

## Article

# A Dual-Population-Based NSGA-III for Constrained Many-Objective Optimization

Huantong Geng <sup>1,2,\*</sup> , Zhengli Zhou <sup>1</sup> , Junye Shen <sup>1</sup>  and Feifei Song <sup>1</sup> 

<sup>1</sup> School of Computer Science, Nanjing University of Information Science and Technology, Nanjing 210044, China

<sup>2</sup> School of Information Technology, Jiangsu Open University, Nanjing 210036, China

\* Correspondence: htgeng@nuist.edu.cn

**Abstract:** The main challenge for constrained many-objective optimization problems (CMaOPs) is how to achieve a balance between feasible and infeasible solutions. Most of the existing constrained many-objective evolutionary algorithms (CMaOEs) are feasibility-driven, neglecting the maintenance of population convergence and diversity when dealing with conflicting objectives and constraints. This might lead to the population being stuck at some locally optimal or locally feasible regions. To alleviate the above challenges, we proposed a dual-population-based NSGA-III, named DP-NSGA-III, where the two populations exchange information through the offspring. The main population based on the NSGA-III solves CMaOPs and the auxiliary populations with different environment selection ignore the constraints. In addition, we designed an  $\varepsilon$ -constraint handling method in combination with NSGA-III, aiming to exploit the excellent infeasible solutions in the main population. The proposed DP-NSGA-III is compared with four state-of-the-art CMaOEs on a series of benchmark problems. The experimental results show that the proposed evolutionary algorithm is highly competitive in solving CMaOPs.

**Keywords:** constrained many-objective optimization; evolutionary algorithm; dual-population; coevolution;  $\varepsilon$ -constraint handling



**Citation:** Geng, H.; Zhou, Z.; Shen, J.; Son, F. A Dual-Population-Based NSGA-III for Constrained Many-Objective Optimization. *Entropy* **2023**, *25*, 13. <https://doi.org/10.3390/e25010013>

Academic Editors: Marcin Sosnowski, Jaroslaw Krzywanski, Yunfei Gao, Karolina Grabowska, Dorian Skrobek, Ghulam Moeen Uddin, Anna Kulakowska, Anna Zylka and Bachil El Fil

Received: 22 November 2022

Revised: 14 December 2022

Accepted: 15 December 2022

Published: 21 December 2022



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Constrained multi-objective optimization problems (CMOPs) widely exist in the real world, such as the design of the water distribution network [1] and the vehicle routing problem with time windows [2]. In this paper, a dual-population-based NSGA-III with an auxiliary population that ignores constraints is proposed. The offspring of its auxiliary population can guide the evolution of the main population.

In the last decade, several new methods have been proposed to solve unconstrained multi/many-objective optimization problems, such as a new competitive mechanism [3], adaptive operator selection [4], and adaptive weight vector generation strategy [5]. In recent years, CMOPs have attracted increasing research interest in the field of evolutionary computation. Constrained multi-objective evolutionary algorithms (CMOEs), such as PPS [6], ToP [7], and c-DPEA [8], have achieved good performance on CMOPs. However, CMaOPs involve both complex constraints and high-dimensional objective spaces, so it is inefficient to solve CMaOPs directly with existing CMOEs. Specifically, the difficulties in solving CMaOPs are as follows: first, even in the unconstrained case, balancing convergence and diversity in a high-dimensional objective space is difficult to achieve [9]. Although CMOEs can handle constraints, they cannot efficiently select excellent solutions in a high-dimensional objective space. Meanwhile, the algorithm should balance the search between feasible and infeasible regions. Particularly, some CMaOPs have complex constraints. It is difficult for individuals to cross multiple disjoint infeasible regions to approach the constraint Pareto

front (CPF), which leads to poor convergence and diversity of the population [10]. In addition, most evolutionary algorithms just simply push solutions toward the feasible boundary, and it is difficult to balance the search between feasible and infeasible regions [11]. This may cause the population to fall into locally optimal or locally feasible areas, especially when the feasible region is discrete, narrow, or far from the unconstrained Pareto front (UPF) [12]. To alleviate the above problems, this article proposes a dual-population-based constrained many-objective evolutionary algorithm, named DP-NSGA-III. It includes two collaborative and complementary populations. The main characteristics of DP-NSGA-III are as follows:

1. To take advantage of the infeasible solution with excellent objectives, we introduce an auxiliary population, and its environmental selection [13] is different from the main population, using  $p$ -norm to select individuals. Two populations cooperate by exchanging offspring. Each population can evolve more efficiently using information from the other population. Specifically, Population1 is the main population, which finds the CPF by the  $\varepsilon$ -constraint handling method. Population2 with less convergence pressure is the auxiliary population, which will eventually converge to UPF due to the neglected constraints.
2. From the point of view of the problem, neither population solves the original CMOEs. Since the  $\varepsilon$ -constraint boundary replaces the original constraint, Population1 solves a problem with looser constraints than the original problem and is equivalent to the original problem at the end of evolution. Population2 solves the unconstrained problem corresponding to the original problem. This strategy indirectly solves the original CMOEs.
3. To make the main population of DP-NSGA-III infeasibility-driven, we design a monotonically decreasing  $\varepsilon$ -constraint handling function that takes no additional parameters. That is, the  $\varepsilon$ -bounded boundary is larger in the early period and close to 0 in the late period. Therefore, this function has a greater impact in the early evolutionary period. The function aims to involve partially infeasible solutions in the evolution, balancing convergence and diversity by using excellent infeasible solutions.

The rest of this article is organized as follows: Section 2 briefly introduces the existing CMOEs or CMOEAs. The details of the proposed DP-NSGA-III are described in Section 3. Sections 4 and 5 present the experimental setup and experimental results, respectively. Finally, the conclusions and future work are given in Section 6.

## 2. Related Work

This section presents some representative CMOEs and CMOEAs from the perspective of constraint handling methods. According to the research trend of constraint handling techniques in evolutionary algorithms in earlier years, early constraint handling techniques can be broadly categorized into (1) penalty function-based methods [14]; (2) feasibility-oriented approaches [15]; (3) stochastic ranking [16]; (4)  $\varepsilon$ -constrained methods [17]; (5) multi-objective optimization-based approaches [18]; and (6) hybrid methods [19].

The constraint-dominance principle (CDP) [15] was one of the earlier well-known constraint handling methods. NSGA-II [15] employs the CDP to compare pairs of solutions. For two infeasible solutions, the one with less constraint violation is better; the two feasible solutions are still compared by the Pareto dominance. Both C-NSGA-III [20] and C-MOEA/D [20] are well-known representative algorithms that embed this technique. Fan et al. [21] proposed an angle-based CDP, which considers the ratio of feasible solutions and the angular relationship between individuals. Among the above algorithms, NSGA-III is designed for many-objective optimization problems, so C-NSGA-III can solve some CMOEs.

In recent years, some evolutionary algorithms based on multiple stages have been designed to solve CMOPs. PPS [6] divides the search process into two stages: Push and Pull. The Push phase does not consider constraints, which helps to cross large infeasible areas and push the population to UPF. In the Pull phase, an improved  $\varepsilon$ -constraint handling method is used to pull the population back to the CPF. Liu et al. [7] proposed a phased framework ToP.

In ToP, the first stage uses a weighted sum to transform the CMOP into an unconstrained single objective optimization problem to find a promising feasible region. The second stage uses the existing CMOEA to find the CPF of the original problem. CMOEA-MS [22] is also a two-stage evolutionary algorithm, which adjusts the fitness evaluation strategies during the evolutionary process to adaptively balance objective optimization and constraint satisfaction.

In addition, dual-population-based or coevolutionary technique has also been applied to CMOEAs or CMOEAs. c-DPEA [8] is a typical dual-population-based evolutionary algorithm. Population1 uses an adaptive penalty function named saPF to handle infeasible solutions, and Population2 is oriented to feasibility. At the same time, an adaptive fitness function called bCAD is designed to better balance the convergence and diversity in the evolution process. Although c-DPEA can handle CMOPs well, it is inefficient in solving CMOEs. C-TAEA [11] is an evolutionary algorithm with two archives. It maintains two collaborative archives simultaneously. The convergence-oriented archive (CA) aims to optimize constraints and objectives to ensure feasibility and convergence, while the diversity-oriented archive (DA) mainly tends to explore the areas that have not been exploited by the CA without considering feasibility. It is worth noting that C-TAEA is one of the few evolutionary algorithms designed for high-dimensional constrained objective spaces.

### 3. Proposed Algorithm: DP-NSGA-III

This section presents the proposed dual-population-based many-objective evolutionary algorithm, named DP-NSGA-III. The procedure of DP-NSGA-III is shown in Figure 1. DP-NSGA-III is a cooperative evolutionary algorithm that evolves two populations, namely, Population1 and Population2. The populations in DP-NSGA-III are evolved separately, only sharing all the offspring in each generation, where Population1 uses the  $\epsilon$ -constraint handling method and Population2 ignores the constraint. Population2 can provide solutions for Population1 that violate the constraint but have excellent objective values to participate in evolution. The final output population of the algorithm is Population1. The  $\epsilon$ -constraint handling method and the detailed evolution process of each population will be described below.

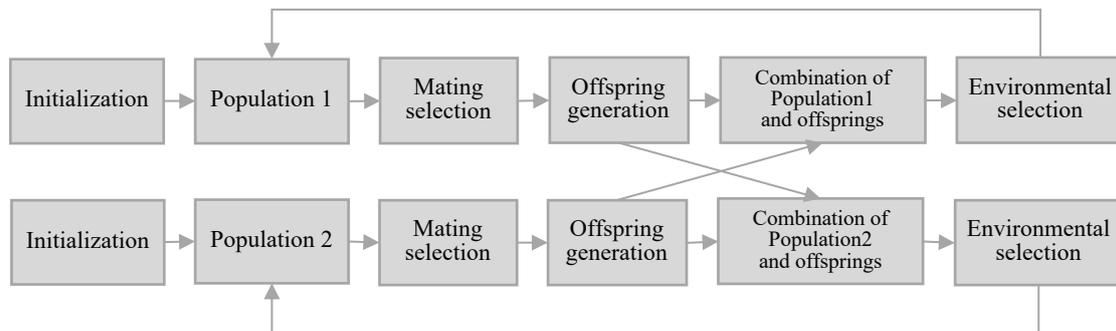


Figure 1. The procedure of DP-NSGA-III.

#### 3.1. Constraint Handling

A CMOP can be mathematically defined as:

$$\begin{aligned}
 \min F(x) &= (f_1(x), f_2(x), \dots, f_M(x)) \\
 \text{s.t. } g(x) &= (g_1(x), \dots, g_k(x)) \leq 0 \\
 \text{where } x &= (x_1, \dots, x_D) \in \Omega
 \end{aligned}
 \tag{1}$$

where  $x$  is a decision vector in decision space  $\Omega$ .  $F(x)$  is the objective vector that consists of  $M$  conflicting objective functions in objective space.  $g_j(x)$  is the  $j$ -th inequality constraint. A solution  $x$  is called a feasible solution in case  $g(x) \leq 0$ ; otherwise, it is an infeasible solution. In particular, when  $M > 3$ , the problem is called a constrained many-objective optimization problem (CMAOP), and the objective space is said to be high-dimensional.

In this article, the constraint violation is defined as the normalized violation of  $x$  on all constraints:

$$cv(x) = \frac{1}{q} \sum_{i=1}^q \frac{G_i(x)}{\max\{\max_{x \in P}\{G_i(x)\}, 1\}} \quad (2)$$

where  $P$  denotes the initial population, and  $G_i(x)$  represents the degree of the constraint violation of  $x$  on the  $i$ -th constraint:

$$G_i(x) = \max\{g_j(x), 0\}, i = 1, 2, \dots, q \quad (3)$$

To avoid a population excessively weighted towards feasibility and to be able to involve infeasible solutions with excellent objective values in evolution, a parameter-free  $\varepsilon$ -constraint boundary function to deal with constraints is proposed, with the detailed formulation given as follows:

$$\varepsilon(t) = \frac{cv_0(x)}{1 + e^{20(\frac{t}{MaxT} - 0.4)}}, t = 1, 2, \dots, MaxT \quad (4)$$

where  $cv_0(x)$  is the constraint violation of population initialization, and  $t$  is the number of iterations. The constraint boundary is a monotonically decreasing function with an initial value approximating  $cv_0(x)$ . As the population iterates, the constraint boundary value approaches 0. Therefore, at the end of evolution, this  $\varepsilon$ -constraint handling method is consistent with Deb's CDP, i.e., feasible solutions are preferred over infeasible solutions.

Based on the above designed  $\varepsilon(t)$ , the evolutionary algorithm solves the problem as shown below:

$$\begin{aligned} \min f(x) &= (f_1(x), f_2(x), \dots, f_M(x)) \\ \text{s.t. } g(x) &= (g_1(x), \dots, g_k(x)) \leq \varepsilon(t) \\ \text{where } \varepsilon(t) &\rightarrow 0 \end{aligned} \quad (5)$$

The problem is equivalent to the original CMAOPs with relaxed constraints in the early stage. In the late stage, since  $g(x) \rightarrow 0$ , it is equivalent to the original problem.

Based on the proposed  $\varepsilon$ -constraint handling method, the proposed evolutionary algorithm will solve Equation (1) indirectly. The main population solves the problem represented by Equation (5), and the auxiliary population solves the unconstrained problem corresponding to Equation (1).

### 3.2. Mating Selection and Offspring Reproduction

Population1 uses the  $\varepsilon$ -constraint handling method to deal with constraints. To make the offspring satisfy the  $\varepsilon$ -constraint boundary to the greatest extent, Population1 prefers an  $\varepsilon$ -feasible solution to an  $\varepsilon$ -infeasible solution, or a solution with a small constraint violation to a solution with a large constraint violation as the parent and puts it into the mating pool1. Algorithm 1 describes the above tournament selection procedure in detail.

Population2 does not consider constraints, so the nondominated solution is preferred as the parent and puts it into the mating pool2. If the two solutions do not dominate each other, then the selection is made by survival score [13].

Once the mating pool is constructed, the offspring of the population of size  $N$  is created using simulated binary crossover (SBX) [23] and polynomial mutation (PM) [24] operators. Finally, coevolution is achieved by merging Population1 (or Population2) with offspring 1 and offspring 2.

