

Use of Two Penalty Values in Multi-objective Evolutionary Algorithm based on Decomposition

Lie Meng Pang, *Member, IEEE*, Hisao Ishibuchi, *Fellow, IEEE*, and Ke Shang, *Member, IEEE*

Abstract—The multi-objective evolutionary algorithm based on decomposition (MOEA/D) with the penalty-based boundary intersection (PBI) function (denoted as MOEA/D-PBI) has been frequently used in many studies in the literature. One essential issue in MOEA/D-PBI is its penalty parameter value specification. However, it is not easy to specify the penalty parameter value appropriately. This is because MOEA/D-PBI shows different search behavior when the penalty parameter values are different. The PBI function with a small penalty parameter value is good for convergence. However, the PBI function with a large value of penalty parameter is needed to preserve the diversity and uniformity of solutions. Although some methods for adapting the penalty parameter value for each weight vector have been proposed, they usually lead to slow convergence. In this paper, we propose the idea of using two different values of penalty parameter simultaneously in MOEA/D-PBI. Although the idea is simple, the proposed algorithm is able to utilize both the convergence ability of a small penalty parameter value and the diversification ability of a large penalty parameter value of the PBI function. Experimental results demonstrate that the proposed algorithm works well on a wide range of test problems.

Index Terms—Decomposition-based evolutionary algorithms, MOEA/D, multi-objective optimization, penalty parameter values, penalty-based boundary intersection.

I. INTRODUCTION

OPTIMIZATION problems with multiple objective functions are frequently encountered in numerous real-world applications [1]-[3], where the objective functions are typically conflicting with one another. This implies that tackling a multi-objective optimization problem will obtain a set of trade-off solutions. The trade-off solutions are often defined using the Pareto dominance relation, which are known as Pareto optimal solutions. When all Pareto optimal solutions are mapped to the objective space, a Pareto front is obtained. Many studies have shown that evolutionary multi-objective optimization (EMO) algorithms are effective tools for tackling multi-objective optimization problems [4]. Due to EMO algorithms' population-based search nature, it is possible to obtain a set of non-dominated solutions in a single run, which

This paragraph of the first footnote will contain the date on which you submitted your paper for review, which is populated by IEEE. This work was supported by National Natural Science Foundation of China (Grant No. 61876075, 62002152), Guangdong Provincial Key Laboratory (Grant No. 2020B121201001), the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (Grant No. 2017ZT07X386), The Stable Support Plan Program of Shenzhen Natural Science Fund (Grant No. 20200925174447003), Shenzhen Science and Technology Program (Grant No. KQTD2016112514355531). (*Corresponding author: Hisao Ishibuchi*).

is then used to approximate the Pareto front. Thus, EMO algorithms are primarily designed to search for a good approximation of the Pareto front considering three criteria: convergence, spread, and uniformity.

One of the most frequently-used EMO algorithms is the multi-objective evolutionary algorithm based on decomposition (MOEA/D) [5]. MOEA/D decomposes a multi-objective optimization problem into several single-objective sub-problems by utilizing a set of weight vectors and a scalarizing function. Each sub-problem is optimized collaboratively together with its neighbors. The penalty-based boundary intersection (PBI) function is frequently used as a scalarizing function in MOEA/D for its ability to handle non-convex Pareto front [5] and its effectiveness in tackling many-objective problems [6].

One key issue in MOEA/D with the PBI function (which is referred to as MOEA/D-PBI) is the specification of its penalty parameter θ . The penalty parameter θ is used by the PBI function for balancing the diversity and convergence of solutions [5]. However, it is not easy to specify an appropriate value of θ . A small value of θ is needed for strong convergence ability [7]-[8]. For example, it has been reported in [6] and [9] that MOEA/D-PBI with a very small penalty parameter value (e.g., $\theta = 0.1$) works well for knapsack problems with many objectives (e.g., 10 objectives). Although a small value of θ is good for convergence, it leads to poor diversity of solutions on concave and linear Pareto fronts [7]. On the other hand, a large value of θ is needed for good uniformity of solutions. However, a large value of θ leads to poor convergence of solutions (e.g., for knapsack problems with many objectives) [6], [9]. Thus, a different penalty parameter value is needed for different types of problems [8].

For further enhancement on the performance of MOEA/D-PBI, several approaches have been proposed [8], [10], [11]. The general idea of this line of research is to specify a different and appropriate value of θ for each weight vector (i.e., sub-problem). Depending on the Pareto front shape around the intersection point with the weight vector, an appropriately specified θ value can maximize the convergence ability and help to find a Pareto optimal solution on the weight vector. This strategy usually leads to a large value of θ for a weight vector close to each of the axis

L. M. Pang, H. Ishibuchi, and K. Shang are with Research Institute of Trustworthy Autonomous Systems, Southern University of Science and Technology, Shenzhen 518055, China; Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China. (e-mail: panglm@sustech.edu.cn; hisao@sustech.edu.cn; kshang@foxmail.com).

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>

in the objective space when the Pareto front is convex. Another important issue is the adaptation of weight vectors for irregular Pareto fronts [12]-[13]. A number of weight vector adaptation or adjustment mechanisms have been proposed [14]-[19]. However, in general, weight vector adaptation leads to slow convergence [20].

In order to utilize both the strong convergence ability of the PBI function with a small value of penalty parameter and its strong diversification ability with a large value of penalty parameter, we propose an MOEA/D-PBI algorithm with two penalty parameter values (referred to as MOEA/D-2PBI) in this paper. The simultaneous use of two different penalty parameter values (one is a very small value and the other is a relatively large value) enables MOEA/D-2PBI to enhance the convergence as well as the uniformity of a solution set. Whereas the proposed algorithm is simple, it performs well on a set of test problems with varying characteristics. The main contributions of this paper are as follows:

- We propose an idea of simultaneously using two penalty parameter values in MOEA/D with the PBI function to achieve strong convergence (by a small penalty parameter value) and strong diversification (by a large penalty parameter value).
- We show that the proposed algorithm performs well on a wide range of test problems whereas it uses the fixed penalty parameter values and the fixed weight vectors.
- We also show that the proposed algorithm outperforms more sophisticated algorithms with adaptive penalty parameter values and adaptive weight vectors on difficult test problems and a real-world problem.
- We clearly explain why the proposed algorithm works well using the contour lines of the PBI function.

The organization of this paper is as follows. Section II provides a brief overview of multi-objective optimization and the standard MOEA/D-PBI algorithm. Section III proposes MOEA/D-2PBI. Section IV reports experimental results for a set of test problems with different characteristics. Section V presents experimental results for the multi- and many-objective knapsack test problem, and a real-world application on the conceptual marine design problem. Finally, Section VI concludes this paper.

II. BACKGROUND

A. Multi-objective Optimization

In general, optimization can be divided into maximization and minimization problems. In this paper, minimization problems are considered. We can transform maximization problems into minimization problems by multiplying -1 to the objective functions. Multi-objective optimization problems are generally formulated as follows:

$$\text{Minimize } \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_M(\mathbf{x}))^\top \text{ subject to } \mathbf{x} \in \Omega, \quad (1)$$

where $\mathbf{x} = (x_1, x_2, \dots, x_D)^\top$ is a solution (i.e., a vector with D decision variables), Ω represents the feasible region of \mathbf{x} , and $f_i(\mathbf{x})$ is the i -th objective to be minimized ($i = 1, 2, \dots, M$).

Given two solutions \mathbf{x}^a and \mathbf{x}^b , we say \mathbf{x}^a dominates \mathbf{x}^b if both of the following two conditions hold:

$$(i). f_i(\mathbf{x}^a) \leq f_i(\mathbf{x}^b), \forall i \in \{1, 2, \dots, M\},$$

$$(ii). f_j(\mathbf{x}^a) < f_j(\mathbf{x}^b), \exists j \in \{1, 2, \dots, M\}.$$

A solution $\mathbf{x}^* \in \Omega$ is a Pareto optimal solution if and only if there exists no solution \mathbf{x} in Ω that dominates \mathbf{x}^* . The Pareto set is a collection of all Pareto optimal solutions, and the Pareto front is formed when the Pareto set is projected onto the objective space.

B. MOEA/D-PBI

We briefly explain MOEA/D with the PBI function here. Let the population size of MOEA/D be denoted by N . In MOEA/D, the size of the population has the same number to the weight vectors and the number of sub-problems. MOEA/D uses a weight vectors set $\mathcal{W} = \{\mathbf{w}^1, \dots, \mathbf{w}^N\}$ to decompose a multi-objective problem with M objectives into N single-objective sub-problems. With the PBI function, each sub-problem with a weight vector \mathbf{w} is defined in (2).

$$\text{Minimize } \mathbf{f}^{\text{PBI}}(\mathbf{x}|\mathbf{w}, \mathbf{z}^*) = d_1 + \theta d_2, \quad (2)$$

where θ is a user-defined positive real number called the penalty parameter. The value of θ is usually set as 5 in many studies in the literature. In (2), d_1 and d_2 are described as follows:

$$d_1 = |(\mathbf{f}(\mathbf{x}) - \mathbf{z}^*)^\top \mathbf{w}| / \|\mathbf{w}\|, \quad (3)$$

$$d_2 = \|\mathbf{f}(\mathbf{x}) - \mathbf{z}^* - d_1(\mathbf{w}/\|\mathbf{w}\|)\|, \quad (4)$$

where $\mathbf{z}^* = (z_1^*, \dots, z_M^*)$ is the reference point. In \mathbf{z}^* , each element z_i^* is determined by calculating the minimum value of each objective value $f_i(\mathbf{x})$ over all the examined solutions.

Fig. 1 illustrates the two distances d_1 and d_2 of the PBI function in the objective space (two-objective case is considered here). As we can see from Fig. 1, d_1 is the distance from the projection of the objective vector $\mathbf{f}(\mathbf{x})$ on the weight vector \mathbf{w} to the reference point \mathbf{z}^* , and d_2 is the distance perpendicular to the weight vector from $\mathbf{f}(\mathbf{x})$. The penalty parameter θ is used to balance the convergence (based on d_1) and the diversity (based on d_2) in the multi-objective search by MOEA/D.

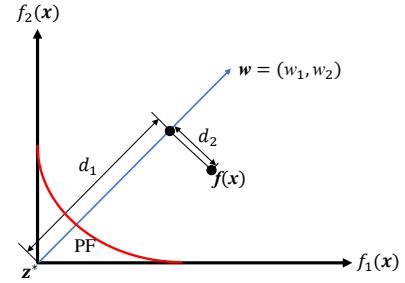


Fig. 1. Illustration of the PBI function.

Figs. 2 and 3 show the contour lines (i.e., the dotted lines) of the PBI function with different penalty parameter values. When $\theta = 0$ (Fig. 2), the contour lines are straight lines perpendicular to the weight vector. This type of contour lines helps to achieve fast convergence since the current solution has a large movable region. For example, Solution A on the weight vector in Fig. 2 can move to any point in the shaded region (e.g., Solution B), which leads to fast convergence and poor uniformity of solutions.

When $\theta = 5$ (Fig. 3), the contour lines have a sharp angle. The larger the value of θ , the sharper the angle of the contour lines. Such a sharp angle leads to good uniformity of solutions and slow

convergence since the movable region is a narrow cone along the weight vector. For example, the movable region of Solution C in Fig. 3 is the shaded region, which is much smaller than that of Solution A in Fig. 2. Solution C can move to Solution D, but cannot move to Solution E.

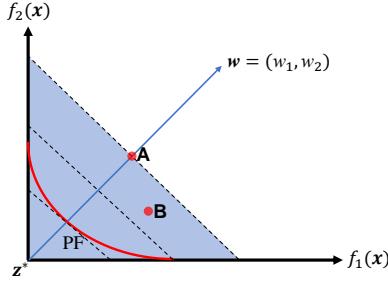


Fig. 2. Illustration of the PBI function with $\theta = 0$ and its contour lines.

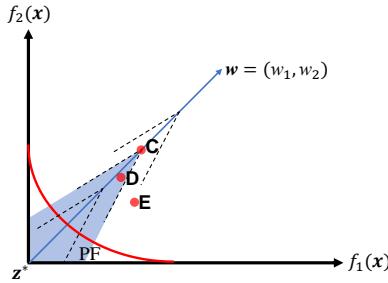


Fig. 3. Illustration of the PBI function with $\theta = 5$ and its contour lines.

In this paper, we use the Das and Dennis method [21] to construct a set of uniformly distributed weight vectors \mathbf{W} systematically. Each weight vector $\mathbf{w}^j = (w_1^j, \dots, w_M^j)^T$ must fulfill the following conditions:

$$\sum_{i=1}^M w_i^j = 1, \quad (5)$$

$$w_i^j \in \{0, 1/H, 2/H, \dots, H/H\}, \quad i = 1, 2, \dots, M, \quad (6)$$

where H (i.e., an integer parameter) determines the number of weight vectors generated (i.e., the size of the population).

The standard procedure of MOEA/D-PBI is described here. During the initialization stage, N individuals are randomly created. Each individual is randomly assigned to a sub-problem. For each weight vector \mathbf{w}^j , a neighborhood (with the T_n closest weight vectors including \mathbf{w}^j itself) is determined by calculating the Euclidean distance in the weight vector space. The size of the neighborhood T_n is a user definable parameter. Then, for each sub-problem $j \in \{1, 2, \dots, N\}$, two parents \mathbf{x}^k and \mathbf{x}^l are randomly chosen from the neighborhood. Genetic operators (i.e., crossover and mutation operators) are applied to the parent individuals to generate a child \mathbf{x}^c . The child \mathbf{x}^c is compared with each solution in its neighbourhood with the weight vector corresponding to the compared solution using the PBI function. The child \mathbf{x}^c replaces the current solution if it has a better PBI value than the current one. The process is repeated until the stopping condition is reached.

III. THE PROPOSED ALGORITHM: MOEA/D-2PBI

This section explains our proposed algorithm, i.e., the MOEA/D algorithm with two penalty parameter values

(denoted as MOEA/D-2PBI). As we have discussed in Section II.B, different penalty parameter values generate different contour lines for the PBI function. The purpose of MOEA/D-2PBI is to combine advantages of both small and large penalty parameter values. In general, the most important issue in the design of EMO algorithms is how to balance between convergence and diversity. As shown in some studies (e.g., [43]), a large θ value is needed to obtain well-distributed solutions. However, such a large θ value deteriorates the convergence ability especially for many-objective problems [6]. Our idea is to use both small and large θ values simultaneously. Two populations are evolved in a collaborative manner. One population with a small θ value helps the convergence of the other population with a large θ value. The use of multiple populations in an EMO algorithm has been shown to be both effective and practical in the literature [30], [41]. Our idea can be easily extended to more than two populations, which improves the performance of MOEA/D-2PBI on some problems and deteriorates it on other problems as discussed in the supplementary file. Our idea of using small and large θ values is implemented as MOEA/D with two populations named MOEA/D-2PBI. Each population in MOEA/D-2PBI has the population size N . The value of the penalty parameter is specified as $\theta = 0$ in one population and $\theta = 5$ in the other population. It is expected that the strong convergence ability of PBI with $\theta = 0$ will help to improve the convergence ability of PBI with $\theta = 5$.

The MOEA/D-2PBI algorithm is described as follows.

- Input:** A multi-objective optimization problem, a termination condition, a population size N , and a neighbourhood size T_n .
- Step 1:** Randomly generate $2N$ solutions as initial solutions, and randomly assign them to the two populations, i.e., Population 1 and Population 2.
- Step 2:** Apply the standard MOEA/D procedure with PBI ($\theta = 0$) to each of the N solutions in Population 1 once. Population 2 with PBI ($\theta = 5$) is updated by each generated solution in Population 1. Population 2 is handled as an archive where the entire population is used as the replacement neighbourhood for each solution generated in Population 1. The same reference point \mathbf{z}^* is used in the PBI function in the two populations. When a new solution is generated in each population, the reference point is always updated.
- Step 3:** Apply the standard MOEA/D procedure with PBI ($\theta = 5$) to each of the N solutions in Population 2 once. Population 1 with PBI ($\theta = 0$) is updated by each generated solution in Population 2. Population 1 is handled as an archive where the entire population is used as the replacement neighbourhood for each solution generated in Population 2. The shared reference point \mathbf{z}^* is always updated.
- Step 4:** Iterate Step 2 and Step 3 until the termination condition is reached. The generation update consists of Step 2 and Step 3. The output is the better population between Population 1 and

Population 2, which is selected based on their hypervolume values (calculated using the estimated ideal and nadir points).

The solution update mechanism of MOEA/D-2PBI is further explained here using a simple example, as presented in Figs. 4-6. Fig. 4 shows two populations of MOEA/D-2PBI with six weight vectors. Each population has six solutions, each of which is associated to one of the six weight vectors. The six red points, i.e., $\{a^1, a^2, a^3, a^4, a^5, a^6\}$ are the current solutions in Population 1, in which a^1, a^2, a^3, a^4, a^5 , and a^6 are associated to w^1, w^2, w^3, w^4, w^5 , and w^6 , respectively. The six yellow points, i.e., $\{b^1, b^2, b^3, b^4, b^5, b^6\}$ are the current solutions in Population 2. The solutions b^1, b^2, b^3, b^4, b^5 , and b^6 are associated to w^1, w^2, w^3, w^4, w^5 , and w^6 , respectively.

Let us assume that in Population 1, an offspring solution (i.e., the green point) is produced for the weight vector w^3 . We also assume that the neighborhood size is three (i.e., the neighborhood of w^3 is $\{w^2, w^3, w^4\}$). The offspring solution of w^3 in Population 1 is compared with its neighboring solutions using the PBI function with $\theta = 0$. The same offspring solution is also compared with all solutions in Population 2 using the PBI function with $\theta = 5$.

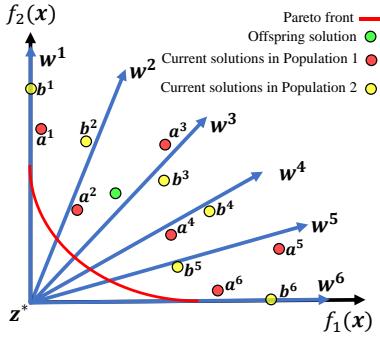


Fig. 4. Illustration of two populations in MOEA/D-2PBI. Each population is associated to the six weight vectors.

The solution update mechanism in Population 1 is illustrated in Fig. 5. The offspring solution is compared with the neighborhood solutions of w^3 (i.e., a^2, a^3 and a^4) using the PBI function with $\theta = 0$. For a better illustration, a contour line for the PBI function with $\theta = 0$ (i.e., the black dotted line) is also included for each current solution in the neighborhood of w^3 . Since the offspring solution has a better PBI function value than a^3 and a^4 , the offspring solution will replace them.

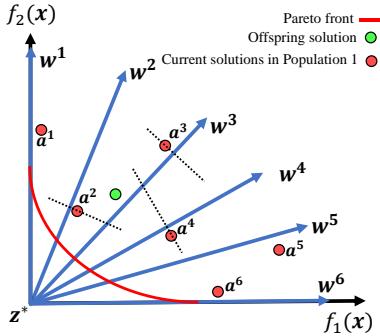


Fig. 5. Illustration of the solution update mechanism for Population 1.

Then, all the solutions in Population 2 are compared with the same offspring solution using the PBI function with $\theta = 5$. In Fig. 6, the contour line for the PBI function with $\theta = 5$ is also included for each current solution of Population 2. Since the offspring solution has a better PBI function value than the current solution b^3 of the weight vector w^3 , it will replace b^3 in Fig. 6. Since there are two different populations involved in MOEA/D-2PBI, it is worth mentioning that the total number of generations of each population in MOEA/D-2PBI is half that of population in the standard MOEA/D-PBI.

