This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Handling Constrained Multiobjective Optimization Problems via Bidirectional Coevolution

Zhi-Zhong Liu, Bing-Chuan Wang, and Ke Tang, *Senior Member, IEEE*

***Abstract*—Constrained multiobjective optimization problems (CMOPs) involve both conflicting objective functions and various constraints. Due to the presence of constraints, CMOPs' Pareto-optimal solutions are very likely lying on constraint boundaries. The experience from the constrained single-objective optimization has shown that to quickly obtain such an optimal solution, the search should surround the boundary of the feasible region from both the feasible and infeasible sides. In this article, we extend this idea to cope with CMOPs and, accordingly, we propose a novel constrained multiobjective evolutionary algorithm with *bi*directional *co*evolution, called BiCo. BiCo maintains two populations, that is: 1) the main population and 2) the archive population. To update the main population, the constraint-domination principle is equipped with an NSGA-II variant to move the population into the feasible region and then to guide the population toward the Pareto front (PF) from the feasible side of the search space. While for updating the archive population, a nondominated sorting procedure and an angle-based selection scheme are conducted in sequence to drive the population toward the PF within the infeasible region while maintaining good diversity. As a result, BiCo can get close to the PF from two complementary directions. In addition, to coordinate the interaction between the main and archive populations, in BiCo, a restricted mating selection mechanism is developed to choose appropriate mating parents. Comprehensive experiments have been conducted on three sets of CMOP benchmark functions and six real-world CMOPs. The experimental results suggest that BiCo can obtain quite competitive performance in comparison to eight state-of-the-art-constrained multiobjective evolutionary optimizers.**

***Index Terms*—Angle-based selection, bidirectional coevolution, constrained multiobjective optimization problems (CMOPs), constraint-handling technique (CHT), evolutionary algorithms.**

I. Introduction

CONSTRAINED multiobjective optimization problems (CMOPs) are frequently encountered in diverse science and engineering disciplines [1]–[3], which involve both conflicting objective functions and various constraints. Without loss of generality, a CMOP can be defined as

$$\begin{aligned} \min \quad & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))^T \mathbb{R}^m \\ \text{s.t.} \quad & g_j(\mathbf{x}) \le 0, \; j = 1, \ldots, q \\ & h_j(\mathbf{x}) = 0, \; j = q+1, \ldots, \ell \\ & \mathbf{x} = (x_1, x_2, \ldots, x_n)^T \in \mathbb{R}^n \end{aligned} \tag{1}$$

where $\mathbf{x}$ refers to an n-dimensional decision vector, $\mathbb{R}^n$ denotes the decision space, $\mathbf{F}(\mathbf{x})$ constitutes m real-valued conflicting objective functions,[1] $\mathbb{R}^m$ represents the objective space, $f_i(\mathbf{x})$ is the ith objective function, and $g_j(\mathbf{x})$ and $h_j(\mathbf{x})$ are the jth inequality constraint and the $(j-q)$th equality constraint, respectively.

In constrained evolutionary optimization, the degree of constraint violation on the jth constraint for $\mathbf{x}$ is calculated as

$$c_j(\mathbf{x}) = \begin{cases} \max\big(0, g_j(\mathbf{x})\big), & \text{if } j \le q \\ \max\big(0, |h_j(\mathbf{x})| - \epsilon\big), & \text{otherwise} \end{cases} \tag{2}$$

where ϵ is a very small positive value (e.g., $\epsilon = 10^{-4}$) to relax the equality constraints. For the degree of constraint violation on all constraints (i.e., overall constraint violations), it is usually computed as

$$\mathrm{CV}(\mathbf{x}) = \sum_{j=1}^{\ell} c_j(\mathbf{x}). \tag{3}$$

$\mathbf{x}$ is feasible if it satisfies $\mathrm{CV}(\mathbf{x}) = 0$; otherwise, $\mathbf{x}$ is infeasible. Then, the feasible space $\mathbb{F}$ can be defined as $\mathbb{F} = \{\mathbf{x} \in \mathbb{R}^n | \mathrm{CV}(\mathbf{x}) = 0\}$. Suppose two decision vectors $\mathbf{x}_u, \mathbf{x}_v \in \mathbb{F}$, if $\forall i \in \{1, 2, \ldots, m\}$, $f_i(\mathbf{x}_u) \le f_i(\mathbf{x}_v)$ and $\exists i \in \{1, 2, \ldots, m\}$, $f_i(\mathbf{x}_u) < f_i(\mathbf{x}_v)$, then $\mathbf{x}_u$ is said to Pareto dominate $\mathbf{x}_v$, denoted as $\mathbf{x}_u \prec \mathbf{x}_v$. A solution $\mathbf{x}_u \in \mathbb{F}$ is called a Pareto-optimal solution if and only if $\neg\exists \mathbf{x}_v \in \mathbb{F}$, $\mathbf{x}_v \prec \mathbf{x}_u$. The set of all Pareto-optimal solutions is called the Pareto set ($\mathcal{PS}$), and the image of $\mathcal{PS}$ in the objective space is called the Pareto front (PF): $\mathcal{PF} = \{\mathbf{F}(\mathbf{x}_u) | \mathbf{x}_u \in \mathcal{PS}\}$. The ultimate goal of the

[1] At least two of them are conflicting with each other.

Manuscript received July 22, 2020; revised November 20, 2020; accepted January 19, 2021. This work was supported in part by the Science and Technology Commission of Shanghai Municipality under Grant 19511120600; in part by the Program for Guangdong Introducing Innovative and Enterpreneurial Teams under Grant 2017ZT07X386; in part by the Shenzhen Peacock Plan under Grant KQTD2016112514355531; and in part by the National Leading Youth Talent Support Program of China. This article was recommended by Associate Editor H. Ishibuchi. *(Corresponding author: Ke Tang.)*

Zhi-Zhong Liu is with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: liuzz3@sustech.edu.cn).

Bing-Chuan Wang is with the School of Automation, Central South University, Changsha 410083, China (e-mail: bingcwang@csu.edu.cn).

Ke Tang is with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China, and also with the Research Institute of Trustworthy Autonomous Systems, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: tangk3@sustech.edu.cn).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TCYB.2021.3056176.

Digital Object Identifier 10.1109/TCYB.2021.3056176

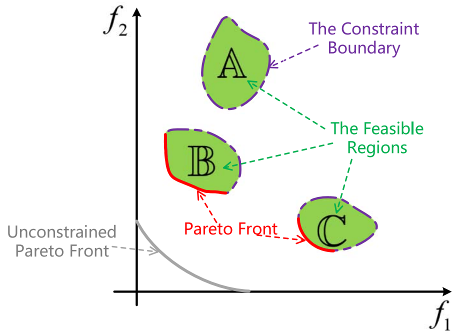

Fig. 1. Illustration of the effects of the constraints.

constrained multiobjective optimization is to obtain a set of well-distributed Pareto-optimal solutions.

However, it is not an easy task to achieve that goal, especially because of the existence of constraints. Specifically, the constraints can divide the entire search space into a series of feasible and infeasible regions; thus, the Pareto-optimal solutions may be scattered in several feasible regions. Under this condition, it would be challenging to discover these feasible regions in a single run since these feasible regions might be very narrow and/or disparately distributed in the search space. Besides, the constraints can make the PF totally different from the unconstrained PF, in terms of both the shape and the location. In this case, the Pareto-optimal solutions are very likely located on constraint boundaries [4]–[7], which are not so easy to attain. To make a clear explanation about the effects of the constraints, an example is presented in Fig. 1. As shown in Fig. 1, feasible regions $\mathbb{A}$, $\mathbb{B}$, and $\mathbb{C}$ are enclosed by the constraint boundaries and they are scattered in the search space. The Pareto-optimal solutions are simultaneously distributed in some constraint boundaries of $\mathbb{B}$ and $\mathbb{C}$. Obviously, it would face great challenges for a CMOEA to approach all these constraint boundaries and to obtain a well-distributed approximation of the entire PF during only one run [8]–[12].

Indeed, to address a CMOP, a straightforward way is first to move the population into the feasible region, then to drive the population toward the PF within the feasible region. However, this method may bring about the following two issues.

1) It can lead to the population being stuck at some locally feasible regions (e.g., feasible region $\mathbb{A}$ in Fig. 1) or local optimal feasible regions (e.g., only feasible region $\mathbb{B}$ or $\mathbb{C}$ in Fig. 1).
2) In addition, the driving force may be limited since the population only evolves from the feasible side of the search space. Thus, this method is unable to push the population toward the PF promptly.

To alleviate the first issue, many researchers have claimed that the information of infeasible solutions should be utilized since this kind of information can help to maintain the diversity of the search and find as many feasible regions as possible. Along this line, some researchers try to take CV as an additional objective function [7], [13] and transform the original m-objective CMOP into an unconstrained $(m+1)$-objective multiobjective optimization problem (MOP). Note that the transformed MOP can be conveniently addressed by the current MOEAs; thus, the population can search with good diversity in this $(m+1)$-dimensional objective space. Another way is to ignore the constraints and only take the objective functions into account, with the aim of providing as many diversified solutions as possible [14]. To remedy the second issue, according to the knowledge from constrained single-objective optimization, the potential infeasible solutions near the constraint boundaries should be retained, since they can provide an advantage to quickly search for the optimal solution by surrounding the boundary of the feasible region from both the feasible and infeasible sides [15]. Unfortunately, few current CMOEAs can obtain such infeasible solutions and make use of them appropriately. In summary, to design an efficient CMOEA: 1) aside from the search in the feasible regions, the search in the infeasible regions should also be carefully considered, which should not only maintain good diversity but also surround the constraint boundaries and 2) the cooperation between these two different kinds of searches should be taken into account, aiming at improving search efficiency.

