




Even Search in a Promising Region for Constrained Multi-Objective Optimization

Fei Ming , Wenyin Gong , *Member, IEEE*, and Yaochu Jin , *Fellow, IEEE*

Abstract—In recent years, a large number of approaches to constrained multi-objective optimization problems (CMOPs) have been proposed, focusing on developing tweaked strategies and techniques for handling constraints. However, an overly fine-tuned strategy or technique might overfit some problem types, resulting in a lack of versatility. In this article, we propose a generic search strategy that performs an even search in a promising region. The promising region, determined by obtained feasible non-dominated solutions, possesses two general properties. First, the constrained Pareto front (CPF) is included in the promising region. Second, as the number of feasible solutions increases or the convergence performance (i.e., approximation to the CPF) of these solutions improves, the promising region shrinks. Then we develop a new strategy named even search, which utilizes the non-dominated solutions to accelerate convergence and escape from local optima, and the feasible solutions under a constraint relaxation condition to exploit and detect feasible regions. Finally, a diversity measure is adopted to make sure that the individuals in the population evenly cover the valuable areas in the promising region. Experimental results on 45 instances from four benchmark test suites and 14 real-world CMOPs have demonstrated that searching evenly in the promising region can achieve competitive performance and excellent versatility compared to 11 most state-of-the-art methods tailored for CMOPs.

Index Terms—Constrained multi-objective optimization, even search, evolutionary algorithms, promising region, real-world problems.

I. INTRODUCTION

CONSTRAINED multi-objective optimization problems (CMOPs) refer to optimization problems with two or three objective functions and some constraint conditions, which widely exist in real-world applications and scientific research, such as vehicle scheduling of the urban bus line [1].

In recent years, many research efforts have been devoted to developing approaches for solving CMOPs [2]. Researchers in the multi-objective optimization field have proposed many

new constrained multi-objective optimization evolutionary algorithms (CMOEAs), most of which focused on algorithmic strategies or constraint-handling techniques (CHTs). Regarding the algorithmic strategies, ignoring or relaxing constraints in an auxiliary problem to help the algorithm solve the original CMOP [3], [4] and balancing the optimization of objectives and satisfaction of constraints [5], [6] are both popular. As for CHTs, the ϵ -constrained technique [7], [8], the penalty function [1], and new CHTs [9] have been improved to handle CMOPs with more complex features by utilizing valuable infeasible solutions. However, as pointed out in the experiments of [10] and according to the No Free Lunch theory [11], a specific strategy or tailored technique, especially if it is labor-intensively designed and complicated, might overfit some problem types¹. As a consequence, these methods lack versatility and may be hard to use in real-world applications that are always subject to unknown features and difficulties.

To overcome the above shortcomings, this work proposes to conduct an even search in the promising region for CMOPs. To be more specific, the main contributions are as follows:

1) We propose an approach that makes use of the obtained feasible non-dominated solutions to define a promising region. The promising region has two properties. First, the constrained Pareto front (CPF) must be included in the region no matter what features the CMOP has. Thus, searching in the promising region helps search the CPF. Second, as either the number of feasible solutions or the convergence (i.e., the approximation degree to the CPF) improves, the promising region will shrink. Thus, the region where the even search should be performed becomes smaller and is more precisely defined.

2) We develop an even search method that utilizes valuable solutions located in the promising region to assist the search for the CPF. Specifically, those non-dominated solutions with good convergence and diversity are used to accelerate the approximation to the CPF and help the search algorithm escape from local optima. Moreover, those feasible solutions under a constraint-relaxed condition with good feasibility and diversity are used to exploit the detected and explore the undetected feasible regions. In addition, we use the minimum Euclidean distance between solutions as a measure to enhance diversity. In this way, the preserved solutions are well-distributed in valuable areas in the promising region to achieve

Manuscript received June 18, 2023; accepted July 25, 2023. This work was partly supported by the National Natural Science Foundation of China (62076225). Recommended by Associate Editor Nian Zhang. (Corresponding authors: Wenyin Gong and Yaochu Jin.)

Citation: F. Ming, W. Gong, and Y. Jin, “Even search in a promising region for constrained multi-objective optimization,” *IEEE/CAA J. Autom. Sinica*, vol. 11, no. 2, pp. 474–486, Feb. 2024.

F. Ming and W. Gong are with the School of Computer Science, China University of Geosciences, Wuhan 430074, China (e-mail: feiming@cug.edu.cn; wygong@cug.edu.cn).

Y. Jin was with the Faculty of Technology, Bielefeld University, North Rhine-Westphalia 33619, Germany. He is now with the School of Engineering, Westlake University, Hangzhou 310030, China (e-mail: jinyaochu@westlake.edu.cn).

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

Digital Object Identifier 10.1109/JAS.2023.123792

¹ Overfit in this work indicates that a method performs extremely well on some problems but unusable (i.e., cannot find a desirable number of applicable solutions) on others.

an even search. Moreover, a simple ε -constrained method, containing only one parameter, is designed to evaluate the feasibility under a constraint-relaxed condition.

3) Based on the above methods, we propose a new CMOEA named constrained multi-objective optimization based on even search (CMOES). CMOES contains two stages, the first stage aims to find feasible non-dominated solutions and approximate the CPF, while the second stage conducts the even search in the determined promising region.

Compared to most existing methods in the constrained multi-objective optimization community, our methods have the following two features. First, since the CPF must be included in the promising region for any CMOP, the strategy to search in the promising region works for any class of CMOPs. Second, our proposed even search method contains only one parameter and does not require problem-specific fine-tuning. The proposed methods are tested on 45 instances from four benchmark suites with very different features and difficulties and 14 real-world CMOPs with unknown features and difficulties. Extensive experimental results have demonstrated that searching in the promising region can achieve competitive performance and excellent versatility compared to 11 most state-of-the-art CMOEAs.

The remainder of this article is organized as follows. Section II introduces the related work and backgrounds. Section III elaborates on the proposed CMOEs. Then, experiments and analysis are presented in Section IV. Finally, conclusions and future research directions are given in Section V.

II. LITERATURE REVIEW AND MOTIVATIONS

A. Preliminaries of CMOPs

Without loss of generality, a CMOP is formulated as follows:

$$\begin{aligned} \text{Minimize} \quad & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))^T \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{S} \\ & g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, p \\ & h_j(\mathbf{x}) = 0, \quad j = p+1, \dots, q \end{aligned} \quad (1)$$

where m denotes the number of objective functions; $\mathbf{x} = (x_1, \dots, x_n)^T$ is an n -dimensional decision vector; n is the number of decision variables; $\mathbf{x} \in \mathcal{S}$, and $\mathcal{S} \subseteq \mathbb{R}^n$ is the search space. $g_i(\mathbf{x})$ and $h_j(\mathbf{x})$ are the i -th inequality and j -th equality constraints, respectively. q denotes the number of constraints.

For a CMOP, the degree of the j -th constraint violation (denoted as $cv_j(\mathbf{x})$) of \mathbf{x} is

$$cv_j(\mathbf{x}) = \begin{cases} \max(0, g_j(\mathbf{x})), & j = 1, \dots, p \\ \max(0, |h_j(\mathbf{x}) - \sigma|), & j = p+1, \dots, q \end{cases} \quad (2)$$

where σ is a small enough positive value to relax the equality constraints into inequality ones. The overall constraint violation (CV) of \mathbf{x} is calculated by

$$CV(\mathbf{x}) = \sum_{j=1}^q cv_j(\mathbf{x}). \quad (3)$$

A solution \mathbf{x} is feasible if $CV(\mathbf{x}) = 0$; otherwise, it is infeasible.

B. Existing Approaches to CMOPs

Based on the basic ideas and foci, existing approaches for CMOPs could be roughly divided into three categories.

1) *Using Auxiliary Evolution*: The first category uses one or more auxiliary problems to assist the evolution of the original problem. Generally, this kind of method performs well on those CMOPs whose CPF is the same as, part of, or near the unconstrained Pareto front (UPF). Liu and Wang [3] proposed a two-phase algorithm that uses the first phase to solve a single-objective optimization problem transferred from the original CMOP to assist in solving the original problem in the second phase. Li *et al.* [4] proposed a two-archive algorithm that uses a convergence-oriented archive to store feasible solutions and pursue convergence and meanwhile uses a diversity-oriented archive to store solutions with good diversity to assist in searching the objective space. Tian *et al.* [12] proposed a new coevolutionary framework that evolves a population for the unconstrained helper problem to assist the original problem, besides, the new framework performs independent mating to avoid generating useless offspring. Jiao *et al.* [13] proposed a multi-tasking framework that conducts an auxiliary task to search for the CPF in feasible regions derived from the original problem through an ε -constrained technique.

2) *Balancing Convergence and Feasibility*: The second category solves CMOPs by balancing the convergence (i.e., approximation to the CPF) and the feasibility (i.e., satisfaction of constraints). Generally, this kind of method could perform well on CMOPs whose feasible regions are relatively large so the switches of preference between constraints and objectives need not be complicated. Tian *et al.* [5] proposed an auto-switched two-stage algorithm that uses one stage for the unconstrained helper problem and the other for the original problem and automatically switches between these two stages according to the state of the population. Liang *et al.* [6] proposed utilizing the relationship between the CPF and the UPF by learning the distribution of the CPF and UPF to balance convergence and feasibility. Yu *et al.* [14] proposed to dynamically adjust the selection preference between convergence and feasibility during the evolutionary process considering the state of the current population. Ma *et al.* [15] proposed a new fitness evaluation function that automatically adjusts the weights of convergence and feasibility according to the current population state to balance the optimization of objectives and constraints.

