

# A Two-Stage Multi-Objective Optimization Algorithm based on Estimated Pareto Front

Changhao Yang<sup>1</sup> and Jin Qin<sup>1\*</sup>

<sup>1</sup>The State Key Laboratory of Public Big Data, Institution of Artificial Intelligence, GuiZhou University, GuiYang, 550025, GuiZhou, China.

\*Corresponding author(s). E-mail(s): [jinqin1@gzu.edu.cn](mailto:jinqin1@gzu.edu.cn);  
Contributing authors: [ych\\_heihei@163.com](mailto:ych_heihei@163.com);

## Abstract

Recently, multi-objective optimization algorithms based on estimated Pareto front (PF) have emerged and achieved good results. However, due to the complexity of PFs and the insufficient convergence of Pareto nondominated solutions during the optimization process, the estimated PF changes dramatically and has a significant error with the true PF, which may mislead the selection of solutions. To better utilize the estimated PF to guide the selection, we propose a two-stage multi-objective optimization algorithm based on estimated Pareto front (TS-MOEA-EPF). In the first stage of this algorithm, a modified growing neural gas (GNG) is used to guide the population to maintain diversity while converging as soon as possible, thus providing Pareto nondominated solutions close enough to the true PF. In the second stage, a neural network is used to fit the PF of any shape by using these Pareto nondominated solutions as training examples. Next, we use the projection points of solutions on the estimated PF to more directly and effectively measure the convergence and diversity of the solutions, thereby preserving those solutions with good diversity and convergence for the next population. To verify the effectiveness of TS-MOEA-EPF, we conducted comparative experiments with five other advanced algorithms on 57 benchmark problems. The experimental results show that TS-MOEA-EPF has achieved good results on most benchmark problems, especially on problems with complicated PF and Pareto nondominated solutions are far from the true PF in the early stages of the optimization.

**Keywords:** Growing neural gas, evolutionary multi-objective optimization, Pareto front estimation, irregular Pareto front.

# 1 Introduction

Multi-objective optimization problems (MOPs) are commonly seen in real-world [1, 2]. MOPs involve two or more conflicting objectives that cannot simultaneously achieve their respective optimal values. Therefore, they can only obtain a set of mutually compromising solutions. The mathematical form is as follows:

$$\min \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_M(\mathbf{x})), s.t. \mathbf{x} \in \Omega \quad (1)$$

where  $\mathbf{x}$  is the  $n$ -dimensional decision vector in the search space  $\Omega$ ,  $\mathbf{f}(\mathbf{x})$  is the objective vector of  $\mathbf{x}$ ,  $f_i(\mathbf{x})$  is the  $i$ -th objective ( $i=1, 2, \dots, M$ ), and  $M$  is the number of objectives [3]. In  $\Omega$ ,  $\mathbf{x}^1$  Pareto dominates  $\mathbf{x}^2$  if they satisfy Eq. (2), denoted as  $\mathbf{x}^1 \prec \mathbf{x}^2$ .

$$\begin{aligned} \forall i \in 1, \dots, M, f_i(\mathbf{x}^1) &\leq f_i(\mathbf{x}^2) \\ \text{and } \exists j \in 1, \dots, M, f_j(\mathbf{x}^1) &< f_j(\mathbf{x}^2) \end{aligned} \quad (2)$$

In  $\Omega$ ,  $\mathbf{x}^*$  is regarded as Pareto optimal solution if there's no other  $\mathbf{x}$  satisfying  $\mathbf{x} \prec \mathbf{x}^*$ . All the Pareto optimal solutions together form the Pareto optimal solution set (PS), and the objective space of PS is called Pareto front (PF). Since PS may be an infinite set, in practice we may only be able to obtain a finite subset of PS for decision-making. The goal of multi-objective optimization algorithms is to find a finite set of solutions with good convergence and diversity for approximating the true PF as much as possible [4]. Many multi-objective optimization algorithms have been proposed. Overall, the current algorithms can be divided into three categories: Pareto dominance-based algorithms [5], indicator-based algorithms [6] and decomposition-based algorithms [7].

Pareto dominance-based algorithms generally prefer solutions with greater diversity under the premise of prioritizing nondominated solutions. However, in many objective optimization problems (MaOPs) with  $M>3$ , traditional Pareto dominance-based algorithms may not be able to measure the convergence of solutions because it is difficult for one solution to dominate others. This leads to the selection of solutions mainly based on diversity, resulting in the preservation of distant solutions with poor convergence, known as the Pareto dominance resistance phenomenon [8]. Algorithms like SDE [9], CV [10],  $\epsilon$ -dominance [11], NAEA [12], F-DEA [13], etc, to some extent, measure the convergence and diversity of solutions in a more comparable way compared to traditional Pareto dominance-based algorithms, but the impact of the parameter settings on the balance of convergence and diversity still needs to be considered [14].

In indicator-based algorithms, the indicators simultaneously consider the diversity and convergence of solutions. Common indicators include hypervolume (HV) [15] and inverted generational distance (IGD) [16]. Since the computational complexity of indicator-based algorithms increases exponentially with the number of objectives [17], methods such as HypE [6] and R2HCA-EMOA [18] approximate the HV value through estimation methods and can reduce the computational complexity to some extent. However, there is still a need to address the issue of balancing estimation accuracy and computational complexity. HV based algorithms may exhibit different preferences between solutions in the middle region and those on the borders of convex/concave

PF [19]. IGD based algorithms require obtaining uniform sampling points on the true PF, unfortunately the true PF is unknown before optimization. MaOEA/IGD [20] uses uniform sampling points on the hyperplane composed of the extreme points to guide the selection of solutions.

In decomposition-based algorithm, the objective space is divided into multiple sub-regions by setting the reference vectors (or the reference points), and the intersection points between the reference vectors and the true PF are the optimal solutions for each subregion [21]. Compared to Pareto dominance-based algorithms, decomposition-based algorithms are less affected by the number of objectives and can effectively converge towards these directions of the reference vectors [22]. However, the distribution of the obtained population is highly sensitive to the setting of the reference vectors. If the intersection points between the reference vectors and the true PF show an uneven distribution, or if there are invalid reference vectors that do not intersect with the PF, then the obtained solutions are difficult to approximate the true PF [23, 24].

The emerging multi-objective algorithms based on estimated PF estimate the PF through a set of reference vectors or models [25]. For instance, RVEA [26], MOEA/D-AWA [27] and SPEA/ARP [28] adjust the reference vectors based on the distribution of Pareto nondominated solutions, so that the reference vectors guide the selection of solutions towards more uniform and more effective directions. To achieve convergence as quickly as possible while maintaining diversity, PeEA [29], and MaOEA-3C [30] attempt to estimate the concavity of the true PF for designing the convergence metric. MOEA-PFC [31] adopts reference vectors starting from the ideal point and the nadir point respectively for local concave PF and local convex PF. MOEA/D-SOM [32], DEA-GNG [33] and RVEA-iGNG [34] adjust the reference vectors by learning the topological structure of the objective space corresponding to the current Pareto nondominated solutions. GFM-MOEA [19] estimates the PF through a polynomial model and measures the convergence and diversity of solutions based on their projection points on the estimated PF. LMPFE [25] goes further by using local models to obtain the corresponding local estimated PFs for the partitioned objective spaces. However, due to the insufficient convergence of Pareto nondominated solutions or the excessive complexity of the true PF, these models may not sufficiently approximate the PF, which may mislead the optimization.

Inspired by DEA-GNG and LMPFE, we propose a two-stage multi-objective optimization algorithm based on estimated Pareto front (TS-MOEA-EPF). This algorithm utilizes a modified growing neural gas (GNG) to gradually learn the distribution of Pareto nondominated solutions in the objective space, thereby obtaining reference vectors that conform to the current distribution of Pareto nondominated solutions. These reference vectors guide the selection of solutions towards more uniform and more effective directions. As the population continues to converge, the current Pareto nondominated solutions get closer to the true PF. We use a shallow neural network with strong generalization ability to estimate the true PF based on the current Pareto nondominated solutions. Next, we evaluate the projection points of solutions on the estimated PF to further measure their convergence and diversity for selecting solutions more effectively. Our contributions are as follows:

1. A new archive filtering mechanism for training GNG and a new structure of GNG have been designed. Through these measures, the archive will retain Pareto non-dominated solutions with good convergence and diversity. The modified GNG will be able to learn the distribution of Pareto nondominated solutions in the archive more accurately and gradually guide the selection of solutions towards more uniform and more effective directions.
2. We use a shallow neural network to estimate the PF based on Pareto nondominated solutions in the later stages of the optimization. Through the neural network and Pareto nondominated solutions with good convergence and diversity, theoretically we can fit PF of any shape. Thus, the estimated PF can better guide the selection of solutions.
3. We conducted comparative experiments with five other advanced algorithms on 57 benchmark problems to verify the effectiveness of the proposed algorithm.

Next, we will introduce our work in the following order. In Section 2, we will elaborate on our research motivation and basic ideas. Section 3 will detail our proposed algorithm. Section 4 will explain our experimental design and results. Finally, in section 5, we will summarize our work.

## 2 Related Work and Motivation

This section will provide a detailed introduction to the GNG and LMPFE which select solutions based on their projection points on the estimated PF, as well as our research motivation.

### 2.1 Growing Neural Gas

GNG can reflect the distribution of a large number of data points through a small number of nodes and edges [35], making it well-suited for representing the distribution of Pareto nondominated solutions in the objective space. In GNG, each edge is associated with an *age*, with a maximum value of  $age_{max}$ . Additionally, each node has an associated *error*. The *age* indicates whether there are data points near the node, while the *error* reflects how many data points are closest to the current node and how close they are to it. In DEA-GNG [33], each node is additionally associated with a *HP*, while in RVEA-iGNG [34], each node is additionally associated with a *flag*. Both the *HP* and the *flag* are used to identify outdated nodes which do not reflect the distribution of current data points, and these nodes will be deleted.

The training process of GNG is as follows: First, find the node  $\mathbf{r}_a$  closest to the current data point  $\boldsymbol{\xi}$  and the second closest node  $\mathbf{r}_b$ . Then, connect  $\mathbf{r}_b$  with  $\mathbf{r}_a$  using an edge, setting the *age* of this edge to 0, and the *ages* of other edges emitted from  $\mathbf{r}_a$  will increase by 1. Use Eq. (3) to update the nodes, where  $nb(\mathbf{r}_a)$  represents other nodes with an edge connected to  $\mathbf{r}_a$  and  $k=1, \dots, |nb(\mathbf{r}_a)|$ .  $\epsilon_a$  and  $\epsilon_{nb}$  are the learning rates.

$$\begin{aligned} \mathbf{r}_a &= \mathbf{r}_a + \epsilon_a (\boldsymbol{\xi} - \mathbf{r}_a) \\ \mathbf{r}_k^{nb(\mathbf{r}_a)} &= \mathbf{r}_k^{nb(\mathbf{r}_a)} + \epsilon_{nb} (\boldsymbol{\xi} - \mathbf{r}_k^{nb(\mathbf{r}_a)}) \end{aligned} \quad (3)$$

Afterwards, update the cumulative *error* of  $\mathbf{r}_a$  using Eq. (4).

$$\text{error}(\mathbf{r}_a) = \text{error}(\mathbf{r}_a) + ||\boldsymbol{\xi} - \mathbf{r}_a||^2 \quad (4)$$

If the *age* of an edge is greater than  $age_{max}$ , then delete it. Additionally, isolated nodes without associated edges are removed. In DEA-GNG, the *HP* of node  $\mathbf{r}_a$  is restored to its maximum value  $HP_{Max}$ , the *HP* of  $\mathbf{r}_b$  remains unchanged, the *HPs* of remaining nodes decrease by 1. If there are nodes with *HP* less than or equal to 0, delete these nodes and theirs associated edges. In RVEA-iGNG, the *flag* of a node that has not been updated for a long time increases by 1, and nodes with larger *flag* are deleted first.

When the number of nodes is less than the upper bound, new nodes need to be generated. First, find the node  $\mathbf{r}_1^{max}$  with the current maximum *error*. Next, find the second node  $\mathbf{r}_2^{max}$  with the highest cumulative *error* from  $nb(\mathbf{r}_1^{max})$ , and generate a new node as shown in Eq. (5).

$$\mathbf{r}_{new} = 0.5 \cdot (\mathbf{r}_1^{max} + \mathbf{r}_2^{max}) \quad (5)$$

Delete the edge between  $\mathbf{r}_1^{max}$  and  $\mathbf{r}_2^{max}$ , and establish edges between  $\mathbf{r}_{new}$  and  $\mathbf{r}_1^{max}$ , as well as between  $\mathbf{r}_{new}$  and  $\mathbf{r}_2^{max}$ . Set the *ages* of the associated edges of  $\mathbf{r}_{new}$  to 0. Use  $\alpha$  to discount the errors of  $\mathbf{r}_1^{max}$  and  $\mathbf{r}_2^{max}$ , and then assign  $\text{error}(\mathbf{r}_1^{max})$  to  $\text{error}(\mathbf{r}_{new})$ . Finally, use  $\delta$  to decay all *errors* associated with these nodes.

## 2.2 LMPFE

In LMPFE [25], the current Pareto nondominated solutions are divided into a specified number of clusters, and the following model is used for each cluster to obtain the local estimated PF, where  $f'_i$  represent the  $i$ -th normalized objective value.

$$(f'_1)^{P_1} + (f'_2)^{P_2} + \dots + (f'_M)^{P_M} = 1 \quad (6)$$

In each cluster, it hoped to find a set of  $P_1, P_2, \dots, P_M$ , so that their normalized objective vectors satisfy Eq. (6) as much as possible. After obtaining the estimated PF, LMPFE evaluates the projection points of solutions on the estimated PF and uses Eq. (7) to measure the convergence of solutions, where  $\mathbf{d}$  represents the projection point of  $\mathbf{x}$  on the estimated PF.

$$con(\mathbf{x}) = \max_{i=1,\dots,M} |f'_i(\mathbf{x}) - d_i| \quad (7)$$

Diversity is calculated using Eq. (8), where  $\mathbf{d}'$  represents the other projection point closest to  $\mathbf{d}$ .

$$div(\mathbf{x}) = \max_{i=1,\dots,M} |d_i - d'_i| \quad (8)$$

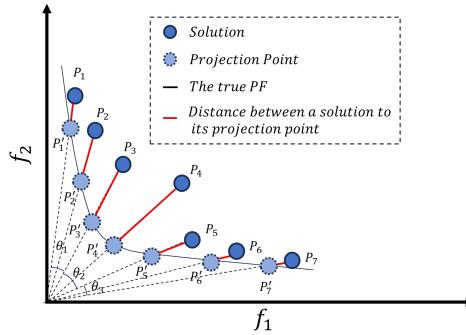
Finally, Eq. (9) is used as the fitness of  $\mathbf{x}$ , where  $\theta$  is determined using an adaptive mechanism.

$$fit(\mathbf{x}) = \theta \times div(\mathbf{x}) + (1 - \theta) \times con(\mathbf{x}) \quad (9)$$

### 2.3 Motivation

If we know the true PF, we can evaluate the projection point of a solution on the PF. The distance between the projection points of solutions will be used to measure diversity, and the distance between a solution and its projection point will be used to measure the convergence of that solution [25]. As shown in Fig. 1, when we need to abandon the most crowded solution, if the diversity of a solution is measured based on the sum of the angle distance or the Euclidean distance between the solution and its two closest solutions,  $P_2$  or  $P_6$  will be considered the most crowded solution, and then  $P_2$  or  $P_6$  will be abandoned.