Following these ideas, in this article, we propose a novel CMOEA with bidirectional coevolution, denoted as BiCo, for properly addressing CMOPs. BiCo can conveniently coevolve the solutions toward the PF from both the feasible and infeasible sides of the search space. Specifically, BiCo maintains two populations—the main population and the archive population—which are the main driving forces toward the PF within the feasible and infeasible regions, respectively. To update the main population, we first employ the constraint-domination principle (CDP) to drive the population into the feasible region, then utilize an NSGA-II variant to guide the population toward the PF from the feasible side of the search space. To update the archive population, we first tend to find these potential infeasible solutions, in other words, the nondominated infeasible solutions by regarding CV as an additional objective function. Afterward, a brand-new angle-based selection scheme is devised to update these infeasible solutions, which can move the population toward the PF in good diversity within the infeasible region. In addition, to coordinate the searches in the main and archive populations, a restricted mating selection mechanism is developed to select appropriate mating parents.

The main contributions of this article are listed as follows.

1) This article makes an attempt to solve CMOPs via bidirectional coevolution, and accordingly, a novel CMOEA, called BiCo, has been proposed. By coevolving two populations (i.e., the main population and the archive population), BiCo can conveniently and effectively drive the solutions toward the PF from both the feasible and infeasible sides of the search space, which is of essential importance in constrained multiobjective optimization.
2) A novel angle-based selection scheme is designed to update the archive population. This scheme can not only maintain the diversity of the search, facilitating the discovering of more feasible regions; but can also retain the infeasible solutions close to the PF, speeding up the search for the Pareto-optimal solutions.
3) To coordinate the interactions between the main and archive populations and make use of the complementary

information of them, a brand-new restricted mating selection mechanism is developed in this article.

4) Systemic experiments have been conducted on three sets of CMOP benchmark functions (e.g., MW [5], CTP [4], and LIR-CMOP [16]) and six real-world CMOPs to validate the effectiveness of BiCo. The empirical results suggest that BiCo can obtain quite competitive performance in comparison to eight peer CMOEAs in terms of both IGD [17] and HV [18]. Furthermore, the benefits of some important algorithmic components in BiCo have been verified.

The remainder of this article is organized as follows. Section II provides a brief literature review of the current CMOEAs. The details of BiCo are given in Section III. The experimental setup is introduced in Section IV and the experiments and discussions are carried out in Section V. Finally, Section VI concludes this article.

II. Literature Review

The past two decades have witnessed a significant progress in the development of EAs for CMOPs. Up to the present, various CMOEAs have been proposed, and in this article, they are roughly grouped into two categories: 1) the feasibility-driven CMOEA and 2) the infeasibility-assisted CMOEA.

A. Feasibility-Driven CMOEA

As the name suggests, this category is mainly driven by feasibility information, which always assigns a higher superiority to feasible solutions than to infeasible ones. Under this consideration, Coello Coello and Christiansen [19] proposed a simple method for constrained multiobjective optimization. In this method, only the feasible solutions are retained while the infeasible solutions are neglected. However, this method fails to distinguish solutions when all solutions are infeasible [14]. Thus, this method would be invalid for CMOPs with a narrow feasible region. To increase the selection pressure, Deb *et al.* [20] proposed the famous NSGA-II-CDP, which incorporates the CDP into NSGA-II for environmental selection. The implementation of CDP is quite simple. Given two solutions $\mathbf{x_u}$ and $\mathbf{x_v}$, $\mathbf{x_u}$ is said to constraint-dominate $\mathbf{x_v}$ if:

1) both $\mathbf{x_u}$ and $\mathbf{x_v}$ are infeasible, and $\mathrm{CV}(\mathbf{x_u}) < \mathrm{CV}(\mathbf{x_v})$;
2) $\mathbf{x_u}$ is feasible yet $\mathbf{x_v}$ is infeasible;
3) both $\mathbf{x_u}$ and $\mathbf{x_v}$ are feasible, and $\mathbf{x_u} \prec \mathbf{x_v}$.

Clearly, CDP has the capability to motivate the population to approach or enter the feasible region very quickly. Due to its simplicity and efficiency, at present, CDP has already been embedded into various MOEAs (i.e., AnD [21], NSGA-III [22], and MOEA/D [23]) and search engines (i.e., particle swarm optimization [24] and differential evolution [25]) for constrained multiobjective optimization.

Similar to Deb *et al.* [20], Jiménez *et al.* [26] proposed a novel CMOEA, called ENORA. In ENORA, a min-max formulation-based evaluation function is employed to evolve the infeasible solutions toward the feasible ones. Afterward, a Pareto-based MOEA is applied to drive the feasible solutions toward the PF. Recently, Miyakawa *et al.* [27] proposed a two-stage nondominated sorting for addressing CMOPs. The main novelty of this work lies in its two-stage nondominated sorting: 1) first, the degree of constraint violation on each constraint is considered as an objective function and the nondominated sorting is implemented to sort all solutions according to their constraint violation values and, thereafter, 2) the solutions in each obtained front after the first stage are reclassified by nondominated sorting according to their objective function values. Miyakawa *et al.* [27] claimed that the usage of this two-stage nondominated sorting leads to finding feasible solutions having better objective functions. Very recently, Liu and Wang [6] proposed a two-phase framework, called ToP, to cope with CMOPs with complex constraints. The uniqueness of ToP lies in its first phase, in which a CMOP is transformed into a constrained single-objective optimization problem by making use of the weighted sum approach. To address this transformed problem, the feasibility rule is employed to handle the constraints [28]. Note that the feasibility rule is a well-known feasibility-driven constraint-handling technique (CHT) in the constrained optimization community [17], [29].

B. Infeasibility-Assisted CMOEA

This category is assisted by infeasibility information, in which some infeasible solutions are also preferred rather than only the feasible ones since the potential information of these infeasible solutions can also benefit the evolutionary search. One representative in this category is IDEA [7], in which a small percentage of infeasible solutions are explicitly maintained during the entire evolution. To update these infeasible solutions, IDEA first transforms the original m-objective CMOP into an unconstrained $(m + 1)$-objective MOP by taking the overall constraint violations as an additional objective function. Afterward, nondominated sorting and crowding distance sorting are conducted to rank these infeasible solutions of $m + 1$ objective functions regarding the transformed problem. Analogously, Peng *et al.* [13] proposed a novel evolutionary algorithm with directed weights for constrained multiobjective optimization. This method also regards the overall constraint violations as an additional objective function, but uses two types of weights—feasible weights and infeasible ones—to guide the search toward the promising regions. Note that these infeasible weights are dynamically changed along with the evolution to prefer infeasible solutions with better objective values and smaller constraint violations. Oyama *et al.* [30] proposed a new CMOEA based on the Pareto-optimality and niching concepts, in which the current population is separated into the feasible and infeasible subpopulations according to the solutions' degree of constraint violations. Similarly, in the CMOEA proposed by Sorkhabi *et al.* [31], the population is also divided into a feasible subpopulation and an infeasible one, which are evolved in a parallel manner.

By modifying the objective functions with the constraints, some potential infeasible solutions can also be reserved. Woldesenbet *et al.* [32] proposed an adaptive penalty function, which is in coupled with a distance measure to search for the Pareto-optimal solutions in the feasible regions and to exploit the important information provided by the infeasible solutions with better objective function values and lower constraint violation values. Jan and Zhang [33] introduced a penalty function that penalizes infeasible solutions based on an adaptive threshold value into the framework of MOEA/D for addressing CMOPs. Jiao *et al.* [34] proposed a novel CMOEA by combining a modified objective function method with a feasible-guiding strategy, which can lead to dominance checking and repair of the infeasible solutions, respectively. By blending a solution's rank in the objective space with its rank in the constrained space, Young [35] proposed a novel CMOEA that can cross the infeasible regions and find the true PF. Similarly, Ning *et al.* [36] incorporated a constrained nondominated rank into an improved version of MOEA/D-M2M [37] for addressing CMOPs. Note that the assignment of a solution's nondominated rank is based on its constraint violations and its Pareto rank at the same time. With the utilization of the ε-constrained method, the constraint boundary can be relaxed to a certain degree, then some infeasible solutions can also survive. The ε-constrained method was extended into the framework of MOEA/D [38] for dealing with CMOPs [39], [40]. As for the ε value, it is adaptively adjusted to achieve superior performance. Besides, the ε-constrained method was also integrated in ECHM [41], in which an ensemble of CHTs are used rather than a single one to tackle CMOPs. Note that for each CHT, in ECHM, a different population is associated with it.

Recently, a two-archive evolutionary algorithm (C-TAEA) was proposed for constrained multiobjective optimization by Li *et al.* [14]. C-TAEA simultaneously maintains two collaborative archives: 1) the convergence-oriented archive (CA) and 2) the diversity-oriented archive (DA), aiming to drive the population toward the PF and maintain the diversity of the population, respectively. Also, Fan *et al.* embedded a CHT, called angle-based-constrained dominance principle (ACDP), into MOEA/D for tackling CMOPs [42]. In ACDP, the angle information among all solutions and the proportion of the feasible solutions are used to adjust the dominance relationship. Very recently, Fan *et al.* [43] incorporated a push-and-pull search (PPS) framework into a constrained MOEA/D for coping with CMOPs. PPS consists of two stages: 1) the push stage and 2) the pull stage. In the push stage, only the objective functions are considered, with the aim of crossing the infeasible regions in front of the unconstrained PF. In the pull stage, both the objective functions and the constraints are considered, and an improved ϵ CHT is implemented to pull the solutions obtained in the push stage toward the true PF. It is worth noting that Fan *et al.* also applied PPS to a multiobjective to multiobjective (M2M) decomposition approach and then proposed a very efficient CMOEA, that is, PPS-M2M in [44].

Our work in this article falls into the second category. Moreover, the feasible solutions and the infeasible ones are coevolved to search for the Pareto-optimal solutions.

III. Proposed Algorithm

A. BiCo

This article tries to solve CMOPs via bidirectional coevolution, and accordingly, a novel CMOEA, called BiCo, is proposed. The motivation behind BiCo comes from the following three aspects.

