

# Multiple Populations for Multiple Objectives Framework with Bias Sorting for Many-objective Optimization

Qi-Te Yang, *Student Member, IEEE*, Zhi-Hui Zhan, *Senior Member, IEEE*, Sam Kwong, *Fellow, IEEE*, and Jun Zhang, *Fellow, IEEE*

**Abstract**—The convergence and diversity enhancement of multiobjective evolutionary algorithms (MOEAs) to efficiently solve many-objective optimization problems (MaOPs) is an active topic in evolutionary computation. By considering the advantages of the multiple populations for multiple objectives (MPMO) framework in solving multi-objective optimization problems and even MaOPs, this paper proposes an MPMO-based algorithm with a bias sorting (BS) method (termed MPMO-BS) for solving MaOPs to achieve both good convergence and diversity performance. For convergence, the BS method is applied to each population of the MPMO framework to enhance the role of nondominated sorting by biasedly paying more attention to the objective optimized by the corresponding population. This way, all the populations in the MPMO framework evolve together to promote the convergence performance on all objectives of the MaOP. For diversity, an elite learning strategy is adopted to generate locally mutated solutions, and a reference vector-based maintenance method is adopted to preserve diverse solutions. The performance of the proposed MPMO-BS algorithm is assessed on 29 widely used MaOP test problems and two real-world application problems. The experimental results show its high effectiveness and competitiveness when compared with seven state-of-the-art MOEAs for many-objective optimization.

**Index Terms**—Many-objective optimization problems (MaOPs), evolutionary computation, multi-objective evolutionary algorithm (MOEA), bias sorting, coevolution.

Manuscript received XXXX; revised XXXX; accepted XXXX. This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB2102102, in part by the National Natural Science Foundations of China under Grant 62176094 and Grant 61873097, in part by the Key-Area Research and Development of Guangdong Province under Grant 2020B010166002, in part by the Guangdong Natural Science Foundation Research Team under Grant 2018B030312003, in part by the National Research Foundation of Korea (NRF-2021H1D3A2A01082705), in part by the Hong Kong GRF-RGC General Research Fund under Grant 11209819 (CityU 9042816) and Grant 11203820 (CityU 9042598), and in part by the Hong Kong Innovation and Technology Commission (InnoHK Project CIMDA). (Corresponding authors: Zhi-Hui Zhan; Jun Zhang.)

Qi-Te Yang and Zhi-Hui Zhan are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China, also with the Pazhou Laboratory, Guangzhou 510330, China, and also with the Guangdong Provincial Key Laboratory of Computational Intelligence and Cyberspace Information, Guangzhou 510006, China (e-mail: zhanapollo@163.com).

Sam Kwong is with the Department of Computer Science, City University of Hong Kong, Hong Kong.

Jun Zhang is with Zhejiang Normal University, Jinhua 321004, China and also with Hanyang University, Ansan 15588, South Korea.

## I. INTRODUCTION

MANY promising multiobjective evolutionary algorithms (MOEAs) have been proposed in recent decades to solve multiobjective optimization problems (MOPs) [1][2]. These MOEAs include the elitist nondominated sorting genetic algorithm (NSGA-II) [3], the improved strength Pareto evolutionary algorithm (SPEA2) [4], the indicator-based evolutionary algorithm (IBEA) [5], the MOEA with decomposition (MOEA/D) [6], and the multiple populations for multiple objectives (MPMO)-based algorithm [7]. However, almost all these MOEAs face a challenge: the proportion of nondominated solutions in the current population becomes larger when solving problems with more than three objectives, i.e., many-objective optimization problems (MaOPs). Such enormous numbers of nondominated solutions lead to insufficient selection pressure, resulting in the degraded performance of MOEAs. This phenomenon is known as *dominance resistance* [8]. Another challenge in dealing with MaOPs is diversity maintenance since many existing diversity management methods may not work well in high-dimensional objective spaces. Due to the limitation of population size, it is more difficult for MOEAs to approximate the whole Pareto front (PF) when the number of objectives increases.

To tackle the challenges in convergence and diversity that MOEAs encounter when dealing with MaOPs, many approaches have been proposed. In general, the most common approaches can be divided into four categories: dominance-based, decomposition-based, indicator-based, and multi-population-based. Although they have shown good performance in solving MaOPs, some side effects have also been observed. For example, dominance-based approaches usually increase selection pressure by enlarging the dominating area, which can cause the loss of diversity [9]. For decomposition-based approaches, the distributed reference vectors in these algorithms may have an adverse effect on diversity maintenance with nonuniform or convex PFs [10][11]. For indicator-based approaches, one well-known drawback is the high computational complexity of indicators such as hypervolume (HV) [12], especially on MaOPs.

Among all the above approaches for solving MaOPs, the inherent multiple-population advantages in the MPMO framework make it very suitable to be extended for solving MaOPs with more objectives. In MPMO, a corresponding

population is adopted to optimize one objective approximating the PF. This strategy can accelerate each population approaching the PF. In every generation, different populations can communicate and share information. Compared with other approaches, the benefits of MPMO lie mainly in two aspects. First, the MPMO does not face the difficulty of sorting enormous numbers of nondominated solutions caused by the many objectives in MaOPs because the MPMO mainly addresses one objective in each population. Second, the MPMO does not need to deal with a large number of subproblems because it does not need to decompose the original MaOP. Nevertheless, as shown in [13], only considering one objective in each population may cause the PF margin phenomenon when dealing with MaOPs. This indicates that each population in an MPMO-based algorithm must consider the priority of the corresponding optimized objective and the equality among other objectives.

In MOEAs, one of the common strategies that can reflect the equal relationship among all objectives is nondominated sorting (NDS) based on the Pareto dominance relationship. However, similar to the inefficiency of dominance-based approaches, the effectiveness of NDS will deteriorate on MaOPs due to the emergence of a large number of nondominated solutions (more details are provided in Section II-C). Therefore, we propose a bias sorting (BS) method that prioritizes the corresponding population objective without ignoring other objectives. The BS method pays more attention to the convergence on the corresponding population objective to increase the selection pressure. Solutions with better values on the corresponding population objective will be preferentially ranked at the lower front (i.e., better front). Therefore, to combine the advantages of both the BS and NDS methods, this paper proposes an adaptive sorting strategy. First, during the early search stage, solutions can quickly approach the PF with the guidance of the BS. Second, by considering the convergence status of each population, the sorting method is adaptively switched from BS to NDS in the later search stage to further promote convergence.

Based on the consideration above, we propose an MPMO-based MOEA with BS (termed MPMO-BS) for solving MaOPs. The MPMO-BS follows the principle of *convergence first and diversity second* [14].

For convergence, each population in MPMO-BS optimizes a specific objective, and all of them together can promote the convergence performance on all objectives of the MaOP. For each population, two steps are carried out in every generation. First, solutions are ranked at several fronts via the more suitable sorting method between the BS or the NDS. Herein, the solutions at the lower front are better than those at the higher front. Second, as for the solutions ranked at the same front, we develop an auxiliary convergence fitness (ACF) strategy to further distinguish their convergence status. Therefore, we can measure the relationship between any two solutions in each population through the above two strategies.

For diversity, we use an archive to preserve well-performing solutions. After the parallel optimization process of all populations, the nondominated solutions in these populations are moved into the archive. An elite learning

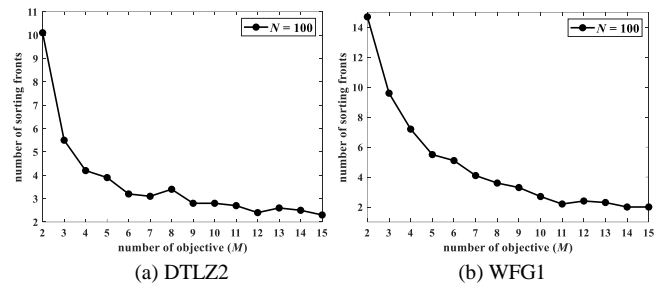


Fig. 1. Curves of the mean number of sorting fronts in the initial population versus  $M$ .

strategy (ELS) is developed to help jump out of possible local optima and help the archiving solutions efficiently converge to the PF. After that, solutions with good diversity in the archive are preserved via predefined uniform reference vectors.

The contributions of this paper are as follows.

1) We propose the BS method, which can increase selection pressure among solutions and accelerate convergence. Via the adaptive sorting strategy, the algorithm can adopt a suitable sorting method between BS and NDS according to the convergence status of the population.

2) We propose the ACF strategy, which can further distinguish the convergence relationship among solutions ranked at the same front and can adapt itself to different sorting methods.

3) Derived from the MPMO framework, multiple populations in MPMO-BS evolve in parallel to ensure convergence on different objectives. The introduced ELS can promote the diversity of solutions selected from all populations.

4) To facilitate communication among populations, we develop a population reallocation strategy between the archive and populations.

The remainder of this paper is organized as follows. Section II provides the background. The details of the proposed MPMO-BS are given in Section III. Section IV presents the experimental results and detailed analysis of MPMO-BS on MaOPs and two real-world application problems. Finally, Section V concludes this paper.

## II. BACKGROUND

### A. MOPs and MaOPs

Real-world problems usually involve multiple conflicting objectives that need to be considered simultaneously. Such complex problems are called MOPs. A MOP with  $M$  objectives to be minimized can be described as:

$$\min F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x}))^T, \quad (1)$$

$$s.t. \mathbf{x} \in \Omega,$$

where  $\Omega$  is the decision (variable) space, and  $\mathbf{x} = (x_1, \dots, x_D)^T$  is a candidate solution.  $F: \Omega \rightarrow R^M$  represents  $M$  real-valued objective functions, and  $R^M$  denotes the objective space. When the number of objectives in a MOP (i.e., the  $M$ ) is larger than three, such a MOP is also called a MaOP.

Because of conflicting objectives, there is no unique optimal solution in a MOP that can optimize all the objectives simultaneously. Thus, the goal of solving a MOP is to obtain a set of optimal solutions, called Pareto optimal solutions. Given

two solutions  $\mathbf{x}, \mathbf{y} \in \Omega$ ,  $\mathbf{x}$  is said to Pareto dominate  $\mathbf{y}$  ( $\mathbf{x} < \mathbf{y}$ ) if  $f_i(\mathbf{x}) \leq f_i(\mathbf{y})$  for  $\forall i \in \{1, \dots, M\}$  and  $\exists j \in \{1, \dots, M\}$  that  $f_j(\mathbf{x}) < f_j(\mathbf{y})$ .  $\mathbf{x}$  is said to be a Pareto optimal solution if there does not exist another  $\mathbf{x}^* \in \Omega$  such that  $\mathbf{x}^* < \mathbf{x}$ . All Pareto optimal solutions form the Pareto set (PS). The PF, the corresponding objective vector set of the Pareto optimal solutions, is defined as:

$$\mathbf{PF} = \{F(\mathbf{x}) \in R^M \mid \mathbf{x} \in \mathbf{PS}\} \quad (2)$$

### B. Related Works

The existing approaches for tackling MaOPs can roughly be divided into four categories.

The first category focuses on dominance-based MOEAs, which increase the selection pressure among nondominated solutions by modifying the Pareto dominance relationship. Some typical recent dominance relationships include  $\varepsilon$ -dominance [15],  $\alpha$ -dominance [16], preference order ranking [17], and  $\theta$ -dominance [18]. For example, a method that can control the dominance area of solutions was proposed in [19]. A fuzzy-based Pareto dominance was proposed in [20] to distinguish Pareto nondominated solutions. Such fuzzy logic was also adopted in some new dominance relations, such as  $L$ -dominance [21] and  $(1-k)$ -dominance [22]. A grid dominance was proposed in [23] to strengthen the selection pressure among solutions through three grid-based criteria. In addition, a strengthened dominance relation (SDR) was proposed in [24] for MaOPs. A niching technique based on the angles between solutions was adopted in SDR to help maintain solutions with the best convergence in each niche.

The second category is decomposition-based approaches. They convert an original MaOP into several single-objective optimization subproblems and use predefined reference vectors to guarantee distribution. Solutions are guided to approximate the PF following the direction of each reference vector. NSGA-III [25] is one of the representative approaches of this category. In addition, many MOEA/D variants [26]-[28] have been developed recently. For example, an MOEA/D with an effective stable matching model (MOEA/D-STM) was proposed in [29]. In [30], a Pareto adaptive scalarizing method was inserted into MOEA/D. In [31], a decomposition-based MOEA called RVEA was proposed, in which an angle penalized distance scalarization approach can equilibrate the diversity and convergence when dealing with MaOPs.

The third category is called indicator-based approaches, in which indicators are used to assess the quality of solutions for selection. This category of approaches includes GDE-MOEA [32], MOMBI-II [33], and AR-MOEA [34]. In [35], a distinctive hypervolume-based MOEA called HypE was proposed, in which a Monte Carlo simulation method was adopted to replace the accurate HV estimation for computationally complexity reduction. However, it is worth noting that the accuracy of this approximate calculation is influenced by the number of sampling points. In [36], a utility tensor was introduced to reduce the computational cost of the hypervolume contribution. This method can improve efficiency by avoiding meaningless repetitive calculations. In [37], an inverted generational distance (IGD) [38] indicator-based MOEA called MaOEA/IGD was proposed, and a decomposition-based nadir point estimation

method was adopted to promote the calculation of IGD.