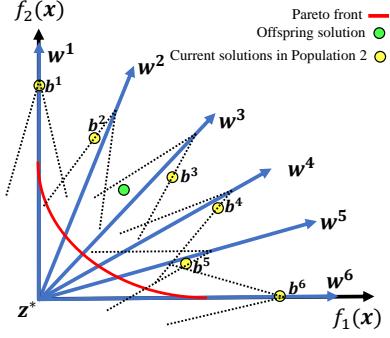


Fig. 6. Illustration of the solution update mechanism for Population 2.

Here, we further discuss the reason for comparing a newly generated solution in one population with not only the neighboring solutions in that population but also all solutions in the other population. Let us consider a test problem with a convex inverted Pareto front (e.g., Minus-DTLZ2 [27]). First, we discuss the handling of a convex Pareto front by the PBI function. In Fig. 7, the relation between the weight vectors and the obtained solutions by the PBI function with $\theta = 5$ and $\theta = 0$ is shown in (a) and (b) for a convex Pareto front of a two-objective problem. Solution 1 and Solution 2 are the best solutions for $w^1 = (1, 0)$ and $w^2 = (0, 1)$, respectively. As depicted in Fig. 7 (a), the two extreme points of the Pareto front cannot be obtained by the PBI function with $\theta = 5$. In order to obtain solutions near to the two extreme points, it is needed to use a larger value of θ (e.g., $\theta = 10$), which further weakens the convergence ability. However, they can be obtained by the PBI function with $\theta = 0$ in Fig. 7 (b).

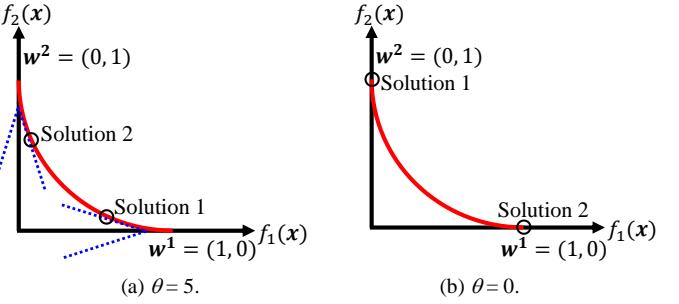


Fig. 7. Relation between the weight vector and the best solution by the PBI function. Solution 1 and Solution 2 are the best solutions for $w^1 = (1, 0)$ and $w^2 = (0, 1)$, respectively. The red curve shows the Pareto front of the normalized bi-objective Minus-DTLZ problem.

It is noteworthy that, in Fig. 7 (b), Solution 1 for $w^1 = (1, 0)$ is obtained at $(0,1)$ and Solution 2 for $w^2 = (0, 1)$ is obtained at $(1, 0)$. That is, the best solution for each weight vector is not obtained around the corresponding weight vector. This is the reason for comparing a newly generated solution in one

population with all solutions in the other population in MOEA/D-2PBI (e.g., all solutions in Population 2 with $\theta = 5$ are compared with a newly generated solution in Population 1 with $\theta = 0$). For example, let us assume that Solution 2 in Fig. 7 (b) on the $f_1(x)$ -axis is generated for $\mathbf{w}^1 = (1, 0)$ in the population with $\theta = 5$ in Fig. 7(a). This solution cannot survive in Fig. 7(a) with $\theta = 5$. However, it can survive in Fig. 7(b) with $\theta = 0$ as the solution for $\mathbf{w}^2 = (0, 1)$.

Since two populations are obtained in MOEA/D-2PBI, the population with a better hypervolume value will be selected as the output of MOEA/D-2PBI. The hypervolume value is calculated based on the estimated ideal and nadir points. The estimated ideal point is the same as the reference point \mathbf{z}^* in the final population. The estimated nadir point will be obtained from the non-dominated solutions among all the solutions in the two final populations. In this paper, the exact hypervolume calculation (i.e., the fast WFG algorithm [22]) is used since computational experiments are performed on test problems with three to eight objectives. For problems with ten objectives or more (i.e., $M \geq 10$), we can use an approximate hypervolume calculation method (e.g., [23]). Alternatively, the IGD⁺ indicator [24] can be used since the hypervolume indicator and the IGD⁺ indicator have similar characteristic feature [25]. In IGD⁺, all the non-dominated solutions from the two final populations will serve as reference points for distance calculation.

IV. EXPERIMENTS

A. Comparison Between the MOEA/D-PBI and MOEA/D-2PBI Algorithms

As an initial step to evaluate the performance of MOEA/D-2PBI, it is first compared with the standard MOEA/D-PBI. The frequently-used DTLZ1-4 test problems [26] and their minus version (i.e., the Minus-DTLZ1-4 test problems) [27] are used in this section. The number of objectives is specified as $M \in \{3, 4, 6, 8\}$. A total of 32 test instances are used. As in [26], the number of decision variables (i.e., D) is specified as $M + k - 1$ for DTLZ and Minus-DTLZ, where k equals 5 for DTLZ1 and Minus-DTLZ1, and k is set as 10 for DTLZ2-4 and Minus-DTLZ2-4. Table I shows the population size used in our experiments for each setting of the number of objectives.

TABLE I.
POPULATION SIZE FOR EACH SETTING OF THE OBJECTIVE NUMBER.

M	Population size (N)
3	91
4	120
6	126
8	156

All experiments are conducted on the PlatEMO platform [28], with the following experimental settings:

Neighborhood size T_n : 10% of the population size,
Crossover operator: Simulated binary crossover (SBX),
Mutation operator: Polynomial mutation,
Crossover probability: 1,
Mutation probability: $1/D$,
Distribution indexes for crossover and mutation: 20,

Stopping conditions: 300 generations for three- and four-objective test problems (i.e., 27,300 and 36,000 solution evaluations for $M = 3$ and 4, respectively), 400 generations for six- and eight-objective test problems (i.e., 50,400 and 62,400 solution evaluations for $M = 6$ and 8, respectively).

In order to provide a fair comparison, it is important to note that the total number of generations of each population in MOEA/D-2PBI is half that of population in the standard MOEA/D-PBI. For example, when the termination condition is set as 300 generations for three- and four-objective test problems, each population in MOEA/D-2PBI is actually evolved for 150 generations only, whereas the population in the standard MOEA/D-PBI is evolved for 300 generations.

Each MOEA/D algorithm is performed on each test problem 31 times independently. The performance evaluation of each algorithm is based on the hypervolume indicator. For the calculation of the hypervolume value, the objective space is normalized using the true ideal and nadir points of each test problem first. In this manner, the ideal point is $(0, 0, \dots, 0)$ and the nadir point is $(1, 1, \dots, 1)$. Then, a reference point is set as $\mathbf{r} = (1.1, 1.1, \dots, 1.1)$ in the normalized objective space. An algorithm with a higher hypervolume value has better performance.

Table II presents the average hypervolume values by each standard MOEA/D algorithm (with different values of θ) and each population of MOEA/D-2PBI on a set of DTLZ and Minus-DTLZ test instances with three, four, six and eight objectives. Since MOEA/D-PBI with $\theta = 5$ is frequently used in many studies for performance comparison (e.g., see [15], [29]-[30]), the statistical significance of differences in the obtained experimental results between MOEA/D-PBI (with $\theta = 5$) and each of the other algorithms are examined. In this paper, the Wilcoxon's rank sum test at the 5% level of significant is used. We use the symbols “+”, “-”, and “=” to show whether the compared algorithm has better, worse, or equivalent performance to the original MOEA/D-PBI (with $\theta = 5$) statistically. The bold typeface and the gray shading are used to highlight the best result of each test instance.

In Table II, we can observe that in the standard MOEA/D-PBI algorithm, a different value of the penalty parameter (i.e., θ) is needed for a different test problem (thus, two different penalty parameter values are used in our proposed algorithm). In MOEA/D-2PBI, better results are obtained from Population 2 with $\theta = 5$ on many test problems. However, on some test problems, better results are obtained from Population 1 with $\theta = 0$ (thus, a better population is selected as the final output of the proposed algorithm). It can also be observed that on many test problems, the results by the proposed MOEA/D-2PBI algorithm are better than or similar to the results of MOEA/D-PBI with the best setting of the penalty parameter value.

Fig. 8 shows the solution sets obtained for the three-objective DTLZ2 problem by the standard MOEA/D-PBI with $\theta = 5$ and the proposed MOEA/D-2PBI. A single run is selected based on the median hypervolume value of 31 runs. Even though the standard MOEA/D-PBI with $\theta = 5$ has statistically better performance than MOEA/D-2PBI on DTLZ2, their average hypervolume values are very similar. From Fig. 8(a) and (b), it can be also clearly observed that their obtained solution distributions are very similar. However, when the setting of the

penalty parameter value is not appropriate, clearly worse results are obtained by MOEA/D-PBI than MOEA/D-2PBI.

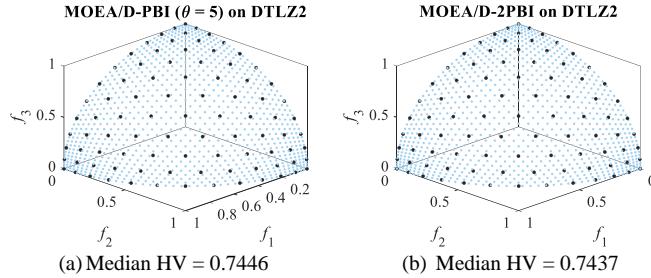


Fig. 8. The solution sets obtained for the three-objective DTLZ2 problem by (a) MOEA/D-PBI with $\theta = 5$, and (b) MOEA/D-2PBI (Population 2), respectively. The true Pareto front is represented by the blue surface, and the solutions are shown by the black points.

Fig. 9 shows the solution sets obtained for the three-objective Minus-DTLZ2 problem by the standard MOEA/D-PBI with $\theta = 5$ and the proposed MOEA/D-2PBI. For Minus-DTLZ2, the output of the proposed MOEA/D-2PBI is Population 1 because it has a better hypervolume value than Population 2. As shown in Fig. 9 (a), the standard MOEA/D-PBI with $\theta = 5$ cannot obtain a well-distributed solutions set over the entire Pareto front. As for MOEA/D-2PBI, the obtained solutions can evenly cover the entire Pareto front (as shown in Fig. 9 (b)).

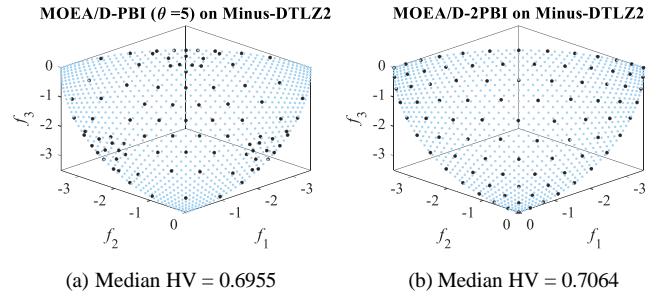


Fig. 9. The solution sets obtained for the three-objective Minus-DTLZ2 by (a) MOEA/D-PBI with $\theta = 5$, and (b) MOEA/D-2PBI (Population 1), respectively. The true Pareto front is represented by the blue surface, and the solutions are shown by the black points.

The experimental results in Table II also indicate that the convergence ability of MOEA/D-2PBI is significantly enhanced by the use of the two penalty parameter values simultaneously. As an example, the improvement of the convergence ability can be demonstrated by the performance of MOEA/D-2PBI (Population 2) on the DTLZ3 test problem. The DTLZ3 test problem is designed to test the convergence ability of an EMO algorithm to the Pareto optimal front. The search space of DTLZ3 contains a huge number of local Pareto optimal fronts, which hinders an EMO algorithm from approaching the global Pareto optimal front [26]. Comparing to the standard MOEA/D-PBI, MOEA/D-2PBI has significantly better performance on the DTLZ3 test problem with three, four, six, and eight objectives.

TABLE II.
THE AVERAGE HYPERVOLUME VALUE OF EACH MOEA/D ALGORITHM ON EACH TEST PROBLEM WITH THREE TO EIGHT OBJECTIVES.

Test Instance	M	Standard MOEA/D-PBI					MOEA/D-2PBI	
		$\theta = 0$	$\theta = 1$	$\theta = 2$	$\theta = 5$	$\theta = 10$	Population 1	Population 2
DTLZ1	3	4.7151E-1 -	1.0790E+0 -	1.1148E+0 =	1.1141E+0	1.1132E+0 =	4.6554E-1 -	1.1169E+0 +
	4	6.9107E-1 -	1.2848E+0 -	1.3720E+0 =	1.3728E+0	1.3715E+0 =	6.2715E-1 -	1.3741E+0 +
	6	1.1373E+0 -	1.6501E+0 -	1.7356E+0 -	1.7497E+0	1.7497E+0 =	1.0443E+0 -	1.7497E+0 =
	8	1.4242E+0 -	2.0668E+0 -	1.8787E+0 -	2.1254E+0	2.1339E+0 +	1.4080E+0 -	2.1135E+0 -
DTLZ2	3	3.3100E-1 -	6.9290E-1 -	7.4469E-1 +	7.4459E-1	7.4447E-1 -	3.3100E-1 -	7.4372E-1 -
	4	4.6410E-1 -	7.2653E-1 -	1.0320E+0 +	1.0316E+0	1.0313E+0 -	4.6410E-1 -	1.0295E+0 -
	6	6.7140E-1 -	6.5737E-1 -	1.5135E+0 +	1.5133E+0	1.5132E+0 -	6.7156E-1 -	1.5117E+0 -
	8	6.7931E-1 -	6.6879E-1 -	1.9810E+0 +	1.9798E+0	1.9792E+0 -	6.7923E-1 -	1.9781E+0 -
DTLZ3	3	2.4715E-1 =	1.7426E-1 =	3.8367E-1 +	2.3000E-1	7.7518E-2 -	1.5339E-1 =	4.4850E-1 +
	4	4.1992E-1 =	3.6783E-1 =	5.4973E-1 =	4.2029E-1	2.9275E-2 -	3.1276E-1 =	7.8548E-1 +
	6	6.3466E-1 +	3.2483E-1 =	5.1720E-1 =	5.2659E-1	4.7388E-1 =	6.3673E-1 +	1.4626E+0 +
	8	6.6528E-1 =	2.8787E-1 -	5.7426E-1 -	1.1855E+0	9.0301E-1 =	6.5017E-1 =	1.9015E+0 +
DTLZ4	3	1.8713E-1 -	4.2941E-1 -	4.9106E-1 =	4.6286E-1	5.7331E-1 +	2.7552E-1 -	5.7853E-1 +
	4	2.2500E-1 -	5.2912E-1 -	6.9226E-1 =	7.5243E-1	7.6992E-1 =	3.6152E-1 -	7.8068E-1 =
	6	3.8190E-1 -	5.8848E-1 -	1.2369E+0 =	1.2341E+0	1.2813E+0 =	5.7718E-1 -	1.2257E+0 =
	8	4.7348E-1 -	6.4324E-1 -	1.6279E+0 -	1.6990E+0	1.7101E+0 =	7.4308E-1 -	1.7647E+0 =
Minus-DTLZ1	3	3.7844E-2 -	2.6143E-1 +	2.6097E-1 +	2.5550E-1	2.4427E-1 -	4.1981E-2 -	2.4983E-1 -
	4	5.9078E-3 -	7.4119E-2 +	7.0758E-2 +	6.8419E-2	6.7352E-2 -	5.2130E-3 -	6.6495E-2 -
	6	1.0504E-4 -	1.6840E-3 +	1.2941E-3 =	1.4151E-3	1.7110E-3 +	9.3530E-5 -	1.9345E-3 +
	8	1.7045E-6 -	3.8094E-5 +	6.3213E-6 =	7.0605E-6	7.4230E-6 =	1.0443E-6 -	4.7327E-5 +
Minus-DTLZ2	3	7.0649E-1 +	5.5322E-1 -	6.8523E-1 -	6.9510E-1	6.7595E-1 -	7.0640E-1 +	6.9377E-1 -
	4	3.6563E-1 +	2.9279E-1 -	3.9609E-1 +	3.5122E-1	3.2180E-1 -	3.6562E-1 +	3.5043E-1 =
	6	1.6121E-2 -	6.7512E-2 +	6.2400E-2 +	5.5207E-2	4.9061E-2 -	1.6323E-2 -	5.4959E-2 =
	8	1.5433E-3 -	1.1177E-2 +	5.3340E-3 -	6.0249E-3	5.3084E-3 -	1.5016E-3 -	6.2322E-3 +
Minus-DTLZ3	3	7.0134E-1 +	5.3947E-1 -	6.7025E-1 +	6.6454E-1	6.2983E-1 -	6.9499E-1 +	6.7947E-1 +
	4	3.6233E-1 +	2.8255E-1 -	3.8400E-1 +	3.3804E-1	2.9498E-1 -	3.5158E-1 +	3.2840E-1 -
	6	1.6587E-2 -	6.6009E-2 +	6.1944E-2 +	5.4997E-2	4.7741E-2 -	1.8561E-2 -	5.1572E-2 -
	8	1.4120E-3 -	1.0854E-2 +	5.1923E-3 -	5.9607E-3	5.1100E-3 -	1.1491E-3 -	5.8831E-3 =
Minus-DTLZ4	3	6.4739E-1 +	4.4428E-1 -	5.5225E-1 =	4.5764E-1	5.4394E-1 =	6.8672E-1 +	6.7367E-1 +
	4	3.3684E-1 +	1.3596E-1 -	2.6737E-1 =	2.6449E-1	2.5670E-1 -	2.9844E-1 +	2.8443E-1 =
	6	1.5409E-2 -	2.2011E-2 -	2.4377E-2 =	2.3047E-2	2.9814E-2 =	1.2848E-2 -	3.7243E-2 +
	8	1.3851E-3 -	9.1026E-4 =	7.6514E-4 -	1.2436E-3	9.5115E-4 -	1.0906E-3 =	4.2891E-3 +
+/- /=		7/21/4	8/20/4	12/8/12		3/18/11	7/21/4	14/10/8

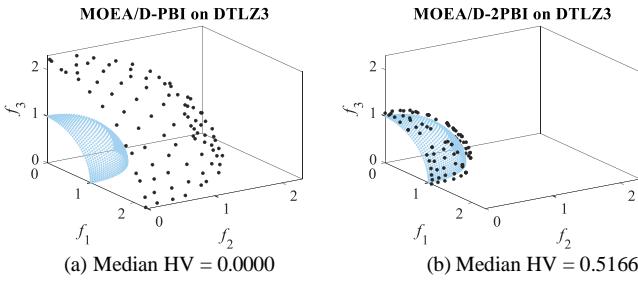


Fig. 10. The solution sets obtained for the three-objective DTLZ3 by (a) MOEA/D-PBI with $\theta = 5$, and (b) MOEA/D-2PBI (Population 2), respectively. The true Pareto front is represented by the blue surface, and the solutions are shown by the black points.

Fig. 10 shows the solution sets obtained for the three-objective DTLZ3 problem by MOEA/D-2PBI and MOEA/D-PBI (with $\theta = 5$), respectively. We can see that the solution set obtained by MOEA/D-2PBI on the three-objective DTLZ3 problem is much closer to the true Pareto front. It is worth noting that the total number of generations of each population in MOEA/D-2PBI is half of the total number of generations in the standard MOEA/D-PBI. That is, the solution set obtained by MOEA/D-2PBI is evolved for 150 generations, whereas the solution set obtained by MOEA/D-PBI is evolved for 300 generations. The obtained solution sets clearly demonstrate that MOEA/D-2PBI with two penalty parameter values can achieve faster convergence than the standard MOEA/D-PBI with a single penalty parameter value. In summary, the experimental results suggest the usefulness of using two penalty parameter values simultaneously in improving the search ability of MOEA/D with the PBI function.