**Algorithm 1:** Tournament selection of Population1

---

```

input : candidate solutions  $x_1$  and  $x_2$ 
output: mating parent  $p$ 
1 if one and only one of  $x_1$  and  $x_2$  is  $\varepsilon$ -feasible then
2   | Select the  $\varepsilon$ -feasible solution as  $p$ 
3 else if  $x_1$  is  $\varepsilon$ -infeasible and  $x_2$  is  $\varepsilon$ -infeasible then
4   | if  $cv(x_1) \neq cv(x_2)$  then
5     | select  $x_1$  or  $x_2$  with smaller  $cv$  as  $p$ 
6   | else
7     |  $p \leftarrow \text{random}(x_1, x_2)$ 
8   | end
9 else
10  |  $p \leftarrow \text{random}(x_1, x_2)$ 
11 end

```

---

**3.3. Environmental Selection of Population1**

The environment selection of Population1 is based on NSGA-III combined with the  $\varepsilon$ -constraint handling method. This stage will select  $N$  outstanding individuals from Population1, offspring1, and offspring2 as new Population1.

The simplest case is that the number of  $\varepsilon$ -feasible solutions  $N_f$  is less than  $N$ . Then, the new population consists of all  $\varepsilon$ -feasible solutions and  $N - N_f$  solutions with the smallest constraint violation.

Otherwise,  $N_f$  is more than  $N$ . The reference-point-based nondominated sorting with two steps is performed. First, different nondominated levels are classified by using the nondominated sorting based on  $M + 1$  objectives:  $(f_1(x), f_2(x), \dots, f_M(x), cv(x))$ . Note that the constraint violation is treated as a new objective in this process. Then, Population1 is distinguished into  $F_1, F_2, \dots, F_k, \dots, F_n$  and satisfies  $|F_1 \cup F_2 \cup \dots \cup F_{k-1}| \leq N$ ,  $|F_1 \cup F_2 \cup \dots \cup F_k| > N$ . Second, reference-point-based elite selection is used to maintain population diversity.  $N - |F_1 \cup F_2 \cup \dots \cup F_{k-1}|$  solutions are selected in  $F_k$  based on the association between solutions and the predefined reference points (see Algorithm 2).

It is worth noting that, since the environmental selection of Population1 tends to  $\varepsilon$ -feasible solutions, the excellent infeasible solutions in offspring2 also have the possibility of entering the next generation. That is, the  $\varepsilon$ -constraint handling method provides the possibility of retaining the excellent offspring in coevolution.

**Algorithm 2:** Reference-point-based nondominated sorting

---

```

input :  $\varepsilon$ -feasible set  $S$ , reference points  $Z$ 
output: Population1  $P_{t+1}$ 
1 Performing nondominated sorting on  $S$  based on  $(f_1(x), f_2(x), \dots, f_M(x), cv(x))$  to form the
  nondomination levels  $F_1, F_2, \dots, F_k, \dots, F_n$ 
2 Let  $P_{t+1} = F_1 \cup F_2 \cup \dots \cup F_{k-1}$ 
3 while  $|P_{t+1}| < N$  do
4   | Normalize  $(f_1(x), f_2(x), \dots, f_M(x))$ 
5   | Association Operation:
6   | Associate each solution of  $P_{t+1}$  with the nearest reference point. The number of solutions
  associated with each reference point  $r$  is called niche count ( $r$ ).
7   |  $p \leftarrow \text{argmin niche count}(r)$ 
8   | Niche-Preservation Operation:
9   | Let  $T_p \subseteq F_k$  be a set of solutions associated with  $p$ 
10  | if  $|T_p| = 0$  then
11    |  $Z = Z \setminus p$ 
12  | else
13    | if  $\text{nichecount}(p) = 0$  then
14      | Select the solution  $x$  which has the smallest distance to  $p$ 
15    | else
16      | Select a solution  $x$  randomly from  $T_p$ 
17    | end
18  | end
19   $P_{t+1} = P_{t+1} \cup \{x\}$ ,  $F_k = F_k \setminus x$ 
20   $\text{nichecount}(p)++$ 
21 end

```

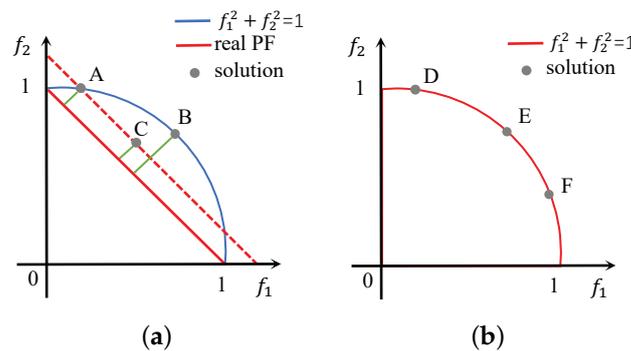
---

### 3.4. Environmental Selection of Population2

Population2 does not consider the constraint and eventually aims to converge to the UPF. Its environment selection [13] evaluates the solutions by  $p$ -norm, which can well distinguish the solutions in the high-dimensional objective space. This stage will select  $N$  outstanding individuals from Population2, offspring1, and offspring2 as new Population2. First, the nondominated levels are still divided by nondominated sorting. The following steps will be completely different from Population1. In each generation, Population2 is assigned fitness according to a fitness function called survival score. The survival score is composed of the ratio of the diversity score and the convergence score.

#### 3.4.1. The $p$ -Norm of the First Front $F_1$

The distance of the solution to the real PF reflects the convergence. However, the real PF is unknown, so the convergence is usually expressed as the Euclidean distance from the solution to the ideal point  $Z^{\min} = (0, 0)$ , i.e., the 2-norm of the solution. In Figure 2a, B is located at  $f_1 + f_2 = 1$ , C is located at  $f_1 + f_2 = 1.2$ , and A is located at the intersection of the red dashed line and the blue line. According to the Euclidean distance, since points A and B lie on the same circle,  $1 = \|A\|_2 = \|B\|_2 > \|C\|_2$ . However, the green line shows that A and C have the same distance to the real PF (assuming we know it), while B is the farthest. Therefore, in this case, 2-norm does not correctly represent the distance from the solution to the real PF. However, 1-norm can derive the correct distance relationship because A and C lie on 1-norm's contour  $f_1^1 + f_2^1 = 1.2$  and B is on the outside of the contour, i.e.,  $1.2 = \|A\|_1 = \|C\|_1 < \|B\|_1$ .



**Figure 2.** Solutions on different contours. (a) The real PF is the contour of 1-norm. (b)  $F_1$  is the contour of 2-norm.

Therefore, a suitable  $p$ -norm is chosen to measure the convergence of the solution. Since the real PF is unknown, assume that the shape of the first front  $F_1$  is the same as real PF and use the  $p$ -norm of  $F_1$  to replace the  $p$ -norm of the real PF. A method to find the  $p$ -norm of  $F_1$  is given below.

After normalizing the populations, the solution  $x = (x_1, \dots, x_D)$  in the  $F_1$  can be regarded as on the unitary hypersurface, satisfying the unique contour equation  $(x_1^p + \dots + x_D^p)^{1/p} = 1$ . This equation is equivalent to finding the zeros of  $h(p) = \ln(x_1^p + \dots + x_D^p)$ . The function is second order derivable and can be solved by Halley's method [25] as follows:

$$p_{n+1} = p_n - \frac{h(p_n)}{h'(p_n)} \left( 1 - \frac{h(p_n)h''(p_n)}{2(h'(p_n))^2} \right)^{-1}, k = 0, 1, 2, \dots \tag{6}$$

The  $p$ -norm of  $F_1$  can be solved by substituting any solution in  $F_1$  into Equation (6). Figure 2b was used to verify the correctness of Equation (6), which is the case of  $p = 2$ .  $D(\cos \frac{7\pi}{16}, \sin \frac{7\pi}{16})$ ,  $E(\cos \frac{\pi}{4}, \sin \frac{\pi}{4})$  and  $F(\cos \frac{\pi}{8}, \sin \frac{\pi}{8})$  are the normalized solutions in  $F_1$ . Let the initial value  $p_0 = 1$ , then substitute D, E or F into Equation (6) to obtain  $p = p_2 \approx 2$

in only two iterations.  $p \approx 2$  is consistent with the actual situation, since the three solutions lie on the contour of the 2-norm.

The computational complexity of Halley’s method is  $O(MT)$ , where  $T$  is the number of iterations, generally around 3, much smaller than the size of the population  $N$ . Therefore, there is no impact on the time complexity of the algorithm.

### 3.4.2. Survival Score

Once the  $p$ -norm of the first nondominated front  $F_1$  is calculated using Equation (6), we can measure both the convergence and diversity of  $F_1$  [13] accordingly:

$$\text{convergence}(x) = \left\| f^n(x) - Z^{\min} \right\|_p = \|f^n(x)\|_p \tag{7}$$

$$\text{diversity}(x, F_1) = \min_{y \neq x \in F_1} \|f^n(x) - f(y)\|_p \tag{8}$$

The convergence score of an individual is the Minkowski distance of the individual to the ideal point  $Z^{\min} = (0, 0)$ . Since the ideal point is the origin, the convergence is the  $p$ -norm of the individual. The diversity score of an individual is computed as the minimum Minkowski distance with the other solutions in the front  $F_1$ . The survival score of each solution  $x \in F_1$  consists of the ratio of diversity to convergence as follows:

$$\text{survival score}(x) = \frac{\text{diversity}(x, F_1)}{\text{convergence}(x)} \tag{9}$$

The procedure for calculating the survival score of the solution in  $F_1$  is shown in Algorithm 3. Note that, if the solution  $x \notin F_1$ , then its survival score is the reciprocal of the convergence score.

---

#### Algorithm 3: Survival Score of $F_1$

---

```

input :  $p$ : exponent of estimated geometry the  $p$ -norm, Population2 -  $P_2$ , offspring1 and offspring2
output: Population2 -  $P_2$ 
1 score[E]  $\leftarrow +\infty$ ; // E = extreme points
2  $\Phi \leftarrow E$ ; // Considered solutions
3  $\bar{\Phi} \leftarrow F_1 \setminus \Phi$ ; // Remaining solutions
4 for each solution  $x \in \bar{\Phi}$  do
5 | convergence( $x$ )  $\leftarrow \|f^n(x)\|_p$ 
6 end
7 for each solution  $x_1 \in F_1$  do
8 | for each solution  $x_2 \in F_1$  do
9 | | dist( $x_1, x_2$ )  $\leftarrow \|f(x_1) - f(x_2)\|_p$ 
10 | end
11 end
12 while  $|\bar{\Phi}| > 0$  do
13 | for each solution  $x \in \Phi$  do
14 | | diversity( $x$ )  $\leftarrow \min_{y \in \Phi} \text{dist}(x, y) + \min_{y \in \Phi} 2\text{dist}(x, y)$ ; // min2: the second minimum
15 | | | distances
16 | | | survival score( $x$ ) = diversity( $x$ )/convergence( $x$ )
17 | end
18 |  $x^* \leftarrow \arg \min_{x \in \Phi} \text{value}(x)$ 
19 | score( $x^*$ )  $\leftarrow \text{value}(x^*)$ 
20 |  $\Phi \leftarrow \Phi \cup \{x^*\}$ ; // Considered solutions
21 |  $\bar{\Phi} \leftarrow \bar{\Phi} \setminus \{x^*\}$ ; // Remaining solutions
22 end

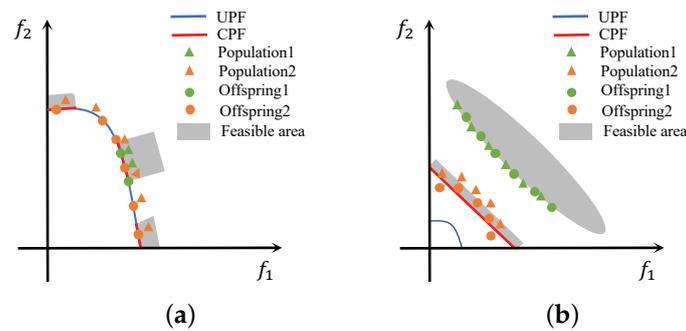
```

---

### 3.5. Discussion

Population2 has a different population distribution than Population1 because it ignores the constraint. On the one hand, Population2 can provide offspring with good diversity. In Figure 3a, Population1 is trapped in a local optimum, and its Offspring1 have difficulty finding other feasible areas. Population2 is widely distributed around the

UPF due to ignoring constraints. Through the coevolution of offspring combinations, the better-diversified Offspring2 can lead Population1 to explore other feasible regions. On the other hand, Population2 can provide offspring with good convergence. In Figure 3b, Population1 falls into a local optimum away from the CPF and Population2, which does not consider constraints and explores towards the UPF. The Offspring2 near the CPF can pull Population1 near the CPF.



**Figure 3.** Role of the Population2. (a) Offspring2 with good diversity; (b) Offspring2 with good convergence.

### 3.6. Time Complexity of DP-NSGA-III

The proposed  $\varepsilon$ -constraint handling method has no effect on the time complexity of NSGA-III, so the time complexity of Population1 is still  $O(MN^2)$  [9,26].

The time complexity of Population2 is influenced by nondominated sorting and the calculation of survival score. The time complexity of nondominated sorting is the same as Population1 with  $O(MN^2)$  [26]. The computational complexity of calculation of survival score of  $F_1$  (Algorithm 3) is  $O(MN^2) + O(N^3)$ —specifically,  $O(MN^2)$  for the loop in lines 7–10, and  $O(N^3)$  for the loop in lines 12–21. The survival score of the other nondominated levels is calculated consistently with lines 4–6 as  $O(MN)$ .

Therefore, the time complexity of DP-NSGA-III is  $O(MN^2) + O(N^3)$ .

## 4. Experimental Setup

### 4.1. Test Instances

In this article, C-DTLZ [20], DC-DTLZ [11], and MW [27] suites have been selected for examining the performance of DP-NSGA-III. The C-DTLZ suite is based on the DTLZ suite, and four constraints are added to the objective space, while the DC-DTLZ suite introduces three types of constraints in the decision space. The DC-DTLZ suite involves many local optima in infeasible regions, which causes the algorithm may easily fall into some infeasible local optima and cannot enter the feasible region. The objective number of the C-DTLZ suite and the DC-DTLZ suite is scalable. The MW suite was recently proposed, and it contains a variety of features, such as a small feasibility area, multiple complex nonlinear constraints, a scalable number of objectives, and high-dimensional decision vectors. Since some of the test functions in the MW suite have a fixed objective number, in this section, we only test MW4, MW8, and MW14 with variable objective space dimensions.

According to the relationship between UPF and CPF, the above test problems can be divided into three categories [20]:

Type-I: The CPF is the same as the UPF. MW4, MW14, C1DTLZ1, C1DTLZ3, DC2DTLZ1, and DC2DTLZ3 are Type-I problems.