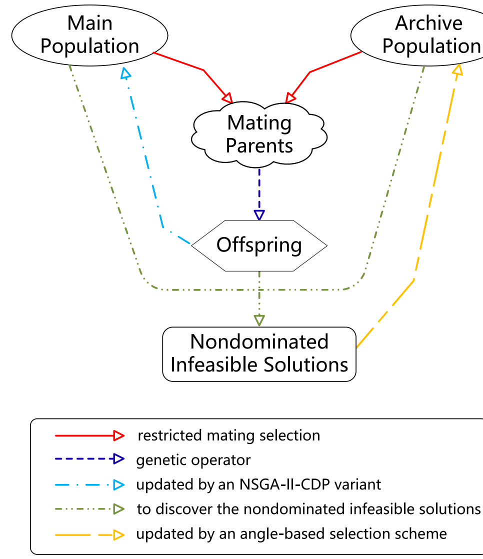

Fig. 2. System architecture of BiCo.

1) Although it is quite necessary to evolve the solutions toward the PF from two complementary directions, that is, from the feasible and infeasible sides of the search space, few current CMOEAs can effectively accomplish this task. For feasibility-driven CMOEAs, they focus mainly on feasible solutions, thus they can only move the solutions toward the PF from the feasible side of the search space. As for most current infeasibility-assisted CMOEAs, they keep both feasible and infeasible solutions during the evolution. Note, however, that these infeasible solutions are mainly employed to maintain the diversity of the search (see Section I) and they lose the capability to offer an effective driving force toward the PF from the infeasible side of the search space.
2) It is indeed the fact that the infeasible solutions are expected to be close to the constraint boundaries, where the Pareto-optimal solutions locate while maintaining good diversity. Nevertheless, to the best of our knowledge, almost no current CMOEAs can obtain such infeasible solutions effectively.
3) Besides the searches from the feasible and infeasible sides of the search space, the interaction between them is also very important. Unfortunately, up to now, there is still not much research focusing on this issue.

BiCo aims to address the previous three issues. As presented in Fig. 2,[2] BiCo maintains two populations: 1) the main population and 2) the archive population, which are updated according to different strategies and they are employed to provide evolutionary forces toward the PF from the feasible and infeasible sides of the search space, respectively. In terms of how to obtain such infeasible solutions around the PF and with good diversity as well, in BiCo, a novel angle-based selection scheme is designed to update the archive population. For

[2]The painting of this figure borrows the ideas of Fig. 2 in [45].

Algorithm 1 Framework of BiCo

Input: a CMOP and the population size N

Output: $\mathcal{P}_{t+1}$

Initialize $t = 0$, main population $\mathcal{P}_0 = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, and archive population $\mathcal{A}_0 = \emptyset$;
while *the stopping criterion is not met* **do**
  Implement the restricted mating selection to select the mating parents $\overline{\mathcal{P}}$ from $\mathcal{P}_t$ and $\mathcal{A}_t$;
  Implement the genetic operations to produce the offspring $\mathcal{Q}_t$ from $\overline{\mathcal{P}}$;
  Update $\mathcal{P}_{t+1}$ from $\mathcal{P}_t$ and $\mathcal{Q}_t$ by making use of an NSGA-II-CDP variant;
  Implement the nondominated sorting procedure to discover the nondominated infeasible solutions (i.e., $\mathcal{V}_t$) in the union of $\mathcal{P}_t$, $\mathcal{A}_t$ and $\mathcal{Q}_t$.
  Update $\mathcal{A}_{t+1}$ from $\mathcal{V}_t$ by using an angle-based selection scheme;
  $t = t + 1$;
end while

coordinating the interaction between the main and archive populations, a restricted mating selection is developed to generate promising offspring.

Algorithm 1 demonstrates the main framework of BiCo. In the initialization process, the main population $\mathcal{P}_0 = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ is randomly sampled from the search space, and the archive population $\mathcal{A}_0$ is initialized to be empty. During the search process, a restricted mating selection and some genetic operations are conducted in sequence to produce the offspring $\mathcal{Q}_t$. Afterward, an NSGA-II-CDP variant is executed to update the main population $\mathcal{P}_{t+1}$. Finally, a nondominated sorting procedure and an angle-based selection scheme are executed to update the archive population $\mathcal{A}_{t+1}$. In brief, BiCo contains three key components: 1) the updating of $\mathcal{P}_t$; 2) the updating of $\mathcal{A}_t$; and 3) the generation of $\mathcal{Q}_t$.

B. Updating of the Main Population

The main population is the main driving force toward the PF from the feasible side of the search space. To update the main population, in BiCo, an NSGA-II-CDP variant is implemented by borrowing the ideas from CDP and NSGA-II. CDP, as introduced in Section II-A, prefers feasible solutions to infeasible ones, while for the two infeasible solutions, it prefers the one with smaller constraint violations. As a result, CDP can push the population toward the feasible region promptly. As for NSGA-II, it is one of the most famous MOEAs in the past two decades. In NSGA-II, a nondominated sort is implemented to divide the entire population into several nondomination levels (i.e., $\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_k$). For the last desired level $\mathcal{F}_l$,[3] the crowding distance sort is conducted to delete unpromising solutions [20]. In general, with the combination of CDP and NSGA-II, NSGA-II-CDP is expected to quickly move the population toward the feasible region, and then guide the population toward the PF within the feasible region. The detailed implementation procedures are presented in Algorithm 2.

[3]Note that $\sum_{i=1}^{l-1} |\mathcal{F}_i| < N < \sum_{i=1}^{l} |\mathcal{F}_i|$, $\mathcal{F}_i$ denotes the ith nondomination level and N is the population size.

Algorithm 2 Updating of the Main Population

Input: $\mathcal{P}_t$ and $\mathcal{Q}_t$

Output: $\mathcal{P}_{t+1}$

1: $\mathcal{U}_t = \mathcal{P}_t \cup \mathcal{Q}_t$
2: Divide $\mathcal{U}_t$ into the feasible solution set $\mathcal{S}_1 = \{\mathbf{u}_i \in \mathcal{U}_t | \mathrm{CV}(\mathbf{u}_i) = 0\}$ and the infeasible solution set $\mathcal{S}_2 = \{\mathbf{u}_i \in \mathcal{U}_t | \mathrm{CV}(\mathbf{u}_i) > 0\}$;
3: **if** $\|\mathcal{S}_1\| \geq N$ **then**
4:   Partition $\mathcal{S}_1$ into several PFs denoted as $\mathcal{F}_1, \ldots, \mathcal{F}_k$ by applying the nondominated sorting;
5:   $\mathcal{P}_{t+1} = \emptyset$ and $i = 1$;
6:   **while** $\|\mathcal{P}_{t+1}\| + \|\mathcal{F}_i\| \leq N$ **do**
7:     $\mathcal{P}_{t+1} \leftarrow \mathcal{P}_{t+1} \cup \mathcal{F}_i$;
8:     $i \leftarrow i + 1$;
9:   **end while**
10:   **while** $\|\mathcal{P}_{t+1}\| + \|\mathcal{F}_i\| > N$ **do**
11:     Calculate the crowding distances in $\mathcal{F}_i$ and delete the solution with the smallest crowding distance from $\mathcal{F}_i$;
12:   **end while**
13:   $\mathcal{P}_{t+1} \leftarrow \mathcal{P}_{t+1} \cup \mathcal{F}_i$;
14: **else**
15:   $\mathcal{P}_{t+1} \leftarrow \mathcal{S}_1$;
16:   Sort solutions in $\mathcal{S}_2$ in ascending order according to CV and place the top $(N - \|\mathcal{S}_1\|)$ best solutions into $\mathcal{P}_{t+1}$;
17: **end if**

From Algorithm 2, it is observed that the NSGA-II-CDP variant used here is slightly different from the original one.

1) First, the crowding distance used here is defined as the minimum Euclidean distance between one solution and the other solutions in F_l regarding the objective space (line 11). Suppose several solutions share the same crowding distance, the second minimum Euclidean distance will be considered, and so on.
2) Furthermore, the solutions in the last desired level $\mathcal{F}_l$ herein are deleted one by one. This is quite different from the original NSGA-II-CDP but similar to SPEA2 [46] and HypE [47]. That means each time, only the solution with the smallest crowding distance (lines 10-12) will be discarded and then the remaining solutions' crowding distances will be re-evaluated. This way, as claimed in [47], can bring better results.

C. Updating of the Archive Population

Unlike the main population, the archive population is the primary driving force toward the PF from the infeasible side of the search space. Algorithm 3 describes how BiCo updates the archive population. In general, it involves two essential elements.

1) Discovery of the Nondominated Infeasible Solutions: In the community of constrained evolutionary optimization, nondominated infeasible solutions are always regarded as

Algorithm 3 Updating of the Archive Population

Input: $\mathcal{P}_t$, $\mathcal{A}_t$, and $\mathcal{Q}_t$

Output: $\mathcal{A}_{t+1}$

$\mathcal{U}_t = \mathcal{P}_t \cup \mathcal{A}_t \cup \mathcal{Q}_t$;
Find the nondominated solutions in $\mathcal{U}_t$ by regarding CV as an additional objective function, and select out the infeasible nondominated solutions (i.e., $\mathcal{V}_t$);
while $\|\mathcal{V}_t\| > N$ **do**
  Find the two solutions (denoted as $\mathbf{u}_i$ and $\mathbf{u}_j$) with the smallest vector angle (i.e., $\theta_{\mathbf{u}_i,\mathbf{u}_j}$) in $\mathcal{V}_t$;
  if $\text{CV}(\mathbf{u}_j) < \text{CV}(\mathbf{u}_i)$ **then**
    $\mathcal{V}_t \leftarrow \mathcal{V}_t \backslash \mathbf{u}_i$; // delete $\mathbf{u}_i$
  else
    $\mathcal{V}_t \leftarrow \mathcal{V}_t \backslash \mathbf{u}_j$; // delete $\mathbf{u}_j$
  end if
end while
$\mathcal{A}_{t+1} \leftarrow \mathcal{V}_t$;

promising ones [48]. Herein, the following steps are conducted to obtain these nondominated infeasible solutions.

1) Take CV in (3) as an additional objective function and transform the original CMOP in (1) into an unconstrained (m+1)-objective MOP

$$\min \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}), \text{CV}(\mathbf{x}))^T. \quad (4)$$

2) Implement a nondominated sorting procedure to discover the nondominated solutions in the union of $\mathcal{P}_t$, $\mathcal{A}_t$, and $\mathcal{Q}_t$, in terms of (4).
3) Pick out the infeasible ones from the obtained nondominated solutions.

