 (2.074e-04)	5.452e-02 = (5.829e-04)	5.637e-02 + (4.368e-04)	5.490e-02 = (1.888e-04)	6.392e-02 + (7.698e-04)	9.412e-02 + (2.835 e-03)
	5	1.527e-01 (8.099e-04)	1.612e-01 + (3.528e-04)	1.553e-01 + (4.447e-03)	1.519e-01 = (8.868e-04)	3.486e-01 + (1.318e-03)	2.095e-01 + (5.578e-03)
	8	2.598e-01 (3.285e-03)	2.675e-01 + (7.748e-04)	2.979e-01 + (5.934e-03)	2.617e-01 = (4.335e-03)	3.500e-01 + (2.287e-03)	4.494e-01 + (5.147e-03)
	10	3.286e-01 (1.116e-03)	3.570e-01 + (9.137e-03)	3.574e-01 + (1.895e-03)	4.600e-01 + (1.013e-03)	4.009e-01 + (6.283e-04)	4.603e-01 + (5.291e-03)
	15	3.188e-01 (3.626e-04)	3.580e-01 + (8.035e-04)	4.547e-01 + (2.230e-03)	3.592e-01 + (1.777e-03)	4.596e-01 + (7.071e-03)	4.583e-01 + (3.264e-03)
DTLZ3	3	5.893e-02 (7.098e-04)	9.937e-02 + (8.864e-04)	5.593e-02 = (1.972e-03)	6.608e-02 + (4.416e-03)	6.385e-02 + (1.490e-03)	9.495e-02 + (1.291e-03)
	5	1.671e-01 (3.529e-03)	1.598e-01 = (4.145e-03)	1.650e-01 + (9.129e-03)	1.583e-01 = (4.052e-03)	5.327e-01 + (1.052e-03)	5.158e-01 + (4.742e-03)
	8	2.857e-01 (2.753e-02)	4.185e-01 + (1.043e-01)	3.706e-01 + (5.212e-02)	3.117e-01 + (2.923e-02)	4.196e-01 + (4.810e-02)	4.032e-01 + (8.360e-02)
	10	3.252e-01 (1.778e-02)	4.751e-01 + (1.321e-02)	6.767e-01 + (4.540e-02)	3.835e-01 + (1.495e-02)	4.401e-01 + (4.446e-03)	7.313e-01 + (5.746e-02)
	15	3.225e-01 (5.028e-03)	5.076e-01 + (4.409e-02)	5.469e-01 + (2.917e-02)	3.636e-01 + (7.496e-03)	4.414e-01 + (2.664e-02)	5.743e-01 + (5.692e-02)
DTLZ4	3	2.998e-03 (1.402e-04)	3.685e-03 + (7.272e-04)	5.537e-02 + (1.937e-01)	3.359e-03 + (2.443e-04)	6.434e-02 + (1.009e-01)	7.938e-02 + (3.162e-02)
	5	9.586e-03 (6.178e-04)	1.173e-02 + (2.162e-03)	1.704e-01 + (1.054e-03)	1.039e-02 + (3.622e-04)	4.485e-02 + (2.562e-03)	1.419e-02 + (2.946e-03)
	8	2.820e-02 (1.099e-03)	3.257e-02 + (2.645e-03)	4.432e-01 + (3.122e-03)	3.082e-02 + (5.237e-04)	2.741e-01 + (2.447e-03)	4.622e-01 + (4.797e-03)
	10	5.562e-02 (3.895e-03)	5.043e-02 = (1.118e-03)	7.208e-01 + (3.446e-03)	5.556e-02 = (4.444e-03)	1.411e-01 + (8.020e-03)	9.292e-02 + (5.714e-03)
	15	9.265e-02 (2.780e-02)	8.040e-02 = (2.634e-03)	1.183e-01 + (5.087e-03)	1.002e-01 + (1.266e-02)	1.303e-01 + (9.904e-03)	1.033e-01 + (3.758e-03)
DTLZ5	3	1.281e-01 (1.585e-02)	1.143e-01 = (8.659e-03)	1.674e-01 + (5.705e-02)	2.057e-01 + (3.254e-03)	4.196e-01 + (2.332e-03)	4.329e-02 = (8.859e-03)
	5	3.854e-01 (6.196e-02)	1.137e+00 + (1.129e-01)	5.398e-01 + (1.447e-01)	5.198e-01 + (8.029e-02)	6.048e-01 + (1.373e-03)	4.785e-01 + (4.553e-02)
	8	3.702e-01 (5.821e-02)	6.228e-01 + (1.042e-02)	7.637e-01 + (3.741e-02)	4.112e-01 + (4.772e-02)	4.647e-01 + (5.300e-02)	4.697e-01 + (3.117e-02)
	10	3.853e-01 (6.602e-02)	7.052e-01 + (5.501e-02)	6.375e-01 + (6.288e-02)	3.821e-01 = (1.585e-02)	7.844e-01 + (2.872e-02)	5.840e-01 + (3.769e-02)
	15	5.905e-01 (6.090e-02)	8.251e-01 + (2.694e-02)	9.463e-01 + (1.180e-02)	6.022e-01 = (4.108e-02)	8.558e-01 + (2.129e-02)	6.296e-01 + (5.632e-02)
DTLZ6	3	9.766e-01 (2.520e-02)	1.516e+00 + (9.127e-02)	1.656e+00 + (5.092e-02)	1.303e+00 + (2.028e-02)	1.515e+00 + (7.586e-03)	1.826e+00 + (3.657e-03)
	5	5.673e-01 (1.851e-02)	6.385e-01 + (3.418e-02)	6.251e-01 + (9.602e-03)	7.416e-01 + (6.407e-03)	7.880e-01 + (9.136e-03)	9.257e-01 + (5.256e-03)
	8	5.141e-01 (4.012e-02)	5.233e-01 + (3.393e-02)	5.460e-01 + (1.476e-02)	5.383e-01 + (3.976e-02)	7.703e-01 + (8.253e-03)	7.437e-01 + (5.732e-03)
	10	4.425e-01 (2.195e-02)	3.994e-01 = (2.948e-02)	5.030e-01 + (2.205e-02)	6.184e-01 + (1.084e-02)	7.130e-01 + (4.135e-03)	6.718e-01 + (5.169e-02)
	15	3.147e-01 (2.727e-02)	3.558e-01 + (1.079e-02)	3.566e-01 + (2.154e-02)	5.502e-01 + (8.639e-02)	4.214e-01 + (3.918e-02)	6.972e-01 + (4.715e-02)
#+/-/-		—	23/2/5	29/0/1	22/5/3	30/0/0	29/0/1

TABLE 6: The HV average and standard deviation of NSGA-III-WA and other five algorithms on DTLZ1-6 testing problems.