The fourth category is multi-population-based approaches. Although many MOEAs have been developed for solving MaOPs, most of them use only one population to simultaneously optimize all objectives. Such a population needs to coordinate the optimized degree of each solution on all objectives, slowing down the approaching speed of the population to the PF. Therefore, it is difficult to explore all objectives by one population. Considering this, the MPMO framework was proposed in [7], where multiple populations are adopted for solving multiple objectives simultaneously. Generally, various optimization algorithms, such as the genetic algorithm (GA) [39]-[41], ant colony system [42]-[45], PSO [7][13][46]-[48], and differential evolution [49]-[52], can be adopted for each population in the MPMO framework. The MPMO framework has shown promising performance in solving MOPs, indicating its potential ability to solve MaOPs. Therefore, an MPMO-based coevolutionary particle swarm optimization was proposed in [13] for solving MaOPs. In [13], the authors proposed a bottleneck objective learning strategy, which can alleviate the side effect of poorly performing objectives (named bottleneck objectives) for particles in approximating the PF. This strategy pays particular attention to two objectives (the bottleneck objective and the optimized objective of the corresponding population) in the process of particle updating while ignoring the influence of other objectives. It does accelerate particles toward the PF but may make populations only focus on their optimized bottleneck objectives. This may cause falling into local optima and be harmful to diversity.

In addition to the above four categories of MOEAs, many studies have been proposed for solving MaOPs. For instance, a hybrid algorithm that mixes the preponderance of Pareto dominance and a reference vector-based decomposition approach, called SPEA/R, was proposed in [53]. As knee points are naturally preferred by decision-makers, the information of knee points is used in [54] to promote the convergence and diversity of the population. A preference-inspired coevolutionary algorithm (PICEA-g) was proposed in [55], coevolving a family of preferences simultaneously with the population for better convergence. In [56], a MaOP was transformed into a MOP with two indicative objectives (i.e., convergence and diversity). During optimization, a population was assigned to different clusters according to the two indicative objective values, and well-performing solutions were selected from each cluster via a clustering-based sequential selection method. In [57], the authors integrated different solution selection methods into an ensemble framework to promote the performance of selected solutions. All these works show that the research into MaOPs is a significant and active research topic.

### C. Nondominated Sorting

NDS has been widely used in MOEAs since it was proposed in [58]. It is a strategy that can rank solutions in the population into different fronts based on their Pareto dominance relationship. Given a population  $P$  with a set of solutions, these solutions can be ranked into  $v$  different fronts, denoted as  $F_i$  ( $i = 1, 2, \dots, v$ ). After NDS, each solution at the front  $F_i$  is

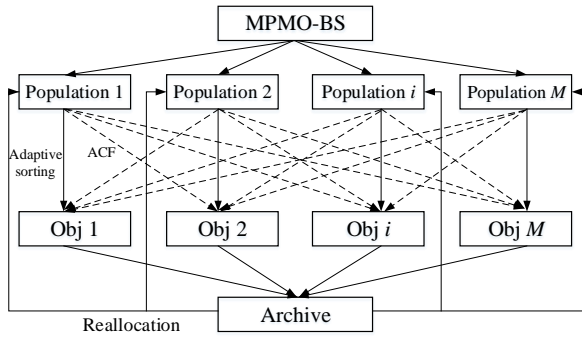


Fig. 2. Framework of MPMO-BS.

dominated by at least one solution ranked at  $F_{i-1}$ .

Over the past few decades, a number of NDS methods have been proposed in dominance-based MOEAs. Among them, the most representative is the fast nondominated sort in NSGA-II. Since then, many approaches have been proposed to simplify the NDS. For example, an arena's principle-based NDS was proposed in [59]. In [60], a dominance tree and divide-and-conquer mechanism were used to reduce the number of redundant comparisons of dominance relations among solutions. In [61], a new  $M$ -front sorting method was proposed. By using the dominance relationship information at the last generation,  $M$ -front can reduce the best-case complexity. Moreover, an efficient nondominated sort was proposed in [62], in which a strategy of presorting was used to make the latter ranked solutions unable to dominate the former. In this paper, we also adopt the efficient nondominated sorting approach in [62] for NDS. In addition to the abovementioned methods, some other methods have also been reported, e.g., [63][64].

NDS has been widely used in dominance-based MOEAs for solving MOPs. For MaOPs, NDS has also been used in MOEAs such as NSGA-III [25] and KnEA [54]. However, one phenomenon may be observed in NDS when solving MaOPs: most solutions in the population are ranked at similar fronts, which will weaken the selection pressure among solutions. Moreover, NDS should consider the priority of each solution on every objective. As the number of objectives increases, NDS becomes increasingly ineffective. Such NDS results will ultimately deteriorate or even hinder the selection process. To illustrate this, we use NSGA-II, the most representative algorithm using NDS, to conduct experiments on the DTLZ2 and WFG1 test instances with different objectives. The number of sorting fronts in the initial population is recorded, and the curves of the mean results in 10 runs are shown in Fig. 1. It can be seen that the number of sorting fronts versus  $M$  declines rapidly. Especially when  $M > 5$ , only two or three fronts remain. This indicates that a large number of solutions are located at the same front, which makes the algorithm unable to distinguish promising solutions using only the NDS method.

### III. PROPOSED MPMO-BS ALGORITHM

#### A. The Framework of the MPMO-BS

For a MaOP with  $M$  objectives, the main idea of MPMO is to use  $M$  populations to optimize the MaOP, with each population (of size  $N_s$ ) corresponding to only one objective and all

#### Algorithm 1 BS ( $p^i, i$ )

**Begin**

- 1: Initialize  $L = \{L^1, \dots, L^M\}$ ;  
/\*  $L$  is a storage of all bi-objective sorting results \*/
- 2:  $L^i = \emptyset$ ;
- 3: **For Each**  $j = 1, \dots, M$  and  $j \neq i$  **Do**
- 4:    $\{L_j^i, \dots\} = \text{NDS on objective } i \text{ and } j$ ;
- 5: **End For**
- 6:  $\{F_1, \dots, F_v\} = \max(L)$ ;
- 7: Rearrange  $\{F_1, \dots, F_v\}$ ;
- 8: **Return**  $\{F_1, \dots, F_v\}$ ;

**End**

populations working together to approximate the whole PF. MPMO-BS follows the MPMO framework, while the adaptive sorting strategy is used to accelerate the convergence in each population. Through adaptive sorting, solutions are ranked at different fronts to distinguish the convergence status. For solutions at the same front, the ACF is adopted to further qualify their convergence status. An external archive  $A$  with size  $N$  is used to store the nondominated solutions from all populations. Finally, all solutions in  $A$  are reassigned to each population for information sharing. The final archive  $A$  is output once the termination condition is satisfied. The framework of MPMO-BS is shown in Fig. 2.

#### B. BS Method

Without loss of generality, population  $p^i$  is taken as an example to describe the process of BS. The pseudocode of the BS process is given in **Algorithm 1**.

First, we carry out  $M-1$  times bi-objective NDS on objective  $i$  and each other objective  $j$  ( $j \in \{1, \dots, M\}$  and  $j \neq i$ ) for all solutions in population  $p^i$  corresponding to the  $i$ -th objective (Lines 3-5 in **Algorithm 1**). Then each solution has  $M-1$  sorting front results, and the largest one is defined as its bias front in  $p^i$  (Line 6 in **Algorithm 1**). Notably, this sorting may cause a front chasm. That is, some solutions may be sorted to the higher front before the lower front has appeared. Under this situation, the bias front should be rearranged to eliminate the chasm (Line 7 in **Algorithm 1**). For example, the bias front of  $p^i$  is  $\{1, 1, 4, 4, 5, 2, 5\}$ , and after rearrangement, it becomes  $\{1, 1, 3, 3, 4, 2, 4\}$  as the final bias front result.

Through the above description of BS, we can see that there are some similarities between BS and the one-objective sorting method (i.e., only objective  $i$  is taken into account when sorting  $p^i$ ). They both tend to accelerate convergence by giving priority to one objective. However, compared with the one-objective sorting method, the BS can accommodate the other objectives and minimize the loss of diversity. It can be regarded as a sorting method lying between one-objective sorting and NDS since it can promote convergence while not ignoring diversity.

Herein we present an example of a 4-objective space to illustrate the BS method and compare the differences among BS, NDS, and one-objective sorting methods.

**Example:** Consider a population  $p^1$  containing 5 solutions in the objective space:  $a(1, 4, 8, 3)$ ,  $b(2, 3, 7, 8)$ ,  $c(2, 4, 6, 5)$ ,  $d(8, 2, 7, 3)$ , and  $e(3, 4, 5, 4)$ , and 3 optimal solutions need to be selected from them.

The front result of these solutions by the NDS method is  $\{1,$

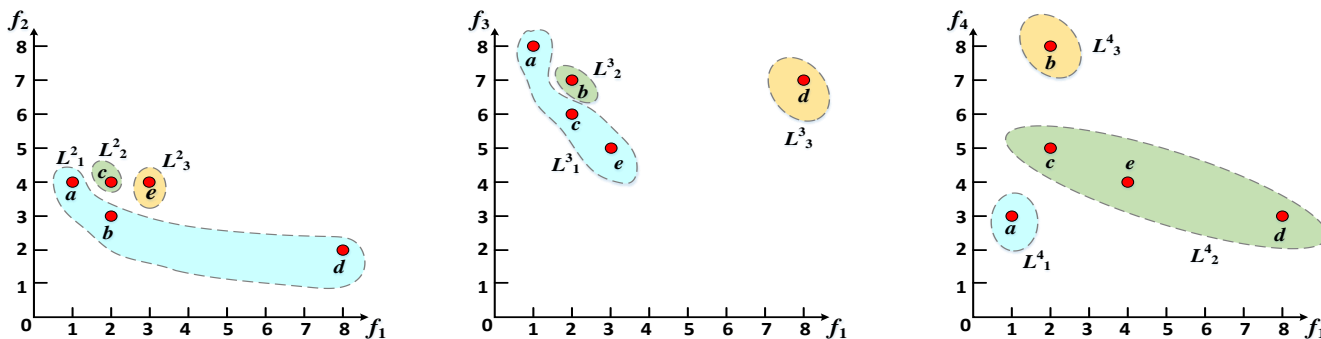
1, 1, 1, 1}, which indicates that these solutions are nondominated with each other. At this time, it is impossible to decide among these solutions which one is better without other auxiliary operators. In other words, they are all regarded as optimal solutions in the current generation. In contrast, these solutions are ranked at {1, 2, 2, 4, 3} if the one-objective sorting method is adopted on  $f_1$ . At this time, solutions  $a$ ,  $b$ , and  $c$  are seen as the 3 optimal solutions. However, if we take into account the other objective values, we can see that  $b$  is severely deficient on  $f_4$  compared with others, whereas  $d$  only performs worse on  $f_1$  and  $e$  performs moderately on all objectives. It seems that choosing  $b$  may not be a good choice if diversity is considered.

However, different results can appear if BS is used. Fig. 3 illustrates the BS process in  $p^1$ , and the bias front result is {1, 3, 2, 3, 3}. Then, it can be distinguished that in  $p^1$ , which is more biased to objective  $f_1$ , only solution  $a$  is regarded as the best one. It is worth noting that  $b$ ,  $d$ , and  $e$  are ranked at the same front, although there are significant differences in their  $f_1$  values. This is the result of BS after comprehensively considering the values of these solutions on other objectives.

The above example shows the advantages of BS compared with NDS and one-objective sorting. On the one hand, the BS can make a more detailed rank division among solutions than NDS. This makes the sorting mechanism restore the capability to distinguish the merits of solutions on MaOPs. On the other hand, different from the one-objective sorting, the BS is not completely biased toward a single objective but takes diversity into account. Combined with the MPMO framework, the BS can work efficiently to eliminate the phenomenon of *dominance resistance* and accelerate convergence rapidly. It is worth noting that herein we use the bi-objective NDS in our BS, but potential performance improvement may be achieved by extending the bi-objective NDS to  $k$ -objective NDS ( $2 < k < M$ ) in the BS. Inevitably, the computational complexity will also increase because there will be more objective combinations in the BS process if  $k$  is larger.

### C. Convergence Maintenance

In MPMO-BS, multiple populations search in parallel, and each population pays attention to its convergence. In this section, we illustrate in detail how each population maintains convergence.



(a) NDS on objective  $f_1$  and  $f_2$

(b) NDS on objective  $f_1$  and  $f_3$

(c) NDS on objective  $f_1$  and  $f_4$

Fig. 3. Example of BS in population  $p^1$ . The bi-objective NDS results in (a), (b) and (c) are  $L^2 = \{1,1,2,1,3\}$ ,  $L^3 = \{1,2,1,3,1\}$ , and  $L^4 = \{1,3,2,2,2\}$ , respectively. So the BS result in  $p^1$  is  $\max\{L^2, L^3, L^4\} = \{1,3,2,3,3\}$ . Noting that there is no front chasm in this example.

### 1) Adaptive Sorting Strategy

During the early search stage, solutions usually do not converge well. At this time, BS is very useful in accelerating convergence since it prioritizes different objectives in different populations. However, as the BS's forward search process and convergence performance improve, there is no need to pay more attention to one specific objective. In contrast, all objectives need to be treated equally to reduce the differences among populations. Considering this, the adaptive sorting strategy is developed here, which can adaptively switch from BS to NDS according to the convergence status of each population.

For any population  $p^i$ , a trigger of the sorting type (denoted as  $ST_i$ ) is set to determine which sorting method should be used. The trigger  $ST_i$  is initialized to 1, which represents the BS method. The value of  $ST_i$  is verified during each generation, and it will be fixed to 2 once verified equal to 2 in a certain generation. In other words, the NDS method is and will be continuously used in current and all generations to come if  $ST_i = 2$ . Otherwise, if  $ST_i = 1$ , BS is used and the trigger  $ST_i$  will be updated.

