A New Many-Objective Evolutionary Algorithm for Adjusting Reference Vectors Based On Vector Angles and Convergence Metrics

WenJie Wang

***Abstract*— In decomposition-based MOEAs, the inconsistency of the distribution of reference vectors with the shape of the PF is a long-standing problem. To address this challenge, strategies to adjust the reference vectors during the evolutionary process have been proposed. However, most of the methods either adjust the reference vectors fixedly in each generation or at a fixed frequency, ignoring the dynamic information during population evolution. To solve the above problem, this work proposes a strategy to judge the timing of reference vector adjustment based on the change of population convergence. By calculating the improvement rate of subproblem convergence, it can reflect the current convergence state of the population. The algorithm performs reference vector adjustment only when the subproblems are considered to be converged in general. To make the reference vectors better adapt to different shapes of the Pareto fronts, this work draws on the concept of vector angle and proposes a method to adjust the reference vector based on the maximum vector angle. Specifically, by maintaining an archive with well-distributed nondominated solutions, the individual among them that possesses the largest vector angle with the current population is selected for the adjustment of the reference vector. Experiments on WFG test problems show that the proposed algorithm is competitive compared to the state-of-the-art algorithms for solving problems with different Pareto fronts.**

***Index Terms*—convergence state, decomposition-based MOEAs, irregular Pareto fronts, reference vectors adjustment.**

I. INTRODUCTION

With the development of science and the renewal of industrial technology, multi-objective optimization problems (MOPs) have become more and more the focus of attention, and are widely used in the fields of scientific research and engineering. When problems have more than four objectives, they are called many-objective optimization problems (MaOPs). A MOP or MaOP can be formulated as follows:

$$\begin{aligned} &min\ F(x) = \left[f_1(x), f_2(x), \ldots, f_m(x) \right] \\ &s.t. x \in \Omega \end{aligned} \quad (1)$$

where $x = (x_1, \ldots, x_n)^T$ is an n-dimensional decision vector in the feasible region Ω of the decision space. $f_i(x)$ represents the i-th objective function in objective space, m is the number of objectives in the optimization problem.

In the MaOPs presented above, multiple objectives often conflict with each other. Optimizing one objective may result in deterioration of other objectives. Therefore, there does not exist a solution that can achieve optimal results on all objectives. For the study of MaOPs, it is significance to explore effective methods to coordinate these conflicting objectives 0and help decision makers obtain the optimal compromise.

With the increasing research on optimization problems, many-objective optimization problems (MaOEAs) have been proposed to solve MaOPs. These algorithms are broadly categorized into three types: (1) Pareto-dominance-based; (2) Indicator-based; (3) Decomposition-based. Based on the Pareto-dominance principle, MOaEAs typically categorize solutions into distinct non-dominated floors, addressing MaOPs by preserving a set of non-dominated solutions. Nevertheless, when confronted with a substantial number of objectives, the traditional Pareto dominance relationship becomes insufficient for selecting non-dominated solutions. Consequently, researchers have introduced various enhanced dominance relationships, including grid dominance [1] and L-dominance [2], to overcome this limitation. Indicator-based MaOEAs often devise specific indicators to comprehensively assess performance of individuals accordingly. Such as the epsilon indicator-based approach [3] and the hypervolume indicator-based method [4] prioritize both convergence and diversity. Nevertheless, as the number of objectives increases, the computational overhead of calculating these performance indicators also escalates. Decomposition-based MaOEAs dividing the MaOPs into numerous subproblems through the weights [5], reference points [6], or reference vectors [7]. These algorithms solve these subproblems cooperatively, offering a more efficient means of tackling MaOPs.

Decomposition-based MaOEAs have achieved excellent performance in solving MaOPs by dividing a MaOP into several subproblems using a set of predefined uniformly distributed reference vectors. In decomposition-based algorithms, if the reference vectors are uniformly distributed, then the optimal solutions will also have good diversity. Based on this assumption, many researchers have proposed a variety of algorithms aiming to ensure the divisibility of the reference vectors on the real Pareto fronts (PFs). With the increasing research on such algorithms, it is recognized that this assumption presupposes that the PF shapes are simplex-like. However, when the shape of the real PF is irregular, such as discontinuous or degenerate, uniformly distributed reference vectors in the objective space are not guaranteed to be uniformly distributed on the real PF as well.

In order to solve the problem of the distributivity of the reference vectors, a straight approach is to adjust the reference vectors during the evolutionary process. There are

Manuscript received May 05, 2024
Wenjie Wang, School of Computer Science and Technology, Tiangong University, Tianjin, China

several important issues that need to be addressed in reference vector adjustment-based algorithms. Although some of these issues have been addressed to some extent. However, there is still room for further research in such algorithms, in terms of when to trigger the adjustment operation and how to adjust the reference vectors to ensure their effectiveness in the face of MaOPs. However, most of the existing reference vector adjustment mechanisms only focus on the process of reference vector adjustment, in which the correlation between the frequency of adjustment and the evolutionary state of the population has received little attention. As long as a preset frequency is reached, the reference vector is adjusted, even though the individuals of the population still have a large convergence space. Adjusting the reference vectors in this case may destroy the diversity of the population, leading to longer convergence time and poorer convergence of the algorithm. In addition, the number of valid reference vectors becomes small in a high-dimensional objective space. Therefore, the reference vector adjustment method independent of the number of objectives has better performance in the solution of MaOPs.

To address the issues mentioned above, a new algorithm to improve the RVEA called CDBEA is proposed in this work. The algorithm designs a reference vector adjustment timing strategy based on convergence metrics in order to periodically and conditionally trigger the reference vector adjustment. In addition, the reference vector adjustment process selects a vector angle adjustment-based method to replace invalid reference vectors in MaOPs. The main contributions of CDBEA can be summarized as follows:

1. The reference vectors are adjusted only when the subproblems are generally considered to be convergent. The current convergence of the population by calculating the improvement rate of the subproblem convergence.

2. Based on the vector angle between the individuals in a well-maintained archive and the reference vectors, this paper proposes a method to adjust the reference vector based on the maximum vector angle. The individual in the archive that has the largest vector angle with the current population is chosen as the reference vector to fit the shape of the PF.