Problem	M	NSGA-III-WA	NSGA-III	VAEA	RVEA	MOEA/D	MOEA/D-M2M
DTLZ1	3	9.727e-01 (1.679e-03)	9.661e-01 = (3.208e-03)	6.745e-01 + (8.305e-03)	9.379e-01 + (9.716e-03)	6.232e-01 + (1.434e-03)	9.595e-01 + (5.082e-03)
	5	9.987e-01 (4.645e-04)	9.941e-01 + (5.491e-03)	9.936e-01 + (8.779e-04)	9.990e-01 - (3.939e-04)	8.516e-01 + (5.741e-03)	8.409e-01 + (1.297e-03)
	8	9.986e-01 (1.009e-03)	9.910e-01 + (8.001e-03)	8.763e-01 + (3.096e-03)	9.724e-01 + (3.224e-03)	8.396e-01 + (4.159e-03)	8.682e-01 + (5.529e-03)
	10	9.983e-01 (5.898e-04)	9.858e-01 + (1.008e-03)	9.082e-01 + (1.463e-03)	9.972e-01 = (3.197e-03)	8.974e-01 + (3.229e-03)	8.754e-01 + (2.519e-03)
	15	9.998e-01 (1.572e-04)	9.980e-01 + (7.871e-04)	9.960e-01 + (1.004e-03)	9.995e-01 + (5.718e-04)	8.830e-01 + (8.141e-03)	7.307e-01 + (2.841e-03)
DTLZ2	3	9.244e-01 (2.256e-03)	9.250e-01 = (2.214e-03)	9.231e-01 + (1.889e-03)	9.251e-01 - (3.107e-03)	7.737e-01 + (1.329e-03)	8.968e-01 + (2.764e-03)
	5	9.909e-01 (1.504e-03)	9.890e-01 + (5.152e-04)	9.899e-01 + (9.500e-04)	9.379e-01 + (7.638e-03)	7.323e-01 + (4.672e-03)	9.760e-01 + (2.302e-03)
	8	9.992e-01 (1.119e-04)	9.984e-01 + (2.264e-04)	9.885e-01 + (1.367e-03)	9.985e-01 + (2.970e-03)	7.386e-01 + (4.697e-03)	8.922e-01 + (6.281e-03)
	10	9.989e-01 (4.487e-04)	9.942e-01 + (4.741e-03)	9.969e-01 + (2.531e-03)	9.973e-01 + (3.720e-03)	7.020e-01 + (5.068e-03)	8.815e-01 + (5.413e-03)
	15	9.999e-01 (2.973e-04)	1.000e+00 = (2.199e-03)	9.998e-01 + (1.731e-03)	9.999e-01 + (1.017e-03)	8.788e-01 + (7.351e-03)	9.082e-01 + (4.352e-03)
DTLZ3	3	9.261e-01 (2.199e-03)	9.202e-01 = (1.894e-03)	9.235e-01 + (1.788e-03)	9.197e-01 + (4.206e-03)	8.482e-01 + (1.881e-03)	9.062e-01 + (7.041e-03)
	5	9.899e-01 (1.063e-03)	9.892e-01 = (9.066e-04)	9.865e-01 = (1.690e-03)	8.921e-01 + (2.124e-03)	6.265e-01 + (7.156e-03)	4.653e-01 + (2.740e-03)
	8	9.831e-01 (1.132e-03)	9.984e-01 - (2.184e-04)	8.619e-01 + (5.486e-03)	9.981e-01 - (7.039e-04)	7.947e-01 + (4.331e-03)	5.237e-01 + (2.762e-03)
	10	9.975e-01 (4.712e-03)	9.789e-01 + (5.968e-03)	8.869e-01 + (1.820e-02)	9.916e-01 + (3.154e-03)	7.193e-01 + (5.231e-02)	3.010e-01 + (7.925e-02)
	15	1.000e+00 (5.351e-04)	9.998e-01 + (1.778e-03)	9.737e-01 + (4.021e-03)	9.999e-01 + (3.956e-04)	8.390e-01 + (2.721e-03)	3.417e-01 + (8.294e-03)
DTLZ4	3	9.252e-01 (2.831e-03)	8.762e-01 = (6.284e-03)	8.950e-01 + (1.073e-03)	9.261e-01 = (2.322e-03)	7.646e-01 + (1.483e-03)	9.097e-01 = (6.935e-03)
	5	9.867e-01 (1.377e-03)	9.887e-01 - (8.484e-04)	9.853e-01 = (9.799e-04)	9.831e-01 + (1.641e-03)	6.276e-01 + (2.529e-03)	9.861e-01 = (1.544e-03)
	8	9.987e-01 (5.391e-04)	9.987e-01 = (2.606e-04)	9.957e-01 + (4.359e-03)	9.989e-01 - (3.131e-04)	7.757e-01 + (5.391e-04)	9.943e-01 + (1.414e-03)
	10	9.998e-01 (1.117e-04)	9.998e-01 = (9.643e-04)	9.996e-01 + (5.604e-04)	9.995e-01 + (5.008e-04)	7.350e-01 + (5.951e-04)	9.964e-01 + (2.945e-03)
	15	1.000e+00 (9.863e-04)	9.999e-01 + (2.858e-04)	9.995e-01 + (1.863e-03)	9.998e-01 + (8.315e-04)	8.456e-01 + (4.485e-03)	9.927e-01 + (4.372e-03)
DTLZ5	3	8.370e-01 (1.361e-03)	8.128e-01 + (5.771e-03)	8.049e-01 + (2.157e-03)	8.689e-01 - (1.159e-03)	8.094e-01 + (1.693e-03)	7.330e-01 - (2.916e-02)
	5	8.622e-01 (3.332e-02)	3.775e-01 + (1.310e-02)	6.676e-01 + (7.426e-02)	8.057e-01 + (2.957e-02)	4.525e-01 + (9.054e-03)	8.099e-01 + (3.771e-03)
	8	8.099e-01 (4.163e-02)	6.056e-01 + (3.357e-02)	5.373e-01 + (4.556e-02)	7.143e-01 + (3.091e-02)	7.205e-01 + (6.349e-02)	6.365e-01 + (5.982e-02)
	10	7.885e-01 (3.761e-02)	7.052e-01 + (5.501e-02)	6.375e-01 + (6.294e-02)	6.399e-01 + (9.123e-02)	6.859e-01 + (5.109e-02)	3.786e-01 + (2.764e-02)
	15	6.892e-01 (7.425e-02)	5.436e-01 + (9.687e-02)	7.714e-01 - (9.537e-02)	5.691e-01 + (9.341e-02)	5.864e-01 + (4.891e-02)	5.144e-01 + (2.467e-02)
DTLZ6	3	1.079e+00 (1.335e-03)	1.043e+00 + (4.541e-03)	1.056e+00 + (4.296e-03)	9.258e-01 + (9.737e-03)	9.493e-01 + (8.997e-03)	1.041e+00 = (3.657e-03)
	5	1.429e+00 (1.647e-02)	1.384e+00 + (5.443e-02)	1.166e+00 + (2.272e-02)	1.402e+00 + (3.644e-02)	1.248e+00 + (2.181e-03)	1.275e+00 + (4.681e-03)
	8	1.468e+00 (5.101e-02)	1.416e+00 + (2.891e-02)	1.123e+00 + (8.664e-02)	1.176e+00 + (9.943e-02)	1.095e+00 + (2.064e-02)	8.201e-01 + (7.193e-02)
	10	1.123e+00 (9.184e-02)	1.127e+00 = (5.759e-02)	9.897e-01 + (2.878e-02)	9.816e-01 + (1.076e-02)	8.064e-01 + (4.445e-02)	9.322e-01 + (4.926e-02)
	15	9.319e-01 (2.287e-02)	9.107e-01 + (7.325e-03)	9.461e-01 = (4.023e-02)	7.651e-01 + (9.930e-02)	7.378e-01 + (3.547e-02)	7.194e-01 + (3.271e-02)
#+/-/-		—	19/9/2	26/3/1	23/2/5	30/0/0	26/3/1