The update of  $ST_i$  is based on the BS result. After BS, the number of solutions at front  $F_1$  is counted as  $N_{opt}$ . Then the solution proportion ( $SP$ ) of  $F_1$  in  $p^i$  is calculated as:

$$SP = \frac{N_{opt}}{N_s} \quad (3)$$

where  $SP$  is a number between (0, 1] that can numerically reflect the current population's convergence status. The larger  $SP$  is, the more optimal solutions based on BS are, indicating that the smaller difference between BS and NDS on the result of sorting is. Therefore, when  $SP$  is greater than a threshold  $\theta$ ,  $ST_i$  is updated to 2 so that NDS can be adopted for subsequent sorting. In this paper,  $\theta$  is set to 0.8, and its sensitivity is investigated in Section IV-F.

### 2) ACF Strategy

After sorting, the solutions at the same front are nondominated with each other. The ACF strategy is proposed to further distinguish the convergence status of the solutions at the same front. Mutual evaluation [65] is a loss function that can evaluate the quality of each solution, and ACF follows the main idea of this approach.

First, the objective values of any solution  $\mathbf{x}$  in  $p^i$  need to be preprocessed to be positive, as follows:

---

**Algorithm 2 Update\_Archive** ( $A, S$ )

---

```

Begin
1: If  $A = \emptyset$  Then
2:    $A = S$ ;
3: Else
4:   For Each  $s \in S$  Do
5:     While  $\exists a \in A$  is dominated by  $s$  Do
6:       Remove  $a$  from  $A$ ;
7:     End While
8:     If  $\forall a \in A$  are nondominated with  $s$  Then
9:       Add  $s$  into  $A$ ;
10:    End If
11:   End For
12: End If
13: Return  $A$ ;
End

```

---

$$\bar{f}_k(\mathbf{x}) = f_k(\mathbf{x}) - \min f_k(\mathbf{y}) + 10^{-6}, \mathbf{y} \in p^i \quad (4)$$

where  $10^{-6}$  is to prevent the value from being equal to 0.

Second, considering that there are  $N_r$  solutions at the  $r$ -th front in  $p^i$ , the calculation method of ACF is designed according to the type of adopted sorting method, which is specified as follows.

If BS is adopted in  $p^i$ , then for any two solutions at that front, the mutual evaluation of  $\mathbf{x}_m$  ( $m = 1, \dots, N_r$ ) evaluated by  $\mathbf{x}_n$  ( $n = 1, \dots, N_r$  and  $n \neq m$ ) is:

$$cv_{mn} = \max_k \frac{\bar{f}_k(\mathbf{x}_n)}{\bar{f}_k(\mathbf{x}_m)}, \quad k = 1, \dots, M \text{ and } k \neq i \quad (5)$$

Otherwise, in the case that NDS is adopted in  $p^i$ , then the mutual evaluation is defined as:

$$cv_{mn} = \max_k \frac{\bar{f}_k(\mathbf{x}_n)}{\bar{f}_k(\mathbf{x}_m)}, \quad k = 1, \dots, M \quad (6)$$

Finally, the ACF value of each solution is as:

$$f_{AC}(\mathbf{x}_m) = \min_{n \neq m} cv_{nm}, \quad n = 1, \dots, N_r \quad (7)$$

ACF can be considered to be a quality indicator to assess the convergence status of solutions at the same front. Solutions with larger ACF values have better convergence status once they are ranked at the same front.

Now let us return to the previous example given in Section III-B. Solutions  $b$ ,  $d$ , and  $e$  are ranked at the same front via BS in  $p^1$ . Now, we calculate their ACF values according to Eq. (5) and (7). The results are  $f_{AC}(b) = 1$ ,  $f_{AC}(d) = 2$ , and  $f_{AC}(e) = 1.4$ . Therefore, solution  $d$  is regarded to be optimal among these three solutions. At this point, through BS and ACF, 3 optimal solutions selected from the 5 solutions are  $a$ ,  $c$ , and  $d$ .

### 3) Environmental Selection

We choose  $N_s$  solutions with better convergence from each population and its offspring population during every generation. First,  $p^i$  and its offspring population  $q^i$  form a combined population of size  $2N_s$ . All solutions in this combined population are ranked at several fronts via the adaptive sorting strategy. Starting from  $F_1$ , the solutions with better ACF values are preserved until the number of preserved solutions reaches  $N_s$ . These preserved solutions are adopted as  $p^i$  for the next generation.

---

**Algorithm 3 Population\_Reallocation** ( $A$ )

---

```

Begin
1: Initialize  $P = \emptyset, S = A$ ;
2: While  $S$  is not empty Do
3:   For Each  $i = 1, \dots, M$  Do
4:     Move solution with the best value on objective  $i$  from  $S$  into  $p^i$ ;
5:   End For
6: End While
7:  $P = \{p^1, p^2, \dots, p^M\}$ ;
8: Return  $P$ ;
End

```

---

### D. Diversity Maintenance

In MPMO-BS, archive  $A$  is used for the remaining well-converged solutions from all populations. Then solutions with good diversity are selected from  $A$  and preserved. In this section, we describe the diversity maintenance process.

#### 1) Archive Update and ELS

Archive  $A$  is the storage of nondominated solutions from all populations. It is initialized to be empty and updated in every generation and is also regarded as the final solution set found by MPMO-BS. **Algorithm 2** gives the process of archive updating. The input solution set  $S$  is copied into  $A$  if  $A$  is empty. Otherwise, each solution  $s \in S$  is compared with all the solutions in  $A$ . First, the solutions that are dominated by  $s$  are removed from  $A$ . Then,  $s$  will be added into  $A$  if all remaining solutions in  $A$  are nondominated with  $s$ .

After storing all nondominated solutions in each population into  $A$ , the ELS proposed in [7] is adopted to help solutions jump out of possible local optima. In MPMO-BS, we randomly select several solutions from  $A$  and copy them into a temporary set. The number of selected solutions equals half of the current size of  $A$ . For each solution  $\mathbf{x} = (x_1, \dots, x_D)$  in this temporary set, a random dimension  $d$  is chosen to mutate by Gaussian distribution as:

$$x_d = x_d + (x_d^{max} - x_d^{min}) \cdot \text{Gaussian}(0, \sigma^2) \quad (8)$$

where  $x_d^{max}$  and  $x_d^{min}$  are the upper and lower bounds of the  $d$ -th dimension, respectively. Gaussian  $(0, \sigma^2)$  is a Gaussian distribution with a mean value of 0 and a standard deviation value of  $\sigma$  ( $\sigma = 0.5$ ). Then  $x_d$  is confirmed to be within the feasible search range of  $[x_d^{min}, x_d^{max}]$ . If not,  $x_d$  is set to the corresponding bound. After the mutation, all solutions in this temporary set are evaluated, and archive  $A$  will be updated by this temporary set. The update process is the same as **Algorithm 2** with this temporary set as input.

#### 2) Solutions Preservation

At the end of every generation, solutions with poor diversity performance are deleted from  $A$  if the size of  $A$  is larger than  $N$ . In this paper, the reference-vector-based solution selection method in [13] is adopted.  $N$  solutions are selected from  $A$  based on reference vectors, which are generated by a double-layer method.

### E. Population Reallocation

The solutions preserved in  $A$  perform better than those in each population because of the ELS and the multi-population coevolution technique. Therefore, if there are enough solutions



---

**Algorithm 4** MPMO-BS
 

---

```

Begin
1: Initialize  $A = \emptyset$ ,  $P = \{p^1, p^2, \dots, p^M\}$ ;
2:  $\{ST_1, \dots, ST_M\} = \{1, \dots, 1\}$ ; /* 1 stands for BS, 2 stands for NDS */
3: For Each  $i = 1, \dots, M$  Do
4:    $\{F_1, \dots, F_v\} = \mathbf{BS}(p^i, i)$ ; //Algorithm 1
5:   Compute the  $f_{AC}$  of each solution in  $p^i$  via Eq. (5) and (7);
6:    $A = \mathbf{Update\_Archive}(A, F_i)$ ; //Algorithm 2
7: End For
8: While not finished Do
   /* Convergence maintenance */
9:   For Each  $i = 1, \dots, M$  Do
10:    Generate offspring population  $q^i$  by  $p^i$ ;
11:     $p^i = p^i \cup q^i$ ;
12:    If  $ST_i == 1$  Then
13:       $\{F_1, \dots, F_v\} = \mathbf{BS}(p^i, i)$ ; //Algorithm 1
14:      Update  $ST_i$ ;
15:      Compute the  $f_{AC}$  of each solution in  $p^i$  via Eq. (5) and (7);
16:    Else If  $ST_i == 2$  Then
17:       $\{F_1, \dots, F_v\} = \mathbf{NDS}(p^i)$ ; //Section II-C
18:      Compute the  $f_{AC}$  of each solution in  $p^i$  via Eq. (6) and (7);
19:    End If
20:     $p^i = \mathbf{Environmental\_Selection}(\{F_1, \dots, F_v\}, f_{AC})$ ; //Section III-C-3
21:     $A = \mathbf{Update\_Archive}(A, p^i)$ ; //Algorithm 2
22:  End For
   /* Diversity maintenance */
23:   $ELS = \mathbf{Elitist\_Learning\_Strategy}(A)$ ; //Section III-D-1
24:   $A = \mathbf{Update\_Archive}(A, ELS)$ ; //Algorithm 2
25:  If  $|A| > N$  Then
26:     $A = \mathbf{Solutions\_Preservation}(A)$ ; //Section III-D-2
27:  End If
   /* Population reallocation */
28:  If  $|A| == N$  Then
29:     $P = \mathbf{Population\_Reallocation}(A)$ ; //Algorithm 3
30:  End If
31: End While
32: Return  $A$ ;
End

```

---

in archive  $A$ , reassigning solutions in  $A$  to each population can greatly improve search efficiency and information communication. Here we develop a population reallocation strategy to achieve this purpose. By this population reallocation strategy, solutions in each population are replaced by those from  $A$ . The pseudocode of the population reallocation process is shown in **Algorithm 3**. First, solutions in  $A$  are copied into a temporary set  $S$ , and all populations are set to the null set. Then, the solution with the best value on objective  $i$  ( $i = 1, 2, \dots, M$ ) in  $S$  is moved to  $p^i$  one by one. The above operation will repeat until  $S$  is empty. Note that the population reallocation strategy is carried out in every generation only when the size of archive  $A$  equals  $N$  so that all the solutions in  $A$  can be allocated to all the populations.

#### F. Complete MPMO-BS and Complexity Analysis

Based on the above operations, we propose an algorithm MPMO-BS for MaOPs. The framework of MPMO-BS can be summarized into three parts: 1) multi-population coevolution for convergence, 2) solution preservation archive for diversity and 3) population reallocation for information sharing. The pseudocode of MPMO-BS is shown in **Algorithm 4**.

Within one generation, MPMO-BS consists of the following seven operations: adaptive sorting, ACF calculation, envi-

ronmental selection, archive update, ELS, solution preservation, and population reallocation. For a MaOP with  $M$  objectives, assume the archive size is  $N$ , and the size of each population is  $N_s$ .

First, we analyze the computational complexity of each operation in a single population. In the adaptive sorting, it takes  $O(MN_s^2)$  in the worst-case for the combined population of size  $2N_s$  if NDS is adopted. Otherwise, a runtime of  $O(M^2)$  is needed if the BS is chosen as the sorting method. Then, it takes  $O(N_s \log N_s)$  time for the ACF calculation. In the environmental selection, it takes  $O(N_s \log N_s)$  for selecting solutions according to their front results and ACF values. For archive updating, the computational complexity of Line 21 of **Algorithm 4** is  $O(NN_s)$ . Therefore, the complexity of the MPMO component with  $M$  populations is  $O(M(MN_s^2 + N_s \log N_s + N_s \log N_s + NN_s))$  if NDS is adopted, and is  $O(M(M^2 + N_s \log N_s + N_s \log N_s + NN_s))$  if BS is adopted. Usually  $N > N_s > M$ , so the worst-case complexity of the MPMO component can be simplified to  $O(M^2 N_s^2 + MNN_s)$ .

The ELS requires  $O(N)$  computations, and the archive updating of Line 24 of **Algorithm 4** requires  $O(N^2)$  computations. In solution preservation, the computational complexity here is  $O(M^2 N + M^3 + N + MN^2)$ , which is  $O(MN^2)$ . Finally, the population reallocation takes  $O(\log N)$ .

Considering all the above computations, the overall worst-case computational complexity of one generation of MPMO-BS is  $O((M^2 N_s^2 + MNN_s) + N + N^2 + MN^2 + \log N)$ , which can be reduced to  $O(M^2 N_s^2 + MN^2)$ . In all of our simulations, we have used  $N_s \approx N/M$ , so the worst-case complexity can finally be reduced to  $O(MN^2)$ .

## IV. EXPERIMENTAL STUDIES

In this section, we present the experimental study of the proposed MPMO-BS and compare it with seven state-of-the-art MOEAs: NSGA-III [25], SPEA/R [53], MaOEA/IGD [37], KnEA [54], NSGA-II/SDR [24], Mo4Ma [56], and VMEF [57]. The codes for the first 5 compared MOEAs come from an open platform called PlatEMO [66], and the code of VMEF can be found on the homepage of the original authors. After that, we give an experimental analysis to show the effectiveness of each strategy in MPMO-BS.

#### A. Benchmark Problems and Parameter Settings

In this paper, 29 widely used test problems, including DTLZ1-7 [67], WFG1-9 [68], MaF1-6 [69][70], Minus DTLZ1-4 (DTLZ1-4<sup>-1</sup>) [71], Convex-DTLZ2 (CDTLZ2), and Scaled DTLZ (SDTLZ)1-2 in [25] are adopted to test the performance of MPMO-BS. More parameter details about the test problems are shown in Table I. For WFG,  $K$  (position parameter) and  $L$  (distance parameter) are set to  $M - 1$  and 10, respectively. The scaling factor for SDTLZ1-2 is set to 2.