However, if the Euclidean distance between the projection points of solutions is used to measure the diversity of solutions, then  $P_4$  will be considered the most crowded solution. Additionally,  $P_4$  is also the solution with the worst convergence compared to  $P_3$ ,  $P_5$ . For these reasons,  $P_4$  will be abandoned. Evaluating the diversity of solutions based on their projection points is similar to evaluating the diversity of the reference vectors based on their intersection points on the PF.



**Fig. 1:** Measure the convergence and diversity of solutions based on their projection points.

The true PF is unknown before optimization. Fortunately, as the solutions of the population converge, they continue to approach the true PF. Therefore, it is feasible to use the Pareto nondominated solutions to approximate the PF when they are close enough to the PF. In the early stages, a solution with poor convergence may be retained due to being away from other solutions with better convergence [36], which is unfavorable for convergence. Therefore, DLEA [37] and TS-NSGA-II [38] focus on convergence in the first stage and then on diversity in the second stage. To obtain nondominated solutions close to the true PF and avoid retaining solutions with poor convergence, in the early stages, we hope to maintain the diversity of the population while allowing it to converge as soon as possible. This approach helps us quickly obtain an estimated PF close to the true PF, thereby more efficiently guiding the selection of solutions.

Decomposition-based algorithms are easier to determine convergence compared to Pareto dominance-based algorithms [22], but the distribution of the reference vectors

needs to conform to the PF. Therefore, we need to adjust the distribution of the reference vectors based on the Pareto nondominated solutions. If the reference vectors used to approximate the PF changes dramatically, a solution with good convergence may be abandoned due to the disappearance of its associated reference vector, which is unfavorable for convergence [34, 39].

Our basic idea is using GNG to gradually learn the distribution of current Pareto nondominated solutions in the objective space, and then obtain the corresponding reference vectors to guide the population to converge as quickly as possible while maintaining diversity. When the Pareto nondominated solutions are close enough to the true PF, we use them to approximate the PF. Next, we evaluate the projection points of solutions on the estimated PF [25], which can further distinguish the convergence and diversity of the solutions for better selection.

### 3 The Proposed Algorithm

This section first introduces the framework of TS-MOEA-EPF, then provides a detailed introduction to its main components, and finally analyzes the computational complexity of TS-MOEA-EPF.

#### 3.1 Framework

The process of the proposed TS-MOEA-EPF is shown in Algorithm 1. Firstly, a population  $P$  containing  $N$  solutions is randomly generated. Then, the archive  $A$ , GNG, and  $\mathbf{z}^{min}$  (Line 4-6) will be initialized.  $\mathbf{z}^{min}$  is composed of the minimum values of each objective encountered during the optimization process.

In the first stage, we use the binary tournament selection to select a mating pool based on the dominance level and use the solutions in the mating pool as parents to generate offspring  $Q$ .  $A$  and  $Q$  will be merged into  $P$ , and the dominance level of the solutions in  $P$  will be calculated using  $\epsilon$ -dominance [11] (line 8-13). The  $i$ -th value of  $\epsilon'$  is the smaller value between  $\epsilon \cdot (z_i^{max} - z_i^{min})$  and  $\epsilon$ , as shown in Eq. (10), where  $z_i^{max}$  is the maximum value of the  $i$ -th objective of the current nondominated solutions. We adopt  $\epsilon$ -dominance to identify outliers that are only slightly better than other solutions on a few objectives but worse on most objectives, as dominated solutions.

$$\epsilon'_i = \epsilon \times \min(1, (z_i^{max} - z_i^{min})) \quad (10)$$

The Pareto nondominated solutions of  $P$  will be used to update  $A$ , and then the objective vectors of  $A$  will be used to update the GNG (Line 14-17), allowing the GNG to consider the newly generated solutions [33]. How to use the GNG to guide the selection of solutions will be explained in 3.4.

After the Pareto nondominated solutions of  $P$  are close enough to the true PF, we use a neural network to approximate the PF (Line 20-22) based on these Pareto nondominated solutions. Firstly, we normalize these solutions using Eq. (11) to obtain  $A'$ .

$$f'_i(\mathbf{x}) = \frac{f_i(\mathbf{x}) - z_i^{min}}{z_i^{max} - z_i^{min}} \quad (11)$$

---

**Algorithm 1** Framework.

---

**Input:**  $N$  (Population size),  $FEs_{max}$  (Maximum number of function evaluations),  $\epsilon$  ( $\epsilon$ -dominance),  $\alpha$  (The proportion of the first stage)

**Output:**  $P$  (a set of solutions)

```

1:  $FEs \leftarrow 0;$ 
2:  $P \leftarrow$  Randomly generate  $N$  solutions;
3:  $R \leftarrow$  Generate  $N$  uniform reference vectors on the hyperplane;
4:  $A \leftarrow$  The Pareto nondominated solutions in  $P$ ;
5:  $GNG \leftarrow$  Initialize the GNG;
6:  $z^{min} \leftarrow$  IdealPoint( $P$ );
7: while  $FEs < \alpha \times FEs_{max}$  do
8:    $Q \leftarrow$  GenerateOffspring( $P$ );
9:    $FEs \leftarrow FEs + N$ ;
10:   $z^{min} \leftarrow$  UpdateIdealPoint( $Q$ ,  $z^{min}$ );
11:   $P \leftarrow A \cup Q \cup P$ ;
12:  Determine  $\epsilon'$  according to Eq. (10);
13:   $Fno \leftarrow$  NDSortByEpsilonion( $P$ ,  $\epsilon'$ );
14:   $A \leftarrow$  The Pareto nondominated solutions of  $P$ ;
15:   $A \leftarrow$  ArchiveUpdate( $A$ ,  $2 \times N$ );
16:   $Objs \leftarrow$  The objective vectors of  $A$ ;
17:   $GNG \leftarrow$  GNGUpdate( $GNG$ ,  $Objs$ );
18:   $P \leftarrow$  Eselection( $P$ ,  $N$ ,  $GNG$ ,  $R$ ,  $Fno$ );
19: end while
20:  $A \leftarrow$  The Pareto nondominated solutions of  $P$ ;
21:  $A' \leftarrow$  Normalization( $A$ );
22:  $front \leftarrow$  Using a neural network to get the estimated PF based on  $A'$ ;
23: while  $FEs < FEs_{max}$  do
24:    $P \leftarrow$  ESelectionOneByOne( $P$ ,  $N$ ,  $front$ ,  $\epsilon$ );
25:    $FEs \leftarrow FEs + N$ ;
26: end while
27: return  $P$ ;

```

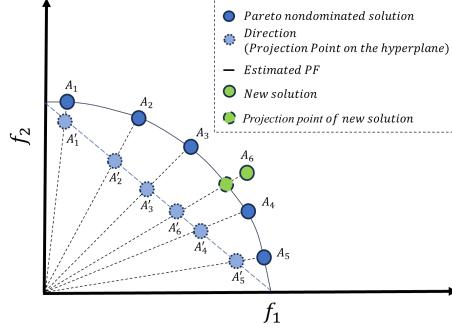
---

Because there is at most one intersection point with the true PF in a direction, this inspires us to use direction as input and find the mapping relationship between each direction and its intersection point on the PF. The training examples of the neural network are extracted from the current Pareto nondominated solutions. The input and target value of an example are computed according to Eq. (12) and Eq. (13). The input data are the directions of current Pareto nondominated solutions (their projection points on the hyperplane). Finally, we use  $input(\mathbf{x}) \cdot target(\mathbf{x})$  to get the projection point of  $\mathbf{x}$  on the estimated PF.

$$input(\mathbf{x}) = \frac{\mathbf{f}'(\mathbf{x})}{\sum_{i=1}^M f'_i(\mathbf{x})} \quad (12)$$

$$target(\mathbf{x}) = \sum_{i=1}^M f'_i(\mathbf{x}) \quad (13)$$

As shown in Fig. 2, the goal of the neural network is to map these projection points of  $A'$  on the hyperplane back to  $A'$  and evaluate the projection point of a solution with any direction, such as  $A_6$ , on the estimated PF. Next, we enter the second stage guided by the estimated PF (Line 23-26).



**Fig. 2:** Use  $A_1$  to  $A_5$  to obtain the estimated PF and evaluate the projection point of new solution  $A_6$  on the estimated PF.

### 3.2 Update Archive

The pseudocode of updating the archive is shown in Algorithm 2. If the size of the archive is greater than  $N_s$ , then it needs to be filtered. Firstly, we normalize the solutions in the archive using Eq. (11). Then, we identify and save the extreme points in the archive (Line 3). The extreme points are determined according to Algorithm 3. In Algorithm 3, we hope to preserve solutions that are close to the boundary, have big angles with each other, and contain the maximum values of some objectives.

Next, in Algorithm 2, we prioritize selecting the solutions that are farthest from the chosen solutions until  $N_s$  solutions are selected (Line 5-9), so that we can find  $N_s$  solutions that are very far from each other. If there are still solutions in  $A$ , then add each solution from  $A$  to  $A'$  one by one (Line 10-15). After this, we identify  $\mathbf{p}$  and  $\mathbf{q}$  with the closest angle to each other in  $A'$ . We use Eq. (14) to compare the advantages over neighbors ( $AN$ ) of these two solutions, where  $nb(\mathbf{x})$  are other  $M$  solutions closest to  $\mathbf{x}$  in  $A'$ , and  $dis(\mathbf{f}'(\mathbf{p}), \mathbf{f}'(\mathbf{x}))$  is the Euclidean distance between  $\mathbf{p}$  and  $\mathbf{x}$  in the objective space. The solution with smaller  $AN$  value will be removed from  $A'$  (Line 13-14).

$$AN(\mathbf{x}) = \sum_{\mathbf{p} \in nb(\mathbf{x})} \max_{i=1, \dots, M} (f'_i(\mathbf{p}) - f'_i(\mathbf{x})) \cdot dis(\mathbf{f}'(\mathbf{p}), \mathbf{f}'(\mathbf{x})) \quad (14)$$

The  $AN$  value can simultaneously reflect the convergence and diversity of a solution compared to its surrounding solutions. We use Fig. 3 to illustrate why the Euclidean distance between solutions is still considered in Eq. (14). The two solutions with the

smallest angle in Fig. 3 are  $A_2$  and  $A_3$ , we prioritize removing the solution with the smallest  $AN$  from these two solutions. If we do not consider the Euclidean distance and only consider the maximum difference in objectives between this solution and its two surrounding solutions, then  $AN(A_2) = 0.05 + 0.15 = 0.20$ , and  $AN(A_3) = 0.10 + 0.16 = 0.26$ , so  $A_2$  will be removed. However, it is clear that  $A_2$  should be saved. After introducing the Euclidean distance,  $AN(A_2) = 0.05 \cdot dis(A_1, A_2) + 0.15 \cdot dis(A_2, A_3) \approx 0.1150$ , while  $AN(A_3) = 0.10 \cdot dis(A_2, A_3) + 0.16 \cdot dis(A_3, A_4) \approx 0.1019$ . Therefore,  $A_3$  will be deleted while retaining  $A_2$ .

---

**Algorithm 2** ArchiveUpdate.

---

**Input:**  $A$  (Archive),  $N_s$  (Maximum size of archive),  
**Output:**  $A$  (Archive after update)

```

1: if  $|A| \geq N_s$  then
2:    $A \leftarrow \text{Normalization}(A)$ 
3:    $A' \leftarrow \text{FindExtremePoints}(A);$ 
4:    $A \leftarrow A \setminus A';$ 
5:   while  $|A'| < N_s$  do
6:      $\mathbf{r} \leftarrow \arg \max_{\mathbf{x} \in A} \min_{\mathbf{p} \in A'} \angle \mathbf{f}'(\mathbf{p}), \mathbf{f}'(\mathbf{x});$ 
7:      $A' \leftarrow A' \cup \mathbf{r};$ 
8:      $A \leftarrow A \setminus \mathbf{r};$ 
9:   end while
10:  for  $\mathbf{q} \in A$  do
11:     $A' \leftarrow A' \cup \mathbf{q};$ 
12:     $\mathbf{p}, \mathbf{q} \leftarrow \arg \min_{\mathbf{x}, \mathbf{y} \in A', \mathbf{x} \neq \mathbf{y}} \angle \mathbf{f}'(\mathbf{x}), \mathbf{f}'(\mathbf{y});$ 
13:     $\mathbf{r} \leftarrow \arg \min_{\mathbf{x} \in \{\mathbf{p}, \mathbf{q}\}} AN(\mathbf{x});$ 
14:     $A' \leftarrow A' \setminus \mathbf{r};$ 
15:  end for
16:   $A \leftarrow A';$ 
17: end if
18: return  $A;$ 

```

---

---

**Algorithm 3** FindExtremePoints.

---

**Input:**  $P$  (Pareto nondominated solutions),  $M$  (Number of objectives)

**Output:**  $EP$  (Extreme Points)

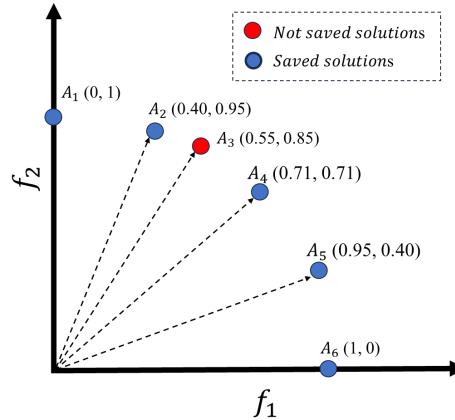
```

1:  $EP \leftarrow \emptyset$ ;
2:  $EM \leftarrow \{(1, 1, \dots, 1)\}$ ;
3:  $minDis(\mathbf{x}) \leftarrow \min_{\mathbf{p} \in EM} (1 - \cos(\angle \mathbf{p}, \mathbf{f}'(\mathbf{x}))) \cdot \max_{i=1, \dots, M} (f'_i(\mathbf{x}))$ , for each  $\mathbf{x} \in P$ ;
4: while  $|EP| < M$  And  $|P| > 0$  do
5:    $\mathbf{q} \leftarrow \arg \max_{\mathbf{x} \in P} minDis(\mathbf{x})$ 
6:    $EM \leftarrow EM \cup \mathbf{f}'(\mathbf{q})$ ;
7:    $EP \leftarrow EP \cup \mathbf{q}$ ;
8:    $P \leftarrow P \setminus \mathbf{q}$ ;
9:    $minDis \leftarrow \text{Update } minDis$ ;
10: end while
11: return  $EP$ ;

```

---

We use the angle distance to select the two solutions that need to be compared, allowing outliers with poor convergence but far from other solutions to participate in comparison and be filtered out. However, this method may be difficult to deal with problems which PF is convex. We will consider this issue in the update of GNG.



**Fig. 3:** By considering both the convergence and diversity of the above solutions  $A_1$  to  $A_6$  through AN,  $A_3$  will be removed.

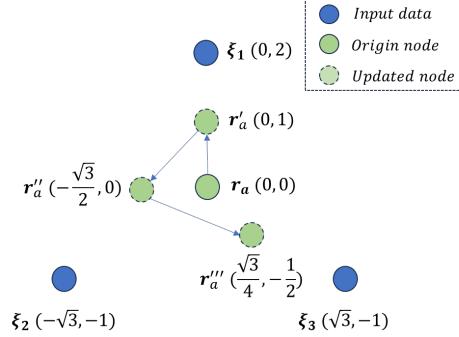
### 3.3 Update GNG network

The Eq. (4) cannot allow the nodes of the GNG to reflect the distribution of their surrounding data points accurately. As shown in Fig. 4, even if  $\mathbf{r}_a$  is already at the center of  $\xi_1$ ,  $\xi_2$  and  $\xi_3$ , it still walks between these three data points. We adopt Eq.

