mathematics

MDPI

Article

A Constrained Multi-Objective Optimization Algorithm with a Population State Discrimination Model

Shaoyu Zhao [1,†], Heming Jia [1,*,†], Yongchao Li [2] and Qian Shi [3]

[1] School of Information Engineering, Sanming University, Sanming 365004, China; zsy@fjsmu.edu.cn
[2] School of Information and Electrical Engineering, Heilongjiang Bayi Agricultural University, Daqing 163319, China; liyongchao2001@byau.edu.cn
[3] School of Mathematics and Statistics, Changchun University of Technology, Changchun 130012, China; 2202412016@stu.ccut.edu.cn
* Correspondence: jiaheming@fjsmu.edu.cn
† These authors contributed equally to this work.

Abstract: The solution to constrained multi-objective optimization problems (CMOPs) requires optimizing the objective functions while satisfying the constraint conditions. To effectively address CMOPs, algorithms must balance objectives and constraints. However, the limited adaptability of specific constraint-handling techniques (CHTs) has hindered the widespread applicability of constrained multi-objective evolutionary algorithms (CMOEAs). To overcome this limitation, this article proposes a population state-based CMOEA. First, a model is developed to identify population states based on the positions of the primary and auxiliary populations. Tailored environmental selection models are then designed for the auxiliary population according to different states, enabling them to guide the evolution of the main population more effectively. By dynamizing the CHTs, the proposed algorithm can adapt to a broader and more complex range of CMOPs. Additionally, state-specific optimal individual selection methods are introduced, allowing the auxiliary population to escape local optima and accelerate exploration. A simple yet effective resource allocation model is incorporated to address the potential computational resource waste associated with dual populations, enhancing the resource utilization. Comprehensive tests, including comparisons with seven state-of-the-art algorithms, were conducted on 47 benchmark functions and 12 real-world problems. The experimental results demonstrate that the proposed CMOEA outperforms existing CMOEAs in its convergence and diversity.

Keywords: constrained multi-objective optimization; multiple population optimization models; population state discrimination

MSC: 90C29; 68W50

check for updates

Academic Editor: Jun He

Received: 16 January 2025
Revised: 13 February 2025
Accepted: 19 February 2025
Published: 20 February 2025

Citation: Zhao, S.; Jia, H.; Li, Y.; Shi, Q. A Constrained Multi-Objective Optimization Algorithm with a Population State Discrimination Model. *Mathematics* **2025**, *13*, 688. https://doi.org/10.3390/math13050688

Copyright: © 2025 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

In real-world scenarios, numerous constrained multi-objective optimization problems arise, such as vehicle scheduling [1], robot gripper optimization [2], and energy-saving strategies [3]. These problems involve the simultaneous consideration of multiple objectives and constraints, which significantly increases their complexity. With the advancement of intelligent computing, there has been growing and increasingly detailed research into addressing these challenges [4].

For a CMOP, it usually consists of the following form:

$$\min F(x) = (f_1(x), f_2(x), \cdots, f_M(x)) \tag{1}$$

$$s.t.\begin{cases} g_i(x) \le 0, & i = 1, \cdots, p \\ h_i(x) = 0, & i = p+1, \cdots, L \\ x \in S. \end{cases} \tag{2}$$

where $F(x)$ is the objective function, which is a vector composed of multiple subobjectives. M represents the number of subobjectives, which is the dimension of the objective function; x is a solution vector consisting in the decision space S, with a dimension of D; and there are a total of L constraints, including p inequality constraints $g(x)$ and $L - p$ equality constraints $h(x)$. The decision space S is defined by $g(x)$ and $h(x)$.

In addition, to compare the quality of the solutions, constraint violation values (CVs) are defined as follows:

$$CV(x) = \sum_{i=1}^{L} C_i(x) \tag{3}$$

There are two different types of methods for calculating C_i :

$$C_i(x) = \begin{cases} \max(0, g_i(x)), & 1 \le i \le p \\ \max(0, |h_i(x)| - \delta), & p+1 \le i \le L \end{cases} \tag{4}$$

where δ is a very small positive parameter used to relax the equality constraints. When the value of the CV is 0, x is a feasible solution; otherwise, it is an infeasible solution.

In the development of intelligent computing, evolutionary algorithms (EAs) have been widely employed to address constrained optimization problems (COPs). Recently, rapid advancements in intelligent computing technology have expanded the application of EAs to solving multi-objective optimization problems (MOPs). However, research on CMOPs remains in its early stages of development [5]. In MOPs, the primary goal is to identify a set of Pareto front (PF) solutions. In contrast, CMOPs involve both unconstrained Pareto front (UPF) and constrained Pareto front (CPF) solutions, which introduces additional complexity and challenges into the solution process. In continuous spaces, EAs can efficiently locate the PF. However, when the constraint conditions partition the decision space into multiple feasible regions, some PFs may reside in smaller subregions [6]. This fragmentation not only poses significant challenges for maintaining solution diversity but also hampers the evolutionary process. Infeasible regions often trap EAs in local optima, adversely affecting convergence. To address these challenges, various CHTs have been proposed, such as the ε model [7], which allows individuals to cross feasible domain boundaries by moderately relaxing the constraints. While such methods can assist with population evolution in specific cases, they face notable limitations. Specifically, it is often difficult to accurately identify potential high-quality solutions. Moreover, while certain CHTs may work effectively for specific CMOPs, it remains challenging to develop a universally applicable CHT that can reliably solve diverse types of CMOPs.

To further enhance the quality of the solution sets, numerous researchers have actively explored and proposed various CMOEAs. Among these, CMOEAs based on multiple populations or tasks have demonstrated remarkable performance improvements and have garnered significant attention from the academic community in recent years. Representative algorithms such as TP-HCEA [8], URCMO [9], and MTCMO [10] exemplify this approach. These algorithms leverage a collaborative optimization framework that utilizes an information exchange mechanism between two populations: one population focuses on evolving within the feasible domain to address convergence issues, while the other explores smaller feasible regions within infeasible areas to enhance the solution diversity. However, this algorithmic model has a notable drawback: the co-evolution process of the two populations requires substantial computational resources [11]. As a result, when significant resources are dedicated to intensive exploration of the UPF, the CPF often remains

insufficiently explored. This creates a critical conflict between the high computational cost and the need for improved algorithm convergence. This imbalance severely limits the optimization potential and overall performance of these algorithms [12].

To address the challenges outlined above and improve the quality of the solutions for CMOPs, this paper proposes a population-state-based constrained multi-objective (PSCMO) evolutionary algorithm. The proposed algorithm aims to mitigate the limitations of using a single CHT by dynamically adapting the CHTs based on the evolutionary state of the population. This approach prevents the population from becoming trapped in local optima and enhances the solution convergence. In PSCMO, the main population focuses on exploring the CPF, while the auxiliary population plays a guiding role in population evolution. The positional relationship between the main and auxiliary populations is continuously monitored, and this information is used to determine the population state. Based on the determined state, three different CHTs are employed to guide the selection and evolution of the auxiliary population. This real-time switching of the CHTs dynamically adjusts the direction of evolution, enabling the auxiliary population to guide the main population better. When the population evolution stagnates, the algorithm generates more diverse offspring to help escape local optima. Conversely, during smooth evolution, it produces more concentrated offspring to accelerate the convergence. To achieve this, the algorithm incorporates two distinct versions of Differential Evolution and designs specific optimal individual selection mechanisms tailored to both the main and auxiliary populations. Finally, to address the computational resource challenges posed by the dual-population model, a simple yet effective resource allocation model is introduced. This model minimizes resource waste by optimizing the balance between the two populations, thereby improving the overall computational efficiency of the algorithm. The contributions of this article are as follows:

1. The evolutionary state of the population is divided into three types: convergence, convergence–diversity balance, and diversity. A population state discrimination model is established to identify these states by recording the distance between the auxiliary and main populations and determining the state based on this relationship. In the convergence state, the population prioritizes convergence, and the selection of auxiliary individuals considers only the distance between the target space and the origin. In the convergence–diversity balance state, both convergence and diversity are considered, with the selection process accounting for the distance to the origin and the cosine similarity between individuals. In the diversity state, the population emphasizes diversity, and the selection considers both the cosine similarity between individuals and the pairwise distances among them. By adapting the population selection to the specific state, the auxiliary population is guided effectively, enhancing the optimization process.
2. Building on the two-stage concept, a specialized evolutionary operator design is applied to the auxiliary population. Specifically, the evolutionary process of the auxiliary population is divided into two stages: in the early stage, DE/current/best/1 is employed, and in the later stage, DE/rand/1 is used. During the initial phase of evolution, different population states are considered to select the optimal individuals, thereby accelerating the convergence of the auxiliary population.
3. Comparing PSCMO with seven state-of-the-art different types of CMOEAs in 47 test functions and 12 real-world problems, the results show that PSCMO has the most balanced and superior performance.

The structure of this article is organized as follows: The literature review is described in Section 2. Section 3 introduces the algorithm in detail. Section 4 provides the experimental settings. Section 5 presents the experimental results and analysis. Section 6 concludes this article and outlines future research directions.

2. Research and Review of Existing CMOAs

The core issues faced by CMOEAs can be attributed to diversity and convergence. In this section, we will introduce some classic or recently developed CMOEAs from different classifications.

2.1. An Optimization Model Based on Gradual Evolution

This model is the most commonly used model, which does not consider the process from the UPF to the CPF. Instead, it uses clever EAs and CHTs to obtain an optimized model with good performance while considering convergence and diversity [13].

This model incorporates several classic algorithms, such as NSGA-II [14], with the core idea primarily relying on CHTs like the constrained dominance principle (CDP) and non-dominated sorting. Based on this approach, NSGA-II-CDP was proposed, which combines the CDP with NSGA-II [15]. Similarly, Ma et al. developed a two-stage linking method (ToR), NSGA-II-ToR [16]. In the CDP, the relationships between feasible solutions, infeasible solutions, and combinations of feasible and infeasible solutions are addressed, with the weights of the constraint conditions effectively applied during environmental selection. This method has seen widespread use. In the ToR, a weighting strategy is employed to process the objectives, gradually reducing the target weight in the early stages to maintain population diversity. In 2022, Jiawei Yuan et al. introduced ICMA [17], which defines a cost-based distance to measure the crowding between points in the objective space, improving the exploration of regions near the origin, which traditional Minkowski distances fail to address. Individual evaluation indicators based on the cost distance and overall constraints were also designed, guiding the population to explore promising areas and preventing them from getting trapped in local regions. In the same year, Qinghua Gu et al. proposed an enhanced CDP in MOEA/D-ICDP [18], allowing the algorithm to search between constrained and unconstrained regions. It merges the constraint value, the Tchebycheff aggregation function value, and a dynamic parameter to evaluate individual fitness. In 2023, Ruwang Jiao et al. introduced MFO [19], a multi-form constrained multi-objective evolutionary framework that improves the population diversity and accelerates convergence by creating auxiliary tasks for the information exchange between the primary and auxiliary tasks. In 2024, Mingcheng Zuo et al. [20] applied machine learning to CMOEAs by using reinforcement learning methods to adaptively select offspring generation operators. Additionally, in 2024, Jingjing Fang et al. proposed FLI [21], a metric that considers both the fitness distance correlation and the current evolutionary generation. FLI then uses this metric to design a strategy for selecting different versions of evolutionary algorithms. These models have addressed the challenges in selection posed by constrained spaces through clever CHTs, but they cannot overcome the lack of diversity in later stages caused by unified CHTs. Consequently, optimization models based on multiple groups or stages have gained greater attention in recent years.

2.2. An Optimization Model Based on Multiple Populations

This model mainly balances the constraint space and target space through collaboration between multiple populations (mainly two populations), exploring feasible and hopeful infeasible domains, respectively.

In 2018, Jiahai Wang et al. proposed CCMOD [22], which combines JADE with multiple populations (M populations) to solve M objectives. Two distinct CHTs are employed for archiving—one for offspring generation and the other for solution storage. Notably, the preserved population does not directly influence the generation of offspring but impacts it indirectly through another archive. In 2021, Zhi Zhong Liu et al. introduced BiCo [23], which features a novel angle-based selection scheme for updating the archive population. Since 2022, various swarm algorithms, such as CMDEIPCM [24], MTCMO [10], DPCPRA [25], and CMOCSO [26], have been proposed, with further advancements expected by 2024. Jianlin Zhang et al. introduced TCT-CMOEA [27], an algorithm that integrates the penalty function method with the CDP to enhance the population diversity and preserve promising solutions. Additionally, an external archive is utilized to balance the objectives and constraints. In 2022, Kangjia Qiao et al. proposed CSEMT [28], which establishes multiple auxiliary tasks based on a single constraint condition and develops an auxiliary task priority model for identifying and retaining highly correlated tasks. Weiwei Zhang et al. introduced MACA [29], which incorporates two strategies: a guided feasible search strategy to approach the constrained Pareto solution set by evaluating the direction and a manifold learning-based exploration strategy to assess the diversity contributions between the CPS and UPS populations. This strategy encourages the population to escape from local optima. However, the use of multiple population evolutionary methods in these algorithms inevitably leads to significant computational resource consumption, potentially compromising the algorithmic convergence. Consequently, the key challenge of this model is how to effectively minimize the waste of computational resources while ensuring robust convergence.

2.3. An Optimization Model Based on Multiple Stages

The core idea of this model is to divide the evolutionary process into multiple stages (mainly dual populations) to balance the convergence and diversity.

In recent years, multi-stage optimization models have garnered increasing attention from researchers. These models often employ a single population for evolution, integrating different stages of evolutionary methods with the UPF to enhance both the convergence and diversity. In 2021, Haiping Ma et al. proposed MSCMO [30], an algorithm that initially considers only a few constraints and then adopts varying CHTs at different stages. In 2022, Ye Tian et al. introduced CMOEA-BS [31], which establishes a fitness evaluation model to assess the priority relationship between the objectives and constraints. In Stage A, the majority of the solutions are infeasible, and the goal is to prioritize the constraints for reaching feasible areas. In Stage B, most of the solutions are feasible to prioritize the diversification of the solutions along the feasible boundary. In 2023, Ruiqing Sun et al. proposed C3M [5], a multi-stage CMOEA that disregards the constraints in the first stage to solve the MOP, incorporates important constraints through a priority model in the second stage, and considers all of the constraints in the final stage. However, a challenge remains in accurately transitioning the population between stages. As a result, two-stage optimization models have gained more popularity in recent years. Fei Ming et al. proposed CMOQLMT [32], which integrates reinforcement learning into evolutionary multitasking and introduces an auxiliary task selection model based on Q-learning and deep-Q learning. During the learning phase, Q-Table or Q-Network is used to select the most appropriate auxiliary tasks, and during the evolutionary phase, only the most suitable tasks are used for knowledge transfer. In 2024, Yuhang Ma et al. introduced CLBKR [33], which features a learning stage in the early phase of the algorithm. They proposed a classification model for distinguishing between single-modal and multimodal test problems. In the later stages, a metric, KTR, is employed to guide the population evolution for a single modality, while a

CHT is applied to guide multimodal population evolution. Also in 2024, Xiaoyu Zhong et al. proposed CEMTFB [34], which, similarly to other models, establishes tasks for finding both the UPF and CPF. However, this model also includes a helper task aimed at exploring promising areas. A reverse haulage strategy was proposed to prevent better feasible regions from being overlooked. The algorithm operates in a two-stage co-evolutionary process: the forward stage solves the CPF and UPF, establishing a knowledge transfer model, while the later stage focuses on the helper task. Furthermore, Wenjuan Zhang et al. proposed MA-TSEA [35], a two-stage multi-archive CMOEA. In the first stage, the parent population stores M+1 non-dominated solutions of the target, while the formal archive stores feasible solutions, processed by two populations. In the second stage, the parent population serves as the auxiliary population, and the formal archive becomes the main population, focusing on approaching the CPF. Three distinct CHTs are used to select and preserve the population for the auxiliary population, the parent population, and the formal archive. Despite the advancements in these models, an ongoing challenge remains in correctly switching between stages. Incorrect transitions can shift the direction of population evolution, making it difficult to effectively reach the CPF.