II. BACKGROUND

A. Related work

Since most of the real-world optimization problems have irregular PFs, the adjustment of the reference vectors is necessary in order to make the reference vectors uniformly distributed over the real PFs. In MOEA/D-AM2M [9], k representative individuals are randomly selected to divide the objective space into k subregions. In each subregion, a vector is randomly selected and a new weight vector is generated using the individual with the largest angle to the selected vector. In the enhanced decomposition-based evolutionary algorithm g-DEBA [10], invalid reference vectors that are not associated with any solution are removed at each iteration. When adding new reference vectors, the closest reference vector to the deleted reference vector is first found. Next, the individual in the population that has the largest vertical distance from that vector is found. Finally, generate a new reference vector based on this solution. In MOEA/D with adaptive weight adjustment (MOEA/D-AWA) [11], the weight vectors of the congested regions are periodically deleted and new weight vectors are added based on the sparsest solution in the external archive. In the MOEA/D algorithm with uniform stochastic adaptive weights (MOEA/D-URAW) [12], MOEA/D-AWA is enhanced by using a uniform strategy to generate the initial weight vectors. In the Adaptive Evolutionary Algorithm for Reference Points Based on IGD-NS Metrics (AR-MOEA) [13], the reference vectors are updated based on the IGD-NS metrics using the individuals in the archive. Lin et al [14] set up the reference vectors based on the nondominated individuals in the external archive that have the maximum multiplicative distance from the kth nearest neighbor. In the Decomposition-based multi-objective Evolutionary Algorithm guided by Growing Neural Gas (DEA-GNG) [15], the structure of the PF is learned by a growing neural gas network. The learned topological neighbors are used to modify the scalarizing functions and the reference vectors. New Two-Stage Based Evolutionary Algorithm (MaOEA-IT) [16] improves the performance of the MOEA/D algorithm by introducing the concept of iteration and weight adjustment. It adjusts the weight vectors iteratively during the optimization process, which enhances the search capability and convergence speed of the algorithm.

B. RVEA

The reference vector-guided evolutionary algorithm RVEA can be seen as a decomposition-based algorithm, which aims to balance convergence and diversity of solutions. In RVEA, reference vectors are used to divide the objective space into several subspaces, each reference vector represents a subproblem, and each subproblem evolves independently under the guidance of the reference vectors. The process of reference vector-guided solution evolution in RVEA consists of the following four main steps:(1) objective value conversion; (2) population division; (3) calculation of angular penalization distance (APD) values; and (4) selection of elite solutions. The ADP function plays an important role in the process of environment selection and is used to evaluate the overall performance of the candidate solutions, i.e., the convergence and diversity of the solutions. The APD is calculated as shown in the following equation:

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

where $\left\|f'_{t,i}\right\|$ is an evaluation of the performance of the i-th solution in terms of convergence, which is expressed by the distance between the objective vector $f'_{t,i}$ of the j-th subproblem and the ideal point. The angle between the solution and the reference vector $v_{t,j}$ is used to represent the diversity of the solutions. $P\left(\theta_{t,i,j}\right)$ is the penalty function on the $\theta_{t,i,j}$;

$$P\left(\theta_{t,i,j}\right) = M \cdot \left(\frac{t}{t_{\max}}\right)^{\alpha} \cdot \frac{\theta_{t,i,j}}{\gamma_{\mathbf{v}_{t,j}}} \tag{3}$$

$$\gamma_{\mathbf{v}_{t,j}} = \min \left\langle \mathbf{v}_{t,i}, \mathbf{v}_{t,j} \right\rangle, \quad i \in \{1,\ldots,N\}, \quad i \neq j \tag{4}$$

where M is the number of objectives, N is the population size, t is the current iteration number, and $t_{\max}$ is a predefined maximum number of iterations. α is a user-defined parameter to control the rate of change of $P(\theta_{t,i,j})$, and a larger value of α indicates a greater focus on convergence. In the evolutionary process, when the initial iteration t = 0, $P(\theta_{t,i,j})$ = 0, which means that solutions with better convergence will be prioritized. As the number of iterations t increases, the algorithm then emphasizes more and more on the diversity of solutions.

III. Proposed Algorithm

In this section, details of the CDBEA will be given. The overall framework of CDBEA is first presented.

A. Overall Framework

The main loop of CDBEA is presented in Algorithm 1. The new component of CDBEA are Archive Update and the Reference Vectors Adjustment.

Algorithm 1: Main Processes of CDBEA	
Input: Population size N; number of objectives M; maximum number of iterations $t_{\max}$; external archive size NA.	
Output: The final population P;	
1	Initialize the population P and a set of uniformly distributed reference vectors $V = \{v_1, v_2, \ldots, v_N\}$;
2	Adding non-dominated solutions from population P to archive A;
3	Calculate the value of the convergence indicator CM for the subproblem; (Algorithm 2)
4	While $t < t_{\max}$ Do
5	Q= Offspring-Creation (P);
6	A = ArchiveUpdate (A, Q, NA);
7	$S = Q \cup P$
8	P= Environmental Selection (S, V);
9	If condition 1 is satisfied
10	Compute IR to reflect the relative improvement in convergence for each subproblem;
11	If condition 2 is satisfied
12	Adaptive adjustment of reference vectors; (Algorithm 3)
13	End If
14	End If
15	End While

Algorithm 1 shows the framework of CDBEA. First, the population P and the set of reference vectors V are initialized (line 1). Next, find the non-dominated solutions in the population P and place them in the archive A (line 2). The degree of convergence of the subproblem is evaluated by calculating the convergence indicator CM by Algorithm 3 (line 3). Lines 4-15 are the main loop of CDBEA. If the current number of iterations does not reach the preset maximum number of iterations, the child generation Q is generated based on the population P (line 5); the individuals in the archive A are updated by the Archive Update method (line 6). The next generation of populations is selected in the joint population S by the APD method, this is the same environment selection as in RVEA. Finally, we determine whether to perform reference vector adjustment by condition 1 and condition 2 and execute the adjustment method of reference vector.

B. Archive Update

In CDBEA, an archive with a set limit is employed during evolution to hold non-dominated solutions. When the archive reaches its capacity, overcrowded individuals are selectively eliminated to preserve a balanced distribution.

For this study, we utilize the population maintenance technique proposed in [17] as a means to guarantee archive diversity. The niche concept serves as a powerful tool for evaluating the crowding of individuals in a population. Here, we assess the crowding level of archive individuals based on the number and position of individuals in the niche. Then, we systematically remove the most crowded individuals until the archive reaches its predetermined capacity. Specifically, the crowding degree of individual p in archive A is defined as follows:

$$D(p) = 1 - \prod_{q \in A, q \neq p} R(p,q) \tag{5}$$

and

$$R(p,q) = \begin{cases} d(p,q)/r, & \text{if } d(p,q) \leq r \\ 1, & \text{otherwise} \end{cases} \tag{6}$$

where $d(p,q)$ denotes the Euclidean distance between solutions p and q in the archive A; The niche radius r is set to the median distance between all solutions and their k-th nearest neighbor, where k corresponds to the number of objectives. In addition, since the different number of objectives in the problem affects the evaluation of individual crowding, all objectives will be normalized using the max-min method.

Notably, the individual crowding degree $D(p)$ ranges from [0,1], where a lower $D(p)$ signifies less crowding. When $D(p)$ is zero, it implies that there are no other individuals in the niche of p. Duplicate individuals exhibit the largest $D(p)$.