B. Comparison Between MOEA/D-2PBI with Other EMO Algorithms

The proposed MOEA/D-2PBI algorithm is compared with other EMO algorithms in this section. NSGA-III [31], RVEA [29], MOEA/D-AWA [14], AdaW [17], MOEA/D-APS [8], Two_Arch2 [38], θ -DEA [39], and R2HCA-EMOA [40] are used for comparison. These compared algorithms are selected based on the following considerations. NSGA-III is one of the most widely used decomposition-based EMO algorithms in the EMO community. RVEA, MOEA/D-AWA, and AdaW are decomposition-based EMO algorithms that adopt certain strategies to adaptively adjust the weight vectors (or reference vectors) along the evolutionary process. MOEA/D-APS uses an adaptive penalty scheme (APS) to linearly adjust the value of θ at different evolution stages. Two_Arch2 and θ -DEA are also used in our study for performance comparison since they have been shown to be effective in solving many-objective problems. R2HCA-EMOA is a recently proposed hypervolume-based algorithm. We use R2HCA-EMOA in our experiments since its performance has been shown to be competitive to other hypervolume-based algorithms [40]. Experimental settings are the same as those used in Section IV.A on the PlatEMO platform. On Pages 2 and 3 in the supplementary file, the nine algorithms are also compared under the following two settings of the

population size: (i) N in the proposed algorithm and $2N$ in all the other algorithms, and (ii) $2N$ in all algorithms.

Table III shows the computational results of each EMO algorithm on the DTLZ1-4 and Minus-DTLZ1-4 problems. 31 runs are independently performed for each test problem. The symbols “+”, “-”, and “=” are used to show whether the compared algorithm has better, worse, or equivalent performance to the proposed MOEA/D-2PBI statistically. The bold typeface is used to highlight the best result of each test instance. On average, MOEA/D-2PBI performs well on most of the examined test instances. On 22 out of 32 test instances, MOEA/D-2PBI ranks in the top five among the nine algorithms. The experimental results indicate that, although MOEA/D-2PBI is not always the best, it does not show very poor performance on any of the test problems. MOEA/D-2PBI is particularly good at handling difficult-to-converge problems, as evidenced by its results on the DTLZ1 and DTLZ3 problems.

The convergence curves of different algorithms on the six-objective and eight-objective DTLZ3 problems are presented in Figs. 11 and 12, respectively. These figures demonstrate that MOEA/D-2PBI converges towards the Pareto fronts faster than the other compared algorithms. In addition, we can also see that MOEA/D-2PBI obtains good performance on many six- and eight-objective test instances. The experimental results clearly indicate that the use of a small penalty parameter value and a large penalty parameter value together can effectively enhance the searchability of MOEA/D in many-objective optimization.

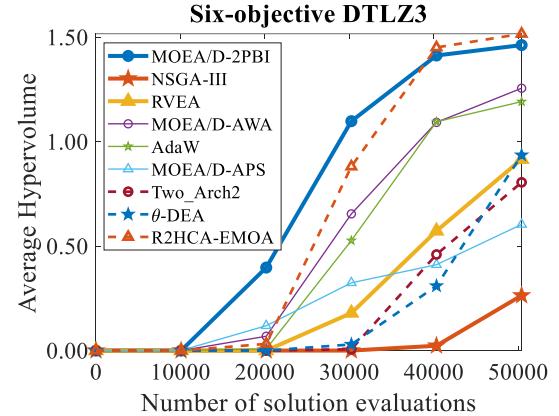


Fig. 11. The convergence curve for six-objective DTLZ3.

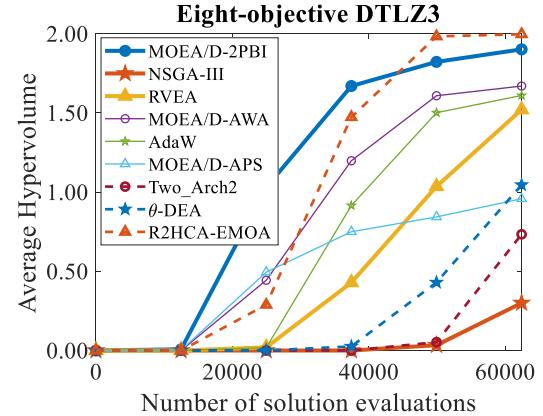


Fig. 12. The convergence curve for eight-objective DTLZ3.

TABLE III.
THE AVERAGE HYPERVOLUME VALUE OF EACH ALGORITHM ON THE DTLZ AND MINUS-DTLZ TEST PROBLEMS.

Test Instance	M	MOEA/D-2PBI	NSGA-III	RVEA	MOEA/D-AWA	AdaW	MOEA/D-APS	Two_Arch2	θ -DEA	R2HCA-EMOA
DTLZ1	3	1.117E+0	1.115E+0 -	1.115E+0 -	1.109E+0 -	1.116E+0 -	1.112E+0 -	1.067E+0 -	1.117E+0 =	1.118E+0 +
	4	1.374E+0	1.372E+0 -	1.374E+0 =	1.357E+0 -	1.371E+0 -	1.372E+0 -	1.362E+0 -	1.374E+0 =	1.375E+0 +
	6	1.750E+0	1.713E+0 -	1.750E+0 +	1.737E+0 -	1.745E+0 -	1.749E+0 -	1.736E+0 -	1.740E+0 -	1.751E+0 +
	8	2.114E+0	2.079E+0 -	2.138E+0 +	2.112E+0 =	2.004E+0 -	2.127E+0 +	2.107E+0 -	2.134E+0 +	2.138E+0 +
DTLZ2	3	7.437E-1	7.446E-1 +	7.444E-1 +	7.464E-1 +	7.441E-1 =	7.446E-1 +	7.417E-1 -	7.446E-1 +	7.527E-1 +
	4	1.030E+0	1.031E+0 +	1.032E+0 +	1.034E+0 +	1.027E+0 -	1.032E+0 +	9.961E-1 -	1.032E+0 +	1.046E+0 +
	6	1.512E+0	1.512E+0 +	1.513E+0 +	1.506E+0 -	1.485E+0 -	1.513E+0 +	1.343E+0 -	1.513E+0 +	1.534E+0 +
	8	1.978E+0	1.963E+0 -	1.980E+0 +	1.941E+0 -	1.947E+0 -	1.980E+0 +	1.613E+0 -	1.980E+0 +	2.006E+0 +
DTLZ3	3	4.485E-1	3.143E-1 =	7.318E-2 -	3.628E-1 =	3.046E-1 =	3.218E-1 =	1.020E-1 -	4.042E-1 =	5.501E-1 +
	4	7.855E-1	2.413E-1 -	1.584E-1 -	8.498E-1 =	6.258E-1 -	3.954E-1 -	2.151E-1 -	5.934E-1 =	9.303E-1 +
	6	1.463E+0	2.638E-1 -	9.161E-1 -	1.256E+0 -	1.192E+0 -	6.040E-1 -	8.055E-1 -	9.357E-1 -	1.517E+0 +
	8	1.901E+0	3.001E-1 -	1.521E+0 -	1.670E+0 -	1.610E+0 -	9.583E-1 -	7.335E-1 -	1.046E+0 -	1.997E+0 +
DTLZ4	3	5.785E-1	6.670E-1 +	7.349E-1 +	7.184E-1 +	7.360E-1 +	4.538E-1 -	7.419E-1 +	6.842E-1 +	5.402E-1 =
	4	7.807E-1	9.581E-1 +	1.032E+0 +	9.739E-1 +	9.929E-1 +	6.628E-1 =	9.909E-1 +	9.995E-1 +	9.881E-1 +
	6	1.226E+0	1.482E+0 +	1.510E+0 +	1.347E+0 +	1.485E+0 +	1.239E+0 =	1.280E+0 =	1.513E+0 +	1.482E+0 +
	8	1.765E+0	1.950E+0 +	1.979E+0 +	1.864E+0 +	1.959E+0 +	1.659E+0 =	1.588E+0 -	1.981E+0 +	1.980E+0 +
Minus-DTLZ1	3	2.498E-1	2.719E-1 +	2.440E-1 -	2.721E-1 +	2.919E-1 +	2.564E-1 +	2.913E-1 +	2.509E-1 =	2.950E-1 +
	4	6.650E-2	5.255E-2 -	3.085E-2 -	5.810E-2 -	8.390E-2 +	6.931E-2 +	8.324E-2 +	5.043E-2 -	8.638E-2 +
	6	1.934E-3	1.350E-3 -	2.884E-4 -	1.706E-3 -	2.737E-3 +	1.270E-3 -	3.002E-3 +	6.097E-4 -	3.106E-3 +
	8	4.733E-5	5.212E-5 -	3.519E-6 -	1.490E-5 -	4.963E-5 +	5.558E-6 -	5.870E-5 +	4.760E-5 =	6.712E-5 +
Minus-DTLZ2	3	7.064E-1	6.924E-1 -	6.705E-1 -	6.950E-1 -	7.136E-1 +	6.885E-1 -	7.117E-1 +	6.879E-1 -	7.097E-1 +
	4	3.656E-1	3.390E-1 -	2.877E-1 -	3.020E-1 -	4.023E-1 +	3.307E-1 -	4.056E-1 +	3.574E-1 -	4.211E-1 +
	6	5.496E-2	2.437E-2 -	2.725E-3 -	3.612E-2 -	5.416E-2 =	5.106E-2 -	6.985E-2 +	1.617E-2 -	9.026E-2 +
	8	6.232E-3	5.211E-3 -	2.858E-3 -	2.970E-3 -	3.758E-3 -	5.528E-3 -	7.777E-3 +	4.878E-3 -	1.336E-2 +
Minus-DTLZ3	3	6.950E-1	6.730E-1 -	6.468E-1 -	6.844E-1 -	6.964E-1 =	6.601E-1 -	7.096E-1 +	6.727E-1 -	7.081E-1 +
	4	3.516E-1	3.070E-1 -	2.662E-1 -	3.031E-1 -	3.780E-1 +	3.134E-1 -	3.922E-1 +	3.415E-1 -	4.198E-1 +
	6	5.157E-2	2.140E-2 -	3.146E-3 -	3.600E-2 -	5.096E-2 =	5.101E-2 =	6.213E-2 +	1.903E-2 -	8.979E-2 +
	8	5.883E-3	3.678E-3 -	2.812E-3 -	2.699E-3 -	3.431E-3 -	5.477E-3 -	5.924E-3 =	4.835E-3 -	1.329E-2 +
Minus-DTLZ4	3	6.867E-1	6.926E-1 +	6.508E-1 -	6.975E-1 +	7.128E-1 +	5.326E-1 -	7.115E-1 +	6.940E-1 +	7.176E-1 +
	4	2.984E-1	3.292E-1 +	2.704E-1 -	2.987E-1 +	3.985E-1 +	1.880E-1 -	4.041E-1 +	3.515E-1 +	4.268E-1 +
	6	3.697E-2	1.376E-2 -	3.670E-3 -	2.338E-2 =	5.033E-2 =	1.827E-2 -	6.934E-2 +	1.135E-2 -	8.980E-2 +
	8	4.279E-3	4.250E-3 =	1.085E-3 -	2.079E-3 -	2.714E-3 =	1.294E-3 -	7.604E-3 +	4.793E-3 =	1.268E-2 +
+/-=		11/19/2	10/21/1	9/19/4	13/12/7	7/20/5	17/13/2	11/14/7	31/0/1	

In Table III, R2HCA-EMOA shows the best performance among all the compared algorithms on many test instances. Since R2HCA-EMOA is a hypervolume-based algorithm that has been designed to maximize the hypervolume of the current population, it is reasonable to expect that R2HCA-EMOA obtains the best performance in terms of hypervolume. It is also observed that MOEA/D-2PBI has lower performance than AdaW and Two_Arch2 on many Minus-DTLZ test instances. Since Minus-DTLZ test instances have inverted triangular Pareto fronts, decomposition-based algorithms with the predefined weight vectors usually perform poorly on these test instances. This is the reason why MOEA/D-2PBI with the predefined weight vectors is outperformed by AdaW with a weight vector adaptation mechanism and Two_Arch2 with a search mechanism based on no weight vectors. However, as shown later, MOEA/D-2PBI outperforms these two algorithms on difficult-to-converge test problems and a real-world problem.

Next, we further examine the performance of MOEA/D-2PBI using a newly proposed many-objective test problem, i.e., HTNY19 [32]-[33]. The formulation of HTNY19 is described as follows:

$$\text{Minimize } f_i(\mathbf{x}) = \max \left\{ 0, x_i - \beta \sum_{j=1}^M x_j \right\}, \quad i = 1, 2, \dots, M, \quad (7)$$

$$\text{subject to } 0 \leq x_i \leq 100, \quad i=1, 2, \dots, M, \quad (8)$$

where

$$f_i(\mathbf{x}) = 10000, \quad i=1, 2, \dots, M, \quad \text{if } \sum_{i=1}^M f_i(\mathbf{x}) < 1. \quad (9)$$

According to [32], the value of β is set as 0.1 for HTNY19. The number of decision variables D equals the number of objectives M . Whereas the HTNY19 problem is defined by a simple formulation, it is a difficult many-objective test problem for most EMO algorithms, as demonstrated in [32]. Especially, a strong convergence ability is necessary in order to obtain good solutions on the Pareto front.

Table IV presents the average hypervolume value of each algorithm on the HTNY19 problem with a stopping condition of 5000 generations (2500 generations for each population of MOEA/D-2PBI: 455,000, 600,000, 630,000, and 780,000 solution evaluations for three-objective, four-objective, six-objective and eight-objective problems, respectively). Each algorithm is performed 31 times independently on each test instance. We use the symbols “+”, “-”, and “=” to show whether the compared algorithm has better, worse, or equivalent performance to the MOEA/D-2PBI statistically. It is observed in Table IV that the search performance of NSGA-III, MOEA/D-AWA and AdaW is deteriorated by increasing the objective number from six to eight. Especially, the performance of MOEA/D-AWA and AdaW is severely degraded (e.g., the average hypervolume values obtained by MOEA/D-AWA and AdaW are zero and 0.2055 for the eight-objective problem, respectively). MOEA/D-2PBI, RVEA, MOEA/D-APS, θ -DEA, and R2HCA-EMOA obtain similar performance on the

HTNY19 problem with three to eight objectives. That is, almost the same average hypervolume values are obtained from these

algorithms (e.g., see the results on the eight-objective HTNY19 problem in Table IV).

TABLE IV.

THE AVERAGE HYPERVOLUME VALUE OF EACH ALGORITHM ON THE HTNY19 PROBLEM. THE NUMBER OF DECISION VARIABLES IS THE SAME AS THE NUMBER OF OBJECTIVES.

Test Instance	M	MOEA/D-2PBI ($\theta = 5$ and $\theta = 0$)	MOEA/D-PBI ($\theta = 0$)	MOEA/D-PBI ($\theta = 5$)	NSGA-III	RVEA	MOEA/D-AWA	AdaW	MOEA/D-APS	Two_Arch2	θ -DEA	R2HCA-EMOA
HTNY19	3	1.1106	0.3310 -	1.1138 +	1.1183 +	1.1164 +	1.0490 -	1.1134 +	1.1190 +	1.0917 -	1.1194 +	1.1197 +
	4	1.3728	0.4641 -	1.3752 +	1.3759 +	1.3745 +	1.2650 -	1.3724 =	1.3763 +	1.3396 -	1.3763 +	1.3768 +
	6	1.7505	0.7457 -	1.7507 +	1.7510 +	1.7505 +	1.6397 -	1.7369 -	1.7510 +	1.6223 -	1.7510 +	1.7490 =
	8	2.1384	1.5725 -	1.9997 -	1.5862 =	2.1375 -	0.0000 -	0.2055 -	2.1383 =	1.9790 -	2.1385 +	2.1337 -
$+/-=$		0/4/0	3/1/0	3/0/1	3/1/0	0/4/0	1/2/1	3/0/1	0/4/0	4/0/0	2/1/1	

TABLE V.

THE AVERAGE HYPERVOLUME VALUE OF EACH ALGORITHM ON THE EIGHT-OBJECTIVE HTNY19 PROBLEM WITH DIFFERENT NUMBER OF DECISION VARIABLES.

Test Instance	D	MOEA/D-2PBI ($\theta = 5$ and $\theta = 0$)	MOEA/D-PBI ($\theta = 0$)	MOEA/D-PBI ($\theta = 5$)	NSGA-III	RVEA	MOEA/D-AWA	AdaW	MOEA/D-APS	Two_Arch2	θ -DEA	R2HCA-EMOA
HTNY19 ($M=8$)	8	2.1384	1.5725 -	1.9997 -	1.5862 =	2.1375 -	0.0000 -	0.2055 -	2.1383 =	1.9790 -	2.1385 +	2.1337 -
	40	2.1372	1.8906 -	0.0000 -	0.0120 -	2.1353 -	0.0000 -	0.0000 -	2.1359 -	2.0499 -	1.8862 =	2.1375 +
	80	2.1364	1.9597 -	0.0000 -	0.0000 -	2.1231 -	0.0000 -	0.0000 -	2.1332 -	2.0547 -	1.1812 -	2.1383 +
	120	2.1356	1.9580 -	0.0000 -	0.0000 -	2.0199 -	0.0000 -	0.0000 -	2.1327 -	2.0657 -	0.0256 -	2.0693 -
$+/-=$		0/4/0	0/4/0	0/3/1	0/4/0	0/4/0	0/4/0	0/4/0	0/3/1	0/4/0	1/2/1	2/2/0

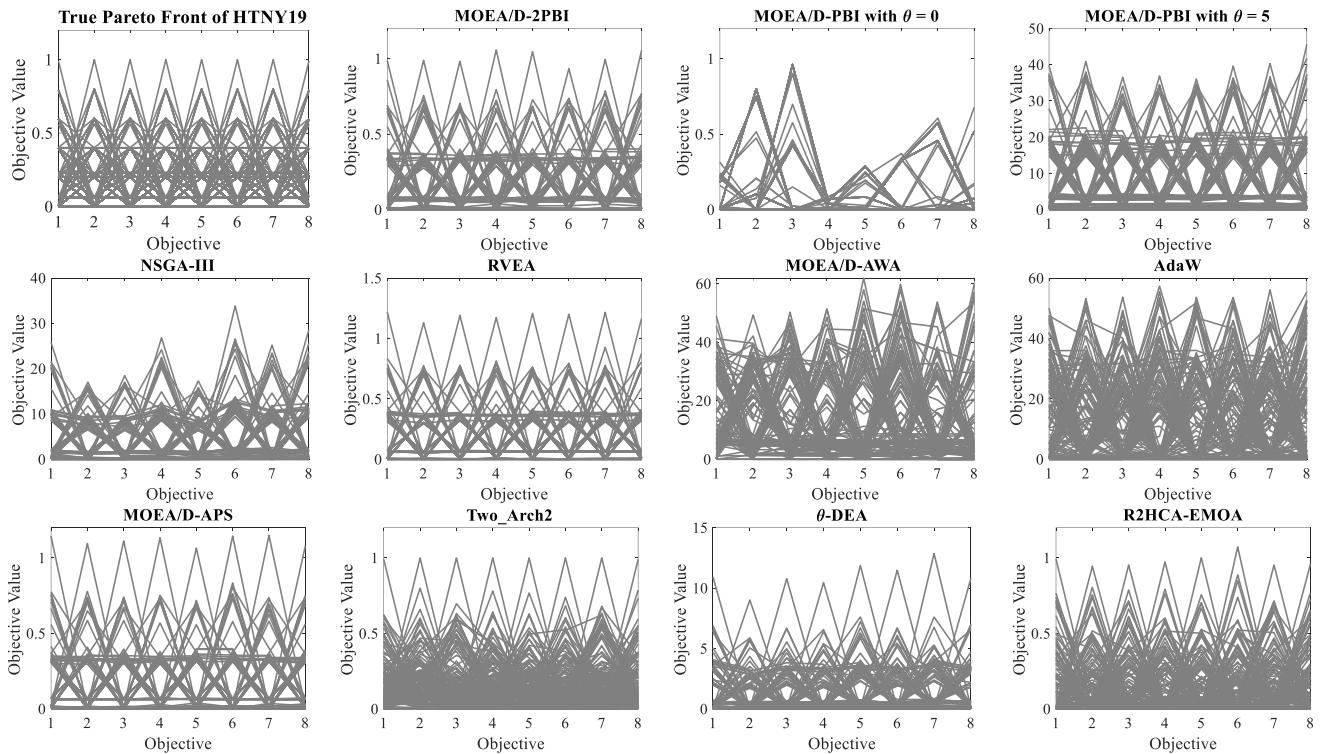


Fig. 13. The true Pareto front of the eight-objective HTNY19 and the obtained solution sets for the eight-objective HTNY19 problem with 120 decision variables by different algorithms.