3. The Proposed Algorithm

3.1. The Framework

In solving CMOPs using CMOEAs, the population state often fluctuates, leading to challenges related to convergence, diversity, and feasibility. To address these issues effectively, it is crucial to focus on the globally optimal feasible domain. Thus, understanding the current state of the population is key during the optimization process. This article introduces PSCMO, which dynamically adjusts the evolutionary strategies based on the population's status. It aims to enhance exploration in infeasible regions while guiding the population toward the globally optimal feasible domain. Specifically, the algorithm records the positional relationship between the main and auxiliary populations in real time, analyzing the population's state based on historical data. Different CHTs are applied to selecting the auxiliary population depending on the population state, with the operators adjusted accordingly. To minimize resource waste in infeasible regions, a simple yet effective resource allocation model is proposed for optimizing the population size, reducing the function evaluations during the evolutionary process. Figure 1 clearly illustrates the process of PSCMO. The pseudocode for PSCMO is presented in Algorithm 1.

Algorithm 1 The evolutionary rocess of PSCMO

Input: *NP*: population size; *MaxFES*: maximal number of fitness evaluations
Output: The feasible Pareto-optimal solutions

1. P_1 ←Initialize NP individuals;
2. P_2 ←Initialize NP individuals;
3. $N_1 = NP/2$;
4. $N_2 = NP/2$;
5. *State* = 1;
6. ***while*** $MaxFES \leq FES$
7. O ← Generate offspring through Algorithm 2;
8. P_1 ← Select the main population through the CDP in O and P_1;
9. P_2 ← Select the auxiliary population through Algorithm 3 in O and P_2;
10. *State* ←Calculate the population state based on Equations (5)–(8);
11. [N_1,N_2] ← Calculate the number of offspring based on Equations (11) and (12);
12. ***end***

Algorithm 2 Search algorithm

Input: N_1: Number of offspring in the main population; N_2: number of offspring in the auxiliary population; P_1: main population; P_2: auxiliary population; *state*: current state of evolution; *FES*: number of function evaluations; *MaxFES*: maximum number of function evaluations.
Output: O: Offspring.
/* The main population's offspring's evolution */

$SD \longleftarrow$ Standardize the objective values and constraint violation values to obtain a matrix of size $NP \times (M + L)$;
$d \longleftarrow$ Calculate the distance of each individual in SD to the origin;
$s \longleftarrow$ Calculate the average cosine value of each individual relative to that of the others in the objective space;
$P_{best} \longleftarrow$ Perform fast non-dominated sorting on d and s, select the optimal $0.1 \times NP$ individuals, and generate $N1$ copies of these individuals;
$[P_{r1}, P_{r2}, P_{r3}] \longleftarrow$ Randomly shuffle the population to obtain three random populations composed of the main population, with each containing $N1$ individuals;
$O_1 \longleftarrow$ Execute *DE/current/test/1*;
$O_1 \longleftarrow$ Execute polynomial mutation;
/* The auxiliary population's offspring's evolution */
if $\frac{FES}{MaxFES} < \frac{1}{2}$
  $SD \longleftarrow$ Standardize the objective values to obtain a matrix of size $NP \times M$;
  $d \longleftarrow$ Calculate the distance of each individual in SD to the origin;
  if state == 1:
    $P_{best} \longleftarrow$ Select the top $0.1 \times NP$ individuals with the smallest d and generate $N2$ copies of these individuals;
  else
    $s \longleftarrow$Calculate the average cosine value of each individual relative to that of the others in the objective space;
    $P_{best} \longleftarrow$ Perform fast non-dominated sorting on d and s, select the optimal $0.1 \times NP$ individuals, and generate $N2$ copies of these individuals;
  end
  $[P_{r1}, P_{r2}, P_{r3}] \longleftarrow$ Randomly shuffle the population to obtain three random populations composed of auxiliary populations, with each containing $N2$ individuals;
  $O_2 \longleftarrow$ Execute *DE/current/test/1*;
else
  $[P_{r1}, P_{r2}, P_{r3}] \longleftarrow$ Randomly shuffle the population to obtain three random populations composed of auxiliary populations, with each containing $N2$ individuals;
  $O_2 \longleftarrow$ Execute *DE/rand/1*;
end
$O_2 \longleftarrow$ Execute polynomial mutation;
$O \leftarrow O_1 \cup O_2$.

Algorithm 3 Environmental selection model for the auxiliary population

Input: *NP*: Population size; P_2: auxiliary population; O: offspring; *state*: the current state of evolution.
Output: P_2: The auxiliary population.

$Pop \leftarrow P_2 \cup O$;
if $state == 1$
  $StandardObjs \leftarrow$ Standardize the objective values of Pop in the objective space;
  $d_1 \leftarrow$ Calculate the Euclidean distance from each individual to the origin using $StandardObjs$;
  $P_2 \leftarrow$ Sort d_1 and select the smallest NP individuals;
elseif $state == 2$
  $StandardObjs$ Standardize the objective values of Pop in the objective space;
  $d_2 \leftarrow$ Calculate the Euclidean distance from each individual to the origin using $StandardObjs$;
  $s \leftarrow$ Calculate the cosine value between each individual in Pop and other individuals, and then take the average;
  $P_2 \leftarrow$ Perform fast non-dominated sorting on d_2 and s to select the optimal NP individuals;
else
  $d_3 \leftarrow$ Calculate the average Euclidean distance between each individual in Pop and P_1;
  $s \leftarrow$ Calculate the cosine value between each individual in Pop and other individuals, and then take the average;
  $P_2 \leftarrow$ Perform fast non-dominated sorting on d_3 and s to select the optimal NP individuals;
end

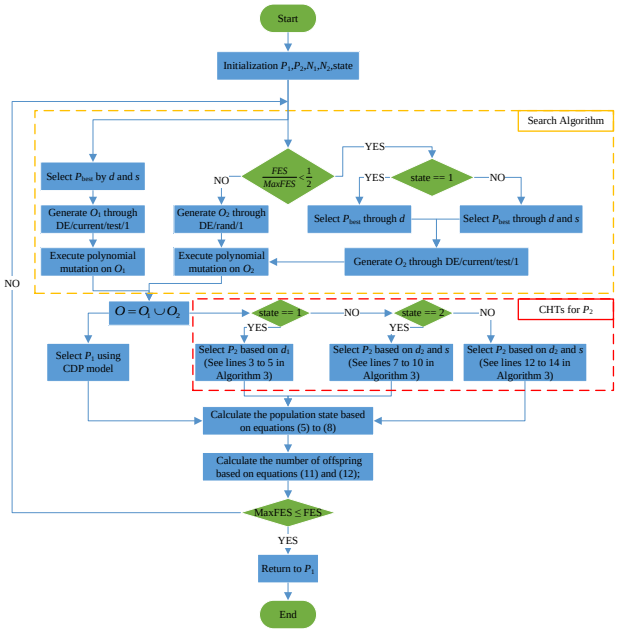


Figure 1. PSCMO flow chart.

3.2. The Population State Discrimination Model

In solving CMOPs, the constraints significantly alter the topological structure of the optimization space. Unlike MOPs, CMOPs introduce complex feasible domain distributions into both the decision and objective spaces due to the presence of infeasible regions. Specifically, the interaction between equality and inequality constraints divides the continuous optimization space into multiple isolated feasible subdomains. This fragmentation can result in a disjointed PF. Such structural characteristics present a dual challenge for evolutionary algorithms: First, the discrete feasible domains hinder effective information exchange between individuals in the population; second, the "barrier effect" caused by infeasible regions can trap populations in local feasible areas, limiting global exploration. To address this issue, this paper proposes determining the population's state by analyzing the positional relationship between the main and auxiliary populations. This approach further enhances the algorithm's performance from the perspective of the population state. Specifically, we identify three distinct states based on the population's configuration:

1. When the distance between the auxiliary population and the main population remains relatively stable, both the auxiliary population and the main population can be considered to have been effectively explored, and the auxiliary population can effectively guide the population to converge.
2. When the auxiliary population is close to the main population, it is considered that there is not much room for improvement in the convergence, and diversity should be given more consideration.
3. When the closest distance occurs between the auxiliary population and the main population, this indicates that the auxiliary population cannot help the population achieve further improvement in convergence. At this time, the main task of the auxiliary population should be to help the main population improve in its diversity.

From this, the following methods can be used to determine these three states of the population:

$$state(t) = \begin{cases} 2, & \overline{\Delta d}(t) < \overline{\Delta d}_{Q1}; \\ 3, & \Delta d(t) < \min\limits_{1 \le i \le t-1} \{\Delta d(i)\}; \\ 1, & \text{otherwise.} \end{cases} \tag{5}$$

$$d(t) = \frac{1}{NP^2} \sum_{j=1}^{NP} \sum_{i=1}^{NP} \left\| {P_1}^i(t) - {P_2}^j(t) \right\|_2 \tag{6}$$

$$\Delta d(t) = |d(t) - d(t-1)| \tag{7}$$

$$\overline{\Delta d}(t) = \frac{1}{t-1} \sum_{i=1}^{t-1} d(i) \tag{8}$$

where ${P_1}^i(t)$ represents the i-th individual of the main population in the t-th iteration, and ${P_2}^j(t)$ represents the j-th individual of the auxiliary population in the j-th iteration; $d(t)$ represents the average Euclidean distance between the main population and the auxiliary population; $\Delta d(t)$ represents the amount of change between two iterations, and D represents the average amount of change in the past t iterations at the t-th iteration; and $\overline{\Delta d}_{Q1}$ represents the quartiles of $\overline{\Delta d}$.

In the above model, first, $\overline{\Delta d}$ and $\overline{\Delta d}_{Q1}$ are used to determine whether the population belongs to the second state. If not, $\Delta d(t)$ and $\min_{1 \le i \le t-1}\{\Delta d(i)\}$ are used to check whether the population belongs to the third state. If neither condition is met, the population is determined to be in the first state. Therefore, it can be seen that by using two population position statistics, the state of the population can be effectively judged.

3.3. The Environmental Selection Model

A key distinction between CMOPs and MOPs is the need to address constraint violations in CMOPs. Solutions that violate constraints are referred to as infeasible solutions, which are typically not desired in the final solution set. While it is possible to accept solutions with poor convergence or diversity, the feasibility of the solutions is a strict requirement. However, in solving CMOPs, completely discarding infeasible solutions can hinder the ability to identify the global optimal solution. This is where the dual-population model proves beneficial for handling CMOPs. This model prioritizes the solution feasibility in the main population, while the auxiliary population leverages infeasible solutions to enhance the convergence and diversity. In this article, the main population employs the CDP for environmental selection, and the auxiliary population adjusts its selection method in real time based on the population state, thereby providing more effective assistance to the main population's evolution.

Deb et al. [15] proposed using the CDP to address the issue of infeasible solutions in solving CMOPs based on the non-dominated solution-solving process in NSGA-II. Due to its simplicity and effectiveness, the CDP model is widely used in various CMOEAs. Given two solutions x and y, if x constrains y (denoted as), then x and y follow the following principles:

1. $\varphi(x) < \varphi(y)$;
2. $\varphi(x) = \varphi(y), x \prec y$;

where $\varphi(x)$ and $\varphi(y)$ represent the constraint violation values of the two solutions. $x \prec y$ represents x Pareto dominance over y. If the constraint violations of the two solutions are the same, $x \prec y$ should satisfy

1. $\forall i \in \{1, 2, ..., m\}, f_i(x) \le f_i(y)$;
2. $\exists i \in \{1, 2, ..., m\}, f_i(x) < f_i(y)$.

To enable the auxiliary population to effectively assist the main population in overcoming obstacles in infeasible regions and enhance the diversity and convergence of the solutions, this article proposes three distinct environmental selection models for the auxiliary population. These models are adjusted in real time based on the population state. In the first state, where there is a stable distance relationship between the auxiliary and

main populations, the auxiliary population can effectively guide the main population's evolution. Thus, in this state, the auxiliary population focuses solely on the convergence in the target space, selecting individuals closest to the origin in the center of the target. These individuals converge most rapidly (see lines 3 to 5 in Algorithm 3). In the second state, the distance between the auxiliary population and the main population decreases, reducing the space for convergence improvements. Therefore, diversity becomes an important consideration. In this state, cosine values are incorporated to regulate the diversity of the auxiliary population (see lines 7 to 10 in Algorithm 3). By factoring in cosine values, the population's approach to the origin is slowed while maintaining a focus on the dispersion between individuals. This increase in diversity for the auxiliary population also helps improve the diversity of the main population. In the third state, when the auxiliary population cannot significantly improve the main population's convergence, the auxiliary population should be pulled closer to the main population to avoid unnecessary ineffective evolution in the infeasible region. Therefore, in this state, we replace the distance to the origin with the distance between the individual and the main population, selecting individuals that maintain diversity but are closer to the main population. The environmental selection model for the auxiliary population is shown in Algorithm 3.

3.4. The Search Algorithm

Previous studies [36] have shown that DE/current/best/1 has a good optimization performance in solving multi-objective optimization problems. The most important feature of this version of DE is the use of the optimal individual to guide the evolution of the population. The DE/rand/1 version of the operator can better ensure the diversity of the population. Therefore, PSCMO chooses the following two versions of operators for generating offspring:

$$DE/current/best/1 : O = P_{r1} + F \times (P_{best} - P_{r1} + P_{r2} - P_{r3}) \tag{9}$$

$$DE/rand/1 : O = P_{r1} + F \times (P_{r2} - P_{r3}) \tag{10}$$

To achieve a balance between convergence and diversity in the evolution of the main population, this article proposes a redesign of the optimal individual selection process. For convergence, the Euclidean distance d, which measures the distance from an individual's point to the origin in the target and constraint spaces, is used as a metric. For diversity, the average cosine value s between each individual and all the other individuals in the target space is considered. Fast non-dominated sorting is then performed based on both d and s. The top 10% of individuals from this sorting are selected as the optimal individuals P_{best} for the main population, ensuring a balance between both convergence and diversity. In the evolution of auxiliary populations, different evolutionary strategies must be applied depending on the population's state. For a CMOEA, convergence is prioritized in the early stages, while diversity is emphasized in the later stages. This two-stage approach has been widely adopted by many researchers. Following this approach, we have integrated a two-stage method into the offspring evolution of the auxiliary population. In the early stage, the auxiliary population evolves using DE/current/best/1 to quickly identify the optimal feasible domain. In the later stage, DE/rand/1 is used to enhance the population's diversity. When selecting optimal individuals for the auxiliary population, we incorporate the population state (as detailed in Section 3.2) to avoid excessive evolution within infeasible domains. If the population is in the first state, convergence is the sole focus, and the distance between individuals and the origin in the objective space is the only evaluation criterion for P_{best}. In the second and third states, the selection process for P_{best} follows the same principles as those for the main population, with the key difference being that

the auxiliary population does not consider the solution feasibility. For the evolutionary operators mentioned, the F parameter settings are [0.6, 0.8, 1.0], and the CR parameter settings are [0.1, 0.2, 1.0], consistent with the settings in reference [36]. After the evolutionary operator is executed, polynomial mutation is applied, as is also described in reference [36] and thus will not be repeated here.

The main population balances convergence and diversity using the DE/current/best/1 operator and the parameters d and s. The d indicator incorporates constraints, thereby simultaneously considering the convergence, diversity, and feasibility, ensuring a fundamental guarantee of the solution's quality. By alternately switching between two different operators and population states, the auxiliary population enhances the algorithm's ability to escape local optima, thereby improving the overall performance.