The finally obtained solutions (i.e., $\mathcal{V}_t$) are the desired ones.

Remark 1: The main population $\mathcal{P}_t$ is employed in the previous procedures, and this is quite necessary since the feasible solutions in $\mathcal{P}_t$ can provide very important information to identify the quality of the infeasible solutions. For ease of understanding, an example is presented in Fig. 3. Suppose **A** and **B** are feasible solutions in $\mathcal{P}_t$, while **C**, **D**, **E**, **F**, **G**, and **H** are infeasible ones in the union of $\mathcal{A}_t$ and $\mathcal{Q}_t$. Due to the existence of **A** and **B**, solutions **C**, **D**, and **E** would become the dominated infeasible solutions, which are obviously not desired and should be discarded. However, without the help of **A** and **B**, infeasible solutions **C**, **D**, and **E** might be retained in the archive population since they are quite close to the constraint boundaries (but are far away from the PF) and **F**, **G**, and **H** cannot dominate them as well.

2) Angle-Based Selection: If the size of $\mathcal{V}_t$ is larger than the archive size N, then some redundant solutions in $\mathcal{V}_t$ need to be discarded. To achieve that goal, in our study, an angle-based selection scheme is developed by making use of the merits of the vector angle [21], [42].

Herein, the vector angle refers to the angle between two solutions in the normalized objective space (without considering the constraints). To calculate the vector angle, it has to find the ideal point $Z^{\min} = (z_1^{\min}, z_2^{\min}, \ldots, z_m^{\min})$ and estimate the nadir point $Z^{\max} = (z_1^{\max}, z_2^{\max}, \ldots, z_m^{\max})$, where $z_i^{\min}$ and $z_i^{\max}$ denote the minimum and maximum values of the ith objective for all solutions in $\mathcal{V}_t$, respectively. Subsequently,

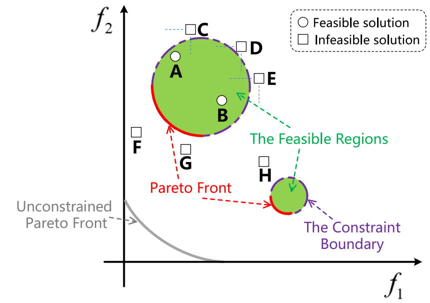

Fig. 3. Illustration of the feasible solutions' effects in the discovery of the nondominated infeasible solutions. Due to the existence of the feasible solutions **A** and **B**, the infeasible solutions **C**, **D**, and **E** become dominated infeasible solutions. While for the infeasible solutions **F**, **G**, and **H**, they are nondominated infeasible solutions. Obviously, the nondominated infeasible solutions are more promising than the dominated infeasible solutions.

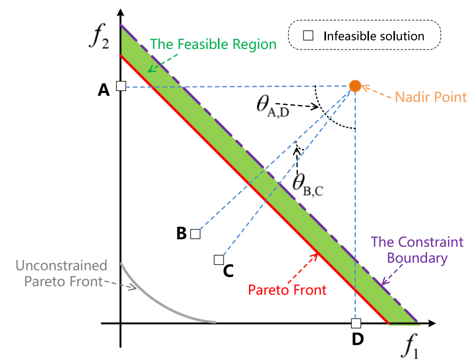

Fig. 4. Illustration of the vector angle in CMOP with two objective functions.

the jth solution's objective vector $\mathbf{F}(\mathbf{v}_j)$ can be normalized as $\mathbf{F}'(\mathbf{v}_j) = (f_1'(\mathbf{v}_j), f_2'(\mathbf{v}_j), \ldots, f_m'(\mathbf{v}_j))$ according to

$$f_i'(\mathbf{v}_j) = \frac{z_i^{\max} - f_i(\mathbf{v}_j)}{z_i^{\max} - z_i^{\min}}, \quad i = 1, 2, \ldots, m. \quad (5)$$

Thereafter, the vector angle between two solutions $\mathbf{v}_j$ and $\mathbf{v}_k$ (referred as $\theta_{\mathbf{v}_j,\mathbf{v}_k}$) can be computed as

$$\theta_{\mathbf{v}_j,\mathbf{v}_k} = \arccos \left| \frac{\mathbf{F}'(\mathbf{v}_j) \bullet \mathbf{F}'(\mathbf{v}_k)}{\|\mathbf{F}'(\mathbf{v}_j)\| \cdot \|\mathbf{F}'(\mathbf{v}_k)\|} \right| \quad (6)$$

where $\mathbf{F}'(\mathbf{v}_j) \bullet \mathbf{F}'(\mathbf{v}_k)$ returns the inner product between $\mathbf{F}'(\mathbf{v}_j)$ and $\mathbf{F}'(\mathbf{v}_k)$, and $\|\cdot\|$ calculates the norm of the vector. In general, the vector angle has a promising property, that is it can reflect the similarities of the search directions between two solutions to some extent. For two solutions searching from very different directions, the vector angle between them would be large; otherwise, the vector angle between them would be relatively small. An example is presented in Fig. 4. Suppose **A**, **B**, **C**, and **D** are nondominated infeasible solutions in $\mathcal{V}_t$. It can be seen that **A** and **D** search from different directions; thus, $\theta_{\mathbf{A},\mathbf{D}}$ is quite large, while **B** and **C** share similar search directions. Hence, $\theta_{\mathbf{B},\mathbf{C}}$ is relatively small.

In our proposed angle-based selection scheme, a "diversity first and feasibility second" mechanism is developed to delete the poor solutions in $\mathcal{V}_t$ one by one. Its implementation is quite simple. Specifically, it first identifies two solutions in $\mathcal{V}_t$ with the minimum vector angle. To maintain the search for diversity,

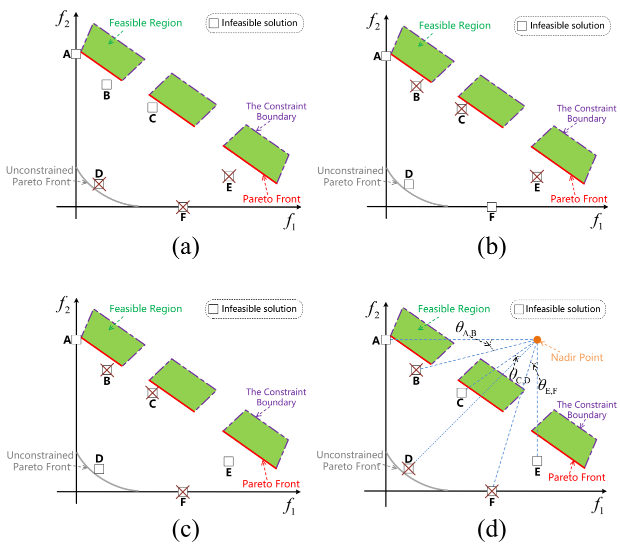

Fig. 5. Illustration of the working principles of four different kinds of CHTs. There are six solutions in the population, that is, **A**, **B**, **C**, **D**, **E**, and **F**. The task is to select three solutions from the population into the next generation. (a) Feasibility-driven CHT. (b) Constraints ignoring CHT. (c) Multiobjective-based CHT. (d) Angle-based selection.

TABLE I
INFORMATION OF THE INFEASIBLE SOLUTIONS

Solution	f_1	f_2	CV
A	0	1	0.1
B	0.2	0.8	0.3
C	0.5	0.65	0.2
D	0.15	0.15	1
E	1	0.2	0.4
F	0.7	0	0.7

intuitively, it is necessary to delete one of them since they search in the most similar directions. To make the deletion wiser, the feasibility information of these two solutions is considered, and the one with the larger value of CV is discarded. These two steps will continue until the size of $\mathcal{V}_t$ is no more than N. In general, by repeatedly deleting one of two infeasible solutions with the most similar search directions in the remaining population, the diversity of the finally obtained population will be well maintained. Note that in the deletion process, the feasibility information of the solutions is considered, which is beneficial to keep these infeasible solutions near the constraint boundaries. As discussed in Section I, the Pareto-optimal solutions are very likely located on constraint boundaries. As a result, the finally obtained infeasible solutions can not only maintain good diversity but are also close to the PF.

An example is presented in Fig. 5 to further explain the difference between our angle-based selection and the other three representative CHTs: 1) feasibility-driven CHT [20], [21]; 2) constraints ignoring CHT [14], [43]; and 3) multiobjective-based CHT [7], [13]. In Fig. 5, **A**, **B**, **C**, **D**, **E**, and **F** are six nondominated infeasible solutions in terms of (4), and their detailed information is given in Table I. Our task is to select three promising infeasible solutions into the next generation. Fig. 5 depicts what happens to these four compared CHTs.

1) In feasibility-driven CHT, the infeasible solutions with smaller constraint violations are preferred. From Table I, it is observed that **A**, **B**, and **C** have the smaller CV; thus, they will be selected into the next generation [see Fig. 5(a)].
2) With respect to the constraints ignoring CHT, the constraints are neglected and only the objective functions are considered. Under this condition, the nondominated solutions in terms of the objective functions are preferred. From Fig. 5(b), it is clear that **A**, **D**, and **F** are such kind of solutions and they will survive into the next generation.
3) For multiobjective-based CHT, CV is regarded as an additional objective function, and the original CMOP is transformed into an unconstrained $(m+1)$-objective MOP. To solve this transformed problem, NSGA-II is supposed to be applied. Since **A**, **B**, **C**, **D**, **E**, and **F** are nondominated infeasible solutions, they cannot be distinguished by the nondominated sorting. In this situation, the crowding distance sort will work. Suppose the crowding distance of one solution is defined as the lth nearest Euclidian distance between this solution to the other solutions, and the setting of l follows the ideas in [18]. Then, l would be 3. Thereafter, we can calculate the crowding distances of **A**, **B**, **C**, **D**, **E**, and **F**, which are 1.2470, 0.9566, 0.7018, 1.0062, 1.0050, and 0.8441, respectively. Obviously, **A**, **D**, and **E** have relatively larger crowding distances. Thus, they will be retained as shown in Fig. 5(c).
4) To implement the angle-based selection, first, we have to compute the vector angles between any two solutions in the entire population. Afterward, we need to find these two solutions with the minimum vector angle and then differentiate them based on their CV values. Clearly, **C** and **D** share the minimum vector angle in the population. Then, **D** is removed since it has a larger CV than **C**. After **D** has been eliminated, **A** and **B** share the minimum vector angle in the remaining population. Note that CV(**A**) > CV(**B**), then **B** is discarded. Similarly, after **B** has been deleted, **E** and **F** have the minimum angle vector. Then, **F** will be deleted since CV(**E**) > CV(**F**). In summary, **B**, **D**, and **F** will be eliminated, and **A**, **C**, and **E** will be survive [see Fig. 5(d)].