We can increase the number of decision variables for HTNY19 problem from M to Mp by splitting each original decision variables in (7)-(8) into p variables [34], as follows:

$$x_i = y_{i1} + y_{i2} + \dots + y_{ip}, \quad i = 1, 2, \dots, M, \quad (10)$$

$$0 \leq y_{ih} \leq 100/p, \quad i = 1, 2, \dots, M, \text{ where } h = 1, 2, \dots, p. \quad (11)$$

For the eight-objective HTNY19 problem, we increase the decision variables number from 8 to 40, 80, and 120 (i.e., $p = 5, 10$, and 15). The results are presented in Table V. The difficulty of the HTNY19 problem is increased by increasing the number of decision variables, as evident from the performance degradation of the other EMO algorithms in Table

V. MOEA/D-2PBI shows robust performance even for the case of 120 decision variables. Fig. 13 presents the solution sets obtained for the eight-objective HTNY19 problem with 120 decision variables by different algorithms. It is clearly observed that a set of well converged and diversified solutions is obtained by MOEA/D-2PBI. Even though MOEA/D-PBI with $\theta = 0$ can converge to the Pareto front, it cannot obtain a set of well-diversified solutions. RVEA, MOEA/D-APS, Two_Arch2, and R2HCA-EMOA show slightly inferior performance to MOEA/D-2PBI on the eight-objective HTNY19 problem with 120 decision variables. From the vertical axis of each figure in Fig. 13, it can be seen that NSGA-III, MOEA/D-AWA, AdaW and θ -DEA need much stronger convergence ability for the eight-objective HTNY19 problem with 120 decision variables.

As we have explained in Section II, MOEA/D-PBI with $\theta = 0$ has strong convergence ability, and MOEA/D-PBI with $\theta = 5$ has strong diversification ability. Our idea is to utilize the strength of each specification simultaneously in a single algorithm. The strong convergence ability of MOEA/D-PBI with $\theta = 0$ is demonstrated in Fig. 13 and Table V on the eight-objective HTNY19 problem where good results are not obtained by MOEA/D-PBI with $\theta = 5$ when the number of decision variables are 40, 80 and 120. The strong diversification ability of MOEA/D-PBI with $\theta = 5$ is demonstrated in Table IV on the test problems with three to eight objectives where good results are not obtained by MOEA/D-PBI with $\theta = 0$. By utilizing the strength of each specification simultaneously, good results are obtained on all test instances in Table IV and Table V by MOEA/D-2PBI. This explains the high search ability of our proposed MOEA/D-2PBI algorithm with the use of $\theta = 0$ and $\theta = 5$ simultaneously.

V. PERFORMANCE ON MANY-OBJECTIVE KNAPSACK PROBLEMS AND A REAL-WORLD PROBLEM

In this section, we further examine the performance of our proposed MOEA/D-2PBI algorithm on the two-, four-, six-, and eight-objective knapsack problems with 500 items. Specifically, we use the two-objective 500-item knapsack problem in [35] and the 500-item knapsack problems with four,

six, and eight objectives in [6]. The detailed formulations of these test problems are also available in the supplementary file. In this paper, the two-objective 500-item problem, four-objective 500-item problem, six-objective 500-item problem, and the eight-objective 500-item problem are referred to as 2-500, 4-500, 6-500 and 8-500 problems, respectively. In accordance with the specifications in [6], the population size is set as 100 for the 2-500 problem, 120 for the 4-500 problem, 126 for the 6-500 problem, and 120 for the 8-500 problem. For the coding implementation, a binary string of length 500 is used. The termination condition is set as 400,000 solution evaluations. As the genetic operators, the uniform crossover and the bit-flip mutation are used. The probability for crossover is set as 1 and the probability for mutation is set as 1/500.

Table VI presents the average hypervolume values of the eleven algorithms on the 2-500, 4-500, 6-500, and 8-500 problems. The symbols “+”, “-”, and “=” are used to show whether the compared algorithm has better, worse, or equivalent performance to the MOEA/D-2PBI statistically. In Table VI, MOEA/D-PBI with $\theta = 0$ obtains the best performance on 2-500, 4-500, 6-500, and 8-500 problems. Although MOEA/D-2PBI do not show the best performance, it has similar performance to MOEA/D-PBI with $\theta = 0$ (i.e., it obtains the second-best performance on the many-objective knapsack problems). Fig. 14 displays the solutions obtained by the eleven algorithms on 2-500 problem. The black dotted line is the true Pareto front (which is provided by the authors of [35]) and the blue points are the solutions obtained by each algorithm. Whereas the solution set obtained by MOEA/D-PBI with $\theta = 0$ has larger spread, it has smaller number of solutions than the other solution sets. In contrast, the solution set obtained by MOEA/D-PBI with $\theta = 5$ has more solutions but smaller spread. It can also be seen that NSGA-III, RVEA, AdaW, MOEA/D-APS, Two_Arch2, θ -DEA, and R2HCA-EMOA have similar performance to MOEA/D-PBI with $\theta = 5$ with respect to their spread of solution set. For the proposed MOEA/D-2PBI, the obtained solution set has the advantage of each specification: large spread (by $\theta = 0$) and many solutions (by $\theta = 5$).

TABLE VI.
THE AVERAGE HYPERVOLUME VALUE OF EACH ALGORITHM ON THE MANY-OBJECTIVE KNAPSACK TEST PROBLEMS. THE BEST VALUE IS HIGHLIGHTED BY BOLD
AND SHADED IN GRAY. THE SECOND-BEST RESULT IS ITALICIZED.

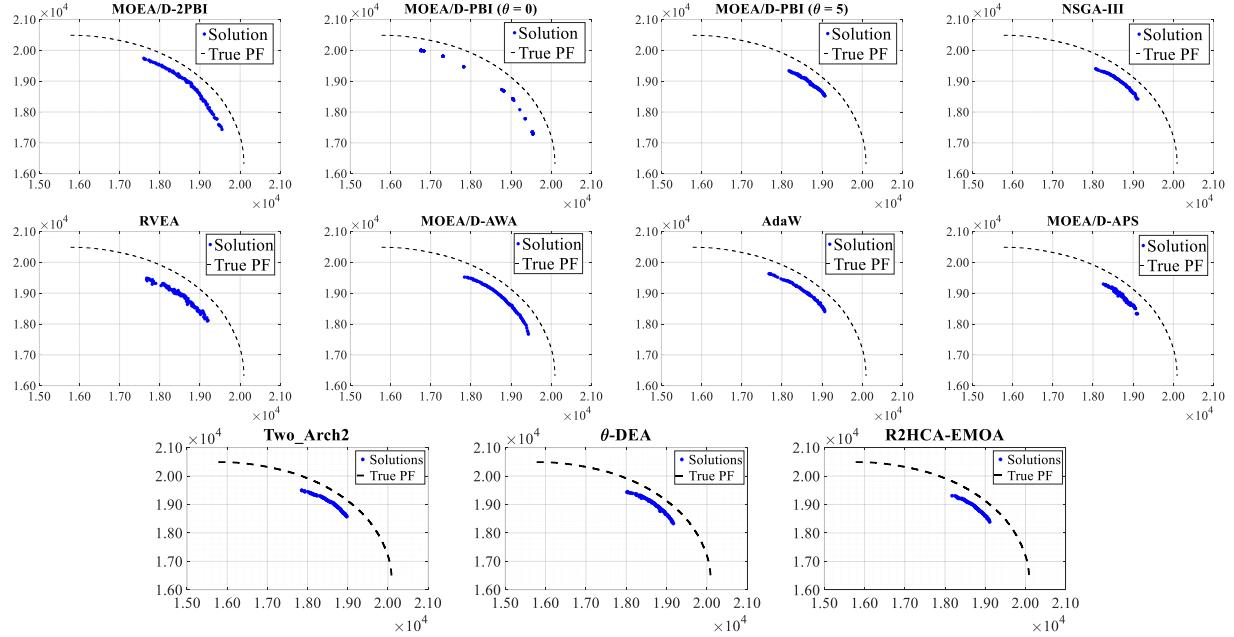


Fig. 14. The obtained solution sets for the two-objective knapsack problem with 500 decision variables by different algorithms. The blue points are the obtained solutions by each algorithm, and the black dotted line is the true Pareto front.

Next, the usefulness of the proposed MOEA/D-2PBI algorithm is further demonstrated on a four-objective conceptual marine design problem [36], [37]. In the conceptual marine design problem, the objectives are (1) to minimize the cost of transportation, (2) to minimize the weight of the light ship, and (3) to maximize the annual cargo transport capacity. In addition, the involved constraints are reformulated as the fourth objective to minimize the sum of constraint violations. Since these four objectives have totally different ranges, a simple normalization mechanism in [42] is used in MOEA/D-2PBI:

$$z_i := \frac{z_i - z_i^*}{z_i^N - z_i^* + \epsilon}, \quad (12)$$

where z_i^* and z_i^N are the estimated ideal and nadir points. The normalization parameter ϵ is set to 10^{-6} to prevent the denominator from being zero. For fair comparison, the same normalization mechanism as (12) is used in MOEA/D-APS.

With respect to the other algorithms, their original implementations are used. This is because each algorithm has its own mechanism to handle objectives with different scales. The population size N is specified as 120 for all algorithms. We consider the conceptual marine design problem as an expensive real-world problem where only 1,200 solutions can be evaluated in the execution of each algorithm, which is equivalent to 10 generations.

Table VII shows our experimental results. In Table VII, MOEA/D-2PBI has the best performance, followed by MOEA/D-AWA, AdaW, and R2HCA-EMOA. Fig. 15 shows the solution sets obtained by these four algorithms. Compared to AdaW and R2HCA-EMOA, the solution set by MOEA/D-2PBI has a larger spread. In comparison with MOEA/D-AWA, the solution set by MOEA/D-2PBI has a better convergence. These results demonstrate the usefulness of MOEA/D-2PBI in solving real-world problems with a limited computational budget.

TABLE VII.
THE AVERAGE HYPERVOLUME VALUE OF EACH ALGORITHM ON THE FOUR-OBJECTIVE CONCEPTUAL MARINE DESIGN PROBLEM. THE BEST VALUE IS HIGHLIGHTED BY BOLD.

M	MOEA/D-2PBI	NSGA-III	RVEA	MOEA/D-AWA	AdaW	MOEA/D-APS	Two_Arch2	θ-DEA	R2HCA-EMOA
4	0.7146	0.6202 -	0.6205 -	0.7046 =	0.6870 -	0.3386 -	0.6019 -	0.6220 -	0.6835 =

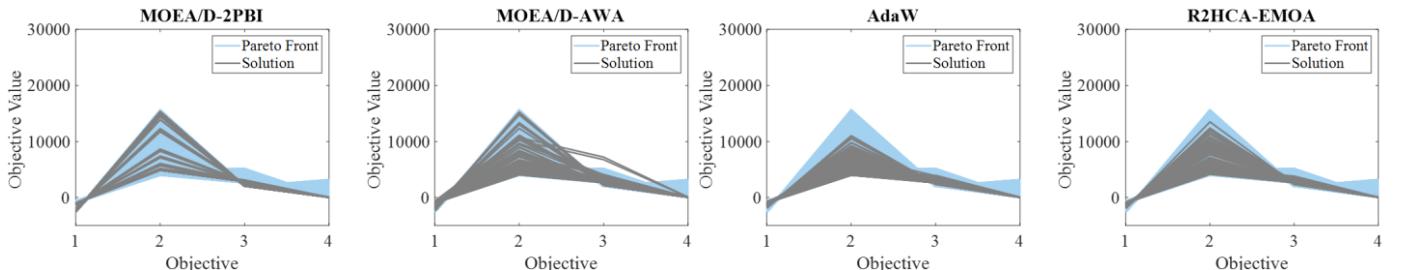


Fig. 15. The obtained solution sets for the four-objective conceptual marine design problem by MOEA/D-2PBI, MOEA/D-AWA, AdaW, and R2HCA-EMOA.

VI. CONCLUSIONS

In order to simultaneously utilize the advantages of small and large penalty parameter values of the PBI function in MOEA/D, we proposed MOEA/D-2PBI in this paper. Two penalty parameter values (i.e., $\theta = 0$ and $\theta = 5$) were used simultaneously in MOEA/D-2PBI. The basic idea was to simultaneously utilize high convergence ability of a small penalty parameter value and high diversification ability of a large penalty parameter value. This idea was implemented as the proposed MOEA/D-2PBI algorithm by using two populations which are evolved in a collaborative manner. The original MOEA/D-PBI and the proposed MOEA/D-2PBI were applied to a wide range of test problems with regular and irregular Pareto fronts. Our experimental results showed that a different penalty parameter value is needed for a different test problem. MOEA/D-2PBI outperformed the original MOEA/D-PBI (i.e., with a single penalty parameter value) on many test problems. MOEA/D-2PBI was also compared with eight state-of-the-art EMO algorithms for many-objective optimization and good results were obtained. The strong convergence ability of the proposed algorithm was also demonstrated by computational experiments on a difficult many-objective test problem (i.e., HTNY19) and many-objective knapsack problems. Furthermore, the proposed algorithm was shown to perform well on a real-world conceptual marine design problem with a limited computational budget.

We also explained why the PBI function with $\theta = 0$ can find the extreme solutions of convex Pareto fronts, which cannot be obtained by the PBI function with $\theta = 5$ (i.e., by the standard setting of the PBI function). That is, it was clearly demonstrated that the PBI function with $\theta = 0$ can help the standard MOEA/D algorithm (i.e., MOEA/D-PBI with $\theta = 5$) to increase both the convergence ability and the diversification ability by finding the extreme solutions of convex Pareto fronts.

Our current implementation of MOEA/D-2PBI chooses one of the two populations as the final output. Alternatively, the final output can be selected from the merged population using a subset selection method. This is an interesting and promising future research topic where different subset selection methods will be examined. In the supplementary file, some preliminary results (Tables S4-S5 and Figs. S2-S3) are given. It is also worthwhile to investigate how continuous values of θ can be used in the proposed algorithm.

REFERENCES

- [1] J. Wang *et al.*, “Multiobjective vehicle routing problems with simultaneous delivery and pickup and time windows: Formulation, instances, and algorithms,” *IEEE Transactions on Cybernetics*, vol. 46, no. 3, pp. 582-594, 2016.
- [2] M. Zhou and J. Liu, “A two-phase multiobjective evolutionary algorithm for enhancing the robustness of scale-free networks against multiple malicious attacks,” *IEEE Transactions on Cybernetics*, vol. 47, no. 2, pp. 539-552, 2017.
- [3] L. Wang *et al.*, “Influence spread in Geo-Social networks: A multiobjective optimization perspective,” *IEEE Transactions on Cybernetics*, vol. 51, no. 5, pp. 2663-2675, 2021.
- [4] C. A. Coello Coello, “Evolutionary multi-objective optimization: a historical view of the field,” *IEEE Computational Intelligence Magazine*, vol. 1, no. 1, pp. 28-36, 2006.
- [5] Q. Zhang and H. Li, “MOEA/D: A multiobjective evolutionary algorithm based on decomposition,” *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712-731, 2007.
- [6] H. Ishibuchi, N. Akedo, and Y. Nojima, “Behaviour of multiobjective evolutionary algorithms on many-objective knapsack problems,” *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 264-283, 2015.
- [7] H. Ishibuchi, K. Doi, and Y. Nojima, “Characteristics of many-objective test problems and penalty parameter specification in MOEA/D,” in *Proceedings of 2016 IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, Canada, July 2016, pp. 1115-1122.
- [8] S. Yang, S. Jiang, and Y. Jiang, “Improving the multiobjective evolutionary algorithm based on decomposition with new penalty schemes,” *Soft Computing*, vol. 21, pp. 4677-4691, 2017.
- [9] H. Ishibuchi, N. Akedo, and Y. Nojima, “A study on the specification of a scalarizing function in MOEA/D for many-objective knapsack problems,” *Lecture Notes in Computer Science Volume 7997: Learning and Intelligent Optimization*, pp. 231-246, Springer, 2013.
- [10] M. Ming, R. Wang, Y. Zha, and T. Zhang, “Pareto adaptive penalty-based boundary intersection method for multi-objective optimization,” *Information Sciences*, vol. 414, pp. 158-174, 2017.
- [11] Y. Qi, D. Liu, X. Li, J. Lei, X. Xu, and Q. Miao, “An adaptive penalty-based boundary intersection method for many-objective optimization problem,” *Information Sciences*, vol. 509, pp. 356-375, 2020.
- [12] X. Ma, Y. Yu, X. Li, Y. Qi, and Z. Zhu, “A survey of weight vector adjustment methods for decomposition-based multiobjective evolutionary algorithms,” *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 4, pp. 634-649, 2020.
- [13] Y. Hua, Q. Liu, K. Hao, and Y. Jin, “A survey of evolutionary algorithms for multi-objective optimization problems with irregular Pareto fronts,” *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 2, pp. 303-318, 2021.
- [14] Y. Qi *et al.*, “MOEA/D with adaptive weight adjustment,” *Evolutionary Computation*, vol. 22, no. 2, pp. 231-264, 2014.
- [15] M. Asafuddoula, H. K. Singh, and T. Ray, “An enhanced decomposition-based evolutionary algorithm with adaptive reference vectors,” *IEEE Transactions on Cybernetics*, vol. 48, no. 8, pp. 2321-2334, 2018.
- [16] X. Cai, Z. Mei, Z. Fan, “A decomposition-based many-objective evolutionary algorithm with two types of adjustments for direction vectors,” *IEEE Transactions on Cybernetics*, vol. 48, no. 8, pp. 2335-2348, 2018.
- [17] M. Li and X. Yao, “What weights work for you? Adapting weights for any Pareto front shape in decomposition-based evolutionary multiobjective optimisation,” *Evolutionary Computation*, vol. 28, no. 2, pp. 227-253, 2020.
- [18] Y. Liu, H. Ishibuchi, N. Masuyama, and Y. Nojima, “Adapting reference vectors and scalarizing functions by growing neural gas to handle irregular Pareto fronts,” *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 3, pp. 439-453, 2020.
- [19] Q. Liu, Y. Jin, M. Heiderich, T. Rodemann, and G. Yu, “An adaptive reference vector-guided evolutionary algorithm using growing neural gas for many-objective optimization of irregular problems,” *IEEE Transactions on Cybernetics*, Early Access.
- [20] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, “Towards understanding the cost of adaptation in decomposition-based optimization algorithms,” *IEEE International Conference on Systems, Man, and Cybernetics*, Manchester, UK, October 2013, pp. 615-620.
- [21] I. Das and J. E. Dennis, “Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems,” *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 631-657, 1998.
- [22] L. While, L. Bradstreet, and L. Barone, “A fast way of calculating exact hypervolumes,” *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, pp. 86-95, 2012.
- [23] K. Shang, H. Ishibuchi, M.-L. Zhang, and Y. Liu, “A new R2 indicator for better hypervolume approximation,” in *Proceedings of the 2018 Genetic and Evolutionary Computation Conference (GECCO)*, Kyoto, Japan, July 2018, pp. 745-752.
- [24] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, “Modified distance calculation in generational distance and inverted generational distance,” in *Proceedings of International Conference on Evolutionary Multi-Criterion Optimization*, March 2015, pp. 110-125.
- [25] H. Ishibuchi, R. Imada, N. Masuyama, and Y. Nojima, “Comparison of hypervolume, IGD and IGD⁺ from the viewpoint of optimal distributions of solutions,” in *Proceedings of International Conference on Evolutionary Multi-Criterion Optimization*, March 2019, pp. 332-345.
- [26] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable test problems for evolutionary multiobjective optimization,” in *Evolutionary Multiobjective Optimization*, London, U.K.: Springer-Verlag, pp. 105-145, 2005.
- [27] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, “Performance of decomposition-based many-objective algorithms strongly depends on