3.5. The Population Resource Allocation Method

The dual-population model often leads to unnecessary computational resource waste during the evolution process. To address this, this paper introduces a population resource allocation model to optimize the resource usage. In CMOPs, the goal is to find a set of solutions, which is determined by the main population returned by the model proposed in this paper. For both the main and auxiliary populations, the greater the number of offspring selected by the main population during environmental selection, the higher the success rate of evolution. Consequently, efficient evolution can be considered a method that enhances the population's ability to converge. To ensure that the final results are both convergent and balanced, efficient evolutionary methods are used to produce fewer offspring in the early stages. This allows for sufficient computational resources in later stages to improve the solution quality. This method is simple yet effective, with its core idea being using the evolutionary success rates to regulate the number of individuals in the main and auxiliary populations, thereby enabling real-time adjustment of the population's resources. The success rate of population evolution and the number of subgenerations for each population are calculated as follows:

$$R_i = \frac{|O_i \in P_i|}{N_i} \tag{11}$$

$$N_i = \max\{1, NP - NP \times R_i\} \tag{12}$$

where $i \in \{1, 2\}$, O_1 represents the offspring produced by the main population, and O_2 represents the offspring produced by the auxiliary population. N_1 represents the number of offspring in the main population, and N_2 represents the number of offspring in the auxiliary population. In the initialization phase, the number of main and auxiliary populations is $NP/2$.

3.6. A Complexity Analysis

In the PSCMO algorithm, the complexity mainly comes from evolutionary operators, CDP models, and fast non-dominated sorting models. The complexity of the evolutionary operators in the selection of intermediate parents and the generation of offspring is $O(NP)$ and $O(D \times NP)$, respectively. The complexity of the CDP model is $O(M \times NP^2)$. The complexity of the fast non-dominated sorting model is $O(M \times NP^2)$. Therefore, the complexity of the entire PSDCMO algorithm is approximately $O(NP^2)$, which is the same as that of most CMOEAs.

4. The Experimental Setup

4.1. Benchmark Functions and Evaluation Indicators

To comprehensively evaluate the feasibility, diversity, and convergence of the PSCMO algorithm, four benchmark test function sets were selected for testing: CF [37], DAS-

CMOP [38], LIR-CMOP [39], and MW [40]. The CF set includes 10 test functions, DAS-CMOP includes 9 functions, and both LIR-CMOP and MW include 14 functions each. Additionally, to assess the effectiveness of PSCMO in addressing real-world problems, 12 representative problems were selected from RWMOP [41]. For clarity, these 12 real-world problems are numbered in Table 1, where their respective objective counts and dimensions are also detailed.

Table 1. Detailed information on real-world problems.

Name	Problem	M	D
RWMOP1	Multi-product Batch Plant	3	10
RWMOP2	Crash Energy Management for High-speed Train	2	6
RWMOP3	Haverly's Pooling Problem	2	9
RWMOP4	Reactor Network Design	2	6
RWMOP5	Process Synthesis Problem	2	2
RWMOP6	Two Reactor Problem	2	7
RWMOP7	Synchronous Optimal Pulse-Width Modulation of 3-level Inverters	2	25
RWMOP8	Synchronous Optimal Pulse-Width Modulation of 5-level Inverters	2	25
RWMOP9	Synchronous Optimal Pulse-Width Modulation of 7-level Inverters	3	25
RWMOP10	Synchronous Optimal Pulse-Width Modulation of 9-level Inverters	2	30
RWMOP11	Synchronous Optimal Pulse-Width Modulation of 11-level Inverters	2	30
RWMOP12	Synchronous Optimal Pulse-Width Modulation of 13-level Inverters	2	30

This study employs three evaluation metrics—Inverted Generational Distance Plus (IGD+) [42], Hypervolume (HV) [43], and the Feasible Solutions Ratio (FSR) [44]—to assess the feasibility, diversity, and convergence of the PSCMO algorithm. IGD+ measures the average distance between the solutions in the obtained set and the PF, with smaller IGD+ values indicating a better algorithmic performance. HV evaluates the volume enclosed by the solutions in the set and a reference point in the objective space, where larger HV values signify a superior performance. The FSR determines the proportion of feasible solutions in the solution set, with higher FSR values reflecting the better feasibility outcomes of the algorithm.

4.2. Comparing Algorithms and Parameter Settings

To demonstrate the performance of PSCMO, this study selects eight of the latest and diverse types of CMOEAs for comparison: BiCo [23], DPPPS [45], tDEACPBI [11], AGEMOEAII [46], DSPCMDE [47], CMOEA-MS [48], TSTI [49], and CTSEA [50]. The parameter settings for these comparison algorithms are consistent with those specified in their respective original publications. All comparative experiments were conducted using the PlateMO framework developed by Tian et al. [51]. The parameters in each CMOEA in our experiments were set as follows:

- Population size: $N = 100$;
- The number of times the algorithm runs independently: 30;
- The number of maximum function evaluations: 100,000;
- Crossover probability $p_c = 1$ and the distribution index $\eta_c = 20$;
- Mutation probability $p_m = 1/D$ and the distribution index $\eta_m = 20$;
- For the DE operator, $F = 0.5$ and $CR = 1.0$.

5. Experimental Analysis

Tables 2 and 3 summarize the statistical results of 30 independent repeated experiments for IGD+ and HV across 47 benchmark test functions and 12 real-world problems, respectively. The numbers in parentheses indicate the standard deviation of IGD+ or HV.

The results of the Wilcoxon rank-sum test are denoted as "+", "-", and "=". A "+" indicates that the comparative algorithm is significantly better than PSCMO, a "-" indicates it is significantly worse, and a "=" indicates no significant difference, with all at a significance level of 0.05. When the population fails to consistently produce feasible solutions in all independent repeated experiments, the IGD+ and HV values are marked as NaN, and the values in parentheses then represent the corresponding FSR.

Table 2. Experimental results on IGD+ of PSCMO and 7 comparative algorithms.

	BiCo	DPPPS	AGEMOEAII	DSPCMDE	CMOEA-MS	TSTI	CTSEA	PSCMO
CF1	3.91×10^{-2} (2.43×10^{-3})+	**5.40×10^{-3} (2.71×10^{-3})-**	4.41×10^{-2} (3.21×10^{-3})+	6.41×10^{-3} (8.20×10^{-4})-	1.01×10^{-1} (1.19×10^{-2})+	5.05×10^{-2} (2.58×10^{-3})+	2.14×10^{-2} (3.11×10^{-3})+	7.56×10^{-3} (6.19×10^{-4})
CF2	1.19×10^{-1} (3.60×10^{-2})+	5.49×10^{-2} (2.01×10^{-2})+	1.33×10^{-1} (2.92×10^{-2})+	5.79×10^{-2} (3.91×10^{-3})+	1.84×10^{-1} (6.37×10^{-2})+	1.37×10^{-1} (3.28×10^{-2})+	1.20×10^{-1} (2.69×10^{-2})+	**2.43×10^{-2} (8.68×10^{-3})**
CF3	3.77×10^{-1} (4.16×10^{-2})+	2.79×10^{-1} (2.97×10^{-2})+	4.70×10^{-1} (3.39×10^{-2})+	5.21×10^{-1} (1.55×10^{-2})+	4.28×10^{-1} (4.23×10^{-2})+	4.74×10^{-1} (3.10×10^{-2})+	3.61×10^{-1} (4.45×10^{-2})+	**8.02×10^{-2} (1.76×10^{-2})**
CF4	2.89×10^{-1} (6.18×10^{-2})+	2.07×10^{-1} (2.48×10^{-2})+	3.48×10^{-1} (4.85×10^{-2})+	4.49×10^{-1} (3.61×10^{-2})+	3.87×10^{-1} (3.96×10^{-2})+	3.63×10^{-1} (5.14×10^{-2})+	3.06×10^{-1} (7.07×10^{-2})+	**7.54×10^{-2} (7.86×10^{-3})**
CF5	4.18×10^{-1} (2.38×10^{-2})+	2.88×10^{-1} (4.46×10^{-2})+	4.16×10^{-1} (1.75×10^{-2})+	1.53 (0.00)+	4.20×10^{-1} (1.93×10^{-2})+	4.31×10^{-1} (2.62×10^{-2})+	4.22×10^{-1} (1.71×10^{-2})+	**2.37×10^{-1} (1.74×10^{-2})**
CF6	3.46×10^{-1} (2.03×10^{-2})+	1.76×10^{-1} (3.17×10^{-2})+	3.42×10^{-1} (4.84×10^{-2})+	2.57×10^{-1} (1.49×10^{-2})+	3.36×10^{-1} (3.01×10^{-2})+	3.22×10^{-1} (1.57×10^{-2})+	3.13×10^{-1} (2.46×10^{-2})+	**6.46×10^{-2} (4.12×10^{-3})**
CF7	4.72×10^{-1} (8.91×10^{-2})+	3.41×10^{-1} (8.17×10^{-2})+	4.87×10^{-1} (8.78×10^{-2})+	1.40 (0.00)+	4.95×10^{-1} (7.43×10^{-2})+	4.97×10^{-1} (7.66×10^{-2})+	4.92×10^{-1} (6.63×10^{-2})+	**1.95×10^{-1} (2.58×10^{-2})**
CF8	NaN (0.00%)+	2.95×10^{-1} (2.64×10^{-2})+	NaN (0.00%)+	NaN (96.67%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (96.67%)+	**1.92×10^{-1} (1.74×10^{-2})**
CF9	5.16×10^{-1} (1.99×10^{-3})+	1.38×10^{-1} (1.21×10^{-2})+	6.03×10^{-1} (2.14×10^{-2})+	2.53×10^{-1} (1.87×10^{-2})+	1.54×10^{-1} (3.90×10^{-2})+	5.16×10^{-1} (3.18×10^{-6})+	1.35×10^{-1} (9.85×10^{-3})+	**6.74×10^{-2} (1.14×10^{-2})**
CF10	NaN (0.00%)+	6.61×10^{-1} (4.17×10^{-2})+	NaN (0.00%)+	NaN (3.33%)+	NaN (48.03%)+	NaN (0.00%)+	NaN (96.67%)+	**3.30×10^{-1} (4.80×10^{-2})**
+/-/=	10/0/0	9/1/0	10/0/0	9/1/0	10/0/0	10/0/0	10/0/0	
DAS-CMOP1	7.66×10^{-1} (5.63×10^{-4})+	6.15×10^{-1} (2.76×10^{-2})+	7.90×10^{-1} (0.00)+	2.97×10^{-3} (1.02×10^{-3})+	7.74×10^{-1} (0.00)+	7.22×10^{-1} (3.40×10^{-3})+	7.58×10^{-1} (8.42×10^{-4})+	**2.51×10^{-3} (9.73×10^{-4})**
DAS-CMOP2	1.61×10^{-1} (3.72×10^{-3})+	1.10×10^{-1} (7.21×10^{-3})+	1.74×10^{-1} (3.62×10^{-3})+	4.86×10^{-3} (1.54×10^{-4})+	1.70×10^{-1} (3.69×10^{-3})+	1.72×10^{-1} (3.14×10^{-3})+	1.52×10^{-1} (1.31×10^{-3})+	**3.58×10^{-3} (6.23×10^{-5})**
DAS-CMOP3	2.01×10^{-1} (1.91×10^{-4})+	1.90×10^{-1} (7.48×10^{-5})+	2.93×10^{-1} (6.46×10^{-2})+	7.65×10^{-3} (7.02×10^{-4})+	2.23×10^{-1} (1.01×10^{-3})+	2.10×10^{-1} (1.92×10^{-3})+	1.95×10^{-1} (6.80×10^{-4})+	**5.85×10^{-3} (4.36×10^{-4})**
DAS-CMOP4	3.79×10^{-1} (1.12×10^{-2})+	4.73×10^{-3} (4.75×10^{-3})+	3.37×10^{-1} (6.43×10^{-3})+	NaN (0.00%)+	2.64×10^{-1} (3.85×10^{-2})+	3.04×10^{-1} (1.46×10^{-2})+	2.57×10^{-3} (2.03×10^{-3})+	**1.83×10^{-3} (6.54×10^{-4})**
DAS-CMOP5	4.37×10^{-1} (4.54×10^{-2})+	5.16×10^{-3} (2.19×10^{-3})+	NaN (0.00%)+	NaN (0.00%)+	4.28×10^{-1} (1.95×10^{-2})+	2.82×10^{-1} (7.49×10^{-2})+	**2.39×10^{-3} (3.19×10^{-4})-**	3.12×10^{-3} (3.35×10^{-4})
DAS-CMOP6	NaN (90.03%)+	3.52×10^{-2} (1.17×10^{-2})+	NaN (0.00%)+	NaN (0.00%)+	6.34×10^{-1} (1.23×10^{-2})+	6.43×10^{-1} (1.14×10^{-2})+	5.48×10^{-2} (1.71×10^{-2})+	**6.15×10^{-3} (6.05×10^{-4})**
DAS-CMOP7	2.86×10^{-2} (5.02×10^{-3})+	4.13×10^{-2} (6.92×10^{-4})+	**2.34×10^{-2} (2.16×10^{-3})-**	NaN (0.00%)+	3.85×10^{-2} (2.12×10^{-2})=	2.50×10^{-2} (2.60×10^{-3})=	2.58×10^{-2} (4.83×10^{-4})+	2.41×10^{-2} (6.31×10^{-4})
DAS-CMOP8	2.07×10^{-2} (1.74×10^{-3})+	2.46×10^{-2} (9.26×10^{-4})+	2.05×10^{-2} (3.40×10^{-4})+	NaN (0.00%)+	2.60×10^{-2} (1.26×10^{-2})=	1.92×10^{-2} (2.71×10^{-3})-	1.90×10^{-2} (5.87×10^{-4})=	**1.90×10^{-2} (4.39×10^{-4})**
DAS-CMOP9	2.97×10^{-1} (4.48×10^{-3})+	2.13×10^{-1} (6.21×10^{-3})+	3.49×10^{-1} (8.61×10^{-3})+	2.89×10^{-2} (9.37×10^{-4})+	3.21×10^{-1} (9.79×10^{-3})+	2.55×10^{-1} (1.16×10^{-2})+	2.98×10^{-1} (7.08×10^{-3})+	**1.88×10^{-2} (5.73×10^{-4})**
+/-/=	9/0/0	9/0/0	8/1/0	9/0/0	7/0/2	7/1/1	7/1/1	
LIR-CMOP1	2.09×10^{-1} (3.34×10^{-3})+	**9.96×10^{-2} (1.24×10^{-2})=**	2.81×10^{-1} (7.45×10^{-3})+	1.75×10^{-1} (2.65×10^{-2})+	3.02×10^{-1} (1.50×10^{-2})+	2.14×10^{-1} (1.09×10^{-2})+	3.04×10^{-1} (1.41×10^{-2})+	1.03×10^{-1} (1.47×10^{-2})
LIR-CMOP2	1.30×10^{-1} (8.10×10^{-3})+	5.90×10^{-2} (1.07×10^{-2})+	1.94×10^{-1} (1.38×10^{-2})+	7.35×10^{-2} (1.57×10^{-2})+	2.09×10^{-1} (1.50×10^{-2})+	1.33×10^{-1} (1.43×10^{-2})+	2.04×10^{-1} (1.22×10^{-2})+	**4.56×10^{-2} (1.12×10^{-2})**
LIR-CMOP3	2.17×10^{-1} (4.20×10^{-3})+	**1.17×10^{-1} (8.00×10^{-3})-**	2.81×10^{-1} (6.62×10^{-3})+	1.98×10^{-1} (1.42×10^{-2})+	3.00×10^{-1} (6.21×10^{-3})+	2.25×10^{-1} (6.54×10^{-3})+	2.88×10^{-1} (9.60×10^{-3})+	1.26×10^{-1} (1.10×10^{-2})
LIR-CMOP4	1.62×10^{-1} (9.15×10^{-3})+	9.41×10^{-2} (1.04×10^{-2})+	2.16×10^{-1} (1.16×10^{-2})+	1.30×10^{-1} (2.13×10^{-2})+	2.19×10^{-1} (1.60×10^{-2})+	1.57×10^{-1} (7.62×10^{-3})+	2.19×10^{-1} (1.13×10^{-2})+	**7.91×10^{-2} (1.16×10^{-2})**
LIR-CMOP5	1.23 (0.00)+	7.08×10^{-2} (2.17×10^{-2})+	1.27 (0.00)+	1.25×10^{-1} (6.80×10^{-2})+	9.22×10^{-1} (5.20×10^{-2})+	1.23 (0.00)+	3.14×10^{-1} (8.36×10^{-3})+	**2.99×10^{-2} (2.84×10^{-3})**
LIR-CMOP6	1.35 (0.00)+	1.82×10^{-1} (1.03×10^{-2})+	1.35 (0.00)+	1.41×10^{-1} (2.63×10^{-2})+	8.52×10^{-1} (4.77×10^{-2})+	1.35 (0.00)+	3.30×10^{-1} (4.20×10^{-3})+	**5.51×10^{-2} (6.12×10^{-3})**
LIR-CMOP7	1.68 (0.00)+	1.18×10^{-1} (3.46×10^{-3})+	1.68 (0.00)+	2.03×10^{-2} (1.29×10^{-2})+	1.47×10^{-1} (9.59×10^{-3})+	2.23×10^{-1} (4.51×10^{-2})+	1.37×10^{-1} (4.32×10^{-3})+	**8.44×10^{-3} (6.96×10^{-4})**
LIR-CMOP8	1.68 (0.00)+	1.85×10^{-1} (3.56×10^{-3})+	1.68 (0.00)+	2.09×10^{-2} (1.24×10^{-2})+	2.36×10^{-1} (8.68×10^{-3})+	5.54×10^{-1} (8.78×10^{-2})+	2.41×10^{-1} (2.68×10^{-3})+	**6.57×10^{-3} (2.24×10^{-4})**
LIR-CMOP9	1.04 (3.82×10^{-2})+	3.16×10^{-1} (5.54×10^{-2})=	1.07 (3.01×10^{-3})+	3.56×10^{-1} (1.98×10^{-2})+	7.38×10^{-1} (4.87×10^{-2})+	5.47×10^{-1} (2.39×10^{-2})+	7.52×10^{-1} (3.43×10^{-2})+	**2.44×10^{-1} (2.26×10^{-3})**