3) *Utilizing Infeasible Solutions by CHTs*: The third category utilizes valuable infeasible solutions by improving or developing new CHTs to assist the search for the CPF. This kind of method could perform well in dealing with CMOPs with complex constraints and large infeasible regions. However, most CHTs were labor-intensively tailored, and thus, they might be overfitting to some benchmark problems but not applicable to others. Zhou *et al.* [16] proposed an infeasible solutions diversity maintenance ε -constrained handling method that utilizes infeasible solutions with good diversity to search in the objective space. Yuan *et al.* [9] proposed using valuable infeasible solutions determined by a comprehensive criterion that evaluates the potential value of each infeasible solution from different aspects. Fan *et al.* [7] proposed a push-pull search framework that pushes the population toward the

UPF by ignoring constraints in the pushing stage and pulls it back to the CPF in the pull stage using an ε -constrained technique. Ma and Wang [1] proposed a shift-based penalty function that first shifts the positions of infeasible solutions according to the near feasible solutions, and then, the shifted infeasible solutions are penalized based on their constraint violations. Sun *et al.* [8] proposed a new constraint relaxation strategy by adopting different constraint relaxations in feasible sub-population, semi-feasible sub-population, and infeasible sub-population.

C. Motivations

As mentioned above, specific and complicated strategies and techniques might be suitable for some problem types but not applicable to other types, resulting in a lack of versatility². If we need to solve a real-world CMOP whose features and types are unknown, it is hard to determine which strategy or technique to use. Some strategies and techniques might even be too complicated to use. Therefore, in this paper, we investigated the properties of a promising region and the effectiveness of searching evenly in it for CMOPs, aiming to develop a simple but versatile approach that could suit CMOPs with different features and challenges.

Fig. 1 illustrates the basic concepts of our proposed promising region definition. As shown in Fig. 1(a), if we obtain a feasible solution, the non-dominated region, marked as the region within the dashed lines, is named the promising region. If we obtain more feasible solutions, the size of the promising region could be reduced. In the following, we prove that the CPF must be contained in the promising region.

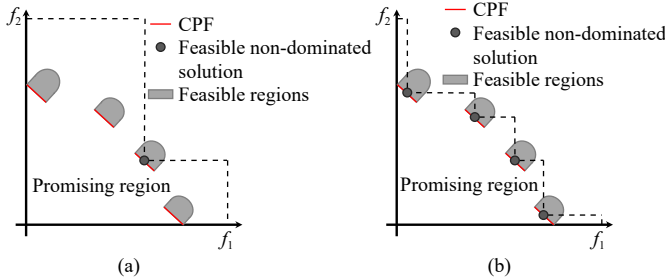


Fig. 1. Illustration of the proposed promising region concept. (a) Promising region determined by one feasible non-dominated solution; (b) Promising region determined by multiple feasible non-dominated solutions.

First, we introduce the basic definitions of constrained multi-objective optimization.

1) *Pareto Dominance*: For two solutions $\mathbf{x}^1, \mathbf{x}^2 \in \mathbb{S}$. \mathbf{x}^1 is said to Pareto dominate \mathbf{x}^2 (denoted as $\mathbf{x}^1 < \mathbf{x}^2$), if and only if $f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2)$ for $\forall i \in \{1, \dots, m\}$ and $f_i(\mathbf{x}^1) < f_i(\mathbf{x}^2)$ for $\exists i \in \{1, \dots, m\}$;

2) *Constrained Pareto Set (CPS)*: A set that for $\forall \mathbf{x}^* \in CPS$, $\nexists \mathbf{x} \in \mathbb{S}$ such that $\phi(\mathbf{x}) = 0$ & $\mathbf{x} < \mathbf{x}^*$;

3) *Constrained Pareto Front (CPF)*: $CPF = \{\mathbf{F}(\mathbf{x}) | \mathbf{x} \in CPS\}$.

Then, given \mathcal{PR} is the point set of the promising region

determined by a solution \mathbf{x} , and \mathcal{S} is the point set distributed on the CPF. $\mathcal{PR} \in \mathbb{S}$, $\mathcal{S} \in \mathbb{S}$.

Suppose $\mathcal{S} \notin \mathcal{PR}$, i.e., not all CPF is in the promising region.

$$\begin{aligned} &\because \mathcal{S} \notin \mathcal{PR}, \\ &\therefore \exists \mathbf{y} \in \mathcal{S}, \mathbf{x} < \mathbf{y}, \\ &\therefore \forall \mathbf{x}^* \in CPS, \nexists \mathbf{x} \in \mathbb{S} \text{ such that } CV(\mathbf{x}) = 0 \text{ \& } \mathbf{x} < \mathbf{x}^*, \\ &\therefore \nexists \mathbf{x}, \mathbf{x} < \mathbf{y}, \\ &\therefore \text{they conflict with each other,} \\ &\therefore \mathcal{S} \in \mathcal{PR}. \end{aligned}$$

Therefore, if we can search evenly in the promising region, the CPF could be obtained no matter how the feasible regions distribute or what the relationship between CPF and UPF is. It is important to note that the term even search (search evenly) in this work means maintaining a set of well-spread solutions in the promising region that covers as many of its areas as possible.

Fig. 2 provides an artificial scene where feasible regions are discontinuous and randomly distributed. The dot represents the old feasible solution and the star represents the new feasible non-dominated solutions. In Fig. 2(a), the old solution will be deleted because it is dominated by the new feasible solution. In Fig. 2(b), two more feasible solutions can be used to determine the promising regions because they are all feasible and non-dominated by each other. As shown in Fig. 2, when the evolution processes, more feasible solutions can be obtained (i.e., the situation of Fig. 2(a)) and the convergence of these solutions can be improved (i.e., the situation of Fig 2(b)). Then, the promising region shrinks so searching evenly in it becomes easier as the evolution proceeds.

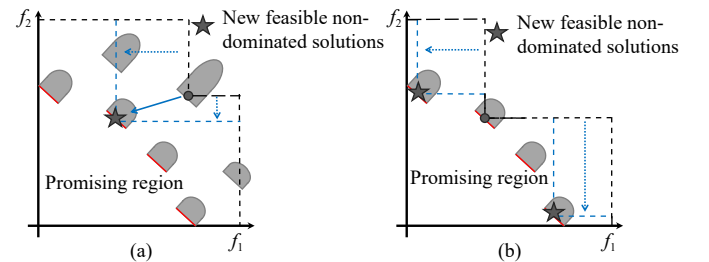


Fig. 2. Illustration of the variations of the promising region during the evolutionary process. (a) Variation of promising region when the convergence of feasible non-dominated solution improves; (b) Variation of promising region when the number of feasible non-dominated solutions increases.

Based on the above considerations, we proposed a new method that conducts an even search in the promising region for handling CMOPs, which will be detailed in the next section.

III. PROPOSED METHODS

A. Determine Solutions in Promising Region

In this part, we first present the method to determine whether a solution is in the promising region. As illustrated and proved in Section II-C, if a solution is not dominated by the obtained feasible solutions, it is in the promising region. Therefore, we use the method presented in Algorithm 1 to achieve this. In our method, a vector \mathbf{PR} is initialized by ones

² Evidence can be found in Section IV, where we perform various experiments to test performances of different CMOPs on different CMOPs. The results clearly reveal that some CMOPs using a specific or complicated strategy (e.g., BiCo and MFOSPEA2) perform very well on one kind of CMOP but cannot handle other features of CMOPs.

with size \mathcal{S} (Line 1). Then, for every solution in \mathcal{S} , if any solution of the obtained feasible solution set \mathcal{F} dominates the solution in \mathcal{S} (Line 4), its mark in **PR** is set to zero (Line 5). Therefore, those solutions in \mathcal{S} marked ones are in the promising region. On the contrary, solutions marked zero are not in the promising region. Since the input of this procedure contains the obtained feasible solution set \mathcal{F} , and \mathcal{F} is updated every generation which will be introduced in Section III-D, the shrink of the promising region can be achieved.

Algorithm 1 Identification of Solutions in PR

Require: \mathcal{S} (solution set), \mathcal{F} (obtained feasible non-dominated solution set)

Output: **PR** (the mark whether in PR)

```

1: PR  $\leftarrow$  Initialize the marks as ones;
2: for each solution  $\mathbf{x}$  in  $\mathcal{F}$  do
3:   for each solution  $\mathbf{y}$  in  $\mathcal{S}$  do
4:     if  $\mathbf{x} < \mathbf{y}$  then
5:       PR $_{\mathbf{x}} \leftarrow 0$ ;
6:     end if
7:   end for
8: end for
9: return PR

```

B. Even Search

In this part, we introduce the proposed even search strategy which is presented in Algorithm 2. This method aims to preserve valuable solutions well-spread in the promising region, achieving an even distribution to assist the search for the CPF. First, all solutions in the promising region are preselected to a temporary solution set \mathcal{S} (Lines 1 and 2). Then, the first proportion of the population for the next generation \mathcal{P}_1 is selected (Lines 3–9). The first proportion contains non-dominated solutions with a good degree of diversity. The diversity is guaranteed by deleting redundant solutions with smaller closest distances. The closest distance of any solution \mathbf{x} is calculated by

$$d = \min_{\mathbf{y} \in \mathcal{S}, \mathbf{y} \neq \mathbf{x}} \text{dist}(\mathbf{x}, \mathbf{y}) \quad (4)$$

where $\text{dist}(\mathbf{x}, \mathbf{y})$ is the Euclidean distance in the objective space.