Type-II: Infeasible area makes the UPF partly feasible and the CPF is a part of the UPF. MW8, C2DTLZ2, DC1DTLZ1, and DC1DTLZ3 are Type-II problems.

Type-III: The UPF is completely infeasible and the CPF is located on the boundary of the feasible region. C3DTLZ4, DC3DTLZ1, and DC3DTLZ3 are Type-III problems.

#### 4.2. Performance Metrics

To measure the performance of DP-NSGA-III and other peer CMAOEs, the following two commonly used metrics are adopted. Furthermore, the Wilcoxon rank sum test with a significance level of 0.05 is adopted to perform statistical analysis.

- (1) Inverted Generational Distance (IGD) [28]: Given  $P^*$  as a set of points uniformly sampled along the PF and  $P$  as the set of solutions obtained from an evolutionary algorithm. The IGD value of  $P$  is calculated as

$$IGD(P, P^*) = \frac{\sum_{x \in P^*} \text{dist}(x, P)}{|P^*|} \tag{10}$$

where  $\text{dist}(x, P)$  is the Euclidean distance between  $x$  and its nearest neighbor in  $P$ .

- (2) Hypervolume (HV) [29]: Let  $z^r = (z_1^r, \dots, z_m^r)^T$  be the worst point dominated by all the Pareto optimal objective vectors. The HV of  $P$  is defined as the volume of the objective space dominated by solutions in  $P$  and bounded by  $z^r$

$$HV(P) = \text{VOL} \left( \bigcup_{z \in P} [z_1, z_1^r] \times \dots \times [z_m, z_m^r] \right) \tag{11}$$

#### 4.3. Algorithms for Comparison

Four state-of-the-art and representative CMAOEs were chosen for performance comparison: C-NSGA-III [20] (Abbreviated as NSGA-III), C-TAEA [11], DCNSGA-III [30], and MOEA/D-2WA [31]. NSGA-III and C-TAEA have been briefly introduced in Section 2. Thus, we briefly describe others' basic ideas as follows:

DCNSGA-III: It is an improved NSGA-III that uses dynamic multi-objective techniques to solve CMAOPs. It is more suitable for processing CMAOPs than NSGA-III due to the upgrade of the constraint processing method.

MOEA/D-2WA: It is a modified MOEA/D using two-type weight adjustments. During the search, the number of infeasible weights is dynamically lowered to help infeasible solutions with greater convergence transcend the infeasible areas, as well as to help infeasible solutions with better diversity identify many feasible sub regions.

#### 4.4. Parameter Settings

- (1) Algorithms and reproduction operators: To make a fair comparison and ensure the performance of each algorithm, the parameters of all the compared algorithm and their reproduction operator parameters, such as the simulated binary crossover and polynomial mutation, are set as suggested in their original papers. All the compared algorithms were implemented in PlatEMO [32].
- (2) Population size and the number of function evaluations: Table 1 shows the number of reference points and population size of each algorithm for different numbers of objectives  $M$  [11,27]. The number of function evaluations is the termination condition of all algorithms [11,27]. The details are shown in Table 2.

**Table 1.** The number of reference points and corresponding population size for different numbers of objectives.

Objectives/Dimension No. $M$	Reference No.	Popsize $N$
3	91	92
5	210	212
10	275	276
15	135	136

**Table 2.** The number of maximum function evaluations for different problems.

Problem	$M = 3$	$M = 5$	$M = 10$	$M = 15$
C1DTLZ1	46,000	127,200	276,000	204,000
C1DTLZ3	92,000	318,000	966,000	680,000
C2DTLZ2	23,000	74,200	207,000	136,000
C3DTLZ4	69,000	265,000	828,000	544,000
DC1DTLZ1	69,000	265,000	828,000	544,000
DC1DTLZ3	69,000	265,000	828,000	544,000
DC2DTLZ1	138,000	530,000	1656,000	1088,000
DC2DTLZ3	138,000	530,000	1656,000	1088,000
DC3DTLZ1	69,000	265,000	828,000	544,000
DC3DTLZ3	69,000	265,000	828,000	544,000
MW4	55,200	127,200	165,600	81,600
MW8	55,200	127,200	165,600	81,600
MW14	55,200	127,200	165,600	81,600

## 5. Experimental Results

This section focuses on analyzing the performance of the proposed DP-NSGA-III with the peer CMAOEA on three test suites. Note that the following population distribution plots or Parallel coordinate plots (PCPs) are taken for the median IGD of 30 independent experiments.

### 5.1. Performance on C-DTLZ Suite

Wilcoxon's test results on C-DTLZ suite are summarized in Table 3. According to the IGD values, the results show that DP-NSGA-III is significantly better than its competitors on at least 7/16 and at most 14/16 test problems. The results indicate that DP-NSGA-III performs much better than its rivals on at least 8/16 and at most 12/16 test problems, based on the HV values.

**Table 3.** Wilcoxon's test results on C-DTLZ suite <sup>1</sup>.

	IGD (−/+/=)	HV (−/+/=)
DP-NSGA-III vs. NSGA-III	2/13/1	4/11/1
DP-NSGA-III vs. C-TAEA	2/13/1	3/11/2
DP-NSGA-III vs. DCNSGA-III	0/14/2	2/12/2
DP-NSGA-III vs. MOEA/D-2WA	6/7/3	6/8/2

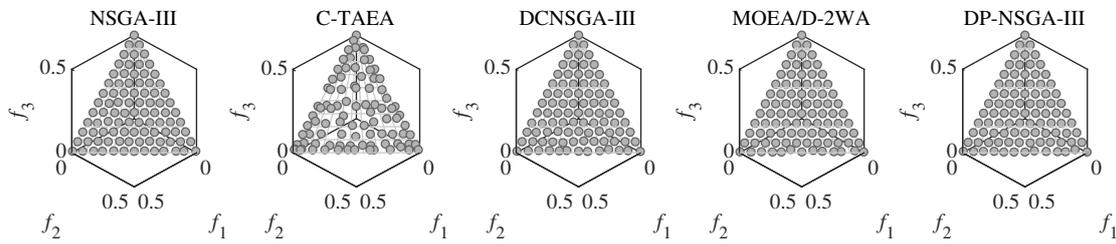
<sup>1</sup> "+", "−", or "=" denote the number of the performance of DP-NSGA-III is significantly better than, worse than, or comparable to its peer, respectively.

Tables A1 and A2 show the mean value and standard deviation of the IGD and HV values obtained from the above-mentioned five algorithms over 30 runs on the C-DTLZ suite with 3, 5, 10, and 15 objectives. The C-DTLZ suite is moderately difficult, and the proposed algorithm achieves good results on all but C3DTLZ4.

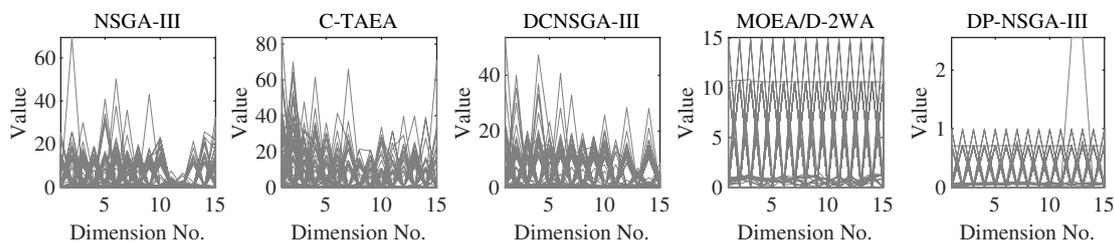
For C1DTLZ1, only a narrow section near PF is feasible. Both the DP-NSGA-III and comparison algorithms converge on this problem. Figure 4 shows that differences in performance metric are mainly influenced by the uniformity of population distribution. C-TAEA has a messy population distribution and therefore has the worst performance metric.

The C1DTLZ3 problem has a highly multimodal landscape and infeasible region barriers that pose a challenge to the convergence of CMAOEA. Populations have difficulty crossing infeasible barriers and are easily trapped in localized feasible areas. The IGD of the proposed DP-NSGA-III is better than other peer algorithms by more than an order of magnitude in all four dimensions. The global optimal value of C1DTLZ3 is between 0 and 1. Figure 5 indicates that DP-NSGA-III converges to the global optimum except for a very small number of solutions. The other algorithms only converge to the boundary of the local feasible region and fall into the local optimum with a large number of solutions with

objective values greater than 1. For DP-NAGA-III, the auxiliary populations that ignore the constraints help the main population to cross a huge infeasibility region.



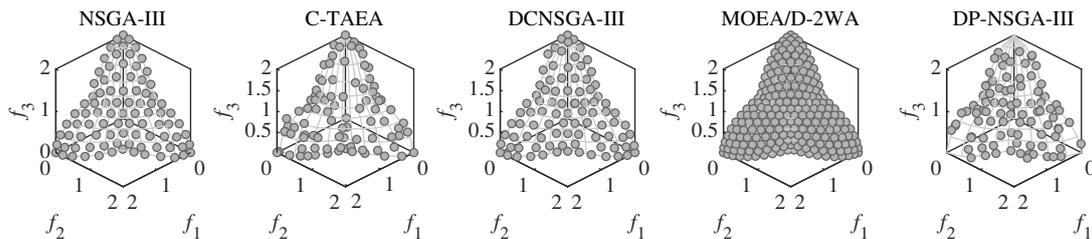
**Figure 4.** Scatter plots of the population obtained by the peer algorithms and DP-NSGA-III on 3-objective C1DTLZ1 (median IGD value). The grey area is the real Pareto front.



**Figure 5.** PCPs of the populations obtained by the peer algorithms and DP-NSGA-III on 15-objective C1DTLZ3 (median IGD value).

The CPF of C2DTLZ2 is part of the UPF, and the areas are not connected. This problem mainly tests the diversity maintenance ability of the algorithm. Most evolutionary algorithms found the CPF, and DP-NSGA-III performed relatively well.

The CPF of C3DLTZ4 is far from the UPF and is the boundary of the feasible area. In this issue, DP-NSGA-III was able to find CPF, but the population diversity was poor. Figure 6 shows the most uniform population distribution obtained by MOEA/D-2WA with two-type weight.



**Figure 6.** Scatter plots of the population obtained by the peer algorithms and DP-NSGA-III on 3-objective C3DTLZ4 (median IGD value). The grey area is the real Pareto front.

5.2. Performance on DC-DTLZ Suite

Wilcoxon’s test results on DC-DTLZ suite are summarized in Table 4. According to the IGD values, the results show that DP-NSGA-III is significantly better than its competitors on at least 14/22 and at most 20/22 test problems. The results indicate that DP-NSGA-III performs much better than its rivals on at least 12/22 and at most 21/22 test problems, based on the HV values.

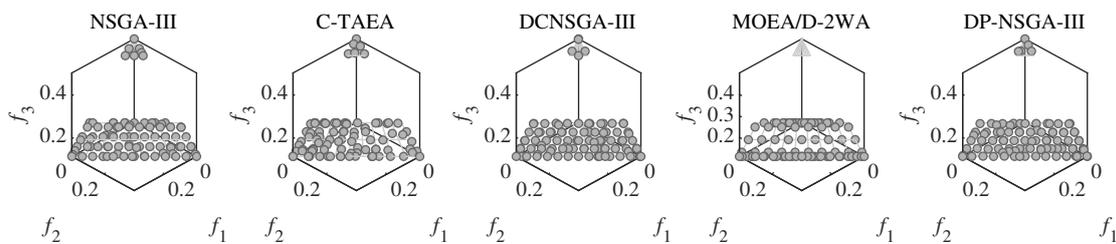
**Table 4.** Wilcoxon’s test results on DC-DTLZ suite <sup>1</sup>.

	IGD (–/+/=)	HV (–/+/=)
DP-NSGA-III vs. NSGA-III	1/20/1	2/18/2
DP-NSGA-III vs. C-TAEA	0/19/3	1/21/0
DP-NSGA-III vs. DCNSGA-III	0/19/3	0/17/5
DP-NSGA-III vs. MOEA/D-2WA	4/14/4	2/12/8

<sup>1</sup> “+”, “–”, or “=” denote the number of the performance of DP-NSGA-III is significantly better than, worse than, or comparable to its peer, respectively.

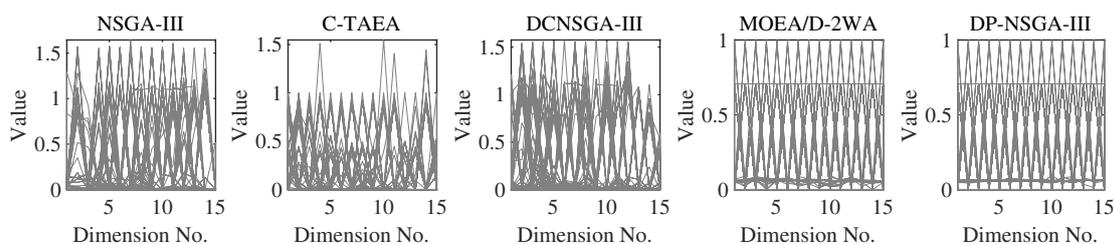
Tables A3 and A4 show the mean value and standard deviation of the IGD and HV values obtained from the above-mentioned five algorithms over 30 runs on the DC-DTLZ suite with 3, 5, 10, and 15 objectives.

DC1DTLZ1 and DC1DTLZ3 are the Type-II problems. The CPF consists of multiple disjoint feasible areas. The obtained population is easily trapped in the local optimal region, and it is difficult to find the complete PF. Performance metrics show that DP-NSGA-III has a greater advantage over DC1DTLZ3 than DC1DTLZ1. Figure 7 illustrates that the population of DP-NSGA-III is evenly distributed on two separated PFs of DC1DTLZ1, while MOEA/D-2WA fails to find the PF above.



**Figure 7.** Scatter plots of the population obtained by the peer algorithms and DP-NSGA-III on 3-objective DC1DTLZ1 (median IGD value). The grey area is the real Pareto front.