The common parameter settings of all algorithms are given as follows.

- 1) Operators: The simulated binary crossover (SBX) [72] with a distribution index of 20 and polynomial mutation [73] with a distribution index of 20 are used to generate offspring. The crossover probability  $p_c$  and mutation probability  $p_m$  are set as 1.0 and  $1/D$ , respectively.

TABLE I  
PARAMETER SETTINGS OF TESTED BENCHMARK PROBLEMS

Problems	$M$	$D$
DTLZ1, DTLZ1 <sup>-1</sup> , SDTLZ1	8, 10, 15	$M + 4$
DTLZ2-6, MaF1-6, DTLZ2-4 <sup>-1</sup> , SDTLZ2, CDTLZ2	8, 10, 15	$M + 9$
DTLZ7	8, 10, 15	$M + 19$
WFG1-9	8, 10, 15	$K + L$

- 2) Population (or archive in MPMO-BS) size:  $N = 156, 275,$  and  $240$  for 8-, 10-, and 15-objective problems, respectively.
- 3) Termination condition: 150000, 200000, and 240000 fitness evaluations for 8-, 10-, and 15-objective problems, respectively.
- 4) The number of runs: Each algorithm runs 30 times independently.

Other parameters not mentioned in the compared algorithms are set according to their original papers for a fair comparison. Specifically, the rate of knee points ( $T$ ) in KnEA on different problems is listed in Table S.I in the supplementary material. For MPMO-BS, the size of each population  $N_s$  is set to  $\lfloor N/M \rfloor$ . Therefore, the size of each population in MPMO-BS is equal to 20, 28, and 16 when  $M = 8, 10,$  and  $15,$  respectively. The number of reference vectors in MPMO-BS is set to  $N$ .

### B. Performance Metrics

IGD and HV are two widely used metrics in MOEAs since they consider both diversity and convergence while evaluating the obtained solutions. A small IGD value typically indicates a better performance of the solution set. In contrast, the smaller the HV value is, the worse the performance of the obtained solution set.

**IGD:** Considering that  $P^*$  is a well-distributed reference point set sampled on the true PF, for an approximate solution set  $P$  obtained by the optimization algorithm, the IGD is defined as:

$$\text{IGD}(P, P^*) = \frac{1}{|P^*|} \left( \sum_{v \in P^*} \min_{\mathbf{x} \in P} (ed(v, F(\mathbf{x}))) \right) \quad (9)$$

where  $ed(v, F(\mathbf{x}))$  is the Euclidean distance between  $v$  and  $F(\mathbf{x})$ . Herein, the number of reference points in  $P^*$  is set as 10000.

**HV:** Let  $\mathbf{z} = (z_1, \dots, z_m)^T$  be a reference point that is dominated by all obtained solutions in the objective space, then the HV metric of solution set  $P$  is defined as:

$$\text{HV}(P) = \text{VOL} \left( \bigcup_{\mathbf{x} \in P} [f_1(\mathbf{x}), z_1] \times \dots \times [f_m(\mathbf{x}), z_m] \right) \quad (10)$$

where  $\text{VOL}(\cdot)$  is the Lebesgue measure. In our experiments,  $\mathbf{z}$  is set as to  $(1.1, \dots, 1.1)^T$  for all test instances. Note that each objective value of the obtained solutions was normalized between  $[0,1]$  according to the nadir point of true PF. In addition, Monte Carlo simulation is used to replace the accurate HV estimation because of the computational complexity. During each HV estimation, 1000000 sampling points are used.

### C. Experimental Results and Discussion

#### 1) Results on DTLZ and WFG

Due to space limitations, only the statistical results in terms of IGD on DTLZ1-7 and WFG1-6 are given in Table II, where

the best result is marked in **boldface** and the second-best result is *italicized*. The details of the entire experimental results can be found in the supplementary material. The statistical results in terms of IGD and HV on DTLZ and WFG problems are given in Table S.II and S.III, respectively. For all test instances, Wilcoxon's rank-sum test [74] at a 5% significance level is conducted to compare the significance of differences between MPMO-BS and the compared algorithms. The symbols '+', '-', and '=' indicate that the metric value of MPMO-BS is significantly better than, worse than, and similar to the corresponding compared algorithm, respectively.

From the IGD results in Table II, we notice that MPMO-BS performs significantly better than the compared seven MOEAs on most instances. When comprehensively considering the best and second-best results, MPMO-BS achieves promising performance on 26 out of 39 instances, superior to all the compared algorithms. Moreover, the significant test results show that MPMO-BS is superior to NSGA-III, SPEA/R, MaOEA/IGD, KnEA, Mo4Ma, VMEF, and NSGA-II/SDR on 29, 31, 35, 24, 29, 31, and 26 instances.

For DTLZ1-3, MPMO-BS and NSGA-II/SDR perform better than the other compared algorithms. MPMO-BS outperforms NSGA-II/SDR on most DTLZ1-3 instances whereas it underperforms NSGA-II/SDR on 15-objective DTLZ3. This may be due to the multimodal characteristic of DTLZ3 with many local optimal fronts. As different populations in MPMO-BS focus on different objectives, they may therefore be drawn into different local optimal fronts. This makes it difficult for MPMO-BS to completely jump out of the local optimum through the ELS. Nevertheless, MPMO-BS outperforms the other six MOEAs on DTLZ3. DTLZ4 is a nonuniform problem, which also creates a challenge for MPMO-BS in terms of diversity maintenance. On this problem, MPMO-BS only outperforms NSGA-II/SDR. DTLZ5 and DTLZ6 are dimension reduction problems, and it can be seen that MPMO-BS obtains the best or second-best IGD results on these instances. In contrast, the other compared MOEAs perform poorly on both DTLZ5 and DTLZ6. This shows that these algorithms cannot handle redundant information well in degeneration cases. DTLZ7 is a test problem with disconnected PF, in which KnEA obtains the best IGD results. However, MPMO-BS still outperforms most of the compared algorithms.

WFG1 is designed to examine whether an algorithm can handle a problem with flat bias and a mixed shape of the PF. MPMO-BS and KnEA perform significantly better than the other six MOEAs. WFG2 has disconnected PF, aiming to examine the ability to handle non-separable, variable dependencies. MPMO-BS does not perform the best on this problem. This may be due to some PF fragments in the middle of the objective space, making it difficult for MPMO-BS to choose the suitable population to converge to these PF fragments. In addition, the disconnectedness of PF also makes it difficult for MPMO-BS to share information among populations. Nevertheless, MPMO-BS generally outperforms most of the compared algorithms on these two problems, with the second best performance on 8-objective and 10-objective WFG1 and the best performance on 8-objective WFG2. For WFG3 with a