Then, the second proportion of the population for the next generation \mathcal{P}_2 is selected (Lines 10–17). The second proportion contains feasible solutions in a relaxed manner. The relaxation factor ε is calculated as

$$\varepsilon_g = CV_{\max} \times \left(1 - \frac{g}{G_{\max}}\right)^\eta \quad (5)$$

where CV_{\max} is the maximum CV found by the population so far; g is the current generation; G_{\max} is the maximum generation, and η is a parameter to control the reducing speed of ε . In this equation, only one parameter η is used. It is set to 2, and the effectiveness of η will be studied in the experiments. The design of ε includes the following underlying considerations. Using CV_{\max} found so far can guarantee a large enough initial value of ε . As the evolution proceeds, $(1 - \frac{g}{G_{\max}})$ reduces from 1 to 0. Moreover, $\eta = 2$ generates a gradually decreasing gradient of ε related to $(1 - \frac{g}{G_{\max}})$. Therefore, at the earlier stage, ε

reduces faster, while at the latter stage, ε reduces slower. Therefore, at the earlier stage, ε is a large value to produce a better exploration of feasible regions. Then, at the latter stage, ε is a small value closer to 0 to ensure the exploitation of feasible regions. With a relaxation factor, all solutions that have a CV value no more than ε are preselected (Lines 10–12). Also, redundant solutions with low diversity are deleted (Lines 13–17). Finally, the closest distance values of \mathcal{P}_1 and the CV values of \mathcal{P}_2 are used for mating selections (Lines 19 and 20).

Algorithm 2 Even Search

Require: N (required size), \mathcal{S} (candidate solution set), \mathcal{F} (obtained feasible solution set), g (current generation), CV_{\max} (maximum constraint violation found so far)

Output: \mathcal{P} (population for next generation), **FP**₁ (fitness values of first proportion of population), **FP**₂ (fitness values of second proportion of population)

```

1: PR  $\leftarrow$  Determine whether solutions of  $\mathcal{F}$  are in promising region by Algorithm 1;
2:  $\mathcal{S} \leftarrow$  Preselect solutions satisfy PR = 1;
3: F  $\leftarrow$  Perform fast non-dominated sorting on  $\mathcal{S}$ ;
4:  $\mathcal{P}_1 \leftarrow$  Select solutions in  $\mathcal{S}$  that satisfy F = 1;
5: while  $|\mathcal{P}_1| > N$  do
6:   D  $\leftarrow$  Calculate the closest distance between each solution to its closest solution using (4);
7:    $\mathbf{x} = \arg \min_{\mathbf{x} \in \mathcal{P}_1} \mathbf{D}_{\mathbf{x}}$ ;
8:   Delete solution  $\mathbf{x}$  with minimum closest distance;
9: end while
10:  $\varepsilon \leftarrow$  Update the value of  $\varepsilon$  using (5);
11: CV  $\leftarrow$  Calculate the constraint violation values of solutions in  $\mathcal{S}$ ;
12:  $\mathcal{P}_2 \leftarrow$  Select solutions in  $\mathcal{S}$  that satisfy CV  $\leq \varepsilon$ ;
13: while  $|\mathcal{P}_2| > N$  do
14:   D  $\leftarrow$  Calculate the closest distance between each solution to its closest solution using (4);
15:    $\mathbf{x} = \arg \min_{\mathbf{x} \in \mathcal{P}_2} \mathbf{D}_{\mathbf{x}}$ ;
16:   Delete solution  $\mathbf{x}$  with minimum closest distance;
17: end while
18:  $\mathcal{P} \leftarrow \mathcal{P}_1 \cup \mathcal{P}_2$ 
19: FP1  $\leftarrow$  The closest distance values of  $\mathcal{P}_1$ ;
20: FP2  $\leftarrow$  The constraint violation values of  $\mathcal{P}_2$ ;
21: return  $\mathcal{P}$ 

```

The benefits of using these two kinds of solutions can be illustrated in Fig. 3. As shown in Fig. 3(a), darker diamonds located on the UPF are preferred, so they could accelerate the approximation to the CPF and help to jump out of local feasible regions. If the UPF overlaps with the CPF, \mathcal{P}_1 could even help to find feasible solutions on the CPF. As shown in Fig. 3(b), darker triangles are preferred. Therefore, those solutions near, or even in, the detected or undetected feasible regions could be preserved. They could help to search the detected feasible regions and detect new feasible regions.

The reasons for developing such a method are as follows. First, an unknown CMOP might have a very large objective space, as well as a large promising region, so we have to further limit the even search to a smaller region. That is to say, we have to design some rules that preserve valuable, rather than all, solutions in the promising region. Second, the pre-

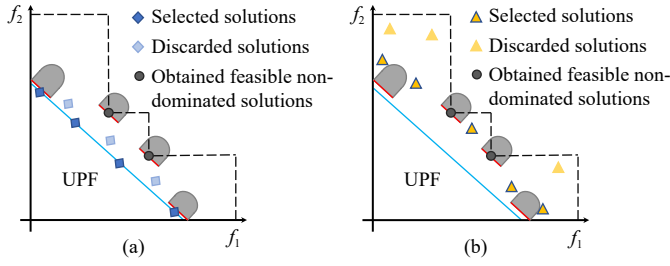


Fig. 3. An illustration of the benefits of the proposed even search method. In Fig. 3(a), the darker diamond solutions are preserved to accelerate the convergence and jump out of local feasible regions; in Fig. 3(b), the darker triangles are preserved to detect feasible regions.

ferred two kinds of solutions are both useful for detecting and searching the CPF. Third, in order to avoid overfitting, we make these rules very simple and extensible.

C. Mating Selection

In this part, we introduce the mating selection method based on the tournament selection mechanism. The pseudo code is presented in Algorithm 3. As mentioned in the last part, we output $-\mathbf{FP}_1$ and \mathbf{FP}_2 as the criteria used in the mating selections of \mathcal{P}_1 and \mathcal{P}_2 , respectively. Therefore, in the mating selection of \mathcal{P}_1 , we select the solution with a larger closest distance from two randomly selected solutions each time, until the size of the mating pool reaches N . Similarly, for \mathcal{P}_2 , we select the solution with a smaller CV value from two each time until the size of the mating pool reaches N .

Algorithm 3 Mating Selection Based on Tournament

Require: N (required size), \mathcal{P} (candidate solution set), \mathbf{F} (fitness criteria)

Output: \mathcal{M} (selected mating pool)

```

1:  $\mathcal{M} \leftarrow \emptyset$ ;
2: while  $|\mathcal{M}| \leq N$  do
3:    $\mathbf{x}, \mathbf{y} \leftarrow$  Randomly select two solution from  $\mathcal{P}$ ;
4:   if  $\mathbf{F}_x > \mathbf{F}_y$  then
5:      $\mathcal{M} \leftarrow \mathcal{M} \cup \mathbf{y}$ ;
6:   els if  $\mathbf{F}_x < \mathbf{F}_y$  then
7:      $\mathcal{M} \leftarrow \mathcal{M} \cup \mathbf{x}$ ;
8:   else
9:      $\mathcal{M} \leftarrow$  Randomly add  $\mathbf{x}$  or  $\mathbf{y}$  to  $\mathcal{M}$ ;
10:  end if
11: end while
12: return  $\mathcal{M}$ 

```

Selecting non-dominated solutions with better diversity as mating parents could improve the search near the UPF. Also, selecting relaxed feasible solutions with smaller CV values as mating parents could enhance the search for feasible regions. Besides, the Tournament Selection mechanism that uses two randomly selected solutions each time could enhance the possibility that worse solutions might also be selected, so the exploration ability could be enhanced.

D. Update the Archive

In this part, we introduce the method used to update the archive that stores obtained feasible solutions. The pseudo

code is presented in Algorithm 4. The archive has two functions. First, it is used to store obtained feasible solutions to determine whether solutions are in the promising region. Second, the archive is used as the final output solution set of CMOES. Therefore, the archive must be updated at each generation and the convergence and diversity of feasible solutions must be enhanced.

Algorithm 4 Update the Archive

Require: N (required size), \mathcal{P} (candidate solution set), \mathcal{A} (archive at last generation)

Output: \mathcal{A} (archive for next generation), \mathbf{FA} (fitness values of archive solutions)

```

1:  $\mathbf{FN} \leftarrow$  Perform the non-dominated sorting using constrained dominance principle in NSGA-II and obtain the front number of each solution in  $\mathcal{P}$ ;
2:  $\mathcal{A} \leftarrow$  Add solutions in  $\mathcal{P}$  that satisfy  $\mathbf{FN} = 1$  to  $\mathcal{A}$ ;
3: if  $|\mathcal{A}| > N$  then
4:    $\mathbf{FA} \leftarrow$  Calculate the Crowding distance of NSGA-II values of solutions in  $\mathcal{A}$ ;
5:    $\mathcal{A}, \mathbf{FA} \leftarrow$  Delete redundant solutions using the Crowding distance and save the Crowding distance values of preserved solutions;
6: end if
7: return  $\mathcal{A}, \mathbf{FA}$ 

```

Based on the above considerations, any update mechanism of any MOEA that balances convergence and diversity could be used. In CMOES, we use the mechanism of NSGA-II, a classical and simple MOEA, for updating the archive. Other mechanisms could also be used as a substitute. In the strategy of this paper, we first include all feasible solutions of the population \mathcal{P} . Then, if the size of \mathcal{A} exceeds N , the redundant feasible solutions with worse crowding distance values are deleted.

E. The Framework of CMOES

Based on the above methods, we introduce the framework of CMOES. The general procedure of CMOES is presented in the flowchart shown in Fig. 4. In order to perform the even search method, we need to first find feasible solutions. Therefore, the proposed CMOES contains two stages. In stage I, CMOES evolves the population by a CMOEA and updates the archive by the obtained feasible solutions. It is worth noting that terminating stage I has two conditions. The first condition indicates that stage II starts when at least one feasible solution is found and stored in \mathcal{A} . The second condition $g > \theta \times G_{\max}$ is designed to allocate more computing resources to push the population to approximate the CPF. Then, in stage II, we generate offspring sets for both the archive and the population. Since the archive contains feasible solutions, we generate offspring using it to better exploit feasible regions.