From this discussion, we can observe that our proposed method can obtain more suitable results compared with its three competitors. For feasibility-driven CHT, it keeps **A**, **B**, and **C**, which are quite close to the PF but are not well distributed. In terms of constraints ignoring CHT and multiobjective-based CHT, their finally obtained solutions can indeed keep good diversity; nevertheless, some of them may be far away from the PF. Note, however, that only in the proposed angle-based selection, can the obtained solutions (i.e., **A**, **C**, and **E**) be close to the constraint boundaries (where the PF lies on) and in diverse search directions as well. Obviously, this kind of infeasible solution is more desired.

D. Offspring Generation

Apart from the evolution of the main and archive populations, the interaction and collaboration between them are

Algorithm 4 Restricted Mating Selection

Input: $\mathcal{P}_t$ and $\mathcal{A}_t$
Output: Mating parents $\mathbf{p}_1$ and $\mathbf{p}_2$
1: **if** $\|\mathcal{A}_t\| < N$ **then**
2: Randomly select two solutions $\mathbf{p}_1$ and $\mathbf{p}_2$ from the union population of $\mathcal{P}_t$ and $\mathcal{A}_t$;
3: **else**
4: Randomly select two solutions $\mathbf{x}_1$ and $\mathbf{a}_1$ from $\mathcal{P}_t$ and $\mathcal{A}_t$, respectively;
5: **if** $\mathrm{CV}(\mathbf{x}_1) \le \mathrm{CV}(\mathbf{a}_1)$ **then**
6: $\mathbf{p}_1 \leftarrow \mathbf{x}_1$;
7: **else**
8: $\mathbf{p}_1 \leftarrow \mathbf{a}_1$;
9: **end if**
10: Randomly select two solutions $\mathbf{x}_2$ and $\mathbf{a}_2$ from $\mathcal{P}_t$ and $\mathcal{A}_t$, respectively;
11: **if** $AD(\mathbf{x}_2) \le AD(\mathbf{a}_2)$ **then**
12: $\mathbf{p}_2 \leftarrow \mathbf{x}_2$;
13: **else**
14: $\mathbf{p}_2 \leftarrow \mathbf{a}_2$;
15: **end if**
16: **end if**

also very important and should be considered carefully. To this end, in BiCo, a restricted mating selection mechanism is developed, which can leverage the complementary information of these two populations for offspring generation. Algorithm 4 presents the detailed information. In general, it considers two conditions.

1) If the size of archive population is less than N, the mating parents $\mathbf{p}_1$ and $\mathbf{p}_2$ are randomly selected from the union of $\mathcal{P}_t$ and $\mathcal{A}_t$.
2) Otherwise, $\mathbf{p}_1$ and $\mathbf{p}_2$ are selected based on CV and angle-based density value (i.e., AD), respectively. To be specific, to select $\mathbf{p}_1$, it first randomly selects two solutions $\mathbf{x}_1$ and $\mathbf{a}_1$ from $\mathcal{P}_t$ and $\mathcal{A}_t$, respectively; afterward, their CV values are compared and the one with the smaller CV is selected. To select $\mathbf{p}_2$, it also first randomly chooses $\mathbf{x}_2$ and $\mathbf{a}_2$ from $\mathcal{P}_t$ and $\mathcal{A}_t$, respectively; thereafter, their AD values are compared and the one with the larger AD is chosen.

CV can reflect the feasibility information of a solution and has been introduced in (3). As for AD, it is designed to measure the diversity of the search directions. To calculate the angle-based density value $AD(\mathbf{x})$ for the jth solution $\mathbf{x}_j$ in $\mathcal{P}_t$, the following steps are implemented.

1) Find the ideal point $\overline{Z}^{\min} = (\overline{z}_1^{\min}, \overline{z}_2^{\min}, \ldots, \overline{z}_m^{\min})$ and estimate the nadir point $\overline{Z}^{\max} = (\overline{z}_1^{\max}, \overline{z}_2^{\max}, \ldots, \overline{z}_m^{\max})$, where $\overline{z}_i^{\min}$ and $\overline{z}_i^{\max}$ denote the minimum and maximum values of the ith objective for all solutions in the joint population of $\mathcal{P}_t$ and $\mathcal{A}_t$, respectively.
2) Normalize the jth solution's objective vector $\mathbf{F}(\mathbf{x}_j)$ as $\mathbf{F}^*(\mathbf{x}_j) = (f_1^*(\mathbf{x}_j), f_2^*(\mathbf{x}_j), \ldots, f_m^*(\mathbf{x}_j))$ according to

$$f_i^*(\mathbf{x}_j) = \frac{f_i(\mathbf{x}_j) - \overline{z}_i^{\min}}{\overline{z}_i^{\max} - \overline{z}_i^{\min}}, \quad i = 1, 2, \ldots, m. \tag{7}$$

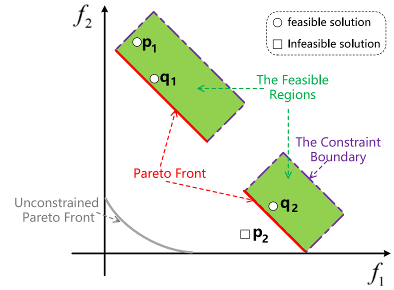

Fig. 6. Simple illustration of the offspring generation. $\mathbf{p}_1$ and $\mathbf{p}_2$ are the parents and $\mathbf{q}_1$ and $\mathbf{q}_2$ are the offsprings. In this scenario, it is observed that $\mathbf{q}_1$ and $\mathbf{q}_2$ can inherit the elite information of their parents.

3) Compute the vector angles between $\mathbf{x}_j$ and the other solutions in $\mathcal{P}_t$ according to

$$\theta'_{\mathbf{x}_j,\mathbf{x}_k} = \arccos\left|\frac{\mathbf{F}^*(\mathbf{x}_j) \bullet \mathbf{F}^*(\mathbf{x}_k)}{\|\mathbf{F}^*(\mathbf{x}_j)\| \cdot \|\mathbf{F}^*(\mathbf{x}_k)\|}\right|$$
$$\mathbf{x}_k \in \mathcal{P}_t \cap \mathbf{x}_k \neq \mathbf{x}_j. \tag{8}$$

4) Assign $AD(\mathbf{x}_j)$ as the kth minimum value in the set of $\{\theta'_{\mathbf{x}_j,\mathbf{x}_k}, \mathbf{x}_k \in \mathcal{P}_t \cap \mathbf{x}_k \neq \mathbf{x}_j\}$, where k is set to $\sqrt{N}$ following the similar idea in [46]. Note that N is the size of $\mathcal{P}_t$.

Clearly, a large AD value is desired for $\mathbf{x}_j$. Following the previous steps, similarly, we can obtain the angle-based density value $AD(\mathbf{a}_j)$ of the jth solution $\mathbf{a}_j$ in $\mathcal{A}_t$.

After the selection of the mating parents, the popular simulated binary crossover and polynomial mutation can be applied to generate the offspring. By mating one solution with good a CV value and one with good a AD value, it is expected that the generated offspring can not only be close to the PF but also with good diversity. As shown in Fig. 6, the generated offspring $\mathbf{q}_1$ and $\mathbf{q}_2$ can inherit the elite information of $\mathbf{p}_1$ and $\mathbf{p}_2$, which are chosen based on CV and AD, respectively. Other reproduction operators can also be conveniently applied with a minor modification [14].

Remark 2: Compared with C-TAEA [14] and DPP [45], BiCo adopts a unique restricted mating selection mechanism. In DPP, to select two mating parents, it selects one from the Pareto-based archive and the other one from the decomposition-based archive. As in C-TAEA, it first combined the CA and the DA together and then computed the proportion of nondominated solution of CA and DA in the union population, which are denoted as ρ_c and ρ_d, respectively. If ρ_c is larger than ρ_d, the first parent is chosen from CA; otherwise, from DA. If ρ_c is larger than a randomly generated number between 0 and 1, the second parent is chosen from CA; otherwise, from DA. Note that in the process of choosing a mating parent from CA (or DA), a binary tournament selection is conducted in C-TAEA by making use of CDP. While in BiCo, as shown in lines 4–15 of Algorithm 4, to select the first parent, it first randomly selects one solution from the main population and the other one from the achieved population. Thereafter, the CV value is employed to differentiate these two selected solutions and the one with a better CV value will be chosen. To select the second parent, similar procedures are conducted but the AD value rather than the CV value is used to distinguish the two solutions and choose the better one. It is apparent that the restricting mating selection mechanism in BiCo is completely different.

E. Discussion

In general, BiCo involves a few parameters and no complicated operators. Regarding the computational time complexity of BiCo, it is indeed acceptable and has been analyzed in Section S-I-A of the supplementary file. Also, in Section S-I-B of the supplementary file, we have investigated the difference between BiCo and several peer CMOEAs, that is, ECHM [41], PPS [43], and C-TAEA [14].

IV. Experimental Setup

A. Test Instances and Performance Metrics

In this article, all experiments were conducted on three sets of CMOP benchmark functions (e.g., MW [5], CTP [4], and LIR-CMOP [40]) and six real-world CMOPs. MW is a recently proposed test suite that covers diverse characteristics extracting from real-world CMOPs. By taking the relationship between the unconstrained and constrained PFs into account, these 14 CMOPs in MW can be classified into four types.