C. Reference Vector Adjustment

1. Adjustment time

In the process of reference vector adjustment, the frequency of adjustment is an important issue. Adjusting the reference vectors with higher frequency may destroy the convergence of the solution. On the contrary, lower frequency adjustments will waste some computational effort on unnecessary subproblems whose reference vectors do not intersect with the true PF. If there is still room for improving convergence in the subproblem, then no adjustment of the reference vectors is needed. Ideally, the reference vector should be adjusted when the population as a whole converges. To solve the above problem, this paper sets two conditions, condition 1 and condition 2, and uses the improvement rate (IR) to control the reference vector adjustment. As shown in Algorithm 1, if condition 1 is satisfied, IR is calculated, and if condition 2 is also satisfied, then the reference vector is adjusted.

At the beginning of evolution, the population is in the process of exploration and it is not appropriate to perform operational adjustment. At the end of evolution, if the reference vectors still be adjusted, then the subproblem will not have enough time to converge. Thus, condition 1 can be expressed as follows: the current iteration t is in the range of 20% to 90% of $t_{\max}$. If condition 1 is true, IR is computed to reflect the relative improvement in convergence for each subproblem.

$$IR = \frac{CM - CM_{\text{old}}}{CM_{\text{old}}} \tag{7}$$

where CM is the value of the convergence metric for the current iteration, and CM_{old} is the value of the convergence metric for the previous computation. As shown in Algorithm 2, First, the population P is normalized, and then the angular distance between each reference vector in V and the solution in the normalized P is computed. The formula for angular distance is as follows.

$$\mathbf{d}(P1,V) = 1 - \cos\left(\mathbf{x}_i, v_j\right) = 1 - \frac{\mathbf{F}'(\mathbf{x}_i)v_j}{\left|\mathbf{F}'(\mathbf{x}_i)\right|_2 \cdot \left|v_j\right|_2} \tag{8}$$

where $F'(x_i)v_j$ is the inner product between $F'(x_i)$ and v_j. $|F'(x_i)|_2$ and $|v_j|_2$ represent the L2-norm $F'(x_i)$ of and v_j. Then, find the solution with smallest angular distance to each reference vector and compute the d1 distance of that solution on its associated reference vector.

Algorithm2: Calculating Convergence Indicator Values

Input：	population P, the set of reference vectors V
Output:	Convergence indicator value CM
1.	Normalized population P;
2.	Calculate the angular distance between the set of reference vectors V and the population P according to Eq. (8);
3.	Associate the reference vector v_i with its nearest solution x_i;
4.	CM = calculate the distance d1 between the projection of x_i on v_i and the ideal point

In this paper, each reference vector is associated with its nearest solution, instead of associating each solution with its nearest reference vector. If a reference vector has more than one associated solution, the solution with the smallest d1 distance is chosen as the final associated solution for that reference vector. Thus, in Fig.1(a), the reference vectors V_1, V_2, V_3, and V_5, V_6, V_8 all have solutions associated with them, which can be used to measure the degree of convergence of their subproblems. In contrast, the degree of convergence of subproblems V_4 and V_7 cannot be represented because there are no solutions associated with them. Although V_4 and V_7 are currently invalid reference vectors, there should be solutions capable of estimating the degree of convergence of their corresponding subproblems before adjusting them. With this in mind, as shown in Figure 4(b), this paper chooses to associate each reference vector with its closest solution. This ensures that each reference vector has an associated solution and that the degree of convergence of its subproblem can be computed.

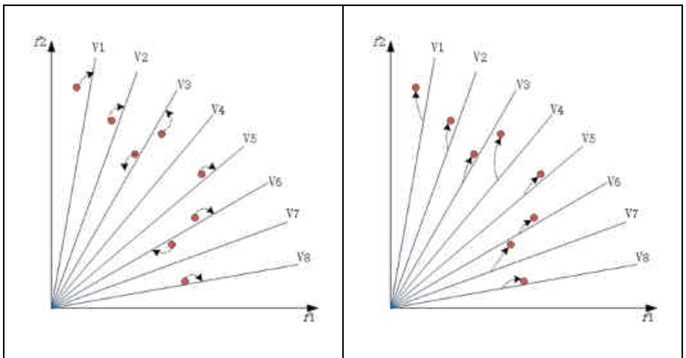

Figure 1 (a) Figure 1 (b)

Figure 1: The way the reference vector is associated with the solution

After calculating IR using Eq. (7), it is converted as follows.

$$IR = \begin{cases} -1, IR < -\alpha \\ 1, \quad IR > \alpha \\ 0, \quad otherwise \end{cases} \tag{9}$$

where α is a parameter that determines whether the subproblem has converged or not. IR <-α indicates that the subproblem still has room to converge. Otherwise, it indicates that the corresponding subproblem is in the process of diversity preservation, and there is not much room for convergent improvement under the current reference vector. Accordingly, the reference vector is adjusted if condition 2 is satisfied. This question defines condition 2 as: when the sum of IR of all subproblems is greater than or equal to zero, the evolution of all subproblems has converged on the whole.

2. Adding and removing reference vectors

If both condition 1 and condition 2 are satisfied, the reference vectors are adjusted using Algorithm3. During evolution, some reference vectors may have no solution associated with them, and these are inactive reference vectors. The main idea of reference vector adjustment in CDBEA is to delete invalid reference vectors, and then to generate new reference vectors based on the solution in the archive that has the largest vector angle to the current set of reference vectors V.

Deletion of invalid reference vectors: the individuals in the population are first normalized (lines 1-4). Associate the individuals in the population with the reference vectors closest to them (lines 5-8). For each reference vector, count the number of solutions associated with it; if no solution is associated with that reference vector, it means that this reference vector is invalid, and such a reference vector will be deleted (lines 9-13).

Adding new reference vectors: the newly generated reference vectors are added one by one to the set of reference vectors V until the number of reference vectors in V reaches a predefined number N. The addition of new references is based on the following considerations. First, when solving MaOPs, the nondominated solutions during evolution can reflect the evolutionary state of the current population. During the evolutionary process, the nondominated solutions will gradually approximate the PF and thus reflect the shape of the real PF. Therefore, these nondominated solutions serve as a reference for exploring the current unexplored areas. Therefore, an archive, in which the nondominated solutions during the evolutionary process are kept, should be used to reflect the current evolutionary direction. Therefore, the newly added reference vectors are referenced by maintaining a good archive. Another consideration is that the newly added reference vectors should maintain the diversity of the reference vector set, which in turn guides the diversity of the solutions. Therefore, new reference vectors are generated based on the solution in the archive that has the largest vector angle with the current reference vector.

The process of adding new reference vectors is shown in lines 14-21 of Algorithm3. First, the number of paradigms is computed for each reference vector in the normalized objective space. For v_j, the $norm(v_j)$ is defined as:

$$\mathrm{norm}(v_j) = \sqrt{\sum_{i=1}^{m} f_i'(v_j)^2} \tag{10}$$

Then the vector (acute angle) between the solution x_j in the archive and the reference vector v_k is defined as:

$$\mathrm{angle}(\mathbf{x}_j, v_k) = \arccos\left|\frac{\mathbf{F}'(\mathbf{x}_j)\bullet\mathbf{F}'(v_k)}{\mathrm{norm}(\mathbf{x}_j)\cdot\mathrm{norm}(v_k)}\right| \tag{11}$$

where $\mathbf{F}'(\mathbf{x}_j)\bullet\mathbf{F}'(v_k)$ is the inner product between $\mathbf{F}'(\mathbf{x}_j)$ and $\mathbf{F}'(v_k)$, it is defined as follows:

$$\mathbf{F}'(\mathbf{x}_j)\bullet\mathbf{F}'(v_k) = \sum_{i=1}^{m} f_i'(\mathbf{x}_j)\cdot f_i'(v_k). \tag{12}$$

After calculating the vector angles between the archive and the reference vectors, the smallest of these angles is selected and the individual in the archive corresponding to the smallest angle is identified, a new reference vector is generated based on this individual and added to the set of reference vectors, and the individual is deleted from the archive.