The pseudocode of CMOES is presented in Algorithm 5. One important point that we should note is the procedure of Lines 12–16. In order to provide criteria for mating selections in stage II, we need to calculate \mathbf{FP}_1 , \mathbf{FP}_2 , and \mathbf{FA} at the beginning of stage II. Therefore, we use a flag s to control this step.

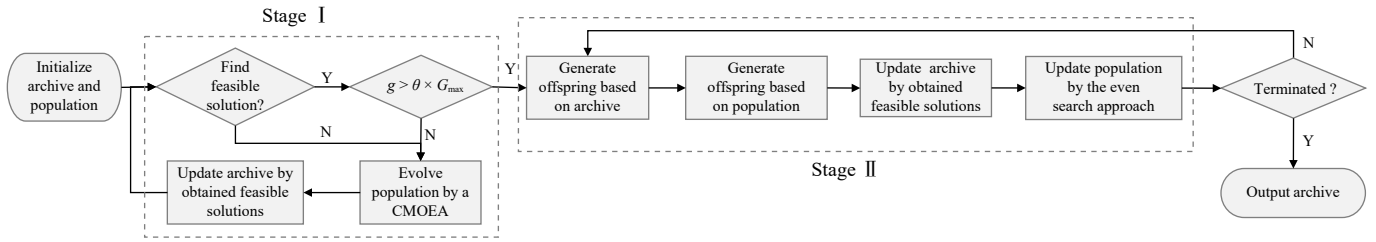


Fig. 4. The overall framework of the proposed CMOES.

Algorithm 5 The Framework of CMOESPR

Require: N (population size), G_{\max} (termination condition), η (evolutionary length for finding feasible solutions)

Output: \mathcal{A} (final archive solution set)

```

1:  $\mathcal{P} \leftarrow$  Generate the initial population randomly;
2:  $CV_{\max} \leftarrow$  Calculate the maximum constraint violation of population;
3:  $\mathcal{A} \leftarrow$  Update the archive by Algorithm 4;
4:  $g \leftarrow$  Set the current generation zero;
5:  $s \leftarrow$  Set the current search stage zero;
6: while  $g < G_{\max}$  do
7:    $g \leftarrow g + 1$ ;
8:   if  $|\mathcal{A}| = 0 \parallel g < \eta \times G_{\max}$  then
9:      $\mathcal{P} \leftarrow$  Evolve the population by a CMOEA;
10:     $\mathcal{A} \leftarrow$  Update the archive by Algorithm 4;
11:   else
12:     if  $s = 0$  then
13:        $s \leftarrow 1$ ;
14:        $\mathbf{FP}_1, \mathbf{FP}_2 \leftarrow$  Calculate the fitness of two proportions of the population;
15:        $\mathbf{FA} \leftarrow$  Calculate the fitness of solutions of the archive;
16:     end if
17:      $\mathcal{MA} \leftarrow$  Tournament selection for the mating pool of archive based on  $\mathbf{FA}$  with size  $|\mathcal{A}|$ ;
18:      $\mathcal{OA} \leftarrow$  Generate offspring by the GA operator based on  $\mathcal{MA}$ ;
19:      $\mathcal{MP}_1 \leftarrow$  Tournament selection for the mating pool of the first proportion of the population based on  $\mathbf{FP}_1$ ;
20:      $\mathcal{OP}_1 \leftarrow$  Generate offspring by the GA operator based on  $\mathcal{MP}_1$ ;
21:      $\mathcal{MP}_2 \leftarrow$  Tournament selection for the mating pool of the second proportion of the population based on  $\mathbf{FP}_2$ ;
22:      $\mathcal{OP}_2 \leftarrow$  Generate offspring by the GA operator based on  $\mathcal{MP}_2$ ;
23:      $\mathcal{O} \leftarrow \mathcal{OA} \cup \mathcal{OP}_1 \cup \mathcal{OP}_2$ ;
24:      $\mathcal{P}, \mathbf{FP}_1, \mathbf{FP}_2 \leftarrow$  Update the population by Algorithm 2;
25:      $\mathcal{A}, \mathbf{FA} \leftarrow$  Update the archive by Algorithm 4;
26:   end if
27:    $CV_{\max} \leftarrow$  Update the maximum constraint violation value found by the population;
28: end while
29: return  $\mathcal{A}$ 

```

F. Computational Complexity

The computational complexity of Algorithm 1 is $O(mN^2)$ since the maximum size of the archive is N . The computa-

tional complexity of Algorithm 2 is $O(mN^2)$ for the fast non-dominated sorting and $O(N)^2$ for calculating the closest distance. The computational complexity of Algorithm 3 is $O(N)$. The computational complexity of Algorithm 4 is $O(mN^2)$ of calculating the crowding distance by NSGA-II. In summary, the computational complexity of CMOES is $O(mN^2)$. However, as the procedure runs sequentially, the practical time complexity could be several times larger.

G. Remarks

The proposed CMOES is different from existing multi-stage-based and multi-population-based CMOEAs, which adopt multiple stages and populations with different CHTs or algorithmic strategies. More specifically, CMOES is not a multi-stage-based CMOEAs because the former stage is adopted only to ensure that at least one feasible solution is obtained to determine the PR. In addition, CMOES is not a multi-population-based CMOEAs because \mathcal{P}_1 and \mathcal{P}_2 are used to maintain two different kinds of promising solutions found in the PR, rather than two populations.

The proposed even search method is also different from the ε -constrained technique (constraint relaxation), which ignores constraints (i.e., CV) to some extent through a relaxation factor and prefers solutions with better convergence (i.e., non-dominated). In even search, all solutions with a CV value smaller than ε , whether dominated or non-dominated, can survive. In addition, the even search method aims to enhance the distribution of these solutions in the PR to explore the CPF. This can provide a more comprehensive search than the constraint relaxation techniques.

Finally, CMOES is different from the methods in [10], [17], which both use constraint relaxation in one of their components (population and archive). In CMOES, the proposed even search method is adopted. Although CMOES also contains a factor ε , it is more simplistic and its function is different. It is used in the even search to determine the second proportion of promising solutions. Comparative studies are conducted and presented in Section IV-F.

IV. EXPERIMENTAL STUDIES

This section presents the experimental studies, including the settings, results, and analysis, on CMOES. All experiments were conducted on PlatEMO [18]³.

A. Experimental Settings

1) *CMOPs for Experiments:* We selected four benchmark

³ Detailed discussions and analyses on all experimental results can be found in the supplementary file at: <https://wewnyin.github.io/wenyingong/Publication/CMOES-supp.pdf>.

TABLE I
A SUMMARY OF STATISTICAL RESULTS OF HV AND IGD+ ON ALL BENCHMARK AND REAL-WORLD PROBLEMS

CMOES vs (+/-/=)	BiCo	CMOEA-MS	MFO-SPEA2	ShiP-A	URCMO	C-TAEA	CCMO	ToP	DSPCMDE	NSGA-II-ToR	CCEA	
DAS-CMOP	HV	0/7/2	1/7/1	0/9/0	0/9/0	1/7/1	0/8/1	2/5/2	0/9/0	0/9/0	0/9/0	2/7/0
	IGD+	0/7/2	1/6/2	0/8/1	0/9/0	1/8/0	0/9/0	1/5/3	0/9/0	0/9/0	0/9/0	1/7/1
LIR-CMOP	HV	0/14/0	2/12/0	0/14/0	4/10/0	1/7/6	0/14/0	1/12/1	0/14/0	2/7/5	0/14/0	2/12/0
	IGD+	0/14/0	2/12/0	0/14/0	5/9/0	1/8/5	0/13/1	1/12/1	0/14/0	3/8/3	0/14/0	2/12/0
LYO	HV	0/3/5	0/7/1	0/6/2	0/8/0	1/7/0	0/8/0	1/7/0	0/7/1	0/7/1	0/8/0	3/3/2
	IGD+	0/3/5	0/7/1	0/5/3	0/8/0	1/7/0	0/8/0	1/7/0	0/7/1	0/7/1	0/8/0	3/3/2
MW	HV	6/5/3	0/10/4	7/2/5	4/6/4	4/5/5	4/9/1	2/2/10	0/13/1	1/13/0	0/14/0	0/8/6
	IGD+	6/4/4	0/10/4	5/1/8	7/5/2	3/5/6	5/7/2	2/2/10	0/13/1	0/13/1	0/14/0	3/7/4
RW-MOP	HV	1/5/8	2/9/3	2/3/9	3/5/6	3/5/6	3/9/2	2/7/5	4/4/6	2/10/2	3/11/0	2/9/3

CMOP test suites and 14 real-world CMOPs to test our proposed CMOES. The four benchmarks include MW [19], LYO [9], LIR-CMOP [20], and DAS-CMOP [21]. The 14 real-world test problems are the former 14 two-objective CMOPs from the test suite for the IEEE CEC 2021 Competition on Real-World Multiobjective Constrained Optimization [22]⁴.

2) *CMOEs for Experiments*: In this paper, we selected 11 state-of-the-art CMOEs for comparison, including methods belonging to all three categories in Section II-B. They are BiCo [23], CMOEA-MS [5], MFO [13] (the MFO-SPEA2 instantiation), ShiP [1] (the ShiP-A instantiation), URCMO [6], C-TAEA [4], CCMO [12], ToP [3], DSPCMDE [14], NSGA-II-ToR [15], and CCEA [9].