TABLE II  
MEAN AND STANDARD DEVIATION IGD VALUES OBTAINED BY EIGHT ALGORITHMS ON DTLZ1-7 AND WFG1-6 PROBLEMS

Problem	M	MPMO-BS	NSGA-III	SPEA/R	MaOEA/IGD	KnEA	Mo4Ma	VMEF	NSGA-II/SDR
DTLZ1	8	<b>9.9598e-2 (1.63e-3)</b>	1.1918e-1 (2.13e-2)+	1.6003e-1 (2.20e-2)+	3.9722e-1 (3.70e-1)+	3.5278e-1 (4.56e-2)+	2.0649e-1 (1.68e-2)+	<i>1.1254e-1 (6.69e-3)+</i>	1.5889e-1 (1.42e-2)+
	10	<b>1.1075e-1 (2.13e-3)</b>	<i>1.2447e-1 (3.61e-2)+</i>	1.6462e-1 (2.80e-2)+	1.6176e-1 (1.68e-1)+	3.2691e-1 (7.52e-2)+	1.9441e-1 (1.87e-2)+	1.5318e-1 (5.38e-2)+	1.7198e-1 (1.23e-2)+
	15	<b>1.3123e-1 (1.73e-2)</b>	<i>1.3286e-1 (1.49e-2)+</i>	3.0790e-1 (1.44e-1)+	2.1452e-1 (2.07e-1)+	2.3537e-1 (3.11e-2)+	2.4365e-1 (1.85e-2)+	9.4587e-1 (5.45e-1)+	2.2438e-1 (2.47e-2)+
DTLZ2	8	<b>3.1685e-1 (3.06e-4)</b>	3.5962e-1 (2.52e-2)+	3.6542e-1 (3.08e-3)+	<i>3.4658e-1 (1.01e-2)+</i>	3.8327e-1 (7.05e-3)+	4.2103e-1 (2.04e-2)+	3.5853e-1 (2.56e-3)+	5.4486e-1 (1.39e-1)+
	10	<b>4.1905e-1 (1.92e-3)</b>	4.7718e-1 (5.96e-2)+	4.3784e-1 (4.34e-4)+	4.3981e-1 (3.97e-3)+	4.5057e-1 (4.39e-2)+	4.6759e-1 (9.53e-3)+	<i>4.2729e-1 (2.40e-3)+</i>	4.4173e-1 (6.63e-3)+
	15	<b>5.2755e-1 (1.03e-2)</b>	5.5941e-1 (3.53e-2)+	<i>5.3101e-1 (6.64e-4)+</i>	9.4432e-1 (2.02e-1)+	7.1229e-1 (2.16e-1)+	6.1706e-1 (1.56e-2)+	5.4863e-1 (1.99e-3)+	5.8054e-1 (1.45e-2)+
DTLZ3	8	<b>3.4326e-1 (3.88e-2)</b>	6.0806e-1 (6.78e-1)+	6.6288e+0 (3.82e+0)+	1.1656e+1 (7.49e+0)+	1.0122e+0 (1.42e-1)+	7.3298e-1 (2.03e-1)+	1.0068e+1 (5.31e+0)+	<i>4.6366e-1 (1.28e-1)+</i>
	10	<i>6.1751e-1 (1.66e-1)</i>	7.2595e-1 (5.73e-1)+	6.3070e+0 (5.41e+0)+	5.7475e+0 (3.84e+0)+	1.1112e+0 (1.23e-1)+	7.7484e-1 (1.98e-1)+	4.7255e+1 (2.18e+1)+	<b>4.3552e-1 (4.79e-3)</b>
	15	8.2015e-1 (6.78e-2)	<i>7.2479e-1 (1.43e-1)-</i>	4.4135e+0 (3.58e+0)+	3.4929e+0 (2.58e+0)+	3.1374e+0 (5.77e+0)+	9.4166e-1 (1.81e-1)+	1.3931e+2 (2.47e+1)+	<b>6.0979e-1 (3.18e-2)</b>
DTLZ4	8	4.2695e-1 (1.41e-1)	4.0857e-1 (7.93e-2)-	<i>3.6696e-1 (2.83e-3)</i>	3.7134e-1 (8.35e-2)-	3.7350e-1 (6.00e-3)-	4.0296e-1 (1.32e-2)-	<b>3.5923e-1 (2.60e-3)</b>	7.6728e-1 (7.28e-2)+
	10	4.4607e-1 (5.97e-2)	4.9374e-1 (7.05e-2)-	4.3800e-1 (5.79e-4)-	4.3641e-1 (1.51e-2)-	<i>4.4350e-1 (1.25e-2)-</i>	4.5348e-1 (1.17e-2)-	<b>4.2934e-1 (2.11e-3)</b>	7.7490e-1 (4.41e-2)+
	15	5.7401e-1 (4.00e-2)	5.5142e-1 (3.70e-2)-	<b>5.2982e-1 (1.57e-3)</b>	5.5212e-1 (8.32e-3)-	<i>5.4435e-1 (3.50e-3)-</i>	5.8720e-1 (9.99e-3)+	5.5204e-1 (2.34e-3)+	7.9198e-1 (3.61e-2)+
DTLZ5	8	<b>1.1545e-1 (3.48e-2)</b>	2.3770e-1 (5.23e-2)+	7.1346e-1 (1.09e-1)+	5.1541e-1 (1.92e-1)+	2.8178e-1 (4.87e-2)+	1.5694e-1 (8.78e-2)+	3.2806e-1 (1.17e-1)+	<i>1.1751e-1 (2.15e-2)</i>
	10	<b>1.0516e-1 (3.23e-2)</b>	3.3785e-1 (8.24e-2)+	6.0281e-1 (1.43e-1)+	4.3854e-1 (1.79e-1)+	2.9500e-1 (4.23e-2)+	1.7726e-1 (5.06e-2)+	3.8512e-1 (1.84e-1)+	<i>1.4944e-1 (3.01e-2)+</i>
	15	<b>1.4962e-1 (2.79e-2)</b>	4.7684e-1 (1.42e-1)+	9.2157e-1 (2.67e-1)+	5.1185e-1 (2.02e-1)+	3.3481e-1 (5.03e-2)+	1.8877e-1 (5.33e-2)+	4.9011e-1 (1.45e-1)+	<i>1.8712e-1 (5.33e-2)+</i>
DTLZ6	8	<i>2.5134e-1 (9.77e-2)</i>	3.7303e-1 (2.43e-1)+	3.7499e+0 (1.33e+0)+	7.0938e-1 (1.32e-1)+	5.0978e-1 (1.20e-1)+	2.9705e-1 (5.65e-2)+	7.9089e-1 (2.47e-1)+	<b>2.1246e-1 (8.82e-2)</b>
	10	<b>2.0015e-1 (5.80e-2)</b>	3.3579e-1 (1.08e-1)+	1.0302e+0 (4.36e-1)+	6.8890e-1 (7.07e-2)+	5.9308e-1 (1.16e-1)+	3.0354e-1 (5.83e-2)+	1.1449e+0 (3.69e-1)+	<i>2.3906e-1 (7.53e-2)</i>
	15	<i>2.4282e-1 (8.89e-2)</i>	6.8457e-1 (4.39e-1)+	1.6881e+0 (3.41e-1)+	6.7913e-1 (1.08e-1)+	6.4922e-1 (1.54e-1)+	3.9318e-1 (1.87e-1)+	1.1000e+0 (3.82e-1)+	<b>2.0395e-1 (8.51e-2)</b>
DTLZ7	8	9.0185e-1 (5.98e-2)	<i>8.1608e-1 (3.83e-2)-</i>	1.3072e+0 (5.46e-2)+	1.1988e+0 (5.51e-2)+	<b>7.0472e-1 (2.79e-2)</b>	1.0792e+0 (1.86e-1)+	9.5646e-1 (7.60e-2)+	9.8921e-1 (1.10e-1)+
	10	1.2277e+0 (1.40e-1)	<i>1.1740e+0 (8.99e-2)-</i>	2.4367e+0 (5.91e-1)+	1.4425e+0 (4.92e-2)+	<b>9.0507e-1 (1.70e-2)</b>	1.2258e+0 (1.66e-1)=	1.3421e+0 (1.77e-1)=	1.5485e+0 (2.81e-1)+
	15	5.2439e+0 (8.19e-2)	4.4358e+0 (7.53e-1)-	<i>2.3938e+1 (1.74e+1)+</i>	<b>2.2002e+0 (8.81e-2)</b>	2.4245e+0 (2.51e-1)-	4.4096e+0 (8.00e-1)-	6.0755e+0 (1.10e-1)+	4.3042e+0 (3.67e-1)-
WFG1	8	<i>9.2697e-1 (4.14e-2)</i>	9.8938e-1 (4.65e-2)+	1.2946e+0 (8.17e-2)+	7.7598e+0 (2.00e+0)+	<b>9.2529e-1 (4.52e-2)</b>	1.2519e+0 (7.04e-2)+	1.0054e+0 (3.26e-2)=	1.6543e+0 (1.66e-1)+
	10	<i>1.0259e+0 (2.48e-2)</i>	1.1022e+0 (7.34e-2)+	1.3450e+0 (8.22e-2)+	7.3825e+0 (3.70e+0)+	<b>9.8753e-1 (1.47e-2)</b>	1.2425e+0 (4.72e-2)+	1.1525e+0 (4.33e-2)+	1.7890e+0 (1.37e-1)+
	15	1.6754e+0 (1.83e-1)	<b>1.4404e+0 (1.20e-1)</b>	1.7814e+0 (9.37e-2)+	1.1255e+1 (5.59e+0)+	<i>1.5607e+0 (7.09e-2)</i>	1.7912e+0 (6.68e-2)+	1.8956e+0 (8.35e-2)+	2.4341e+0 (4.97e-2)+
WFG2	8	<b>1.0257e+0 (1.08e-1)</b>	1.1589e+0 (1.82e-1)+	1.0930e+0 (1.88e-2)+	2.2537e+0 (2.62e-1)+	1.0741e+0 (2.70e-2)+	1.0694e+0 (4.58e-2)+	<i>1.0297e+0 (2.65e-2)</i>	1.4381e+0 (1.63e-1)+
	10	1.2156e+0 (2.58e-1)	1.3787e+0 (1.66e-1)+	1.1802e+0 (2.06e-2)-	2.3799e+0 (4.77e-1)+	<i>1.1787e+0 (3.22e-2)</i>	<b>1.1047e+0 (3.05e-2)</b>	1.2370e+0 (4.43e-2)+	1.5958e+0 (1.25e-1)+
	15	1.8144e+0 (3.75e-1)	<i>1.5885e+0 (6.05e-2)=</i>	<b>1.4391e+0 (1.76e-2)</b>	3.2157e+0 (6.10e-1)+	1.6986e+0 (5.10e-2)=	1.6885e+0 (3.67e-2)=	1.9876e+0 (8.63e-2)+	2.3587e+0 (9.83e-2)+
WFG3	8	1.2343e+0 (1.02e+0)	1.2838e+0 (4.77e-1)+	1.4926e+0 (9.98e-2)+	8.8784e+0 (3.14e-2)+	<i>1.0283e+0 (1.78e-1)</i>	<b>2.1784e-1 (6.98e-2)</b>	1.1174e+0 (1.43e-1)-	1.0720e+0 (1.93e-1)+
	10	<i>8.9504e-1 (8.52e-1)</i>	2.6651e+0 (4.73e-1)+	2.7195e+0 (1.46e-1)+	1.1185e+1 (1.84e-2)+	1.1844e+0 (1.59e-1)+	<b>2.1745e-1 (8.22e-2)</b>	1.2266e+0 (1.81e-1)+	1.6377e+0 (3.76e-1)+
	15	<i>1.6873e+0 (5.92e-1)</i>	2.7566e+0 (1.02e+0)+	4.1363e+0 (2.92e-1)+	9.7540e+0 (5.57e+0)+	2.4288e+0 (4.72e-1)+	<b>2.8143e-1 (4.26e-2)</b>	3.0670e+0 (5.56e-1)+	4.1154e+0 (1.29e+0)+
WFG4	8	<b>2.9589e+0 (1.16e-2)</b>	3.3180e+0 (8.88e-2)+	3.3240e+0 (1.59e-2)+	9.7270e+0 (2.35e-1)+	3.7913e+0 (3.65e-2)+	3.3984e+0 (8.98e-2)+	<i>3.1566e+0 (3.23e-2)+</i>	3.2451e+0 (4.80e-2)+
	10	4.5444e+0 (2.72e-2)	5.1554e+0 (1.46e-1)+	5.1149e+0 (7.98e-3)+	1.2294e+1 (1.31e+0)+	5.1493e+0 (3.67e-2)+	4.5639e+0 (1.12e-1)=	<i>4.4725e+0 (3.53e-2)-</i>	<b>4.3951e+0 (4.42e-2)</b>
	15	8.2538e+0 (1.52e-1)	<b>8.0695e+0 (3.22e-2)</b>	<i>8.0800e+0 (1.58e-2)-</i>	2.0723e+1 (4.26e+0)+	9.0139e+0 (1.01e-1)+	8.3584e+0 (2.25e-1)+	8.3736e+0 (6.90e-2)+	8.1105e+0 (1.87e-1)+
WFG5	8	<b>2.9405e+0 (7.18e-3)</b>	3.2600e+0 (2.02e-3)+	3.2673e+0 (6.03e-3)+	1.4390e+1 (7.28e-1)+	3.3457e+0 (3.04e-2)+	3.3580e+0 (9.27e-2)+	<i>3.1405e+0 (2.92e-2)+</i>	3.2665e+0 (5.21e-2)+
	10	<i>4.4722e+0 (1.47e-2)</i>	5.0743e+0 (3.01e-3)+	5.0786e+0 (6.15e-3)+	1.5922e+1 (4.83e+0)+	4.4963e+0 (3.00e-2)=	4.5972e+0 (1.21e-1)+	4.4726e+0 (3.49e-2)=	<b>4.4477e+0 (6.25e-2)</b>
	15	<i>7.9019e+0 (6.09e-2)</i>	7.9790e+0 (8.81e-2)=	8.0239e+0 (3.46e-2)=	2.8465e+1 (4.24e+0)+	<b>7.3101e+0 (9.30e-2)</b>	8.4038e+0 (1.86e-1)+	8.3472e+0 (7.91e-2)+	8.0933e+0 (2.54e-1)+
WFG6	8	<b>2.9626e+0 (7.97e-3)</b>	3.2745e+0 (1.12e-2)+	3.3200e+0 (2.10e-2)+	8.1576e+0 (4.01e+0)+	3.4765e+0 (5.71e-2)+	3.6664e+0 (1.56e-1)+	<i>3.2284e+0 (4.77e-2)+</i>	3.3761e+0 (7.84e-2)+
	10	<i>4.5529e+0 (1.34e-2)</i>	5.0811e+0 (7.35e-3)+	5.1043e+0 (1.73e-2)+	8.9743e+0 (4.85e+0)+	4.5985e+0 (4.88e-2)+	4.9777e+0 (1.38e-1)+	4.6105e+0 (3.98e-2)+	<b>4.5076e+0 (6.88e-2)</b>
	15	<i>8.1225e+0 (7.04e-2)</i>	8.1754e+0 (1.69e-1)+	8.1687e+0 (1.17e-1)-	2.0229e+1 (8.80e+0)+	<b>8.0980e+0 (6.29e-1)</b>	9.5278e+0 (3.94e-1)+	8.4303e+0 (1.08e-1)+	8.3076e+0 (7.74e-1)=
+/-/=		DTLZ	15/5/1	18/2/1	17/3/1	15/5/1	17/2/2	18/1/2	14/2/5
		WFG	14/2/2	13/4/1	18/0/0	9/3/6	12/4/2	13/2/3	12/4/2
		Total	29/7/3	31/6/2	35/3/1	24/8/7	29/6/4	31/3/5	26/6/7

degeneration PF, MPMO-BS only performs worse than Mo4Ma and obtains the second-best results on 10- and 15-objective instances. For WFG4 with multimodal characteristics, MPMO-BS still performs well and obtains the best IGD value on the 8-objective instance, although it is beaten by NSGA-II/SDR on 10- and 15-objective instances. WFG5 is a problem with deceptiveness, and MPMO-BS outperforms the other seven MOEAs on the 8-objective instance and is only beaten by NSGA-II/SDR on the 10-objective instance and by KnEA on the 15-objective instance.

Through the statistical test results both on IGD and HV, MPMO-BS shows excellent performance on DTLZ and WFG test problems. To better understand MPMO-BS, the distribution of the solutions obtained by each algorithm with the median IGD value on 15-objective DTLZ1 is shown in Fig. 4. From Fig. 4, it can be seen that SPEA/R, KnEA, Mo4Ma, and VMEF cannot approximate the PF well. The appearance of inferior solutions in SPEA/R may be due to the phenomenon of falling into local optima. The main idea of KnEA is to use knee points to enhance the search performance in MaOPs, while DTLZ1 is a problem with linear hyperplane PF but no knee

point. Therefore, it is impractical to solve DTLZ1 by KnEA. NSGA-III, MaOEA/IGD, and NSGA-II/SDR can converge well to the true PF, but they have poor performance in terms of diversity. In addition, it is worth noting that NSGA-II/SDR approximates only some parts (especially the center) of the PF and biases toward a few objectives, such as  $f_2$  and  $f_5$ . This may be because the PF of DTLZ1 is a hyperplane, while the dominance area shape of SDR resembles a curve bending outward. Such a dominance area can have a greater probability of eliminating boundary solutions and making the obtained solutions concentrated in the PF center. However, the solutions obtained by MPMO-BS can approximate the whole PF. Limited by space, the distribution of obtained solutions with the median IGD value on 15-objective WFG3 is shown in Fig. S.1 in the supplementary material. It can be seen that only the solutions obtained by MPMO-BS and Mo4Ma can approximate the true PF very well, although those by MPMO-BS show some deficiencies in diversity.

## 2) Results on Other Test Problems

To further validate the performance of MPMO-BS on problems with more complex PF shapes, comparison

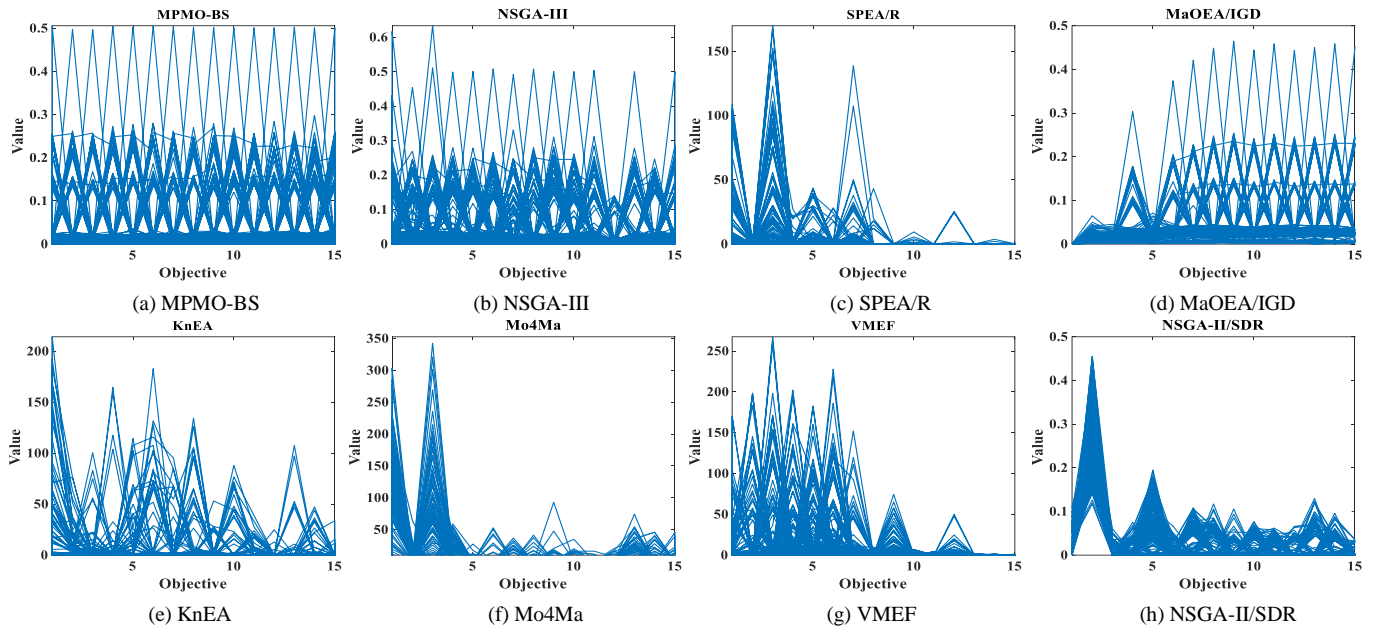


Fig. 4. Distribution of the obtained solutions with the median IGD value for the eight algorithms on DTLZ1 with 15 objectives.