(15) for the update of  $\mathbf{r}_a$ , while Eq. (16) is used for the update of adjacent  $\mathbf{r}_c$  that has an edge connected to  $\mathbf{r}_a$ .

$$\mathbf{r}_a = \frac{n_{\mathbf{r}_a} \cdot \mathbf{r}_a + \epsilon_a \cdot \boldsymbol{\xi} + (1 - \epsilon_a)\mathbf{r}_a^o}{n_{\mathbf{r}_a} + 1} \quad (15)$$

$$\mathbf{r}_c = \frac{n_{\mathbf{r}_c} \cdot \mathbf{r}_c + \epsilon_{nb} \cdot \boldsymbol{\xi} + (1 - \epsilon_{nb})\mathbf{r}_c^o}{n_{\mathbf{r}_c} + 1} \quad (16)$$



**Fig. 4:** For clarity, we set  $\epsilon_a$  to 0.5,  $\mathbf{r}_a$  first approaches  $\boldsymbol{\xi}_1$ , then approaches  $\boldsymbol{\xi}_2$ , and finally approaches  $\boldsymbol{\xi}_3$ , the updated node  $\mathbf{r}_a'''$  tends to be closer to the last data point  $\boldsymbol{\xi}_3$ .

Where  $\mathbf{r}_a^o$  is the location of  $\mathbf{r}_a$  before the GNG update, and the same applies to  $\mathbf{r}_c^o$ ,  $n_{\mathbf{r}_a}$  and  $n_{\mathbf{r}_c}$  are the current number of updates for  $\mathbf{r}_a$  and  $\mathbf{r}_c$ , respectively, which are initially 0 and will increase by 1 after each update. When  $\epsilon_a$  and  $\epsilon_{nb}$  are set to 1, a node will be updated to the center of its surrounding data points in one iteration update.

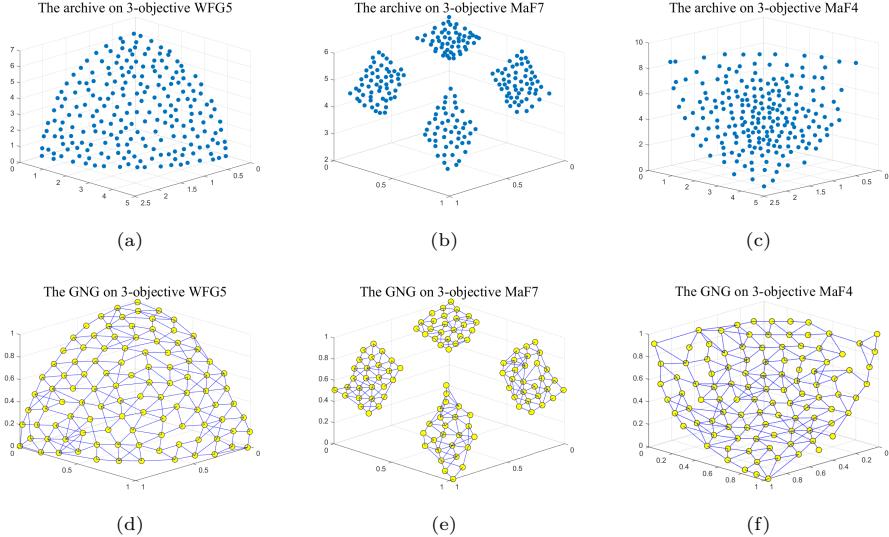
We have modified the traditional definition of the second nearest node  $\mathbf{r}_b$  as shown in Eq. (17), where  $nb(\boldsymbol{\xi})$  represents the set of  $M$  nodes closest to  $\boldsymbol{\xi}$ , excluding  $\mathbf{r}_a$ . Here,  $\theta$  is  $\angle(\mathbf{r} - \mathbf{r}_a, \boldsymbol{\xi} - \mathbf{r}_a)$ .

$$\mathbf{r}_b = \arg \min_{\mathbf{r} \in nb(\boldsymbol{\xi})} \|\mathbf{r} - \boldsymbol{\xi}\|^2 \cdot (1 - \cos(\theta)) \quad (17)$$

We use Eq. (18) to update the *error* and bind it to the edge between  $\mathbf{r}_a$  and  $\mathbf{r}_b$ , where  $d_1$  is  $\|\mathbf{r}_a - \boldsymbol{\xi}\|^2$  and  $d_2$  is  $\|\mathbf{r}_b - \boldsymbol{\xi}\|^2$ .

$$error(\mathbf{r}_a, \mathbf{r}_b) = error(\mathbf{r}_a, \mathbf{r}_b) + d_1 + d_2 \quad (18)$$

By using the defined  $\mathbf{r}_b$ , we can more accurately record the location of the data point  $\boldsymbol{\xi}$  through the edge  $\mathbf{r}_a - \mathbf{r}_b$ . If there are a large number of data points near this edge, it will accumulate greater error for this edge. Therefore, when a new node is needed, it is more important to generate a new node at the center of the edge with the maximum error.



**Fig. 5:** The archives obtained by TS-MOEA-EPF on 3-objective WFG5, 3-objective MaF7, and 3-objective MaF4 are shown in (a), (b), and (c), respectively, while the resulting GNG are shown in (d), (e), and (g).

Because the nodes in the GNG may tend to be closer to the center and farther away from the boundary [34], we use Eq. (19) to expand the nodes in the GNG, where  $\gamma$  is the average Euclidean distance between every pair of nearest nodes in the GNG, and  $r_{ns}$  is the closest node of  $r$ .

$$r = r - 0.02 \cdot \frac{\gamma(r_{ns} - r)}{\|r_{ns} - r\|^2 + 10^{-6}} \quad (19)$$

As shown in Fig. 5, the modified GNG can cover the boundary of the PF and obtain a more accurate distribution, even in MaF7 with the disconnected PF. In the MaF4 problem with the convex PF, the archive tends to retain solutions close to the center, while the node expansion method extends the nodes outward to a certain extent. This allows the GNG to automatically adjust its nodes to approximate the convex PF even if the archive cannot approximate the convex PF well. Consequently, population selection guided by the GNG can also approximate the convex PF well.

### 3.4 Environmental Selection in the First Stage

The selection algorithm for the first stage is shown in Algorithm 4. The extended nodes in the GNG are represented by  $NodeS$ . When  $|NodeS|$  is less than  $N$ , we select the  $N-|NodeS|$  reference vectors with the maximum angle distance to the nearest one of  $NodeS$  from  $R$ , and add them together with  $NodeS$  to form the reference vectors used for guiding the selection of solutions (Line 5).

We associate these solutions with the reference vectors that have the smallest angle to them, where  $C(\mathbf{x}) = i$ , indicating that  $\mathbf{x}$  is associated with the  $i$ -th reference vector (Line 7). Next, the *fitness* of a solution is calculated based on APD [26]. For  $P(\theta_{t,\mathbf{x},i})$ , we modify it as Eq. (21), where  $\theta_{t,\mathbf{x},i}$  is the angle between  $\mathbf{x}$  and the  $i$ -th reference vector,  $\gamma_{t,i}$  is the average angle of the  $M$  other reference vectors closest to the  $i$ -th reference vector,  $t$  is the current generation, and  $t_{max}$  is the maximum number of generations. We set the *fitness* of the extreme points to 0.

$$d_{t,\mathbf{x},i} = (1 + P(\theta_{t,\mathbf{x},i})) \cdot \|f'(\mathbf{x})\|^2 \quad (20)$$

$$P(\theta_{t,\mathbf{x},i}) = \left( \frac{t}{t_{max}} \right) \cdot \frac{\theta_{t,\mathbf{x},i}}{\gamma_{t,i}} \quad (21)$$

For the solutions associated with the same reference vector, we obtain their *FitRank* based on the increasing order of their *fitness*. Finally, we select the  $N - |P'|$  solutions with the smallest *FitRank* in  $F_l$  to join  $P'$  (Line 12).

---

**Algorithm 4** Eselection.

---

**Input:**  $P$  (Population),  $FNo$  (Dominance level of population),  $N$  (Population size),  $GNG$  (GNG network),  $R$  (Default reference vectors)

**Output:**  $P$  (Population)

- 1: Divide  $P$  into  $\{F_1, \dots, F_i, \dots\}$  where  $F_i$  is composed of the solutions which  $FNo$  are  $i$ ;
  - 2:  $P \leftarrow F_1 \cup \dots \cup F_l$  where  $|F_1 \cup \dots \cup F_{l-1}| \leq N$  and  $|F_1 \cup \dots \cup F_l| > N$  ;
  - 3:  $P \leftarrow \text{Normalization}(P)$ ;
  - 4:  $NodeS \leftarrow$  The expanded nodes of  $GNG$ ;
  - 5:  $NodeS \leftarrow NodeS \cup$  the  $N - |NodeS|$  vectors of  $R$  which are farthest from  $NodeS$ ;
  - 6:  $P' \leftarrow F_1 \cup \dots \cup F_{l-1} \cup$  the extreme points;
  - 7:  $C \leftarrow$  Assign the solutions of  $P$  to the reference vectors with the smallest angle to them;
  - 8:  $fitness \leftarrow$  Compute the fitness of  $P$  according to Eq. (20);
  - 9:  $FitRank \leftarrow$  Assign each solution a rank number according to the ascending order of their *fitness* in their cluster.
  - 10:  $P \leftarrow P \setminus P'$ ;
  - 11:  $P \leftarrow$  Sort  $P$  according to the ascend order of *FitRank*;
  - 12:  $P' \leftarrow P' \cup$  the first  $N - |P'|$  solutions of  $P$ ;
  - 13:  $P \leftarrow P'$ ;
  - 14: **return**  $P$ ;
- 

### 3.5 Environmental Selection in the Second Stage

After obtaining the estimated PF, we evaluate the projection point  $pp(\mathbf{x})$  of a solution  $\mathbf{x}$  on the estimated PF. Next, the convergence of the solution is calculated as shown

in Eq. (22).

$$con(\mathbf{x}) = \frac{1.1 \times \sum_{i=1}^M pp_i(\mathbf{x}) - \sum_{i=1}^M f'_i(\mathbf{x})}{1.1 \times \sum_{i=1}^M pp_i(\mathbf{x})} \quad (22)$$

The diversity is shown in Eq. (23), where  $\mathbf{pp}'$  represents the other projection point which is second closest to  $\mathbf{pp}(\mathbf{x})$ , and  $dis(\mathbf{pp}(\mathbf{x}), \mathbf{pp}')$  is used to calculate the Euclidean distance between  $\mathbf{pp}(\mathbf{x})$  and  $\mathbf{pp}'$ .

$$div(\mathbf{x}) = dis(\mathbf{pp}(\mathbf{x}), \mathbf{pp}') \quad (23)$$

In order to avoid the distribution differences of  $div$  and  $con$  causing excessive preference for convergence or diversity, we use Eq. (24) to scale  $div$  and  $con$  respectively.  $\max(val)$  represents the maximum value among all  $val$ , and  $avg(val)$  represents the average value of all  $val$ .

$$val' = \frac{val + S \cdot avg(val)}{\max(val) + S \cdot avg(val)} \quad (24)$$

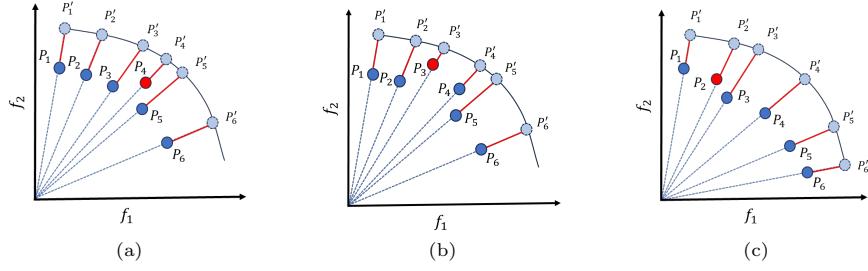
$S$  is defined as Eq. (25), and  $c$  ( $c < 1$ ) is the parameter that controls the difference in the scaled value, ensuring that the ratio of the scaled average value to the scaled maximum value will not be smaller than  $c$ . For example, if we scale  $cov$  now and  $c_{cov} = 0.8$ , with  $\max(cov) = 1$  and  $avg(cov) = 0.5$ . Based on Eq. (25), we obtain that  $S_{cov}$  is 3. Therefore,  $cov' = (cov + 3 \cdot avg(cov)) / (\max(cov) + 3 \cdot avg(cov))$ , and then  $avg(cov') / \max(cov') = (0.5 + 1.5) / (1 + 1.5) = 0.8$ .

$$S = \lceil \frac{c \cdot \max(val) - avg(val)}{avg(val) \cdot (1 - c)} \rceil \quad (25)$$

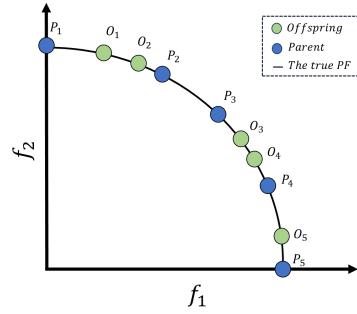
Finally, the fitness of the solution  $\mathbf{x}$  is calculated using Eq. (26), where  $cov'(\mathbf{x})$  is the scaled value of  $cov(\mathbf{x})$ , and the same applies to  $div'(\mathbf{x})$ .

$$fit(\mathbf{x}) = cov'(\mathbf{x}) + div'(\mathbf{x}) \quad (26)$$

In Algorithm 5, a total of  $N$  offspring will be generated. Each generated offspring will be integrated into the current population, and the dominance level will be updated accordingly. The dominance level of the solutions is calculated using  $\epsilon$ -dominance. Afterwards, if there is only one dominated solution in the last layer, then abandon that solution directly as shown in Fig. 6a. If there are more than one dominated solutions in the last layer, then we will discard the one with poorer  $cov'$  from the solutions in the last layer as shown in Fig. 6b. On the contrary, when there are no dominated solutions, find the two solutions which projection points are closest, and abandon the one with the smaller  $fit$  as shown in Fig. 6c.



**Fig. 6:** Illustration of ESelectionOneByOne, we choose 5 solutions from 6 solutions. The red solution is the abandoned solution, and  $P'_i$  is the projection point of the solution  $P_i$  on the estimated PF. (a) Directly abandon the only dominated solution in the last layer. (b) Abandon the solution with the worst convergence from the dominated solutions in the last layer. (c) Abandon the solution with the smaller  $fit$  value among the two solutions which projection points are closest.



**Fig. 7:** The order in which offspring appear is from  $O_1$  to  $O_5$ . For the sake of discussion, we assume that  $P_1$  to  $P_5$  and  $O_1$  to  $O_5$  all converge to the true PF. Consequently, their projection points can be regarded as themselves, and their  $cov'$  are all the same. Therefore, the difference in  $fit$  mainly comes from  $div'$ .