3) *Parameter Settings and Genetic Operators*: For algorithms that use the operators of GA, the simulated binary crossover (SBX) and polynomial mutation (PM) were adopted with the following parameter settings:

a) Crossover probability was $p_c = 1$; distribution index was $\eta_c = 20$;

b) Mutation probability was $p_m = 1/n$; distribution index was $\eta_m = 20$.

For algorithms that use the operators of DE, the parameters CR and F in the DE operator were set to 1 and 0.5, respectively.

The population size N was set to 100, the maximum evaluation E_{\max} was 100 000, the maximum generation G_{\max} was E_{\max}/N . This work adopts $\sigma = 10^{-4}$ for the tested problems as the relaxation for equality constraints. All other parameter settings of the comparison methods were the same as in their original literature (i.e., the default settings in PlatEMO). The parameters for our proposed methods include η used to calculate ε , it is set to 2 and the influence of this parameter will be studied in Section IV-E.

4) *Performance Indicators*: Inverted generational distance based on modified distance calculation (IGD+) [24] and hypervolume (HV) [25] were adopted as indicators to evaluate the performance of different algorithms. IGD+ is used because IGD is proved Pareto non complaint [24]. We used two indicators to achieve a sound and fair comparison [26].

Suppose \mathcal{Z} is a set of uniformly distributed points on CPF and \mathcal{A} is the solution set. In IGD+, $d(a_i, z_j)$ is calculated as

$$d(a, z) = \sqrt{\sum_{k=1}^M (\max\{a_k - z_k, 0\})^2} \quad (6)$$

then, IGD+ is calculated as

$$\text{IGD}+(\mathcal{Z}, \mathcal{A}) = \frac{1}{|\mathcal{Z}|} \sum_{j=1}^{|\mathcal{Z}|} \min_{a_i \in \mathcal{A}} d(a_i, z_j) \quad (7)$$

where $d(a_i, z_j)$ is the Euclidean distance between a and z .

A smaller IGD+ value indicates a better performance.

HV [25] measures the volume or hypervolume of the objective space enclosed by the obtained solution set and the predefined reference point \mathbf{z}^r , HV of a solution set \mathcal{A} can be formulated as

$$\text{HV}(\mathcal{A}) = \text{VOL}\left(\cup[z_1, z_1^r] \times \dots \times [z_m, z_m^r]\right) \quad (8)$$

where VOL indicates the Lebesgue measure. A larger HV value indicates better performance obtained.

10 000 uniformly distributed points were sampled on the true CPF for the calculation of IGD+ according to [12]. As for HV, the objective values were first normalized, and then, (1.1, 1.1, ..., 1.1) was adopted as the reference point in the normalized objective space.

5) *Statistical Analysis*: Each algorithm executed over independent 30 runs on each test instance. The mean and standard deviation values of IGD+ and HV were recorded. The Wilcoxon rank-sum test with a significance level of 0.05 was employed to perform the statistical analysis. Additionally, “+”, “−” and “=” were used to show that the results of other algorithms were significantly better than, significantly worse than, and statistically similar to those obtained by our methods to the Wilcoxon test, respectively.

B. On Benchmark CMOPs

This part presents the results and analyses of benchmark CMOPs. A short summary of the statistical results is presented in Table I. In the Supplementary file, more detailed explanations and analyses are provided.

1) *On DAS-CMOPs*: DAS-CMOPs are featured as adjustable difficulties on convergence, diversity, and feasibility, which tests the versatility and generic ability of an algorithm. The statistical results on HV and IGD+ obtained by CMOES and other CMOEs are recorded in Tables S-II and

⁴ Dimensions of m and n of the selected benchmarks are reported in Table S-I in the Supplementary file.

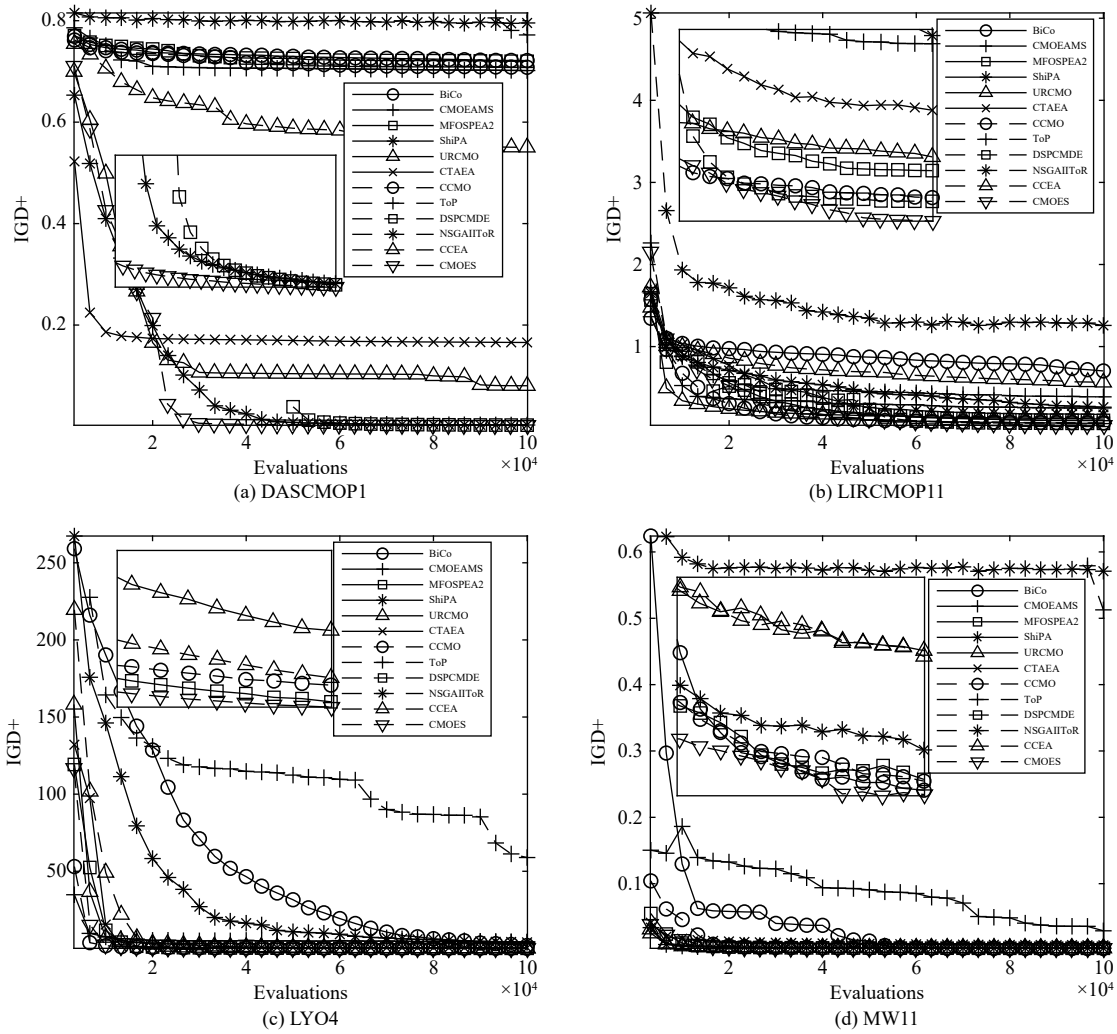


Fig. 5. The convergence profiles on IGD+ of CMOES and other methods on DAS-CMOP1, LIR-CMOP11, LYO4, and MW11 with the median IGD+ values among 30 runs.

S-III in the Supplementary file. The results show that CMOES outperformed all other methods on DAS-CMOP1 to DAS-CMOP6, while CMOEAMS, CCMO, URCMO, and CCEA outperformed CMOES on DAS-CMOP7, DAS-CMOP8, DAS-CMOP9, and DAS-CMOP7-8, respectively. The feasible and non-dominated solution sets obtained by CMOES on each instance of DAS-CMOPs with the median IGD+ values among 30 runs are presented in Fig. S-I in the Supplementary file. It is apparent that CMOES had finally found the CPF and achieved good diversity in each instance. To further compare the results, we depicted feasible and non-dominated solution sets obtained by all methods on DAS-CMOP3 and DAS-CMOP9 in Figs. S-2 and S-3 in the Supplementary file, respectively. It could be found that most of the methods in comparison could not handle these two CMOPs well. Although DSPCMDE could find the CPF of DAS-CMOP3, it performed poorly on DAS-CMOP9. Similarly, although URCMO performed well on DAS-CMOP0, it performed poorly on DAS-CMOP3. However, CMOES could only find most of, not all, the feasible regions on DAS-CMOP7-9, and the diversity performances were not very good. The reason could be that in our update strategy of the archive (which is used as the

final output), the crowding distance of NSGA-II was adopted, but it is not very suitable for three-objective problems. In addition, a three-objective space is much larger, and achieving an even search is much harder. Nevertheless, CMOES could solve all instances of DAS-CMOPs, demonstrating that it has the best versatility. Fig. 5(a) presents the convergence profile on IGD+ of CMOES and other methods on DAS-CMOP1. Compared to other methods, CMOES obtained not only fast convergence speed but also the best final IGD+ value.