Table 2. *Cont.*

	BiCo	DPPPS	AGEMOEAII	DSPCMDE	CMOEA-MS	TSTI	CTSEA	PSCMO
LIR-CMOP10	1.03 (2.56×10^{-4})+	3.17×10^{-1} (6.36×10^{-3})+	9.90×10^{-1} (1.91×10^{-2})+	2.02×10^{-1} (2.34×10^{-2})+	7.04×10^{-1} (7.17×10^{-2})+	9.56×10^{-1} (4.30×10^{-3})+	5.20×10^{-1} (2.81×10^{-2})+	**1.62×10^{-1}** **(1.18×10^{-2})**
LIR-CMOP11	9.85×10^{-1} (1.37×10^{-2})+	1.23×10^{-1} (2.62×10^{-2})=	9.15×10^{-1} (1.51×10^{-2})+	2.11×10^{-1} (4.93×10^{-2})+	5.76×10^{-1} (5.58×10^{-2})+	7.51×10^{-1} (1.87×10^{-3})+	4.02×10^{-1} (3.20×10^{-2})+	**1.20×10^{-1}** **(5.49×10^{-3})**
LIR-CMOP12	9.57×10^{-1} (1.47×10^{-2})+	1.81×10^{-1} (1.57×10^{-2})+	9.66×10^{-1} (1.37×10^{-2})+	2.94×10^{-1} (5.78×10^{-2})+	4.45×10^{-1} (5.11×10^{-2})+	3.70×10^{-1} (7.55×10^{-2})+	4.74×10^{-1} (3.17×10^{-2})+	**1.48×10^{-1}** **(1.80×10^{-3})**
LIR-CMOP13	1.32 (1.02×10^{-4})+	5.55×10^{-2} (2.65×10^{-3})-	1.31 (8.26×10^{-5})+	8.74×10^{-2} (2.81×10^{-3})+	4.86×10^{-2} (2.32×10^{-2})-	1.32 (1.28×10^{-4})+	**4.49×10^{-2}** **(1.13×10^{-3})-**	6.03×10^{-2} (1.57×10^{-3})
LIR-CMOP14	1.28 (3.20×10^{-4})+	5.26×10^{-2} (2.11×10^{-3})+	1.26 (7.63×10^{-5})+	6.30×10^{-2} (3.32×10^{-3})+	**4.57×10^{-2}** **(1.37×10^{-3})-**	1.27 (3.16×10^{-4})+	4.59×10^{-2} (1.54×10^{-3})-	4.74×10^{-2} (1.48×10^{-3})
+/-/=	14/0/0	9/2/3	14/0/0	14/0/0	12/2/0	14/0/0	12/2/0	
MW1	NaN (0.00%)+	NaN (100.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	**2.02×10^{-3}** **(1.82×10^{-4})**
MW2	**4.24×10^{-2}** **(1.82×10^{-2})-**	9.95×10^{-2} (5.48×10^{-2})+	1.31×10^{-1} (6.04×10^{-2})+	NaN (76.67%)+	1.09×10^{-1} (4.36×10^{-2})+	1.06×10^{-1} (4.45×10^{-2})+	7.39×10^{-2} (3.91×10^{-2})+	5.70×10^{-2} (1.15×10^{-2})
MW3	7.57×10^{-3} (1.28×10^{-3})+	7.86×10^{-3} (7.51×10^{-4})+	7.50×10^{-1} (0.00)+	4.84×10^{-3} (3.54×10^{-4})=	1.12×10^{-1} (1.85×10^{-1})+	1.24×10^{-1} (1.27×10^{-1})+	5.95×10^{-3} (1.44×10^{-3})+	**4.79×10^{-3}** **(2.73×10^{-4})**
MW4	NaN (0.00%)+	6.38×10^{-2} (1.35×10^{-1})+	NaN (0.00%)+	NaN (90.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	**4.05×10^{-2}** **(6.12×10^{-4})**
MW5	NaN (0.00%)+	5.88×10^{-2} (8.81×10^{-2})+	NaN (0.00%)+	NaN (36.67%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	**3.25×10^{-3}** **(4.37×10^{-4})**
MW6	**4.74×10^{-2}** **(1.82×10^{-2})-**	1.49×10^{-1} (5.45×10^{-2})-	5.52×10^{-1} (3.98×10^{-2})+	NaN (86.67%)+	4.89×10^{-1} (2.96×10^{-2})+	5.52×10^{-1} (4.09×10^{-2})+	2.02×10^{-1} (6.96×10^{-2})=	1.65×10^{-1} (2.99×10^{-2})
MW7	5.63×10^{-3} (1.40×10^{-3})+	5.06×10^{-3} (8.37×10^{-4})+	3.06×10^{-1} (7.63×10^{-4})+	**2.51×10^{-3}** **(3.21×10^{-4})-**	9.15×10^{-3} (1.74×10^{-3})+	9.38×10^{-3} (1.67×10^{-3})+	3.23×10^{-3} (8.87×10^{-4})-	3.82×10^{-3} (8.63×10^{-4})
MW8	**5.44×10^{-2}** **(3.50×10^{-2})-**	6.15×10^{-2} (2.88×10^{-2})-	NaN (93.33%)+	NaN (0.00%)+	NaN (93.33%)+	NaN (96.67%)+	8.97×10^{-2} (5.92×10^{-2})=	7.58×10^{-2} (1.94×10^{-2})
MW9	NaN (0.00%)+	6.75×10^{-1} (0.00)+	NaN (0.00%)+	NaN (90.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	**3.30×10^{-3}** **(1.39×10^{-3})**
MW10	**1.69×10^{-1}** **(4.62×10^{-2})-**	2.06×10^{-1} (6.37×10^{-2})=	NaN (0.00%)+	NaN (0.00%)+	NaN (20.43%)+	NaN (40.00%)+	NaN (86.67%)+	2.00×10^{-1} (2.52×10^{-2})
MW11	1.11×10^{-1} (7.12×10^{-2})+	3.29×10^{-3} (2.64×10^{-4})+	5.36×10^{-1} (2.02×10^{-4})+	NaN (0.00%)+	1.97×10^{-1} (7.93×10^{-2})+	2.45×10^{-1} (8.51×10^{-2})+	2.89×10^{-3} (2.91×10^{-4})+	**2.78×10^{-3}** **(2.83×10^{-4})**
MW12	NaN (73.50%)+	7.48×10^{-1} (0.00)+	NaN (0.00%)+	NaN (96.67%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	**2.92×10^{-3}** **(2.20×10^{-4})**
MW13	**9.31×10^{-1}** **(1.80×10^{-2})-**	1.29×10^{-1} (4.03×10^{-2})-	9.96×10^{-1} (5.03×10^{-2})+	7.24×10^{-1} (4.83×10^{-2})+	2.90×10^{-1} (5.14×10^{-2})+	3.98×10^{-1} (9.21×10^{-2})+	1.42×10^{-1} (4.80×10^{-2})-	1.47×10^{-1} (1.19×10^{-2})
MW14	2.94×10^{-1} (2.50×10^{-2})+	5.87×10^{-1} (1.53×10^{-2})+	1.49×10^{-1} (2.05×10^{-2})+	5.65×10^{-1} (9.26×10^{-3})+	2.36×10^{-1} (2.99×10^{-2})+	1.71×10^{-1} (2.62×10^{-2})+	1.99×10^{-1} (4.14×10^{-2})+	**1.24×10^{-1}** **(7.47×10^{-3})**
+/-/=	9/5/0	10/3/1	14/0/0	12/1/1	14/0/0	14/0/0	10/2/2	
RWMOP1	9.81×10^{4} (1.03×10^{4})+	7.24×10^{4} (1.09×10^{4})+	1.24×10^{5} (3.96×10^{3})+	NaN (0.00%)+	1.18×10^{5} (8.90×10^{3})+	9.37×10^{4} (1.16×10^{4})+	1.15×10^{5} (1.02×10^{4})+	**4.51×10^{4}** **(6.50×10^{3})**
RWMOP2	2.05×10^{-2} (1.59×10^{-2})+	1.61×10^{-2} (8.92×10^{-5})+	1.61×10^{-2} (1.92e-13)-	1.61×10^{-2} (7.06e-18)+	1.61×10^{-2} (7.06e-18)+	2.26×10^{-2} (3.60×10^{-2})+	1.61×10^{-2} (7.06e-18)+	**1.61×10^{-2}** **(1.17e-08)**
RWMOP3	NaN (0.00%)+	1.49×10^{3} (2.24×10^{2})+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	**1.28×10^{3}** **(2.60)**
RWMOP4	NaN (96.67%)+	3.29 (4.87×10^{-2})-	NaN (66.67%)+	NaN (0.00%)+	**2.75** **(4.96×10^{-1})-**	NaN (96.67%)+	NaN (36.67%)+	3.32 (4.83×10^{-2})
RWMOP5	7.44×10^{-1} (2.22×10^{-5})+	7.44×10^{-1} (1.32×10^{-5})+	7.44×10^{-1} (5.24×10^{-6})+	7.44×10^{-1} (6.76×10^{-5})+	7.44×10^{-1} (1.38×10^{-5})+	7.44×10^{-1} (9.99×10^{-6})+	7.44×10^{-1} (1.50×10^{-5})+	**7.44×10^{-1}** **(3.83×10^{-7})**
RWMOP6	NaN (0.00%)+	3.42×10^{1} (5.44×10^{-4})+	NaN (0.00%)+	NaN (0.00%)+	NaN (90.00%)+	3.10×10^{1} (1.99)+	NaN (0.00%)+	**7.30** **(3.34)**
RWMOP7	NaN (0.00%)+	1.82×10^{-1} (6.71×10^{-2})+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	**1.07×10^{-1}** **(5.09×10^{-3})**
RWMOP8	NaN (0.00%)+	4.12×10^{-1} (1.62×10^{-1})+	NaN (0.00%)+	NaN (10.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	**6.28×10^{-2}** **(2.48×10^{-2})**
RWMOP9	NaN (0.00%)+	9.56×10^{-1} (6.22×10^{-1})+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	**1.12×10^{-1}** **(2.55×10^{-2})**
RWMOP10	NaN (0.00%)+	3.78 (1.79)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	**8.15×10^{-1}** **(4.81×10^{-1})**
RWMOP11	NaN (0.00%)+	3.00 (1.14)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	**1.17** **(6.47×10^{-1})**
RWMOP12	NaN (0.00%)+	**3.36** **(1.43)+**	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	4.02 (1.04)
+/-/=	12/0/0	11/1/0	11/1/0	12/0/0	11/1/0	12/0/0	12/0/0	

Table 3. Experimental results on HV of PSCMO and 7 comparative algorithms.