TABLE 7: Summary of statistical test results in Table 4.

NSGA-III-WA	Objective number	vs. NSGA-III	vs. VAEA	vs. RVEA	vs. MOEA/D	vs. MOEA/D-M2M
GD	3	+: 6, =: 0, -: 0	+: 6, =: 0, -: 0	+: 6, =: 0, -: 0	+: 6, =: 0, -: 0	+: 5, =: 0, -: 1
	5	+: 5, =: 0, -: 1	+: 6, =: 0, -: 0	+: 6, =: 0, -: 0	+: 5, =: 1, -: 0	+: 5, =: 0, -: 1
	8	+: 5, =: 0, -: 1	+: 6, =: 0, -: 0	+: 5, =: 1, -: 0	+: 6, =: 0, -: 0	+: 5, =: 1, -: 0
	10	+: 5, =: 0, -: 1	+: 5, =: 0, -: 1	+: 5, =: 1, -: 0	+: 6, =: 0, -: 0	+: 5, =: 0, -: 1
	15	+: 5, =: 0, -: 1	+: 5, =: 0, -: 1	+: 6, =: 0, -: 0	+: 6, =: 0, -: 0	+: 6, =: 0, -: 0

Note: "+", "=", and "-" represent wins, equal to, and lose.

TABLE 8: Summary of statistical test results in Table 5.

NSGA-III-WA	Objective number	vs. NSGA-III	vs. VAEA	vs. RVEA	vs. MOEA/D	vs. MOEA/D-M2M
IGD	3	+: 3, =: 2, -: 1	+: 5, =: 0, -: 1	+: 5, =: 1, -: 0	+: 6, =: 0, -: 0	+: 5, =: 0, -: 1
	5	+: 5, =: 0, -: 1	+: 6, =: 0, -: 0	+: 4, =: 1, -: 1	+: 6, =: 0, -: 0	+: 6, =: 0, -: 0
	8	+: 6, =: 0, -: 0	+: 6, =: 0, -: 0	+: 4, =: 1, -: 1	+: 6, =: 0, -: 0	+: 6, =: 0, -: 0
	10	+: 4, =: 0, -: 2	+: 6, =: 0, -: 0	+: 4, =: 1, -: 1	+: 6, =: 0, -: 0	+: 6, =: 0, -: 0
	15	+: 5, =: 0, -: 1	+: 6, =: 0, -: 0	+: 5, =: 1, -: 0	+: 6, =: 0, -: 0	+: 6, =: 0, -: 0

Note: "+", "=", and "-" represent wins, equal to, and lose.

TABLE 9: Summary of statistical test results in Table 6.

NSGA-III-WA	Objective number	vs. NSGA-III	vs. VAEA	vs. RVEA	vs. MOEA/D	vs. MOEA/D-M2M
HV	3	+: 2, =: 4, -: 0	+: 6, =: 0, -: 0	+: 3, =: 1, -: 2	+: 6, =: 0, -: 0	+: 3, =: 2, -: 1
	5	+: 4, =: 1, -: 1	+: 4, =: 2, -: 0	+: 5, =: 1, -: 0	+: 6, =: 0, -: 0	+: 5, =: 1, -: 0
	8	+: 5, =: 1, -: 1	+: 6, =: 0, -: 0	+: 4, =: 0, -: 2	+: 6, =: 0, -: 0	+: 6, =: 0, -: 0
	10	+: 4, =: 2, -: 0	+: 6, =: 0, -: 0	+: 5, =: 0, -: 1	+: 6, =: 0, -: 0	+: 6, =: 0, -: 0
	15	+: 5, =: 1, -: 0	+: 4, =: 1, -: 1	+: 6, =: 0, -: 0	+: 6, =: 0, -: 0	+: 6, =: 0, -: 0

Note: "+", "=", and "-" represent wins, equal to, and lose.

adjustment strategy and adjust every four generations in this paper. In this time, the objective vectors have approached the PFs, so the guidance of the weight vectors is relatively accurate, and the population update is relatively stable (i.e., it is close to the PF).

4. Discussion

The previous section described the NSGA-III-WA algorithm in detail. In this section, we compare the similarities and differences between NSGA-III-WA, NSGA-III, and VAEA.

4.1. The Similarities and Differences between NSGA-III-WA and NSGA-III

- Both algorithms use Pareto dominance to select individuals.
- The evolution of the two algorithms is different. NSGA-III adopts the original genetic evolution strategy, while NSGA-III-WA adopts a new differential evolution strategy to optimize individuals. It has better effect on convergence speed and solution accuracy.
- Both algorithms have different strategies for dividing the objective space. The NSGA-III algorithm uses the original method of generating weight vectors to evenly divide the objective space.

The NSGA-III-WA algorithm divides the objective space into several subspaces and adjusts the weight vectors according to the individual density of the objective space. This method can better ensure the uniformity of the weight vectors on the objective surface, thus ensuring the uniformity of the solution set.

4.2. The Similarities and Differences between NSGA-III-WA and VAEA

- Both algorithms use Pareto dominance to select individuals.
- Both algorithms need to normalize the population. The difference is that VAEA normalizes the population according to the ideal and lowest point of the population, while NSGA-III-WA obtains the intercept of each objective axis by calculating the ASF and then normalizes the population. The latter is more universal and more reasonable.
- Both algorithms have associated operations. VAEA does not relate to the association of reference points. It achieves the association between individuals and individuals and thus cannot guarantee individual distribution. NSGA-III-WA associates individuals with weight vectors to improve the distribution of the algorithm.

FIGURE 3: Boxplots of GD, IGD, and HV index by the four algorithms with 5 objectives on DTLZ1 problem.