- Pareto front shapes," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 2, pp. 169-190, 2017.
- [28] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [Educational Forum]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73-87, 2017.
- [29] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773-791, 2016.
- [30] Y.-H. Zhang *et. al.*, "DECAL: Decomposition-based coevolutionary algorithm for many-objective optimization," *IEEE Transactions on Cybernetics*, vol. 49, no. 1, pp. 27-41, 2019.
- [31] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577-601, 2014.
- [32] H. Ishibuchi, T. Matsumoto, N. Masuyama, and Y. Nojima, "Effects of dominance resistant solutions on the performance of evolutionary multi-objective and many-objective algorithms," in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference (GECCO)*, Cancún, Mexico, June 2020, pp. 507-515.
- [33] L. M. Pang, H. Ishibuchi, and K. Shang, "NSGA-II with simple modification works well on a wide variety of many-objective problems," *IEEE Access*, vol. 8, pp. 190240-190250, 2020.
- [34] L. M. Pang, K. Shang, L. Chen, H. Ishibuchi, and W. Chen, "Proposal of a new test problem for large-scale multi- and many-objective optimization," *IEEE International Conference on Systems, Man, and Cybernetics*, Melbourne, Australia, October 2021, pp. 484-491.
- [35] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257-271, 1999.
- [36] M. G. Parsons and R. L. Scott, "Formulation of multicriterion design optimization problems for solution with scalar numerical optimization methods," *Journal of Ship Research*, vol. 48, no. 1, pp. 61-76, 2004.
- [37] R. Tanabe and H. Ishibuchi, "An easy-to-use real-world multi-objective optimization problem suite," *Applied Soft Computing*, vol. 89, 2020, Article no. 106078.
- [38] H. Wang, L. Jiao, and X. Yao, "Two_Arch2: An improved two-archive algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 524-541, 2015.
- [39] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 16-37, 2016.
- [40] K. Shang and H. Ishibuchi, "A new hypervolume-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 5, pp. 839-852, 2020.
- [41] X. Cai *et al.*, "A decomposition-based coevolutionary multiobjective local search for combinatorial multiobjecive optimization," *Swarm and Evolutionary Computation*, vol. 49, pp. 178-193, 2019.
- [42] H. Ishibuchi, N. Akedo, and Y. Nojima, "On the effect of normalization in MOEA/D for multi-objective and many-objective optimization," *Complex & Intelligent Systems*, vol. 3, no. 4, pp. 279-294, 2017.
- [43] L. Chen, K. Deb, H.-L. Liu, Q. Zhang, "Effect of objective normalization and penalty parameter on penalty boundary intersection decomposition-based evolutionary many-objective optimization algorithms," *Evolutionary Computation*, vol. 29, no. 1, pp. 157-186, 2021.



Hisao Ishibuchi (M'93–SM'10–F'14) received the B.S. and M.S. degrees from Kyoto University in 1985 and 1987, respectively, and the Ph.D. degree from Osaka Prefecture University in 1992. From 1987 to 2017, he was with Osaka Prefecture University. Since 2017, he has been a Chair Professor at Southern University of Science and Technology, Shenzhen, China.

Dr. Ishibuchi was the IEEE CIS Vice-President for Technical Activities from 2010 to 2013 and the Editor-in-Chief of the IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE from 2014 to 2019. He is currently an IEEE CIS AdCom Member, an IEEE CIS Distinguished Lecturer, and the General Chair of IEEE WCCI 2024.



Ke Shang received his BS and PhD degrees from Xi'an Jiaotong University in 2009 and 2016, respectively. He is currently a Research Assistant Professor in Southern University of Science and Technology. His research interests include multiobjective optimization and artificial intelligence. He received GECCO 2018 and 2021 Best Paper Awards, and PPSN 2020 Best Paper Nomination.



Lie Meng Pang (M'18) received her Bachelor of Engineering degree in Electronic and Telecommunication Engineering and Ph.D. degree in Electronic Engineering from the Faculty of Engineering, Universiti Malaysia Sarawak, in 2012 and 2018, respectively. She is currently a Research Associate with the Department of Computer Science and Engineering, Southern University of Science and Technology (SUSTech), China. Her research interests include evolutionary multiobjective optimization and fuzzy systems.

Supplementary File for “Use of Two Penalty Values in Multi-objective Evolutionary Algorithm based on Decomposition”

Lie Meng Pang, *Member, IEEE*, Hisao Ishibuchi, *Fellow, IEEE*, and Ke Shang, *Member, IEEE*

1. Search Behavior of MOEA/D-PBI with Different θ Values

Fig. S1 shows the search behavior of MOEA/D-PBI with different penalty values ($\theta \in [0, 10]$) on DTLZ2, DTLZ3, Minus-DTLZ2, and Minus-DTLZ3 (with three, four, six, and eight objectives). Based on Fig. S1, it can be seen that different θ values have different performance for different types of problems. In particular, for DTLZ2, any value of θ larger than 1 performs well independent of the number of objectives. For DTLZ3, it is difficult to determine which value of θ is appropriate since different θ values are needed for different number of objectives. For Minus-DTLZ2 and Minus-DTLZ3, it seems that a small value of θ (which is smaller than 1) is a better choice independent of the number of objectives. However, its best specification depends on the number of objectives.

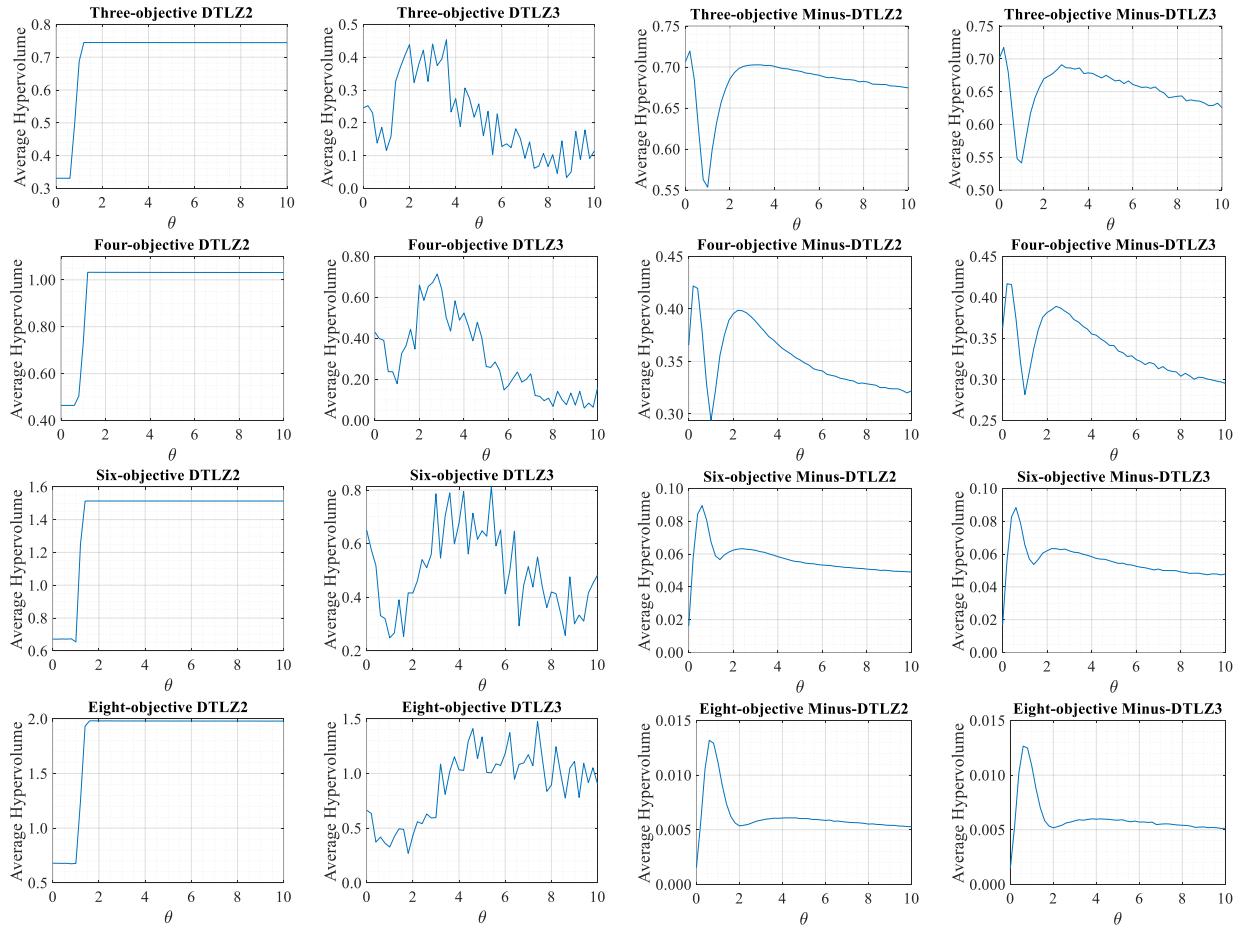


Fig. S1: The performance of the standard MOEA/D-PBI with a single population using a wide variety of θ values in the interval of $[0, 10]$ on DTLZ2, DTLZ3, Minus-DTLZ2, and Minus-DTLZ3.

2. Effect of Population Size: N vs $2N$

In this section, an experiment that tests some other algorithms with a double population size was conducted. In the experiment, the population size for the proposed MOEA/D-2PBI is specified as N . The population size for the other algorithms is specified as $2N$. However, since the Das and Dennis method is used to generate uniformly distributed weight vectors for decomposition-based EMO algorithms (i.e., NSGA-III, RVEA, MOEA/D-AWA, AdaW, MOEA/D-APS, and θ -DEA), their population size cannot be exactly set as $2N$ in some cases. In this case, their population size is specified as a settable integer closest to $2N$. In Table S1, the actual specifications of the population size for each algorithm are listed.

TABLE S1.
THE POPULATION SIZE FOR EACH ALGORITHM.

M	N	MOEA/D-2PBI	NSGA-III, RVEA, MOEA/D-AWA, AdaW, MOEA/D-APS, θ -DEA	Two_Arch2, R2HCA-EMOA
3	91	91	171	182
4	120	120	220	240
6	126	126	252	252
8	156	156	330	330

Following the parameter specification in Section IV.A of the paper, the termination condition is specified as 27300, 36000, 50400, and 62400 solution evaluations, for $M = 3, 4, 6$, and 8 , respectively. For each algorithm, 31 independent runs are performed. It should also be noted that whereas the proposed MOEA/D-2PBI uses two populations, its final output is selected from one of the two populations with N solutions. For the other compared algorithms, the final output is a population with $2N$ solutions. The experimental results are shown in the following Table S2.

TABLE S2.
THE AVERAGE HYPERVOLUME VALUE OF EACH ALGORITHM ON A GIVEN TEST PROBLEM. THE POPULATION SIZE FOR MOEA/D-2PBI IS SPECIFIED AS N , AND THE
POPULATION SIZE FOR OTHER ALGORITHMS IS SPECIFIED AS $2N$.

Test Instance	M	N	MOEA/D-2PBI	NSGA-III	RVEA	MOEA/D-AWA	AdaW	MOEA/D-APS	Two_Arch2	θ -DEA	R2HCA-EMOA
DTLZ1	3	91	1.117E+0	1.122E+0 +	9.623E-1 -	1.123E+0 +	1.124E+0 +	1.121E+0 +	9.917E-1 -	1.125E+0 +	1.131E+0 +
	4	120	1.374E+0	1.377E+0 +	1.279E+0 -	1.357E+0 -	1.380E+0 +	1.382E+0 +	1.362E+0 -	1.382E+0 +	1.388E+0 +
	6	126	1.750E+0	1.748E+0 =	1.732E+0 -	1.718E+0 -	1.742E+0 =	1.754E+0 +	1.747E+0 -	1.702E+0 =	1.758E+0 +
	8	156	2.114E+0	1.947E+0 =	2.131E+0 +	2.109E+0 =	1.975E+0 =	2.129E+0 +	2.117E+0 +	2.116E+0 +	2.141E+0 +
DTLZ2	3	91	7.437E-1	7.608E-1 +	7.584E-1 +	7.637E-1 +	7.619E-1 +	7.611E-1 +	7.618E-1 +	7.612E-1 +	7.691E-1 +
	4	120	1.030E+0	1.054E+0 +	1.053E+0 +	1.055E+0 +	1.054E+0 +	1.056E+0 +	1.036E+0 +	1.055E+0 +	1.072E+0 +
	6	126	1.512E+0	1.540E+0 +	1.547E+0 +	1.531E+0 +	1.531E+0 +	1.549E+0 +	1.415E+0 -	1.547E+0 +	1.569E+0 +
	8	156	1.978E+0	1.994E+0 +	2.023E+0 +	1.957E+0 =	1.950E+0 -	2.024E+0 +	1.766E+0 -	2.022E+0 +	2.041E+0 +
DTLZ3	3	91	4.485E-1	2.469E-2 -	0.000E+0 -	4.061E-1 =	3.885E-1 =	2.084E-1 -	4.417E-2 -	3.700E-2 -	1.898E-1 -
	4	120	7.855E-1	1.378E-2 -	0.000E+0 -	8.331E-1 +	7.983E-1 =	3.631E-1 -	1.064E-1 -	9.150E-2 -	2.842E-1 -
	6	126	1.463E+0	4.556E-3 -	1.510E-1 -	1.135E-0 -	1.144E+0 -	5.758E-1 -	4.660E-3 -	3.459E-1 -	1.111E+0 -
	8	156	1.901E+0	0.000E+0 -	7.187E-2 -	1.671E+0 -	1.494E+0 -	1.029E+0 =	4.325E-3 -	7.113E-2 -	1.996E+0 +
DTLZ4	3	91	5.785E-1	7.218E-1 +	7.587E-1 +	6.952E-1 +	7.624E-1 +	5.061E-1 =	7.615E-1 +	6.999E-1 +	7.102E-1 +
	4	120	7.807E-1	1.046E+0 +	1.054E+0 +	9.848E-1 +	1.036E+0 +	7.524E-1 =	1.031E+0 +	1.047E+0 +	1.065E+0 +
	6	126	1.226E+0	1.533E+0 +	1.549E+0 +	1.437E+0 +	1.534E+0 +	1.152E+0 =	1.379E+0 +	1.548E+0 +	1.566E+0 +
	8	156	1.765E+0	2.012E+0 +	2.026E+0 +	1.943E+0 +	2.012E+0 +	1.665E+0 =	1.733E+0 -	2.025E+0 +	2.040E+0 +
Minus-DTLZ1	3	91	2.498E-1	2.903E-1 +	2.274E-1 -	2.930E-1 +	3.017E-1 +	2.686E-1 +	3.066E-1 +	2.825E-1 +	3.106E-1 +
	4	120	6.650E-2	7.044E-2 +	3.894E-2 -	6.358E-2 -	8.955E-2 +	7.229E-2 +	9.160E-2 +	4.054E-2 -	9.756E-2 +
	6	126	1.934E-3	1.559E-3 -	7.613E-5 -	2.283E-3 +	3.629E-3 +	1.236E-3 -	3.534E-3 +	5.884E-5 -	4.164E-3 +
	8	156	4.733E-5	6.857E-5 +	3.959E-6 -	2.006E-5 -	8.406E-5 +	8.618E-6 -	7.698E-5 +	5.969E-5 +	1.116E-4 +
Minus-DTLZ2	3	91	7.064E-1	7.164E-1 +	6.963E-1 -	7.193E-1 +	7.361E-1 +	7.151E-1 +	7.354E-1 +	7.158E-1 +	7.343E-1 +
	4	120	3.656E-1	3.758E-1 +	3.459E-1 -	3.526E-1 -	4.357E-1 +	3.815E-1 +	4.377E-1 +	3.938E-1 +	4.515E-1 +
	6	126	5.496E-2	3.887E-2 -	3.538E-2 -	3.954E-2 -	7.411E-2 +	7.091E-2 +	8.397E-2 +	1.182E-2 -	1.039E-1 +
	8	156	6.232E-3	3.731E-3 -	1.885E-3 -	4.742E-3 -	6.815E-3 +	8.162E-3 +	1.034E-2 +	6.646E-3 +	1.654E-2 +
Minus-DTLZ3	3	91	6.950E-1	6.718E-1 -	6.446E-1 -	7.068E-1 +	7.009E-1 =	6.748E-1 -	7.223E-1 +	6.865E-1 -	7.294E-1 +
	4	120	3.516E-1	3.274E-1 -	3.276E-1 -	3.422E-1 -	3.884E-1 +	3.532E-1 =	3.986E-1 +	3.736E-1 +	4.460E-1 +
	6	126	5.157E-2	2.865E-2 -	3.762E-2 -	3.860E-2 -	6.233E-2 +	6.706E-2 +	6.751E-2 +	1.645E-2 -	1.030E-1 +
	8	156	5.883E-3	2.948E-3 -	2.278E-3 -	4.061E-3 -	5.312E-3 -	8.167E-3 +	6.955E-3 +	7.361E-3 +	1.646E-2 +
Minus-DTLZ4	3	91	6.867E-1	7.184E-1 +	6.987E-1 +	7.218E-1 +	7.357E-1 +	4.732E-1 =	7.354E-1 +	7.203E-1 +	7.389E-1 +
	4	120	2.984E-1	3.672E-1 +	3.445E-1 +	3.529E-1 +	4.328E-1 +	2.718E-1 -	4.378E-1 +	3.890E-1 +	4.553E-1 +
	6	126	3.697E-2	2.626E-2 -	1.421E-2 -	3.226E-2 =	6.856E-2 +	1.895E-2 =	8.369E-2 +	2.212E-3 -	1.040E-1 +
	8	156	4.279E-3	2.031E-3 -	7.052E-4 -	4.072E-3 =	5.039E-3 =	1.194E-3 -	9.954E-3 +	5.785E-3 =	1.603E-2 +
$+/-=$			17/12/3	11/21/0	15/12/5	22/4/6	16/8/8	22/10/0	20/10/2	29/3/0	

In Table S2, we can see that the performance of the other compared algorithms (except for RVEA) with $2N$ population size on the 32 test instances are generally better than the proposed MOEA/D-2PBI. However, this table does not necessarily mean that

MOEA/D-2PBI performs worse than the other compared algorithms. This is because N solutions obtained by MOEA/D-2PBI are compared with $2N$ solutions obtained by each of the other algorithms. Since most performance indicators such as the hypervolume and IGD depend on the solution set size (i.e., in general, a larger solution set has a better indicator value), we need to compare different algorithms under the same solution set size.