	BiCo	DPPPS	AGEMOEAII	DSPCMDE	CMOEA-MS	TSTI	CTSEA	PSCMO
CF1	5.17×10^{-1} (2.43×10^{-3})+	**5.59×10^{-1} (2.71×10^{-3})-**	5.11×10^{-1} (3.21×10^{-3})+	5.58×10^{-1} (8.20×10^{-4})-	4.44×10^{-1} (1.19×10^{-2})+	5.03×10^{-1} (2.58×10^{-3})+	5.38×10^{-1} (3.11×10^{-3})+	5.57×10^{-1} (6.19×10^{-4})
CF2	4.98×10^{-1} (3.60×10^{-2})+	6.01×10^{-1} (2.01×10^{-2})+	4.99×10^{-1} (2.92×10^{-2})+	5.94×10^{-1} (3.91×10^{-3})+	4.59×10^{-1} (6.37×10^{-2})+	4.87×10^{-1} (3.28×10^{-2})+	5.07×10^{-1} (2.69×10^{-2})+	**6.35×10^{-1} (8.68×10^{-3})**
CF3	1.47×10^{-1} (4.16×10^{-2})+	1.97×10^{-1} (2.97×10^{-2})+	1.08×10^{-1} (3.39×10^{-2})+	8.42×10^{-2} (1.55×10^{-2})+	1.19×10^{-1} (4.23×10^{-2})+	1.02×10^{-1} (3.10×10^{-2})+	1.40×10^{-1} (4.45×10^{-2})+	**2.72×10^{-1} (1.76×10^{-2})**
CF4	2.31×10^{-1} (6.18×10^{-2})+	2.98×10^{-1} (2.48×10^{-2})+	1.81×10^{-1} (4.85×10^{-2})+	9.49×10^{-2} (3.61×10^{-2})+	1.49×10^{-1} (3.96×10^{-2})+	1.73×10^{-1} (5.14×10^{-2})+	2.17×10^{-1} (7.07×10^{-2})+	**4.19×10^{-1} (7.86×10^{-3})**
CF5	1.20×10^{-1} (2.38×10^{-2})+	2.29×10^{-1} (4.46×10^{-2})+	1.22×10^{-1} (1.75×10^{-2})+	0.00 (0.00)+	1.18×10^{-1} (1.93×10^{-2})+	1.09×10^{-1} (2.62×10^{-2})+	1.17×10^{-1} (1.71×10^{-2})+	**2.68×10^{-1} (1.74×10^{-2})**
CF6	3.59×10^{-1} (2.03×10^{-2})+	5.04×10^{-1} (3.17×10^{-2})+	3.59×10^{-1} (4.84×10^{-2})+	4.42×10^{-1} (1.49×10^{-2})+	3.65×10^{-1} (3.01×10^{-2})+	3.75×10^{-1} (1.57×10^{-2})+	3.84×10^{-1} (2.46×10^{-2})+	**6.02×10^{-1} (4.12×10^{-3})**
CF7	1.96×10^{-1} (8.91×10^{-2})+	3.01×10^{-1} (8.17×10^{-2})+	2.04×10^{-1} (8.78×10^{-2})+	0.00 (0.00)+	1.90×10^{-1} (7.43×10^{-2})+	1.96×10^{-1} (7.66×10^{-2})+	1.52×10^{-1} (6.63×10^{-2})+	**4.07×10^{-1} (2.58×10^{-2})**
CF8	NaN (0.00%)+	1.70×10^{-1} (2.64×10^{-2})+	NaN (0.00%)+	NaN (96.67%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (96.67%)+	**2.53×10^{-1} (1.74×10^{-2})**
CF9	9.13×10^{-2} (1.99×10^{-3})+	2.65×10^{-1} (1.21×10^{-2})+	8.53×10^{-2} (2.14×10^{-2})+	1.77×10^{-1} (1.87×10^{-2})+	2.83×10^{-1} (3.90×10^{-2})+	9.09×10^{-2} (3.18×10^{-6})+	2.69×10^{-1} (9.85×10^{-3})+	**4.01×10^{-1} (1.14×10^{-2})**
CF10	NaN (0.00%)+	4.34×10^{-2} (4.17×10^{-2})+	NaN (0.00%)+	NaN (3.33%)+	NaN (48.03%)+	NaN (0.00%)+	NaN (96.67%)+	**1.05×10^{-1} (4.80×10^{-2})**
+/-/=	10/0/0	9/1/0	10/0/0	9/1/0	10/0/0	10/0/0	10/0/0	
DAS-CMOP1	3.98×10^{-4} (5.63×10^{-4})+	3.30×10^{-2} (2.76×10^{-2})+	0.00 (0.00)+	**2.10×10^{-1} (1.02×10^{-3})=**	0.00 (0.00)+	7.58×10^{-3} (3.40×10^{-3})+	6.30×10^{-4} (8.42×10^{-4})+	2.10×10^{-1} (9.73×10^{-4})
DAS-CMOP2	2.51×10^{-1} (3.72×10^{-3})+	2.82×10^{-1} (7.21×10^{-3})+	2.46×10^{-1} (3.62×10^{-3})+	3.54×10^{-1} (1.54×10^{-4})+	2.53×10^{-1} (3.69×10^{-3})+	2.57×10^{-1} (3.14×10^{-3})+	2.60×10^{-1} (1.31×10^{-3})+	**3.55×10^{-1} (6.23×10^{-5})**
DAS-CMOP3	2.08×10^{-1} (1.91×10^{-4})+	2.08×10^{-1} (7.48×10^{-5})+	1.83×10^{-1} (6.46×10^{-2})+	3.10×10^{-1} (7.02×10^{-4})+	2.08×10^{-1} (1.01×10^{-3})+	2.09×10^{-1} (1.92×10^{-3})+	2.07×10^{-1} (6.80×10^{-4})+	**3.11×10^{-1} (4.36×10^{-4})**
DAS-CMOP4	5.93×10^{-2} (1.12×10^{-2})+	1.95×10^{-1} (4.75×10^{-3})+	6.92×10^{-2} (6.43×10^{-3})+	NaN (0.00%)+	1.12×10^{-1} (3.85×10^{-2})+	8.13×10^{-2} (1.46×10^{-2})+	1.96×10^{-1} (2.03×10^{-3})+	**2.02×10^{-1} (6.54×10^{-4})**
DAS-CMOP5	9.98×10^{-2} (4.54×10^{-2})+	3.48×10^{-1} (2.19×10^{-3})+	NaN (0.00%)+	NaN (0.00%)+	9.89×10^{-2} (1.95×10^{-2})+	1.72×10^{-1} (7.49×10^{-2})+	**3.51×10^{-1} (3.19×10^{-4})-**	3.50×10^{-1} (3.35×10^{-4})
DAS-CMOP6	NaN (90.03%)+	2.93×10^{-1} (1.17×10^{-2})+	NaN (0.00%)+	NaN (0.00%)+	3.54×10^{-2} (1.23×10^{-2})+	3.21×10^{-2} (1.14×10^{-2})+	2.88×10^{-1} (1.71×10^{-2})+	**3.11×10^{-1} (6.05×10^{-4})**
DAS-CMOP7	2.84×10^{-1} (5.02×10^{-3})+	2.84×10^{-1} (6.92×10^{-4})+	**2.88×10^{-1} (2.16×10^{-3})-**	NaN (0.00%)+	2.82×10^{-1} (2.12×10^{-2})-	2.88×10^{-1} (2.60×10^{-3})-	2.87×10^{-1} (4.83×10^{-4})-	2.87×10^{-1} (6.31×10^{-4})
DAS-CMOP8	2.06×10^{-1} (1.74×10^{-3})+	2.04×10^{-1} (9.26×10^{-4})+	2.05×10^{-1} (3.40×10^{-4})+	NaN (0.00%)+	2.04×10^{-1} (1.26×10^{-2})-	**2.07×10^{-1} (2.71×10^{-3})-**	2.07×10^{-1} (5.87×10^{-4})-	2.06×10^{-1} (4.39×10^{-4})
DAS-CMOP9	1.13×10^{-1} (4.48×10^{-3})+	1.39×10^{-1} (6.21×10^{-3})+	9.86×10^{-2} (8.61×10^{-3})+	2.01×10^{-1} (9.37×10^{-4})+	1.06×10^{-1} (9.79×10^{-3})+	1.15×10^{-1} (1.16×10^{-2})+	1.13×10^{-1} (7.08×10^{-3})+	**2.06×10^{-1} (5.73×10^{-4})**
+/-/=	9/0/0	9/0/0	8/1/0	8/0/1	7/2/0	7/2/0	6/3/0	
LIR-CMOP1	1.29×10^{-1} (3.34×10^{-3})+	1.77×10^{-1} (1.24×10^{-2})=	1.02×10^{-1} (7.45×10^{-3})+	1.63×10^{-1} (2.65×10^{-2})+	9.28×10^{-2} (1.50×10^{-2})+	1.30×10^{-1} (1.09×10^{-2})+	9.36×10^{-2} (1.41×10^{-2})+	**1.79×10^{-1} (1.47×10^{-2})**
LIR-CMOP2	2.49×10^{-1} (8.10×10^{-3})+	3.02×10^{-1} (1.07×10^{-2})+	2.11×10^{-1} (1.38×10^{-2})+	3.11×10^{-1} (1.57×10^{-2})+	2.04×10^{-1} (1.50×10^{-2})+	2.54×10^{-1} (1.43×10^{-2})+	2.06×10^{-1} (1.22×10^{-2})+	**3.21×10^{-1} (1.12×10^{-2})**
LIR-CMOP3	1.11×10^{-1} (4.20×10^{-3})+	**1.52×10^{-1} (8.00×10^{-3})-**	9.27×10^{-2} (6.62×10^{-3})+	1.41×10^{-1} (1.42×10^{-2})=	9.60×10^{-2} (6.21×10^{-3})+	1.10×10^{-1} (6.54×10^{-3})+	9.24×10^{-2} (9.60×10^{-3})+	1.46×10^{-1} (1.10×10^{-2})
LIR-CMOP4	2.06×10^{-1} (9.15×10^{-3})+	2.47×10^{-1} (1.04×10^{-2})+	1.76×10^{-1} (1.16×10^{-2})+	2.40×10^{-1} (2.13×10^{-2})+	1.77×10^{-1} (1.60×10^{-2})+	2.10×10^{-1} (7.62×10^{-3})+	1.74×10^{-1} (1.13×10^{-2})+	**2.62×10^{-1} (1.16×10^{-2})**
LIR-CMOP5	0.00 (0.00)+	2.36×10^{-1} (2.17×10^{-2})+	0.00 (0.00)+	2.50×10^{-1} (6.80×10^{-2})+	3.60×10^{-2} (5.20×10^{-2})+	0.00 (0.00)+	1.19×10^{-1} (8.36×10^{-3})+	**2.78×10^{-1} (2.84×10^{-3})**
LIR-CMOP6	0.00 (0.00)+	1.48×10^{-1} (1.03×10^{-2})+	0.00 (0.00)+	1.34×10^{-1} (2.63×10^{-2})+	4.69×10^{-2} (4.77×10^{-2})+	0.00 (0.00)+	9.84×10^{-2} (4.20×10^{-3})+	**1.69×10^{-1} (6.12×10^{-3})**
LIR-CMOP7	0.00 (0.00)+	2.41×10^{-1} (3.46×10^{-3})+	0.00 (0.00)+	2.87×10^{-1} (1.29×10^{-2})=	2.32×10^{-1} (9.59×10^{-3})+	2.16×10^{-1} (4.51×10^{-2})+	2.34×10^{-1} (4.32×10^{-3})+	**2.92×10^{-1} (6.96×10^{-4})**
LIR-CMOP8	0.00 (0.00)+	2.30×10^{-1} (3.56×10^{-3})+	0.00 (0.00)+	2.89×10^{-1} (1.24×10^{-2})+	2.18×10^{-1} (8.68×10^{-3})+	1.72×10^{-1} (8.78×10^{-2})+	2.21×10^{-1} (2.68×10^{-3})+	**2.94×10^{-1} (2.24×10^{-4})**
LIR-CMOP9	9.67×10^{-2} (3.82×10^{-2})+	3.81×10^{-1} (5.54×10^{-2})=	8.46×10^{-2} (3.01×10^{-3})+	3.12×10^{-1} (1.98×10^{-2})+	2.01×10^{-1} (4.87×10^{-2})+	2.37×10^{-1} (2.39×10^{-2})+	2.33×10^{-1} (3.43×10^{-2})+	**4.00×10^{-1} (2.26×10^{-3})**
LIR-CMOP10	5.27×10^{-2} (2.56×10^{-4})+	4.87×10^{-1} (6.36×10^{-3})+	5.95×10^{-2} (1.91×10^{-2})+	5.79×10^{-1} (2.34×10^{-2})+	1.67×10^{-1} (7.17×10^{-2})+	5.43×10^{-2} (4.30×10^{-3})+	2.55×10^{-1} (2.81×10^{-2})+	**6.09×10^{-1} (1.18×10^{-2})**
LIR-CMOP11	1.60×10^{-1} (1.37×10^{-2})+	**6.01×10^{-1} (2.62×10^{-2})=**	1.63×10^{-1} (1.51×10^{-2})+	5.33×10^{-1} (4.93×10^{-2})+	2.90×10^{-1} (5.58×10^{-2})+	2.27×10^{-1} (1.87×10^{-3})+	3.80×10^{-1} (3.20×10^{-2})+	5.99×10^{-1} (5.49×10^{-3})
LIR-CMOP12	1.82×10^{-1} (1.47×10^{-2})+	5.09×10^{-1} (1.57×10^{-2})+	1.86×10^{-1} (1.37×10^{-2})+	4.23×10^{-1} (5.78×10^{-2})+	3.97×10^{-1} (5.11×10^{-2})+	3.81×10^{-1} (7.55×10^{-2})+	4.23×10^{-1} (3.17×10^{-2})+	**5.30×10^{-1} (1.80×10^{-3})**
LIR-CMOP13	7.87×10^{-5} (1.02×10^{-4})+	5.43×10^{-1} (2.65×10^{-3})-	4.16×10^{-4} (8.26×10^{-5})+	5.03×10^{-1} (2.81×10^{-3})+	5.51×10^{-1} (2.32×10^{-2})-	9.26×10^{-5} (1.28×10^{-4})+	**5.55×10^{-1} (1.13×10^{-3})-**	5.39×10^{-1} (1.57×10^{-3})
LIR-CMOP14	4.93×10^{-4} (3.20×10^{-4})+	5.48×10^{-1} (2.11×10^{-3})+	1.01×10^{-3} (7.63×10^{-5})+	5.30×10^{-1} (3.32×10^{-3})+	**5.55×10^{-1} (1.37×10^{-3})-**	5.53×10^{-4} (3.16×10^{-4})+	5.55×10^{-1} (1.54×10^{-3})-	5.53×10^{-1} (1.48×10^{-3})
+/-/=	14/0/0	9/2/3	14/0/0	12/0/2	12/2/0	14/0/0	12/2/0	

Table 3. *Cont.*

	BiCo	DPPPS	AGEMOEAII	DSPCMDE	CMOEA-MS	TSTI	CTSEA	PSCMO
MW1	NaN (0.00%)+	3.55×10^{-1} (1.95×10^{-1})+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	**4.88×10^{-1} (1.82×10^{-4})**
MW2	**5.21×10^{-1} (1.82×10^{-2})-**	4.47×10^{-1} (5.48×10^{-2})+	4.09×10^{-1} (6.04×10^{-2})+	NaN (76.67%)+	4.34×10^{-1} (4.36×10^{-2})+	4.38×10^{-1} (4.45×10^{-2})+	4.79×10^{-1} (3.91×10^{-2})+	5.01×10^{-1} (1.15×10^{-2})
MW3	5.37×10^{-1} (1.28×10^{-3})+	5.37×10^{-1} (7.51×10^{-4})+	0.00 (0.00)+	**5.41×10^{-2} (3.54×10^{-4})=**	4.60×10^{-1} (1.85×10^{-1})+	4.32×10^{-1} (1.27×10^{-1})+	5.40×10^{-1} (1.44×10^{-3})+	5.41×10^{-1} (2.73×10^{-4})
MW4	NaN (0.00%)+	7.99×10^{-1} (1.35×10^{-1})+	NaN (0.00%)+	NaN (90.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	**8.29×10^{-1} (6.12×10^{-4})**
MW5	NaN (0.00%)+	2.74×10^{-1} (8.81×10^{-2})+	NaN (0.00%)+	NaN (36.67%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	**3.22×10^{-1} (4.37×10^{-4})**
MW6	**2.69×10^{-1} (1.82×10^{-2})-**	2.24×10^{-1} (5.45×10^{-2})-	7.75×10^{-2} (3.98×10^{-2})+	NaN (86.67%)+	1.11×10^{-1} (2.96×10^{-2})+	8.34×10^{-2} (4.09×10^{-2})+	2.00×10^{-1} (6.96×10^{-2})=	2.10×10^{-1} (2.99×10^{-2})
MW7	4.05×10^{-1} (1.40×10^{-3})+	4.07×10^{-1} (8.37×10^{-4})+	2.44×10^{-1} (7.63×10^{-4})+	**4.12×10^{-1} (3.21×10^{-4})-**	4.01×10^{-1} (1.74×10^{-3})+	4.01×10^{-1} (1.67×10^{-3})+	4.10×10^{-1} (8.87×10^{-4})-	4.08×10^{-1} (8.63×10^{-4})
MW8	**4.90×10^{-1} (3.50×10^{-2})-**	4.76×10^{-1} (2.88×10^{-2})-	NaN (93.33%)+	NaN (0.00%)+	NaN (93.33%)+	NaN (96.67%)+	4.28×10^{-1} (5.92×10^{-2})=	4.52×10^{-1} (1.94×10^{-2})
MW9	NaN (0.00%)+	0.00 (0.00)+	NaN (0.00%)+	NaN (90.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	**3.96×10^{-1} (1.39×10^{-1})**
MW10	**3.28×10^{-1} (4.62×10^{-2})-**	3.06×10^{-1} (6.37×10^{-2})=	NaN (0.00%)+	NaN (0.00%)+	NaN (20.43%)+	NaN (40.00%)+	NaN (86.67%)+	3.08×10^{-1} (2.52×10^{-2})
MW11	4.11×10^{-1} (7.12×10^{-2})+	4.47×10^{-1} (2.64×10^{-4})=	2.73×10^{-1} (2.02×10^{-4})+	NaN (0.00%)+	3.70×10^{-1} (7.93×10^{-2})+	3.67×10^{-1} (8.51×10^{-2})+	**4.47×10^{-1} (2.91×10^{-4})=**	4.47×10^{-1} (2.83×10^{-4})
MW12	NaN (73.50%)+	0.00 (0.00)+	NaN (0.00%)+	NaN (96.67%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	**6.05×10^{-1} (2.20×10^{-4})**
MW13	**4.13×10^{-1} (1.80×10^{-2})-**	3.91×10^{-1} (4.03×10^{-2})-	2.12×10^{-1} (5.03×10^{-2})+	1.53×10^{-1} (4.83×10^{-2})+	2.99×10^{-1} (5.14×10^{-2})+	2.57×10^{-1} (9.21×10^{-2})+	3.83×10^{-1} (4.80×10^{-2})-	3.82×10^{-1} (1.19×10^{-2})
MW14	3.48×10^{-1} (2.50×10^{-2})+	2.21×10^{-1} (1.53×10^{-2})+	4.30×10^{-1} (2.05×10^{-2})+	2.36×10^{-1} (9.26×10^{-3})+	3.72×10^{-1} (2.99×10^{-2})+	4.18×10^{-1} (2.62×10^{-2})+	4.02×10^{-1} (4.14×10^{-2})+	**4.44×10^{-1} (7.47×10^{-3})**
+/-/=	9/5/0	9/3/2	14/0/0	12/1/1	14/0/0	14/0/0	9/2/3	
RWMOP1	3.59×10^{-1} (6.50×10^{-3})+	3.45×10^{-1} (1.24×10^{-2})+	3.25×10^{-1} (8.87×10^{-3})+	NaN (0.00%)+	3.21×10^{-1} (1.87×10^{-2})+	3.49×10^{-1} (8.69×10^{-3})+	3.29×10^{-1} (7.88×10^{-3})+	**3.73×10^{-1} (2.88×10^{-3})**
RWMOP2	3.18×10^{-2} (1.80×10^{-6})=	3.18×10^{-2} (2.44×10^{-6})+	**3.18×10^{-2} (3.82×10^{-7})-**	3.18×10^{-2} (7.82×10^{-7})+	3.18×10^{-2} (8.60×10^{-7})+	3.18×10^{-2} (3.24×10^{-6})+	3.18×10^{-2} (1.18×10^{-6})+	3.18×10^{-2} (8.93×10^{-7})
RWMOP3	NaN (0.00%)+	6.36×10^{-1} (2.13×10^{-1})=	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	**7.03×10^{-1} (9.69×10^{-2})**
RWMOP4	NaN (96.67%)+	**9.02×10^{-1} (6.87×10^{-3})+**	NaN (66.67%)+	NaN (0.00%)+	7.15×10^{-1} (1.14×10^{-1})+	NaN (96.67%)+	NaN (36.67%)+	9.02×10^{-1} (1.61×10^{-2})
RWMOP5	**2.42×10^{-1} (7.71×10^{-6})=**	2.42×10^{-1} (7.41×10^{-6})+	2.41×10^{-1} (4.57×10^{-5})+	2.41×10^{-1} (7.42×10^{-5})+	2.32×10^{-1} (1.03×10^{-5})+	2.32×10^{-1} (7.83×10^{-6})+	2.42×10^{-1} (6.98×10^{-6})=	2.42×10^{-1} (1.00×10^{-5})
RWMOP6	NaN (0.00%)+	2.29×10^{-2} (1.66×10^{-4})+	NaN (0.00%)+	NaN (0.00%)+	NaN (90.00%)+	**3.85×10^{-2} (1.67×10^{-2})-**	NaN (0.00%)+	3.08×10^{-2} (9.77×10^{-3})
RWMOP7	NaN (0.00%)+	5.14×10^{-1} (2.72×10^{-1})=	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	**5.95×10^{-1} (1.73×10^{-1})**
RWMOP8	NaN (0.00%)+	9.09×10^{-2} (2.26×10^{-1})+	NaN (0.00%)+	NaN (10.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	**2.54×10^{-1} (1.73×10^{-1})**
RWMOP9	NaN (0.00%)+	4.81×10^{-2} (1.84×10^{-1})+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	**4.34×10^{-1} (2.63×10^{-1})**
RWMOP10	NaN (0.00%)+	0.00 (0.00)=	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	**1.89×10^{-4} (1.03×10^{-3})**
RWMOP11	NaN (0.00%)+	0.00 (0.00)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	**1.36×10^{-2} (7.43×10^{-2})**
RWMOP12	NaN (0.00%)+	2.85×10^{-1} (1.45×10^{-1})+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	NaN (0.00%)+	**4.33×10^{-1} (1.36×10^{-1})**
+/-/=	10/0/2	9/0/3	11/1/0	12/0/0	12/0/0	11/1/0	11/0/1	