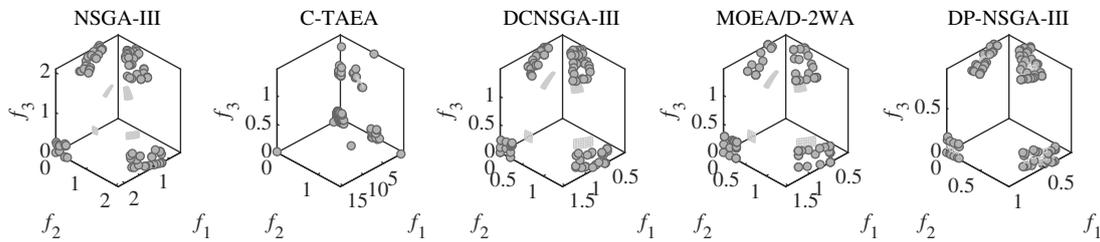
DC2DTLZ1 and DC2DTLZ3 belong to the Type-I problems. Although it has the same CPF as UPF, the infeasibility region is huge and only the region near PF is feasible. Moreover, there exist many local optima in infeasible regions, making it much harder for solutions to escape. DP-NSGA-III achieved the best results in all four dimensions of DC2DTLZ1. Figure 8 demonstrates that DP-NSGA-III and MOEA/D-2WA perform best on DC2DTLZ3; in particular, the convergence and diversity of DP-NSGA-III are improved over NSGA-III and DCNSGA-III. NSGA-III and DCNSGA-III have objective values greater than 1 in multiple dimensions, indicating that there are solutions that fail to converge to the global optimum. Ignoring the few solutions with objective values greater than 1, the polyline in parallel coordinates for C-TAEA was not uniformly distributed compared to DP-NSGA-III, indicating an uneven distribution of the population.



**Figure 8.** PCPs of the populations obtained by the peer algorithms and DP-NSGA-III on 15-objective DC2DTLZ3 (median IGD value).

DC3DTLZ1 and DC3DTLZ3 can be generated as Type-III problems, which are a mixture of the first two kinds of test instances. As the number of objectives increases, the

feasible region becomes smaller and smaller. Note that no feasible solutions are found for all algorithms on the DC3DTLZ1 problems when  $M \geq 10$ . Figure 9 shows that, except for DP-NSGA-III, all the algorithms fail to jump out of the infeasible region on DC3DTLZ3. DP-NSGA-III has the best convergence and distribution.



**Figure 9.** Scatter plots of the population obtained by the peer algorithms and DP-NSGA-III on 3-objective DC3DTLZ3 (median IGD value). The grey area is the real Pareto front.

5.3. Performance on MW Suite

Wilcoxon’s test results on MW suite are summarized in Table 5. According to the IGD values, the results show that DP-NSGA-III is significantly better than its competitors on at least 6/12 and at most 10/12 test problems. The results indicate that DP-NSGA-III performs much better than its rivals on at least 6/12 and at most 10/12 test problems, based on the HV values.

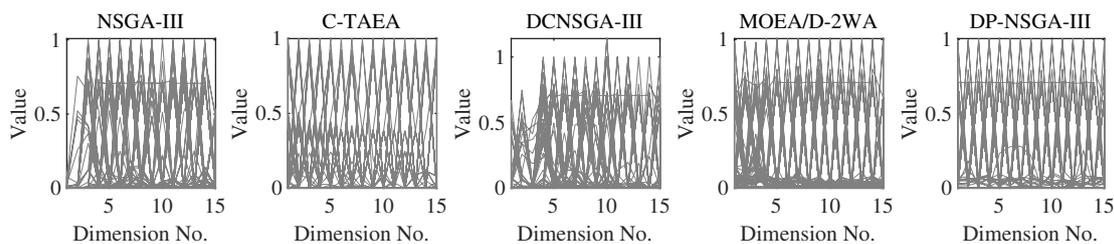
**Table 5.** Wilcoxon’s test results on MW suite <sup>1</sup>.

	IGD (-/+/=)	HV (-/+/=)
DP-NSGA-III vs. NSGA-III	3/8/1	2/6/4
DP-NSGA-III vs. C-TAEA	2/10/0	2/10/0
DP-NSGA-III vs. DCNSGA-III	1/9/2	2/7/3
DP-NSGA-III vs. MOEA/D-2WA	6/6/0	1/7/4

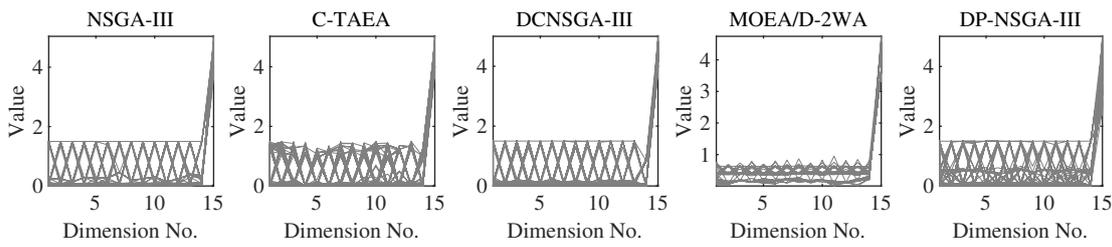
<sup>1</sup> “+”, “-”, or “=” denote the number of the performance of DP-NSGA-III is significantly better than, worse than, or comparable to its peer, respectively.

Tables A5 and A6 show the mean value and standard deviation of the IGD and HV values obtained from the above-mentioned five algorithms over 30 runs on the MW suite with 3, 5, 10, and 15 objectives.

MW4, MW8, and MW14 are the problems with the adjustable number of objectives in this suite. From Figure 10, we can see that DP-NSGA-III and MOEA/D-2WA perform comparably and better than other algorithms on the MW8 problem. NSGA-III and DCNSGA-III do not cover areas with a value of 1 in the first few dimensions. Different objectives of the MW14 have different ranges of value domains, with a maximum value of 5 when  $M = 15$ . Figure 11 shows that MOEA/D-2WA, which relies entirely on the reference vector, has degraded performance in this problem. When  $M \leq 14$ , the population obtained by MOEA/D-2WA fails to find areas with an objective value of 1, while DP-NSGA-III still performs well in higher dimensions.



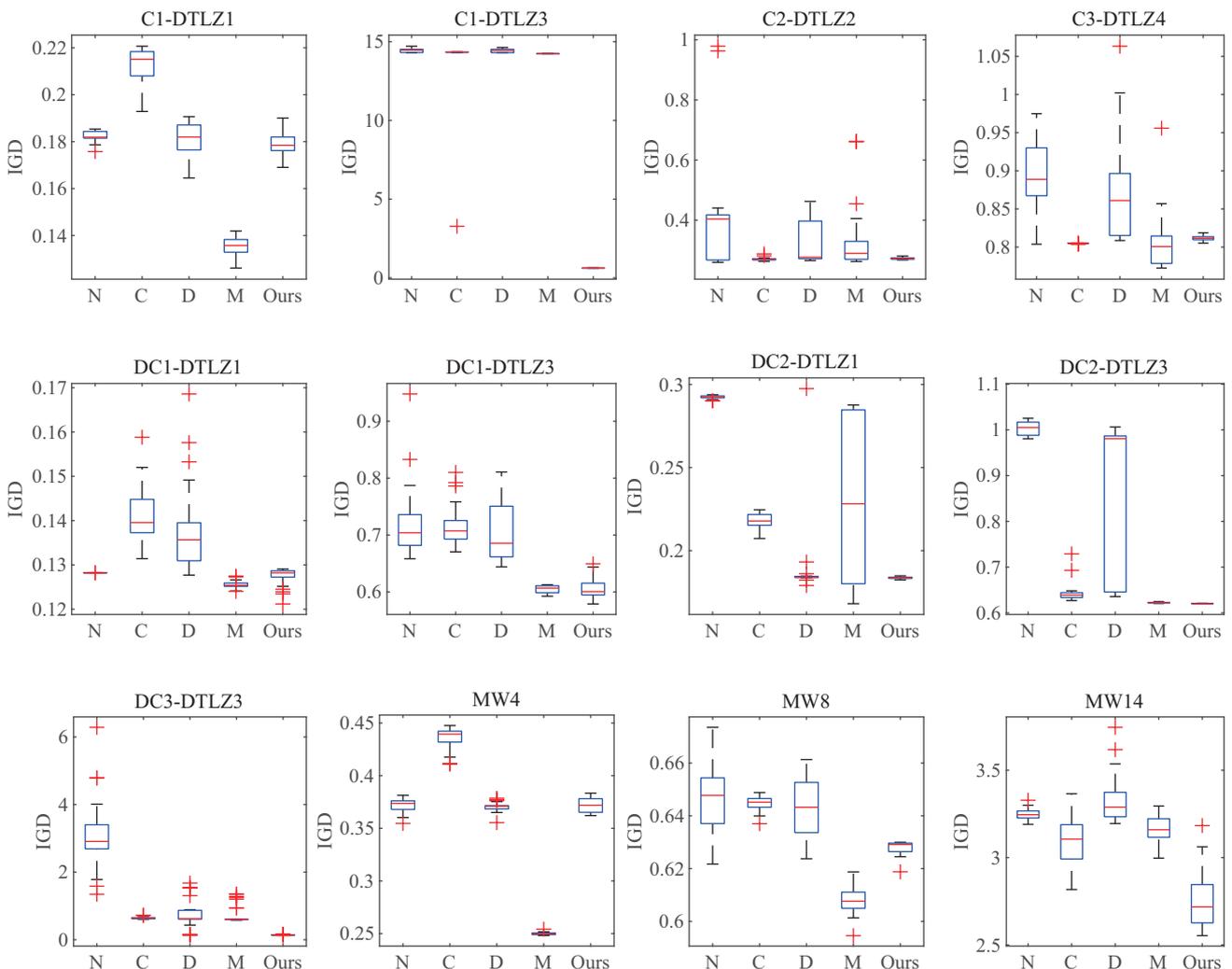
**Figure 10.** PCPs of the populations obtained by the peer algorithms and DP-NSGA-III on 15-objective MW8 (median IGD value).



**Figure 11.** PCPs of the populations obtained by the peer algorithms and DP-NSGA-III on 15-objective MW14 (median IGD value).

5.4. Stability of DP-NSGA-III

To evaluate the stability of each algorithm, box plots are used to represent the statistical results of IGD metrics for 30 experiments of the algorithms on each problem ( $M = 15$ ). For convenience, each algorithm is abbreviated with its initials. Each subplot corresponds to the case of  $M = 15$  in Tables A1, A3, and A5. Figure 12 shows that, while the other algorithms have more outliers and larger deviations, DP-NSGA-III has only a few outliers and small deviations on individual problems. DP-NSGA-III has the smallest inter-quartile range and the best stability for most problems. DP-NSGA-III has obvious advantages in median, inter-quartile range and outliers, reflecting that DP-NSGA-III can better balance convergence and diversity on 15-dimensional CMAOPs and has better algorithm stability.



**Figure 12.** Box plots of IGD for DP-NAGA-III and other peer algorithms on the 15-objective problems.

## 6. Conclusions

This paper proposed a dual-population-based constrained many-objective evolutionary algorithm, named DP-NSGA-III. Compared with existing CMOEAs, DP-NSGA-III coevolves by exchanging the offspring of two populations. Two populations use different environmental selection to select superior individuals. Among them, the Auxiliary population deals with CMOOP that ignores constraints, and the main population focuses on solving the original CMOOP. The main population can make full use of the information of infeasible solutions in the auxiliary population to solve the constraint problem more effectively. From the problem perspective, the designed dual population solves the original CMOOPs indirectly. The main population solves the problem of loose constraints and the auxiliary population solves the problem of ignoring constraints. In addition, to improve the ability of the main population to handle constraints, this paper designed a parameter-free  $\epsilon$ -constraint handling method. In the main populations driven by infeasible solutions,  $\epsilon$ -feasible solutions can provide the impetus for evolution.

The performance of DP-NSGA-III has been studied for a series of suites with complex constraints with up to 15 objectives. Thanks to the proposed constraint handling methods that reduce the difficulty of handling constraints, the experimental results fully demonstrate the competitiveness of DP-NSGA-III in CMOOPs compared to the four state-of-the-art CMOEAs, especially in DC-DTLZ suite with highly complex constraints. However, if the CPF is not a regular surface, or if the UPF is separated from the CPF, DP-NSGA-III performs poorly. On C3DTLZ4 with both features, DP-NSGA-III converged to CPF, but the populations were not as well distributed as NSGA-III. The solutions provided by the auxiliary population have a side effect instead. In the future, it is necessary to further explore the underlying mechanisms of dual-population and design better dual-population strategies. In terms of environment selection, more efficient scalar methods need to be designed to replace Pareto dominance relationships that make it difficult to distinguish individuals in a high-dimensional objective space.

**Author Contributions:** Conceptualization, H.G.; methodology, H.G.; software, Z.Z.; writing—original draft preparation, Z.Z.; writing—review and editing, Z.Z.; visualization, J.S. and F.S.; funding acquisition, H.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China under Grant 51977100.

**Institutional Review Board Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

CMOP	Constrained multi-objective optimization problem
CMaOP	Constrained many-objective optimization problem
CMOEA	Constrained multi-objective evolutionary algorithm
CMaOEA	Constrained many-objective evolutionary algorithm
CDP	Constraint-dominance principle
PF	Pareto front
UPF	Unconstrained Pareto front
CPF	Constrained Pareto front
IGD	Inverted generational distance
HV	Hypervolume
PCP	Parallel coordinate plot

Appendix A  
Appendix A.1

Table A1. IGD values obtained by DP-NSGA-III and other peer algorithms in the C-DTLZ suite <sup>1</sup>.