In Table S3, we further compare the performance of MOEA/D-2PBI with $2N$ population size with other algorithms. This is to ensure that the hypervolume value is calculated based on the solution sets of the same size (i.e., $2N$ in this case) for a fair performance comparison. It can be observed that performance comparison results in Table S3 is similar to the reported results (i.e., Table III on Page 7) in our paper. That is, MOEA/D-2PBI outperforms NSGA-III, RVEA, MOEA/D-AWA, MOEA/D-APS and θ -DEA, and is outperformed by AdaW, Two_Arch2 and R2HCA-EMOA. As shown by other experimental results, MOEA/D-2PBI outperforms these three better algorithms (i.e., AdaW, Two_Arch2 and R2HCA-EMOA) on more difficult test problems.

TABLE S3.

THE AVERAGE HYPERVOLUME VALUE OF EACH ALGORITHM ON A GIVEN TEST PROBLEM. THE POPULATION SIZE FOR MOEA/D-2PBI AND THE COMPARED ALGORITHMS IS SPECIFIED AS $2N$.

Test Instance	M	N	MOEA/D-2PBI	NSGA-III	RVEA	MOEA/D-AWA	AdaW	MOEA/D-APS	Two_Arch2	θ -DEA	R2HCA-EMOA
DTLZ1	3	91	1.125E+0	1.122E+0 =	9.623E-1 -	1.123E+0 =	1.124E+0 =	1.121E+0 =	9.917E-1 -	1.125E+0 =	1.131E+0 +
	4	120	1.384E+0	1.377E+0 -	1.279E+0 -	1.357E+0 -	1.380E+0 -	1.382E+0 -	1.362E+0 -	1.382E+0 -	1.388E+0 +
	6	126	1.755E+0	1.748E+0 -	1.732E+0 -	1.718E+0 -	1.742E+0 -	1.754E+0 -	1.747E+0 -	1.702E+0 -	1.758E+0 +
	8	156	2.129E+0	1.947E+0 =	2.131E+0 =	2.109E+0 -	1.975E+0 -	2.129E+0 =	2.117E+0 =	2.116E+0 -	2.141E+0 +
DTLZ2	3	91	7.582E-1	7.608E-1 +	7.584E-1 =	7.637E-1 +	7.619E-1 +	7.611E-1 +	7.618E-1 +	7.612E-1 +	7.691E-1 +
	4	120	1.051E+0	1.054E+0 +	1.053E+0 +	1.055E+0 +	1.054E+0 +	1.056E+0 +	1.036E+0 -	1.055E+0 +	1.072E+0 +
	6	126	1.543E+0	1.540E+0 -	1.547E+0 +	1.531E+0 -	1.531E+0 -	1.549E+0 +	1.415E+0 -	1.547E+0 +	1.569E+0 +
	8	156	2.017E+0	1.994E+0 -	2.023E+0 +	1.957E+0 -	1.950E+0 -	2.024E+0 +	1.766E+0 -	2.022E+0 +	2.041E+0 +
DTLZ3	3	91	3.417E-1	2.469E-2 -	0.000E+0 -	4.061E-1 =	3.885E-1 =	2.084E-1 =	4.417E-2 -	3.700E-2 -	1.898E-1 -
	4	120	6.637E-1	1.378E-2 -	0.000E+0 -	8.331E-1 =	7.983E-1 =	3.631E-1 -	1.064E-1 -	9.150E-2 -	2.842E-1 -
	6	126	1.435E+0	4.556E-3 -	1.510E-1 -	1.135E+0 -	1.144E+0 -	5.758E-1 -	4.660E-3 -	3.459E-1 -	1.111E+0 -
	8	156	1.938E+0	0.000E+0 -	7.187E-2 -	1.671E+0 -	1.494E+0 -	1.029E+0 -	4.325E-3 -	7.113E-2 -	1.996E+0 +
DTLZ4	3	91	5.859E-1	7.218E-1 +	7.587E-1 +	6.952E-1 +	7.624E-1 +	5.061E-1 =	7.615E-1 +	6.999E-1 +	7.102E-1 +
	4	120	7.921E-1	1.046E+0 +	1.054E+0 +	9.848E-1 +	1.036E+0 +	7.524E-1 =	1.031E+0 +	1.047E+0 +	1.065E+0 +
	6	126	1.360E+0	1.533E+0 +	1.549E+0 +	1.437E+0 +	1.534E+0 +	1.152E+0 -	1.379E+0 +	1.548E+0 +	1.566E+0 +
	8	156	1.799E+0	2.012E+0 +	2.026E+0 +	1.943E+0 +	2.012E+0 +	1.665E+0 -	1.733E+0 =	2.025E+0 +	2.040E+0 +
Minus-DTLZ1	3	91	2.656E-1	2.903E-1 +	2.274E-1 -	2.930E-1 +	3.017E-1 +	2.686E-1 =	3.066E-1 +	2.825E-1 +	3.106E-1 +
	4	120	7.151E-2	7.044E-2 =	3.894E-2 -	6.358E-2 -	8.955E-2 +	7.229E-2 +	9.160E-2 +	4.054E-2 -	9.756E-2 +
	6	126	2.049E-3	1.559E-3 -	7.613E-5 -	2.283E-3 +	3.629E-3 -	1.236E-3 -	3.534E-3 +	5.884E-5 -	4.164E-3 +
	8	156	6.987E-5	6.857E-5 =	3.959E-6 -	2.006E-5 -	8.406E-5 +	8.618E-6 -	7.698E-5 +	5.969E-5 -	1.116E-4 +
Minus-DTLZ2	3	91	7.327E-1	7.164E-1 -	6.963E-1 -	7.193E-1 -	7.361E-1 +	7.151E-1 -	7.354E-1 +	7.158E-1 -	7.343E-1 +
	4	120	4.133E-1	3.758E-1 -	3.459E-1 -	3.526E-1 -	4.357E-1 +	3.815E-1 -	4.377E-1 +	3.938E-1 -	4.515E-1 +
	6	126	6.807E-2	3.887E-2 -	3.538E-2 -	3.954E-2 -	7.411E-2 +	7.091E-2 +	8.397E-2 +	1.182E-2 -	1.039E-1 +
	8	156	8.027E-3	3.731E-3 -	1.885E-3 -	4.742E-3 -	6.815E-3 -	8.162E-3 +	1.034E-2 +	6.646E-3 -	1.654E-2 +
Minus-DTLZ3	3	91	7.196E-1	6.718E-1 -	6.446E-1 -	7.068E-1 -	7.009E-1 -	6.748E-1 -	7.223E-1 =	6.865E-1 -	7.294E-1 +
	4	120	3.956E-1	3.274E-1 -	3.276E-1 -	3.422E-1 -	3.884E-1 -	3.532E-1 -	3.986E-1 =	3.736E-1 -	4.460E-1 +
	6	126	6.054E-2	2.865E-2 -	3.762E-2 -	3.860E-2 -	6.233E-2 +	6.706E-2 +	6.751E-2 +	1.645E-2 -	1.030E-1 +
	8	156	7.146E-3	2.948E-3 -	2.278E-3 -	4.061E-3 -	5.312E-3 -	8.167E-3 +	6.955E-3 -	7.361E-3 =	1.646E-2 +
Minus-DTLZ4	3	91	7.121E-1	7.184E-1 +	6.987E-1 -	7.218E-1 +	7.357E-1 +	4.732E-1 -	7.354E-1 +	7.203E-1 +	7.389E-1 +
	4	120	3.771E-1	3.672E-1 -	3.445E-1 -	3.529E-1 -	4.328E-1 +	2.718E-1 -	4.378E-1 +	3.890E-1 +	4.553E-1 +
	6	126	5.831E-2	2.626E-2 -	1.421E-2 -	3.226E-2 -	6.856E-2 +	1.895E-2 -	8.369E-2 +	2.212E-3 -	1.040E-1 +
	8	156	4.688E-3	2.031E-3 -	7.052E-4 -	4.072E-3 =	5.039E-3 =	1.194E-3 -	9.954E-3 +	5.785E-3 =	1.603E-2 +
$+/-=/$			8/20/4	7/23/2	9/19/4	17/11/4	9/17/6	17/11/4	11/18/3	29/3/0	

3. Selection of the Final Output from the Merged Population

It is possible that a better solution set can be obtained from the two populations than choosing one population. It is likely that some good solutions are included in one population and other good solutions in the other population. This section compares two variants of MOEA/D-2PBI with the proposed MOEA/D-2PBI. For the first variant, the final output of MOEA/D-2PBI is the net set of non-dominated solutions in the merged population (i.e., denoted as MOEA/D-2PBI-ND). The second variant uses a hypervolume-based subset selection algorithm to select the final output from the merged population (i.e., denoted as MOEA/D-2PBI-HVSel). The experimental results are presented in the following two subsections:

3.1 Use of all the Non-Dominated Solutions

The comparison results between MOEA/D-2PBI and MOEA/D-2PBI-ND are shown in Table S4 (as shown on the next page). Table S4 shows that (i) MOEA/D-2PBI-ND and MOEA/D-2PBI have almost the same performance on the DTLZ test problems,

and (ii) MOEA/D-2PBI-ND outperforms MOEA/D-2PBI on the Minus-DTLZ test problems. The first observation for DTLZ can be explained as follows. Since the DTLZ test problems have linear and concave Pareto fronts, all solutions by $\theta = 0$ are only on the boundary of the Pareto front of each test problem. This is because the PBI function with $\theta = 0$ is the same as the weighted sum. Many solutions are overlapping (i.e., the diversity of the obtained solutions is small). For those linear and concave Pareto fronts, the PBI function with $\theta = 5$ works very well. Well-distributed solutions are obtained over the entire Pareto front including its boundary. As a result, the population with $\theta = 5$ is always selected, and the hypervolume contributions of the solutions from $\theta = 0$ are very small in non-dominated solutions of the merged population. The second observation can be explained as follows. Since the Minus-DTLZ test problems have inverted triangular Pareto fronts, well-distributed solutions cannot be obtained by MOEA/D (i.e., by decomposition-based EMO algorithms with the predefined fixed weight vectors [1]). Thus, the obtained solution sets with $\theta = 5$ are not good for Minus-DTLZ. The obtained solution sets with $\theta = 0$ for Minus-DTLZ2-4 can be better since the PBI function with $\theta = 0$ (i.e., the weighted sum) works well on those test problems with convex Pareto front. However, the uniformity of the obtained solution sets by the PBI function with $\theta = 0$ is not very good. Moreover, different solutions are obtained from $\theta = 5$ and $\theta = 0$. As the result, the hypervolume of the non-dominated solutions of the merged population is clearly larger than that of each population. The second observation suggests the usefulness of subset selection from the merged population instead of choosing one of the two populations.

The above-mentioned comparison is somewhat unfair since more solutions are obtained by MOEA/D-2PBI-ND than MOEA/D-2PBI (i.e., since a larger solution set usually has a better hypervolume value than a smaller solution set). For example, Fig. S2 shows the obtained solution sets by MOEA/D-2PBI and MOEA/D-2PBI-ND on the three-objective Minus-DTLZ2 problem. A single run based on the median hypervolume over 31 runs is selected for each algorithm. As can be seen in Fig. S2, the final output of MOEA/D-2PBI-ND includes more solutions than that of MOEA/D-2PBI.

TABLE S4.
THE AVERAGE HYPERVOLUME VALUE OF EACH ALGORITHM ON A GIVEN TEST PROBLEM.

Test Instance	<i>M</i>	MOEA/D-2PBI	MOEA/D-2PBI-ND
DTLZ1	3	1.117E+0 (2.01E-3)	1.117E+0 (2.05E-3) =
	4	1.374E+0 (1.27E-3)	1.374E+0 (1.25E-3) =
	6	1.750E+0 (5.76E-4)	1.750E+0 (5.52E-4) +
	8	2.114E+0 (9.30E-3)	2.117E+0 (7.91E-3) =
DTLZ2	3	7.437E-1 (2.82E-4)	7.437E-1 (2.82E-4) =
	4	1.030E+0 (6.98E-4)	1.030E+0 (6.98E-4) =
	6	1.512E+0 (7.61E-4)	1.512E+0 (7.61E-4) =
	8	1.978E+0 (1.34E-3)	1.978E+0 (1.34E-3) =
DTLZ3	3	4.485E-1 (2.35E-1)	4.485E-1 (2.35E-1) =
	4	7.855E-1 (3.03E-1)	7.855E-1 (3.03E-1) =
	6	1.463E+0 (2.61E-2)	1.463E+0 (2.59E-2) =
	8	1.901E+0 (2.30E-2)	1.902E+0 (2.29E-2) =
DTLZ4	3	5.785E-1 (1.85E-1)	5.785E-1 (1.85E-1) =
	4	7.807E-1 (1.96E-1)	7.807E-1 (1.96E-1) =
	6	1.226E+0 (2.00E-1)	1.226E+0 (2.00E-1) =
	8	1.765E+0 (1.73E-1)	1.765E+0 (1.73E-1) =
Minus-DTLZ1	3	2.498E-1 (4.97E-3)	2.500E-1 (4.95E-3) =
	4	6.650E-2 (1.13E-3)	6.723E-2 (1.15E-3) +
	6	1.934E-3 (7.30E-6)	1.978E-3 (1.23E-5) +
	8	4.733E-5 (3.10E-6)	4.809E-5 (3.09E-6) +
Minus-DTLZ2	3	7.064E-1 (9.79E-5)	7.240E-1 (7.03E-4) +
	4	3.656E-1 (3.93E-4)	4.056E-1 (1.04E-3) +
	6	5.496E-2 (3.85E-4)	5.746E-2 (3.70E-4) +
	8	6.232E-3 (5.50E-5)	7.154E-3 (6.39E-5) +
Minus-DTLZ3	3	6.950E-1 (1.02E-2)	7.111E-1 (9.55E-3) +
	4	3.516E-1 (8.50E-3)	3.833E-1 (1.08E-2) +
	6	5.157E-2 (2.17E-3)	5.468E-2 (2.18E-3) +
	8	5.883E-3 (3.63E-4)	6.537E-3 (3.64E-4) +
Minus-DTLZ4	3	6.867E-1 (1.09E-1)	7.035E-1 (1.12E-1) +
	4	2.984E-1 (1.26E-1)	3.293E-1 (1.41E-1) +
	6	3.697E-2 (1.92E-2)	3.896E-2 (1.98E-2) +
	8	4.279E-3 (2.50E-3)	4.838E-3 (2.74E-3) +
+/-=		16/0/16	

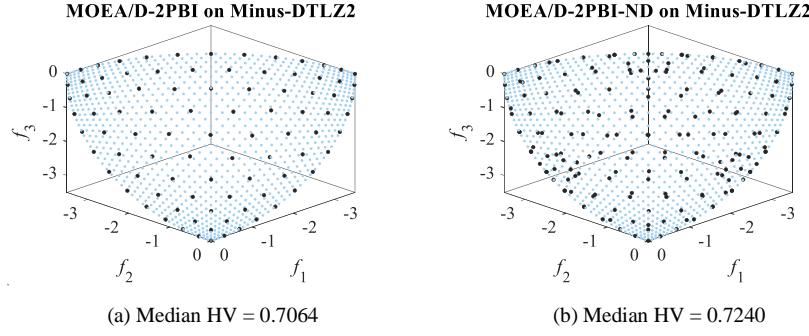


Fig. S2. The obtained solution sets by MOEA/D-2PBI and MOEA/D-2PBI-ND on the three-objective Minus-DTLZ2 problem. A single run based on the median hypervolume value over 31 runs is selected.

3.2 Use of Subset Selection

In MOEA/D-2PBI-HVSel, the final output is selected from a combined population using the greedy hypervolume subset selection process [2]. That is, N solutions are selected from $2N$ solutions in MOEA/D-2PBI-HVSel. The reference point for hypervolume contribution calculation is specified as $(1.1, \dots, 1.1)$ for all test problems independent of the number of objectives.

The comparison results between MOEA/D-2PBI and MOEA/D-2PBI-HVSel are shown in Table S5. As shown in Table S5, it is advantageous to use the hypervolume subset selection process in the proposed MOEA/D-2PBI for maximizing the hypervolume value of the final output.

TABLE S5.

Test Instance	<i>M</i>	MOEA/D-2PBI	MOEA/D-2PBI-HVSeI
DTLZ1	3	1.1169E+0 (2.01E-3)	1.1163E+0 (1.99E-3) =
	4	1.3741E+0 (1.27E-3)	1.3738E+0 (1.56E-3) =
	6	1.7497E+0 (5.76E-4)	1.7498E+0 (6.17E-4) =
	8	2.1135E+0 (9.30E-3)	2.1163E+0 (9.11E-3) =
DTLZ2	3	7.4372E-1 (2.82E-4)	7.4369E-1 (3.03E-4) =
	4	1.0295E+0 (6.98E-4)	1.0299E+0 (5.37E-4) =
	6	1.5117E+0 (7.61E-4)	1.5119E+0 (4.86E-4) =
	8	1.9781E+0 (1.34E-3)	1.9784E+0 (1.44E-3) =
DTLZ3	3	4.4850E-1 (2.35E-1)	3.9033E-1 (2.96E-1) =
	4	7.8548E-1 (3.03E-1)	6.8973E-1 (3.75E-1) =
	6	1.4626E+0 (2.61E-2)	1.4563E+0 (2.53E-2) =
	8	1.9015E+0 (2.30E-2)	1.8956E+0 (2.37E-2) =
DTLZ4	3	5.7853E-1 (1.85E-1)	5.3290E-1 (2.10E-1) =
	4	7.8068E-1 (1.96E-1)	8.1688E-1 (2.32E-1) =
	6	1.2257E+0 (2.00E-1)	1.2881E+0 (1.60E-1) =
	8	1.7647E+0 (1.73E-1)	1.7694E+0 (1.31E-1) =
Minus-DTLZ1	3	2.4983E-1 (4.97E-3)	2.4984E-1 (6.94E-3) =
	4	6.6495E-2 (1.13E-3)	6.7034E-2 (7.97E-4) +
	6	1.9345E-3 (7.30E-6)	1.9738E-3 (1.03E-5) +
	8	4.7327E-5 (3.10E-6)	4.8467E-5 (1.42E-6) +
Minus-DTLZ2	3	7.0640E-1 (9.79E-5)	7.1417E-1 (4.23E-4) +
	4	3.6562E-1 (3.93E-4)	3.9374E-1 (1.08E-3) +
	6	5.4959E-2 (3.85E-4)	5.1697E-2 (3.33E-4) -
	8	6.2322E-3 (5.50E-5)	6.7223E-3 (5.88E-5) +
Minus-DTLZ3	3	6.9499E-1 (1.02E-2)	7.0239E-1 (9.82E-3) +
	4	3.5158E-1 (8.50E-3)	3.7346E-1 (1.03E-2) +
	6	5.1572E-2 (2.17E-3)	5.1501E-2 (2.20E-3) =
	8	5.8831E-3 (3.63E-4)	6.3017E-3 (3.95E-4) +
Minus-DTLZ4	3	6.8669E-1 (1.09E-1)	6.5439E-1 (1.85E-1) -
	4	2.9844E-1 (1.26E-1)	3.7141E-1 (7.99E-2) +
	6	3.6973E-2 (1.92E-2)	4.4081E-2 (1.35E-2) =
	8	4.2792E-3 (2.50E-3)	4.1218E-3 (2.35E-3) -

However, one potential difficulty with the use of the hypervolume subset selection is that uniformly distributed solutions may not always be obtained as the final output. As discussed in [3], the optimal solution distribution of hypervolume maximization may not always result in a uniformly distributed solution set. As an example, Fig. S3 shows the obtained solution sets by MOEA/D-2PBI and MOEA/D-2PBI-HVSel on the three-objective Minus-DTLZ2 problem. A single run based on the median

hypervolume value of 31 runs is selected. The hypervolume value of each solution set is also shown in the caption. It can be seen that whereas the solution set obtained by MOEA/D-2PBI-HVSel has a larger hypervolume value than the solution set obtained by MOEA/D-2PBI, it is not very uniformly distributed. Another difficulty is that an efficient greedy algorithm does not always find a near optimal solution set (as shown in Table S5 where better results are not always obtained from subset selection). Whereas better results will be obtained from exact subset selection algorithms, their use for many-objective problems is usually very difficult due to their long computation time.