1) Type I—-the constrained PF is the same with the unconstrained PF, that is, MW2, MW4, and MW14.
2) Type II—-the constrained PF is a part of the unconstrained PF, that is, MW1, MW5, MW6, and MW8.
3) Type III—-the constrained PF consists of a part of the unconstrained PF and a part of the boundary of the feasible region, that is, MW3, MW7, MW10, and MW13.
4) Type IV—-the unconstrained PF is all located outside the feasible region, that is, MW9, MW11, and MW12.

CTP [4] might be the most famous CMOP test suite that contains eight test functions. LIR-CMOP [40] comprises 12 CMOPs with two objective functions and two CMOPs with three objective functions, and all of them are with large infeasible regions. Note that in this article, the number of decision variables in LIR-CMOP was set to 10. The six real-world CMOPs are: 1) CONSTR [49]; 2) disc-brake design (DBD) [50]; 3) OSY [51]; 4) SRN [52]; 5) TNK [53]; and 6) welded beam design (WBD) [50].

To assess the performance of different CMOEAs, two commonly used metrics: 1) IGD [17] and 2) HV [18], were employed in our experiments. Specifically, the IGD and HV integrated in the platform developed by Tian *et al.* [54] were adopted in our study.

B. Algorithms for Comparison

For performance comparison, two well-known feasibility-driven CMOEAs (i.e., NSGA-II-CDP [20] and A-NSGA-III [22]) and six famous infeasibility-assisted CMOEAs (i.e., IDEA [7], SP [32], C-MOEA/DD [55], MOEA/D-ACDP [42], PPS [43], and C-TAEA [14]) were under our consideration. More detailed information about them can be found in Section S-II of the supplementary file for interested readers.

C. Parameter Settings

1) *Population Size:* It was set to 100 for each algorithm on each test function [5], [20].
2) *Parameter Settings for Genetic Operators:* According to the suggestions in [20], for each algorithm, the crossover and mutation probability were set to 1.0 and $1/n$, respectively, and the distribution indexes of both SBX and the polynomial mutation were set to 20.
3) *Number of Independent Runs and Termination Condition:* All algorithms were independently run 30 times on each test function. For LIR-CMOP, the maximum number of function evaluations (FEs) was set to 300 000 [40], while for the others, the maximum number of FES was set to 60 000 [5].
4) *Parameter Settings for Algorithms:* The parameters of all the compared algorithms were kept identical with their original papers.

All experiments in this article were conducted on the platform developed by Tian *et al.* [54].

V. Results and Discussion

A. Comparison With Eight State-of-the-Art CMOEAs

First, we compare the performance of BiCo with that of the eight peer CMOEAs in Section IV-B on the MW test suite [5], CTP test suite [4], LIR-CMOP test suite [40], and six real-world CMOPs in sequence. The results are summarized in Table II and Tables S-II–S-VIII of the supplementary file.[4] In each table, Wilcoxon's rank-sum test at a 0.05 significance level was performed to test the statistical significance of the experimental results between two algorithms, and for convenience, "$+$," "$-$," and "$\approx$" denote that a peer CMOEA performs better than, worse than, and similar to BiCo, respectively. At our first glance, BiCo can achieve the best performance on most test instances with respect to both IGD and HV. As for the other algorithms, the infeasibility-assisted CMOEAs (IDEA, SP, C-MOEA/DD, MOEA/D-ACDP, PPS, and C-TAEA) can obtain overall better performance than the feasibility-driven ones (NSGA-II-CDP and A-NSGA-III). To visualize the results, we plotted the final populations resulting from the nine compared algorithms in a typical run on four representative CMOPs in Fig. 7 and Figs. S-1–S-3. Herein, a typical run denotes a run producing the median IGD value among all runs. Next, we a give detailed discussion.

1) MW Test Suite: From Tables II and S-II, it is observed that BiCo exhibited the overall best performance in terms of both IGD and HV. In general, BiCo obtained the best IGD results on all test instances (except for MW10) as presented in Table II, and achieved the best HV results on ten out of 14 test instances as demonstrated in Table S-II.

For CMOPs of Type-I and Type-II, their constrained PFs are the same with and are parts of the unconstrained PFs, respectively. That means these two types of CMOPs mainly challenge an algorithm's capability of maintaining the diversity

[4]Tables S-I–S-XXIII and Figs. S-1–S-4 are placed in the supplementary file.

TABLE II
COMPARISON RESULTS ON IGD METRIC (MEAN) FOR BICO AND THE OTHER EIGHT PEER ALGORITHMS ON MW TEST SUITE. THE BEST AND SECOND-BEST AVERAGE IGD VALUES AMONG ALL ALGORITHMS ON EACH TEST FUNCTION ARE HIGHLIGHTED IN GRAY AND LIGHT GRAY, RESPECTIVELY

Type	Problem	m	BiCo	NSGA-II-CDP	A-NSGA-III	IDEA	SP	C-MOEA/DD	MOEA/D-ACDP	PPS	C-TAEA
I	MW2	2	1.1704e-2	2.4621e-2 −	2.3701e-2 −	1.8693e-2 −	2.5740e-2 −	5.1307e-2 −	5.7135e-2 −	3.8061e-2 −	1.8181e-2 −
	MW4	3	4.1320e-2	5.4939e-2 −	4.4313e-2 −	6.2222e-2 −	5.5676e-2 −	4.1512e-2 ≈	4.1641e-2 −	5.3903e-2 −	4.6770e-2 −
	MW14	3	9.7706e-2	1.2571e-1 −	1.1736e-1 −	1.4941e-1 −	1.2617e-1 −	2.6378e-1 −	2.9153e-1 −	2.9893e-1 −	1.1158e-1 −
II	MW1	2	1.6410e-3	2.9638e-2 −	1.4242e-2 −	2.5251e-3 −	1.9925e-2 −	1.9658e-2 −	1.4615e-1 −	1.9696e-2 −	2.3995e-3 −
	MW5	2	7.9353e-4	3.4049e-1 −	3.2084e-1 −	4.8215e-3 −	1.6857e-1 −	1.6780e-2 −	2.9580e-2 −	1.3981e-1 −	1.7847e-2 −
	MW6	2	8.4277e-3	6.5919e-2 −	6.9394e-2 −	1.9357e-2 −	4.3142e-2 −	1.6534e-1 −	1.6004e-2 −	6.5690e-2 −	1.0849e-2 −
	MW8	3	4.4809e-2	6.3316e-2 −	9.5319e-2 −	6.2090e-2 −	5.9748e-2 −	5.5427e-2 −	5.2316e-2 −	6.9785e-2 −	5.3104e-2 −
III	MW3	2	5.1753e-3	3.6789e-2 −	4.1904e-2 −	7.3768e-3 −	6.2090e-3 −	8.2797e-3 −	4.8238e-1 −	7.6360e-3 −	5.4540e-3 −
	MW7	2	5.3150e-3	7.1199e-2 −	9.8922e-2 −	6.5519e-3 −	8.5984e-3 ≈	1.3324e-2 −	5.6536e-3 −	1.9538e-2 −	7.5100e-3 −
	MW10	2	2.1631e-2	1.8016e-1 −	1.8455e-1 −	2.8543e-2 −	1.8372e-1 −	9.6192e-2 −	1.4761e-1 −	1.3381e-1 −	1.2349e-2 ≈
	MW13	2	2.6424e-2	3.7073e-1 −	3.8645e-1 −	7.3242e-2 −	1.5667e-1 −	2.3118e-1 −	8.8536e-2 −	1.1262e-1 −	3.5214e-2 −
IV	MW9	2	4.7035e-3	5.4422e-3 −	7.1531e-3 −	1.7095e-2 −	6.2659e-3 −	3.2760e-2 −	7.3208e-2 −	5.1378e-2 −	2.1019e-2 −
	MW11	2	5.9808e-3	3.9296e-1 −	6.5204e-1 −	8.1488e-3 −	3.0318e-1 −	1.8223e-2 −	1.0690e-2 −	1.2728e-2 −	1.6460e-2 −
	MW12	2	4.7909e-3	2.4485e-2 −	2.0105e-1 −	6.8324e-3 −	5.5682e-2 −	2.1637e-2 −	5.6199e-3 −	2.9445e-2 −	7.9574e-3 −
+/ − / ≈			\	0/14/0	0/14/0	0/14/0	0/13/1	0/13/1	0/14/0	0/14/0	0/13/1

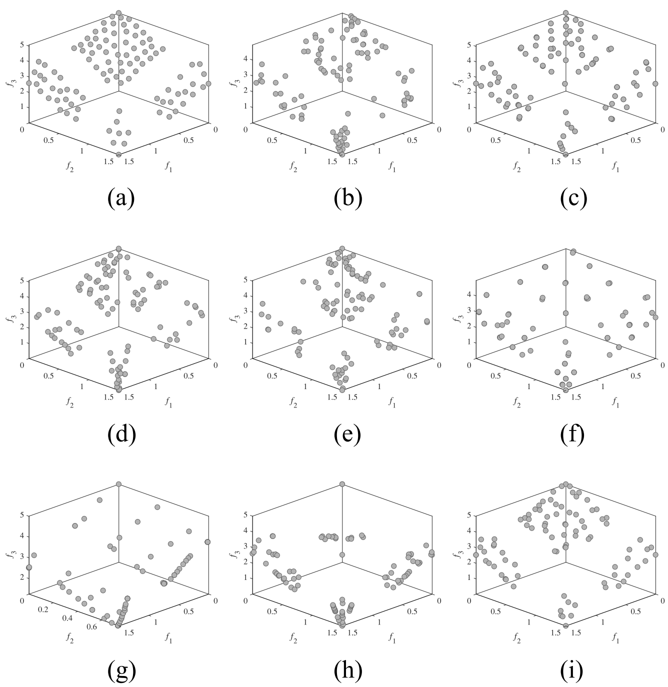

Fig. 7. Scatter plots of the population obtained by BiCo and the eight peer CMOEAs on MW14. (a) BiCo. (b) NSGA-II-CDP. (c) A-NSGA-III. (d) IDEA. (e) SP. (f) C-MOEA/DD. (g) MOEA/D-ACDP. (h) PPS. (i) C-TAEA.