Problem	M	NSGA-III	C-TAEA	DCNSGA-III	MOEA/D-2WA	DP-NSGA-III
C1DTLZ1	3	2.0316 × 10 <sup>-2</sup> (1.66 × 10 <sup>-4</sup> )=	2.3218 × 10 <sup>-2</sup> (5.64 × 10 <sup>-4</sup> )–	<b>2.0292</b> × 10 <sup>-2</sup> ( <b>1.01</b> × 10 <sup>-4</sup> )=	2.0347 × 10 <sup>-2</sup> (1.06 × 10 <sup>-4</sup> )=	2.0342 × 10 <sup>-2</sup> (9.94 × 10 <sup>-5</sup> )
	5	5.2160 × 10 <sup>-2</sup> (2.95 × 10 <sup>-4</sup> )–	5.9101 × 10 <sup>-2</sup> (8.92 × 10 <sup>-4</sup> )–	5.2108 × 10 <sup>-2</sup> (3.15 × 10 <sup>-4</sup> )–	5.1891 × 10 <sup>-2</sup> (2.70 × 10 <sup>-4</sup> )=	<b>5.1815</b> × 10 <sup>-2</sup> ( <b>2.52</b> × 10 <sup>-4</sup> )
	10	1.0934 × 10 <sup>-1</sup> (3.38 × 10 <sup>-3</sup> )–	1.3960 × 10 <sup>-1</sup> (2.37 × 10 <sup>-3</sup> )–	1.1167 × 10 <sup>-1</sup> (7.28 × 10 <sup>-3</sup> )–	<b>1.0487</b> × 10 <sup>-1</sup> ( <b>5.67</b> × 10 <sup>-4</sup> )+	1.0799 × 10 <sup>-1</sup> (5.86 × 10 <sup>-4</sup> )
	15	1.8281 × 10 <sup>-1</sup> (3.39 × 10 <sup>-3</sup> )–	2.1208 × 10 <sup>-1</sup> (7.95 × 10 <sup>-3</sup> )–	1.8062 × 10 <sup>-1</sup> (7.67 × 10 <sup>-3</sup> )=	<b>1.3518</b> × 10 <sup>-1</sup> ( <b>4.24</b> × 10 <sup>-3</sup> )+	1.7936 × 10 <sup>-1</sup> (4.43 × 10 <sup>-3</sup> )
C1DTLZ3	3	3.2850 × 10 <sup>0</sup> (3.93 × 10 <sup>0</sup> )–	7.3722 × 10 <sup>-2</sup> (1.25 × 10 <sup>-2</sup> )–	5.1328 × 10 <sup>0</sup> (3.85 × 10 <sup>0</sup> )–	4.3406 × 10 <sup>0</sup> (4.00 × 10 <sup>0</sup> )–	<b>5.4508</b> × 10 <sup>-2</sup> ( <b>2.67</b> × 10 <sup>-5</sup> )
	5	4.9508 × 10 <sup>0</sup> (5.52 × 10 <sup>0</sup> )–	8.9646 × 10 <sup>0</sup> (4.88 × 10 <sup>0</sup> )–	5.7600 × 10 <sup>0</sup> (5.56 × 10 <sup>0</sup> )–	6.2536 × 10 <sup>0</sup> (5.77 × 10 <sup>0</sup> )–	<b>1.6513</b> × 10 <sup>-1</sup> ( <b>5.32</b> × 10 <sup>-5</sup> )
	10	5.0854 × 10 <sup>0</sup> (6.58 × 10 <sup>0</sup> )–	1.1983 × 10 <sup>1</sup> (5.15 × 10 <sup>0</sup> )–	6.6092 × 10 <sup>0</sup> (6.80 × 10 <sup>0</sup> )–	1.1406 × 10 <sup>1</sup> (5.59 × 10 <sup>0</sup> )–	<b>4.2023</b> × 10 <sup>-1</sup> ( <b>4.00</b> × 10 <sup>-4</sup> )
	15	1.2705 × 10 <sup>1</sup> (4.54 × 10 <sup>0</sup> )–	1.1425 × 10 <sup>1</sup> (5.39 × 10 <sup>0</sup> )–	1.2603 × 10 <sup>1</sup> (4.75 × 10 <sup>0</sup> )–	1.3781 × 10 <sup>1</sup> (2.49 × 10 <sup>0</sup> )–	<b>6.2255</b> × 10 <sup>-1</sup> ( <b>9.07</b> × 10 <sup>-4</sup> )
C2DTLZ2	3	4.8008 × 10 <sup>-2</sup> (4.54 × 10 <sup>-4</sup> )–	5.7025 × 10 <sup>-2</sup> (1.50 × 10 <sup>-3</sup> )–	4.7951 × 10 <sup>-2</sup> (4.62 × 10 <sup>-4</sup> )–	4.9627 × 10 <sup>-2</sup> (4.03 × 10 <sup>-4</sup> )–	<b>4.7472</b> × 10 <sup>-2</sup> ( <b>4.66</b> × 10 <sup>-4</sup> )
	5	1.3891 × 10 <sup>-1</sup> (2.42 × 10 <sup>-4</sup> )–	1.4714 × 10 <sup>-1</sup> (1.25 × 10 <sup>-3</sup> )–	1.3881 × 10 <sup>-1</sup> (3.48 × 10 <sup>-4</sup> )–	1.3911 × 10 <sup>-1</sup> (6.53 × 10 <sup>-4</sup> )–	<b>1.3759</b> × 10 <sup>-1</sup> ( <b>6.92</b> × 10 <sup>-4</sup> )
	10	3.3032 × 10 <sup>-1</sup> (1.65 × 10 <sup>-1</sup> )–	3.4636 × 10 <sup>-1</sup> (3.60 × 10 <sup>-2</sup> )–	2.6499 × 10 <sup>-1</sup> (2.11 × 10 <sup>-3</sup> )–	2.8945 × 10 <sup>-1</sup> (6.53 × 10 <sup>-2</sup> )–	<b>2.5890</b> × 10 <sup>-1</sup> ( <b>2.36</b> × 10 <sup>-3</sup> )
	15	4.0334 × 10 <sup>-1</sup> (1.69 × 10 <sup>-1</sup> )–	<b>2.7026</b> × 10 <sup>-1</sup> ( <b>5.04</b> × 10 <sup>-3</sup> )+	3.3055 × 10 <sup>-1</sup> (7.04 × 10 <sup>-2</sup> )–	3.3590 × 10 <sup>-1</sup> (1.19 × 10 <sup>-1</sup> )–	2.7237 × 10 <sup>-1</sup> (2.85 × 10 <sup>-3</sup> )
C3DTLZ4	3	2.1730 × 10 <sup>-1</sup> (2.85 × 10 <sup>-1</sup> )–	1.1259 × 10 <sup>-1</sup> (2.48 × 10 <sup>-3</sup> )=	1.4665 × 10 <sup>-1</sup> (1.89 × 10 <sup>-1</sup> )–	<b>5.7976</b> × 10 <sup>-2</sup> ( <b>9.37</b> × 10 <sup>-5</sup> )+	1.4219 × 10 <sup>-1</sup> (4.48 × 10 <sup>-2</sup> )
	5	2.4579 × 10 <sup>-1</sup> (8.96 × 10 <sup>-4</sup> )+	2.9056 × 10 <sup>-1</sup> (2.02 × 10 <sup>-3</sup> )–	2.8371 × 10 <sup>-1</sup> (1.09 × 10 <sup>-1</sup> )–	<b>2.4406</b> × 10 <sup>-1</sup> ( <b>4.52</b> × 10 <sup>-4</sup> )+	2.5222 × 10 <sup>-1</sup> (4.96 × 10 <sup>-3</sup> )
	10	5.7797 × 10 <sup>-1</sup> (4.66 × 10 <sup>-2</sup> )+	6.2424 × 10 <sup>-1</sup> (4.86 × 10 <sup>-3</sup> )–	7.0682 × 10 <sup>-1</sup> (6.66 × 10 <sup>-2</sup> )–	<b>5.6050</b> × 10 <sup>-1</sup> ( <b>1.92</b> × 10 <sup>-3</sup> )+	6.0233 × 10 <sup>-1</sup> (4.34 × 10 <sup>-2</sup> )
	15	8.8829 × 10 <sup>-1</sup> (5.01 × 10 <sup>-2</sup> )–	<b>8.0495</b> × 10 <sup>-1</sup> ( <b>3.51</b> × 10 <sup>-4</sup> )+	8.7126 × 10 <sup>-1</sup> (6.00 × 10 <sup>-2</sup> )–	8.0516 × 10 <sup>-1</sup> (3.61 × 10 <sup>-2</sup> )+	8.2850 × 10 <sup>-1</sup> (6.02 × 10 <sup>-2</sup> )
+/-/=	2/13/1	2/13/1	0/14/2	6/7/3		

<sup>1</sup> “+”, “–”, or “=” indicates that the corresponding algorithm is significantly better than, worse than, or comparable to DP-NSGA-III. The best average value among these algorithms on each test problem is highlighted in bold.

Table A2. HV values obtained by DP-NSGA-III and other peer algorithms in the C-DTLZ suite <sup>1</sup>.

Problem	M	NSGA-III	C-TAEA	DCNSGA-III	MOEA/D-2WA	DP-NSGA-III
C1DTLZ1	3	8.3136 × 10 <sup>-1</sup> (8.55 × 10 <sup>-3</sup> )–	8.3071 × 10 <sup>-1</sup> (6.34 × 10 <sup>-3</sup> )–	8.3558 × 10 <sup>-1</sup> (3.94 × 10 <sup>-3</sup> )–	8.3578 × 10 <sup>-1</sup> (4.04 × 10 <sup>-3</sup> )=	<b>8.3647</b> × 10 <sup>-1</sup> ( <b>4.30</b> × 10 <sup>-3</sup> )
	5	9.7520 × 10 <sup>-1</sup> (3.95 × 10 <sup>-3</sup> )=	9.7465 × 10 <sup>-1</sup> (4.62 × 10 <sup>-3</sup> )=	<b>9.7625</b> × 10 <sup>-1</sup> ( <b>4.68</b> × 10 <sup>-3</sup> )=	9.7527 × 10 <sup>-1</sup> (3.61 × 10 <sup>-3</sup> )=	9.7538 × 10 <sup>-1</sup> (3.26 × 10 <sup>-3</sup> )
	10	9.9394 × 10 <sup>-1</sup> (6.75 × 10 <sup>-3</sup> )–	9.9842 × 10 <sup>-1</sup> (1.32 × 10 <sup>-3</sup> )=	9.9659 × 10 <sup>-1</sup> (3.92 × 10 <sup>-3</sup> )–	9.8515 × 10 <sup>-1</sup> (8.06 × 10 <sup>-3</sup> )–	<b>9.9867</b> × 10 <sup>-1</sup> ( <b>8.77</b> × 10 <sup>-4</sup> )
	15	9.9446 × 10 <sup>-1</sup> (6.76 × 10 <sup>-3</sup> )–	9.9484 × 10 <sup>-1</sup> (7.10 × 10 <sup>-3</sup> )–	9.9862 × 10 <sup>-1</sup> (2.56 × 10 <sup>-3</sup> )–	8.9223 × 10 <sup>-1</sup> (2.59 × 10 <sup>-2</sup> )–	<b>9.9985</b> × 10 <sup>-1</sup> ( <b>7.57</b> × 10 <sup>-5</sup> )
C1DTLZ3	3	3.0072 × 10 <sup>-1</sup> (2.78 × 10 <sup>-1</sup> )–	5.3960 × 10 <sup>-1</sup> (9.57 × 10 <sup>-3</sup> )–	1.8095 × 10 <sup>-1</sup> (2.62 × 10 <sup>-1</sup> )–	2.2632 × 10 <sup>-1</sup> (2.76 × 10 <sup>-1</sup> )–	<b>5.5855</b> × 10 <sup>-1</sup> ( <b>6.05</b> × 10 <sup>-4</sup> )
	5	3.6162 × 10 <sup>-1</sup> (3.96 × 10 <sup>-1</sup> )–	1.7036 × 10 <sup>-1</sup> (3.14 × 10 <sup>-1</sup> )–	2.7564 × 10 <sup>-1</sup> (3.78 × 10 <sup>-1</sup> )–	3.4886 × 10 <sup>-1</sup> (3.91 × 10 <sup>-1</sup> )–	<b>8.1262</b> × 10 <sup>-1</sup> ( <b>4.19</b> × 10 <sup>-4</sup> )
	10	5.8279 × 10 <sup>-1</sup> (4.74 × 10 <sup>-1</sup> )–	8.4921 × 10 <sup>-1</sup> (2.46 × 10 <sup>-1</sup> )–	4.0252 × 10 <sup>-1</sup> (4.69 × 10 <sup>-1</sup> )–	1.9396 × 10 <sup>-1</sup> (3.95 × 10 <sup>-1</sup> )–	<b>9.6961</b> × 10 <sup>-1</sup> ( <b>2.85</b> × 10 <sup>-4</sup> )
	15	9.5280 × 10 <sup>-2</sup> (2.91 × 10 <sup>-1</sup> )–	5.5117 × 10 <sup>-2</sup> (2.10 × 10 <sup>-1</sup> )–	1.1506 × 10 <sup>-1</sup> (3.01 × 10 <sup>-1</sup> )–	3.3022 × 10 <sup>-2</sup> (1.81 × 10 <sup>-1</sup> )–	<b>9.9059</b> × 10 <sup>-1</sup> ( <b>1.37</b> × 10 <sup>-4</sup> )
C2DTLZ2	3	5.0609 × 10 <sup>-1</sup> (1.76 × 10 <sup>-3</sup> )+	5.0209 × 10 <sup>-1</sup> (3.47 × 10 <sup>-3</sup> )–	5.0232 × 10 <sup>-1</sup> (2.22 × 10 <sup>-3</sup> )–	<b>5.1261</b> × 10 <sup>-1</sup> ( <b>6.12</b> × 10 <sup>-4</sup> )+	5.0530 × 10 <sup>-1</sup> (1.20 × 10 <sup>-3</sup> )
	5	<b>75624</b> × 10 <sup>-1</sup> ( <b>5.87</b> × 10 <sup>-4</sup> )+	7.3985 × 10 <sup>-1</sup> (1.77 × 10 <sup>-3</sup> )–	7.5597 × 10 <sup>-1</sup> (6.33 × 10 <sup>-4</sup> )+	7.4867 × 10 <sup>-1</sup> (1.36 × 10 <sup>-3</sup> )+	7.4777 × 10 <sup>-1</sup> (1.18 × 10 <sup>-3</sup> )
	10	8.4100 × 10 <sup>-1</sup> (1.74 × 10 <sup>-1</sup> )–	8.7347 × 10 <sup>-1</sup> (4.96 × 10 <sup>-3</sup> )–	8.9450 × 10 <sup>-1</sup> (1.44 × 10 <sup>-3</sup> )–	8.1762 × 10 <sup>-1</sup> (1.29 × 10 <sup>-1</sup> )–	<b>9.9540</b> × 10 <sup>-1</sup> ( <b>1.64</b> × 10 <sup>-3</sup> )
	15	8.7433 × 10 <sup>-1</sup> (1.84 × 10 <sup>-1</sup> )–	9.3273 × 10 <sup>-1</sup> (3.23 × 10 <sup>-3</sup> )–	9.2978 × 10 <sup>-1</sup> (2.75 × 10 <sup>-2</sup> )–	8.3762 × 10 <sup>-1</sup> (2.39 × 10 <sup>-1</sup> )–	<b>9.5717</b> × 10 <sup>-1</sup> ( <b>3.37</b> × 10 <sup>-4</sup> )
C3DTLZ4	3	7.5074 × 10 <sup>-1</sup> (1.00 × 10 <sup>-1</sup> )–	7.8462 × 10 <sup>-1</sup> (1.02 × 10 <sup>-3</sup> )+	7.7286 × 10 <sup>-1</sup> (6.60 × 10 <sup>-2</sup> )+	<b>8.0612</b> × 10 <sup>-1</sup> ( <b>8.42</b> × 10 <sup>-5</sup> )+	7.6916 × 10 <sup>-1</sup> (1.83 × 10 <sup>-2</sup> )
	5	9.6435 × 10 <sup>-1</sup> (5.90 × 10 <sup>-4</sup> )+	9.5697 × 10 <sup>-1</sup> (6.41 × 10 <sup>-4</sup> )–	9.5681 × 10 <sup>-1</sup> (1.85 × 10 <sup>-2</sup> )–	<b>9.6535</b> × 10 <sup>-1</sup> ( <b>2.08</b> × 10 <sup>-4</sup> )+	9.6111 × 10 <sup>-1</sup> (1.78 × 10 <sup>-3</sup> )
	10	9.9912 × 10 <sup>-1</sup> (1.35 × 10 <sup>-3</sup> )+	9.9904 × 10 <sup>-1</sup> (4.95 × 10 <sup>-5</sup> )+	9.9549 × 10 <sup>-1</sup> (6.71 × 10 <sup>-3</sup> )–	<b>9.9947</b> × 10 <sup>-1</sup> ( <b>1.94</b> × 10 <sup>-5</sup> )+	9.9891 × 10 <sup>-1</sup> (1.22 × 10 <sup>-3</sup> )
	15	9.9929 × 10 <sup>-1</sup> (6.53 × 10 <sup>-4</sup> )–	<b>9.9997</b> × 10 <sup>-1</sup> ( <b>5.39</b> × 10 <sup>-6</sup> )+	9.9924 × 10 <sup>-1</sup> (1.62 × 10 <sup>-3</sup> )–	9.9975 × 10 <sup>-1</sup> (7.88 × 10 <sup>-4</sup> )+	9.9942 × 10 <sup>-1</sup> (2.61 × 10 <sup>-3</sup> )
+/-/=	4/11/1	3/11/2	2/12/2	6/8/2		

<sup>1</sup> “+”, “–”, or “=” indicates that the corresponding algorithm is significantly better than, worse than, or comparable to DP-NSGA-III. The best average value among these algorithms on each test problem is highlighted in bold.