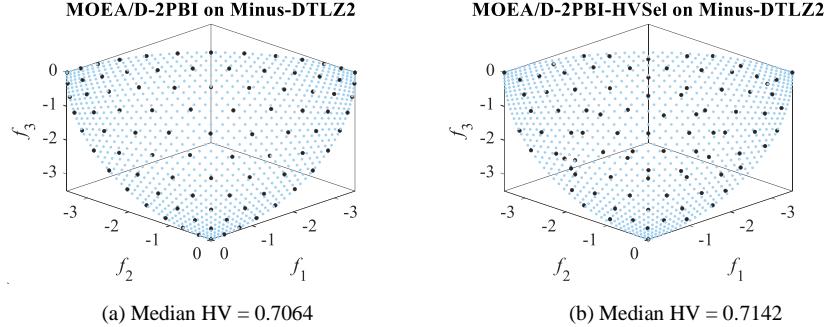


Fig. S3. The obtained solution sets by MOEA/D-2PBI and MOEA/D-2PBI-HVSel on the three-objective Minus-DTLZ2 problem. A single run based on the median hypervolume value over 31 runs is selected.

4. Use of More than Two Populations

This section studies the effect of using more than two populations in the proposed algorithm. Here, a variant of the proposed algorithm with six populations each with $\theta = 0, 1, 2, 3, 4$, and 5 are considered (denoted as MOEA/D-6PBI). The basic algorithm framework of MOEA/D-6PBI is the same as that for MOEA/D-2PBI, explained as follows:

Algorithm S1: MOEA/D-6PBI

- Input:** A multi-objective optimization problem, a termination condition, a population size N , and a neighbourhood size T_n .
- Step 1:** Randomly generate $6N$ solutions as initial solutions, and randomly assign them to the six populations, i.e., Population 1 ($\theta = 0$), Population 2 ($\theta = 1$), Population 3 ($\theta = 2$), Population 4 ($\theta = 3$), Population 5 ($\theta = 4$), and Population 6 ($\theta = 5$).
- Step 2:** Apply the standard MOEA/D procedure with PBI ($\theta = 0$) to each of the N solutions in Population 1 once. Population 2, Population 3, Population 4, Population 5, and Population 6 are updated by each generated solution in Population 1. Populations 2, 3, 4, 5 and 6 are handled as an archive where the entire population is used as the replacement neighbourhood for each solution generated in Population 1. When a new solution is generated in each population, the reference point is always updated.
- Step 3:** Apply the standard MOEA/D procedure with PBI ($\theta = 1$) to each of the N solutions in Population 2 once. Population 1, Population 3, Population 4, Population 5, and Population 6 are updated by each generated solution in Population 2. Populations 1, 3, 4, 5 and 6 are handled as an archive where the entire population is used as the replacement neighbourhood for each solution generated in Population 2. The shared reference point \mathbf{z}^* is always updated.
- Step 4:** Apply the standard MOEA/D procedure with PBI ($\theta = 2$) to each of the N solutions in Population 3 once. Population 1, Population 2, Population 4, Population 5, and Population 6 are updated by each generated solution in Population 3. Populations 1, 2, 4, 5, and 6 are handled as an archive where the entire population is used as the replacement neighbourhood for each solution generated in Population 3. The shared reference point \mathbf{z}^* is always updated.
- Step 5:** Apply the standard MOEA/D procedure with PBI ($\theta = 3$) to each of the N solutions in Population 4 once. Population 1, Population 2, Population 3, Population 5, and Population 6 are updated by each generated solution in Population 4. Populations 1, 2, 3, 5, and 6 are handled as an archive where the entire population is used as the replacement neighbourhood for each solution generated in Population 4. The shared reference point \mathbf{z}^* is always updated.
- Step 6:** Apply the standard MOEA/D procedure with PBI ($\theta = 4$) to each of the N solutions in Population 5 once. Population 1, Population 2, Population 3, Population 4, and Population 6 are updated by each generated solution in Population 5. Populations 1, 2, 3, 4, and 6 are handled as an archive where the entire population is used as the replacement neighbourhood for each solution generated in Population 5. The shared reference point \mathbf{z}^* is always updated.
- Step 7:** Apply the standard MOEA/D procedure with PBI ($\theta = 5$) to each of the N solutions in Population 6 once. Population 1, Population 2, Population 3, Population 4, and Population 5 are updated by each generated solution in Population 6. Populations 1, 2, 3, 4, and 5 are handled as an archive where the entire population is used as the replacement neighbourhood for each solution generated in Population 6. The shared reference point \mathbf{z}^* is always updated.
- Step 8:** Iterate Step 2 to Step 7 until the termination condition is reached. The generation update consists of Step 2 to Step 7. The output is the best population among the six populations, which is selected based on their hypervolume values (calculated using the estimated ideal and nadir points).

The experimental results are presented in Tables S6-S8. Table S6 shows that MOEA/D-6PBI with six populations outperforms MOEA/D-2PBI on 22 test instances. The experimental results suggest that using more than two populations simultaneously can potentially further improve the performance of our proposed algorithm. This observation can be explained as follows. All the test problems in Table S6 are not difficult. Thus, it is likely that most populations in MOEA/D-2PBI and MOEA/D-6PBI are well converged to the Pareto front and they cover the entire Pareto front. Thus, the best population among the six populations is likely to be better than the best population among the two populations as shown in Table S6.

However, MOEA/D-2PBI outperforms MOEA/D-6PBI on multi-objective knapsack problems in Table S7 and large-scale many-objective problems in Table S8. Since our computational experiments on each test problem is performed under the pre-specified computation load (i.e., the pre-specified number of solutions to be examined), the increase in the number of populations decreases the total number of generations. This has a negative effect on the performance of MOEA/D-6PBI on difficult test problems which need a large number of generations. As a result, MOEA/D-2PBI outperforms MOEA/D-6PBI on large-scale many-objective problems in Table S8 and multi-objective knapsack problems in Table S9.

TABLE S6.

THE AVERAGE HYPERVOLUME VALUE OF EACH ALGORITHM ON A GIVEN TEST PROBLEM.

Test Instance	<i>M</i>	MOEA/D-2PBI	MOEA/D-6PBI
DTLZ1	3	1.117E+0 (2.01E-3)	1.117E+0 (1.80E-3) =
	4	1.374E+0 (1.27E-3)	1.374E+0 (1.69E-3) =
	6	1.750E+0 (5.76E-4)	1.750E+0 (7.74E-4) =
	8	2.114E+0 (9.30E-3)	2.099E+0 (1.57E-2) -
DTLZ2	3	7.437E-1 (2.82E-4)	7.446E-1 (5.84E-5) +
	4	1.030E+0 (6.98E-4)	1.032E+0 (1.21E-4) +
	6	1.512E+0 (7.61E-4)	1.513E+0 (1.32E-4) +
	8	1.978E+0 (1.34E-3)	1.980E+0 (7.13E-4) +
DTLZ3	3	4.485E-1 (2.35E-1)	4.372E-1 (3.03E-1) =
	4	7.855E-1 (3.03E-1)	9.267E-1 (1.78E-1) +
	6	1.463E+0 (2.61E-2)	1.484E+0 (1.72E-2) +
	8	1.901E+0 (2.30E-2)	1.952E+0 (1.47E-2) +
DTLZ4	3	5.785E-1 (1.85E-1)	6.776E-1 (1.23E-1) +
	4	7.807E-1 (1.96E-1)	8.682E-1 (1.21E-1) +
	6	1.226E+0 (2.00E-1)	1.352E+0 (1.48E-1) +
	8	1.765E+0 (1.73E-1)	1.798E+0 (1.37E-1) =
Minus-DTLZ1	3	2.498E-1 (4.97E-3)	2.631E-1 (5.38E-4) +
	4	6.650E-2 (1.13E-3)	7.254E-2 (2.79E-4) +
	6	1.934E-3 (7.30E-6)	2.122E-3 (7.81E-6) +
	8	4.733E-5 (3.10E-6)	4.499E-5 (4.89E-6) -
Minus-DTLZ2	3	7.064E-1 (9.79E-5)	7.060E-1 (1.82E-4) -
	4	3.656E-1 (3.93E-4)	3.981E-1 (1.04E-3) +
	6	5.496E-2 (3.85E-4)	6.556E-2 (5.98E-4) +
	8	6.232E-3 (5.50E-5)	1.041E-2 (1.50E-4) +
Minus-DTLZ3	3	6.950E-1 (1.02E-2)	6.871E-1 (1.40E-2) -
	4	3.516E-1 (8.50E-3)	3.763E-1 (1.09E-2) +
	6	5.157E-2 (2.17E-3)	6.178E-2 (1.99E-3) +
	8	5.883E-3 (3.63E-4)	9.149E-3 (5.00E-4) +
Minus-DTLZ4	3	6.867E-1 (1.09E-1)	6.864E-1 (1.09E-1) -
	4	2.984E-1 (1.26E-1)	3.444E-1 (1.24E-1) +
	6	3.697E-2 (1.92E-2)	5.110E-2 (2.21E-2) +
	8	4.279E-3 (2.50E-3)	6.576E-3 (3.87E-3) +
$+/-=$		22/5/5	

TABLE S7.

THE AVERAGE HYPERVOLUME VALUE OF EACH ALGORITHM ON THE HTNY19 PROBLEM. THE NUMBER OF DECISION VARIABLES IS THE SAME AS THE NUMBER OF OBJECTIVES.

Test Instance	<i>M</i>	MOEA/D-2PBI	MOEA/D-6PBI
HTNY19	3	1.1106 (2.01E-3)	1.1203 (2.17E-5) +
	4	1.3728 (1.82E-3)	1.3766 (1.88E-5) +
	6	1.7505 (9.41E-4)	1.7512 (3.27E-5) +
	8	2.1384 (9.90E-5)	2.1384 (2.31E-4) =
$+/-=$		3/0/1	

TABLE S8.

THE AVERAGE HYPERVOLUME VALUE OF EACH ALGORITHM ON THE EIGHT-OBJECTIVE HTNY19 PROBLEM WITH DIFFERENT NUMBER OF DECISION VARIABLE.

Test Instance	<i>D</i>	MOEA/D-2PBI	MOEA/D-6PBI
HTNY19 (<i>M</i> =8)	8	2.1384 (9.90E-5)	2.1384 (2.31E-4) =
	40	2.1372 (4.19e-4)	2.1349 (2.27e-3) -
	80	2.1364 (6.27e-4)	2.1336 (2.30e-3) -
	120	2.1356 (6.51e-4)	2.1330 (2.53e-3) -
$+/-=$		0/3/1	

TABLE S9.

THE AVERAGE HYPERVOLUME VALUE OF EACH ALGORITHM ON THE MANY-OBJECTIVE KNAKPSACK TEST PROBLEMS.

Test Instance	<i>M</i>	MOEA/D-2PBI	MOEA/D-6PBI
2-500	2	3.843E+8 (1.46E+6)	3.765E+8 (1.72E+6) -
4-500	4	1.013E+17 (7.83E+14)	9.551E+16 (9.69E+14) -
6-500	6	2.073E+25 (3.94E+23)	1.892E+25 (3.57E+23) -
8-500	8	4.513E+33 (3.08E+32)	4.140E+33 (5.65E+31) -
$+/-=$		0/4/0	

5. Sensitivity Analysis for the Penalty Parameter Value in the Second Population

In this section, we explain how the penalty parameter values were chosen for the two populations in MOEA/D-2PBI. It has been shown in [4] and [5] that a very small value of θ is good for convergence. Thus, the value 0 is chosen as the parameter to represent a very small value of θ . Since $\theta = 5$ is a frequently used in many MOEA/D-PBI studies (and good experimental results are reported using this setting), we used this setting as a large penalty parameter value. The other values ($\theta \in \{1, 2, 3, 4, 6, 7, 8, 9, 10\}$) were also examined for Population 2. The experimental results are shown in Table S10. It can be seen from Table S10 that the best setting is problem dependent. It can also be seen that similar results are obtained on many test problems from all settings except for $\theta = 1$. Whereas the specification of $\theta = 5$ is not necessarily the best in Table S10, we use this setting since good results are obtained from $\theta = 5$ on the other test problems in the paper and $\theta = 5$ has been frequently used in the literature.

TABLE S10.

THE AVERAGE HYPERVOLUME OF EACH ALGORITHM ON A GIVEN TEST PROBLEM. θ IS SPECIFIED AS 0 IN POPULATION 1. θ_2 DENOTES THE SPECIFIED PENALTY PARAMETER VALUE IN POPULATION 2.

Test Instance	M	$\theta_2 = 5$	$\theta_2 = 1$	$\theta_2 = 2$	$\theta_2 = 3$	$\theta_2 = 4$	$\theta_2 = 6$	$\theta_2 = 7$	$\theta_2 = 8$	$\theta_2 = 9$	$\theta_2 = 10$
DTLZ1	3	1.117E+0	1.076E+0 -	1.117E+0 =	1.116E+0 =	1.116E+0 -	1.115E+0 =	1.113E+0 -	1.115E+0 =	1.115E+0 -	
	4	1.374E+0	1.268E+0 -	1.374E+0 =	1.374E+0 =	1.374E+0 =	1.373E+0 =	1.373E+0 =	1.374E+0 =	1.372E+0 =	
	6	1.750E+0	1.643E+0 -	1.725E+0 -	1.747E+0 -	1.749E+0 =	1.750E+0 +	1.750E+0 +	1.750E+0 +	1.750E+0 +	
	8	2.114E+0	1.902E+0 -	1.889E+0 -	2.074E+0 -	2.101E+0 -	2.118E+0 +	2.121E+0 +	2.123E+0 +	2.124E+0 +	
DTLZ2	3	7.437E-1	6.761E-1 -	7.442E-1 +	7.440E-1 +	7.439E-1 =	7.435E-1 -	7.434E-1 -	7.432E-1 -	7.432E-1 -	7.432E-1 -
	4	1.030E+0	5.985E-1 -	1.031E+0 +	1.030E+0 +	1.030E+0 +	1.029E+0 =	1.029E+0 -	1.029E+0 -	1.029E+0 -	1.028E+0 -
	6	1.512E+0	6.716E-1 -	1.513E+0 +	1.512E+0 +	1.512E+0 +	1.511E+0 -	1.511E+0 =	1.511E+0 -	1.511E+0 -	1.511E+0 -
	8	1.978E+0	6.795E-1 -	1.979E+0 =	1.980E+0 +	1.978E+0 =	1.978E+0 =	1.976E+0 -	1.976E+0 -	1.975E+0 -	
DTLZ3	3	4.485E-1	4.918E-1 =	5.371E-1 +	3.903E-1 =	3.823E-1 =	2.800E-1 -	3.139E-1 =	2.819E-1 -	2.979E-1 =	2.427E-1 -
	4	7.855E-1	4.901E-1 -	9.073E-1 +	8.746E-1 -	7.580E-1 =	7.045E-1 =	6.505E-1 =	6.298E-1 =	5.303E-1 -	6.194E-1 =
	6	1.463E+0	6.546E-1 -	1.449E+0 =	1.458E+0 =	1.463E+0 +	1.444E+0 =	1.404E+0 =	1.402E+0 =	1.347E+0 -	1.310E+0 -
	8	1.901E+0	6.673E-1 -	1.837E+0 -	1.894E+0 =	1.893E+0 =	1.887E+0 -	1.896E+0 =	1.895E+0 =	1.896E+0 =	1.894E+0 =
DTLZ4	3	5.785E-1	4.372E-1 -	4.921E-1 =	5.224E-1 =	5.315E-1 =	5.291E-1 =	5.251E-1 =	5.152E-1 =	4.635E-1 =	5.905E-1 =
	4	7.807E-1	5.021E-1 -	7.910E-1 =	7.997E-1 =	8.125E-1 =	8.639E-1 =	8.621E-1 =	8.272E-1 =	8.425E-1 =	8.078E-1 =
	6	1.226E+0	5.446E-1 -	1.213E+0 =	1.225E+0 =	1.217E+0 =	1.273E+0 =	1.302E+0 =	1.251E+0 =	1.282E+0 =	1.303E+0 =
	8	1.765E+0	6.316E-1 -	1.419E+0 -	1.515E+0 -	1.714E+0 =	1.749E+0 =	1.761E+0 =	1.747E+0 =	1.788E+0 =	1.759E+0 =
Minus-DTLZ1	3	2.498E-1	2.625E-1 +	2.604E-1 +	2.574E-1 +	2.525E-1 +	2.463E-1 -	2.411E-1 -	2.392E-1 -	2.374E-1 -	2.327E-1 -
	4	6.650E-2	7.387E-2 +	7.172E-2 +	6.841E-2 +	6.715E-2 +	6.606E-2 =	6.555E-2 -	6.547E-2 -	6.540E-2 -	6.518E-2 -
	6	1.934E-3	1.676E-3 -	2.125E-3 +	2.002E-3 +	1.934E-3 =	1.936E-3 =	1.933E-3 =	1.939E-3 =	1.934E-3 =	1.935E-3 =
	8	4.733E-5	4.967E-5 +	4.793E-5 =	4.698E-5 =	4.781E-5 =	4.633E-5 =	4.757E-5 =	4.733E-5 =	4.750E-5 =	4.712E-5 =
Minus-DTLZ2	3	7.064E-1	7.063E-1 -	7.064E-1 =	7.064E-1 =	7.064E-1 =	7.064E-1 =	7.064E-1 =	7.064E-1 =	7.064E-1 =	7.064E-1 =
	4	3.656E-1	3.658E-1 +	3.994E-1 +	3.849E-1 +	3.658E-1 =	3.657E-1 =	3.657E-1 =	3.656E-1 =	3.658E-1 =	3.657E-1 =
	6	5.496E-2	6.685E-2 +	6.182E-2 +	6.183E-2 +	5.814E-2 +	5.301E-2 -	5.150E-2 -	4.993E-2 -	4.886E-2 -	4.820E-2 -
	8	6.232E-3	1.104E-2 +	5.035E-3 -	5.941E-3 -	6.260E-3 =	6.093E-3 -	5.911E-3 -	5.712E-3 -	5.559E-3 -	5.411E-3 -
Minus-DTLZ3	3	6.950E-1	6.959E-1 =	6.976E-1 =	6.933E-1 =	6.942E-1 =	6.965E-1 =	6.907E-1 =	6.956E-1 =	6.897E-1 =	6.893E-1 -
	4	3.516E-1	3.599E-1 +	3.798E-1 +	3.637E-1 +	3.543E-1 =	3.534E-1 =	3.532E-1 =	3.527E-1 =	3.522E-1 =	3.528E-1 =
	6	5.157E-2	6.181E-2 +	5.923E-2 +	5.870E-2 +	5.433E-2 +	4.878E-2 -	4.771E-2 -	4.604E-2 -	4.542E-2 -	4.421E-2 -
	8	5.883E-3	1.030E-2 +	5.165E-3 -	5.801E-3 =	5.970E-3 =	5.669E-3 -	5.559E-3 -	5.299E-3 -	5.005E-3 -	4.829E-3 -
Minus-DTLZ4	3	6.867E-1	6.615E-1 -	6.473E-1 =	6.867E-1 =	6.867E-1 =	6.866E-1 -	6.278E-1 -	7.063E-1 =	6.667E-1 -	6.668E-1 -
	4	2.984E-1	3.046E-1 =	3.348E-1 +	3.521E-1 +	2.986E-1 =	3.268E-1 =	3.462E-1 =	2.986E-1 =	3.270E-1 =	3.176E-1 =
	6	3.697E-2	4.579E-2 +	4.226E-2 =	4.539E-2 +	5.026E-2 +	4.159E-2 +	4.035E-2 +	3.511E-2 -	3.946E-2 =	2.758E-2 -
	8	4.279E-3	5.736E-3 =	3.906E-3 -	3.644E-3 -	4.245E-3 -	4.543E-3 +	3.532E-3 -	4.157E-3 -	3.435E-3 -	4.111E-3 -
+/-=		10/18/4	13/7/12	14/5/13	7/2/23	4/11/17	3/10/19	2/14/16	2/14/16	2/17/13	

6. Computational Complexity of the Proposed MOEA/D-2PBI Algorithm

In order to analyze the computational complexity of the proposed MOEA/D-2PBI algorithm, we first consider the computational complexity of the standard MOEA/D algorithm [6]. In each generation, the computational complexity of the standard MOEA/D was reported as $O(MNT_n)$ [6], where M is the number of objectives, N is the population size and T_n is the neighborhood size. In Step 2 and Step 3 of the MOEA/D-2PBI algorithm, the standard MOEA/D procedure is applied to one population and another population is treated as an archive. When a new offspring is generated in these steps, the entire population is used as the replacement neighborhood. Since each newly generated solution needs to be compared with the entire population of N solutions, the computational complexity of MOEA/D-2PBI in one generation is dominated by $O(MN^2)$. In addition, since MOEA/D-2PBI needs to calculate the hypervolume value of two populations at the end of the algorithm in order to determine the final output, the overall complexity of the algorithm is dominated by the hypervolume calculation. In our study, we use the WFG algorithm [7], one of the fastest methods for hypervolume calculation in the literature. The worst-case time complexity of the WFG algorithm was reported as $O(N^{M-1})$ and the lower bound was reported as $\Omega(N^{M/2} \log N)$ [8]. In terms of space complexity, the proposed MOEA/D-2PBI algorithm requires $2N$ space since two populations of size N must be maintained during the search process, which is two times larger than the standard MOEA/D algorithm.