We immediately screen the population and the newly generated offspring to avoid the impact of a large number of randomly distributed offspring on measuring the diversity of solutions. In Fig. 7, we would like to retain  $P_1, O_2, P_3, P_4$  and  $P_5$ . If we perform the selection after generating all offspring, then in the comparison between  $P_2$  and  $O_2$  (their projection points are closest), the second closest solution to  $P_2$  is  $O_1$ , the second closest solution to  $O_2$  is  $O_1$ , and  $dis(P_2, O_1)$  is greater than  $dis(O_1, O_2)$ . Thus, we would retain  $P_2$  and abandon  $O_2$ . However, if a new offspring is generated and immediately participates in the screening, we first compare  $P_1$  with  $O_1$ , and  $O_1$  will be abandoned due to  $dis(P_2, O_1) < dis(P_1, P_2)$ . In the comparison between  $P_2$  and  $O_2$ ,  $P_2$  will be abandoned due to  $dis(P_2, O_2) < dis(P_1, O_2)$ . Similarly,  $O_3$  will be abandoned due to  $dis(P_4, O_3) < dis(P_3, P_4)$ ,  $O_4$  will be abandoned due to  $dis(O_4, P_3) < dis(P_4, P_3)$ , and finally,  $O_5$  will be abandoned due to  $dis(O_5, P_4) < dis(P_5, P_4)$ . In the end, we will obtain the expected population of  $P_1, O_2, P_3, P_4$  and  $P_5$ .

Additionally, we use the two solutions whose projection points are closest for comparison because their projection points are similar, making the comparison of their convergence and diversity more fair.

---

**Algorithm 5** ESelectionOneByOne.

---

**Input:**  $P$  (Population),  $N$  (Population size),  $Front$  (Estimated Pareto front),  $\epsilon$  ( $\epsilon$ -dominance)

**Output:**  $P$  (Population)

```

1:  $pp \leftarrow$  The projection points of  $P$  in the estimated PF;
2:  $\epsilon' \leftarrow$  Determined  $\epsilon'$  according to Eq. (10);
3:  $FNo \leftarrow$  NDSortByEplision( $P$ ,  $\epsilon'$ );
4:  $times \leftarrow 0$ ;
5: while  $times \leq N$  do
6:    $offspring \leftarrow$  Generate a new solution based on  $P$ ;
7:    $times \leftarrow times + 1$ ;
8:    $P \leftarrow P \cup offspring$ ;
9:    $FNo \leftarrow$  Update the  $FNo$  of  $P$ ;
10:  Divide  $P$  into  $\{F_1, \dots, F_l\}$  where  $F_i$  is composed of the solutions which  $FNo$  are  $i$ ;
11:   $cov' \leftarrow$  The scaled  $cov$  value of  $P$  ;
12:   $div' \leftarrow$  The scaled  $div$  value of  $P$ ;
13:   $EP \leftarrow$  FindExtremePoints( $P$ );
14:  if  $l > 1$  then
15:    if  $|F_l| = 1$  then
16:       $P \leftarrow F_1 \cup \dots \cup F_{l-1}$ ;
17:    else
18:       $r \leftarrow \arg \min_{x \in F_l} cov'(x)$ ;
19:       $P \leftarrow F_1 \cup \dots \cup F_l \setminus r$ ;
20:    end if
21:  else
22:     $p, q \leftarrow \arg \min_{x, y \in P, x \neq y} dis(pp(x), pp(y))$ ;
23:    if  $EP \cap q = \emptyset$  and  $EP \cap p = \emptyset$  then
24:       $r \leftarrow \arg \min_{x \in \{p, q\}} fit(x)$ ;
25:       $P \leftarrow F_1 \setminus r$ ;
26:    else if  $EP \cap p = \emptyset$  then
27:       $P \leftarrow F_1 \setminus p$ ;
28:    else
29:       $P \leftarrow F_1 \setminus q$ ;
30:    end if
31:  end if
32: end while
33: return  $P$ ;

```

---

### 3.6 Computational Complexity of TS-MOEA-EPF

In the first stage of TS-MOEA-EPF, we need to calculate the dominance level of all solutions. Adopting the fast dominance method in NSGA-II will result in a complexity of  $O(MN\log(N))$ , where  $M$  is the number of objectives and  $N$  is the population size. The complexity of archive update is  $O(MN^2)$  because we need to calculate the angle distance and Euclidean distance between the solutions of the archive.

Updating the GNG involves finding the closest node to the current input data point. With a maximum of  $2N$  data points and  $N$  GNG nodes, the complexity of the update of the GNG is  $O(MN^2)$ . The selection algorithm in the first stage needs to associate the solutions with the reference vectors and calculate their APD values. With a maximum of  $N$  reference vectors and  $4N$  solutions, the complexity is at most  $O(MN^2)$ .

In the second stage, we generate a new solution and add it to the current population for selection. Since the dominance relationship of the parent is known, only the change in dominance level caused by the new solution needs to be considered. The complexity of updating the dominance level of the population is  $O(M\log(N))$ . Meanwhile, it is necessary to calculate the Euclidean distance between the projection point of the new solution and the projection points of the parent, requiring the complexity of  $O(MN)$ . The complexity of generating a solution and adding it to the population for selection is  $O(M\log(N) + MN)$ , which is repeated  $N$  times. Therefore, the complexity of the second stage is  $O(N(M\log(N) + MN))$ .

In summary, the complexity of both stages can be approximated as  $O(MN^2)$ , which is similar to the complexity of NSGA-II.

## 4 Experiments

This section introduces the problems selected for testing, the algorithms to be compared with TS-MOEA-EPF<sup>1</sup>, the performance indicator, and the parameter settings. All experiments were conducted on the PlatEMO [40] platform.

### 4.1 Test Problems

The multi-objective optimization problems chosen for testing include VNT1-VNT3 [41], MaF1-MaF9 [42] and WFG1-WFG9 [43]. VNT1-VNT3 will be tested on 3 objectives, MaF1-MaF9 and WFG1-WFG9 will be tested on 3, 10, and 15 objectives. For MaF1-MaF9, multi-objective optimization algorithms often obtain nondominated solutions that are far from the true PF in the early stages, which is not conducive to estimating the PF. Moreover, the PF of MaF1-MaF9 is more complex than that of WFG4-WFG9. The characteristics of these problems are shown in Table 1. VNT1-VNT3 have complex PF and the PF of VNT1-VNT2 are highly convex, and we hope to use these three problems to test whether TS-MOEA-EPF can effectively handle high convexity and complex PFs.

---

<sup>1</sup>TS-MOEA-EPF: <https://github.com/YCH-heihei/TS-MOEA-EPF>

**Table 1:** The characteristic of test problems

Problem	characteristic
VNT1	Partial, Highly convex
VNT2	Degenerated, Highly convex
VNT3	Degenerated
MaF1	Linear, Inversed
MaF2	Concave, Partial
MaF3	Convex, Multimodal
MaF4	Convex, Inverted, Multimodal, Badly-scaled
MaF5	Concave, Biased, Badly-scaled
MaF6	Concave, Degenerate
MaF7	Mixed, Disconnected, Multimodal
MaF8	Convex, Partial, Inversed
MaF9	Linear, Multimodal
WFG1	Mixed, Biased
WFG2	Convex, Disconnected, Nonseparable
WFG3	Linear, Degenerate, Nonseparable
WFG4	Concave, Multimodal
WFG5	Concave, Deceptive
WFG6	Concave, Nonseparable
WFG7	Concave, Biased
WFG8	Concave, Nonseparable, Biased
WFG9	Concave, Nonseparable, Deceptive, Biased

We represent the number of decision variables as  $D$ . For WFG test suite and MaF1-MaF6,  $D$  is  $M+9$ , where  $M$  is the number of objectives. For MaF7,  $D$  is  $M+19$ , while for MaF8, MaF9 and VNT1-VNT3,  $D$  is 2.

## 4.2 Compared algorithms

To verify the effectiveness of TS-MOEA-EPF, we selected CLIA [44], DEA-GNG [33], GFM-MOEA [19], PeEA [29], and LMPFE [25] as the comparative algorithms. CLIA uses support vector machine to predict the effectiveness of reference vectors, and prioritizes selecting reference vectors with better effectiveness from a large number of uniformly generated reference vectors to guide the selection of solutions. DEA-GNG learns the topological structure of the objective space of nondominated solutions to obtain reference vectors. PeEA estimates the convexity of the PF to set convergence metrics. GFM-MOEA and LMPFE use polynomial models to obtain the estimated PF, and the quality of a solution is measured based on its projection point on the estimated PF. TS-MOEA-EPF is similar to DEA-GNG in its first stage, while its second stage is similar to GFM-MOEA and LMPFE.

## 4.3 Parameter setting

For objectives of 3, 10, and 15, the corresponding population sizes are set to 105, 275 and 135, respectively. The maximum number of function evaluations  $FEs_{max}$  for each problem is shown in Table 2. Each algorithm will be independently run 30 times.

**Table 2:** The  $FEs_{max}$  of each test problems

Problem	3	10	15
MaF1	20,000	100,000	150,000
MaF2	20,000	70,000	100,000
MaF3	60,000	12,0000	150,000
MaF4	60,000	150,000	200,000
MaF5	10,000	20,000	40,000
MaF6	20,000	40,000	60,000
MaF7	30,000	70,000	100,000
MaF8	60,000	80,000	100,000
MaF9	60000	60,000	100,000
WFG1-3	52,500	275,000	202,500
WFG4-9	31,500	137,500	135,000
VNT1-3	20000	—	—

For all algorithms, we use the simulated binary crossover (SBX) [45] and polynomial mutation (PM) [46] to generate offspring, with a crossover probability of 1 and a mutation probability of  $1/D$ , where  $D$  is the number of decision variables, and the distribution indexes of SBX and PM are both 20. For reference vectors sampled on the hyperplane [33], when the number of objectives is less than 8, the systematic method [47] is used to generate a set of uniformly distributed reference vectors. Otherwise, the two-layer approach [48] is employed.

We use IGD and HV as metrics to measure the performance of these algorithms. The number of sampling points for IGD is 10000. The reference point for HV is set to (1.1, ..., 1.1). For calculating HV, we normalize the population using the nadir point and the ideal point of the PF, and then calculate the HV value of the population. When  $M$  is greater than or equal to 10, we estimate the HV value using Monte Carlo estimation [6]. For the IGD indicator, a smaller IGD value indicates that the population better approximates the PF. A larger HV value indicates that the population can Pareto dominate more solutions in the objective space. We use Wilcoxon's test at the 0.05 level to analyze the results. +, -, and  $\approx$  respectively indicate that the algorithm has better, worse, and similar performance compared to TS-MOEA-EPF.

All parameters of each algorithm comply with the settings in their original paper. For TS-MOEA-EPF,  $age_{max}$  of the GNG is set to  $10M$ , where  $M$  is the number of objectives.  $HP_{max}$  is set to  $2N$ , where  $N$  is the population size.  $\epsilon_a$  is set to 0.3 and  $\epsilon_{nb}$  is set to 0.03.  $\alpha$  is set to 0.5 like DEA-GNG,  $\delta$  is set to 0.99.

The neural network structure used by TS-MOEA-EPF for estimating the PF comprises three hidden layer with 20, 40, and 30 units in each hidden layer. The activation function of the three hidden layers is  $tansig(x) = 2/(1 + e^{-2x}) - 1$ . The loss function is the mean square error (MSE).

#### 4.4 Parameter Analysis

For simplicity, the  $M$ -objective WFG1 is denoted as WFG1( $M$ ). To measure the impact of parameters  $c_{cov}$  and  $c_{div}$  on TS-MOEA-EPF, we conducted tests on

WFG2(3), MaF9(3), WFG7(10), MaF2(10), MaF8(10), and MaF5(15). The extreme points of these problems are not easily misjudged due to the presence of outliers, and the impact of  $\epsilon$  is not significant. Thus,  $\epsilon$  is set to  $5 \times 10^{-4}$ . The corresponding results are shown in Table 3. From Table 3, we can see that the algorithm is not very sensitive to  $c_{cov}$  and  $c_{cov}$ . Based on the Friedman test at the 0.05 level [49] and the average ranking, we set  $c_{div}$  and  $c_{cov}$  to 0.60 and 0.80, respectively.

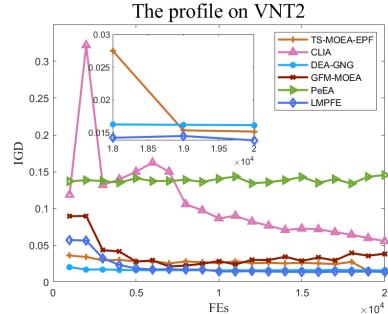
For the experiment with  $\epsilon$ , we set  $c_{div}$  to 0.60 and  $c_{cov}$  to 0.80. The test will be conducted on VNT(3), WFG2(3), MaF3(10), MaF4(10), MaF7(10), and MaF9(15). The obtained IGD results are shown in Table 4, where it can be seen that  $\epsilon$  has a significant impact on problems such as MaF3, MaF4 and VNT2. For MaF3 and MaF4, the extreme points are often selected from the outliers, the  $\epsilon$ -dominance method can determine some outliers as dominated solutions to avoid mistakenly identifying outliers as extreme points. However, due to the high convexity of VNT2's PF, using  $\epsilon$ -dominance may erroneously determine two solutions on the true PF as dominance relationships. Thus,  $\epsilon$  should not be too large or too small. Based on the average ranking shown in Table 4, we set  $\epsilon$  to  $5 \times 10^{-4}$ .

## 4.5 Results and discussions

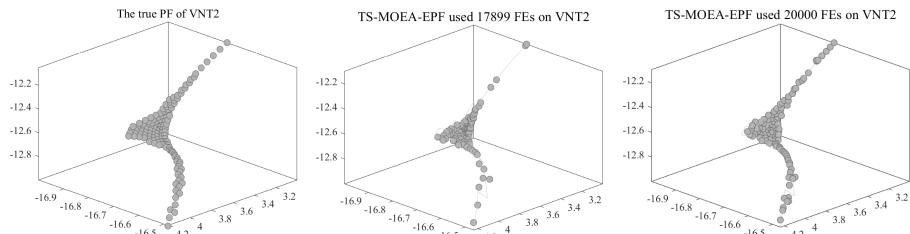
### 4.5.1 Analysis about VNT test problems

Table 5 and Table 6 present the results of each algorithm on VNT1-VNT3, highlighting the optimal values in gray and bold. All these PF are highly convex or have long tails. In the first stage of our algorithm, we use the angle distance to filter the solutions of the archive, which may not effectively retain enough solutions at the long tail. Even if the GNG expands, it may not sufficiently cover the long tail.

Fortunately, in the second stage, we adopted the Euclidean distance between the projection points for screening the solutions. This allowed the new solutions to fill the gaps at the long tail in a timely manner, making the results relatively comparable to other advanced algorithms, as show in Fig. 8 and Fig. 9. Overall, we achieved good results and verified that the neural networks can be used to fit complex PFs when the Pareto nondominated solutions can cover the true PF and converge well.



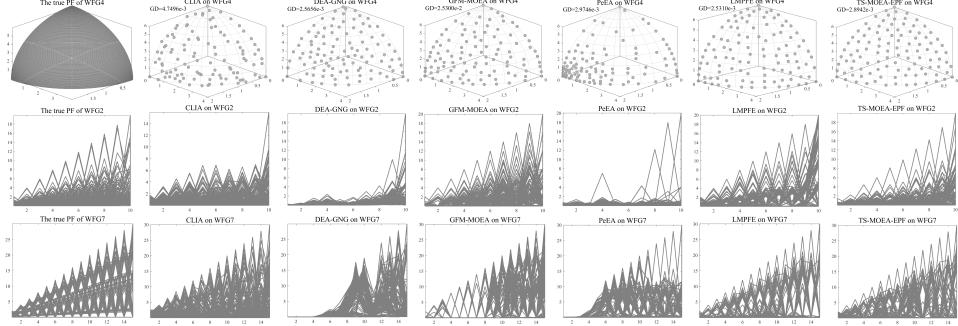
**Fig. 8:** Although filtering solutions based on angle distance may not well cover the long tail of VNT2's PF during the archive update, in the second stage, filtering solutions based on Euclidean distance can timely fill the gap in the tail of VNT2's PF and improve IGD.