5.1. Comparison on the CF Test Set

From Tables 2 and 3, it is evident that the PSCMO algorithm achieves the best IGD and HV results across the nine test functions (CF2 to CF10) compared to those of the other algorithms. This superior performance is attributed to PSCMO's ability to dynamically adjust the evolutionary strategies based on the population states, enhancing its search capability around the CPF and improving both the convergence and distribution of the population. In contrast, the DPPPS algorithm only attains optimal IGD+ and HV values for CF1, as its collaborative evolution framework with three populations is limited to achieving the optimal solutions on a small subset of functions. Regarding feasible solution rates, PSCMO and DPPPS consistently find feasible solutions on the CF test set, whereas the remaining six algorithms fail to do so across all functions. PSCMO achieves optimal solutions on nine test functions, significantly outperforming DPPPS, which only achieves optimality on one function. This stark difference highlights the limitations of DPPPS

compared to PSCMO. Overall, across various test problems, the PSCMO algorithm exhibits a more balanced and superior performance in terms of convergence and diversity while also ensuring feasibility, surpassing the seven other algorithms in the comparison.

5.2. Comparison on the DAS-CMOP Test Set

The DAS-CMOP problems are characterized by Pareto fronts composed of disjoint line segments obscured by extensive infeasible regions, with some feasible regions situated far from the unconstrained Pareto frontier. As shown in Table 2, PSCMO demonstrates a superior performance, achieving the optimal IGD+ for DAS-CMOP1 to DAS-CMOP4, DAS-CMOP6, DAS-CMOP8, and DAS-CMOP9. However, CTSEA obtains the best IGD+ only for DAS-CMOP5. Similarly, Table 3 highlights PSCMO's achievement of the best HV among the compared algorithms for DAS-CMOP2 to DAS-CMOP4, DAS-CMOP6, and DAS-CMOP9. The other algorithms, including AGEMOEAII, DSPCMDE, TSTI, and CTSEA, achieve the best HV for DAS-CMOP7, DAS-CMOP1, DAS-CMOP8, and DAS-CMOP5, respectively. Although CTSEA performs well for DAS-CMOP5, achieving both the best IGD+ and HV, its performance is less competitive for the other test functions, ranking as low as sixth in terms of the IGD+ and HV values for DAS-CMOP9. This suggests that CTSEA is effective in specific problems but lacks general applicability across a broader range of DAS-CMOP functions. Regarding feasible solution rates, algorithms such as BiCo, AGEMOEAII, and DSPCMDE fail to consistently obtain feasible solutions for all test problems in this set. This limitation may stem from the challenges associated with acquiring feasible solutions in multiple swarm algorithm frameworks. In summary, PSCMO stands out among the compared algorithms, consistently demonstrating an excellent performance in its convergence, diversity, and feasibility across the DAS-CMOP benchmark functions.

5.3. Comparison on the LIR-CMOP Test Set

The LIRCMOP test suite is designed to challenge algorithms in handling large infeasible regions, with numerous constraints and a vast infeasible domain. This requires algorithms to effectively explore infeasible regions and identify the CPF. From Table 2, it is evident that the PSCMO algorithm outperforms the others by achieving the optimal IGD+ in LIR-CMOP2 and LIR-CMOP4 to LIR-CMOP12. This highlights PSCMO's proficiency in managing large infeasible regions (LIR-CMOP5 to LIR-CMOP8) and disconnected feasible regions (LIR-CMOP9 to LIR-CMOP12). While DPPPS achieves the best IGD+ in LIR-CMOP1 and LIR-CMOP3 and CTSEA excels in LIR-CMOP13, PSCMO maintains a superior performance in most of the test cases. As shown in Table 3, PSCMO also achieves the optimal HV in LIR-CMOP1, LIR-CMOP2, LIR-CMOP4 to LIR-CMOP10, and LIR-CMOP12. In contrast, DPPPS secures the optimal HV in LIR-CMOP3 and LIR-CMOP11, CMOEA-MS excels in LIR-CMOP14, and CTSEA leads in LIR-CMOP13. The remaining four algorithms do not achieve any optimal results, which may be attributed to the poor performance of BiCo, AGEMOEAII, and DSPCMDE due to their complex multiple swarm algorithm frameworks. TSTI's two-stage framework may also hinder its performance in this test suite. Overall, PSCMO demonstrates a clear advantage in the LIR-CMOP test set, reflecting its robust convergence and diversity optimization capabilities. Its strong performance positions it as a highly competitive algorithm for solving complex constrained multi-objective problems.

5.4. Comparison on the MW Test Set

The MW test suite, designed for general applicability, presents a diverse set of feasible region characteristics, including disconnected regions, large infeasible separations, and regions distant from the unconstrained Pareto front. Table 2 summarizes the IGD+ results for the seven comparison algorithms across MW1 to MW14. PSCMO demonstrates the best

overall performance, achieving the optimal IGD+ on eight test functions and competitive results on the remaining six. This success can be attributed to the relatively large feasible areas in the MW test set, which are more accessible for processing. Regarding the HV, as shown in Table 3, BiCo, DSPCMDE, and CTSEA achieve the highest HV values on five, two, and one test functions, respectively. PSCMO outperforms the others on six test functions and remains competitive on the remaining eight. AGEMOEAII fails to achieve the best results for any of the 14 test functions, likely due to its resource-intensive approach, which avoids assumptions about the geometric shape of the Pareto front. This hinders its ability to balance global and local search effectively. Similarly, CMOEA-MS and TSTI do not achieve the best results on any test function, primarily due to the poor solution distribution caused by their two-stage evolutionary algorithm frameworks. From the perspective of a feasible solution rate, PSCMO consistently produces feasible solutions, aligning with its performance on the CF test set and demonstrating strong feasibility. PSCMO achieves the highest number of optimal IGD+ and HV results across the test functions, underscoring its superior convergence and distribution capabilities. This highlights PSCMO as an effective and robust algorithm for solving constrained multi-objective optimization problems in diverse scenarios.

5.5. Comparison on Real-World Problems

From Table 2, it is evident that PSCMO achieved the best IGD+ values across 10 test functions (RWMOP1 to RWCMOP3 and RWMOP5 to RWMOP11), demonstrating its superior convergence in real-world problems. In comparison, DPPPS and CMOEA-MS achieved the best IGD+ in RWMOP4 and RWCMOP12, respectively, but none of the other five comparison algorithms reached the best IGD+, further emphasizing PSCMO's strong performance. As shown in Table 3, PSCMO excelled in HV as well, obtaining the best HV values in RWMOP1, RWMOP3, and RWMOP7 to RWMOP12. In contrast, BiCo, DPPPS, AGEMOEAII, and TSTI each achieved the best HV in only one of the test functions (RWMOP5, RWMOP4, RWMOP2, and RWMOP6, respectively). This indicates that these algorithms tend to perform well on specific real-world problems but lack broad applicability. In terms of a feasible solution rate, PSCMO consistently provided feasible solutions, aligning with the results from the CF and MW test sets and confirming its excellent feasibility. Additionally, Figure 2 highlights that PSCMO outperformed the other seven comparison algorithms in terms of both IGD+ and HV, as evidenced by the significantly better average rankings in the Friedman test. Overall, PSCMO achieved the best IGD+ and HV, demonstrating its superior convergence, diversity, and feasibility, thus confirming its effectiveness and versatility in solving real-world constrained multi-objective optimization problems.

From the analysis of the four sets of test functions and real-world problems, it is evident that PSCMO is a highly competitive algorithm in terms of its convergence, diversity, and feasibility. The Friedman test results in Figure 2 further support this conclusion, with PSCMO consistently achieving the best overall average ranking. Although the DPPPS algorithm ranks second in the Friedman test, it still lags behind PSCMO, likely due to its weaker convergence performance based on a three-population co-evolution framework. While BiCo exhibits a comparable performance to that of PSCMO in the MW test set, PSCMO significantly outperforms BiCo in the other three test sets and the real-world problems. This is reflected in PSCMO's superior average ranking in the Friedman test compared to that of BiCo, further highlighting PSCMO's balanced and superior capabilities across a wide range of CMOPs. In summary, PSCMO demonstrates a robust and reliable performance across diverse problem types, excelling in its convergence, diversity, and feasibility. This conclusion is fully validated by the comprehensive comparison results

and statistical analyses, positioning PSCMO as an effective and versatile solution for solving CMOPs.

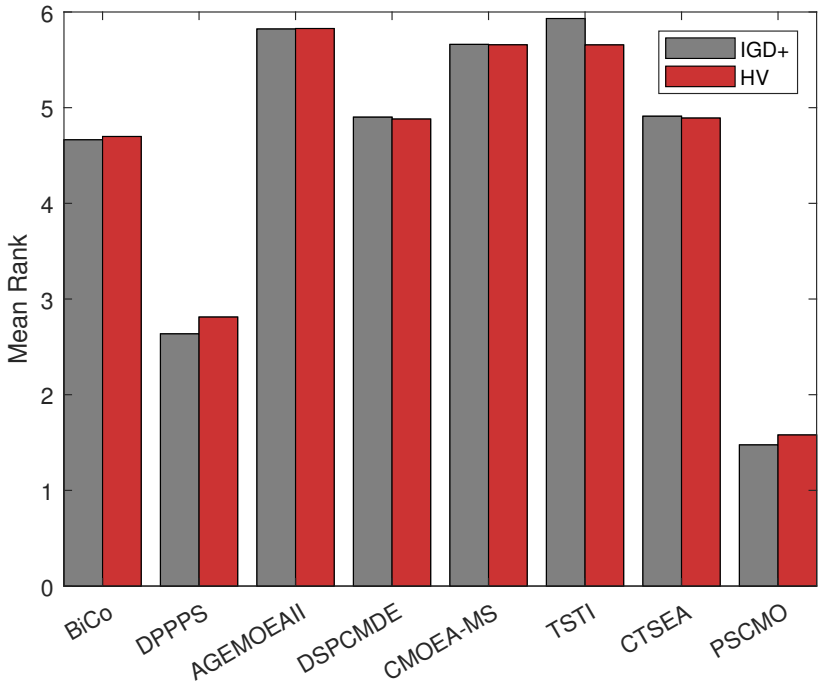


Figure 2. Average ranking of 8 algorithms in the Friedman test for IGD+ and HV across all test functions.

5.6. Convergence Speed Analysis

Figure 3 illustrates the IGD+ convergence curves for PSCMO and six comparative algorithms after 100,000 evaluations, with only the IGD+ values of the feasible solutions plotted. The results indicate that the PSCMO algorithm excels in its convergence speed, especially in problems with narrow feasible regions like DAS-CMOP4, LIR-CMOP2, and MW13. This is due to the ability of both the subpopulations and unconstrained populations in PSCMO to approximate the CPF from both the infeasible and feasible regions of the search space. PSCMO also performs well in problems with large feasible domains, such as LIR-CMOP5 and LIR-CMOP6, demonstrating its capability to efficiently handle such cases. Additionally, PSCMO excels in problems where the CPF is composed of separate segments, such as CF3, DAS-CMOP5, and DAS-CMOP7. The complementarity and collaboration between subpopulations and unconstrained populations allow PSCMO to quickly find a well-distributed set of solutions across the entire CPF, leading to rapid convergence. Compared to the other seven algorithms, PSCMO shows significantly faster convergence as the evaluations increase. While PSCMO initially shows higher IGD+ values on the test functions CF4, CF6, MW2, and MW14, it rapidly converges to the Pareto front after around 30,000 evaluations, highlighting its strong ability to escape local optima. Even though DPPPS achieves faster convergence compared to that of the other algorithms, it still falls short of PSCMO's performance. Overall, PSCMO outperforms the other algorithms in terms of both its convergence speed and final convergence quality, demonstrating its superiority in solving constrained multi-objective optimization problems.