7. Performance Comparison in Terms of Runtime

The comparison with the other algorithms in running time is given in Tables S11-S14 below. In particular, Table S11 shows the average running time of all algorithms on DTLZ and Minus-DTLZ test instances, Table S12 and Table S13 show the average running time of all algorithms on the HTNY19 test instances (with different specifications of the number of objectives and the number of decision variables), and Table S14 shows the average running time of all algorithms on the many-objective knapsack test problems.

Comparing MOEA/D-2PBI to the hypervolume-based algorithm (i.e., R2HCA-EMOA), MOEA/D-2PBI is clearly more efficient. This is because hypervolume calculation is performed only at the final stage of MOEA/D-2PBI in order to choose a better population between the two populations whereas R2HCA-EMOA needs to calculate the hypervolume contribution of each solution every generation. In comparison with AdaW and Two_Arch2, MOEA/D-2PBI seems to have better running efficiency especially when many objectives are involved. NSGA-III, RVEA, MOEA/D-APS, MOEA/D-AWA, and θ -DEA are more efficient in terms of running time than MOEA/D-2PBI.

TABLE S11.
THE AVERAGE RUNTIME (IN SECONDS) OF EACH ALGORITHM ON THE DTLZ AND MINUS-DTLZ TEST PROBLEMS.

Test Instance	M	MOEA/D-2PBI	NSGA-III	RVEA	MOEA/D-AWA	AdaW	MOEA/D-APS	Two_Arch2	θ -DEA	R2HCA-EMOA
DTLZ1	3	5.22	1.20 +	1.06 +	6.62 -	7.95 -	3.68 +	5.32 =	0.89 +	52.00 -
	4	7.52	1.89 +	0.87 +	9.36 -	12.72 -	4.87 +	10.32 -	1.45 +	135.35 -
	6	12.93	2.34 +	1.01 +	13.70 -	20.46 -	6.95 +	22.74 -	1.87 +	278.38 -
	8	62.64	3.24 +	1.57 +	18.29 +	46.36 +	8.90 +	42.14 +	2.83 +	628.29 -
DTLZ2	3	5.18	1.19 +	0.56 +	6.70 -	10.41 -	3.55 +	9.88 -	0.95 +	80.10 -
	4	7.49	1.55 +	0.74 +	9.83 -	18.36 -	4.81 +	20.02 -	1.31 +	199.50 -
	6	12.22	2.31 +	1.04 +	14.59 -	33.90 -	6.90 +	36.39 -	2.01 +	410.37 -
	8	42.52	3.32 +	1.46 +	20.26 +	127.76 -	8.77 +	64.28 -	2.89 +	846.73 -
DTLZ3	3	5.23	1.18 +	0.57 +	6.62 -	5.31 =	3.63 +	3.23 +	0.94 +	39.79 -
	4	7.56	1.63 +	0.74 +	9.67 -	9.83 -	4.92 +	5.92 +	1.25 +	100.53 -
	6	11.85	2.17 +	1.05 +	11.98 =	18.10 -	7.00 +	18.94 -	1.71 +	221.82 -
	8	33.11	2.72 +	1.42 +	17.04 +	42.23 -	8.96 +	35.56 -	2.29 +	563.18 -
DTLZ4	3	5.33	1.33 +	0.55 +	6.88 -	10.19 -	3.67 +	8.71 -	1.03 +	65.84 -
	4	7.57	1.53 +	0.76 +	9.10 -	18.63 -	4.96 +	18.16 -	1.23 +	239.30 -
	6	11.09	2.19 +	1.08 +	13.36 -	35.13 -	7.07 +	34.35 -	1.80 +	409.72 -
	8	15.88	3.07 +	1.46 +	18.83 -	145.28 -	8.99 +	61.05 -	2.59 +	866.68 -
Minus-DTLZ1	3	5.24	1.21 +	0.39 +	6.62 -	9.14 -	3.60 +	8.30 -	0.94 +	69.76 -
	4	7.46	1.48 +	0.48 +	8.42 -	15.94 -	4.85 +	16.98 -	1.20 +	193.92 -
	6	11.07	2.43 +	0.68 +	12.99 -	30.83 -	7.00 +	32.42 -	1.99 +	408.30 -
	8	17.43	3.28 +	0.86 +	17.85 -	145.25 -	8.78 +	57.10 -	2.66 +	867.84 -
Minus-DTLZ2	3	5.28	1.25 +	0.48 +	7.10 -	9.74 -	3.57 +	8.67 -	0.98 +	80.37 -
	4	7.72	1.53 +	0.57 +	8.73 -	18.42 -	4.82 +	17.86 -	1.26 +	228.49 -
	6	12.73	2.47 +	0.77 +	13.17 -	34.98 -	7.00 +	33.39 -	2.11 +	426.90 -
	8	44.94	3.30 +	1.05 +	18.48 +	148.93 -	8.73 +	58.84 -	2.68 +	887.98 -
Minus-DTLZ3	3	5.33	1.21 +	0.49 +	6.44 -	7.71 -	3.66 +	7.29 -	0.95 +	57.72 -
	4	7.73	1.51 +	0.58 +	8.51 -	14.68 -	4.93 +	14.76 -	1.22 +	174.01 -
	6	12.81	2.42 +	0.78 +	13.12 -	28.46 -	7.08 +	30.13 -	2.06 +	375.88 -
	8	42.67	3.27 +	1.07 +	18.52 +	139.59 -	8.92 +	54.31 -	2.68 +	828.69 -
Minus-DTLZ4	3	5.31	1.26 +	0.48 +	6.72 -	9.63 -	3.68 +	8.71 -	1.02 +	83.35 -
	4	7.71	1.57 +	0.59 +	8.97 -	18.35 -	4.93 +	18.02 -	1.28 +	229.12 -
	6	12.2	2.52 +	0.78 +	13.53 -	34.73 -	7.17 +	34.14 -	2.21 +	431.38 -
	8	31.03	3.40 +	1.04 +	18.93 +	153.17 -	8.97 +	60.44 -	2.83 +	878.54 -
+/-=		32/0/0	32/0/0	6/25/1	1/30/1	32/0/0	3/28/1	32/0/0	0/32/0	

TABLE S12.
THE AVERAGE RUNTIME (IN SECONDS) OF EACH ALGORITHM ON THE HTNY19 PROBLEM. THE NUMBER OF DECISION VARIABLES IS THE SAME AS THE NUMBER OF OBJECTIVES.

Test Instance	M	MOEA/D-2PBI ($\theta = 5$ and $\theta = 0$)	MOEA/D-PBI ($\theta = 0$)	MOEA/D-PBI ($\theta = 5$)	NSGA-III	RVEA	MOEA/D-AWA	AdaW	MOEA/D-APS	Two_Arch2	θ -DEA	R2HCA-EMOA
HTNY19	3	89.08	60.91 +	61.92 +	12.24 +	10.12 +	62.68 +	125.70 -	61.96 +	70.20 +	9.81 +	679.31 -
	4	127.48	82.67 +	83.67 +	15.37 +	13.86 +	87.31 +	206.02 -	84.65 +	134.42 -	11.88 +	1548.62 -
	6	142.76	89.34 +	90.26 +	17.13 +	14.78 +	95.99 +	257.36 -	90.31 +	210.04 -	13.05 +	2578.70 -
	8	537.75	110.20 +	112.16 +	24.12 +	20.31 +	175.10 +	1528.34 -	112.31 +	378.58 +	19.23 +	6544.40 -
+/-=		4/0/0	4/0/0	4/0/0	4/0/0	4/0/0	0/4/0	4/0/0	2/2/0	4/0/0	0/4/0	

TABLE S13.

THE AVERAGE RUNTIME (IN SECONDS) OF EACH ALGORITHM ON THE EIGHT-OBJECTIVE HTNY19 PROBLEM WITH DIFFERENT NUMBER OF DECISION VARIABLES.

Test Instance	D	MOEA/D-2PBI ($\theta = 5$ and $\theta = 0$)	MOEA/D-PBI ($\theta = 0$)	MOEA/D-PBI ($\theta = 5$)	NSGA-III	RVEA	MOEA/D-AWA	AdaW	MOEA/D-APS	Two_Arch2	θ -DEA	R2HCA-EMOA
HTNY19 ($M=8$)	8	537.75	110.20 +	112.16 +	24.12 +	20.31 +	175.10 +	1528.34 -	112.31 +	378.58 +	19.23 +	6544.40 -
	40	830.7	215.60 +	212.61 +	60.02 +	30.05 +	284.56 +	1852.08 -	125.22 +	472.37 +	52.77 +	7517.23 -
	80	949.73	221.14 +	224.34 +	65.54 +	35.41 +	301.55 +	1940.93 -	132.99 +	536.05 +	59.11 +	7818.07 -
	120	1017.78	234.77 +	241.18 +	75.17 +	39.61 +	301.11 +	1937.15 -	139.90 +	585.33 +	66.52 +	6688.52 -
$+/-=$		4/0/0	4/0/0	4/0/0	4/0/0	4/0/0	4/0/0	0/4/0	4/0/0	4/0/0	4/0/0	0/4/0

TABLE S14.

THE AVERAGE RUNTIME (IN SECONDS) OF EACH ALGORITHM ON THE MANY-OBJECTIVE KNAPSACK TEST PROBLEMS.

Test Instance	M	MOEA/D-2PBI ($\theta = 5$ and $\theta = 0$)	MOEA/D-PBI ($\theta = 0$)	MOEA/D-PBI ($\theta = 5$)	NSGA-III	RVEA	MOEA/D-AWA	AdaW	MOEA/D-APS	Two_Arch2	θ -DEA	R2HCA-EMOA
2-500	2	80.36	51.35 +	52.32 +	20.77 +	19.29 +	55.18 +	112.48 -	52.72 +	74.16 +	18.33 +	683.73 -
4-500	4	88.73	57.42 +	56.86 +	23.09 +	16.90 +	65.29 +	143.32 -	57.13 +	110.74 -	19.88 +	957.00 -
6-500	6	94.21	61.44 +	59.31 +	26.84 +	19.26 +	69.16 +	171.25 -	61.39 +	149.70 -	24.41 +	1288.36 -
8-500	8	94.69	62.42 +	58.54 +	28.34 +	18.26 +	65.88 +	171.58 -	58.81 +	170.99 -	23.31 +	1582.04 -
$+/-=$		4/0/0	4/0/0	4/0/0	4/0/0	4/0/0	4/0/0	0/4/0	4/0/0	1/3/0	4/0/0	0/4/0

8. Obtained Solution Sets by Different Algorithms on the Four-objective Conceptual Marine Design Problem

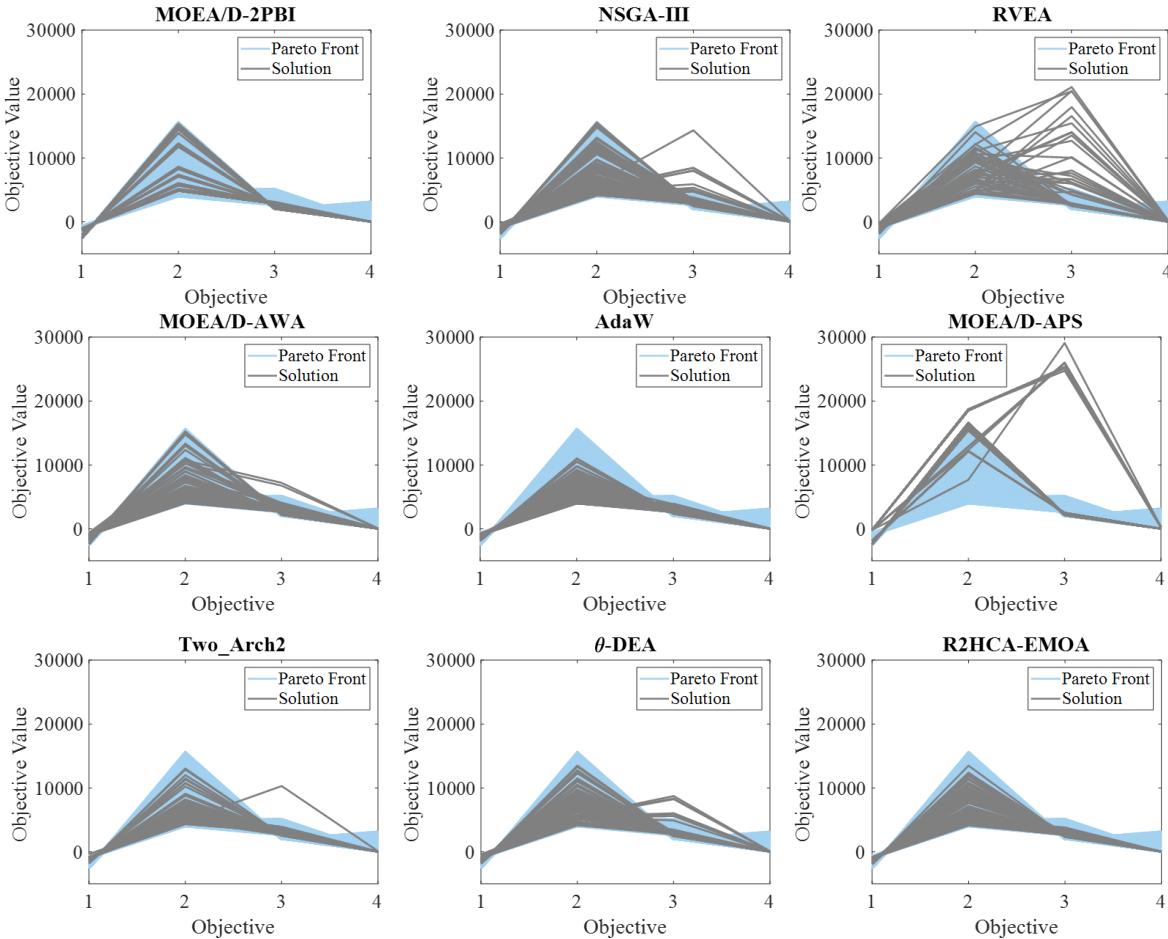


Fig. S4. The obtained solution sets for the four-objective conceptual marine design problem by different algorithms.

9. Formulations of the Multi-objective Knapsack Problems Used in Section V of the Paper

The formulation of the two-objective knapsack problem with 500 items is the one used in Zitzler and Thiele [9], which has the following formulation:

$$\text{Maximize } \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))^\top, \quad (1)$$

$$\text{subject to } \omega_{i1}x_1 + \omega_{i2}x_2 + \dots + \omega_{i500}x_{500} \leq cap_i, \quad i = 1, 2, \quad (2)$$

$$x_j = 0 \text{ or } 1, \quad j = 1, 2, \dots, 500, \quad (3)$$

$$\text{where } f_i(\mathbf{x}) = \alpha_{i1}x_1 + \alpha_{i2}x_2 + \dots + \alpha_{i500}x_{500}, \quad i = 1, 2. \quad (4)$$

Here, \mathbf{x} is a 500-dimensional binary vector, ω_{ij} is the weight of item j with respect to knapsack i , α_{ij} is the profit of item j with respect to knapsack i , and cap_i is the capacity of knapsack i ($i = 1, 2$ and $j = 1, 2, \dots, 500$). In [9], the values of ω_{ij} and α_{ij} were generated at random as integers between 10 and 100, and each knapsack i was specified to have a capacity cap_i of 50% of the total weight of all items in knapsack i . As in [4], we refer this two-objective 500-item knapsack problem as 2-500 problem in this paper. For the four-, six-, and eight-objective 500-item knapsack problems, the problems generated in [4] are used. They were defined in the following manner. The first two objectives were the same as in the 2-500 problem. The remaining objectives were generated as follows:

$$f_i(\mathbf{x}) = \alpha_{i1}x_1 + \alpha_{i2}x_2 + \dots + \alpha_{i500}x_{500}, \quad i = 3, 4, \dots, 8. \quad (5)$$

The same constraint conditions in (2) and (3) were used by all the test problems with 500 items.

References

- [1] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "Performance of decomposition-based many-objective algorithms strongly depends on Pareto front shapes," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 2, pp. 169-190, 2017.
- [2] W. Chen, H. Ishibuchi, and K. Shang, "Fast greedy subset selection from large candidate solution sets in evolutionary multi-objective optimization," *IEEE Transactions on Evolutionary Computation*, Early Access.
- [3] K. Shang, H. Ishibuchi, L. He, and L. M. Pang, "A survey on the hypervolume indicator in evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 1, pp. 1-20, 2021.
- [4] H. Ishibuchi, N. Akedo, and Y. Nojima, "Behaviour of multiobjective evolutionary algorithms on many-objective knapsack problems," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 264-283, 2015.
- [5] H. Ishibuchi, N. Akedo, and Y. Nojima, "A study on the specification of a scalarizing function in MOEA/D for many-objective knapsack problems," *Lecture Notes in Computer Science Volume 7997: Learning and Intelligent Optimization*, pp. 231-246, Springer, 2013.
- [6] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712-731, 2007.
- [7] L. While, L. Bradstreet, and L. Barone, "A fast way of calculating exact hypervolumes," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, pp. 86-95, 2012.
- [8] R. Lacour, K. Klamroth, and C. M. Fonseca, "A box decomposition algorithm to compute the hypervolume indicator," *Computer & Operations Research*, vol. 79, pp. 347-360, 2017.
- [9] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257-271, 1999.