**Fig. 9:** The second stage of TS-MOEA-EPF will cover the long tail of VNT2's PF that was not well covered in the first stage.

#### 4.5.2 Analysis about WFG test problems

Table 8 and Table 9 respectively show the IGD and HV values obtained by each algorithm on WFG1-WFG9. In terms of IGD, TS-MOEA-EPF achieved 11 optimal values out of 27 instances, while CLIA and GFM-MOEA achieved 5 and 7 optimal values respectively. Except for the WFG3 problem, TS-MOEA-EPF either achieves the optimal value or approaches it on the remaining problems. Although CLIA performs well in terms of IGD, its HV values are not satisfactory. This could be attributed to the population of CLIA tending to be closer to the center, resulting in insufficient boundary coverage. We measure the degree to which the population is far from the center on 15-objective WFG7 using Eq. (27) (boundary coverage). The population obtained by each algorithm on 15-objective WFG7 are shown in Fig. 10. Additionally, the obtained GD, Spread, and BC are shown in Table 7. From Table 7, we can see that although CLIA's Spread performs well on 15-objective WFG7, its insufficient



**Fig. 10:** The sets of nondominated solutions with the median IGD among 30 runs obtained by CLIA, DEA-GNG, GFM-MOEA, PeEA, LMPFE, and TS-MOEA-EPF on WFG4, WFG2 and WFG7.

convergence and boundary coverage result in poor HV.

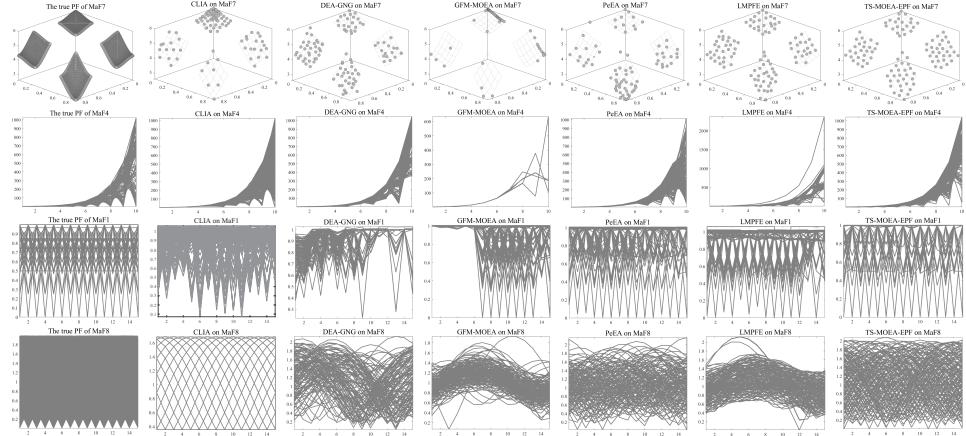
$$BC(P) = \sum_{\mathbf{p} \in P} \frac{\angle(\mathbf{p}, (1, 1, \dots, 1))}{|P|} \quad (27)$$

In terms of HV, LMPFE achieved 13 optimal values out of 27 test instances, GFM-MOEA obtained 9 optimal values out of 27 test instances. Although TS-MOEA-PF did not achieve the optimal value on HV, its results are relatively close to the optimal value, especially when the objective is 15.

On HV, the reason why LMPFE and GFM-MOEA outperform TS-MOEA-PF may be that they directly adopt an indicator that combines convergence and diversity to delete multiple solutions, while TS-MOEA-PF prioritizes screening from the two solutions whose projection points are closest. This could make TS-MOEA-EPF more focused on diversity compared to LMPFE and GFM-MOEA, while convergence is relatively slower. As shown in the result of 3-objective WFG4 in Fig. 10, the distribution of TS-MOEA-PF is more uniform compared to LMPFE and GFM-MOEA, but its GD value is slightly worse than the first two. Moreover, for WFG4-WFG9 with a regular PF, the polynomial model used by LMPFE and GFM-MOEA can easily obtain an accurately estimated PF in the early stages of the optimization progress, enabling them to screen the population earlier than TS-MOEA-PF based on the projection points on the accurately estimated PF.

However, the IGD and HV values of TS-MOEA-EPF remain comparable to those of LMPFE and GFM-MOEA on WFG4-WFG9. Meanwhile, we can also see that algorithms based on the projection points on the estimated PF can achieve better results compared to other algorithms when the estimated PF is closer to the true PF.

On the WFG2 problem, TS-MOEA-EPF achieved optimal IGD on both 3 and 15 objectives, while its IGD on 10 objectives was comparable to CLIA. At the same time, the HV of TS-MOEA-EPF is also very close to the optimal value obtained by LMPFE. This could be attributed to the fact that our modified GNG can accurately record the



**Fig. 11:** The sets of nondominated solutions with the median IGD among 30 runs obtained by CLIA, DEA-GNG, GFM-MOEA, PeEA, LMPFE, and TS-MOEA-EPF on MaF7, MaF4, MaF1 and MaF8.

current distribution of the nondominated solutions, and we obtained an estimated PF that is closer to the true PF in the later stages of optimization progress.

#### 4.5.3 Analysis about MaF test problems

Table 10 and Table 11 respectively show the IGD and HV values obtained by each algorithm on MaF1-MaF9. In terms of IGD, TS-MOEA-EPF achieved 14 optimal values out of these 27 instances, while PeEA achieved 5 optimal values out of these 27 instances. In terms of HV, TS-MOEA-EPF and PeEA achieved 9 and 7 optimal values out of these 27 instances respectively. Compared to its predecessors GFM-MOEA and LMPFE, TS-MOEA-EPF performs better on MaF1-MaF9.

This might be attributed to the significant difference between the estimated PF obtained by GFM-MOEA and LMPFE and the true PF on MaF1-MaF9, as well as the significant fluctuations of the estimated PF during the optimization process. This is not conducive to guiding the selection of solutions based on the estimated PF. The instances that can reflect the above assumption are MaF3, MaF4, and MaF8. The maximum values used for normalization on MaF3 and MaF4 are prone to change, which can cause changes in the relative positions of the normalized solutions. In the early stages of optimization, nondominated solutions are far from the true PF, leading to significant differences between the estimated PF and the true PF, this can easily mislead the true optimization.

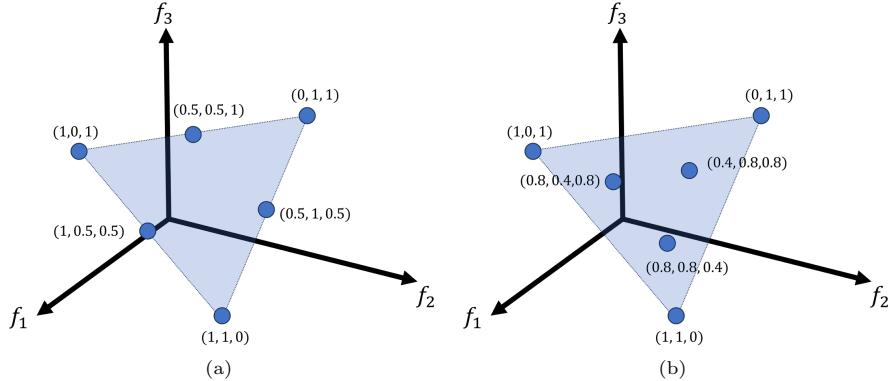
The extreme point determination method adopted by TS-MOEA-EPF selects extreme points in relatively constant directions, which may prevent the maximum value of Pareto nondominated solutions from changing too frequently. Additionally, TS-MOEA-EPF only obtains the estimated PF based on the population closer to the true PF in the later stages of optimization. This approach may help guide the selection

of solutions based on the estimated PF closer to the true PF, making the judgments on the diversity and convergence of solutions more accurate.

From Fig. 11, on 15-objective MaF8 problem, the distribution obtained by TS-MOEA-EPF outperforms the compared algorithms. The inferior performance of GFM-MOEA and LMPFE on the MaF8 problem may be attributed to their lower coverage of the entire PF compared to TS-MOEA-EPF. TS-MOEA-EPF can preserve solutions with good diversity and slightly poor convergence by expanding the nodes of the GNG, thereby increasing the diversity of the population. The progressive update method of the modified GNG also helps the population converge in the relatively constant directions. Additionally, TS-MOEA-EPF prioritizes comparing solutions from the two solutions which projection points are closest, this could significantly improve diversity while optimizing convergence. These factors may be the reason why the distribution of TS-MOEA-EPF on MaF8 is significantly better than that of the comparison algorithms.

On MaF8, the IGD value of TS-MOEA-EPF is better, but the HV value is slightly worse. We believe that for inverted convex PF, the solutions at the center will contribute more to HV compared to those near the boundary [19]. We continued to use GD, Spread, and BC to verify this assumption, and the results obtained are shown in Table 12. It can be seen that PeEA is slightly worse than TS-MOEA-EPF in both GD and Spread, but its BC value is smaller, which means PeEA is closer to the center on MaF8. This may be the reason why its HV value is better than TS-MOEA-EPE.

As shown in Fig. 11, it can be observed that the distribution of TS-MOEA-EPF on 15-objective MaF1 is not too poor, but its IGD and HV values are not very good. To understand what characteristics of the distribution lead TS-MOEA-EPF to perform worse than the compared algorithms in terms of HV and IGD on 10-objective and 15-objective MaF1, we will continue to use GD, Spread, and BC to further analyze the distribution of each algorithm on 15-objective MaF1. TS-MOEA-EPF has the highest BC value, indicating that its distribution is closest to the boundary. Hence, we can assume that the excessive coverage of the boundary of MaF1's PF may lead to less good IGD and HV results. An example is illustrated in Fig. 12.



**Fig. 12:** (a) These 6 solutions are closer to the boundary, with an IGD value of 2.4668e-1 and a HV value of 7.9638e-2. (b) These 6 solutions are closer to the center, with an IGD value of 1.9148e-1 and a HV value of 1.1044e-1. Both in terms of IGD and HV, the latter results are superior to the former.

For the MaF7 problem, the IGD value of TS-MOEA-EPF is optimal or close to optimal when the number of objectives is 3 and 10, but TS-MOEA-EPF is not so good when the number of objectives is 15. The APD we adopt involves the Euclidean distance from the normalized objective vector to the ideal point. However, emphasizing convergence in the early stages of optimization is not conducive to selecting solutions whose last objective is the smallest and the first  $M - 1$  objectives are all the maximum values of each objective. Therefore, the population obtained by TS-MOEA-EPF in the first stage did not cover the entire PF of MaF7 well when the number of objectives is more than 10. However, the performance on 3-objective MaF7 and 10-objective MaF7 can also verify the new method of judging the second nearest node  $r_b$  and the way of associating the cumulative errors with the edges, which is indeed feasible when dealing with the discontinuous PF.

On the MaF6 problem, the performance of TS-MOEA-EPF and LMPFE is not as good as other algorithms, as demonstrated in early experiments with WFG3. This may not be suitable for using the neural networks to fit the degenerated PF on many objectives.

## 5 CONCLUSION

To avoid misleading the selection of solutions based on an estimated PF that differs significantly from the true PF, we first use the modified GNG to ensure that the population maintains diversity while converging as quickly as possible, allowing Pareto nondominated solutions to approach the true PF quickly. When the nondominated solutions are close enough to the true PF, we use a neural network to estimate the true PF, and then remove solutions with poor convergence and diversity from the population and offspring base on their projection points on the estimated PF.

From the comparison results with other advanced algorithms, our algorithm is comparable to other algorithms on most problems. However, in many objectives, there is still room for improvement in using neural network to estimate the true PF. If we can determine that the true PF is concave, we can set the target of the neural network to the Euclidean distance between the normalized objective vectors and the ideal point. For linear cases, we can set the target of the neural network to the sum of each normalized objective. For convex cases, we can set the target of the neural network as the Euclidean distance from the normalized objective vectors to the nadir point. We hope to reduce the fluctuation of the target values through these methods and avoid overfitting when the input size is too small. Although neural networks can find the mapping relationship between the given input data and the target data, the obtained estimated PF may not necessarily satisfy the property that any projection point on the estimated PF cannot Pareto dominate other projection points, especially when the number of training samples is very small.

Additionally, in TS-MOEA-EPF, the switching time between the two stages is fixed. If we can determine that the population sufficiently cover the entire PF and is already close to the true PF, we can decide to adopt the estimated PF to guide the selection of solutions at this time.

**CRediT authorship contribution statement.** **Changhao Yang:** Writing - original draft, Methodology. **Jin Qin:** Writing - review & editing, Methodology, Funding acquisition.

**Data availability.** Data will be made available on request.

**Acknowledgements.** This work was supported by the National Natural Science Foundation of China under Grant No. 62162007.

## Declarations

**Conflicts of Interest:** The authors declare that there are no conflicts of interest regarding the publication of this paper.