2) *On LIR-CMOPs*: LIR-CMOPs are featured as large infeasible regions that propose difficulties for an algorithm to locate very feasible regions, especially for those feasibility-preferred CMOEAs. Tables S-IV and S-V in the Supplementary file present the statistical results on HV and IGD+ obtained by CMOES and other methods, respectively. The results show that ShiP-A performed better than CMOES on LIR-CMOP1-4, DSPCMDE performed better on LIR-CMOP5-6 and LIR-CMOP7, and some CMOEAs performed better on LIR-CMOP13-14. For three-objective LIR-CMOP13-14, the reasons are the same. To further investigate the results, we depicted the feasible and non-dominated solu-

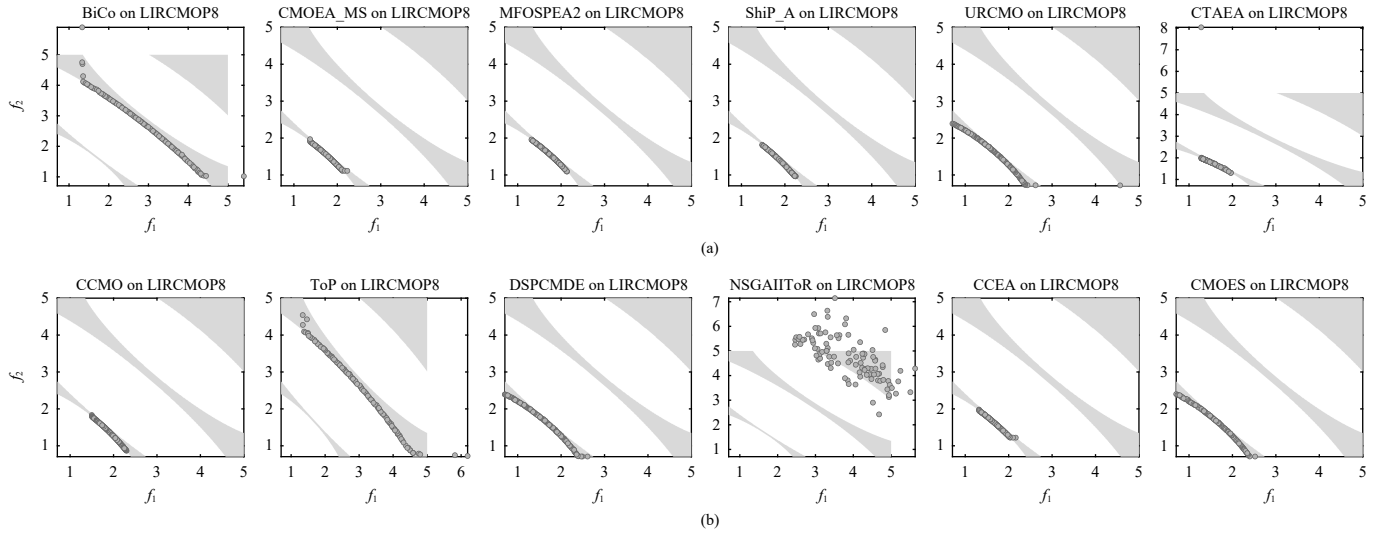


Fig. 6. Feasible and non-dominated solution sets obtained by CMOES and other methods on LIR-CMOP8 with the median IGD+ value among 30 runs. (a) Contains the results of BiCo, CMOEA-MS, MFO-SPEA2, ShiP-A, URCMO, and C-TAEA; (b) Contains the results of CCMO, ToP, DSPCMDE, NSGA-II-ToR, CCEA, and CMOES.

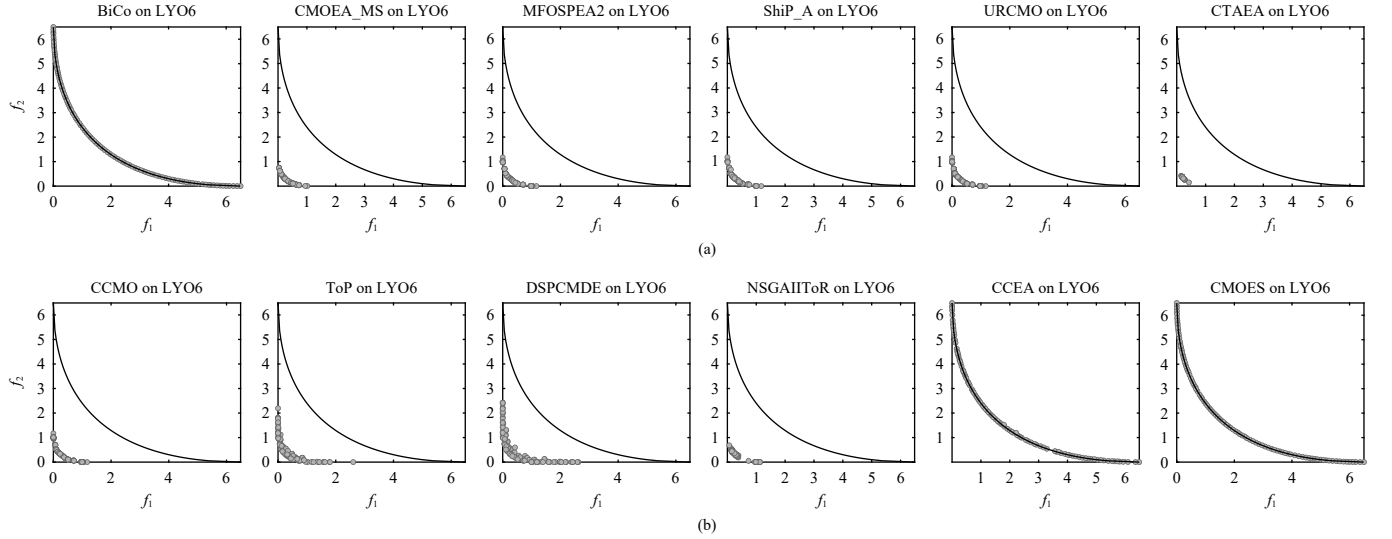


Fig. 7. Feasible and non-dominated solution sets obtained by CMOES and other methods on LYO6 with the median IGD+ value among 30 runs. (a) Contains the results of BiCo, CMOEA-MS, MFO-SPEA2, ShiP-A, URCMO, and C-TAEA; (b) Contains the results of CCMO, ToP, DSPCMDE, NSGA-II-ToR, CCEA, and CMOES.

tion sets obtained by CMOES on each instance of LIR-CMOPs with the median IGD+ values among 30 runs in Fig. S-4 in the Supplementary file. Also, we depicted the feasible and non-dominated solution sets obtained by all methods in Fig. 6. It is apparent that CMOES had found only some parts of the CPF on LIR-CMOP1-4 and the better indicator results on LIR-CMOP5-10 were because CMOES could find the CPF impeded by the large infeasible regions. Although CMOES outperformed most of the methods in comparison, these results reveal that the effectiveness of CMOES in dealing with CMOPs with very small feasible regions has to be improved. In summary, CMOES could handle LIR-CMOPs with large infeasible regions because the even search method could explore the objective space where the small feasible regions locate. Fig. 5(b) presents the convergence profile on IGD+ of CMOES and other methods on LIR-CMOP11. Com-

pared to other methods, CMOES obtained the best final IGD+ value.

3) *On LYOs*: LYOs are a set of special benchmark CMOPs that the initial population arises in the complex infeasible regions below the CPF in the objective space. Therefore, it proposes difficulties for an algorithm to make use of infeasible solutions to search back to the CPF. The statistical results on HV and IGD+ obtained by CMOES and other CMOEAs are reported in Tables S-VI and S-VII in the Supplementary file. CCEA and CMOES obtained the best overall results, except that only URCMO performed better on LYO2. Most CMOEAs in comparison could not handle LYO5-8, except for CCEA and CMOES. However, CMOES performed worse than CCEA, an algorithm proposed for the LYOs. To further analyze the results, we depicted the obtained final solution sets by CMOES on all instances with the median IGD+ val-

ues among 30 runs in Fig. S-6 in the Supplementary file, and the obtained solution sets by all methods on LY06 in Fig. 7. From Fig. 7 we can see that only BiCo, CCEA, and CMOES could find the CPF. From Fig. S-6 in the Supplementary file it can be found that CMOES could not find all the segments of the CPF on LY07-8. Since the distribution of feasible and infeasible regions was not given in LY0, it can be only inferred that the reason is few or no offspring could be generated in the sparse and blank areas when dealing with LY07-8. Nevertheless, CMOES performed well on this special test suite compared to most state-of-the-art CMOEAs, demonstrating the effectiveness of CMOES in using valuable solutions in the promising region. Fig. 5(c) presents the convergence profile on IGD+ of CMOES and other methods on LY04. Compared to other methods, CMOES obtained the best final IGD+ value. Additionally, the extremely bad performances of CMOEAs using a specific or complicated strategy lack versatility.

4) *On MWs*: MWs are featured as four relationships between the CPF and the UPF, which comprehensively represent the problem types in terms of the position and distribution of the CPF relative to the UPF. However, according to our experimental experience, MWs lack difficulties in the distribution of infeasible regions. Therefore, ignoring constraints could well assist the search for the CPF for most instances of MW, especially the Type-I to Type-III problems. However, these constraint-ignoring-assisted CMOEAs performed worse in dealing with Type-IV problems. Therefore, as the statistical results reported in Tables S-VIII and S-IX in the Supplementary file reflected, those CMOEAs using a constraint-ignored auxiliary problem (BiCo, CMOEAs-MS, MFPSPEA2, and CCMO) performed well. Besides, although MFO-SPEA2 and BiCo performed extremely well on MWs, they also performed extremely poorly on DAS-CMOPs, LIR-CMOPs, and LYOs. This indicates that some advanced approaches lack versatility but overfit some kinds of problems. To further analyze the results, we depicted the obtained final solution sets of CMOES on all instances of MW in Fig. S-8 in the Supplementary file. It can be found that CMOES finally obtained very good results in terms of both convergence and diversity, except that it could not find the CPF segment in a very small feasible region on MW10. Fig. 5(d) presents the convergence profile on IGD+ of CMOES and other methods on MW11. It can be found many state-of-the-art CMOEAs could solve this problem very well.