FIGURE 4: Boxplots of GD, IGD, and HV index by the four algorithms with 5 objectives on DTLZ2 problem.

FIGURE 5: Boxplots of GD, IGD, and HV index by the four algorithms with 5 objectives on DTLZ3 problem.

5. Simulation Results

In order to verify the performance of the proposed algorithm on MAOPs, this paper selects the general test function set DTLZ and WFG in the field of many-objective optimization for simulation experiments. The proposed algorithm is

compared with five reliable algorithms, MOEA/D, NSGA-III, VAEA, RAEA, and MOEA/D-M2M, and representative algorithms with the objectives 3, 5, 8, 10, and 15 on the DTLZ1-6 test function and WFG1-4 test instances. Performance indicators GD [29], IGD [30], and HV [31, 32] are used for comparative analysis: first, to briefly introduce the

FIGURE 6: Boxplots of GD, IGD, and HV index by the four algorithms with 5 objectives on DTLZ4 problem.

FIGURE 7: Boxplots of GD, IGD, and HV index by the four algorithms with 5 objectives on DTLZ5 problem.

FIGURE 8: Boxplots of GD, IGD, and HV index by the four algorithms with 5 objectives on DTLZ6 problem.

corresponding parameter settings for each algorithm, and second, to explain the experimental results, compare, and analyze them.

5.1. General Experimental Settings. The number of decision variables for all test functions is $V = M + k - 1$, and M is the number of objective functions, $k = 5$ for DTLZ1, and $k = 10$ for DTLZ2-6. The number of decision variables for all WFG test functions is $V = k + 1$, where the position

variable is $k = M - 1$ and the objective dimension is M ; the distance variable is $l = 10$. The population sizes of NSGA-III and RVEA are related to the uniformly distributed weight scale and determined by the combination number of M and the number of p on each objective. The double-layer distribution method in [13] is adopted in order to tackle the problem. The specific parameter settings are given in Table 1. For fair comparison, the population size is the same as the other three algorithms.

FIGURE 9: Boxplots of GD, IGD, and HV index by the four algorithms with 15 objectives on DTLZ1 problem.

FIGURE 10: Boxplots of GD, IGD, and HV index by the four algorithms with 15 objectives on DTLZ2 problem.

FIGURE 11: Boxplots of GD, IGD, and HV index by the four algorithms with 15 objectives on DTLZ3 problem.

The algorithm runs independently for 30 times on each test function. The algorithm uses the maximum function evaluation (MFE) as the terminal condition for each run.

Due to objective dimensions of the solution problem, MFE is also not the same. According to [16], the specific settings are shown in Table 2. The maximum number of iterations is calculated by $\text{gen_max} = \text{MFE}/N$. The parameter settings of NSGA-III and MOEA/D are shown in Table 3. In addition, the number of MOEA/D neighbor weight T is 10; the RVEA penalty factor change rate α is 2, and the VAEA angle threshold is expressed as $\delta = (\pi/2)/(N + 1)$.

5.2. Results and Analysis. In order to verify the performance saliency of the proposed NSGA-III-WA algorithm on many objective optimization problems, general performance evaluation indicators GD, IGD and HV were used. It is compared with five good algorithms, MOEA/D, NSGA-III, VAEA, RAEA, and MOEA/D-M2M, and representative algorithms with the objectives 3, 5, 8, 10, and 15 on the DTLZ1-6 test function and WFG1-4 test instances.

5.2.1. Testing and Analysis of DTLZ Series Functions.

FIGURE 12: Boxplots of GD, IGD, and HV index by the four algorithms with 15 objectives on DTLZ4 problem.

FIGURE 13: Boxplots of GD, IGD, and HV index by the four algorithms with 15 objectives on DTLZ5 problem.

FIGURE 14: Boxplots of GD, IGD, and HV index by the four algorithms with 15 objectives on DTLZ6 problem.

This section shows the results and analysis of the GD, IGD, and HV performance test data of the DTLZ1-6 test function. The experimental results are shown in Tables 4–6. They are the average values and standard deviations of 30 independent running results. The best results are shown in black and bold, and the values in parentheses indicate the standard deviation; the number in square brackets is the algorithm performance ranking, which is based on the Whitney–Wilcoxon rank-sum test [33]. To investigate whether NSGA-III-WA is statistically superior to other algorithms, Wilcoxon’s rank-sum test is performed at a 0.05 significance level between NSGA-III-WA and each

competing algorithm on each test case. The test results are given at the end of each cell, represented by the symbols “+,” “=,” or “–,” which indicate that the NSGA-III performance is better than the algorithm in the corresponding column, equal to, and worse. At the same time, the last row of Tables 4–6 summarizes the number of test instances that NSGA-III-WA is significantly better than, equal to, and below its competitors. Tables 7–9 show the results of the comparison of NSGA-III-WA algorithm with the other five algorithms under different objective numbers.

It can be seen from the experimental results in Table 4 that the GD values of NSGA-III-WA on the DTLZ1-4 are

FIGURE 15: Parallel graph of the final solution set of each algorithm on the 15-objective DTLZ2 test problem. (a) NSGA-III-WA. (b) NSGA-III. (c) MOEA/D. (d) RVEA. (e) VAEA. (h) MOEA/D-M2M.

superior to the other five algorithms, only 8th, 10th, and 15th dimensions are superior on the DTLZ5, and NSGA-III on the DTLZ6 gets the best results. It shows that in solving

many-objective problems, the convergence of NSGA-III-WA is more effective than that of NSGA-III algorithm and is better than other algorithms. Table 7 shows summary

TABLE 10: The IGD average and standard deviation of NSGA-III-WA and other five algorithms on WFG1-4 testing problems.