TABLE III  
SIGNIFICANCE TEST BETWEEN MPMO-BS AND OTHER SEVEN MOEAs IN TERMS OF IGD ON MaF, DTLZ<sup>-1</sup>, CDTLZ2, AND SDTLZ1-2 PROBLEMS

Algorithm	NSGA-III	SPEA/R	MaOEA/IGD	KnEA	Mo4Ma	VMEF	NSGA-II/SDR
+/-/=	25/11/3	34/4/1	37/2/0	24/13/2	28/8/3	21/12/6	27/9/3

experiments are conducted on MaF, DTLZ<sup>-1</sup>, CDTLZ2, and SDTLZ1-2. The significance test results in terms of IGD are summarized in Table III, where the complete statistical results in terms of IGD and HV are given in Table S.IV and S.V in the supplementary material. From the results, we can see that MPMO-BS performs evidently better than NSGA-III, SPEA/R, and MaOEA/IGD, and shows strong competitiveness compared with KnEA, Mo4Ma, VMEF, and NSGA-II/SDR.

MaF1 and MaF4 are two inverted DTLZ variants. MPMO-BS does not perform the best on these two problems but is still promising. The PF shapes of these problems are inverted, which means that the population tends to focus on one boundary point when only optimizing one objective. This creates a significant challenge for MPMO-BS in diversity maintenance when dealing with these kinds of problems. MaF2 is a DTLZ2 variant with more difficulty in convergence, and MPMO-BS performs better than the others except for KnEA on this problem. MaF3, MaF5, and CDTLZ2 are three problems with convex PF shapes. MPMO-BS achieves the best or second-best performance on these three problems. This indicates that MPMO-BS is capable of handling these problems with convex PF. Fig. S.2 in the supplementary material gives the distribution of obtained solutions on the 15-objective MaF5 instance, in which only MPMO-BS, Mo4Ma, and VMEF obtain the solutions with good performance both in convergence and diversity. Similar to DTLZ5, MaF6 is a problem for examining whether an algorithm can deal with degenerated PF. From Table S.IV and S.V, we can see that MPMO-BS performs worse than Mo4Ma and NSGA-II/SDR on the two metrics. Fig. S.3 in the supplementary material gives the distribution of the obtained solutions on the 15-objective MaF6 instance. As seen

from it, there are still many solutions obtained by VMEF and NSGA-II/SDR that do not converge to the PF, while solutions obtained by MPMO-BS have converged well with poor diversity. SDTLZ1-2 are two problems with scaled PF shapes, and MPMO-BS obtains promising performance on most instances of these two problems. Fig. S.4 in the supplementary material gives the distribution of the obtained solutions on the 15-objective SDTLZ1 instance.

Moreover, we further verify the performance of MPMO-BS on DTLZ<sup>-1</sup> problems with inverted triangular PF shapes. The results from Table S.IV and S.V show that MPMO-BS is still competitive among all compared algorithms, although it may not perform particularly well on some instances (i.e., DTLZ4<sup>-1</sup> instances), whereas it is still competitive among all the compared algorithms. Specifically, MPMO-BS can beat NSGA-III, SPEA/R, MaOEA/IGD, Mo4Ma, and VMEF on most DTLZ1-3<sup>-1</sup> instances. Overall, the performance of MPMO-BS on DTLZ<sup>-1</sup> is generally promising, although the advantage is not so significant as that on DTLZ. The reasons may be twofold: (1) MPMO-BS uses reference vectors to maintain diversity, while problems with inverted PF shapes are very challenging for these reference vector-based algorithms in diversity maintenance. For this point, more analysis can be found in [71]. (2) DTLZ<sup>-1</sup> can be viewed as a DTLZ version with maximized objectives. Therefore, the gap between different objectives gradually widens with convergence. This makes it more difficult to share information among different populations and poses a challenge for ELS to improve diversity. Fig. S.5 and S.6 in the supplementary material give the distributions of the obtained solutions on the 8-objective DTLZ1<sup>-1</sup> and DTLZ2<sup>-1</sup> instances. It can be seen that all

TABLE IV  
SIGNIFICANCE TEST BETWEEN MPMO-BS AND ITS VARIANTS IN TERMS OF IGD ON DTLZ AND MaF PROBLEMS

Problem	1objS	noBS	noAS	noACF	noELS	noReA
DTLZ	17/1/3	14/5/2	17/1/3	11/1/9	18/2/1	16/3/2
MaF	14/1/3	10/2/6	12/2/4	4/0/14	11/1/6	12/0/6
+/-/=	31/2/6	24/7/8	29/3/7	15/1/23	29/3/7	28/3/8

algorithms have poor diversity performance on these instances to varying degrees, whereas MPMO-BS is competitive in both convergence and diversity.

In summary, MPMO-BS can be applied to deal with MaOPs with different characteristics. The statistical results from the comparison experiments show the high competitiveness and robustness of MPMO-BS.

#### D. Contribution of Archive

In this section, we discuss the contribution of the archive adopted in MPMO-BS. We design a variant of MPMO-BS (named noArch) and conduct comparison experiments between it and MPMO-BS. The noArch is a variant that replaces archive  $A$  with the set of populations  $P$  after environmental selection at each generation. In other words, all operations in **Algorithm 4** related to  $A$ , e.g., ELS in Line 23, solution preservation in Line 26, and population reallocation in Line 29, are replaced by using  $P$ .

Table S.VI in the supplementary material shows the mean and standard deviation IGD values given by noArch and MPMO-BS on DTLZ and MaF test suites, where the best is marked in **boldface**. It can be seen that MPMO-BS performs better than noArch on almost all instances. The experimental results confirm that the adoption of the archive can significantly improve the performance of MPMO-BS. The reason for this difference is that the optimal solutions of all generations up to now are preserved in the archive. This is equivalent to an elite reservation strategy.

Fig. S.7 in the supplementary material plots the obtained solution set by noArch and MPMO-BS on 10-objective DTLZ2. It can be seen that the diversity of the solution set obtained by MPMO-BS is much better than that obtained by noArch. This shows the role of the archive in promoting diversity.

#### E. Analysis of Each Strategy

In this section, the contribution of each strategy in MPMO-BS is exhibited. Adaptive sorting, BS, ACF, ELS, and population reallocation strategies are the core of MPMO-BS. Accordingly, we design six variants with different combinations of the strategies:

- 1objS: variant that adopts the one-objective sorting method, and a random selection strategy is used to replace the ACF strategy.
- noBS: variant that only adopts the NDS, and a random selection strategy is used to replace the ACF strategy.
- noAS: variant that only adopts the BS. In other words, the adaptive sorting strategy is abandoned. In addition, the ACF strategy is replaced by a random selection strategy.
- noACF: a random selection strategy is used to replace the ACF strategy in MPMO-BS.

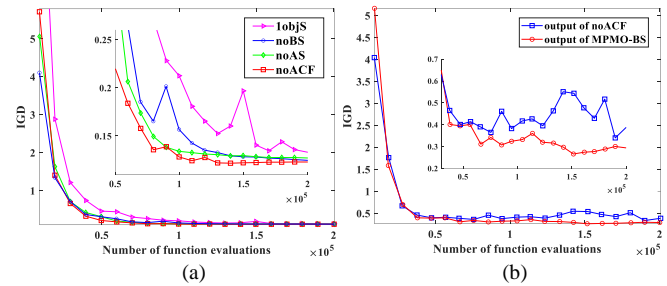


Fig. 5. The mean IGD curves of (a) 1objS, noBS, noAS, and noACF on DTLZ1 with 10 objectives, and (b) output of noACF and MPMO-BS on DTLZ6 with 10-objectives.

- noELS: variant that without ELS.
- noReA: variant that without population reallocation strategy.

The variants are compared with MPMO-BS on DTLZ and MaF test suites. The IGD and HV value details are presented in Table S.VII and Table S.VIII in the supplementary material. In this paper, only the significance test results in terms of the IGD metric are summarized in Table IV, which is based on Wilcoxon's rank-sum test at a 5% significance level. From Table IV, it can be seen that MPMO-BS outperforms all its variants. Specifically, 1objS, noBS, noAS, noACF, noELS, and noReA are beaten by MPMO-BS on 17, 14, 17, 11, 18, and 16 out of 21 DTLZ instances, and on 14, 10, 12, 4, 11, and 12 out of 18 MaF instances. This comparison result shows that the BS, adaptive sorting, ELS, and population reallocation strategies play significant roles in improving the performance of the MPMO-BS. On the one hand, the comparison among 1objS, noBS, and noAS shows that the performance of noAS is somewhere between that of 1objS and noBS. This result confirms our previous positioning of BS, which can be considered a sorting method between the one-objective sorting and NDS. On the other hand, the comparison among noBS, noAS, and noACF shows that the use of ACF strategies may not make a large difference but can slightly improve the performance of MPMO-BS.

Here, we choose the 15-objective DTLZ1 instance and compare the performance of these variants on it. Fig. S.8 in the supplementary material shows the distribution of the obtained solution set with the median IGD values for MPMO-BS and its variants. In Fig. S.8, it can be seen that MPMO-BS is the best-performing algorithm whether in terms of convergence or diversity. However, noBS does not perform well both in terms of convergence and diversity, whereas 1objS and noAS only have defects in diversity. This further confirms the role of BS or one-objective sorting in convergence maintenance. However, the comparison between 1objS and noAS shows that the solutions obtained by 1objS tend to favor some objectives more seriously, resulting in worse diversity performance. In addition, noACF performs better than both noBS and noACF but slightly worse than MPMO-BS. Solutions obtained by noELS and noReA have a serious deficiency in diversity. These results suggest that they play an important role in diversity maintenance for MPMO-BS.

##### 1) Analysis of Adaptive Sorting Strategy

BS is a strategy that can accelerate convergence. The comparison between noBS and noAS shows that BS alone

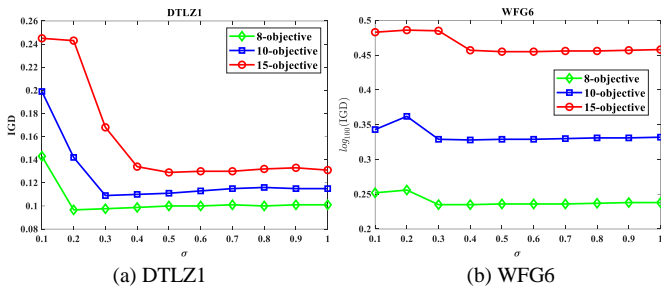


Fig. 6. IGD values of MPMO-BS with different setting for  $\sigma$ , averaging over 20 independent runs.

cannot promote performance heavily since BS and NDS are complementary. Meanwhile, the adaptive sorting strategy is considered a balanced strategy between different sorting methods, and it can improve the performance of MPMO-BS by switching the sorting method between BS and NDS. The cooperation of the two can promote the performance of MPMO-BS to the best.

To illustrate this, we plot the mean IGD curves of 1objS, noBS, noAS, and noACF on 10-objective DTLZ1, as shown in Fig. 5(a). From Fig. 5(a), it can be seen that the IGD of 1objS is always the worst since the IGD is a metric that comprehensively considers convergence and diversity. However, the one-objective sorting method is only excellent in convergence, whereas diversity maintenance is ignored. In addition, the IGD values by the two algorithms with BS (noAS and noACF) can drop faster than noBS at the early search stage. More specifically, noAS and noACF can reduce IGD values to approximately 0.14 by 50,000 fitness evaluations, while the IGD value of noBS is still above 0.16 at that time. This illustrates that the use of BS can accelerate convergence in the search process. Meanwhile, noBS gradually narrows the IGD gap with noAS and surpasses it. However, noACF is almost always ahead in the IGD value, showing that NDS is more advantageous than BS as the population approximates the true PF. In other words, noACF, the variant adopting the adaptive sorting strategy, can choose the better sorting method appropriately to promote the search capability. The curve comparison of Fig. 5(a) further confirms that the adaptive sorting strategy with BS is one of the keys to the high performance of MPMO-BS.

### 2) Analysis of ACF Strategy

ACF is an auxiliary strategy that can further distinguish the convergence status among solutions that are at the same sorting front to increase the selection pressure. To investigate the influence of ACF on the performance of MPMO-BS, we use MPMO-BS and its variant noACF to conduct comparison experiments on 10-objective DTLZ6. Note that only the set of all populations  $P$  after environmental selection is taken as the output at the end of every generation. However, archive  $A$  is not removed, which is different from the variant noArch mentioned above. This is done to eliminate the influence of other strategies on the results as much as possible, and more clearly explain the role of ACF in MPMO-BS.