5) *Summaries and Conclusions*: The results of these four benchmark suites show the following phenomena. BiCo and MFO-SPEA2 performed very well on the MW test suite but extremely poorly on the other three. CCEA performed very well on the LY0 test suite but performed poorly on the others. ShiP-A performed very well on LIR-CMOP1-4 but poorly on other instances of LIR-CMOPs, also, it performed well on some Type-I, Type-II, and Type-III instances (MW2, MW6, MW10, and MW13) of MW but performed poorly on others, especially on Type-IV instances. Therefore, these methods are somehow overly fine-tuned and overfitting. As we analyzed

above, although CMOES could not achieve the best indicator values, it can find the CPF and obtain good distribution on it in the face of most instances, especially DAS-CMOPs with adjustable and multiple features. It can be concluded that searching in the promising region can achieve the best versatility.

C. On Real-World CMOPs

In this part, we present the experiment on real-world CMOPs. We selected the former 14 two-objective CMOPs from the IEEE CEC 2021 Competition on Real-World Multi-objective Constrained Optimization test suite in this experiment. It should be noted that the features and difficulties of these problems are unknown, and they include CMOPs from very different application fields. Since the true CPFs are unknown, IGD+ is not applicable, so only HV is used. The statistical results of HV are reported in Table S-X in the Supplementary file. It could be seen that CMOES generally outperformed all the other methods except for ToP. Also, they obtained the largest number of best results (four) among these 14 instances. From the results, we could draw the following three conclusions. First, different algorithms perform very differently among various CMOPs. Second, without complex strategies or techniques, CMOES and ToP performed better than those with complex strategies and techniques on real-world problems. Third, although different algorithms performed differently, CMOES generally obtained better versatility.

D. Ablation Studies on the Effectiveness of Even Search

In this part, we present the ablation study conducted to verify the effectiveness of our proposed even search method. First, we created two variants as follows

- 1) *CMOESP1*: Only the first proportion of the population is preserved and evolved in mating and selection;
- 2) *CMOESP2*: Only the second proportion of the population is preserved and evolved in mating and selection.

The results on HV and IGD+ on the four benchmarks are presented in Tables S-XI–S-XVIII in the Supplementary file. In summary, we could draw the same conclusion as mentioned in Section II-B. Preserving only the first proportion performed better on CMOPs whose CPF is close to or contained by the UPF, on the contrary, preserving only the second proportion performed better on CMOPs whose CPF is far from the UPF, especially LYOs.

Then, we depicted the distributions of the second proportion of the population at the middle stage of evolution (i.e., $G_{\max}/2$) in dealing with DAS-CMOP1, DAS-CMOP5, LIR-CMOP1, LIR-CMOP11, LY01, and MW13 with the median IGD+ values among 30 runs in Fig. 8. From these figures, we could see that the second proportion of the population had even distributions near all segments of the CPF on DAS-CMOP1 and DAS-CMOP5. Also, it covered the most promising regions that contain the CPF on LIR-CMOP1 and LIR-CMOP11. It should be noted that LIR-CMOP1 is rather difficult due to the very small feasible region (long and thin line). It reached the CPF from several directions and finally found

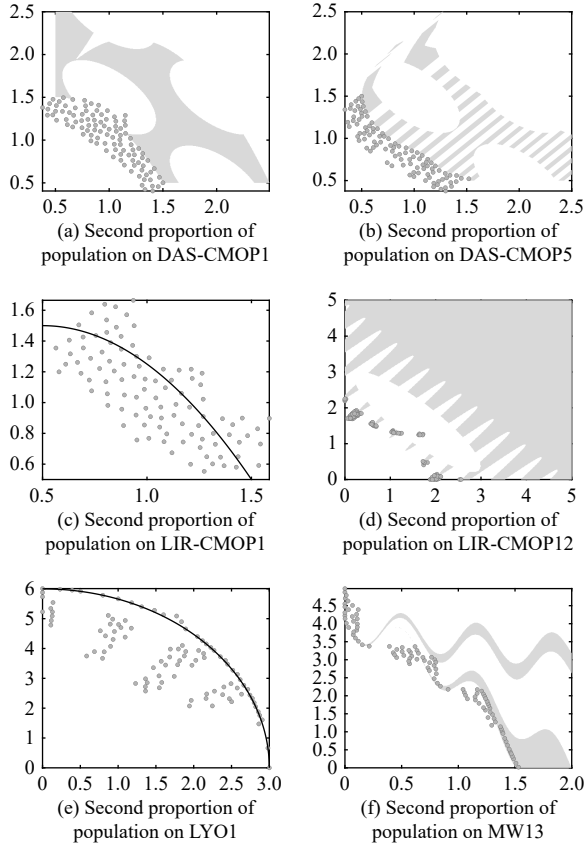


Fig. 8. The distributions of the second proportion of the population at the middle stage of evolution in dealing with DAS-CMOP1, DAS-CMOP5, LIR-CMOP1, LIR-CMOP11, LYO1, and MW13 with the median IGD+ values among 30 runs.

all segments of the CPF on LYO1. It also had good distribution near all regions that the CPF is in on MW13, including $f_1 \in [0, 1]$, the CPF segments covered by very small feasible regions. In summary, the second proportion of the population could cover most of the CPF segments and help to detect new feasible regions in dealing with these CMOPs with very different features and challenges, revealing that our proposed even search method is versatile and robust.

Therefore, from the above results and analysis, we have demonstrated the effectiveness of the proposed even search method. The first proportion of the population could enhance the convergence and suit some kinds of CMOPs, while the second could help to solve other kinds of CMOPs.

E. Parameter Analyses

Since our proposed methods contain two important parameters, η used to calculate ε and θ used to control the length of the former stage, we also conducted experiments on studying the sensitivity of these parameters. Specifically, we changed η to $1/2$, 1 , $3/2$, and 3 respectively, instead of 2 in the original setting, to study the effectiveness of the proposed ε under an incremental ($1/2$), steady (1), gently-decreasing ($3/2$), and rapidly-decreasing (3) gradient. The statistical results on HV and IGD+ on the four benchmarks are presented in Tables S-XIX–S-XXVI in the Supplementary file.

For DAS-CMOPs and LIR-CMOPs, $\eta = 2$ is the most suitable setting for most cases. For LYOs, $\eta = 2$ and $\eta = 3$ perform similarly while $\eta = 2$ has better general performance. For MWs, a larger ε is also more suitable for those instances with very small feasible regions (MW2, MW6, MW10, and MW13). However, $\eta = 2$ is more suitable for LIR-CMOPs. This also demonstrates the announcement that a specified and complicated technique might overfit some problems, and a simple strategy or technique is more applicable for real-world applications.

For DAS-CMOPs, $\theta = 2$ is the best setting. For LIR-CMOPs, a smaller θ results in a shorter former stage and longer even search stage, bringing better performance on LIR-CMOP1-4 which has the smallest feasible region, revealing that our proposed even search strategy can enhance the performance when the feasible regions are considerably small. The setting of θ does not have a significant influence on LYOs and MWs.

F. Comparison With Constraint Relaxation-Based CMOEAs

In this part, the proposed CMOES is compared to the constraint relaxation-based CMOEAs in TriP [10] and PPTA [17] to prove the superiority of even search. HV and IGD+ results on benchmark problems are reported in Table S-XXVII in the Supplementary file. TriP and PPTA outperformed CMOES only on LIR-CMOP1-6, whose feasible regions are considerably small. Similarly, TriP can only outperform CMOES on MW instances with very small feasible regions. Meanwhile, CMOES outperformed them in almost all the other instances, from which the following two observations can be made. TriP and PPTA both adopt a fine-tuned ε -constrained technique, which can enhance the performance on several special instances but fail to adapt to different kinds of problems. On the contrary, CMOES, which uses only simplistic designed techniques and a general even search method, has better versatility.

V. CONCLUSIONS AND FUTURE WORK

In this article, we proposed to search evenly in the promising region to solve CMOPs. We investigated that the promising region contains the CPF and has good properties. Then, we designed an even search method that utilizes two kinds of valuable solutions to search evenly in the promising region for the CPF. A new CMOEA, termed CMOES, is developed based on simple strategies and techniques, and the experiments have demonstrated the effectiveness and the best versatility of our proposed methods on both benchmark and real-world CMOPs. The results prove that a specified, especially complicated, strategy or technique might overfit some problems with specific properties, resulting in a lack of versatility. On the contrary, methods that use simple strategies and techniques could be more applicable to real-world problems whose features and difficulties are unknown.

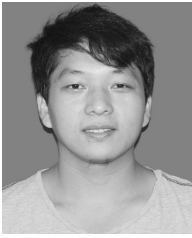
However, the proposed CMOES also has some limitations that need to be improved. The ineffectiveness of the proposed CMOES on LYO7-8 needs further investigation. It indicates that the proposed ε -constrained technique is still sensitive to

the parameters and design. Besides, how to better locate feasible regions while searching evenly in the promising region is expected referring to Fig. 8(f), especially when dealing with CMOPs with very small feasible regions. Furthermore, using machine learning techniques to achieve an even search in the promising region is worth trying [27]–[29]. Last but not least, extending the even search method to other types of multi-objective optimization problems to enhance the performances of existing algorithms is also valuable [30], [31].

The codes of CMOES can be obtained from the authors upon request.