Algorithm 3: Reference Vector Adjustment.

Input: population P, reference vector set V, archive A, number of populations N.
Output: updated population P, updated reference vector V.
Normalization process
1 Calculate the ideal point Z^{min};
2 For $i = 1 \rightarrow |P|$ do
3 $f_i = f_i - Z^{\min}$;
4 End For
5 For $i = 1 \rightarrow |P|$ do
6 $k \leftarrow argmin_{j\in(1,2,\ldots,|V|)} \langle P_i, V_j \rangle$;
7 $S_k = S_k \cup P_i$;#Dividing the subproblems
8 End For
Remove inactive reference vectors
9 For $i = 1 \rightarrow |V|$ do
10 If $|S_i| > 0$ Then
11 $V = V \setminus V_i$;
12 End If
13 End For
Add new reference vectors
14 While $|V| < N$
15 computing norm ($A \cup V$) according to Eq. (10);
16 Calculate the vector angle between the solution in the archive and V according to Eq. (11);
17 Select the individual $A_{\max}$ in the archive with the largest vector angle to V
18 Generate a new reference vector $V_{\max}$ based on $A_{\max}$
19 $V = V \cup V_{\max}$;
20 $A = A \setminus A_{\max}$;
21 End While

IV. Experimental Simulation and Analysis

In this section, the performance of the proposed algorithm is examined on WFG1-WFG9 [18]. MOEAD [5], RVEA [7], DEAGNG [15], MaOEAIT [16] and hpaEA [19] are adopted to compare with the proposed algorithm.

A. *Performance Indicators*

In this paper, the performance of the algorithms is evaluated by two widely used metrics: hypervolume (HV) [20] and inverse generation distance (IGD) [21]. These two metrics serve as comprehensive evaluation metrics capable of measuring algorithm convergence and diversity.

(1) IGD evaluates the performance of a population by calculating the average Euclidean distance between a reference point on the true PF and a solution in the population. The smaller value of IGD indicates a better performance of the algorithm. The real PF is represented by a set of solutions located in it. In this experiment, the number of these solutions is set to 10000. the IGD is defined by the following formula:

$$\mathrm{IGD}(P^*, P) = \frac{\sum_{v\in P^*} D_{\min}(v, P)}{|P^*|} \tag{13}$$

where P^* is the set of uniformly distributed reference points on the real PF and P is the set of objective vectors of the population. The value of $D_{min}(v,P)$ is the minimum Euclidean distance from point v in P^* to all points in P.

(2) HV evaluates the convergence and distribution of the algorithm in terms of the reference points surrounded by the hypervolume. A larger value of HV indicates better performance of the algorithm. The reference point $Z = (Z_1, Z_2, \ldots, Z_M)$, are dominated by all Pareto optimal solutions. The HV of a set of solutions can be defined as the volume of the objective space dominated by the final solution set P and bounded by the reference point Z. It can be expressed as follows:

$$HV(P,Z) = \text{Volume}\left(\bigcup_{x\in P}[f_1(x), Z_1]\times\ldots[f_M(x), Z_M]\right) \tag{14}$$

where Volume(.) is the Lebesgue metric, which is used to measure the volume in space. In this experiment, the size of each dimension of the reference point exceeds the upper limit of each dimension of the real PF by 10%. In general, in order to reduce the computational complexity of HV on MaOPs, Monte Carlo methods are usually used to approximate the HV values with 1000000 sampling points when M>4.

B. *Experimental Settings*

In the design of algorithm parameters for the (MOEAs) under comparison, the parameter settings are the same as the recommendations provided in their original papers.

Specifically, for MOEA/D, the neighborhood size is set to 10% of the total population size, while the probability of selecting a parent solution from the neighborhood is set at 0.9. Additionally, the maximum number of solutions that can be replaced during each iteration is limited to 1% of the population size. For the RVEA, the rate of change for the penalty function is set to 2, and the frequency of adaptive adjustment for the reference vector is specified as 0.1. These parameter configurations ensure consistency with the original implementations and facilitate a fair comparison of the algorithms' performance.

Since the population size of the multi-objective evolutionary algorithm (MOEA/D) is significantly influenced by the number of objectives, a tailored approach is adopted for weight reduction problems. When the number of objectives exceeds 8, a two-layer reference vector generation strategy is employed. This strategy generates weight vectors not only along the outer boundary of the Pareto front, but also within an inner layer to capture a broader range of potential weight reductions. In the current experiment, the population sizes for the six algorithms on the 9 test problems are determined based on the number of objectives. Specifically, when the number of objectives is 3, 5, 8, and 10, the corresponding population sizes are set to 100, 212, 156, and 275, respectively.

In this experiment, the maximum number of evolutionary iterations for the 9 test functions is set to 100,000 on 3, 5, 8, and 10 objectives. Upon reaching the maximum iteration limit, the algorithms terminated their execution. Furthermore, for the 9 test problems each of the six algorithms is evaluated independently 30 times, and the mean and standard deviation of the metrics were documented. To assess the significant differences in the hypervolume (HV) and inverted generational distance (IGD) metrics against the five comparison algorithms of CDBEA, we employed the Wilcoxon rank sum test with a significance level of 0.05. Specifically, the symbol "+" denotes that the other multi-objective evolutionary algorithms (MOEAs) exhibited significantly superior results compared to CDBEA, whereas "-" signifies the converse. Lastly, the symbol "=" indicates that the algorithms performed similarly.

C. Results and Discussions

Tables 1 and 2 collect the mean and standard deviation of the HV performance metrics and IGD performance metrics for the six algorithms in WFG1-WFG9 on 3, 5, 8, and 10 objectives. In these tables, the best results for each test suite are marked with a gray background.

In Table 1, CDBEA shows better performance on the HV test metrics. It outperforms all the comparison algorithms on 17 out of 36 test instances, while RVEA, hpaEA, MOEAD, MaOEAIT, and DEAGNG, perform best on 1, 4, 3, 1, and 8 test instances, respectively. Based on the Wilcoxon rank sum test to test the performance difference with the five comparison algorithms of CDBEA on HV metrics it can be concluded that CDBEA outperforms the other five comparison algorithms on 21, 17, 29, 32, and 19 test suites, respectively. CDBEA outperforms the other five comparison algorithms on 5, 8, 10 objectives of WFG6, WFG7, WFG8, and WFG9, from which it can be inferred that CDBEA has a competitive advantage in dealing with MaOPs, which may be attributed to the fact that the reference vector adjustment method based on the vector angles can well guide the evolution of the populations thus obtaining solutions that can balance the convergence and diversity.