Fig. 5(b) shows the mean IGD curves of the output by MPMO-BS and noACF. In Fig. 5(b), the IGD curves of MPMO-BS and noACF both decrease rapidly to below 1 in very few evaluations. However, the curve of MPMO-BS drops

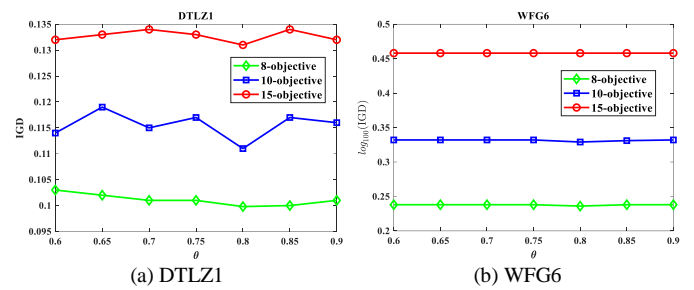


Fig. 7. IGD values of MPMO-BS with different setting for  $\theta$ , averaging over 20 independent runs.

faster than that of noACF. This shows that MPMO-BS converges faster than noACF because of the ACF strategy. In addition, the IGD curve of noACF fluctuates greatly compared with that of MPMO-BS. This is because, in noACF, solutions are randomly selected but not through the ACF value when turning to the second condition in the environmental selection. This random selection method may retain solutions with poor convergence. Through the IGD curve comparison, we can see that the ACF promotes the performance of MPMO-BS, although its effect is slight.

## F. Sensitivity Analysis of Parameter Settings

### 1) Sensitivity Analysis of ELS

ELS was first designed to jump out of the local optima, and it has shown its effectiveness in diversity maintenance. According to our previous research experience in [75], the effectiveness of ELS is affected by the value of  $\sigma$ . In this subsection, we investigate the sensitivity of parameter  $\sigma$  in ELS. The value of  $\sigma$  varies from 0.1 to 1. Because of the space limitation, Fig. 6 only shows the curves of different IGD values for different  $\sigma$  settings on DTLZ1 and WFG6 with different objectives, averaging over 20 independent runs. Note that  $\log_{10}(\cdot)$  function is appended to the IGD calculation on WFG6 instances for easy visual comparison. It can be seen that the performance of MPMO-BS is heavily affected by the value of  $\sigma$  on DTLZ1, whereas it is only slightly affected on WFG6. This indicates that the sensitivity of the  $\sigma$  parameter is not fixed for different optimization problems. However,  $\sigma$  values that are too large or too small are inappropriate for ELS since the IGD curves appear to be concave. In Fig. 6(a), the optimal setting for  $\sigma$  gets larger from 0.2 to 0.5 with the increase of objectives. In Fig. 6(b), there is no obvious correlation between the optimal setting of  $\sigma$  and the number of objectives, whereas the optimal  $\sigma$  value is always approximately 0.4. More sensitivity investigation results on other problems can be seen in Fig. S.9 in the supplementary material. Through the experimental comparison in Fig. 6 and Fig. S.9, we can conclude that  $\sigma$  in ELS can be set to between 0.2 and 0.7, and a larger  $\sigma$  seems to be appropriate for problems with more objectives. In this paper, we set  $\sigma$  to 0.5.

### 2) Sensitivity Analysis of Threshold $\theta$

The threshold  $\theta$  is a trigger that determines the conversion from BS to NDS in the adaptive sorting strategy. In this subsection, we investigate the sensitivity of  $\theta$ , which varies from 0.6 to 0.9. Fig. 7 shows the IGD curves for different  $\theta$  settings on DTLZ1 and WFG6 with different objectives, averaging over 20 independent runs. Fig. 7 shows that the performance of MPMO-BS is not particularly sensitive to the value of  $\theta$ . It can



TABLE V  
MEAN AND STANDARD DEVIATION HV VALUES OBTAINED BY EIGHT ALGORITHMS ON TWO REAL-WORLD APPLICATION PROBLEMS

Reference point	Problem	MPMO-BS	NSGA-III	SPEA/R	MaOEA/IGD	KnEA	Mo4Ma	VMEF	NSGA-II/SDR
$(1.1, \dots, 1.1)^T$	water resource planning	<b>1.5052e+0</b> <b>(4.16e-3)</b>	1.4939e+0 (7.20e-3) +	1.4597e+0 (6.27e-3) +	1.3027e+0 (2.42e-2) +	1.4823e+0 (7.99e-3) +	1.4990e+0 (8.11e-3) +	1.4959e+0 (5.05e-3) +	1.4087e+0 (2.78e-2) +
	car cab design	2.8926e-1 (1.11e-2)	2.4236e-1 (1.63e-2) +	2.8107e-1 (1.45e-2) +	1.2878e-1 (1.16e-2) +	2.1787e-1 (1.49e-2) +	<b>3.4273e-1</b> <b>(8.76e-3)</b> -	3.1099e-1 (8.03e-3) -	3.4374e-2 (3.77e-3) +
$(1.25, \dots, 1.25)^T$	water resource planning	<b>3.3610e+0</b> <b>(7.29e-3)</b>	3.3363e+0 (1.50e-2) +	3.2794e+0 (1.21e-2) +	2.9969e+0 (4.31e-2) +	3.3223e+0 (1.30e-2) +	3.3507e+0 (1.47e-2) +	3.3435e+0 (8.88e-3) +	3.1789e+0 (5.61e-2) +
$(4/3, \dots, 4/3)^T$	car cab design	2.9567e+0 (1.00e-1)	2.5013e+0 (1.49e-1) +	2.8412e+0 (1.26e-1) +	1.4666e+0 (1.10e-1) +	2.3079e+0 (1.38e-1) +	<b>3.2762e+0</b> <b>(7.64e-2)</b> -	3.0338e+0 (6.09e-2) -	5.4406e-1 (4.25e-2) +

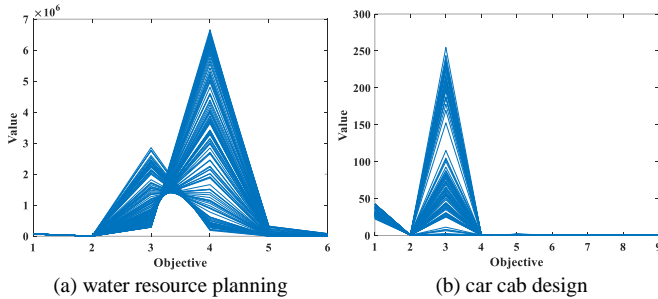


Fig. 8. Distribution of the obtained solutions by MPMO-BS with the median HV value ( $\mathbf{z} = (1.1, \dots, 1.1)^T$ ) on two real-world application problems.

be seen that all IGD curves show a concave shape with less obvious curvature, and a suitable setting range for  $\theta$  is between 0.7 and 0.85. Such situations can also be seen on other problems such as DTLZ4 and WFG8 (Fig. S.10 in the supplementary material). However, the performance of MPMO-BS is very sensitive to  $\theta$  on other problems, such as MaF6 shown in Fig. S.10(c), and the suitable setting range for  $\theta$  is between 0.8 and 0.85. Considering the above analysis, we set  $\theta$  to 0.8 in this paper.

### G. Experiments on Real-world Problems

In this subsection, we try to investigate the performance of MPMO-BS in dealing with real-world MaOPs. Two application problems (named water resource planning and car cab design) derived from the real world are adopted herein, which were proposed in [76]. Some properties of these two problems and the parameter settings for all compared algorithms are given in Table S.IX, and the statistical results in terms of HV are given in Table V. Note that since the true PF is unknown, before HV calculation, the non-dominated solutions obtained by all algorithms after 30 independent runs are formed into a set, and the final solutions of each algorithm are normalized by the ideal and nadir points of this set. We first set the reference point  $\mathbf{z}$  as  $(1.1, \dots, 1.1)^T$  for HV calculation, just like in the above experiments. To make the experimental results more comprehensive, we also use the reference point setting method proposed in [77] for HV calculation. According to this method,  $\mathbf{z}$  is set as  $(1.25, \dots, 1.25)^T$  and  $(4/3, \dots, 4/3)^T$  on water resource planning and car cab design, respectively. In addition, the IGD metric is not adopted since the true PF of these two real-world problems is unknown.

From Table V we can see that MPMO-BS obtains the best and third-best HV results on water resource planning and car cab design problems, respectively. These results show that our proposed algorithm has the potential to solve real-world ap-

plication problems. Fig. 8 plots the distribution of the solutions obtained by MPMO-BS with the median HV value on these problems when  $\mathbf{z} = (1.1, \dots, 1.1)^T$ , and those obtained by the other algorithms are given in Fig. S.11 and S.12 in the supplementary material, respectively.

## V. CONCLUSION

This paper focuses on the MPMO framework and proposes a multi-population coevolutionary MOEA for many-objective optimization. The proposed MPMO-BS algorithm uses different sorting methods to boost population convergence depending on the convergence status. The BS method is employed for accelerating convergence at the early search stage while switching to the NDS method at the later search stage for all objective equality. For those solutions at the same sorting front, the ACF strategy can further distinguish their convergence status. At environmental selection, well-converged solutions are preserved according to two conditions: 1) at the lower sorting front, and 2) with a larger ACF value.

In addition, with the assistance of the MPMO framework and the ELS in diversity maintenance, MPMO-BS can remain solutions with good convergence and diversity performance. The performance of MPMO-BS is further improved by incorporating a population reallocation strategy, which strengthens communication and information sharing among populations.

To assess the performance of MPMO-BS, the experiments and comparisons have been conducted on 29 test problems with 5, 8, and 10 objectives. The results confirm the effectiveness and robustness of the MPMO-BS in solving MaOPs. The results on water resource planning and car cab design problems also show its effectiveness in solving real-world application problems.

## REFERENCES

- [1] Z.-H Zhan, L. Shi, K. C. Tan, and J. Zhang, "A survey on evolutionary computation for complex continuous optimization," *Artif. Intell. Rev.*, vol. 55, pp. 59–110, 2022.
- [2] L. M. Antonio and C. A. C. Coello, "Coevolutionary multiobjective evolutionary algorithms: Survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 22, no. 6, pp. 851–865, 2018.
- [3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.
- [4] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," *TIK-report*, vol. 103, 2001.
- [5] E. Zitzler and S. Kunzli, "Indicator-based selection in multiobjective search," in *Proc. Para. Problem Solv. Nature*, 2004, pp. 832–842.
- [6] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no.