Problem	M	NSGA-III-WA	NSGA-III	VAEA	RVEA	MOEA/D	MOEA/D-M2M
WFG1	3	1.171e+00 (2.727e-01)	1.370e+00 + (3.356e-01)	1.324e+00 + (2.315e-01)	1.047e+00 - (2.417e-01)	1.216e+00 + (2.173e-01)	1.211e+00 + (3.725e-01)
	5	2.828e+00 (1.057e-01)	2.927e+00 + (3.726e-01)	3.203e+00 + (2.941e-01)	3.171e+00 + (3.173e-01)	3.701e+00 + (2.053e-01)	1.953e+00 - (3.278e-01)
	8	5.721e+00 (1.714e-01)	5.230e+00 - (1.572e-01)	5.139e+00 - (1.437e-01)	5.520e+00 - (1.635e-01)	6.623e+00 + (1.052e-01)	5.769e+00 = (1.678e-01)
	10	7.146e+00 (3.182e-02)	7.071e+00 - (5.709e-02)	7.238e+00 + (4.083e-02)	7.182e+00 + (4.281e-02)	9.541e+00 + (3.219e-01)	7.816e+00 + (9.023e-02)
	15	8.942e+00 (1.284e-01)	9.079e+00 + (1.673e-01)	9.057e+00 + (3.073e-01)	9.149e+00 + (3.726e-01)	1.183e+01 + (2.961e-01)	1.235e+00 + (3.618e-01)
WFG2	3	2.149e-01 (6.137e-02)	2.839e-01 + (1.040e-01)	3.218e-01 + (8.931e-02)	3.157e-01 + (4.366e-02)	1.317e+00 + (7.013e-02)	3.714e-01 + (4.827e-02)
	5	5.237e-01 (9.814e-02)	6.125e-01 + (1.375e-01)	9.052e-01 + (2.781e-01)	7.026e-01 + (1.437e-01)	3.971e+00 + (2.739e-01)	1.411e+00 + (3.894e-01)
	8	2.316e+00 (1.835e-01)	3.146e+00 + (1.379e-01)	2.007e+00 - (2.371e-01)	2.572e+00 + (1.638e-01)	8.837e+00 + (5.462e-01)	2.885e+00 + (4.732e-01)
	10	2.037e+00 (2.171e-01)	2.923e+00 + (4.518e-01)	3.592e+00 + (4.178e-01)	2.964e+00 + (3.926e-01)	1.027e+01 + (1.001e+00)	2.1416e+00 = (9.287e-01)
	15	5.187e+00 (1.373e-01)	6.223e+00 + (2.381e-01)	5.250e+00 = (5.936e-01)	4.945e+00 = (3.826e-01)	1.346e+01 + (2.964e-01)	4.014e+00 - (5.382e-01)
WFG3	3	2.163e-01 (2.647e-02)	3.791e-01 + (8.167e-02)	1.489e-01 - (6.916e-03)	1.977e-01 - (3.283e-02)	1.793e-01 - (3.034e-02)	2.361e-01 + (4.624e-02)
	5	4.746e-01 (6.437e-03)	5.274e-01 + (7.136e-03)	4.793e-01 + (6.374e-03)	4.827e-01 + (5.378e-03)	5.418e-01 + (2.542e-02)	7.633e-01 + (1.091e-01)
	8	1.308e+00 (2.748e-02)	1.709e+00 + (1.061e-01)	1.427e+00 + (2.135e-02)	1.604e+00 + (3.375e-02)	1.829e+00 + (4.873e-02)	2.487e+00 + (3.823e-02)
	10	1.864e+00 (2.073e-02)	2.176e+00 + (2.874e-01)	1.725e+00 - (3.271e-02)	1.845e+00 = (4.273e-02)	2.966e+00 + (7.627e-02)	3.369e+00 + (6.379e-02)
	15	2.815e+00 (2.733e-01)	4.206e+00 + (1.537e-01)	2.963e+00 + (2.736e-01)	3.028e+00 + (1.893e-01)	5.265e+00 + (8.733e-02)	6.738e+00 + (1.284e-01)
WFG4	3	2.043e-01 (2.274e-03)	2.147e-01 + (3.859e-04)	2.317e-01 + (7.352e-03)	2.272e-01 + (3.72e-03)	2.475e-01 + (3.758e-03)	3.581e-01 + (3.146e-03)
	5	9.635e-01 (3.762e-03)	9.865e-01 + (4.873e-03)	9.535e-01 + (5.378e-03)	9.526e-01 - (3.288e-03)	1.284e+00 + (3.725e-02)	1.676e+00 + (1.003e-02)
	8	3.021e+00 (4.887e-03)	3.262e+00 + (6.256e-03)	3.023e+00 = (1.567e-02)	3.114e+00 + (6.331e-03)	6.642e+00 + (1.526e-02)	4.6209e+00 + (2.462e-02)
	10	4.063e+00 (1.879e-02)	4.621e+00 + (2.834e-02)	3.982e+00 = (3.274e-02)	3.870e+00 - (2.716e-01)	9.826e+00 + (1.873e-01)	6.698e+00 + (2.706e-01)
	15	8.926e+00 (2.768e-01)	9.732e+00 + (3.381e-02)	8.541e+00 - (2.834e-01)	8.737e+00 - (2.762e-01)	1.496e+01 + (9.276e-03)	1.103e+01 + (1.241e-01)
# +/=-		—	18/0/2	12/3/5	12/2/6	19/0/1	16/2/2

TABLE 11: Summary of statistical test results in Table 10.

NSGA-III-WA	Objective number	vs. NSGA-III	vs. VAEA	vs. RVEA	vs. MOEA/D	vs. MOEA/D-M2M
IGD	3	+: 4, =: 0, -: 0	+: 3, =: 0, -: 1	+: 2, =: 0, -: 2	+: 3, =: 0, -: 1	+: 4, =: 0, -: 0
	5	+: 4, =: 0, -: 0	+: 4, =: 0, -: 0	+: 3, =: 0, -: 1	+: 4, =: 0, -: 0	+: 3, =: 0, -: 1
	8	+: 3, =: 0, -: 1	+: 1, =: 1, -: 2	+: 3, =: 0, -: 1	+: 4, =: 0, -: 0	+: 3, =: 1, -: 0
	10	+: 3, =: 0, -: 1	+: 2, =: 1, -: 1	+: 2, =: 1, -: 1	+: 4, =: 0, -: 0	+: 3, =: 1, -: 0
	15	+: 4, =: 0, -: 0	+: 2, =: 1, -: 1	+: 2, =: 1, -: 1	+: 4, =: 0, -: 0	+: 3, =: 0, -: 1

Note: “+,” “=,” and “-” represent wins, equal to, and lose.

of statistical test results from Table 4. NSGA-III-WA is compared to five other more advanced multiobjective algorithms and counts the number of wins (+), equal to (=), and number of loses (-). As can be seen from the table, NSGA-III-WA is clearly superior to the five most advanced designs selected.

From Table 5, the NSGA-III-WA can get the best results especially the objectives 5, 10, and 15 on the DTLZ1, 8, 10, and 15 on the DTLZ2 and DTLZ3, 3, 5, and 8 on the DTLZ4, and 5, 8, and 15 on the DTLZ5. Moreover, the objectives 3 and 8 on the DTLZ1, the objectives 3 and 5 on the DTLZ2, the objective 3 on the DTLZ3, and the objective 10 on the DTLZ5 achieve the second best results. Nevertheless, on the DTLZ5 and DTLZ6, the results of NSGA-III-WA are not significant because the DTLZ5 and DTLZ6 are used to test the ability to converge to a curve. Owing to the reason that NSGA-III-WA needs to build a hypersurface, M extreme points cannot be found in the later stage of the algorithm to construct the hypersurface, and it cannot converge to a curve well. In addition, NSGA-III-WA has noticeable effects on other test functions and is a kind of stable and relatively comprehensive algorithm. Table 8 shows summary of statistical test results from Table 5. It can be seen from the table that NSGA-III-WA performs best on the other five algorithms.