Table A3. IGD values obtained by DP-NSGA-III and other peer algorithms in the DC-DTLZ suite <sup>1</sup>.

Problem	M	NSGA-III	C-TAEA	DCNSGA-III	MOEA/D-2WA	DP-NSGA-III
DC1DTLZ1	3	1.4149 × 10 <sup>-2</sup> (1.95 × 10 <sup>-4</sup> )–	1.5618 × 10 <sup>-2</sup> (4.13 × 10 <sup>-4</sup> )–	1.3807 × 10 <sup>-2</sup> (1.95 × 10 <sup>-4</sup> )–	2.7167 × 10 <sup>-2</sup> (5.37 × 10 <sup>-3</sup> )–	<b>1.3601</b> × 10 <sup>-2</sup> ( <b>1.51</b> × 10 <sup>-4</sup> )
	5	<b>3.9846</b> × 10 <sup>-2</sup> ( <b>1.65</b> × 10 <sup>-4</sup> )+	4.2228 × 10 <sup>-2</sup> (4.69 × 10 <sup>-4</sup> )–	4.0444 × 10 <sup>-2</sup> (2.61 × 10 <sup>-4</sup> )–	4.3698 × 10 <sup>-2</sup> (3.61 × 10 <sup>-5</sup> )–	4.0141 × 10 <sup>-2</sup> (1.91 × 10 <sup>-4</sup> )
	10	8.6732 × 10 <sup>-2</sup> (3.27 × 10 <sup>-3</sup> )–	9.6071 × 10 <sup>-2</sup> (2.55 × 10 <sup>-3</sup> )–	8.8756 × 10 <sup>-2</sup> (1.35 × 10 <sup>-2</sup> )–	<b>7.9532</b> × 10 <sup>-2</sup> ( <b>1.44</b> × 10 <sup>-4</sup> )+	8.5734 × 10 <sup>-2</sup> (1.12 × 10 <sup>-3</sup> )
	15	1.2823 × 10 <sup>-1</sup> (1.99 × 10 <sup>-5</sup> )=	1.4122 × 10 <sup>-1</sup> (5.93 × 10 <sup>-3</sup> )–	1.3728 × 10 <sup>-1</sup> (9.13 × 10 <sup>-3</sup> )–	<b>1.2557</b> × 10 <sup>-1</sup> ( <b>7.74</b> × 10 <sup>-4</sup> )+	1.2736 × 10 <sup>-1</sup> (1.97 × 10 <sup>-3</sup> )
DC1DTLZ3	3	4.6098 × 10 <sup>-2</sup> (9.37 × 10 <sup>-4</sup> )–	4.4586 × 10 <sup>-2</sup> (1.17 × 10 <sup>-3</sup> )=	4.7993 × 10 <sup>-2</sup> (1.13 × 10 <sup>-2</sup> )–	4.6526 × 10 <sup>-2</sup> (7.32 × 10 <sup>-3</sup> )–	<b>4.4474</b> × 10 <sup>-2</sup> ( <b>1.04</b> × 10 <sup>-3</sup> )
	5	1.4445 × 10 <sup>-1</sup> (6.38 × 10 <sup>-4</sup> )–	2.2595 × 10 <sup>-1</sup> (2.12 × 10 <sup>-2</sup> )–	1.4455 × 10 <sup>-1</sup> (8.29 × 10 <sup>-4</sup> )–	1.4340 × 10 <sup>0</sup> (7.70 × 10 <sup>-1</sup> )–	<b>1.4325</b> × 10 <sup>-1</sup> ( <b>9.35</b> × 10 <sup>-4</sup> )
	10	5.0342 × 10 <sup>-1</sup> (4.18 × 10 <sup>-2</sup> )–	5.1141 × 10 <sup>-1</sup> (2.40 × 10 <sup>-2</sup> )–	4.8251 × 10 <sup>-1</sup> (4.40 × 10 <sup>-2</sup> )–	<b>3.9529</b> × 10 <sup>-1</sup> ( <b>4.18</b> × 10 <sup>-3</sup> )+	4.0617 × 10 <sup>-1</sup> (6.62 × 10 <sup>-3</sup> )
	15	7.2158 × 10 <sup>-1</sup> (5.90 × 10 <sup>-2</sup> )–	7.1515 × 10 <sup>-1</sup> (3.52 × 10 <sup>-2</sup> )–	7.0274 × 10 <sup>-1</sup> (4.72 × 10 <sup>-2</sup> )–	6.1623 × 10 <sup>-1</sup> (6.01 × 10 <sup>-2</sup> )–	<b>6.0734</b> × 10 <sup>-1</sup> ( <b>1.82</b> × 10 <sup>-2</sup> )
DC2DTLZ1	3	1.4465 × 10 <sup>-1</sup> (6.13 × 10 <sup>-2</sup> )–	2.3327 × 10 <sup>-2</sup> (1.95 × 10 <sup>-4</sup> )–	6.9379 × 10 <sup>-2</sup> (7.56 × 10 <sup>-2</sup> )–	2.0561 × 10 <sup>-2</sup> (8.29 × 10 <sup>-6</sup> )=	<b>2.0559</b> × 10 <sup>-2</sup> ( <b>6.25</b> × 10 <sup>-6</sup> )
	5	1.4000 × 10 <sup>-1</sup> (3.94 × 10 <sup>-2</sup> )–	6.1657 × 10 <sup>-2</sup> (3.61 × 10 <sup>-4</sup> )–	8.0926 × 10 <sup>-2</sup> (5.34 × 10 <sup>-2</sup> )–	5.2707 × 10 <sup>-2</sup> (3.66 × 10 <sup>-6</sup> )–	<b>5.2706</b> × 10 <sup>-2</sup> ( <b>4.76</b> × 10 <sup>-6</sup> )
	10	1.7004 × 10 <sup>-1</sup> (2.17 × 10 <sup>-2</sup> )–	1.4388 × 10 <sup>-1</sup> (8.15 × 10 <sup>-4</sup> )–	1.1874 × 10 <sup>-1</sup> (2.31 × 10 <sup>-2</sup> )–	1.1113 × 10 <sup>-1</sup> (1.19 × 10 <sup>-2</sup> )–	<b>1.0951</b> × 10 <sup>-1</sup> ( <b>3.82</b> × 10 <sup>-5</sup> )
	15	2.7788 × 10 <sup>-1</sup> (3.77 × 10 <sup>-2</sup> )–	2.1802 × 10 <sup>-1</sup> (4.29 × 10 <sup>-3</sup> )–	1.9029 × 10 <sup>-1</sup> (2.61 × 10 <sup>-2</sup> )–	2.3111 × 10 <sup>-1</sup> (5.32 × 10 <sup>-2</sup> )=	<b>1.8368</b> × 10 <sup>-1</sup> ( <b>6.14</b> × 10 <sup>-4</sup> )

Table A3. Cont.

Problem	M	NSGA-III	C-TAEA	DCNSGA-III	MOEA/D-2WA	DP-NSGA-III
DC2DTLZ3	3	$5.6067 \times 10^{-1}(2.37 \times 10^{-4})-$	$6.3740 \times 10^{-2}(1.24 \times 10^{-3})-$	$5.3745 \times 10^{-1}(1.56 \times 10^{-1})-$	$3.7557 \times 10^{-1}(2.48 \times 10^{-1})-$	<b><math>5.4472 \times 10^{-2}(8.30 \times 10^{-6})</math></b>
	5	$5.9661 \times 10^{-1}(4.30 \times 10^{-4})-$	$1.8737 \times 10^{-1}(1.98 \times 10^{-3})-$	$3.4857 \times 10^{-1}(2.31 \times 10^{-1})-$	$1.6513 \times 10^{-1}(1.04 \times 10^{-5})=$	<b><math>1.6513 \times 10^{-1}(1.11 \times 10^{-5})</math></b>
	10	$8.3041 \times 10^{-1}(3.53 \times 10^{-2})-$	$4.3599 \times 10^{-1}(3.26 \times 10^{-3})-$	$5.9260 \times 10^{-1}(1.94 \times 10^{-1})-$	<b><math>4.2079 \times 10^{-1}(4.12 \times 10^{-4})+</math></b>	$4.2125 \times 10^{-1}(1.40 \times 10^{-4})$
	15	$1.0035 \times 10^0(1.64 \times 10^{-2})-$	$6.4346 \times 10^{-1}(2.09 \times 10^{-2})-$	$8.6106 \times 10^{-1}(1.69 \times 10^{-1})-$	$6.6804 \times 10^{-1}(1.19 \times 10^{-1})-$	<b><math>6.2027 \times 10^{-1}(4.59 \times 10^{-4})</math></b>
DC3DTLZ1	3	$2.2305 \times 10^{-1}(1.61 \times 10^{-1})-$	<b><math>9.9253 \times 10^{-3}(5.08 \times 10^{-4})=</math></b>	$1.0108 \times 10^{-2}(3.19 \times 10^{-4})=$	$2.0631 \times 10^{-2}(6.73 \times 10^{-3})-$	$9.9561 \times 10^{-3}(3.22 \times 10^{-4})$
	5	$8.8082 \times 10^{-2}(5.76 \times 10^{-2})-$	$3.3210 \times 10^{-2}(6.29 \times 10^{-3})-$	$2.5873 \times 10^{-2}(1.13 \times 10^{-2})=$	$5.9310 \times 10^{-2}(1.62 \times 10^{-2})-$	<b><math>2.2646 \times 10^{-2}(8.55 \times 10^{-4})</math></b>
DC3DTLZ3	3	$1.5363 \times 10^0(4.87 \times 10^{-1})-$	$4.2066 \times 10^{-2}(1.63 \times 10^{-2})=$	$6.2999 \times 10^{-1}(2.56 \times 10^{-1})-$	$5.8570 \times 10^{-1}(1.32 \times 10^{-1})-$	<b><math>3.0312 \times 10^{-2}(1.72 \times 10^{-3})</math></b>
	5	$9.5209 \times 10^{-1}(4.83 \times 10^{-1})-$	$1.0169 \times 10^{-1}(1.03 \times 10^{-2})-$	$3.1576 \times 10^{-1}(2.48 \times 10^{-1})-$	$2.4876 \times 10^{-1}(1.25 \times 10^{-1})-$	<b><math>8.7801 \times 10^{-2}(2.90 \times 10^{-3})</math></b>
	10	$1.6601 \times 10^0(6.41 \times 10^{-1})-$	$1.8491 \times 10^{-1}(6.92 \times 10^{-2})-$	$6.3620 \times 10^{-1}(3.84 \times 10^{-1})-$	$4.9978 \times 10^{-1}(6.75 \times 10^{-2})-$	<b><math>1.2029 \times 10^{-1}(9.22 \times 10^{-3})</math></b>
	15	$3.0696 \times 10^0(1.00 \times 10^0)-$	$6.2199 \times 10^{-1}(3.64 \times 10^{-1})-$	$7.5702 \times 10^{-1}(4.11 \times 10^{-1})-$	$7.3173 \times 10^{-1}(2.68 \times 10^{-1})-$	<b><math>1.4118 \times 10^{-1}(9.31 \times 10^{-3})</math></b>
+/-/=	1/20/1	0/19/3	0/19/3	4/14/4		

<sup>1</sup> “+”, “-”, or “=” indicates that the corresponding algorithm is significantly better than, worse than, or comparable to DP-NSGA-III. The best average value among these algorithms on each test problem is highlighted in bold.

Table A4. HV values obtained by DP-NSGA-III and other peer algorithms in the DC-DTLZ suite <sup>1</sup>.