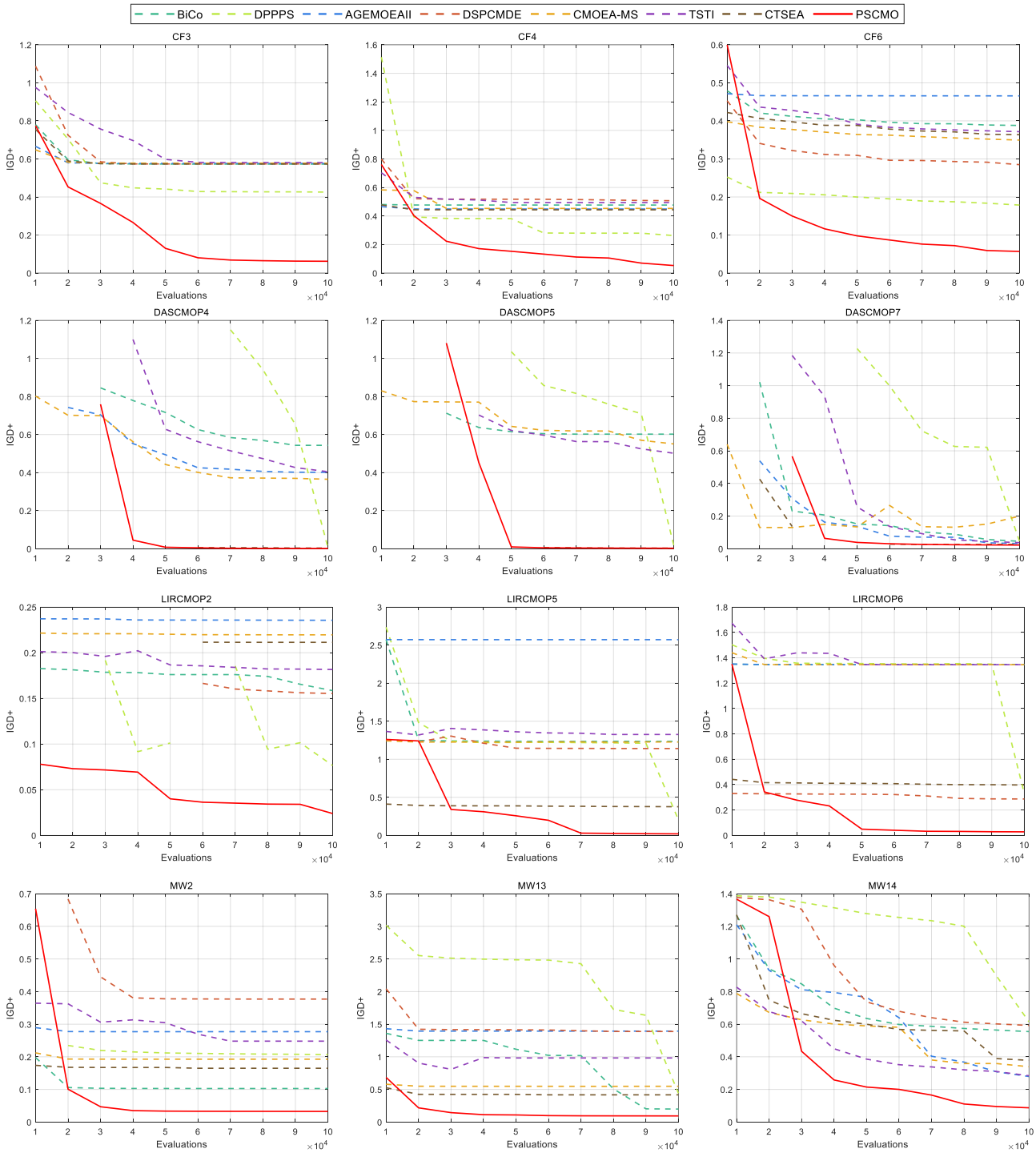


Figure 3. IGD+ convergence graph of 8 algorithms under 100,000 evaluations.

5.7. Diversity Analysis

Figure 4 displays the population distribution of PSCMO and seven comparative algorithms across four different test functions, with each algorithm using a population of 100 individuals and a maximum evaluation count of 100,000. The results show that PSCMO consistently achieves the best distribution across these functions, outperforming the other algorithms in both convergence and diversity. In particular, AGEMOEAII, CMOEA-MS, TSTI, and CTSEA perform poorly in terms of convergence and diversity. In the CF2 test function, none of these algorithms can reach the Pareto front region, and in DAS-CMOP9, they fail to reach smaller feasible domains. This suggests that these algorithms struggle to relax the constraints sufficiently, preventing them from escaping local optima when

they are encountered. The same issue is observed with BiCo. For CMOPs, it is crucial to appropriately relax the constraints to allow the algorithm to explore the entire feasible domain and reach all feasible regions, especially smaller ones. On the other hand, DPPPS and DSPCMDE demonstrate good convergence and diversity, performing similarly to PSCMO in CF2, LIR-CMOP12, and MW14. However, as seen in DAS-CMOP9, the diversity of DPPPS and DSPCMDE significantly lags behind that of PSCMO. This may be due to their susceptibility to infeasible regions, which hinders their ability to maintain a well-distributed population across the solution space. Figure 5 shows the convergence curve for HV, representing the evolution of the diversity indicators throughout the population's evolution process. The graph reveals that PSCMO does not perform optimally during the first 30,000 iterations, a pattern also observed in the IGD+ curve. However, PSCMO consistently demonstrates a rapid improvement in the population diversity after this initial phase, as seen in cases such as CF3, CF4, DAS-CMOP4, and MW14. From CF3 and CF4, it is evident that the other comparative algorithms tend to become trapped in local optima, preventing them from escaping and causing their HV values to plateau before reaching an optimal solution. In contrast, as seen for CF2, DAS-CMOP4, MW2, MW13, and MW14, PSCMO exhibits a relatively faster increase in diversity. This is due to PSCMO's ability to switch between different CHTs based on varying population states, which enhances both the population diversity and the algorithm's capacity to explore global optima. Overall, PSCMO outperforms the other algorithms in terms of the diversity and the convergence speed for HV.

5.8. Statistics and Testing

Tables 4 and 5 provide the Wilcoxon signed rank test results for IGD+ and HV between PSCMO and the seven comparative algorithms. In general, a larger difference between R^+ and R^- indicates the greater advantage of PSCMO over the comparison algorithms. The results show that the number of R^- (instances where PSCMO performs worse) is significantly smaller than the number of R^+ (instances where PSCMO performs better). This trend strongly suggests that PSCMO outperforms the seven compared algorithms. Moreover, all of the *P*-values are less than 0.05, further confirming that the advantage of PSCMO is statistically significant. Based on the statistical analysis, it can be concluded that PSCMO demonstrates a superior performance in terms of both convergence and diversity, making it a highly competitive algorithm for constrained multi-objective optimization problems.

Table 4. Wilcoxon signed rank test results of PSCMO and IGD+ of 7 comparative algorithm problems.

PSCMO VS	R^+	R^-	*P*	Level = 0.05
BiCo	1066	419	0.002709	YES
DPPPS	1309	231	0.000003	YES
AGEMOEAII	1105	380	0.000914	YES
DSPCMDE	1016	469	0.009372	YES
CMOEA-MS	1137	348	0.000346	YES
TSTI	1143	342	0.000286	YES
CTSEA	1050	435	0.004105	YES

Table 5. Wilcoxon signed rank test results of PSCMO and HV of 7 comparative algorithm problems.

PSCMO VS	R^+	R^-	*P*	Level = 0.05
BiCo	1584	69	0.00000	YES
DPPPS	1477	176	0.00000	YES
AGEMOEAII	1648	5	0.00000	YES
DSPCMDE	1566	87	0.00000	YES
CMOEA-MS	1627	26	0.00000	YES
TSTI	1642	11	0.00000	YES
CTSEA	1546	107	0.00000	YES

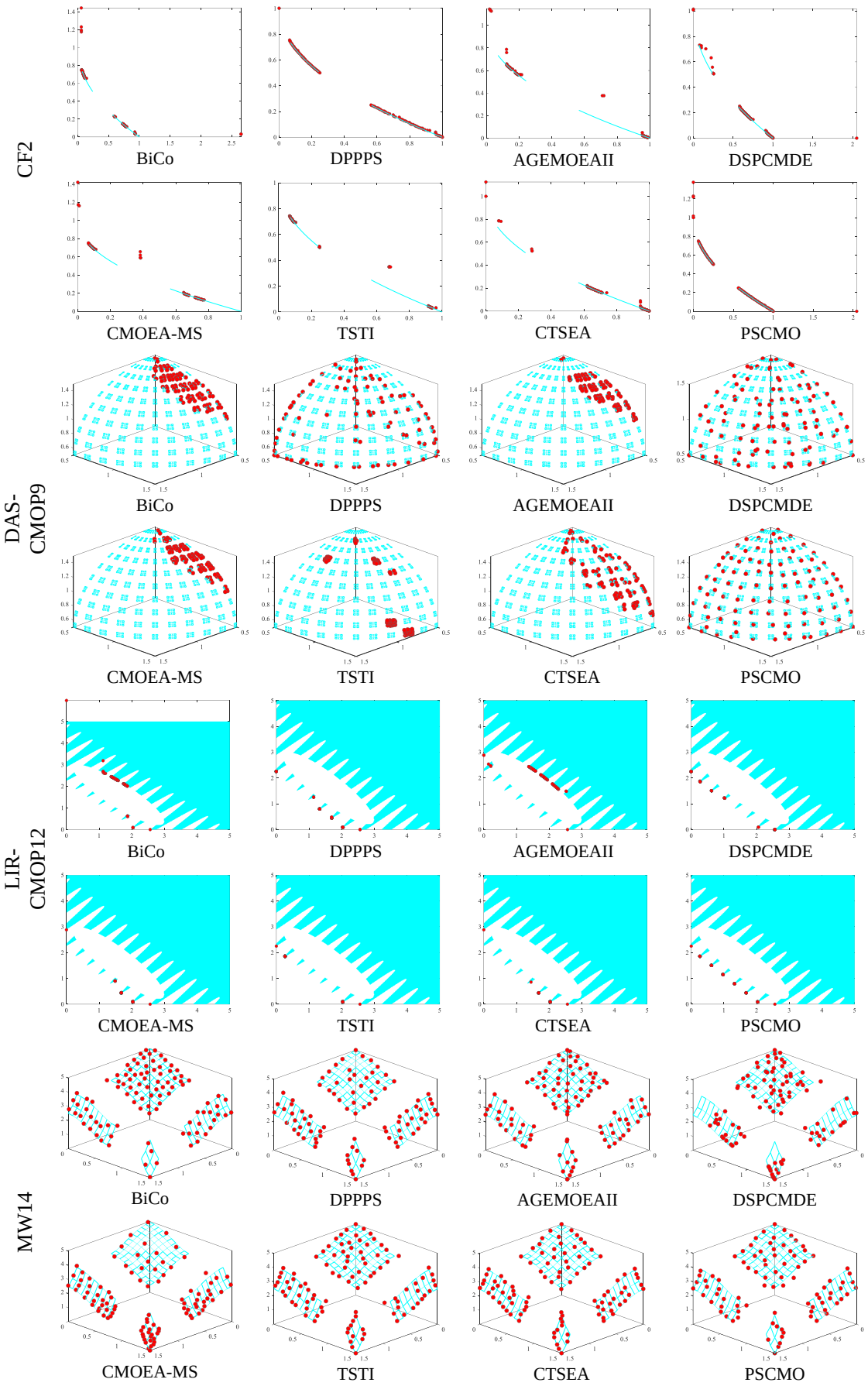


Figure 4. Population distribution of 8 algorithms under 4 different functions.

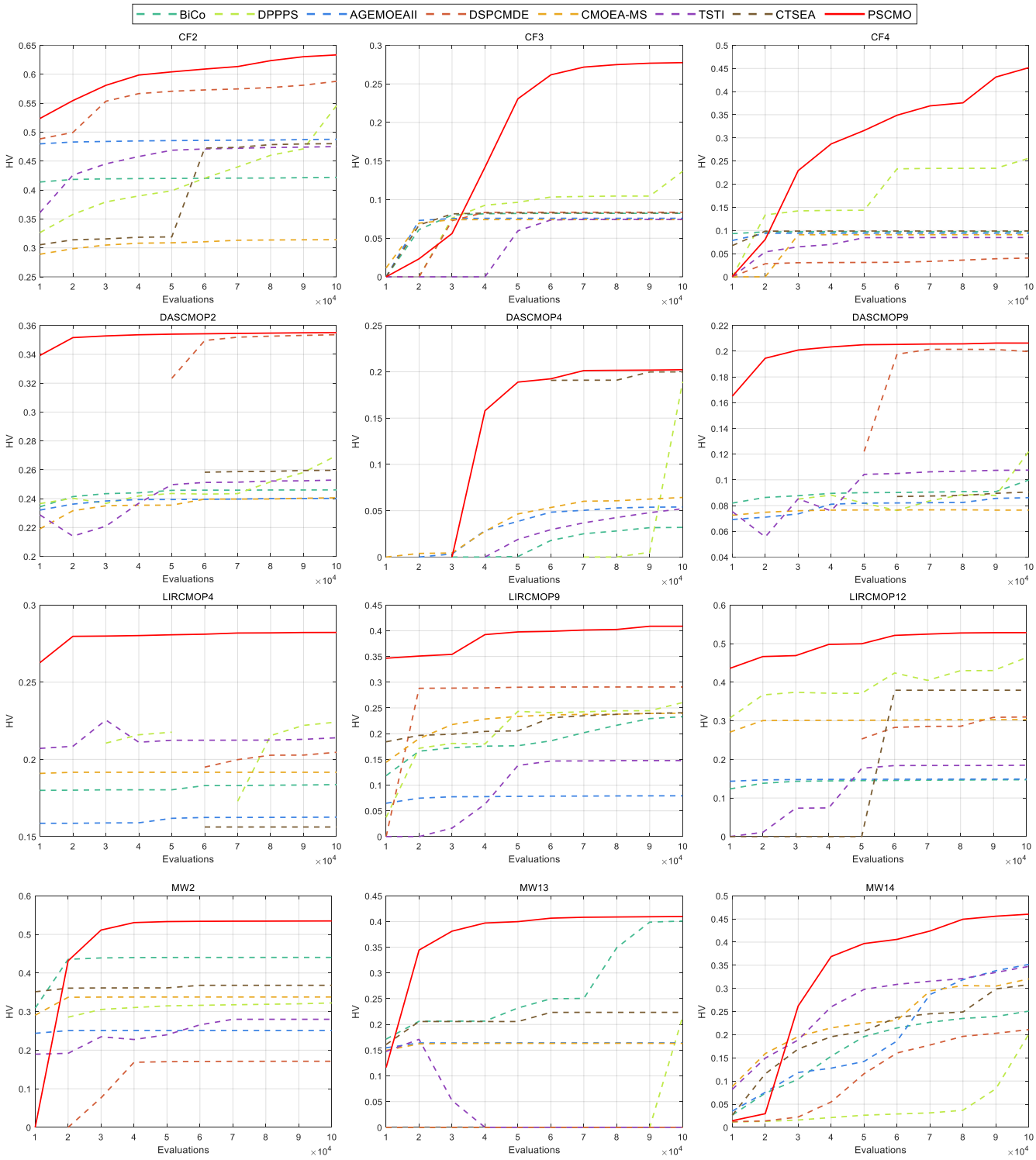


Figure 5. HV convergence graph of 8 algorithms under 100,000 evaluations.

5.9. The Ablation Experiment

PSCMO is a multi-strategy CMOEA that adapts to different situations by combining various strategies. In this section, different strategy combinations were tested, as shown in Table 6. Table 7 presents the experimental results for 47 test functions under five different combinations. From the results in Table 7, it is clear that the number of "+" (where PSCMO outperformed the compared algorithms) is significantly higher than the number of "-" (where PSCMO performed worse), and the significance levels are all below 0.05. This indicates the validity of the experimental results. The performance of PSCMO-A, PSCMO-

B, and PSCMO-C is found to be inferior to that of PSCMO, suggesting that adjusting the population state through different combinations can affect the performance. Specifically, this paper highlights that by changing the CHTs according to the population's changing state, the convergence and diversity of the population can be enhanced effectively. PSCMO-D, which uses a different environmental selection approach, is also found to be inferior to PSCMO, indicating that the environmental selection proposed in this study is better for assisting in the population selection. PSCMO-E, which incorporates population resource allocation, is similarly found to be inferior to PSCMO. This suggests that resource allocation can improve the algorithm's performance. Overall, it can be concluded that combining multiple strategies allows PSCMO to achieve a balanced and superior performance in both its diversity and convergence, reinforcing its robustness and adaptability in tackling constrained multi-objective optimization problems.