From Table 6, it can be seen that the NSGA-III-WA can effectively handle most test problems. It can get the best results especially the objectives 3, 8, 10, and 15 on the DTLZ1, 5 and 10 on the DTLZ2, 3, 5, 10, and 15 on the DTLZ3, 10 and 15 on the DTLZ4, 5, 8, and 10 on the DTLZ5, and 3, 5, and 8 on the DTLZ6. Moreover, the objective 5 on the DTLZ1, objective 15 on the DTLZ2, objectives 3 and 5 on the DTLZ4, objectives 3 and 15 on the DTLZ5, and objectives 10 and 15 on the DTLZ6 achieve the second best results. However, the performances of objective 3 on the DTLZ2 and 8 on the DTLZ3 are poor. Although NSGA-III, VAEA, RVEA, and MOEA/D can obtain optimal values for a particular dimension in the function, NSGA-III-WA has the best overall performance considering all dimensional objective results. Table 9 shows summary of statistical test results from Table 6. As can be seen from the table, NSGA-III-WA hypervolume performance is better than the other five algorithms.

In order to express the effect of the algorithm more intuitively, the performance of the algorithm is presented in the form of a box diagram. Due to space limitations, only the analysis of the box diagrams of the four algorithms under five goals and fifteen goals is given here. Figures 3–8 show the performance box diagram under the four goals, and

Figures 9–14 show the performance box diagram under the fifteen goals. Each box diagram is calculated by inputting 30 independent running results. It reflects the median, maximum, minimum, upper quartile, lower quartile, and outliers of the five algorithms on indicators GD, IGD, and HV.

From Figures 3–8, it can be seen that NSGA-III-WA can achieve better results when dealing with most test problems. Its convergence and breadth are significantly better than the other five algorithms. The overall performance indicators achieve the best results on the DTLZ1 and DTLZ4 and get the second best results on the DTLZ2 and DTLZ3. Although the minimum value is obtained on the DTLZ5 and DTLZ6, there exist abnormal values, indicating that the algorithm is relatively unstable. This is because DTLZ5 and DTLZ6 test the ability of the algorithm to converge to a straight line, while NSGA-III-WA needs to build a hypersurface, so it cannot converge to a curve well. However, the overall robustness of NSGA-III-WA is relatively better with all test function results.

From Figures 9–14, it can be seen that the NSGA-III-WA has the ability to handle most problems under the 15 objectives. The convergence on the DTLZ1-DTLZ5 is significantly better than the other five algorithms. The overall performance obtains the best results on the DTLZ1-DTLZ3, DTLZ5, and DTLZ6. The NSGA-III-WA under the 15 objectives can get the minimum on DTLZ5 and DTLZ6 but is relatively unstable. The breadth achieves the best results on the DTLZ1, DTLZ3, and DTLZ4 and gets the second best results on the DTLZ2, DTLZ5, and DTLZ6, and there exist abnormal values. On the 15 objectives, it is evident that the outliers of each algorithm increase. That is explained by the fact that the stability of algorithms in the high-dimensional space will decline due to the increase of the spatial breadth. Depending on the results of all test functions, NSGA-III-WA has better stability.

In order to visually reflect the distribution of the solution set in the high-dimensional target space, parallel coordinates are used to visualize the high-dimensional data as shown in Figure 15.

From Figure 15, it can be seen that NSGA-III-WA and RVEA find the final solution set in this problem to be similar in convergence and distribution. In contrast, MOEA/D-M2M and NSGA-III are slightly less distributed than the above three algorithms. VAEA finds that the distribution of the solution is poor. MOEA/D appears the concentrated solution. Lose extreme solutions at 12 objectives and the distribution of MOEA/D is seriously missing.

5.2.2. Testing and Analysis of WFG Series Functions. The performance indicators of the WFG test function are mainly

TABLE 12: The HV average and standard deviation of NSGA-III-WA and other five algorithms on WFG1-4 testing problems.

Problem	M	NSGA-III-WA	NSGA-III	VAEA	RVEA	MOEA/D	MOEA/D-M2M
WFG1	3	5.114e-01 (2.425e-02)	5.013e-01 + (2.279e-02)	5.217e-01 - (2.725e-02)	4.963e-01 + (2.673e-02)	4.927e-01 + (9.245e-03)	4.824e-01 + (5.245e-02)
	5	4.725e-01 (2.469e-03)	4.632e-01 + (5.352e-03)	5.172e-01 - (4.281e-03)	4.824e-01 + (3.729e-03)	5.793e-01 - (4.736e-03)	4.875e-01 + (4.237e-03)
	8	4.481e-01 (2.724e-03)	4.116e-01 + (3.861e-03)	4.480e-01 + (3.783e-03)	4.376e-01 + (2.751e-02)	4.472e-01 + (2.747e-02)	4.324e-01 + (3.783e-02)
	10	6.063e-01 (3.217e-02)	5.937e-01 + (6.273e-02)	5.997e-01 + (4.238e-02)	6.218e-01 - (3.628e-02)	4.926e-01 + (1.026e-01)	5.382e-01 + (7.375e-02)
	15	6.279e-01 (2.781e-02)	6.163e-01 + (3.273e-02)	6.181e-01 + (2.672e-02)	6.197e-01 + (2.418e-02)	3.472e-01 + (2.163e-01)	4.781e-01 + (1.784e-01)
WFG2	3	8.373e-01 (4.263e-02)	8.524e-01 - (3.279e-02)	8.393e-01 + (3.789e-02)	8.334e-01 + (2.274e-02)	7.251e-01 + (3.194e-02)	8.425e-01 - (2.785e-02)
	5	9.602e-01 (3.793e-02)	9.547e-01 + (3.926e-02)	9.482e-01 + (4.821e-02)	9.376e-01 + (4.245e-02)	9.172e-01 + (8.278e-02)	9.318e-01 + (6.351e-02)
	8	9.223e-01 (2.891e-02)	9.502e-01 - (3.268e-02)	9.172e-01 + (2.753e-02)	9.514e-01 - (2.724e-02)	8.702e-01 + (6.932e-02)	8.945e-01 + (4.837e-02)
	10	9.492e-01 (2.893e-02)	9.471e-01 + (2.062e-02)	9.261e-01 + (3.625e-02)	9.372e-01 + (1.341e-01)	8.981e-01 + (1.826e-01)	9.148e-01 + (4.782e-01)
	15	9.715e-01 (2.715e-02)	9.678e-01 + (2.361e-02)	9.483e-01 + (2.936e-02)	9.572e-01 + (3.798e-02)	7.815e-01 + (3.781e-01)	8.147e-01 + (3.461e-01)
WFG3	3	8.068e-01 (3.278e-03)	8.136e-01 - (2.267e-03)	7.978e-01 + (4.624e-03)	5.749e-01 + (2.411e-02)	7.371e-01 + (4.267e-02)	5.361e-01 + (3.283e-02)
	5	8.723e-01 (2.798e-03)	8.825e-01 - (3.142e-03)	8.734e-01 + (3.682e-03)	5.921e-01 + (5.274e-02)	7.702e-01 + (9.257e-02)	5.032e-01 + (3.863e-02)
	8	9.267e-01 (3.278e-03)	9.241e-01 + (5.261e-03)	9.257e-01 + (4.258e-02)	7.026e-01 + (2.794e-02)	7.315e-01 + (8.903e-02)	8.461e-01 + (6.352e-02)
	10	9.349e-01 (1.392e-03)	9.352e-01 + (3.267e-03)	9.392e-01 - (2.674e-03)	5.252e-01 + (2.493e-02)	4.315e-01 + (1.367e-01)	7.947e-01 + (6.375e-01)
	15	9.318e-01 (2.717e-03)	9.264e-01 + (3.257e-03)	9.381e-01 - (2.916e-03)	6.735e-01 + (4.784e-02)	7.106e-01 + (8.628e-02)	5.375e-01 + (7.783e-02)
WFG4	3	6.997e-01 (3.278e-03)	6.805e-01 + (2.581e-03)	6.885e-01 + (3.528e-03)	7.293e-01 - (3.271e-03)	6.697e-01 + (3.275e-02)	6.019e-01 + (4.251e-02)
	5	8.674e-01 (2.678e-03)	8.640e-01 + (4.216e-03)	8.601e-01 + (3.728e-03)	8.756e-01 - (3.782e-02)	8.602e-01 + (5.272e-03)	8.327e-01 + (6.429e-03)
	8	9.147e-01 (2.791e-03)	9.020e-01 + (3.736e-03)	9.103e-01 + (3.827e-03)	9.128e-01 + (3.782e-03)	7.502e-01 + (2.861e-02)	8.462e-01 + (4.571e-02)
	10	8.573e-01 (3.728e-03)	8.517e-01 + (4.286e-03)	8.237e-01 + (3.183e-03)	8.603e-01 - (4.247e-03)	7.136e-01 + (2.271e-02)	8.354e-01 + (1.180e-02)
	15	9.114e-01 (1.273e-03)	9.077e-01 + (7.263e-03)	9.105e-01 + (3.726e-03)	8.982e-01 + (4.251e-03)	4.525e-01 + (1.597e-01)	7.239e-01 + (2.743e-01)
# +/=-	—	—	13/3/4	9/7/4	11/5/4	17/2/1	17/2/1