of the search, which can help to avoid the loss of some parts of the PF. However, for NSGA-II-CDP, and A-NSGA-III, as observed in Tables II and S-II, none of them could obtain any best or second-best results on these two types of CMOPs. The reason might be that these two algorithms are feasibility-driven CMOEAs. As introduced in Section II, these feasibility-driven CMOEAs always give priority to the satisfaction of the constraints, while ignoring the maintenance of the population's diversity, which is of essential importance for properly solving these two types of CMOPs. In terms of IDEA and SP, the former could obtain an overall better performance than the latter. Specifically, IDEA obtained the second-best IGD values on MW5, and the second-best HV values on MW1 and MW5; while SP could not obtain any best or second-best results. The relatively poor performance of SP might be attributed to its infeasible stage, in which the objective functions are totally disregarded and only the constraints are considered to compare the solutions [32]. With respect to C-MOEA/DD and MOEA/D-ACDP, the former achieved the second-best HV value on MW4, while the latter obtained the second-best IGD value on MW8 and attained the best and second-best HV values on MW4 and MW8, respectively. The slightly better performance of MOEA/D-ACDP is not surprising, since, in MOEA/D-ACDP, an additional archive is employed to update the feasible solutions, which can help to obtain a denser representation of solutions than C-MOEA/DD [42]. Regarding PPS and C-TAEA, the latter outperformed the former in most cases. PPS failed to obtain any best or second-best results, while C-TAEA gained the second-best IGD and HV values on MW2, MW14, MW1, and MW6, and on MW2, MW14, and MW6, respectively. This phenomenon implies the merits of maintaining two populations. Interestingly, the latter four algorithms (i.e., C-MOEA/DD, MOEA/D-ACDP, PPS, and C-TAEA) failed to obtain promising results on CMOPs with disconnected PFs, such as MW14, as presented in Figs. 7(f)–(i). This is reasonable since these four algorithms are decomposition-based approaches, whose performance may be dramatically deteriorated in problems with complex PFs [56]. While for BiCo, clearly, it obtained the best IGD and HV values on most test instances in these two types of CMOPs. The highly competitive performance of BiCo can be attributed to the usage of the archive population, in which the angle-based selection scheme is employed to maintain some promising infeasible solutions with diverse search directions. In addition, the NSGA-II variant used in the main population can make the feasible solutions be more evenly distributed along the PF. From Figs. 7 and S-1, it is evident that on both MW14 and MW8, BiCo can achieve the best results among the compared nine algorithms.

As for CMOPs of Type-III and Type-IV, their constrained PFs are partially and all located in the infeasible regions, respectively. To effectively address these two types of CMOPs, not only the diversity of the search should be maintained but also the information of the infeasible solutions close to the PF should be employed. Again, the two feasibility-driven CMOEAs (i.e., NSGA-II-CDP and A-NSGA-III) all failed to obtain promising results on these two types of CMOPs. Specifically, none of them could achieve any best or second-best IGD or HV values (except for NSGA-II-CDP, which obtained the second-best IGD and HV values on MW9). The poor performance of these two feasibility-driven CMOEAs is not difficult to grasp. First, they cannot maintain population diversity as analyzed in Section II. Second, they can only search from the feasible side of the search space. Note that the PFs of these two types of CMOPs are partially or all located in the constraint boundaries and the searches from both the feasible and infeasible sides of the search space are very necessary [7]. Regarding IDEA and SP, IDEA obtained the second-best IGD and HV values on MW11 while SP could not obtain any best or second-best results. This phenomenon suggests that IDEA can obtain an overall better performance than SP. It is easy to understand since, in SP, the penalty factor is employed to help SP preserve some infeasible solutions; however, how to set an appropriate penalty factor is still a challenging issue in the current constrained optimization research field [17], [57]. Then, the infeasible solutions kept in SP maybe not as promising as those preserved by IDEA and, thus, SP cannot be better than IDEA overall in these two types of CMOPs. C-MOEA/DD did not obtain any best or second-best results on these two types of CMOPs. The reason might be that in C-MOEA/DD, an infeasible solution is discarded if it coexists with a feasible solution in the same subregion, while this infeasible solution may have good objective function values, which can benefit the evolutionary search significantly. In terms of MOEA/D-ACDP, it obtained the second-best IGD values on two instances (i.e., MW7 and MW12) and attained the best and second-best HV values on MW7 and MW12, respectively. Similar to C-MOEA/DD, PPS failed to obtain any best or second-best results. The reason might be that the main advantage of PPS lies in its push stage, which can help PPS cross the infeasible regions in front of the unconstrained PF. However, the infeasible regions in front of the unconstrained PF in MW are not difficult to pass through, then the strength of PPS cannot be exhibited. Regarding C-TAEA, it achieved the best and second-best IGD values on one instance (i.e., MW10) and two instances (i.e., MW3 and MW13), respectively, and reached the best HV values on MW3 and MW10 and the second-best HV value on MW13. As for BiCo, again, it obtained the best overall performance in these two types of CMOPs. The reason is quite straightforward: by coevolving two populations, BiCo can not only maintain the diversity of the search but can also coevolve the solutions toward the PF from both the feasible and infeasible sides of the search space.

2) CTP Test Suite: Unlike in MW, the CMOPs in the CTP test suite have large feasibility ratios. Analogously, we compared the performance of BiCo with that of eight peer algorithms on CTP problems. The results are summarized in Tables S-III and S-IV. From these two tables, again, it is observed that BiCo exhibited the best performance on most instances in terms of both IGD and HV.

For CTP1, only one-third of the PF comes from the original unconstrained PF, while the others come from the constraint boundaries. Actually, based on the classification method in MW [5], CTP1 is a CMOP of Type III. In this instance, again, it is obverted that BiCo achieved the best results in terms of both IGD and HV. This is easy to understand since BiCo can not only maintain the diversity of the search but can also coevolve the solutions toward the PF from both the feasible and infeasible sides of the search space. Similar phenomena can be seen in CTP3-CTP5 and CTP8 (see Fig. S-2). As for CTP2 and CTP7, their PFs are divided into several disconnected segments. To properly solve them, the key point is to approach the PF from diverse search directions. Interestingly, BiCo and SP obtained the best and second-best results on these two instances, respectively. Concerning CTP6, its PF is a continuous line on the boundary of a feasible region. Then, CTP6 mainly challenges an algorithm's capability of maintaining the diversity of the search in the feasible regions. BiCo and NSGA-II-CDP were two winners for this instance.

In general, the CMOEAs using the reference points or vectors (i.e., A-NSGA-III, C-MOEA/DD, MOEA/D-ACDP, PPS, and C-TAEA) failed to obtain promising results on these CTP problems, which might be attributed to the complex characteristics (i.e., disconnected and discrete) of the PFs in these CMOPs [56]. As for BiCo, IDEA, SP, and NSGA-II-CDP, the first three algorithms (infeasibility-assisted CMOEAs) obtained an overall better performance than the latter one (feasibility-driven CMOEA). It is again verified that to design an appropriate CMOEA, the infeasible solutions should be considered carefully.

3) LIR-CMOP Test Suite: Aside from MW and CTP, we also considered the CMOPs in the LIR-CMOP test suite. Note that the CMOPs in the LIR-CMOP test suite are with large infeasible regions. Again, we compared the performance of BiCo with that of eight peer algorithms on LIR-CMOP problems. The results are summarized in Tables S-V and S-VI. From these two tables, once again, BiCo achieved the overall best performance on most instances in terms of both IGD and HV.

In terms of LIR-CMOP1–LIR-CMOP4, they have large infeasible regions. On these four test instances, BiCo and MOEA/D-ACDP were two winners. It seems that the algorithms using the vector angle can obtain promising results on these four instances. Regarding LIR-CMOP5 and LIR-CMOP6, large infeasible regions lay in front of the PFs, but these PFs are the same as those of their unconstrained counterparts. To solve this kind of CMOPs, the CMOEAs using multiobjective-based CHTs and constraint ignoring CHTs are more suitable. As for BiCo, it could not gain promising results on them. For LIR-CMOP7 and LIR-CMOP8, and LIR-CMOP11 and LIR-CMOP12, their PFs are situated on the constraint boundaries. Undoubtedly, BiCo gained the overall best performance on these four test instances. As for LIR-CMOP9 and LIR-CMOP10, their PFs are parts of their

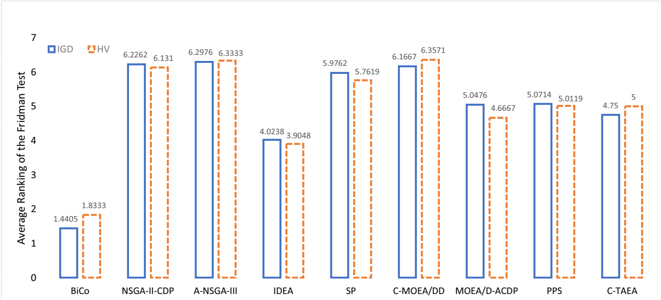

Fig. 8. Friedman test between BiCo and its eight competitors in terms of IGD and HV. The lower the ranking, the better the performance of an algorithm.

unconstrained PFs and the main difficulties lie in how to maintain the diversity of the search. The two algorithms using the vector angle, that is, BiCo and MOEA/D-ACDP, obtained the overall better performance again. The reason might be that the usage of vector angle is very helpful to maintain the diversity of the search. As for LIR-CMOP13 and LIR-CMOP14, they are CMOPs with three objective functions. For LIR-CMOP13, its PF is the same as its unconstrained PF. For this test instance, C-MOEA/DD obtained the best IGD and HV results, followed by MOEA/D-ACDP and BiCo. As for LIR-CMOP14, its PF is located on the constraint boundaries. In this case, BiCo and MOEA/D-ACDP obtained the overall best performance. Specifically, on LIR-CMOP14, BiCo obtained the best IGD value and the second-best HV value, and MOEA/D-ACDP obtained the second-best IGD value and the best HV value.

From the above discussion, it is easy to conclude that BiCo is also suitable for solving CMOPs with large infeasible regions. The images of the final population provided by these nine compared CMOEAs on LIR-CMOP1 are plotted in Fig. S-3.