The results of the comparison between CDBEA and the other five algorithms on the IGD metrics are shown in Table 2. According to the results CDBEA has a competitive advantage in IGD metrics for WFG test problems. As shown in the table, CDBEA achieves the best results on 20 out of 36 test instances, while RVEA, hpaEA, MOEAD, MaOEAIT, and DEAGNG perform best on 4, 5, 0, 1, and 6 test instances, respectively. Based on the Wilcoxon rank sum test to examine the performance differences with the five CDBEA comparison algorithms on IGD metrics, it can be concluded that CDBEA outperforms the other five comparison algorithms on 16, 21, 34, 34 and 15 test suites, respectively. In addition, CDBEA obtains the best performance on the test cases WFG2-WFG9 the highest number of times and the algorithm still performs well when the number of objectives increases, proving that CDBEA has a competitive advantage in terms of comprehensive performance. DEAGNG performs second only to CDBEA, and also solves the WFG test problem well. In addition, since WFG3 is a degenerate problem, none of the six algorithms converge to the PF on the objectives 8 and 10.

In order to visualize the comparison between CDBEA and RVEA, their output populations with maximum HV values over 30 runs on the 10-objective WFG7 are plotted in parallel coordinates. Figure 2 gives the results of the comparison between CDBEA and RVEA on WFG7 at 10-objective.

WFG7 is a separable unimodal problem. As can be observed from the figure, RVEA outputs only a few solutions and does not cover the middle part of each objective well. For RVEA, it fails to converge to the true PF on some of the objectives. CDBEA obtains a solution set with better convergence and diversity due to its focus on the convergence of the solutions during the evolutionary process and adjusts the reference vectors according to the overall convergence. RVEA lacks an effective method of adaptive adjustment of the reference vectors, and thus obtains solutions with poorer diversity.

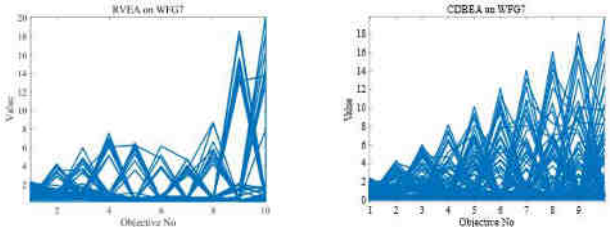

Figure 2: Parallel coordinates of the final solution set obtained by RVEA and CDBEA on the 10-objective WFG7

Table 1: Statistical results of the HV values obtained by the six algorithms in the WFG test suite