TABLE 13: Summary of statistical test results in Table 12.

NSGA-III-WA	Objective number	vs. NSGA-III	vs. VAEA	vs. RVEA	vs. MOEA/D	vs. MOEA/D-M2M
HV	3	+: 2, =: 0, -: 2	+: 1, =: 2, -: 1	+: 2, =: 1, -: 1	+: 4, =: 0, -: 0	+: 3, =: 0, -: 1
	5	+: 2, =: 1, -: 1	+: 2, =: 1, -: 1	+: 2, =: 1, -: 1	+: 2, =: 1, -: 1	+: 3, =: 1, -: 0
	8	+: 3, =: 0, -: 1	+: 2, =: 2, -: 0	+: 1, =: 2, -: 1	+: 3, =: 1, -: 0	+: 3, =: 1, -: 0
	10	+: 2, =: 2, -: 0	+: 2, =: 1, -: 1	+: 2, =: 1, -: 1	+: 4, =: 0, -: 0	+: 4, =: 0, -: 0
	15	+: 4, =: 0, -: 0	+: 2, =: 1, -: 1	+: 4, =: 0, -: 0	+: 4, =: 0, -: 0	+: 4, =: 0, -: 0

Note: “+,” “=,” and “-” represent wins, equal to, and lose.

IGD and HV indicators. Therefore, this section tests the WFG1-4 test instance and analyzes the results, as shown in Tables 10–13.

From the results in Table 10, it can be seen that NSGA-III-WA can handle most of the considered examples well. In particular, it achieved the best overall performance on the objectives 3, 5, and 10 on WFG2 instances and the objectives 5, 8, and 15 on WFG3 instances. In addition, it achieves the best performance on the objective 15 on WFG8 and the objectives 3 and 8 on WFG9. The VAEA performed well on the objective 8 on WFG1 and WFG2 test instances and also achieved good results on the objectives 3 and 10 on WFG3 and the objective 4 on WFG3. RVEA obtains the best IGD value on the objective 3 on WFG1 and the objectives 5 and 10 on WFG4. It is worth noting that RVEA performs poorly for WFG2 and WFG3 instances. But it performs relatively well compared to the NSGA3 and MOEA/D algorithms. NSGA-III and MOEA/D-M2M typically have moderate performance on most WFG problems, and good results can only be achieved on specific WFG test instances. MOEA/D does not produce satisfactory results in all WFG test instances. As the number of objectives increases, the results gradually deteriorate. Table 11 shows summary of statistical test results from Table 10. It can be seen from the table that the performance of NSGA-III-WA is significantly better than that of the other five algorithms.

From the results in Table 12, it can be seen that NSGA-III-WA has obtained the best performance for most of the high-dimensional objective problems. NSGA-III works well on the WFG1 and WFG2 test instances, and VAEA also gets good results on the objectives 10 and 15 on WFG3 test instances. RVEA obtains the best HV value on the objectives 3, 5, and 10 on WFG4. MOEA/D and MOEA/D-M2M are not quite effective on these five instances. Table 13 shows summary of statistical test results from Table 12. The three-dimensional performance of the NSGA-III-WA algorithm is not very prominent. The performance under the eight-dimensional algorithm is the same as that of the RVEA algorithm, but the NSGA-III-WA algorithm can achieve better performance in high-dimensional objective problems. In general, the NSGA-III-WA algorithm outperforms the other five algorithms in this performance.

In summary, after comparing the test results of GD, IGD, and HV performance, the performance of the NSGA-III-WA algorithm is superior.

6. Conclusion

This paper proposes a many-objective optimization algorithm based on weight vector adjustment, which increases

the individual’s ability to evolve through new differential evolution strategies, and at the same time, dynamically adjust the weight vector by means of the K -means to make the weight vector as evenly distributed as possible on the objective surface. The NSGA-III-WA algorithm has good convergence ability and good distribution. To prove its effectiveness, the NSGA-III-WA is experimentally compared with the other five most advanced algorithms on the DTLZ test set and WFG test instances. The experimental results show that the proposed NSGA-III-WA performs well on the DTLZ test set and WFG test instances we studied, and the obtained solution set has good convergence and distribution. However, the proposed algorithm has high complexity and it only plays the role of alleviating sensitive frontiers. Further research will be conducted on the above 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 there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant nos. 61501107 and 61501106, the Education Department of Jilin Province Science and Technology Research Project of “13th Five-Year” under Grant no. 95, and the Project of Scientific and Technological Innovation Development of Jilin under Grant nos. 201750219 and 201750227.