Table 6. Testing algorithms under different combinations.

Algorithm Name	Describe
PSCMO-A	Compared with PSCMO, the population state of this algorithm is all in the first state.
PSCMO-B	Compared with PSCMO, the population state of this algorithm is entirely in the second state.
PSCMO-C	Compared with PSCMO, the population state of this algorithm is all in the third state.
PSCMO-D	Compared to PSCMO, the optimal individuals for this algorithm are selected by the CDP model.
PSCMO-E	Compared with PSCMO, the population size of this algorithm is fixed at *NP*.

Table 7. Results of the ablation experiment.

PSCMO VS	IGD+ $(+/-/=)$	*P*	HV $(+/-/=)$	*P*
PSCMO-A	42/0/5	0.000031	39/0/8	0.000000
PSCMO-B	41/1/5	0.000129	40/1/6	0.000016
PSCMO-C	37/1/9	0.000052	37/0/10	0.000000
PSCMO-D	42/1/4	0.000003	42/0/5	0.000000
PSCMO-E	39/0/8	0.000048	39/0/8	0.000000

6. Conclusions

This article introduces a population state-based CMOEA to solve CMOPs efficiently while minimizing computational resource waste. Traditionally, the individual selections were primarily determined by CHTs; however, due to the varying difficulty of CMOPs, many existing algorithms lack universal adaptability. To address this issue, this paper proposes a method for identifying the state of the population by analyzing the positional relationships between the auxiliary and main populations. Based on this, different CHTs are designed to improve the algorithm's applicability. Further, by selecting the auxiliary population based on different CHTs, the evolution of the main population is guided more effectively. This paper also proposes distinct optimal individual selection models based on the population state, enabling the population to escape local optima and accelerate the convergence. Additionally, a simple and effective resource allocation model is introduced to optimize the use of computing resources and avoid the inefficiencies associated with a dual-population model. This model improves the overall efficiency of resource utilization. Through extensive comparative experiments on 47 benchmark test functions and 12 real-world problems, PSCMO has been thoroughly validated, demonstrating superior convergence, diversity, and feasibility.

Although PSCMO demonstrates a strong performance and can obtain high-quality solution sets, there are areas where it could be further improved. Specifically, its performance on the MW test functions is not as superior as its performance on the other test functions, and its diversity in the LIR-CMOP test functions also requires enhancement. These limitations are primarily attributed to the shortcomings of restrictive constraint-handling techniques. Furthermore, ref. [52] can be referenced, which attempts to control

the population state using a second-order transfer function. Future research will focus on developing more effective constraint-handling methods to address these challenges. Despite these areas for improvement, PSCMO has shown excellent performance across 47 test functions and 12 real-world problems, establishing it as a robust and competitive algorithm in evolutionary computing.

The source code is available at https://github.com/YongchaoLucky/PSCMO (accessed on 15 January 2025).

Author Contributions: Conceptualization: S.Z. and Y.L. Methodology: S.Z. Software: Y.L. Validation: H.J., S.Z. and Y.L. Formal analysis: Y.L. Investigation: S.Z. Resources: Q.S. Data curation: Y.L. Writing—original draft preparation: S.Z. Writing—review and editing: Q.S. Visualization: H.J. Supervision: H.J. Project administration: H.J. Funding acquisition: H.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Guiding Science and Technology Projects in Sanming City (2023-G-5).

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors would like to thank the support of Fujian Key Lab of AgricultureIOT Application, IOT Application Engineering Research Center of Fujian Province Colleges and Universities.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Jozefowiez, N.; Semet, F.; Talbi, E.G. Multi-objective vehicle routing problems. *Eur. J. Oper. Res.* **2008**, *189*, 293–309. [CrossRef]
2. Ma, Z.; Wang, Y. Shift-Based Penalty for Evolutionary Constrained Multiobjective Optimization and Its Application. *IEEE Trans. Cybern.* **2023**, *53*, 18–30. [CrossRef] [PubMed]
3. Siddall, J.N. *Optimal Engineering Design: Principles and Applications*; CRC Press: Boca Raton, FL, USA, 1982.
4. Xu, B.; Gong, D.; Zhang, Y.; Yang, S.; Wang, L.; Fan, Z.; Zhang, Y. Cooperative co-evolutionary algorithm for multi-objective optimization problems with changing decision variables. *Inf. Sci.* **2022**, *607*, 278-296. [CrossRef]
5. Sun, R.; Zou, J.; Liu, Y.; Yang, S.; Zheng, J. A Multistage Algorithm for Solving Multiobjective Optimization Problems With Multiconstraints. *IEEE Trans. Evol. Comput.* **2023**, *27*, 1207–1219. [CrossRef]
6. Gan, X.; Sun, J.; Gong, D.; Jia, D.; Dai, H.; Zhong, Z. An Adaptive Reference Vector-Based Interval Multiobjective Evolutionary Algorithm. *IEEE Trans. Evol. Comput.* **2022**, *27*, 1235–1249. [CrossRef]
7. Takahama, T.; Sakai, S. Constrained optimization by the ϵ constrained differential evolution with an archive and gradient-based mutation. In Proceedings of the IEEE Congress on Evolutionary Computation, Barcelona, Spain, 18–23 July 2010; pp. 1–9. [CrossRef]
8. Wang, Q.; Li, Y.; Hou, Z.; Zou, J.; Zheng, J. A novel multi-population evolutionary algorithm based on hybrid collaboration for constrained multi-objective optimization. *Swarm Evol. Comput.* **2024**, *87*, 101581. : 10.1016/j.swevo.2024.101581 [CrossRef]
9. Liang, J.; Qiao, K.; Yu, K.; Qu, B.; Yue, C.; Guo, W.; Wang, L. Utilizing the Relationship Between Unconstrained and Constrained Pareto Fronts for Constrained Multiobjective Optimization. *IEEE Trans. Cybern.* **2023**, *53*, 3873–3886. [CrossRef] [PubMed]
10. Qiao, K.; Yu, K.; Qu, B.; Liang, J.; Song, H.; Yue, C.; Lin, H.; Tan, K.C. Dynamic Auxiliary Task-Based Evolutionary Multitasking for Constrained Multiobjective Optimization. *IEEE Trans. Evol. Comput.* **2023**, *27*, 642–656. [CrossRef]
11. Ming, F.; Gong, W.; Wang, L.; Gao, L. A Constraint-Handling Technique for Decomposition-Based Constrained Many-Objective Evolutionary Algorithms. *IEEE Trans. Syst. Man Cybern. Syst.* **2023**, *53*, 7783–7793. [CrossRef]
12. Liang, J.; Ban, X.; Yu, K.; Qu, B.; Qiao, K.; Yue, C.; Chen, K.; Tan, K.C. A Survey on Evolutionary Constrained Multiobjective Optimization. *IEEE Trans. Evol. Comput.* **2023**, *27*, 201–221. [CrossRef]
13. Wang, B.C.; Li, H.X.; Li, J.P.; Wang, Y. Composite Differential Evolution for Constrained Evolutionary Optimization. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 1482–1495. [CrossRef]
14. Srinivas, N.; Deb, K. Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evol. Comput.* **1994**, *2*, 221–248. [CrossRef]
15. 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]

16. Ma, Z.; Wang, Y.; Song, W. A New Fitness Function with Two Rankings for Evolutionary Constrained Multiobjective Optimization. *IEEE Trans. Syst. Man Cybern.-Syst.* **2021**, *51*, 5005–5016. [CrossRef]
17. Yuan, J.; Liu, H.L.; Ong, Y.S.; He, Z. Indicator-Based Evolutionary Algorithm for Solving Constrained Multiobjective Optimization Problems. *IEEE Trans. Evol. Comput.* **2022**, *26*, 379–391. [CrossRef]
18. Gu, Q.; Bai, J.; Li, X.; Xiong, N.; Lu, C. A constrained multi-objective evolutionary algorithm based on decomposition with improved constrained dominance principle. *Swarm Evol. Comput.* **2022**, *75*, 101162. [CrossRef]
19. Jiao, R.; Xue, B.; Zhang, M. A Multiform Optimization Framework for Constrained Multiobjective Optimization. *IEEE Trans. Cybern.* **2023**, *53*, 5165–5177. [CrossRef] [PubMed]
20. Zuo, M.; Xue, Y. Population Feasibility State Guided Autonomous Constrained Multi-Objective Evolutionary Optimization. *Mathematics* **2024**, *12*, 913. [CrossRef]
21. Fang, J.; Liu, H.L.; Gu, F. A constrained multi-objective evolutionary algorithm based on fitness landscape indicator. *Appl. Soft Comput.* **2024**, *166*, 112128. : 10.1016/j.asoc.2024.112128 [CrossRef]
22. Wang, J.; Liang, G.; Zhang, J. Cooperative differential evolution framework for constrained multiobjective optimization. *IEEE Trans. Cybern.* **2018**, *49*, 2060–2072. [CrossRef]
23. 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]
24. Liang, J.; Ban, X.; Yu, K.; Qiao, K.; Qu, B. Constrained multiobjective differential evolution algorithm with infeasible-proportion control mechanism. *Knowl.-Based Syst.* **2022**, *250*, 109105. [CrossRef]
25. Qiao, K.; Chen, Z.; Qu, B.; Yu, K.; Yue, C.; Chen, K.; Liang, J. A dual-population evolutionary algorithm based on dynamic constraint processing and resources allocation for constrained multi-objective. *Expert Syst. Appl.* **2024**, *238*, 121707. [CrossRef]
26. Ming, F.; Gong, W.; Li, D.; Wang, L.; Gao, L. A competitive and cooperative swarm optimizer for constrained multiobjective optimization problems. *IEEE Trans. Evol. Comput.* **2022**, *27*, 1313–1326. [CrossRef]
27. Zhang, J.; Cao, J.; Zhao, F.; Chen, Z. Two cooperative constraint handling techniques with an external archive for constrained multi-objective optimization. *Memetic Comput.* **2024**, *16*, 115–137. [CrossRef]
28. Qiao, K.; Liang, J.; Yu, K.; Ban, X.; Yue, C.; Qu, B.; Suganthan, P.N. Constraints Separation Based Evolutionary Multitasking for Constrained Multi-Objective Optimization Problems. *IEEE/CAA J. Autom. Sin.* **2024**, *11*, 1819–1835. [CrossRef]
29. Zhang, W.; Yang, J.; Li, G.; Zhang, W.; Yen, G.G. Manifold-assisted coevolutionary algorithm for constrained multi-objective optimization. *Swarm Evol. Comput.* **2024**, *91*, 101717. : 10.1016/j.swevo.2024.101717 [CrossRef]
30. Ma, H.; Wei, H.; Tian, Y.; Cheng, R.; Zhang, X. A multi-stage evolutionary algorithm for multi-objective optimization with complex constraints. *Inf. Sci.* **2021**, *560*, 68–91. [CrossRef]
31. 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]
32. Ming, F.; Gong, W.; Gao, L. Adaptive Auxiliary Task Selection for Multitasking-Assisted Constrained Multi-Objective Optimization [Feature]. *IEEE Comput. Intell. Mag.* **2023**, *18*, 18–30. [CrossRef]
33. Ma, Y.; Shen, B.; Pan, A. Constrained evolutionary optimization based on dynamic knowledge transfer. *Expert Syst. Appl.* **2024**, *240*, 122450. [CrossRef]
34. Zhong, X.; Yao, X.; Qiao, K.; Gong, D. A Multitasking-Based Constrained Multi-Objective Evolutionary Algorithm With Forward and Backward Stages. *IEEE Trans. Emerg. Top. Comput. Intell.* **2024**, *8*, 3474–3488. [CrossRef]
35. Zhang, W.; Liu, J.; Zhang, W.; Liu, Y.; Tan, S. A two-stage evolutionary algorithm assisted by multi-archives for constrained multi-objective optimization. *Appl. Soft Comput.* **2024**, *162*, 111840. : 10.1016/j.asoc.2024.111840 [CrossRef]
36. Qiao, K.; Liang, J.; Yu, K.; Yue, C.; Lin, H.; Zhang, D.; Qu, B. Evolutionary Constrained Multiobjective Optimization: Scalable High-Dimensional Constraint Benchmarks and Algorithm. *IEEE Trans. Evol. Comput.* **2024**, *28*, 965–979. [CrossRef]
37. Zhang, Q.; Zhou, A.; Zhao, S.; Suganthan, P.; Liu, W.; Tiwari, S. *Multiobjective optimization Test Instances for the CEC 2009 Special Session and Competition*; University of Essex: Colchester, UK, 2008.
38. Fan, Z.; Li, W.; Cai, X.; Li, H.; Wei, C.; Zhang, Q.; Deb, K.; Goodman, E. Difficulty Adjustable and Scalable Constrained Multiobjective Test Problem Toolkit. *Evol. Comput.* **2020**, *28*, 339–378. _a_00259 [CrossRef] [PubMed]
39. Fan, Z.; Li, W.; Cai, X.; Huang, H.; Fang, Y.; You, Y.; Mo, J.; Wei, C.; Goodman, E. An improved epsilon constraint-handling method in MOEA/D for CMOPs with large infeasible regions. *Soft Comput.* **2019**, *23*, 12491–12510. [CrossRef]
40. Ma, Z.; Wang, Y. Evolutionary Constrained Multiobjective Optimization: Test Suite Construction and Performance Comparisons. *IEEE Trans. Evol. Comput.* **2019**, *23*, 972–986. [CrossRef]
41. Kumar, A.; Wu, G.; Ali, M.Z.; Luo, Q.; Mallipeddi, R.; Suganthan, P.N.; Das, S. A Benchmark-Suite of real-World constrained multi-objective optimization problems and some baseline results. *Swarm Evol. Comput.* **2021**, *67*, 100961. [CrossRef]
42. Ishibuchi, H.; Masuda, H.; Nojima, Y. Sensitivity of performance evaluation results by inverted generational distance to reference points. In Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 24–29 July 2016; pp. 1107–1114. [CrossRef]

43. 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]
44. Yang, Y.; Liu, J.; Tan, S.; Wang, H. A multi-objective differential evolutionary algorithm for constrained multi-objective optimization problems with low feasible ratio. *Appl. Soft Comput.* **2019**, *80*, 42–56. [CrossRef]
45. Ming, F.; Gong, W.; Wang, L.; Lu, C. A tri-population based co-evolutionary framework for constrained multi-objective optimization problems. *Swarm Evol. Comput.* **2022**, *70*, 101055. [CrossRef]
46. Panichella, A. An improved Pareto front modeling algorithm for large-scale many-objective optimization. In Proceedings of the Genetic and Evolutionary Computation Conference, Boston, MA, USA, 9–13 July 2022; pp. 565–573.
47. Yu, K.; Liang, J.; Qu, B.; Luo, Y.; Yue, C. Dynamic selection preference-assisted constrained multiobjective differential evolution. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *52*, 2954–2965. [CrossRef]
48. Song, Z.; Wang, H.; Xue, B.; Zhang, M.; Jin, Y. Balancing Objective Optimization and Constraint Satisfaction in Expensive Constrained Evolutionary Multiobjective Optimization. *IEEE Trans. Evol. Comput.* **2024**, *28*, 1286–1300. [CrossRef]
49. Dong, J.; Gong, W.; Ming, F.; Wang, L. A two-stage evolutionary algorithm based on three indicators for constrained multi-objective optimization. *Expert Syst. Appl.* **2022**, *195*, 116499. [CrossRef]
50. Ming, F.; Gong, W.; Zhen, H.; Li, S.; Wang, L.; Liao, Z. A simple two-stage evolutionary algorithm for constrained multi-objective optimization. *Knowl.-Based Syst.* **2021**, *228*, 107263. [CrossRef]
51. Tian, Y.; Cheng, R.; Zhang, X.; Jin, Y. PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization. *IEEE Comput. Intell. Mag.* **2017**, *12*, 73–87. [CrossRef]
52. Tavakoli, A.; Forouzanfar, M. Adaptive droop control for current sharing and bus voltage stability in DC microgrids. *Int. Trans. Electr. Energy Syst.* **2021**, *31*, e12753. [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.