problem	objective	RVEA	hpaEA	MOEAD	MaOEAIT	DEAGNG	CDBEA
WFG1	3	1.2828e-4 (6.97e-4) =	3.7391e-3 (1.19e-2) =	3.0964e-3 (6.09e-3) +	3.0609e-2 (2.63e-2) +	1.2612e-2 (1.30e-2) +	1.0167e-4 (5.32e-4)
	5	8.3192e-4 (1.64e-3) -	2.2586e-2 (3.15e-2) +	1.3198e-1 (8.42e-2) +	1.0187e-2 (1.52e-2) =	4.0491e-3 (6.30e-3) =	3.2731e-3 (3.41e-3)
	8	1.3739e-3 (3.23e-3) -	6.1516e-3 (9.77e-3) =	1.0216e-1 (9.67e-2) +	4.8957e-4 (2.02e-3) -	1.5265e-4 (2.54e-4) -	5.0394e-3 (1.06e-2)
	10	5.3289e-3 (1.04e-2) =	7.0233e-3 (1.31e-2) =	3.2004e-2 (9.54e-2) =	2.2337e-4 (1.16e-3) -	3.0871e-4 (8.75e-4) -	7.9198e-3 (1.01e-2)
WFG2	3	8.9007e-1 (9.61e-3) +	9.0409e-1 (2.54e-2) +	8.2499e-1 (3.07e-2) -	8.5530e-1 (2.52e-2) -	9.0879e-1 (5.68e-3) +	8.7694e-1 (1.42e-2)
	5	9.2564e-1 (2.05e-2) -	9.4352e-1 (1.41e-2) =	8.1579e-1 (4.59e-2) -	8.6670e-1 (4.23e-2) -	9.3554e-1 (1.02e-2) =	9.3678e-1 (2.29e-2)
	8	8.8265e-1 (4.19e-2) -	9.0679e-1 (5.07e-2) -	7.5345e-1 (7.46e-2) -	8.0961e-1 (4.97e-2) -	9.3060e-1 (3.11e-2) =	9.2820e-1 (4.25e-2)
	10	8.9640e-1 (5.41e-2) -	9.0631e-1 (6.98e-2) =	7.4702e-1 (7.31e-2) -	7.4921e-1 (6.31e-2) -	9.3564e-1 (3.23e-2) =	9.3548e-1 (3.36e-2)
WFG3	3	3.0435e-1 (1.37e-2) +	3.5215e-1 (8.58e-3) +	2.8308e-1 (3.50e-2) +	9.2546e-2 (2.52e-2) -	3.5898e-1 (7.59e-3) +	2.5567e-1 (3.22e-2)
	5	5.3130e-2 (2.71e-2) +	5.6637e-2 (1.85e-2) +	8.8807e-4 (4.21e-3) -	0.0000e+0 (0.00e+0) -	5.3364e-2 (2.77e-2) +	3.6463e-3 (4.88e-3)
	8	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0)
	10	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0)
WFG4	3	5.1792e-1 (4.02e-3) -	5.2669e-1 (7.05e-3) =	4.9014e-1 (8.99e-3) -	4.2587e-1 (1.56e-2) -	5.3540e-1 (2.91e-3) +	5.2606e-1 (4.68e-3)
	5	6.9995e-1 (8.73e-3) -	6.1852e-1 (3.32e-2) -	4.8267e-1 (4.78e-2) -	3.8975e-1 (3.22e-2) -	6.4160e-1 (1.16e-2) -	7.0936e-1 (1.20e-2)
	8	7.2328e-1 (4.05e-2) -	4.9043e-1 (3.68e-2) -	2.4928e-1 (4.61e-2) -	3.2809e-1 (3.74e-2) -	6.2539e-1 (3.77e-2) -	7.6258e-1 (6.40e-2)
	10	7.2686e-1 (4.54e-2) =	5.0394e-1 (4.59e-2) -	2.2505e-1 (4.59e-2) -	3.0058e-1 (2.93e-2) -	6.3422e-1 (3.92e-2) -	7.2907e-1 (8.77e-2)
WFG5	3	4.9526e-1 (4.80e-3) =	5.0430e-1 (4.27e-3) +	4.7675e-1 (8.58e-3) -	4.1477e-1 (1.97e-2) -	4.9538e-1 (3.09e-3) =	4.9429e-1 (7.85e-3)
	5	6.8342e-1 (6.97e-3) +	6.6255e-1 (1.18e-2) -	4.8094e-1 (3.34e-2) -	3.8149e-1 (2.62e-2) -	6.0565e-1 (1.29e-2) -	6.7661e-1 (8.13e-3)
	8	7.1796e-1 (2.58e-2) -	5.0244e-1 (2.50e-2) -	2.2107e-1 (2.78e-2) -	2.7075e-1 (2.86e-2) -	5.7293e-1 (3.69e-2) -	7.5043e-1 (1.87e-2)
	10	7.5575e-1 (3.07e-2) -	5.9347e-1 (2.96e-2) -	2.0202e-1 (3.59e-2) -	2.4894e-1 (3.40e-2) -	5.6079e-1 (4.57e-2) -	7.9510e-1 (2.29e-2)
WFG6	3	4.5416e-1 (1.61e-2) =	4.7649e-1 (1.35e-2) +	4.4671e-1 (1.96e-2) -	3.7278e-1 (1.95e-2) -	4.7740e-1 (1.21e-2) +	4.6461e-1 (1.99e-2)
	5	6.2754e-1 (3.51e-2) =	6.2457e-1 (2.09e-2) -	3.4857e-1 (4.60e-2) -	3.4405e-1 (3.15e-2) -	5.7996e-1 (2.12e-2) -	6.4024e-1 (2.23e-2)
	8	5.8851e-1 (4.27e-2) -	4.8955e-1 (4.58e-2) -	1.1589e-1 (4.31e-2) -	2.4089e-1 (2.97e-2) -	5.6380e-1 (4.63e-2) -	6.6063e-1 (4.58e-2)
	10	4.8918e-1 (9.69e-2) -	5.6550e-1 (4.18e-2) -	1.0071e-1 (4.26e-2) -	2.2326e-1 (2.62e-2) -	5.2791e-1 (4.31e-2) -	7.4826e-1 (4.70e-2)
WFG7	3	5.1770e-1 (4.29e-3) -	5.3505e-1 (2.77e-3) +	4.3496e-1 (2.60e-2) -	3.6553e-1 (2.30e-2) -	5.4125e-1 (3.87e-3) +	5.2780e-1 (7.36e-3)
	5	7.0494e-1 (1.46e-2) =	6.6669e-1 (1.64e-2) -	4.3262e-1 (6.22e-2) -	3.5436e-1 (2.76e-2) -	6.4969e-1 (1.73e-2) -	7.0852e-1 (1.76e-2)
	8	6.8928e-1 (5.48e-2) -	5.0803e-1 (4.18e-2) -	2.1721e-1 (5.72e-2) -	3.0674e-1 (1.96e-2) -	6.1545e-1 (4.69e-2) -	7.8553e-1 (3.23e-2)
	10	6.5129e-1 (1.47e-1) -	5.6368e-1 (4.15e-2) -	1.6571e-1 (6.27e-2) -	2.9443e-1 (2.11e-2) -	6.2434e-1 (4.42e-2) -	8.3339e-1 (6.80e-2)
WFG8	3	4.2712e-1 (6.10e-3) =	4.4457e-1 (3.92e-3) +	4.0929e-1 (1.73e-2) -	3.4192e-1 (1.85e-2) -	4.4898e-1 (4.18e-3) +	4.2488e-1 (1.00e-2)
	5	5.6096e-1 (1.45e-2) -	5.6832e-1 (1.06e-2) =	2.0751e-1 (6.92e-2) -	3.0589e-1 (3.89e-2) -	5.2040e-1 (1.71e-2) -	5.7084e-1 (1.46e-2)
	8	4.7996e-1 (8.85e-2) -	4.8682e-1 (3.56e-2) -	4.5023e-2 (8.57e-2) -	2.0418e-1 (2.55e-2) -	5.0886e-1 (3.24e-2) -	6.9030e-1 (7.12e-2)
	10	3.8240e-1 (1.61e-1) -	5.6392e-1 (2.05e-2) -	2.0282e-1 (2.94e-1) -	1.7766e-1 (2.94e-2) -	5.4232e-1 (3.64e-2) -	7.4395e-1 (3.21e-2)
WFG9	3	4.8188e-1 (2.19e-2) -	5.1237e-1 (5.05e-3) +	4.0868e-1 (4.64e-2) -	4.0984e-1 (1.84e-2) -	5.1135e-1 (2.10e-2) +	4.9550e-1 (1.07e-2)
	5	6.0872e-1 (7.91e-2) =	6.1302e-1 (3.16e-2) =	4.1999e-1 (8.19e-2) -	3.3762e-1 (3.25e-2) -	6.2089e-1 (2.64e-2) =	6.2304e-1 (3.09e-2)
	8	6.0355e-1 (4.05e-2) -	4.6556e-1 (3.07e-2) -	1.8760e-1 (7.08e-2) -	2.5460e-1 (2.84e-2) -	5.9442e-1 (3.81e-2) -	6.6099e-1 (5.17e-2)
	10	5.7139e-1 (6.68e-2) -	5.2202e-1 (4.54e-2) -	1.0877e-1 (7.14e-2) -	2.3153e-1 (2.19e-2) -	5.6452e-1 (4.52e-2) -	6.7294e-1 (8.91e-2)
+/-/=		4/21/11	9/17/10	4/29/3	1/32/3	9/19/8	

Table 2: Statistical results of the IGDvalues obtained by the six algorithms in the WFG test suite