- 6, pp. 712–731, 2007.
- [7] Z.-H. Zhan, J. Li, J. Cao, J. Zhang, H. S.-H. Chung, and Y.-H. Shi, “Multiple populations for multiple objectives: A coevolutionary technique for solving multiobjective optimization problems,” *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 445–463, 2013.
- [8] Q. Yang, Z. Wang, and H. Ishibuchi, “It is hard to distinguish between dominance resistant solutions and extremely convex Pareto optimal solutions,” in *Proc. Evol. Multi-Criterion Optim.*, 2021, pp. 3–14.
- [9] Z. Wang, Q. Li, Q. Yang, and H. Ishibuchi, “The dilemma between eliminating dominance-resistant solutions and preserving boundary solutions of extremely convex Pareto fronts,” *Complex Intell. Syst.*, 2021, DOI: 10.1007/s40747-021-00543-2.
- [10] H. Ishibuchi, L. He, and K. Shang, “Regular Pareto front shape is not realistic,” in *Proc. Conf. Evol. Comput.*, 2019, pp. 2034–2041.
- [11] L. He, A. Camacho, and H. Ishibuchi, “Another difficulty of inverted triangular Pareto fronts for decomposition-based multi-objective algorithms,” in *Proc. Genet. Evol. Comput.*, 2020, pp. 498–506.
- [12] K. Shang, H. Ishibuchi, L. He, and L. M. Pang, “A survey on the hypervolume indicator in evolutionary multiobjective optimization,” *IEEE Trans. Evol. Comput.*, vol. 25, no. 1, pp. 1–20, 2021.
- [13] X.-F. Liu, Z.-H. Zhan, Y. Gao, J. Zhang, S. Kwong, and J. Zhang, “Coevolutionary particle swarm optimization with bottleneck objective learning strategy for many-objective optimization,” *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 587–602, 2019.
- [14] H. Liu, L. Chen, K. Deb, and E. D. Goodman, “Investigating the effect of imbalance between convergence and diversity in evolutionary multiobjective algorithms,” *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 408–425, 2017.
- [15] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, “Combining convergence and diversity in evolutionary multi-objective optimization,” *Evol. Comput.*, vol. 10, no. 3, pp. 263–282, 2002.
- [16] K. Ikeda, H. Kita, and S. Kobayashi, “Failure of Pareto-based MOEAs: Does non-dominated really mean near to optimal?” in *Proc. Cong. Evol. Comput.*, vol. 2, 2001, pp. 957–962.
- [17] F. Di Pierro, S.-T. Khu, and D. A. Savic, “An investigation on preference order ranking scheme for multi-objective evolutionary optimization,” *IEEE Trans. Evol. Comput.*, vol. 11, no. 1, pp. 17–45, 2007.
- [18] Y. Yuan, H. Xu, B. Wang, and X. Yao, “A new dominance relation-based evolutionary algorithm for many-objective optimization,” *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 16–37, 2016.
- [19] H. Sato, H. Aguirre, and K. Tanaka, “Controlling dominance area of solutions and its impact on the performance of MOEAs,” in *Proc. Evol. Multi-Criterion Optim.*, 2007, pp. 5–20.
- [20] Z. He, G. G. Yen, and J. Zhang, “Fuzzy-based Pareto optimality for many-objective evolutionary algorithms,” *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 269–285, 2014.
- [21] X. Zou, Y. Chen, M. Liu, and L. Kang, “A new evolutionary algorithm for solving many-objective optimization problems,” *IEEE Trans. Syst., Man, and Cybern. -Part B*, vol. 38, no. 5, pp. 1402–1412, 2008.
- [22] M. Farina and P. Amato, “A fuzzy definition of ‘optimality’ for many-criteria optimization problems,” *IEEE Trans. Syst., Man, and Cybern. -Part A*, vol. 34, no. 3, pp. 315–326, 2004.
- [23] S. Yang, M. Li, X. Liu, and J. Zheng, “A grid-based evolutionary algorithm for many-objective optimization,” *IEEE Trans. Evol. Comput.*, vol. 17, no. 5, pp. 721–736, 2013.
- [24] Y. Tian, R. Cheng, X. Zhang, Y. Su, and Y. Jin, “A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization,” *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 331–345, 2019.
- [25] K. Deb and H. Jain, “An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, part I: solving problems with box constraints,” *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, 2014.
- [26] K. Li, K. Deb, Q. Zhang, and S. Kwong, “An evolutionary many-objective optimization algorithm based on dominance and decomposition,” *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 694–716, 2015.
- [27] Y. Yuan, H. Xu, B. Wang, B. Zhang, and X. Yao, “Balancing convergence and diversity in decomposition-based many-objective optimizers,” *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 180–198, 2016.
- [28] M. Wu, K. Li, S. Kwong, and Q. Zhang, “Evolutionary many-objective optimization based on adversarial decomposition,” *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 753–764, 2020.
- [29] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, “Stable matching-based selection in evolutionary multiobjective optimization,” *IEEE Trans. Evol. Comput.*, vol. 18, no. 6, pp. 909–923, 2014.
- [30] R. Wang, Q. Zhang, and T. Zhang, “Decomposition-based algorithms using Pareto adaptive scalarizing methods,” *IEEE Trans. Evol. Comput.*, vol. 20, no. 6, pp. 821–837, 2016.
- [31] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, “A reference vector guided evolutionary algorithm for many-objective optimization,” *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, 2016.
- [32] A. M. Mendez and C. A. C. Coello, “GDE-MOEA: a new MOEA based on the generational distance indicator and  $\epsilon$ -dominance,” in *Proc. Cong. Evol. Comput.*, 2015, pp. 947–955.
- [33] H. Gómez, Raquel, and C. A. C. Coello, “Improved metaheuristic based on the R2 indicator for many-objective optimization,” in *Proc. Genet. Evol. Comput.*, 2015, pp. 679–686.
- [34] Y. Tian, R. Cheng, X. Zhang, F. Cheng, and Y. Jin, “An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility,” *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 609–622, 2018.
- [35] J. Bader and E. Zitzler, “HypE: An algorithm for fast hypervolume-based many-objective optimization,” *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, 2011.
- [36] K. Shang and H. Ishibuchi, “A new hypervolume-based evolutionary algorithm for many-objective optimization,” *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 839–852, 2020.
- [37] Y. Sun, G. G. Yen, and Z. Yi, “IGD indicator-based evolutionary algorithm for many-Objective optimization problems,” *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 173–187, 2019.
- [38] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, “Performance assessment of multiobjective optimizers: An analysis and review,” *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, 2003.
- [39] S. -H. Wu, Z. -H. Zhan, and J. Zhang, “SAFE: Scale-adaptive fitness evaluation method for expensive optimization problems,” *IEEE Trans. Evol. Comput.*, vol. 25, no. 3, pp. 478–491, 2021.
- [40] J. -Y. Li, Z. -H. Zhan, C. Wang, H. Jin, and J. Zhang, “Boosting data-driven evolutionary algorithm with localized data generation,” *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 923–937, 2020.
- [41] S. -C. Liu, Z. -G. Chen, Z. -H. Zhan, S. -W. Jeon, S. Kwong, and J. Zhang, “Many-objective job shop scheduling: A multiple populations for multiple objectives-based genetic algorithm approach,” *IEEE Trans. Cybern.*, 2021, DOI: 10.1109/TCYB.2021.3102642.
- [42] X. Zhang, Z. -H. Zhan, W. Fang, P. Qian, and J. Zhang, “Multi population ant colony system with knowledge based local searches for multiobjective supply chain configuration,” *IEEE Trans. Evol. Comput.*, vol. 26, no. 3, pp. 512–526, 2021.
- [43] J. -Y. Li, *et al.*, “A multipopulation multiobjective ant colony system considering travel and prevention costs for vehicle routing in COVID-19-like epidemics,” *IEEE Trans. Intelli. Transp. Syst.*, 2022, DOI:10.1109/ITITS.2022.3180760.
- [44] Z. -G. Chen, *et al.*, “Multiobjective cloud workflow scheduling: A multiple populations ant colony system approach,” *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 2912–2926, 2019.
- [45] S. -Z. Zhou, Z. -H. Zhan, Z. -G. Chen, S. Kwong, and J. Zhang, “A multi-objective ant colony system algorithm for airline crew rostering problem with fairness and satisfaction,” *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 6784–6798, 2021.
- [46] J. Kennedy, and E. Russell, “Particle swarm optimization,” in *Proc. Neur. Net.*, 1995, vol. 4, pp. 1942–1948.
- [47] J. -R. Jian, Z. -G. Chen, Z. -H. Zhan, and J. Zhang, “Region encoding helps evolutionary computation evolve faster: A new solution encoding scheme in particle swarm for large-scale optimization,” *IEEE Trans. Evol. Comput.*, vol. 25, no. 4, pp. 779–793, 2021.
- [48] X. Xia, *et al.*, “Triple archives particle swarm optimization,” *IEEE Trans. Cybern.*, vol. 50, no. 12, pp. 4862–4875, 2020.
- [49] K. Price, R. M. Storn, and J. A. Lampinen, “Differential evolution: A practical approach to global optimization,” *Natural Comput. Seca.*, 2005.
- [50] Z. -H. Zhan, Z. -J. Wang, H. Jin, and J. Zhang, “Adaptive distributed differential evolution,” *IEEE Trans. Cybern.*, vol. 50, no. 11, pp. 4633–4647, 2020.
- [51] Z. -J. Wang, J. -R. Jian, Z. -H. Zhan, Y. Li, S. Kwong, and J. Zhang, “Gene targeting differential evolution: A simple and efficient method for large scale optimization,” *IEEE Trans. Evol. Comput.*, 2022, DOI:10.1109/TEVC.2022.3185665.
- [52] Z. -J. Wang, *et al.*, “Dual-strategy differential evolution with affinity propagation clustering for multimodal optimization problems,” *IEEE*



- Trans. Evol. Comput.*, vol. 22, no. 6, pp. 894–908, 2018.
- [53] S. Jiang and S. Yang, “A strength Pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization,” *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 329–346, 2017.
- [54] X. Zhang, Y. Tian, and Y. Jin, “A knee point-driven evolutionary algorithm for many-objective optimization,” *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 761–776, 2015.
- [55] R. Wang, R. C. Purshouse, and P. J. Fleming, “Preference-inspired co-evolutionary algorithms for many-objective optimization,” *IEEE Trans. Evol. Comput.*, vol. 17, no. 4, pp. 474–494, 2013.
- [56] S.-C. Liu, Z.-H. Zhan, K. C. Tan, and J. Zhang, “A multiobjective framework for many-objective optimization,” *IEEE Trans. Cybern.*, 2021, DOI: 10.1109/TCYB.2021.3082200.
- [57] W. Qiu, *et al.*, “Ensemble many-objective optimization algorithm based on voting mechanism,” *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 52, no. 3, pp. 1716–1730, 2022.
- [58] D. E. Goldberg, “Genetic algorithms in search, optimization, and machine learning,” *Addison Wesley*, 1988.
- [59] S. Tang, Z. Cai, and J. Zheng, “A fast method of constructing the non-dominated set: arena’s principle,” in *Proc. Natural Comput.*, vol. 1, 2008, pp. 391–395.
- [60] H. Fang, Q. Wang, Y.-C. Tu, and M. F. Horstemeyer, “An efficient non-dominated sorting method for evolutionary algorithms,” *Evol. Comput.*, vol. 16, no. 3, pp. 355–384, 2008.
- [61] M. Drozd’ik, Y. Akimoto, H. Aguirre, and K. Tanaka, “Computational cost reduction of nondominated sorting using the m-front,” *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 659–678, 2014.
- [62] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, “An efficient approach to nondominated sorting for evolutionary multiobjective optimization,” *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 201–213, 2015.
- [63] K. McClymont and E. Keedwell, “Deductive sort and climbing sort: New methods for non-dominated sorting,” *Evol. Comput.*, vol. 20, no. 1, pp. 1–26, 2012.
- [64] X. Zhang, Y. Tian, and Y. Jin, “Approximate non-dominated sorting for evolutionary many-objective optimization,” *Info. Sci.*, vol. 369, pp. 14–33, 2016.
- [65] J. Yuan, H.-L. Liu, and F. Gu, “A cost value based evolutionary many-objective optimization algorithm with neighbor selection strategy,” in *Proc. Conf. Evol. Comput.*, 2018, pp. 1–8.
- [66] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, “PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum],” *IEEE Comput. Int. Mag.*, vol. 12, no. 4, pp. 73–87, 2017.
- [67] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable test problems for evolutionary multiobjective optimization,” in *Proc. Evol. Multi-Criterion Optim.* Springer, 2005, pp. 105–145.
- [68] S. Huband, P. Hingston, L. Barone, and L. While, “A review of multi-objective test problems and a scalable test problem toolkit,” *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, 2006.
- [69] R. Cheng, *et al.*, “A benchmark test suite for evolutionary many-objective optimization,” *Compl. & Intelli. Syst.*, vol. 3, no. 1, pp. 67–81, 2017.
- [70] L. Pan, L. Li, C. He, and K. C. Tan, “A subregion division-based evolutionary algorithm with effective mating selection for many-objective optimization,” *IEEE Trans. Cybern.*, vol. 50, no. 8, pp. 3477–3490, 2020.
- [71] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, “Performance of decomposition-based many-objective algorithms strongly depends on Pareto front shapes,” *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 169–190, 2017.
- [72] K. Deb, *et al.*, “Simulated binary crossover for continuous search space,” *Compl. Syst.*, vol. 9, no. 2, pp. 115–148, 1995.
- [73] K. Deb and M. Goyal, “A combined genetic adaptive search for engineering design,” *Comput. Sci. Info.*, vol. 26, pp. 30–45, 1996.
- [74] F. Wilcoxon, “Individual comparisons by ranking methods,” in *Break. in Statist.* Springer, 1992, pp. 196–202.
- [75] Z.-H. Zhan, J. Zhang, Y. Li, and H. S.-H. Chung, “Adaptive particle swarm optimization,” *IEEE Trans. Syst., Man, and Cybern. B, Cybern.*, vol. 39, no. 6, pp. 1362–1381, 2009.
- [76] R. Tanabe and H. Ishibuchi, “An easy-to-use real-world multi-objective optimization problem suite,” *Appli. Soft Comput.*, vol. 89, 2020, DOI: 10.1016/j.asoc.2020.106078.
- [77] H. Ishibuchi, R. Imada, Y. Setoguchi, and Y. Nojima, “How to specify a reference point in hypervolume calculation for fair performance comparison,” *Evol. Comput.*, vol. 26, no. 3, pp. 411–440, 2018.



**Qi-Te Yang** (Student Member, IEEE) received the B.S. degree from Dalian University, Dalian, China, in 2017, and the M.S. degree from Xiangtan University, Xiangtan, China, in 2020. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China.

His research interests mainly include computational intelligence, data-driven optimization, multiobjective optimization, and their applications in real-world problems. He has been invited as a reviewer of the *Swarm and Evolutionary Computation* and the *Neurocomputing*.



**Zhi-Hui Zhan** (Senior Member, IEEE) received the Bachelor’s degree and the Ph. D. degree in Computer Science from the Sun Yat-Sen University, Guangzhou China, in 2007 and 2013, respectively.

He is currently the Changjiang Scholar Young Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China.

His current research interests include evolutionary computation, swarm intelligence, and their applications in real-world problems and in environments of cloud computing and big data.

Dr. Zhan was a recipient of the IEEE Computational Intelligence Society (CIS) Outstanding Early Career Award in 2021, the Outstanding Youth Science Foundation from National Natural Science Foundations of China (NSFC) in 2018, and the Wu Wen-Jun Artificial Intelligence Excellent Youth from the Chinese Association for Artificial Intelligence in 2017. He is one of the World’s Top 2% Scientists for both Career-Long Impact and Year Impact in Artificial Intelligence and one of the Highly Cited Chinese Researchers in Computer Science. He is currently an Associate Editor of the *IEEE Transactions on Evolutionary Computation*, the *Neurocomputing*, the *Memetic Computing*, and the *Machine Intelligence Research*.



**Sam Kwong** (Fellow, IEEE) received the Ph.D. degree from the University of Hagen, Germany, in 1996.

He is currently a Chair Professor with the Department of Computer Science, City University of Hong Kong. His research interests include pattern recognition, evolutionary computations, and video analytics.

Prof. Kwong was elevated to an IEEE Fellow for his contributions to optimization techniques for cybernetics and video coding in 2014. He is the President of the IEEE Systems, Man, and Cybernetics (SMC). He was also appointed as an IEEE Distinguished Lecturer of the IEEE SMC Society in March 2017. He is currently an Associate Editor of the *IEEE Transactions on Evolutionary Computation*.



**Jun Zhang** (Fellow, IEEE) received the Ph.D. degree from the City University of Hong Kong SAR in 2002.

He is currently a Korea Brain Pool Fellow Professor with Hanyang University, South Korea. His current research interests include computational intelligence, cloud computing, operations research, and power electronic circuits. He has published over more than 150 IEEE Transactions papers in his research areas.

Dr. Zhang was a recipient of the Changjiang Chair Professor from the Ministry of Education, China, in 2013, The National Science Fund for Distinguished Young Scholars of China in 2011 and the First-Grade Award in Natural Science Research from the Ministry of Education, China, in 2009. He is currently an Associate Editor of the *IEEE Transactions on Evolutionary Computation* and the *IEEE Transactions on Cybernetics*.