REFERENCES

- [1] Z. Ma and Y. Wang, “Shift-based penalty for evolutionary constrained multiobjective optimization and its application,” *IEEE Trans. Cyber.*, vol. 53, no. 1, pp. 18–30, Jan. 2023.
- [2] Y. Hua, Q. Liu, K. Hao, and Y. Jin, “A survey of evolutionary algorithms for multi-objective optimization problems with irregular Pareto fronts,” *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 2, pp. 303–318, 2021.
- [3] Z. Liu and Y. Wang, “Handling constrained multiobjective optimization problems with constraints in both the decision and objective spaces,” *IEEE Trans. Evolutionary Computation*, vol. 23, no. 5, pp. 870–884, 2019.
- [4] K. Li, R. Chen, G. Fu, and X. Yao, “Two-archive evolutionary algorithm for constrained multiobjective optimization,” *IEEE Trans. Evolutionary Computation*, vol. 23, no. 2, pp. 303–315, 2019.
- [5] Y. Tian, Y. Zhang, Y. Su, X. Zhang, K. C. Tan, and Y. Jin, “Balancing objective optimization and constraint satisfaction in constrained evolutionary multiobjective optimization,” *IEEE Trans. Cyber.*, vol. 52, no. 9, pp. 9559–9572, Sept. 2022.
- [6] J. Liang, K. Qiao, K. Yu, B. Qu, C. Yue, W. Guo, and L. Wang, “Utilizing the relationship between unconstrained and constrained Pareto fronts for constrained multiobjective optimization,” *IEEE Trans. Cyber.*, vol. 53, no. 6, pp. 3873–3886, Jun. 2023.
- [7] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, and E. Goodman, “Push and pull search for solving constrained multi-objective optimization problems,” *Swarm and Evolutionary Computation*, vol. 44, pp. 665–679, 2019.
- [8] Z. Sun, H. Ren, G. G. Yen, T. Chen, J. Wu, H. An, and J. Yang, “An evolutionary algorithm with constraint relaxation strategy for highly constrained multiobjective optimization,” *IEEE Trans. Cyber.*, vol. 53, no. 5, pp. 3190–3204, May 2023.
- [9] J. Yuan, H.-L. Liu, and Z. He, “A constrained multi-objective evolutionary algorithm using valuable infeasible solutions,” *Swarm and Evolutionary Computation*, vol. 68, p. 101020, 2022.
- [10] F. Ming, W. Gong, L. Wang, and C. Lu, “A tri-population based coevolutionary framework for constrained multi-objective optimization problems,” *Swarm and Evolutionary Computation*, vol. 70, p. 101055, 2022.
- [11] D. Wolpert and W. Macready, “No free lunch theorems for optimization,” *IEEE Trans. Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.
- [12] Y. Tian, T. Zhang, J. Xiao, X. Zhang, and Y. Jin, “A coevolutionary framework for constrained multiobjective optimization problems,” *IEEE Trans. Evolutionary Computation*, vol. 25, no. 1, pp. 102–116, 2021.
- [13] R. Jiao, B. Xue, and M. Zhang, “A multiform optimization framework for constrained multiobjective optimization,” *IEEE Trans. Cyber.*, vol. 53, no. 8, pp. 5165–5177, Aug. 2023.
- [14] K. Yu, J. Liang, B. Qu, Y. Luo, and C. Yue, “Dynamic selection preference-assisted constrained multiobjective differential evolution,” *IEEE Trans. Systems, Man, and Cyber.: Systems*, vol. 52, no. 5, pp. 2954–2965, 2022.
- [15] Z. Ma, Y. Wang, and W. Song, “A new fitness function with two rankings for evolutionary constrained multiobjective optimization,” *IEEE Trans. Systems, Man, and Cyber.: Systems*, vol. 51, no. 8, pp. 5005–5016, 2021.
- [16] J. Zhou, J. Zou, J. Zheng, S. Yang, D. Gong, and T. Pei, “An infeasible solutions diversity maintenance epsilon constraint handling method for evolutionary constrained multiobjective optimization,” *Soft Computing*, vol. 25, no. 13, pp. 8051–8062, 2021.
- [17] C. Qin, F. Ming, W. Gong, and Q. Gu, “Constrained multi-objective optimization via two archives assisted push-pull evolutionary algorithm,” *Swarm and Evolutionary Computation*, vol. 75, p. 101178, 2022.
- [18] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, “PlatEMO: A MATLAB platform for evolutionary multi-objective optimization,” *IEEE Computational Intelligence Magazine*, vol. 12, pp. 73–87, 2017.
- [19] Z. Ma and Y. Wang, “Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons,” *IEEE Trans. Evolutionary Computation*, vol. 23, no. 6, pp. 972–986, 2019.
- [20] Z. Fan, W. Li, X. Cai, H. Huang, Y. Fang, Y. Yugen, J. Mo, C. Wei, and E. Goodman, “An improved epsilon constraint-handling method in MOEA/D for CMOPs with large infeasible regions,” *Soft Computing*, vol. 23, pp. 12491–12510, 2019.
- [21] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, and E. Goodman, “Difficulty adjustable and scalable constrained multiobjective test problem toolkit,” *Evolutionary Computation*, vol. 28, no. 3, pp. 339–378, 2020.
- [22] A. Kumar, G. Wu, M. Z. Ali, Q. Luo, R. Mallipeddi, P. N. Suganthan, and S. Das, “A benchmark-suite of real-world constrained multiobjective optimization problems and some baseline results,” *Swarm and Evolutionary Computation*, vol. 67, p. 100961, 2021.
- [23] Z.-Z. Liu, B.-C. Wang, and K. Tang, “Handling constrained multiobjective optimization problems via bidirectional coevolution,” *IEEE Trans. Cyber.*, vol. 52, no. 10, pp. 10163–10176, Oct. 2022.
- [24] H. Ishibuchi, R. Imada, N. Masuyama, and Y. Nojima, “Comparison of hypervolume, IGD and IGD+ from the viewpoint of optimal distributions of solutions,” in *Proc. Evolutionary Multi-Criterion Optimization*, 2019, pp. 332–345.
- [25] E. Zitzler and L. Thiele, “Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach,” *IEEE Trans. Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.
- [26] H. Ishibuchi, L. M. Pang, and K. Shang, “Difficulties in fair performance comparison of multi-objective evolutionary algorithms [research frontier],” *IEEE Computational Intelligence Magazine*, vol. 17, no. 1, pp. 86–101, 2022.
- [27] Z. Lv, L. Wang, Z. Han, and J. Zhao, “Surrogate-assisted particle swarm optimization algorithm with Pareto active learning for expensive multiobjective optimization,” *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 3, pp. 838–849, 2019.
- [28] Y. Tian, L. Si, X. Zhang, K. C. Tan, and Y. Jin, “Local modelbased Pareto front estimation for multiobjective optimization,” *IEEE Trans. Systems, Man, and Cyber.: Systems*, vol. 53, no. 1, pp. 623–634, Jan. 2023.
- [29] Y. Tian, H. Chen, H. Ma, X. Zhang, K. Tan, and Y. Jin, “Integrating conjugate gradients into evolutionary algorithms for large-scale continuous multi-objective optimization,” *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 10, p. 1801–1817, 2022.
- [30] W. Li, X. Yao, K. Li, R. Wang, and T. Zhang, “Coevolutionary framework for generalized multimodal multi-objective optimization,” *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 7, pp. 1544–1556, 2022.
- [31] Y. Tian, Y. Feng, X. Zhang, and C. Sun, “A fast clustering based evolutionary algorithm for super-large-scale sparse multi-objective optimization,” *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 4, pp. 1048–1063, 2023.



Fei Ming received the B.Sc. degree in computer science from China University of Geosciences in 2019. He is currently a Ph.D. candidate with the School of Computer Science, China University of Geosciences. His current research interests include evolutionary multiobjective optimization methods and their applications.



Wenyin Gong (Member, IEEE) received the B.Eng., M.Eng. and Ph.D. degrees in computer science from China University of Geosciences, in 2004, 2007 and 2010, respectively. He is currently a Professor with School of Computer Science, China University of Geosciences. His research interests include evolutionary algorithms, evolutionary optimization, and their applications.

Professor Gong has published over 80 research papers in journals and international conferences. He served as a Referee for over 30 international journals, such as *IEEE Transactions on Evolutionary Computation*, *IEEE Transactions on Cybernetics*, *IEEE Transactions on Systems, MAN, and Cybernetics: Systems*, *IEEE Computational Intelligence Magazine*, *ACM Transactions on Intelligent Systems and Technology*, *Information Sciences*, *European Journal of Operational Research*, *Applied Soft Computing*, *Journal of Power Sources*, etc. He currently serves as Associate Editor of *Expert Systems With Applications*, *Inter-*

national Journal of Bio-Inspired Computation, *Memetic Computing*, etc.



Yaochu Jin (Fellow, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in automatic control from Zhejiang University, in 1988, 1991 and 1996, respectively, and the Dr.-Ing. degree in neuroinformatics from Ruhr University Bochum, Germany in 2001. He joined the Westlake University in October 2023 as a Chair Professor of AI, leading the Trustworthy and General AI Laboratory. He was an Alexander von Humboldt Professor for AI with the Faculty of Technology, Bielefeld University, Germany from 2021 to 2023, and a Distinguished Chair, Professor in Computational Intelligence with the Department of Computer Science, University of Surrey, UK. He was a “Finland Distinguished Professor” with the University of Jyväskylä, Finland, “Changjiang Distinguished Visiting Professor” with Northeastern University, China, and “Distinguished Visiting Scholar”, University of Technology Sydney, Australia. His main research interests include multi-objective and data-driven evolutionary optimization, evolutionary multi-objective learning, trustworthy AI, and evolutionary developmental AI.

Professor Jin is presently the President of the IEEE Computational Intelligence Society and the Editor-in-Chief of *Complex & Intelligent Systems*. He is the recipient of the 2018, 2021 and 2023 IEEE Transactions on Evolutionary Computation Outstanding Paper Award, and the 2015, 2017, and 2020 IEEE Computational Intelligence Magazine Outstanding Paper Award. He was named by the Clarivate as “a Highly Cited Researcher” from 2019 to 2023 consecutively. He is a Member of Academia Europaea and Fellow of IEEE.