problem	objective	RVEA	hpaEA	MOEAD	MaOEAIT	DEAGNG	CDBEA
WFG1	3	3.2894e-1 (1.05e-1) -	5.9861e-1 (2.30e-1) -	2.0140e-1 (1.00e-1) =	4.0404e-1 (2.17e-1) -	1.6735e-1 (5.81e-2) +	2.4760e-1 (9.08e-2)
	5	3.8960e-1 (1.25e-1) =	8.7275e-1 (2.54e-1) -	8.7808e-1 (1.94e-1) -	8.2355e-1 (1.90e-1) -	4.1714e-1 (1.05e-1) =	3.6111e-1 (1.14e-1)
	8	6.1541e-1 (1.95e-1) +	9.8290e-1 (2.39e-1) -	1.6354e+0 (3.01e-1) -	9.2653e-1 (2.38e-1) =	7.4963e-1 (1.61e-1) +	8.7811e-1 (2.19e-1)
	10	1.0879e+0 (3.11e-1) =	1.3600e+0 (3.02e-1) =	1.5694e+0 (4.57e-1) -	8.7162e-1 (2.47e-1) +	1.0929e+0 (2.11e-1) +	1.2206e+0 (2.57e-1)
WFG2	3	2.1344e-1 (1.10e-2) +	1.9126e-1 (6.32e-2) +	3.1870e-1 (3.63e-2) -	2.8047e-1 (6.50e-2) -	1.7479e-1 (7.25e-3) +	2.2707e-1 (2.19e-2)
	5	4.9992e-1 (1.95e-2) +	5.2294e-1 (2.21e-2) +	9.5608e-1 (6.53e-2) -	7.1310e-1 (2.21e-1) -	5.2067e-1 (3.28e-2) +	5.7180e-1 (2.81e-2)
	8	1.1783e+0 (7.30e-2) =	1.3713e+0 (4.29e-1) -	1.9507e+0 (3.29e-1) -	1.4191e+0 (3.92e-1) -	1.2686e+0 (2.97e-1) -	1.1665e+0 (6.52e-2)
	10	1.3627e+0 (5.49e-2) =	1.9276e+0 (8.82e-1) -	2.0942e+0 (5.07e-1) -	2.0367e+0 (7.09e-1) -	1.6206e+0 (4.16e-1) -	1.3608e+0 (1.20e-1)
WFG3	3	2.6978e-1 (3.09e-2) +	1.4219e-1 (1.61e-2) +	3.2258e-1 (1.31e-1) =	7.6700e-1 (1.02e-1) -	1.3769e-1 (1.28e-2) +	2.9411e-1 (5.34e-2)
	5	7.1449e-1 (8.46e-2) +	5.4642e-1 (4.91e-2) +	1.3394e+0 (3.18e-1) -	1.1274e+0 (1.30e-1) -	6.7192e-1 (2.11e-1) +	8.4336e-1 (1.14e-1)
	8	2.3915e+0 (5.63e-1) -	1.2851e+0 (4.95e-1) =	3.9174e+0 (3.24e-1) -	1.6786e+0 (1.55e-1) -	1.0712e+0 (3.76e-1) =	1.0707e+0 (4.41e-1)
	10	4.2717e+0 (1.31e+0) -	1.4034e+0 (4.17e-1) -	5.8137e+0 (4.47e-1) -	2.0077e+0 (1.50e-1) -	1.4735e+0 (6.36e-1) -	1.1022e+0 (3.64e-1)
WFG4	3	2.6552e-1 (7.58e-3) -	2.4999e-1 (3.11e-2) =	2.9299e-1 (1.10e-2) -	4.5607e-1 (5.72e-2) -	2.5180e-1 (3.97e-3) -	2.4996e-1 (4.06e-3)
	5	1.4289e+0 (7.81e-3) -	1.5978e+0 (1.76e-1) -	2.6098e+0 (2.45e-1) -	2.2050e+0 (3.01e-1) -	1.4263e+0 (1.79e-2) -	1.3798e+0 (3.97e-2)
	8	4.6757e+0 (1.04e-1) -	6.1650e+0 (6.05e-1) -	8.4381e+0 (3.61e-1) -	6.4427e+0 (6.93e-1) -	4.2675e+0 (2.42e-1) =	4.1749e+0 (6.90e-1)
	10	6.0599e+0 (1.30e-1) +	9.5323e+0 (9.80e-1) -	1.0849e+1 (4.18e-1) -	9.7675e+0 (1.04e+0) -	6.1306e+0 (5.71e-1) +	7.2891e+0 (1.28e+0)
WFG5	3	2.5955e-1 (7.25e-3) -	2.5582e-1 (2.69e-3) =	2.7624e-1 (1.13e-2) -	4.5265e-1 (6.39e-2) -	2.5634e-1 (2.95e-3) -	2.5536e-1 (5.77e-3)
	5	1.2167e+0 (6.94e-3) +	1.2315e+0 (4.67e-2) +	2.6231e+0 (1.39e-1) -	1.8680e+0 (2.06e-1) -	1.2211e+0 (1.96e-2) +	1.3532e+0 (3.27e-2)
	8	3.7627e+0 (8.77e-2) =	4.4919e+0 (2.08e-1) -	8.1445e+0 (1.81e-1) -	5.7340e+0 (5.42e-1) -	3.7961e+0 (2.12e-1) =	3.7279e+0 (1.66e-1)

	10	6.0077e+0 (9.30e-2) -	6.7534e+0 (4.77e-1) -	1.0405e+1 (2.62e-1) -	8.3947e+0 (8.99e-1) -	6.0971e+0 (4.48e-1) -	5.6125e+0 (2.42e-1)
WFG6	3	3.3477e-1 (2.27e-2) -	2.6959e-1 (1.59e-2) +	3.3353e-1 (2.47e-2) -	5.2529e-1 (7.56e-2) -	2.7149e-1 (1.18e-2) +	3.0475e-1 (2.12e-2)
	5	1.4768e+0 (6.97e-2) -	1.5757e+0 (3.90e-2) -	3.1025e+0 (2.43e-1) -	1.9341e+0 (2.62e-1) -	1.4607e+0 (2.55e-2) -	1.4115e+0 (5.77e-2)
	8	4.0300e+0 (2.34e-1) =	4.5953e+0 (3.13e-1) -	8.5315e+0 (1.43e-1) -	6.0325e+0 (7.86e-1) -	3.9035e+0 (2.70e-1) +	4.1655e+0 (2.17e-1)
	10	7.1097e+0 (4.55e-1) -	6.9772e+0 (5.49e-1) -	1.0902e+1 (2.64e-1) -	8.2670e+0 (8.38e-1) -	6.5592e+0 (5.96e-1) -	5.8451e+0 (2.80e-1)
WFG7	3	2.7575e-1 (1.39e-2) -	2.2469e-1 (2.92e-3) +	4.0280e-1 (3.77e-2) -	6.5080e-1 (9.13e-2) -	2.2608e-1 (4.78e-3) +	2.6517e-1 (1.85e-2)
	5	1.2416e+0 (1.14e-2) +	1.3000e+0 (5.32e-2) +	2.8667e+0 (3.08e-1) -	2.0152e+0 (2.39e-1) -	1.2367e+0 (3.01e-2) +	1.4358e+0 (3.67e-2)
	8	3.9892e+0 (1.38e-1) =	4.9240e+0 (4.01e-1) -	8.5650e+0 (6.56e-1) -	6.1730e+0 (5.07e-1) -	3.9043e+0 (2.98e-1) =	3.9021e+0 (2.20e-1)
	10	5.9998e+0 (2.64e-1) =	7.8948e+0 (6.70e-1) -	1.1397e+1 (2.69e-1) -	8.7406e+0 (6.83e-1) -	6.3581e+0 (4.61e-1) -	5.9968e+0 (6.47e-1)
WFG8	3	3.7483e-1 (1.35e-2) -	3.1390e-1 (5.17e-3) +	3.7530e-1 (2.81e-2) -	6.1452e-1 (7.30e-2) -	3.1145e-1 (5.46e-3) +	3.6128e-1 (1.24e-2)
	5	1.3642e+0 (1.65e-2) -	1.3611e+0 (7.08e-2) -	2.3552e+0 (2.45e-1) -	2.1440e+0 (2.30e-1) -	1.3624e+0 (4.31e-2) -	1.3579e+0 (2.39e-2)
	8	3.9650e+0 (2.11e-1) =	4.9407e+0 (3.56e-1) -	7.1496e+0 (1.10e+0) -	5.9800e+0 (4.51e-1) -	4.3766e+0 (2.55e-1) -	3.9646e+0 (3.07e-1)
	10	6.6573e+0 (5.22e-1) =	7.1723e+0 (4.14e-1) -	8.8883e+0 (1.73e+0) -	8.8017e+0 (8.49e-1) -	7.0086e+0 (5.32e-1) -	6.5619e+0 (5.00e-1)
WFG9	3	2.7896e-1 (3.27e-2) -	2.2499e-1 (5.03e-3) +	3.7693e-1 (7.08e-2) -	4.5304e-1 (4.98e-2) -	2.2810e-1 (2.27e-2) +	2.5786e-1 (8.98e-3)
	5	1.2538e+0 (2.11e-1) +	1.2006e+0 (3.00e-2) +	2.5047e+0 (1.90e-1) -	1.8730e+0 (1.55e-1) -	1.2084e+0 (2.10e-2) +	1.3173e+0 (3.31e-2)
	8	3.6766e+0 (8.08e-2) =	4.6297e+0 (3.72e-1) -	7.8498e+0 (4.23e-1) -	6.0122e+0 (5.25e-1) -	3.6923e+0 (1.29e-1) -	3.6319e+0 (1.94e-1)
	10	5.7915e+0 (1.96e-1) -	7.1310e+0 (7.38e-1) -	1.0493e+1 (3.90e-1) -	9.0798e+0 (5.80e-1) -	6.0252e+0 (4.90e-1) -	5.7847e+0 (6.90e-1)
+/-/=		9/16/11	11/21/4	0/34/2	1/34/1	16/15/5	