## References

- [1] Ponsich, A., Jaimes, A.L., Coello, C.A.C.: A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications. *IEEE Transactions on Evolutionary Computation* **17**(3), 321–344 (2013) <https://doi.org/10.1109/TEVC.2012.2196800>
- [2] Zhang, Z., He, C., Ye, J., Xu, J., Pan, L.: Switching ripple suppressor design of the grid-connected inverters: A perspective of many-objective optimization with constraints handling. *Swarm and evolutionary computation* **44**, 293–303 (2019)
- [3] Coello, C.C.: Evolutionary multi-objective optimization: a historical view of the field. *IEEE computational intelligence magazine* **1**(1), 28–36 (2006)

- [4] Bao, C., Gao, D., Gu, W., Xu, L., Goodman, E.D.: A new adaptive decomposition-based evolutionary algorithm for multi-and many-objective optimization. *Expert Systems with Applications* **213**, 119080 (2023)
- [5] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation* **6**(2), 182–197 (2002) <https://doi.org/10.1109/4235.996017>
- [6] Bader, J., Zitzler, E.: Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary computation* **19**(1), 45–76 (2011)
- [7] Zhang, Q., Li, H.: Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation* **11**(6), 712–731 (2007)
- [8] Purshouse, R.C., Fleming, P.J.: On the evolutionary optimization of many conflicting objectives. *IEEE Transactions on Evolutionary Computation* **11**(6), 770–784 (2007) <https://doi.org/10.1109/TEVC.2007.910138>
- [9] Li, M., Yang, S., Liu, X.: Shift-based density estimation for pareto-based algorithms in many-objective optimization. *IEEE Transactions on Evolutionary Computation* **18**(3), 348–365 (2014) <https://doi.org/10.1109/TEVC.2013.2262178>
- [10] Yuan, J., Liu, H.-L., Gu, F.: A cost value based evolutionary many-objective optimization algorithm with neighbor selection strategy. In: 2018 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8 (2018). <https://doi.org/10.1109/CEC.2018.8477649>
- [11] Deb, K., Mohan, M., Mishra, S.: Evaluating the  $\epsilon$ -domination based multi-objective evolutionary algorithm for a quick computation of pareto-optimal solutions. *Evolutionary Computation* **13**(4), 501–525 (2005) <https://doi.org/10.1162/106365605774666895>
- [12] Zhou, J., Zou, J., Yang, S., Zheng, J., Gong, D., Pei, T.: Niche-based and angle-based selection strategies for many-objective evolutionary optimization. *Information Sciences* **571**, 133–153 (2021)
- [13] Das, S.S., Islam, M.M., Arafat, N.A.: Evolutionary algorithm using adaptive fuzzy dominance and reference point for many-objective optimization. *Swarm and evolutionary computation* **44**, 1092–1107 (2019)
- [14] Li, M., Yang, S., Liu, X., Shen, R.: A comparative study on evolutionary algorithms for many-objective optimization. In: Purshouse, R.C., Fleming, P.J., Fonseca, C.M., Greco, S., Shaw, J. (eds.) *Evolutionary Multi-Criterion Optimization*, pp. 261–275. Springer, Berlin, Heidelberg (2013)

- [15] Jiang, S., Zhang, J., Ong, Y.-S., Zhang, A.N., Tan, P.S.: A simple and fast hypervolume indicator-based multiobjective evolutionary algorithm. *IEEE Transactions on Cybernetics* **45**(10), 2202–2213 (2015) <https://doi.org/10.1109/TCYB.2014.2367526>
- [16] Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Fonseca, V.G.: Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation* **7**(2), 117–132 (2003) <https://doi.org/10.1109/TEVC.2003.810758>
- [17] Li, B., Li, J., Tang, K., Yao, X.: Many-objective evolutionary algorithms: A survey. *ACM Computing Surveys (CSUR)* **48**(1), 1–35 (2015)
- [18] Shang, K., Ishibuchi, H.: A new hypervolume-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation* **24**(5), 839–852 (2020) <https://doi.org/10.1109/TEVC.2020.2964705>
- [19] Tian, Y., Zhang, X., Cheng, R., He, C., Jin, Y.: Guiding evolutionary multiobjective optimization with generic front modeling. *IEEE Transactions on Cybernetics* **50**(3), 1106–1119 (2020) <https://doi.org/10.1109/TCYB.2018.2883914>
- [20] Sun, Y., Yen, G.G., Yi, Z.: Igd indicator-based evolutionary algorithm for many-objective optimization problems. *IEEE Transactions on Evolutionary Computation* **23**(2), 173–187 (2019) <https://doi.org/10.1109/TEVC.2018.2791283>
- [21] Wang, R., Zhang, Q., Zhang, T.: Decomposition-based algorithms using pareto adaptive scalarizing methods. *IEEE Transactions on Evolutionary Computation* **20**(6), 821–837 (2016) <https://doi.org/10.1109/TEVC.2016.2521175>
- [22] Asafuddoula, M., Ray, T., Sarker, R.: A decomposition-based evolutionary algorithm for many objective optimization. *IEEE Transactions on Evolutionary Computation* **19**(3), 445–460 (2015) <https://doi.org/10.1109/TEVC.2014.2339823>
- [23] Ishibuchi, H., Setoguchi, Y., Masuda, H., Nojima, Y.: Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes. *IEEE Transactions on Evolutionary Computation* **21**(2), 169–190 (2017) <https://doi.org/10.1109/TEVC.2016.2587749>
- [24] Hua, Y., Liu, Q., Hao, K., Jin, Y.: A survey of evolutionary algorithms for multi-objective optimization problems with irregular pareto fronts. *IEEE/CAA Journal of Automatica Sinica* **8**(2), 303–318 (2021) <https://doi.org/10.1109/JAS.2021.1003817>
- [25] Tian, Y., Si, L., Zhang, X., Tan, K.C., Jin, Y.: Local model-based pareto front estimation for multiobjective optimization. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **53**(1), 623–634 (2023) <https://doi.org/10.1109/TSMC>.

- [26] Cheng, R., Jin, Y., Olhofer, M., Sendhoff, B.: A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation* **20**(5), 773–791 (2016) <https://doi.org/10.1109/TEVC.2016.2519378>
- [27] Qi, Y., Ma, X., Liu, F., Jiao, L., Sun, J., Wu, J.: MOEA/D with Adaptive Weight Adjustment. *Evolutionary Computation* **22**(2), 231–264 (2014) [https://doi.org/10.1162/EVCO\\_a.00109](https://doi.org/10.1162/EVCO_a.00109) [https://direct.mit.edu/evco/article-pdf/22/2/231/1510396/evco\\_a\\_00109.pdf](https://direct.mit.edu/evco/article-pdf/22/2/231/1510396/evco_a_00109.pdf)
- [28] Li, X., Li, X., Wang, K., Yang, S.: A strength pareto evolutionary algorithm based on adaptive reference points for solving irregular fronts. *Information Sciences* **626**, 658–693 (2023) <https://doi.org/10.1016/j.ins.2023.01.073>
- [29] Li, L., Yen, G.G., Sahoo, A., Chang, L., Gu, T.: On the estimation of pareto front and dimensional similarity in many-objective evolutionary algorithm. *Information Sciences* **563**, 375–400 (2021) <https://doi.org/10.1016/j.ins.2021.03.008>
- [30] Wang, X., Zhang, F., Yao, M.: A many-objective evolutionary algorithm with estimating the convexity-concavity of pareto fronts and clustering. *Information Sciences* **644**, 119289 (2023) <https://doi.org/10.1016/j.ins.2023.119289>
- [31] Feng, W., Gong, D., Yu, Z.: Multi-objective evolutionary optimization based on online perceiving pareto front characteristics. *Information Sciences* **581**, 912–931 (2021) <https://doi.org/10.1016/j.ins.2021.10.007>
- [32] Gu, F., Cheung, Y.-M.: Self-organizing map-based weight design for decomposition-based many-objective evolutionary algorithm. *IEEE Transactions on Evolutionary Computation* **22**(2), 211–225 (2017)
- [33] Liu, Y., Ishibuchi, H., Masuyama, N., Nojima, Y.: Adapting reference vectors and scalarizing functions by growing neural gas to handle irregular pareto fronts. *IEEE Transactions on Evolutionary Computation* **24**(3), 439–453 (2020) <https://doi.org/10.1109/TEVC.2019.2926151>
- [34] Liu, Q., Jin, Y., Heiderich, M., Rodemann, T., Yu, G.: An adaptive reference vector-guided evolutionary algorithm using growing neural gas for many-objective optimization of irregular problems. *IEEE Transactions on Cybernetics* **52**(5), 2698–2711 (2022) <https://doi.org/10.1109/TCYB.2020.3020630>
- [35] Fritzke, B.: A growing neural gas network learns topologies. *Advances in neural information processing systems* (1995)
- [36] Wagner, T., Beume, N., Naujoks, B.: Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In: Obayashi, S., Deb, K., Poloni, C.,

Hiroyasu, T., Murata, T. (eds.) Evolutionary Multi-Criterion Optimization, pp. 742–756. Springer, Berlin, Heidelberg (2007)

- [37] Li, G., Wang, G.-G., Dong, J., Yeh, W.-C., Li, K.: Dlea: A dynamic learning evolution algorithm for many-objective optimization. *Information sciences* **574**, 567–589 (2021)
- [38] Ming, F., Gong, W., Wang, L.: A two-stage evolutionary algorithm with balanced convergence and diversity for many-objective optimization. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **52**(10), 6222–6234 (2022) <https://doi.org/10.1109/TSMC.2022.3143657>
- [39] Asafuddoula, M., Singh, H.K., Ray, T.: An enhanced decomposition-based evolutionary algorithm with adaptive reference vectors. *IEEE Transactions on Cybernetics* **48**(8), 2321–2334 (2018) <https://doi.org/10.1109/TCYB.2017.2737519>
- [40] Tian, Y., Cheng, R., Zhang, X., Jin, Y.: Platemo: A matlab platform for evolutionary multi-objective optimization [educational forum]. *IEEE Computational Intelligence Magazine* **12**(4), 73–87 (2017) <https://doi.org/10.1109/MCI.2017.2742868>
- [41] Vlennet, R., Fonteix, C., Marc, I.: Multicriteria optimization using a genetic algorithm for determining a pareto set. *International Journal of Systems Science* **27**(2), 255–260 (1996)
- [42] Cheng, R., Li, M., Tian, Y., Zhang, X., Yang, S., Jin, Y., Yao, X.: A benchmark test suite for evolutionary many-objective optimization. *Complex & Intelligent Systems* **3**, 67–81 (2017)
- [43] Huband, S., Hingston, P., Barone, L., While, L.: A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation* **10**(5), 477–506 (2006)
- [44] Ge, H., Zhao, M., Sun, L., Wang, Z., Tan, G., Zhang, Q., Chen, C.L.P.: A many-objective evolutionary algorithm with two interacting processes: Cascade clustering and reference point incremental learning. *IEEE Transactions on Evolutionary Computation* **23**(4), 572–586 (2019) <https://doi.org/10.1109/TEVC.2018.2874465>
- [45] Deb, K.: Multi-objective optimization using evolutionary algorithms. Wiley (2001)
- [46] Deb, K., Goyal, M., *et al.*: A combined genetic adaptive search (geneas) for engineering design. *Computer Science and informatics* **26**, 30–45 (1996)
- [47] Das, I., Dennis, J.E.: Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM journal*

- on optimization **8**(3), 631–657 (1998)
- [48] Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation* **18**(4), 577–601 (2014) <https://doi.org/10.1109/TEVC.2013.2281535>
- [49] Derrac, J., García, S., Molina, D., Herrera, F.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation* **1**(1), 3–18 (2011) <https://doi.org/10.1016/j.swevo.2011.02.002>

**Table 3:** The HV for different  $c_{cov}$  and  $c_{div}$

$c_{div}, c_{cov}$	Statistic	WFG1(3)	MaF9(3)	WFG7(10)	MaF2(10)	MaF8(10)	MaF5(15)	Average ranking
0.60, 0.80	Mean	9.4280e-1[1]	8.3541e-1[1]	9.3517e-1[1]	1.9873e-1[1]	1.0781e-2[1]	9.7821e-1[1]	1.0000e+0
	Std.Dev.	(2.75E-3)	(5.04E-3)	(4.34E-3)	(4.26E-3)	(1.01E-4)	(2.79E-3)	
0.60, 0.85	Mean	9.4097e-1[6]	8.3682e-1[1]	9.3261e-1[1]	1.9844e-1[1]	1.0765e-2[1]	9.7745e-1[5]	2.3333e+0
	Std.Dev.	(1.82E-3)	(2.13E-3)	(4.39E-3)	(5.15E-3)	(1.20E-4)	(2.73E-3)	
0.60, 0.90	Mean	9.4116e-1[3]	8.3656e-1[1]	9.3129e-1[1]	1.9846e-1[1]	1.0778e-2[1]	9.7839e-1[2]	1.5000e+0
	Std.Dev.	(3.23E-3)	(2.46E-3)	(4.50E-3)	(5.18E-3)	(1.04E-4)	(2.06E-3)	
0.70, 0.80	Mean	9.4057e-1[5]	8.3663e-1[1]	9.3324e-1[1]	1.9759e-1[1]	1.0775e-2[1]	9.7800e-1[4]	2.1667e+0
	Std.Dev.	(2.73E-3)	(1.97E-3)	(4.56E-3)	(5.38E-3)	(1.20E-4)	(1.96E-3)	
0.70, 0.85	Mean	9.4019e-1[4]	8.3604e-1[1]	9.3260e-1[1]	1.9920e-1[1]	1.0773e-2[1]	9.7836e-1[3]	1.8333e+0
	Std.Dev.	(7.32E-3)	(1.93E-3)	(4.68E-3)	(5.74E-3)	(1.48E-4)	(1.75E-3)	
0.70, 0.90	Mean	9.4243e-1[2]	8.3646e-1[1]	9.3233e-1[1]	1.9953e-1[1]	1.0732e-2[1]	9.7656e-1[6]	2.0000e+0
	Std.Dev.	(1.85E-3)	(1.84E-3)	(3.95E-3)	(5.19E-3)	(1.59E-4)	(2.89E-3)	

**Table 4:** The IGD for different  $\epsilon$

$\epsilon$	Statistic	VNT2(3)	WFG2(3)	MaF3(10)	MaF4(10)	MaF7(15)	MaF3(15)	Average ranking
1e-5	Mean	1.4556e-2[2]	1.5422e-1[1]	1.3545e-1[6]	5.2123e+1[5]	8.1881e-1[1]	1.9069e-1[4]	3.1667e+0
	Std.Dev.	(3.09E-3)	(3.65E-3)	(5.12E-2)	(2.01E+1)	(5.48E-2)	(7.88E-2)	
5e-5	Mean	1.3881e-2[1]	1.5346e-1[1]	9.5668e-2[3]	4.7584e+1[2]	8.1059e-1[1]	1.9293e-1[4]	2.0000e+0
	Std.Dev.	(8.47E-4)	(3.64E-3)	(2.93E-2)	(1.86E+0)	(2.61E-2)	(7.34E-2)	
1e-4	Mean	1.4235e-2[1]	1.5342e-1[1]	1.0119e-1[4]	4.8233e+1[4]	8.1200e-1[1]	1.8113e-1[4]	2.5000e+0
	Std.Dev.	(1.03E-3)	(4.45E-3)	(3.38E-2)	(2.34E+0)	(2.91E-2)	(5.03E-2)	
5e-4	Mean	1.4928e-2[3]	1.5320e-1[1]	9.2328e-2[2]	4.6673e+1[1]	8.1111e-1[1]	1.6505e-1[2]	1.6667e+0
	Std.Dev.	(1.60E-3)	(3.13E-3)	(2.91E-2)	(1.86E+0)	(2.95E-2)	(5.01E-2)	
1e-3	Mean	1.5446e-2[4]	1.5337e-1[1]	1.0047e-1[5]	4.7717e+1[3]	8.0784e-1[1]	1.6431e-1[3]	2.8333e+0
	Std.Dev.	(9.99E-4)	(3.63E-3)	(3.12E-2)	(2.03E+0)	(3.32E-2)	(3.96E-2)	
5e-3	Mean	2.4323e-2[5]	1.5222e-1[1]	8.2506e-2[1]	4.8619e+1[6]	8.1371e-1[1]	1.3879e-1[1]	2.5000e+0
	Std.Dev.	(3.89E-3)	(2.81E-3)	(1.83E-2)	(2.94E+0)	(4.79E-2)	(3.78E-2)	

**Table 5:** The IGD on VNT1-VNT3

Problem	M	CLIA	DEA-GNG	GFM-MOEA	PeEA	LMPFE	TS-MOEA-EPF
VNT1	3	1.7227e-1(1.34e-2)-	1.3897e-1(9.93e-3)-	1.1665e-1(2.97e-3)+	1.8004e-1(1.27e-2)-	1.2849e-1(5.60e-3)-	1.2188e-1(4.13e-3)
VNT2	3	5.5706e-2(6.12e-3)-	1.6095e-2(1.29e-3)-	3.8533e-2(3.29e-2)-	1.4554e-1(1.56e-2)+	1.3770e-2(1.22e-3)-	1.5114e-2(1.53e-3)
VNT3	3	7.8696e-2(5.32e-3)-	4.7295e-2(1.06e-2)-	2.0945e-1(5.32e-1)-	8.1098e-2(1.35e-2)-	3.1271e-2(1.27e-3)+	3.2265e-2(1.39e-3)
+ / - / $\approx$		0/3/0	0/3/0	1/2/0	0/3/0	2/1/0	2/1/0