Problem	M	NSGA-III	C-TAEA	DCNSGA-III	MOEA/D-2WA	DP-NSGA-III
DC1DTLZ1	3	<b><math>6.2908 \times 10^{-1}(1.80 \times 10^{-3})+</math></b>	$6.2624 \times 10^{-1}(1.18 \times 10^{-3})+$	$6.2456 \times 10^{-1}(2.30 \times 10^{-3})=$	$5.9725 \times 10^{-1}(1.81 \times 10^{-2})-$	$6.2544 \times 10^{-1}(1.51 \times 10^{-3})$
	5	<b><math>7.7284 \times 10^{-1}(8.57 \times 10^{-4})+</math></b>	$7.6956 \times 10^{-1}(8.87 \times 10^{-4})-$	$7.6977 \times 10^{-1}(7.37 \times 10^{-4})-$	$7.6525 \times 10^{-1}(1.74 \times 10^{-4})-$	$7.7057 \times 10^{-1}(1.18 \times 10^{-3})$
	10	$7.9618 \times 10^{-1}(2.24 \times 10^{-4})-$	$7.9609 \times 10^{-1}(1.14 \times 10^{-4})-$	$7.9345 \times 10^{-1}(1.66 \times 10^{-2})-$	$7.5413 \times 10^{-1}(5.97 \times 10^{-3})-$	<b><math>7.9681 \times 10^{-1}(1.11 \times 10^{-4})</math></b>
	15	$7.9782 \times 10^{-1}(7.60 \times 10^{-6})-$	$7.9700 \times 10^{-1}(5.69 \times 10^{-4})-$	$7.9658 \times 10^{-1}(2.55 \times 10^{-3})-$	$6.3463 \times 10^{-1}(2.12 \times 10^{-2})-$	<b><math>7.9784 \times 10^{-1}(2.01 \times 10^{-5})</math></b>
DC1DTLZ3	3	$4.6783 \times 10^{-1}(1.08 \times 10^{-3})=$	$4.5876 \times 10^{-1}(2.86 \times 10^{-3})-$	$4.6354 \times 10^{-1}(2.02 \times 10^{-2})=$	<b><math>4.6891 \times 10^{-1}(9.39 \times 10^{-3})+</math></b>	$4.6701 \times 10^{-1}(1.88 \times 10^{-3})$
	5	$7.6344 \times 10^{-1}(1.04 \times 10^{-3})=$	$6.8741 \times 10^{-1}(2.11 \times 10^{-2})-$	$7.6321 \times 10^{-1}(8.69 \times 10^{-4})-$	$5.6919 \times 10^{-3}(1.88 \times 10^{-2})-$	<b><math>7.6382 \times 10^{-1}(9.38 \times 10^{-4})</math></b>
	10	$8.6557 \times 10^{-1}(9.56 \times 10^{-2})-$	$9.0533 \times 10^{-1}(1.34 \times 10^{-2})-$	$8.8507 \times 10^{-1}(6.10 \times 10^{-2})-$	<b><math>9.6234 \times 10^{-1}(6.81 \times 10^{-4})+</math></b>	$9.6086 \times 10^{-1}(1.77 \times 10^{-3})$
	15	$7.1236 \times 10^{-1}(2.05 \times 10^{-1})-$	$7.6602 \times 10^{-1}(2.15 \times 10^{-1})-$	$7.5661 \times 10^{-1}(1.87 \times 10^{-1})-$	$9.6225 \times 10^{-1}(1.39 \times 10^{-1})-$	<b><math>9.7225 \times 10^{-1}(1.22 \times 10^{-2})</math></b>
DC2DTLZ1	3	$5.2980 \times 10^{-1}(1.50 \times 10^{-1})-$	$8.3802 \times 10^{-1}(3.19 \times 10^{-4})-$	$7.1724 \times 10^{-1}(1.93 \times 10^{-1})-$	$8.4167 \times 10^{-1}(1.11 \times 10^{-4})=$	<b><math>8.4168 \times 10^{-1}(9.28 \times 10^{-5})</math></b>
	5	$8.5826 \times 10^{-1}(5.50 \times 10^{-2})-$	$9.7691 \times 10^{-1}(2.39 \times 10^{-4})-$	$9.3819 \times 10^{-1}(8.13 \times 10^{-2})-$	$9.7987 \times 10^{-1}(1.24 \times 10^{-4})=$	<b><math>9.7989 \times 10^{-1}(1.22 \times 10^{-4})</math></b>
	10	$9.7880 \times 10^{-1}(7.28 \times 10^{-3})-$	$9.9944 \times 10^{-1}(2.90 \times 10^{-5})-$	$9.9753 \times 10^{-1}(7.40 \times 10^{-3})=$	$9.9893 \times 10^{-1}(4.09 \times 10^{-3})=$	<b><math>9.9968 \times 10^{-1}(1.55 \times 10^{-5})</math></b>
	15	$9.7478 \times 10^{-1}(1.01 \times 10^{-2})-$	$9.9978 \times 10^{-1}(8.11 \times 10^{-5})-$	$9.9493 \times 10^{-1}(1.60 \times 10^{-2})-$	$9.8577 \times 10^{-1}(1.44 \times 10^{-2})-$	<b><math>9.9992 \times 10^{-1}(9.13 \times 10^{-6})</math></b>
DC2DTLZ3	3	$8.1376 \times 10^{-3}(1.24 \times 10^{-4})-$	$5.4647 \times 10^{-1}(1.30 \times 10^{-3})-$	$4.8563 \times 10^{-2}(1.47 \times 10^{-1})-$	$2.1007 \times 10^{-1}(2.70 \times 10^{-1})-$	<b><math>5.5933 \times 10^{-1}(2.64 \times 10^{-4})</math></b>
	5	$7.1752 \times 10^{-2}(3.92 \times 10^{-4})-$	$7.9251 \times 10^{-1}(2.19 \times 10^{-3})-$	$5.1276 \times 10^{-1}(3.74 \times 10^{-1})-$	<b><math>8.1267 \times 10^{-1}(3.45 \times 10^{-4})=</math></b>	$8.1257 \times 10^{-1}(4.34 \times 10^{-4})$
	10	$2.0234 \times 10^{-1}(4.70 \times 10^{-2})-$	$9.6831 \times 10^{-1}(1.11 \times 10^{-3})-$	$6.6125 \times 10^{-1}(3.78 \times 10^{-1})-$	$9.6980 \times 10^{-1}(1.76 \times 10^{-4})=$	<b><math>9.6981 \times 10^{-1}(1.25 \times 10^{-4})</math></b>
	15	$1.8326 \times 10^{-2}(1.06 \times 10^{-2})-$	$9.4186 \times 10^{-1}(3.48 \times 10^{-2})-$	$3.6589 \times 10^{-1}(4.69 \times 10^{-1})-$	$8.5857 \times 10^{-1}(3.43 \times 10^{-1})=$	<b><math>9.9069 \times 10^{-1}(1.18 \times 10^{-4})</math></b>
DC3DTLZ1	3	$9.6576 \times 10^{-2}(1.19 \times 10^{-1})-$	$5.2180 \times 10^{-1}(3.16 \times 10^{-3})-$	<b><math>5.2776 \times 10^{-1}(1.74 \times 10^{-3})=</math></b>	$5.0561 \times 10^{-1}(3.22 \times 10^{-2})=$	$5.2768 \times 10^{-1}(1.33 \times 10^{-3})$
	5	$5.4568 \times 10^{-2}(6.35 \times 10^{-2})-$	$8.8297 \times 10^{-2}(1.19 \times 10^{-2})-$	$1.2713 \times 10^{-1}(5.84 \times 10^{-3})-$	$9.3507 \times 10^{-2}(1.42 \times 10^{-2})-$	<b><math>1.2965 \times 10^{-1}(1.61 \times 10^{-3})</math></b>
DC3DTLZ3	3	$0.0000 \times 10^0(0.00 \times 10^0)-$	$3.3638 \times 10^{-1}(2.15 \times 10^{-2})-$	$1.2002 \times 10^{-2}(6.57 \times 10^{-2})-$	$1.2125 \times 10^{-2}(6.64 \times 10^{-2})-$	<b><math>3.5909 \times 10^{-1}(1.89 \times 10^{-3})</math></b>
	5	$0.0000 \times 10^0(0.00 \times 10^0)-$	$5.2458 \times 10^{-1}(2.00 \times 10^{-2})-$	$3.0684 \times 10^{-1}(2.92 \times 10^{-1})-$	$5.2956 \times 10^{-1}(5.67 \times 10^{-2})=$	<b><math>5.7579 \times 10^{-1}(3.94 \times 10^{-3})</math></b>
	10	$0.0000 \times 10^0(0.00 \times 10^0)-$	$6.6668 \times 10^{-1}(5.64 \times 10^{-2})-$	$2.5906 \times 10^{-1}(3.73 \times 10^{-1})-$	$6.3365 \times 10^{-1}(4.59 \times 10^{-2})-$	<b><math>7.8243 \times 10^{-1}(7.14 \times 10^{-3})</math></b>
	15	$0.0000 \times 10^0(0.00 \times 10^0)-$	$7.4586 \times 10^{-3}(2.03 \times 10^{-2})-$	$1.0836 \times 10^{-2}(2.96 \times 10^{-2})-$	$8.4142 \times 10^{-2}(4.73 \times 10^{-2})-$	<b><math>1.1732 \times 10^{-1}(1.54 \times 10^{-2})</math></b>
+/-/=	2/18/2	1/21/0	0/17/5	2/12/8		

<sup>1</sup> “+”, “-”, or “=” indicates that the corresponding algorithm is significantly better than, worse than, or comparable to DP-NSGA-III. The best average value among these algorithms on each test problem is highlighted in bold.

Table A5. IGD values obtained by DP-NSGA-III and other peer algorithms in the MW suite. <sup>1</sup>

Problem	M	NSGA-III	C-TAEA	DCNSGA-III	MOEA/D-2WA	DP-NSGA-III
MW4	3	<b><math>4.1535 \times 10^{-2}(1.08 \times 10^{-4})+</math></b>	$4.6663 \times 10^{-2}(3.46 \times 10^{-4})-$	$4.2120 \times 10^{-2}(5.01 \times 10^{-4})=$	$4.1993 \times 10^{-2}(9.15 \times 10^{-4})+$	$4.2103 \times 10^{-2}(3.36 \times 10^{-4})$
	5	$1.0474 \times 10^{-1}(2.36 \times 10^{-4})+$	$1.2218 \times 10^{-1}(1.18 \times 10^{-3})-$	$1.0500 \times 10^{-1}(1.69 \times 10^{-4})+$	<b><math>1.0430 \times 10^{-1}(3.57 \times 10^{-4})+</math></b>	$1.0521 \times 10^{-1}(1.78 \times 10^{-4})$
	10	$2.1816 \times 10^{-1}(1.12 \times 10^{-2})+$	$2.7641 \times 10^{-1}(6.24 \times 10^{-3})-$	$2.2047 \times 10^{-1}(7.34 \times 10^{-3})-$	<b><math>2.0360 \times 10^{-1}(8.93 \times 10^{-4})+</math></b>	$2.1994 \times 10^{-1}(1.54 \times 10^{-3})$
	15	$3.7194 \times 10^{-1}(6.44 \times 10^{-3})=$	$4.3583 \times 10^{-1}(9.90 \times 10^{-3})-$	$3.7041 \times 10^{-1}(4.42 \times 10^{-3})=$	<b><math>2.4986 \times 10^{-1}(1.21 \times 10^{-3})+</math></b>	$3.7182 \times 10^{-1}(6.68 \times 10^{-3})$
MW8	3	$6.2302 \times 10^{-2}(3.83 \times 10^{-2})-$	$5.5022 \times 10^{-2}(3.58 \times 10^{-3})-$	$4.9781 \times 10^{-2}(2.19 \times 10^{-3})-$	$5.1319 \times 10^{-2}(3.30 \times 10^{-3})-$	<b><math>4.8680 \times 10^{-2}(2.65 \times 10^{-3})</math></b>
	5	$1.5959 \times 10^{-1}(3.49 \times 10^{-2})-$	$1.7035 \times 10^{-1}(1.87 \times 10^{-3})-$	$1.5266 \times 10^{-1}(2.67 \times 10^{-4})-$	$1.5301 \times 10^{-1}(4.01 \times 10^{-4})-$	<b><math>1.5216 \times 10^{-1}(3.66 \times 10^{-4})</math></b>
	10	$4.2660 \times 10^{-1}(5.27 \times 10^{-2})-$	$4.1017 \times 10^{-1}(3.11 \times 10^{-3})-$	$4.0947 \times 10^{-1}(4.64 \times 10^{-2})-$	<b><math>3.9667 \times 10^{-1}(2.06 \times 10^{-3})+</math></b>	$3.9871 \times 10^{-1}(1.65 \times 10^{-3})$
	15	$6.4708 \times 10^{-1}(1.38 \times 10^{-2})-$	$6.4468 \times 10^{-1}(2.82 \times 10^{-3})-$	$6.4353 \times 10^{-1}(1.22 \times 10^{-2})-$	<b><math>6.0777 \times 10^{-1}(4.53 \times 10^{-3})+</math></b>	$6.2793 \times 10^{-1}(2.55 \times 10^{-3})$
MW14	3	$1.3425 \times 10^{-1}(1.02 \times 10^{-2})-$	<b><math>1.1616 \times 10^{-1}(3.84 \times 10^{-3})+</math></b>	$1.3657 \times 10^{-1}(2.50 \times 10^{-2})-$	$2.0790 \times 10^{-1}(5.07 \times 10^{-3})-$	$1.2706 \times 10^{-1}(3.38 \times 10^{-3})$
	5	$3.6249 \times 10^{-1}(2.95 \times 10^{-2})-$	<b><math>3.0987 \times 10^{-1}(6.06 \times 10^{-3})+</math></b>	$3.8139 \times 10^{-1}(4.39 \times 10^{-2})-$	$4.6584 \times 10^{-1}(7.65 \times 10^{-3})-$	$3.3364 \times 10^{-1}(2.80 \times 10^{-2})$
	10	$1.1445 \times 10^0(2.49 \times 10^{-2})-$	$1.4023 \times 10^0(5.04 \times 10^{-2})-$	$1.1844 \times 10^0(2.65 \times 10^{-2})-$	$1.4500 \times 10^0(2.06 \times 10^{-1})-$	<b><math>1.0925 \times 10^0(2.83 \times 10^{-2})</math></b>
	15	$3.2374 \times 10^0(7.09 \times 10^{-2})-$	$3.0938 \times 10^0(1.41 \times 10^{-1})-$	$3.3312 \times 10^0(1.33 \times 10^{-1})-$	$3.1616 \times 10^0(7.02 \times 10^{-2})-$	<b><math>2.7544 \times 10^0(1.58 \times 10^{-1})</math></b>
+/-/=	3/8/1	2/10/0	1/9/2	6/6/0		

<sup>1</sup> “+”, “-”, or “=” indicates that the corresponding algorithm is significantly better than, worse than, or comparable to DP-NSGA-III. The best average value among these algorithms on each test problem is highlighted in bold.