V. Conclusions

The algorithm proposes a method to judge the timing of reference vector adjustment based on the change of population convergence in order to periodically and conditionally trigger the reference vector adjustment. This strategy ensures that the reference vector adjustment is based on the actual evolution of the population, avoids unnecessary adjustment overhead, and improves the efficiency of the algorithm. In addition, the reference vector adjustment process chooses a vector angle-based adjustment method. Specifically, by maintaining well-distributed non-dominated solutions in archive, the individual among them that possesses the largest vector angle with the current population is selected as the basis for reference vector adjustment. This method enables the reference vectors to be more evenly distributed over the PFs, increasing the validity of the reference vectors and thus better adapting to PFs with different shapes. In the future research, we will work on designing novel methods to evaluate the convergence of populations in order to reflect more comprehensively and accurately the convergence state of populations in high-dimensional spaces.

References

[1] S. Yang, M. Li, X. Liu, et al. A Grid-Based Evolutionary Algorithm for Many-Objective Optimization[J]. IEEE Transactions on Evolutionary Computation, 2013, 17(5):721-736.

[2] X. Zou, Y. Chen, M. Liu, et al. A New Evolutionary Algorithm for Solving Many-Objective Optimization Problems[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B ,2008, 38(5): 1402-1412.

[3] E. Zitzler, S. Künzli. Indicator-based selection in multi objective search[J]. International Conference on Parallel Problem Solving from Nature. Springer. 2004, 3242:(832–842).

[4] J. Bader and E. Zitzler. HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization[J]. Evolutionary Computation, 2011,19(1): 45-76.

[5] Q. Zhang and H. Li. MOEA/D: A Multi-objective Evolutionary Algorithm Based on Decomposition[J]. IEEE Transactions on Evolutionary Computation, 2007,11(6):712-731.

[6] J. Shen, P. Wang and X. Wang. A Controlled Strengthened Dominance Relation for Evolutionary Many-Objective Optimization[J]. IEEE Transactions on Cybernetics, 2022,52(5): 3645-3657.

[7] R. Cheng, Y. Jin, M. Olhofer and B. Sendhoff. A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization[J] IEEE Transactions on Evolutionary Computation, 2016, 20(5): 773-791.

[8] Wang Z, Zhang Q, Li H, Ishibuchi H, Jiao L. On the use of two reference points in decomposition based multi-objective evolutionary algorithms[J], Swarm. Evol Comput, 2017, 34:89–102.

[9] Liu H. L, Chen L, Zhang Q and Deb K. Adaptively Allocating Search Effort in Challenging Many-Objective Optimization Problems[J]. IEEE Transactions on Evolutionary Computation, 2018, 22(3): 433-448.

[10] Asafuddoula M, Singh H. K and Ray. T. An Enhanced Decomposition-Based Evolutionary Algorithm with Adaptive Reference Vectors[J]. IEEE Transactions on Cybernetics, 2018, 48(8): 2321-2334.

[11] Qi Y, Ma X, Liu F, Jiao L, et al. MOEA/D with Adaptive Weight Adjustment. Evolutionary Computation[J], 2014, 22(2): 231-264.

[12] L. R. C. deFarias, P. H. M. Braga, H. F. Bassani, et al. MOEA/D with uniformly randomly adaptive weights[J]. Genetic and Evolutionary Computation Conference, 2018: 641–648.

[13] Tian Y, Cheng R, Zhang X, Cheng F, et al. An Indicator-Based Multiobjective Evolutionary Algorithm with Reference Point Adaptation for Better Versatility[J]. IEEE Transactions on Evolutionary Computation, 2018, 22(4): 609-622.

[14] Lin Y, Liu H, and Jiang Q. Dynamic reference vectors and biased crossover use for inverse model based evolutionary multi-objective optimization with irregular Pareto fronts[J]. Applied Intelligence, 2022, 48(9): 3116–3142.

[15] Liu Y, Ishibuchi. H, Masuyama. N, and Nojima. Y. Adapting reference vectors and scalarizing functions by growing neural gas to handle irregular Pareto fronts[J]. IEEE Trans. Evol. Comput, early access, 2019.

[16] Sun Y, Xue B, Zhang M and G. G. Yen. A New Two-Stage Evolutionary Algorithm for Many-Objective Optimization[J]. IEEE Transactions on Evolutionary Computation, 2021, 23(5): 748-761.

[17] Li M, Yang S and Liu X. Pareto or Non-Pareto: Bi-Criterion Evolution in Multi-objective Optimization[J]. IEEE Transactions on Evolutionary Computation,2016, 20(5): 645-665.

[18] Huband S, Hingston P, Barone L, et al. A review of multi-objective test problems and a scalable test problem toolkit[J]. IEEE Transactions on Evolutionary Computation, 2006, 10(5): 477-506.

[19] Chen H, Tian Y, Pedrycz W, et al. Hyperplane Assisted Evolutionary Algorithm for Many-Objective Optimization Problems[J]. *IEEE Transactions on Cybernetics*, 2020, 50(7): 3367-3380

[20] Zitzler. E and Thiele. L. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach[J]. IEEE Transactions on Evolutionary Computation, 1999, 3(4): 257-271.

[21] P. A. N. Bosman and D. Thierens. The balance between proximity and diversity in multiobjective evolutionary algorithms[J]. IEEE Transactions on Evolutionary Computation, 2003,7(2): 174-188.