**Table 6:** The HV on VNT1-VNT3

Problem	M	CLIA	DEA-GNG	GFM-MOEA	PeEA	LMPFE	TS-MOEA-EPF
VNT1	3	3.3995e-1(1.51e-3)-	3.4226e-1(1.24e-3)-	3.4427e-1(2.90e-4)=	3.3895e-1(1.09e-3)-	3.4423e-1(2.99e-4)=	3.4417e-1(9.42e-4)
VNT2	3	3.3066e-1(1.19e-3)-	3.3322e-1(2.70e-4)-	3.3041e-1(4.50e-3)-	3.2267e-1(8.09e-3)-	3.3340e-1(2.42e-4)=	3.3342e-1(1.69e-4)
VNT3	3	1.6809e-1(1.20e-3)-	1.7605e-1(4.31e-5)+	1.7564e-1(1.53e-4)-	1.7264e-1(7.05e-4)-	1.7589e-1(3.45e-5)-	1.7595e-1(3.26e-5)
+/- / ≈		0/3/0	1/2/0	0/2/1	0/3/0	0/1/2	

**Table 7:** The more mean metrics value of each algorithem on 15-objective WFG7

Problem	GD	Spread	BC
CLIA	4.20e-1(6.6e-3)	3.17e-1(3.0e-2)	8.86e-1(1.4e-2)
DEA-GNG	4.67e-1(4.1e-2)	5.38e-1(6.6e-2)	9.93e-1(3.8e-2)
GFM-MOEA	3.75e-1(9.5e-2)	2.92e-1(3.9e-1)	1.12e+0(7.3e-2)
PeEA	3.40e-1(7.2e-2)	4.97e-1(2.0e-1)	1.17e+0(4.4e-2)
LMPFE	2.47e-1(1.1e-2)	3.87e-1(3.8e-2)	1.14e+0(3.6e-3)
TS-MOEA-EPP	2.99e-1(2.4e-2)	3.23e-1(3.1e-2)	1.09e+0(9.2e-3)

**Table 8:** The IGD on WFG1-WFG9

Problem	M	CLIA	DEA-GNG	GFM-MOEA	PeEA	LMPFEE	TS-MOEA-EPF
WFG1	3	2.0389e-1(2.55e-2)-	2.0060e-1(2.16e-2)-	1.4901e-1(8.17e-3)=	2.7048e-1(3.56e-2)-	1.6120e-1(9.09e-3)-	1.54492e-1(1.59e-2)
	10	1.0990e+0(2.16e-2)+	9.4933e-1(1.44e-2)+	1.3873e+0(2.45e-1)-	1.3754e+0(1.34e-1)-	1.1836e+0(6.24e-2)-	9.6737e-1(2.13e-2)
	15	1.8412e+0(3.53e-2)-	1.7322e+0(5.27e-2)-	2.4498e+0(3.05e-1)-	1.9756e+0(9.16e-2)-	2.0275e+0(5.43e-2)-	1.6621e+0(6.44e-2)
WFG2	3	1.6153e-1(1.66e-3)-	1.5486e-1(9.08e-3)=	1.6910e-1(5.43e-3)-	2.0872e-1(9.61e-3)-	1.7743e-1(4.90e-3)-	1.5288e-1(2.91e-3)
	10	1.0596e+0(1.23e-2)=	1.0921e+0(3.67e-2)-	1.3649e+0(2.21e-1)-	1.3448e+0(4.56e-2)-	1.5903e+0(1.03e-1)-	1.0639e+0(2.78e-2)
	15	1.7962e+0(3.44e-2)-	2.2634e+0(2.42e-1)-	2.0249e+0(3.15e-1)-	1.9365e+0(1.22e-1)-	2.4514e+0(2.42e-1)-	1.6868e+0(5.52e-2)
WFG3	3	1.1359e-1(5.56e-3)-	1.1053e-1(1.34e-2)-	7.9676e-2(5.52e-3)+	1.8537e-1(2.55e-2)-	1.1525e-1(1.22e-2)-	9.1426e-2(5.27e-3)
	10	2.1283e+0(1.40e-2)-	1.6215e+0(1.45e+0)+	1.7143e+0(9.46e-1)+	1.3501e+0(2.94e-1)+	2.5661e+0(6.19e-1)-	2.0159e+0(2.30e-1)
	15	4.5839e+0(2.05e-1)-	1.3386e+0(4.32e-1)+	9.6696e+0(1.00e+0)-	2.5055e+0(4.51e-1)+	5.2840e+0(1.24e+0)-	4.2325e+0(7.01e-1)
WFG4	3	2.3071e-1(6.33e-3)-	2.1436e-1(3.87e-3)-	2.0606e-1(1.66e-3)=	2.6641e-1(9.56e-3)-	2.0979e-1(1.79e-3)-	2.0647e-1(2.40e-3)
	10	4.01338e+0(2.48e-2)-	4.0953e+0(2.79e-2)-	4.0707e+0(2.84e-2)-	5.0050e+0(1.70e-1)-	4.3292e+0(3.09e-2)-	3.9891e+0(1.92e-2)
	15	8.3159e+0(5.61e-2)+	1.0046e+1(8.05e-1)-	9.6722e+0(1.15e+0)-	9.8917e+0(6.23e-1)-	8.5504e+0(1.19e-1)-	8.4735e+0(1.43e-1)
WFG5	3	2.2935e-1(2.07e-3)-	2.2065e-1(2.59e-3)-	2.1490e-1(1.59e-3)+	2.6644e-1(5.99e-3)-	2.2011e-1(1.41e-3)-	2.1610e-1(2.30e-3)
	10	4.01010e+0(3.27e-2)=	4.2120e+0(6.39e-2)-	4.0172e+0(3.71e-2)-	4.9495e+0(7.69e-2)-	4.3403e+0(3.22e-2)-	3.9923e+0(3.12e-2)
	15	8.2016e+0(7.24e-2)+	9.6160+0(6.28e-1)-	9.3713e+0(1.22e+0)-	9.4597e+0(3.54e-1)-	8.4929e+0(1.14e-1)-	8.3086e+0(1.13e-1)
WFG6	3	2.5042e-1(1.46e-2)-	2.4018e-1(1.23e-2)-	2.2501e-1(9.83e-3)+	2.8874e-1(1.61e-2)-	2.3173e-1(8.95e-3)=	2.3216e-1(8.33e-3)
	10	4.0399e+0(2.95e-2)=	4.3839e+0(8.82e-2)-	4.0463e+0(2.82e-2)=	5.1958e+0(1.78e-1)-	4.5080e+0(2.72e-2)-	4.0510e+0(3.88e-2)
	15	8.2937e+0(1.02e-1)+	1.2648e+1(9.64e-1)-	1.0830e+1(2.00e+0)-	1.0061e+1(6.28e-1)-	8.8896e+0(2.12e-1)-	8.4637e+0(1.60e-1)
WFG7	3	2.2390e-1(3.30e-3)-	2.0813e-1(3.34e-3)-	2.0532e-1(2.28e-3)=	2.7149e-1(1.37e-2)-	2.0881e-1(2.16e-3)-	2.0476e-1(1.73e-3)
	10	4.0314e+0(2.84e-2)-	4.3028e+0(3.25e-1)-	4.0731e+0(3.87e-2)-	4.9388e+0(7.81e-2)-	4.3059e+0(2.82e-2)-	3.9620e+0(2.06e-2)
	15	8.3594e+0(9.09e-2)=	1.2618e+1(1.36e+0)-	9.3557e+0(1.16e+0)-	9.7900e+0(5.76e-1)-	8.5652e+0(8.04e-2)-	8.3334e+0(1.16e-1)
WFG8	3	3.1224e-1(5.16e-3)-	2.8536e-1(6.92e-3)-	2.6925e-1(3.97e-3)+	3.6393e-1(1.64e-2)-	2.8353e-1(4.02e-3)-	2.7330e-1(3.26e-3)
	10	4.2684e+0(4.08e-2)-	5.3964e+0(2.19e-1)-	4.3820e+0(8.51e-1)=	4.5492e+0(1.08e-1)-	4.3839e+0(3.04e-2)-	4.0707e+0(5.22e-2)
	15	8.5296e+0(1.36e-1)=	1.3342+1(6.43e-1)-	1.1078e+1(1.63e+0)-	1.0243e+1(5.22e-1)-	8.3541e+0(1.01e-1)+	8.4961e+0(1.43e-1)
WFG9	3	2.2092e-1(2.83e-3)-	2.0587e-1(3.32e-3)-	2.0280e-1(2.03e-3)=	2.6089e-1(1.38e-2)-	2.0709e-1(1.95e-3)-	2.0340e-1(2.54e-3)
	10	4.0527e+0(6.07e-2)-	4.3145e+0(1.07e-1)-	4.0478e+0(2.53e-1)-	4.2050e+0(3.68e-2)-	3.9536e+0(2.53e-2)	
	15	8.6571e+0(1.16e-1)-	1.0395e+1(8.86e-1)-	8.7739e+0(4.41e-1)-	9.0246e+0(4.70e-1)-	8.1855e+0(2.69e-1)-	8.0139e+0(1.05e-1)
+ / - / ≈		3/19/5	3/23/1	5/16/6	2/25/0	1/25/1	

**Table 9:** The HV on WFG1-WFG9

Problem	M	CLIA	DEA-GNG	GFM-MOEA	PeEA	LMPFE	TS-MOEA-EPF
WFG1	3	9.3163e-1(5.08e-3)-	9.2723e-1(4.51e-3)+	9.4284e-1(2.13e-3)+	9.1590e-1(7.75e-3)-	9.4464e-1(1.78e-3)+	9.4118e-1(2.99e-3)
	10	9.9969e-1(8.99e-5)+	9.9420e-1(1.66e-3)-	9.9509e-1(1.81e-2)-	9.9083e-1(3.80e-2)-	9.9999e-1(3.45e-5)+	9.9924e-1(2.16e-4)
	15	9.9978e-1(7.36e-5)+	9.9221e-1(2.72e-3)-	9.9497e-1(1.33e-2)=	9.9829e-1(5.26e-4)-	9.9999e-1(7.34e-6)+	9.9815e-1(3.93e-4)
WFG2	3	9.3140e-1(7.47e-4)-	9.2498e-1(4.04e-3)+	9.3825e-1(6.18e-4)+	9.0996e-1(7.50e-3)-	9.3306e-1(1.02e-3)-	9.3615e-1(8.11e-4)
	10	9.9543e-1(8.92e-4)-	9.8735e-1(2.97e-3)+	9.9921e-1(1.78e-3)+	9.9056e-1(5.85e-3)-	9.9960e-1(2.66e-4)+	9.9810e-1(3.46e-4)
	15	9.9621e-1(1.21e-3)-	9.7432e-1(6.97e-3)-	9.9798e-1(1.92e-3)=	9.9432e-1(2.47e-3)-	9.9942e-1(4.38e-4)+	9.9875e-1(7.71e-4)
WFG3	3	3.7645e-1(3.16e-3)-	3.8454e-1(5.64e-3)-	3.9784e-1(1.91e-3)+	3.5350e-1(1.09e-2)-	3.8233e-1(5.01e-3)-	3.8965e-1(2.57e-3)
	10	9.6615e-5(5.29e-4)=	0.0000e+0(0.00e+0)=	8.3135e-2(2.92e-2)+	3.4273e-2(1.96e-2)+	5.7249e-2(1.23e-2)+	7.0564e-5(3.86e-4)
	15	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	5.3663e-1(5.67e-3)-	5.4831e-1(2.25e-3)-	5.5632e-1(1.45e-3)+	5.3908e-1(2.54e-3)-	5.5530e-1(1.16e-3)=	5.5511e-1(1.60e-3)
	10	8.3675e-1(1.02e-2)-	8.3145e-1(8.22e-3)-	9.2572e-1(2.72e-3)+	9.1586e-1(4.86e-3)+	9.5442e-1(1.76e-3)+	9.1267e-1(4.16e-3)
	15	8.1018e-1(2.12e-2)-	8.3339e-1(3.31e-2)-	9.2992e-1(3.91e-2)=	9.2672e-1(1.14e-2)-	9.7814e-1(2.60e-3)+	9.4733e-1(7.86e-3)
WFG5	3	5.1321e-1(9.65e-4)-	5.0230e-1(2.61e-3)-	5.1895e-1(1.75e-3)=	5.0418e-1(2.02e-3)-	5.1788e-1(2.03e-3)=	5.1807e-1(3.78e-3)
	10	8.0995e-1(7.69e-3)-	7.6947e-1(2.11e-2)-	8.6983e-1(3.25e-3)+	8.6335e-1(4.16e-3)+	8.8791e-1(1.85e-3)+	8.5706e-1(4.89e-3)
	15	7.6886e-1(1.68e-2)-	7.6236e-1(2.49e-2)-	8.6044e-1(4.25e-2)-	8.7001e-1(1.10e-2)-	9.0434e-1(1.57e-3)+	8.9195e-1(3.81e-3)
WFG6	3	4.9411e-1(1.27e-2)-	4.9793e-1(1.34e-2)-	5.0626e-1(1.18e-2)=	4.8988e-1(1.54e-2)-	5.0601e-1(1.24e-2)=	5.0393e-1(9.37e-3)
	10	7.8486e-1(1.52e-2)-	7.6972e-1(1.95e-2)-	8.5309e-1(2.44e-2)=	8.2457e-1(1.49e-2)-	8.8648e-1(1.93e-2)+	8.4742e-1(1.57e-2)
	15	7.4707e-1(2.60e-2)-	6.8332e-1(5.65e-2)-	7.6944e-1(9.06e-2)-	8.1290e-1(2.38e-2)-	9.0305e-1(2.10e-2)+	8.7308e-1(1.85e-2)
WFG7	3	5.4852e-1(4.67e-3)-	5.5252e-1(2.64e-3)-	5.5968e-1(1.02e-3)+	5.4083e-1(3.63e-3)-	5.5803e-1(1.07e-3)-	5.5836e-1(1.06e-3)
	10	8.2807e-1(1.07e-2)-	8.3754e-1(3.18e-2)-	9.3255e-1(6.15e-3)=	9.2571e-1(5.30e-3)-	9.6156e-1(1.11e-3)+	9.3302e-1(4.37e-3)
	15	7.9227e-1(1.89e-2)-	7.5523e-1(6.61e-2)-	9.3400e-1(4.42e-2)-	9.3499e-1(2.23e-2)-	9.8339e-1(9.56e-4)+	9.7024e-1(2.20e-3)
WFG8	3	4.5700e-1(3.45e-3)-	4.6543e-1(4.43e-3)-	4.7521e-1(1.58e-3)=	4.4522e-1(4.07e-3)-	4.6610e-1(2.23e-3)-	4.7408e-1(1.86e-3)
	10	7.0697e-1(1.41e-2)-	7.1707e-1(3.02e-2)-	8.1568e-1(8.98e-2)=	8.3239e-1(15.50e-2)+	8.2404e-1(4.74e-3)+	7.9665e-1(3.75e-2)
	15	7.0132e-1(4.22e-2)-	6.8699e-1(5.44e-2)-	5.8562e-1(2.13e-1)-	8.9963e-1(7.07e-3)+	8.9382e-1(7.51e-3)+	8.2904e-1(1.97e-2)
+/-/≈		2/23/2	0/25/2	11/5/11	7/19/1	17/5/5	