**Table A6.** HV values obtained by DP-NSGA-III and other peer algorithms in the C-DTLZ suite <sup>1</sup>.

Problem	M	NSGA-III	C-TAEA	DCNSGA-III	MOEA/D-2WA	DP-NSGA-III
MW4	3	<b>8.4121</b> × 10 <sup>-1</sup> ( <b>1.51</b> × 10 <sup>-4</sup> ) <b>+</b>	8.3793 × 10 <sup>-1</sup> (3.18 × 10 <sup>-4</sup> ) <b>-</b>	8.4069 × 10 <sup>-1</sup> (4.56 × 10 <sup>-4</sup> ) <b>=</b>	8.4026 × 10 <sup>-1</sup> (2.58 × 10 <sup>-3</sup> ) <b>=</b>	8.4076 × 10 <sup>-1</sup> (2.82 × 10 <sup>-4</sup> )
	5	<b>9.7976</b> × 10 <sup>-1</sup> ( <b>1.64</b> × 10 <sup>-4</sup> ) <b>+</b>	9.7721 × 10 <sup>-1</sup> (2.88 × 10 <sup>-4</sup> ) <b>-</b>	9.7969 × 10 <sup>-1</sup> (1.07 × 10 <sup>-4</sup> ) <b>+</b>	9.7957 × 10 <sup>-1</sup> (1.51 × 10 <sup>-4</sup> ) <b>=</b>	9.7954 × 10 <sup>-1</sup> (1.78 × 10 <sup>-4</sup> )
	10	9.9955 × 10 <sup>-1</sup> (6.84 × 10 <sup>-4</sup> ) <b>=</b>	9.9952 × 10 <sup>-1</sup> (5.81 × 10 <sup>-5</sup> ) <b>-</b>	9.9966 × 10 <sup>-1</sup> (7.69 × 10 <sup>-5</sup> ) <b>=</b>	9.6297 × 10 <sup>-1</sup> (9.11 × 10 <sup>-3</sup> ) <b>-</b>	<b>9.9967</b> × 10 <sup>-1</sup> ( <b>2.65</b> × 10 <sup>-5</sup> )
	15	9.9990 × 10 <sup>-1</sup> (9.83 × 10 <sup>-6</sup> ) <b>=</b>	9.9984 × 10 <sup>-1</sup> (2.52 × 10 <sup>-5</sup> ) <b>-</b>	9.9990 × 10 <sup>-1</sup> (9.07 × 10 <sup>-6</sup> ) <b>=</b>	8.2556 × 10 <sup>-1</sup> (2.54 × 10 <sup>-2</sup> ) <b>-</b>	<b>9.9991</b> × 10 <sup>-1</sup> ( <b>1.00</b> × 10 <sup>-5</sup> )
MW8	3	5.0730 × 10 <sup>-1</sup> (4.92 × 10 <sup>-2</sup> ) <b>-</b>	5.1953 × 10 <sup>-1</sup> (1.50 × 10 <sup>-2</sup> ) <b>-</b>	5.2886 × 10 <sup>-1</sup> (9.19 × 10 <sup>-3</sup> ) <b>-</b>	5.3090 × 10 <sup>-1</sup> (1.40 × 10 <sup>-2</sup> ) <b>=</b>	<b>5.3278</b> × 10 <sup>-1</sup> ( <b>1.18</b> × 10 <sup>-2</sup> )
	5	8.0433 × 10 <sup>-1</sup> (2.12 × 10 <sup>-2</sup> ) <b>=</b>	7.8742 × 10 <sup>-1</sup> (5.44 × 10 <sup>-3</sup> ) <b>-</b>	<b>8.1027</b> × 10 <sup>-1</sup> ( <b>2.94</b> × 10 <sup>-3</sup> ) <b>+</b>	8.0856 × 10 <sup>-1</sup> (4.10 × 10 <sup>-3</sup> ) <b>=</b>	8.1020 × 10 <sup>-1</sup> (2.88 × 10 <sup>-3</sup> )
	10	9.5443 × 10 <sup>-1</sup> (2.29 × 10 <sup>-2</sup> ) <b>=</b>	9.6348 × 10 <sup>-1</sup> (2.09 × 10 <sup>-3</sup> ) <b>-</b>	9.6277 × 10 <sup>-1</sup> (1.94 × 10 <sup>-2</sup> ) <b>-</b>	<b>9.6889</b> × 10 <sup>-1</sup> ( <b>1.33</b> × 10 <sup>-3</sup> ) <b>+</b>	9.6775 × 10 <sup>-1</sup> (6.76 × 10 <sup>-4</sup> )
	15	9.7484 × 10 <sup>-1</sup> (1.10 × 10 <sup>-2</sup> ) <b>-</b>	9.4017 × 10 <sup>-1</sup> (4.48 × 10 <sup>-3</sup> ) <b>-</b>	9.7489 × 10 <sup>-1</sup> (9.28 × 10 <sup>-3</sup> ) <b>-</b>	9.8542 × 10 <sup>-1</sup> (1.19 × 10 <sup>-3</sup> ) <b>-</b>	<b>9.9014</b> × 10 <sup>-1</sup> ( <b>2.21</b> × 10 <sup>-4</sup> )
MW14	3	4.5894 × 10 <sup>-1</sup> (6.58 × 10 <sup>-3</sup> ) <b>-</b>	4.6358 × 10 <sup>-1</sup> (2.95 × 10 <sup>-3</sup> ) <b>-</b>	4.5976 × 10 <sup>-1</sup> (1.11 × 10 <sup>-2</sup> ) <b>-</b>	4.3775 × 10 <sup>-1</sup> (3.53 × 10 <sup>-3</sup> ) <b>-</b>	<b>4.6539</b> × 10 <sup>-1</sup> ( <b>2.91</b> × 10 <sup>-3</sup> )
	5	3.4702 × 10 <sup>-1</sup> (1.53 × 10 <sup>-2</sup> ) <b>-</b>	<b>3.9414</b> × 10 <sup>-1</sup> ( <b>3.36</b> × 10 <sup>-3</sup> ) <b>+</b>	3.6253 × 10 <sup>-1</sup> (1.99 × 10 <sup>-2</sup> ) <b>-</b>	3.2687 × 10 <sup>-1</sup> (2.63 × 10 <sup>-3</sup> ) <b>-</b>	3.8548 × 10 <sup>-1</sup> (7.65 × 10 <sup>-3</sup> )
	10	2.0239 × 10 <sup>-1</sup> (4.52 × 10 <sup>-3</sup> ) <b>-</b>	<b>2.2912</b> × 10 <sup>-1</sup> ( <b>4.66</b> × 10 <sup>-3</sup> ) <b>+</b>	2.0483 × 10 <sup>-1</sup> (3.25 × 10 <sup>-3</sup> ) <b>-</b>	1.5608 × 10 <sup>-1</sup> (2.01 × 10 <sup>-2</sup> ) <b>-</b>	2.0902 × 10 <sup>-1</sup> (5.06 × 10 <sup>-3</sup> )
	15	1.5134 × 10 <sup>-1</sup> (2.58 × 10 <sup>-3</sup> ) <b>-</b>	1.5365 × 10 <sup>-1</sup> (3.57 × 10 <sup>-3</sup> ) <b>-</b>	1.5918 × 10 <sup>-1</sup> (1.04 × 10 <sup>-2</sup> ) <b>-</b>	7.7070 × 10 <sup>-2</sup> (1.17 × 10 <sup>-2</sup> ) <b>-</b>	<b>1.6459</b> × 10 <sup>-1</sup> ( <b>3.82</b> × 10 <sup>-3</sup> )
+/-/=	2/6/4	2/10/0	2/7/3	1/7/4		

<sup>1</sup> “+”, “-”, or “=” indicates that the corresponding algorithm is significantly better than, worse than, or comparable to DP-NSGA-III. The best average value among these algorithms on each test problem is highlighted in bold.

## References

- Mala-Jetmarova, H.; Sultanova, N.; Savic, D. Lost in optimisation of water distribution systems? A literature review of system operation. *Environ. Model. Softw.* **2017**, *93*, 209–254. [\[CrossRef\]](#)
- Dong, W.; Zhou, K.; Qi, H.; He, C.; Zhang, J. A tissue P system based evolutionary algorithm for multi-objective VRPTW. *Swarm Evol. Comput.* **2018**, *39*, 310–322. [\[CrossRef\]](#)
- Pan, J.S.; Liu, N.; Chu, S.C. A competitive mechanism based multi-objective differential evolution algorithm and its application in feature selection. *Knowl.-Based Syst.* **2022**, *245*, 108582. [\[CrossRef\]](#)
- Geng, H.; Xu, K.; Zhang, Y.; Zhou, Z. A classification tree and decomposition based multi-objective evolutionary algorithm with adaptive operator selection. *Complex Intell. Syst.* **2022**, 1–18. [\[CrossRef\]](#)
- Wang, F.; Li, Y.; Zhang, H.; Hu, T.; Shen, X.L. An adaptive weight vector guided evolutionary algorithm for preference-based multi-objective optimization. *Swarm Evol. Comput.* **2019**, *49*, 220–233. [\[CrossRef\]](#)
- Fan, Z.; Li, W.; Cai, X.; Li, H.; Wei, C.; Zhang, Q.; Deb, K.; Goodman, E. Push and pull search for solving constrained multi-objective optimization problems. *Swarm Evol. Comput.* **2019**, *44*, 665–679. [\[CrossRef\]](#)
- Liu, Z.Z.; Wang, Y. Handling constrained multiobjective optimization problems with constraints in both the decision and objective spaces. *IEEE Trans. Evol. Comput.* **2019**, *23*, 870–884. [\[CrossRef\]](#)
- Ming, M.; Trivedi, A.; Wang, R.; Srinivasan, D.; Zhang, T. A dual-population-based evolutionary algorithm for constrained multiobjective optimization. *IEEE Trans. Evol. Comput.* **2021**, *25*, 739–753. [\[CrossRef\]](#)
- Deb, K.; Jain, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Trans. Evol. Comput.* **2013**, *18*, 577–601. [\[CrossRef\]](#)
- Zhou, Y.; Zhu, M.; Wang, J.; Zhang, Z.; Xiang, Y.; Zhang, J. Tri-goal evolution framework for constrained many-objective optimization. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *50*, 3086–3099. [\[CrossRef\]](#)
- Li, K.; Chen, R.; Fu, G.; Yao, X. Two-archive evolutionary algorithm for constrained multiobjective optimization. *IEEE Trans. Evol. Comput.* **2018**, *23*, 303–315. [\[CrossRef\]](#)
- Wang, J.; Liang, G.; Zhang, J. Cooperative differential evolution framework for constrained multiobjective optimization. *IEEE Trans. Cybern.* **2018**, *49*, 2060–2072. [\[CrossRef\]](#) [\[PubMed\]](#)
- Panichella, A. An adaptive evolutionary algorithm based on non-Euclidean geometry for many-objective optimization. In Proceedings of the Genetic and Evolutionary Computation Conference, Prague, Czech Republic, 13–17 July 2019; pp. 595–603.
- Woldesenbet, Y.G.; Yen, G.G.; Tessema, B.G. Constraint handling in multiobjective evolutionary optimization. *IEEE Trans. Evol. Comput.* **2009**, *13*, 514–525. [\[CrossRef\]](#)
- Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [\[CrossRef\]](#)
- Jan, M.A.; Khanum, R.A. A study of two penalty-parameterless constraint handling techniques in the framework of MOEA/D. *Appl. Soft Comput.* **2013**, *13*, 128–148. [\[CrossRef\]](#)
- Takahama, T.; Sakai, S. Efficient constrained optimization by the  $\epsilon$  constrained adaptive differential evolution. In Proceedings of the IEEE Congress on Evolutionary Computation, Barcelona, Spain, 18–23 July 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 1–8.
- Wang, Y.; Cai, Z. Combining multiobjective optimization with differential evolution to solve constrained optimization problems. *IEEE Trans. Evol. Comput.* **2012**, *16*, 117–134. [\[CrossRef\]](#)
- Tasgetiren, M.F.; Suganthan, P.N.; Pan, Q.K.; Mallipeddi, R.; Sarman, S. An ensemble of differential evolution algorithms for constrained function optimization. In Proceedings of the IEEE Congress on Evolutionary Computation, Barcelona, Spain, 18–23 July 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 1–8.

20. Jain, H.; Deb, K. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach. *IEEE Trans. Evol. Comput.* **2013**, *18*, 602–622. [[CrossRef](#)]
21. Fan, Z.; Fang, Y.; Li, W.; Cai, X.; Wei, C.; Goodman, E. MOEA/D with angle-based constrained dominance principle for constrained multi-objective optimization problems. *Appl. Soft Comput.* **2019**, *74*, 621–633. [[CrossRef](#)]
22. Tian, Y.; Zhang, Y.; Su, Y.; Zhang, X.; Tan, K.C.; Jin, Y. Balancing objective optimization and constraint satisfaction in constrained evolutionary multiobjective optimization. *IEEE Trans. Cybern.* **2021**, *52*, 9559–9572. [[CrossRef](#)]
23. Deb, K.; Agrawal, R.B. Simulated binary crossover for continuous search space. *Complex Syst.* **1995**, *9*, 115–148.
24. Deb, K.; Goyal, M. A combined genetic adaptive search (GeneAS) for engineering design. *Comput. Sci. Inform.* **1996**, *26*, 30–45.
25. Alefeld, G. On the convergence of Halley’s Method. *Am. Math. Mon.* **1981**, *88*, 530–536. [[CrossRef](#)]
26. Zhang, X.; Tian, Y.; Cheng, R.; Jin, Y. An efficient approach to nondominated sorting for evolutionary multiobjective optimization. *IEEE Trans. Evol. Comput.* **2014**, *19*, 201–213. [[CrossRef](#)]
27. Ma, Z.; Wang, Y. Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons. *IEEE Trans. Evol. Comput.* **2019**, *23*, 972–986. [[CrossRef](#)]
28. Bosman, P.A.; Thierens, D. The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Trans. Evol. Comput.* **2003**, *7*, 174–188. [[CrossRef](#)]
29. Zitzler, E.; Thiele, L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Trans. Evol. Comput.* **1999**, *3*, 257–271. [[CrossRef](#)]
30. Jiao, R.; Zeng, S.; Li, C.; Yang, S.; Ong, Y.S. Handling constrained many-objective optimization problems via problem transformation. *IEEE Trans. Cybern.* **2020**, *51*, 4834–4847. [[CrossRef](#)] [[PubMed](#)]
31. Jiao, R.; Zeng, S.; Li, C.; Ong, Y.S. Two-type weight adjustments in MOEA/D for highly constrained many-objective optimization. *Inf. Sci.* **2021**, *578*, 592–614. [[CrossRef](#)]
32. Tian, Y.; Cheng, R.; Zhang, X.; Jin, Y. PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]. *IEEE Comput. Intell. Mag.* **2017**, *12*, 73–87. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.