References

- [1] H. Son and C. Kim, “Evolutionary many-objective optimization for retrofit planning in public buildings: a comparative study,” *Journal of Cleaner Production*, vol. 190, pp. 403–410, 2018.
- [2] M. Pal, S. Saha, and S. Bandyopadhyay, “DECOR: differential evolution using clustering based objective reduction for many-objective optimization,” *Information Sciences*, vol. 423, pp. 200–218, 2017.
- [3] Q. Zhang, W. Zhu, B. Liao, X. Chen, and L. Cai, “A modified PBI approach for multi-objective optimization with complex Pareto fronts,” *Swarm & Evolutionary Computation*, vol. 40, pp. 216–237, 2018.

- [4] A. Pan, H. Tian, L. Wang, and Q. Wu, "A decomposition-based unified evolutionary algorithm for many-objective problems using particle swarm optimization," *Mathematical Problems in Engineering*, vol. 2016, Article ID 6761545, 15 pages, 2016.
- [5] W. Gong, Z. Cai, and L. Zhu, "An efficient multiobjective differential evolution algorithm for engineering design," *Structural & Multidisciplinary Optimization*, vol. 38, no. 2, pp. 137–157, 2012.
- [6] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization," *Evolutionary Computation*, vol. 10, no. 3, pp. 263–282, 2005.
- [7] A. G. Hernández-Díaz, L. V. Santana-Quintero, C. A. Coello Coello, and J. Molina, "Pareto-adaptive ϵ -dominance," *Evolutionary Computation*, vol. 15, no. 4, pp. 493–517, 2014.
- [8] Y. Yuan, H. Xu, and B. Wang, "An improved NSGA-III procedure for evolutionary many-objective optimization," in *Proceedings of Conference on Genetic & Evolutionary Computation*, pp. 661–668, ACM, Nanchang, China, October 2014.
- [9] Y. Jingfeng, L. Meilian, X. Zhijie, and X. Jin, "A simple pareto adaptive ϵ -domination differential evolution algorithm for multi-objective optimization," *Open Automation & Control Systems Journal*, vol. 7, no. 1, pp. 338–345, 2015.
- [10] X. Yue, Z. Guo, Y. Yin, and X. Liu, "Many-objective E-dominance dynamical evolutionary algorithm based on adaptive grid," *Soft Computing*, vol. 22, no. 1, pp. 137–146, 2016.
- [11] M. M. Lin, H. Zhou, and L. P. Wang, "Research of many-objective evolutionary algorithm based on alpha dominance," *Computer Science*, vol. 44, no. 1, pp. 264–270, 2017.
- [12] C. Dai and Y. Wang, "A new multiobjective evolutionary algorithm based on decomposition of the objective space for multiobjective optimization," *Journal of Applied Mathematics*, vol. 2014, Article ID 906147, 9 pages, 2014.
- [13] Q. 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.
- [14] Y. Yuan, H. Xu, B. Wang et al., "A new dominance relation-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 16–37, 2016.
- [15] C. Segura and G. Miranda, "A multi-objective decomposition-based evolutionary algorithm with enhanced variable space diversity control," in *Proceedings of Genetic and Evolutionary Computation Conference Companion*, pp. 1565–1571, ACM, Berlin, Germany, July 2017.
- [16] Y. Xiang, Y. Zhou, M. Li, and Z. Chen, "A vector angle-based evolutionary algorithm for unconstrained many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 1, pp. 131–152, 2017.
- [17] X. Guo, Y. Wang, and X. Wang, "Using objective clustering for solving many-objective optimization problems," *Mathematical Problems in Engineering*, vol. 2013, Article ID 584909, 12 pages, 2013.
- [18] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired co-evolutionary algorithm using adaptively generated goal vectors," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 916–923, Cancun, QROO, Mexico, June 2013.
- [19] 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.
- [20] 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.
- [21] H. L. Liu, F. Gu, and Q. Zhang, "Decomposition of a multi-objective optimization problem into a number of simple multiobjective subproblems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 450–455, 2014.
- [22] X. Bi and C. Wang, "An improved NSGA-III algorithm based on objective space decomposition for many-objective optimization," *Soft Computing*, vol. 21, no. 15, p. 4269, 2016.
- [23] M. Zhang and H. Li, "A reference direction and entropy based evolutionary algorithm for many-objective optimization," *Applied Soft Computing*, vol. 70, pp. 108–130, 2018.
- [24] 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.
- [25] Y. Li, Y. Kou, and Z. Li, "An improved nondominated sorting genetic algorithm III method for solving multiobjective weapon-target assignment Part I: the value of fighter combat," *International Journal of Aerospace Engineering*, vol. 2018, Article ID 8302324, 23 pages, 2018.
- [26] Y. Sun, G. G. Yen, and Z. Yi, "IGD indicator-based evolutionary algorithm for many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 99–114, 2018.
- [27] J. Xiao, J. J. Li, X. X. Hong et al., "An improved MOEA/D based on reference distance for software project portfolio optimization," *Mathematical Problems in Engineering*, vol. 2018, Article ID 3051854, 16 pages, 2018.
- [28] J. Shi, M. Gong, W. Ma, and L. Jiao, "A multipopulation coevolutionary strategy for multiobjective immune algorithm," *The Scientific World Journal*, vol. 2014, Article ID 539128, 23 pages, 2014.
- [29] K. Deb and S. Jain, "Running performance metrics for evolutionary multi-objective optimization," Technical Report No. 2002004, Indian Institute of Technology Kanpur, Kanpur, India, 2002.
- [30] S. Chand and M. Wagner, "Evolutionary many-objective optimization: a quick-start guide," *Surveys in Operations Research & Management Science*, vol. 20, no. 2, pp. 35–42, 2015.
- [31] H. R. Maier, Z. Kapelan, J. Kasprzyk et al., "Evolutionary algorithms and other metaheuristics in water resources: current status, research challenges and future directions," *Environmental Modelling & Software*, vol. 62, pp. 271–299, 2014.
- [32] A. Díaz-Manríquez, G. Toscano, J. H. Barron-Zambrano, and E. Tello-Leal, "R2-based multi/many-objective particle swarm optimization," *Computational Intelligence and Neuroscience*, vol. 2016, Article ID 1898527, 10 pages, 2016.
- [33] K. Li, S. Kwong, and K. Deb, "A dual-population paradigm for evolutionary multiobjective optimization," *Information Sciences*, vol. 309, pp. 50–72, 2015.