Table 10: The IGD on MaF1-MaF9

Problem	M	CLIA	DEA-GNG	GFM-MOEA	PeEA	LMPFE	TS-MOEA-EPF
MaF1	3	4.2926e-2(5.86e-4)-	3.9078e-2(5.30e-4)-	3.9550e-2(2.76e-4)-	4.2078e-2(4.47e-4)-	4.1082e-2(7.33e-4)-	3.8482e-2(2.11e-4)
MaF1	10	2.4685e-1(3.34e-3)-	2.2871e-1(9.97e-3)+	2.2802e-1(5.70e-3)+	2.0725e-1(2.22e-3)+	2.2520e-1(4.06e-3)+	2.3571e-1(2.51e-3)
MaF1	15	3.0207e-1(3.95e-3)+	3.5485e-1(1.91e-2)+	3.9516e-1(4.34e-2)=	3.4016e-1(5.00e-3)+	2.9902e-1(3.61e-3)+	3.8747e-1(9.54e-3)
MaF2	3	2.8384e-2(1.26e-4)-	2.7863e-2(5.54e-4)-	2.6988e-2(1.95e-4)-	3.0069e-2(4.79e-4)-	2.7671e-2(3.22e-4)-	2.6845e-2(1.59e-4)
MaF2	10	1.5337e-1(1.95e-3)-	3.4183e-1(3.26e-2)-	1.5963e-1(3.12e-3)-	1.5860e-1(2.31e-3)-	1.5229e-1(3.13e-3)-	1.4739e-1(2.48e-3)
MaF2	15	3.4465e-1(4.76e-2)-	4.6631e-1(4.20e-2)-	2.6345e-1(3.99e-2)-	3.8867e-1(2.16e-2)-	1.7666e-1(2.52e-3)+	1.8721e-1(5.60e-3)
MaF3	3	4.2312e-2(4.86e-3)-	6.3565e-2(4.73e-2)-	9.0620e-2(1.39e-1)=	6.4776e-2(1.23e-2)-	4.5412e-2(2.18e-2)-	3.3678e-2(1.52e-3)
MaF3	10	1.8181e+5(2.66e+5)-	3.8798e+1(1.15e+2)-	3.9653e+1(1.24e+2)-	1.4057e-1(1.14e-1)-	1.3815e+6(1.02e+6)-	8.6420e-2(1.82e-2)
MaF3	15	1.1999e+9(3.19e+9)-	1.8727e+1(5.28e+1)-	1.9433e+0(4.36e+0)-	1.2292e-1(5.48e-3)=	1.6317e+6(1.06e+6)-	1.5566e-1(5.35e-2)
MaF4	3	2.9190e-1(8.20e-3)-	3.7197e-1(1.45e-1)-	2.5504e-1(6.08e-3)-	6.5952e-1(5.88e-1)-	5.9825e-1(3.15e-1)-	2.4853e-1(7.30e-3)
MaF4	10	8.0973e+1(2.56e+0)-	7.6194e+1(1.54e+1)-	4.2433e+2(3.54e+1)-	1.0619e+2(1.19e+1)-	7.8482e+1(1.64e+1)-	4.8408e+1(3.02e+0)
MaF4	15	3.2662e+3(4.02e+1)-	2.1198e+3(2.96e+2)-	1.3431e+4(1.04e+3)-	3.9240e+3(8.73e+2)-	3.9689e+3(2.91e+3)-	1.8531e+3(2.22e+2)
MaF5	3	1.1561e+0(1.30e+0)-	8.2899e-1(7.90e-1)-	5.1521e-1(6.43e-1)=	3.2222e-1(1.57e-2)+	7.0577e-1(7.27e-1)-	4.2223e-1(4.87e-1)
MaF5	10	7.4053e+1(2.64e+1)-	5.0782e+1(1.41e+0)-	1.4916e+2(9.13e+0)-	1.1117e+2(5.11e+0)-	7.8245e+1(7.34e+0)-	4.5187e+1(1.56e+0)
MaF5	15	2.4396e+3(8.05e+2)-	1.6640e+3(1.29e+2)=	6.6854e+3(5.85e+2)-	3.4380e+3(5.93e+2)-	2.5039e+3(3.43e+2)-	1.6557e+3(1.18e+2)
MaF6	3	8.0352e-3(1.18e-3)-	5.55775e-3(9.57e-4)-	3.9200e-3(4.23e-5)+	2.7099e-2(8.38e-3)-	4.3164e-3(1.20e-3)+	4.3630e-3(1.57e-4)
MaF6	10	2.2762e-2(7.13e-3)+	2.5291e+0(1.05e+0)=	3.1415e-1(3.48e-1)+	2.0455e-1(4.52e-2)+	1.9107e+0(1.07e+0)+	2.6157e+0(1.23e+0)
MaF6	15	1.9771e-1(1.84e-1)+	1.0276e+1(1.45e+1)+	2.0285e-1(1.29e-1)+	8.1039e-2(1.68e-2)+	1.1661e+0(7.59e-1)=	1.1316e+1(3.88e+1)
MaF7	3	1.7958e-1(1.89e-1)-	6.57782e-2(5.53e-2)+	1.6350e-1(2.01e-1)-	8.2704e-2(4.31e-3)-	1.0705e-1(1.10e-1)-	7.3851e-2(7.40e-2)
MaF7	10	2.9094e+0(4.06e-1)-	1.0299e+0(1.53e-1)-	8.3700e-1(1.61e-2)-	9.595e-1(2.43e-2)-	4.0775e+0(4.28e-1)-	8.1099e-1(3.30e-2)
MaF7	15	1.3670e+1(2.55e+0)-	7.4649e+0(1.77e+0)-	2.4585e+0(2.71e+0)+	1.6428e+0(2.33e-2)+	3.0423e+0(6.74e-1)-	2.6016e+0(4.72e-1)
MaF8	3	7.0753e-2(3.48e-3)-	7.8275e-2(9.68e-3)-	5.9802e-2(1.21e-3)=	7.0533e-2(3.34e-3)-	6.4507e-2(4.85e-3)-	6.0380e-2(1.47e-3)
MaF8	10	1.3490e-1(4.23e-3)-	1.4792e-1(2.59e-2)-	2.2994e-1(9.25e-2)-	1.2459e-1(3.10e-2)-	1.2459e-1(3.10e-2)-	9.6724e-2(4.09e-3)
MaF8	15	5.8878e-1(7.76e-5)-	2.5841e-1(4.26e-2)-	6.2544e-1(1.52e-1)-	2.3710e-1(1.52e-2)-	6.3389e-1(2.74e-1)-	1.7279e-1(2.71e-3)
MaF9	3	6.8617e-2(3.17e-3)-	2.8871e-1(1.67e-1)-	5.7728e-2(5.84e-4)=	6.9188e-2(2.84e-3)-	2.8693e-1(7.33e-2)-	5.9095e-2(2.56e-3)
MaF9	10	2.6342e-1(3.49e-2)+	5.2078e-1(1.87e-1)+	1.1719e-1(2.80e-2)+	1.7038e-1(3.23e-3)+	1.5538e+0(1.14e+0)-	6.5383e-1(2.47e-1)
MaF9	15	6.6979e-1(8.73e-2)-	4.4989e-1(1.15e-1)-	4.2995e+0(5.69+0)=	2.7445e-1(8.84e-3)-	1.0444e+0(3.78e-1)-	2.3091e-1(3.21e-2)
+ / - / ≈		4/23/0	5/20/2	6/15/6	7/19/1	5/21/1	

Table 11: The HV on MaF1-MaF9

Problem	M	CLIA	DEA-GNG	GFM-MOEA	FpEA	LMPFE	TS-MOEA-EPF
MaF1	3	2.1720e-1(5.59e-4)-	2.2092e-1(6.53e-4)-	2.2336e-1(2.53e-4)=	2.1926e-1(1.51e-3)-	2.1779e-1(8.84e-4)-	2.2373e-1(1.95e-4)
	10	1.3333e-7(3.46e-7)-	6.0185e-7(2.67e-7)+	4.4309e-7(5.04e-07)+	5.8540e-7(4.57e-7)+	3.1956e-7(4.21e-7)=	1.6656e-7(3.79e-7)
	15	0.0000e+0(0.00e+0)=	1.3042e-12(3.38e-12)+	7.1853e-13(8.40e-13)+	0.0000e+0(0.00e+0)=	0.0000e+0(0.00e+0)=	0.0000e+0(0.00e+0)
MaF2	3	2.4017e-1(3.96e-4)-	2.4234e-1(1.24e-3)-	2.4357e-1(1.16e-3)-	2.4567e-1(4.77e-4)-	2.4403e-1(1.01e-3)-	2.4637e-1(5.64e-4)
	10	2.1335e-1(2.74e-3)+	1.8747e-1(6.00e-3)-	2.0414e-1(3.06e-3)+	2.0985e-1(4.16e-3)+	1.7188e-1(7.18e-3)-	1.9948e-1(4.94e-3)
	15	1.9554e-1(8.11e-3)+	1.7201e-1(8.17e-3)+	1.8019e-1(8.15e-3)+	2.1625e-1(2.60e-3)+	1.0577e-1(1.96e-2)-	1.2949e-1(9.74e-3)
MaF3	3	9.5606e-1(4.25e-3)-	9.3778e-1(3.90e-2)-	8.8298e-1(1.98e-1)-	9.4253e-1(1.06e-2)-	9.5274e-1(1.71e-2)=	9.5951e-1(2.28e-3)
	10	0.0000e+0(0.00e+0)-	4.2296e-1(4.33e-1)-	1.6249e-1(3.25e-1)-	9.7071e-1(1.46e-1)-	0.0000e+0(0.00e+0)-	9.9970e-1(1.33e-3)
	15	0.0000e+0(0.00e+0)-	3.7203e-1(4.50e-1)-	4.4773e-1(4.15e-1)-	9.9663e-1(2.89e-3)=	0.0000e+0(0.00e+0)-	9.7177e-1(4.89e-2)
MaF4	3	5.2099e-1(5.60e-3)-	5.1441e-1(3.02e-2)-	5.3318e-1(4.63e-3)=	4.9317e-1(9.38e-2)-	4.7521e-1(5.80e-2)-	5.3355e-1(5.49e-3)
	10	3.5553e-6(9.88e-7)-	2.2557e-4(8.75e-5)+	2.7140e-8(7.36e-9)-	2.5661e-4(1.38e-5)+	7.8403e-5(1.27e-4)-	7.8799e-5(1.86e-5)
	15	8.7625e-12(7.12e-13)-	8.9831e-8(2.38e-8)+	1.1710e-13(3.59e-14)-	1.5917e-7(1.41e-8)+	1.7091e-8(4.70e-8)+	9.6573e-10(3.99e-9)
MaF5	3	4.3778e-1(1.49e-1)-	4.7746e-1(1.01e-1)-	5.2485e-1(18.08e-2)-	5.3631e-1(3.31e-3)+	4.9424e-1(9.75e-2)-	5.3059e-1(7.35e-2)
	10	8.1195e-1(4.04e-2)-	9.5930e-1(1.80e-3)+	4.1106e-1(1.26e-1)-	7.4988e-1(2.23e-2)-	5.2449e-1(1.55e-1)-	9.5567e-1(2.95e-3)
	15	2.7684e-2(2.60e-2)-	9.8330e-1(1.13e-3)+	2.1920e-10(1.15e-1)-	7.9451e-1(1.20e-1)-	8.7099e-1(1.54e-1)-	9.7840e-1(2.28e-3)
MaF6	3	1.9748e-1(8.79e-4)-	1.9968e-1(8.81e-4)-	2.0003e-1(9.70e-5)=	1.8546e-1(2.00e-3)-	1.9955e-1(1.81e-3)-	2.0002e-1(1.80e-4)
	10	9.9351e-2(1.07e-3)+	0.0000e+0(0.00e+0)=	4.9239e-2(5.01e-2)+	2.2777e-3(7.48e-3)-	1.0079e-2(3.08e-2)=	3.3619e-3(1.84e-2)
	15	5.7388e-2(4.32e-2)+	0.0000e+0(0.00e+0)=	6.0224e-2(3.11e-2)+	8.7487e-2(2.33e-3)+	2.9206e-3(1.31e-2)=	0.0000e+0(0.00e+0)
MaF7	3	2.5971e-1(2.12e-2)-	2.7658e-1(6.94e-3)-	2.5961e-1(3.78e-2)-	2.7143e-1(1.70e-3)-	2.7142e-1(1.32e-2)-	2.7757e-1(9.38e-3)
	10	3.9238e-2(1.12e-2)-	1.9049e-1(5.16e-3)+	1.5098e-1(8.54e-3)+	1.4071e-01(5.15e-3)+	1.1080e-1(1.81e-2)=	1.0688e-1(2.10e-2)
	15	1.5611e-5(4.09e-5)-	1.3432e-1(1.63e-2)+	8.2572e-2(2.47e-2)+	7.7172e-2(3.73e-3)+	1.5078e-3(6.58e-3)=	5.0347e-2(2.89e-2)
MaF8	3	2.7617e-1(1.49e-3)-	2.7015e-1(4.53e-3)-	2.7997e-1(5.61e-4)-	2.7308e-1(1.98e-3)-	2.7575e-1(2.97e-3)-	2.8069e-1(8.67e-4)
	10	9.2879e-3(1.70e-4)-	1.0103e-2(5.62e-4)-	1.0924e-2(5.66e-4)+	1.1126e-2(8.44e-5)+	1.0977e-2(9.34e-5)+	1.0787e-2(1.85e-4)
	15	4.7227e-5(1.90e-6)-	4.9750e-4(3.74e-5)-	5.4292e-4(3.68e-5)+	5.8810e-4(1.91e-5)+	5.0134e-4(8.07e-5)=	5.1671e-4(2.06e-5)
+ / - / ≈		5/19/3	10/15/2	11/12/4	12/13/2	2/18/7	

**Table 12:** The more mean metrics value of each algothem  
on 15-objective MaF8

Problem	GD	Spread	BC
CLIA	2.06e-3(8.4e-6)	1.00e+0(2.0e-4)	3.96e-1(1.1e-5)
DEA-GNG	3.70e-3(1.8e-3)	8.43e-1(1.2e-1)	3.44e-1(1.2e-2)
GFM-MOEA	1.96e-3(1.3e-4)	7.35e-1(8.9e-2)	1.97e-1(2.6e-2)
PeEA	4.37e-3(2.0e-3)	4.56e-1(2.9e-2)	2.90e-1(6.1e-3)
LMPFE	4.14e-3(2.8e-3)	6.09e-1(1.2e-1)	2.67e-1(4.0e-2)
TS-MOEA-EPF	3.11e-3(8.4e-4)	1.60e-1(1.5e-2)	3.69e-1(3.1e-3)

**Table 13:** The more mean metrics value of each algothem  
on 15-objective MaF1

Problem	GD	Spread	BC
CLIA	1.55e-2(1.0e-3)	4.77e-1(2.9e-2)	1.36e-1(3.6e-3)
DEA-GNG	1.43e-2(4.9e-4)	6.98e-1(9.2e-2)	1.22e-1(8.7e-3)
GFM-MOEA	1.02e-2(2.2e-3)	1.41e-1(8.1e-2)	1.63e-1(1.2e-2)
PeEA	1.23e-2(3.1e-4)	8.31e-2(2.7e-2)	1.66e-1(1.2e-3)
LMPFE	9.15e-3(7.5e-4)	3.69e-1(6.7e-2)	1.68e-1(3.3e-3)
TS-MOEA-EPF	6.84e-3(5.7e-4)	1.31e-1(9.7e-3)	1.86e-1(1.2e-3)