4) Real-World CMOPs: One might be interested in whether BiCo can obtain superior performance on real-world CMOPs. To answer this question, we compared the performance of BiCo with that of eight peer algorithms on six real-world CMOPs. The results are summarized in Tables S-VII and S-VIII in terms of IGD and HV, respectively.

It is observed that BiCo obtained the overall best performance on these six instances. Regarding IGD, BiCo obtained the best performance on DBD, OSY, TNK, and WBD and the second-best performance on SRN. As for HV, BiCo achieved the best performance on CONSTR, DBD, and TNK, and the second-best performance on SRN. This phenomenon suggests that BiCo is also suitable for solving real-world CMOPs, rather than only for artificial ones. It is thus concluded that BiCo is a promising alternative CMOEA for dealing with a wide range of CMOPs.

5) Discussion: To analyze the overall performance of the compared nine CMOEAs on all test instances, the Friedman test was implemented in terms of both IGD and HV by taking advantage of the KEEL software [58]. Note that in the Friedman test, the lower the ranking, the better the performance of an algorithm. From Fig. 8, it is evident that BiCo achieves the lowest ranking in terms of both IGD and HV, followed by IDEA, C-TAEA, MOEA/D-ACDP, and PPS. This phenomenon suggests the necessity of maintaining some infeasible solutions during evolution. As for CMOEAs using reference points or vectors (i.e., A-NSGA-III, C-MOEA/DD, MOEA/D-ACDP, PPS, and C-TAEA), the latter three algorithms have a lower ranking than the former two. The poor performance of A-NSGA-III is easy to understand since it cannot maintain some promising infeasible solutions during the evolution. Regarding C-MOEA/DD, its unsatisfactory performance might be attributed to its relatively poor convergence performance, which is of critical importance to optimize CMOPs with two or three objective functions. To further understand BiCo, next, we will investigate its important algorithmic components.

B. Further Analysis

1) Necessity of the Archive Population: First, we want to investigate the necessity of the archive population. To this end, a variant of BiCo, called WoA-BiCo, was devised in our study. The only alteration is that in WoA-BiCo, the archive population is eliminated and only the main population is maintained. We compared the performance of BiCo with that of WoA-BiCo on the MW test suite, and the comparison results are summarized in Tables S-IX and S-X in terms of IGD and HV, respectively.

From these two tables, it is evident that BiCo outperformed its variant on most test instances in terms of both IGD and HV. To be specific, BiCo outperformed WoA-BiCo on nine and ten out of 14 instances regarding IGD and HV, respectively, while WoA-BiCo did not beat BiCo on any test instances. It is thus concluded that the archive population is an indispensable part of BiCo.

2) Benefit of the Restricted Mating Selection: One may be interested in the benefit of the restricted mating selection in BiCo. Can it be replaced by random selection? Or can we only use CV or *AD* in the restricted mating selection? To answer these questions, we considered three variants of BiCo, called: 1) WoR-BiCo; 2) CV-BiCo; and 3) AD-BiCo, respectively. In WoR-BiCo, the solutions are randomly selected from the union of the main and archive populations; while in CV-BiCo and AD-BiCo, the solutions are distinguished only by making use of CV and *AD* in the mating selection, respectively. The comparative experiments between BiCo and its three variants were carried out on the MW test suite. The results are summarized in Tables S-XI and S-XII in terms of IGD and HV, respectively.

From Tables S-XI and S-XII, it is observed that BiCo obtained the overall best performance with respect to both IGD and HV. In terms of IGD, BiCo obtained better performance than WoR-BiCo, CV-BiCo, and AD-BiCo on nine, eight, and four instances, respectively, while these three variants did not outperform BiCo on any instances. With respect to HV, BiCo beats WoR-BiCo, CV-BiCo, and AD-BiCo on seven, 11, and two instances, respectively, while losing on none instance. It is easy to conclude that the restricted mating selection plays a very important role in BiCo and both CV and *AD* are necessary for the restricted mating selection.

Remark 3: We also investigated the effectiveness of the updating mechanism for the archive population in Section S-III-A of the supplementary file, the influence of the parameter setting in BiCo in Section S-III-B of the supplementary file, the availability of BiCo on MEMS [59] in Section S-III-C of the supplementary file, the effect of the reference point in Section S-III-D of the supplementary file, the necessity of the feasible information in the angle-based selection scheme in Section S-III-E of the supplementary file, and the extending of BiCo for constrained many-objective optimization in Section S-IV-F of the supplementary file.

VI. Conclusion

In this article, we have proposed a novel CMOEA, called BiCo, by coevolving two populations—the main population and the archive population—from two complementary directions. To update the main population, we employed a variant of NSGA-II-CDP to push the population into the feasible region and then to guide the population toward the PF from the feasible side of the search space. To update the archive population, we first employed a nondominated sorting procedure to find out the nondominated infeasible solutions and then developed an angle-based selection scheme to drive the population toward the PF from the infeasible side of the search space while maintaining good diversity. Besides, to coordinate the interaction between the main and archive populations, we designed a restricted mating selection mechanism to produce promising offspring. We compared the performance of BiCo with that of eight other peer CMOEAs on up to 42 CMOPs. The empirical results demonstrate that BiCo obtained the overall best performance in terms of both IGD and HV. It is thus believed that BiCo is a promising alternative to CMOEA for dealing with a wide variety of CMOPs.

In the future, we will design some other novel approaches to update the main population and the archive population, respectively. Meanwhile, we are going to develop some other efficient mechanisms to make use of the complementary information in the main and archive populations. It is also our plan to devote BiCo to coping with more CMOPs encountered in real-life applications that may involve both equality and inequality constraints [6]. The MATLAB source codes of BiCo can be downloaded from: https://github.com/zhi-zhong/BiCo.

References

[1] R. Chai, A. Savvaris, A. Tsourdos, Y. Xia, and S. Chai, “Solving multiobjective constrained trajectory optimization problem by an extended evolutionary algorithm,” *IEEE Trans. Cybern.*, vol. 50, no. 4, pp. 1630–1643, Apr. 2020.

[2] H. Wang and Y. Jin, “A random forest-assisted evolutionary algorithm for data-driven constrained multiobjective combinatorial optimization of trauma systems,” *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 536–549, Feb. 2020.

[3] B.-S. Chen, W.-Y. Chen, C.-T. Young, and Z. Yan, “Noncooperative game strategy in cyber-financial systems with Wiener and Poisson random fluctuations: LMIs-constrained moea approach,” *IEEE Trans. Cybern.*, vol. 48, no. 12, pp. 3323–3336, Dec. 2018.

[4] K. Deb, A. Pratap, and T. Meyarivan, “Constrained test problems for multi-objective evolutionary optimization,” in *Proc. 1st Int. Conf. Evol. Multicrit. Optim. (EMO)*, 2000, pp. 284–298.

[5] Z. Ma and Y. Wang, “Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons,” *IEEE Trans. Evol. Comput.*, vol. 23, no. 6, pp. 972–986, Dec. 2019.

[6] Z.-Z. Liu and Y. Wang, “Handling constrained multiobjective optimization problems with constraints in both the decision and objective spaces,” *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 870–884, Oct. 2019.

[7] T. Ray, H. K. Singh, A. Isaacs, and W. Smith, “Infeasibility driven evolutionary algorithm for constrained optimization,” in *Constraint-Handling in Evolutionary Optimization*. Heidelberg, Germany: Springer, 2009, pp. 145–165.

[8] Z. Liu, Y. Wang, and B. Wang, “Indicator-based constrained multiobjective evolutionary algorithms,” *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, Dec. 5, 2019, doi: 10.1109/TSMC.2019.2954491.

[9] C. Peng, H.-L. Liu, and E. D. Goodman, “A cooperative evolutionary framework based on an improved version of directed weight vectors for constrained multiobjective optimization with deceptive constraints,” *IEEE Trans. Cybern.*, early access, Jun. 19, 2020, doi: 10.1109/TCYB.2020.2998038.

[10] C. Peng, H.-L. Liu, and E. D. Goodman, “Handling multi-objective optimization problems with unbalanced constraints and their effects on evolutionary algorithm performance,” *Swarm Evol. Comput.*, vol. 55, Jun. 2020, Art. no. 100676, doi: 10.1016/j.swevo.2020.100676.

[11] R. Jiao, S. Zeng, and C. Li, “A feasible-ratio control technique for constrained optimization,” *Inf. Sci.*, vol. 502, no. 4, pp. 201–217, 2019.

[12] R. Jiao, S. Zeng, C. Li, S. Yang, and Y. S. Ong, “Handling constrained many-objective optimization problems via problem transformation,” *IEEE Trans. Cybern.*, early access, Nov. 18, 2020, doi: 10.1109/TCYB.2020.3031642.

[13] C. Peng, H. L. Liu, and F. Gu, “An evolutionary algorithm with directed weights for constrained multi-objective optimization,” *Appl. Soft Comput.*, vol. 60, pp. 613–622, Nov. 2017.

[14] K. Li, R. Chen, G. Fu, and X. Yao, “Two-archive evolutionary algorithm for constrained multi-objective optimization,” *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 303–315, Apr. 2019.

[15] Y. Wang, B.-C. Wang, H.-X. Li, and G. G. Yen, “Incorporating objective function information into the feasibility rule for constrained evolutionary optimization,” *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2938–2952, Dec. 2016.

[16] Z. Fan *et al.*, “An improved epsilon constraint-handling method in MOEA/D for CMOPs with large infeasible regions,” *Soft Comput.*, vol. 23, pp. 12491–12510, Feb. 2019.

[17] C. A. Coello Coello *et al.*, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Heidelberg, Germany: Springer, 2007.

[18] E. Zitzler and L. Thiele, “Multiobjective optimization using evolutionary algorithms—A comparative case study,” in *Proc. Int. Conf. Parallel Problem Solving Nat.*, 1998, pp. 292–301.

[19] C. A. Coello Coello and A. D. Christiansen, “MOSES: A multiobjective optimization tool for engineering design,” *Eng. Optim.*, vol. 31, no. 3, pp. 337–368, 1999.

[20] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[21] Z.-Z. Liu, Y. Wang, and P.-Q. Huang, “AnD: A many-objective evolutionary algorithm with angle-based selection and shift-based density estimation,” *Inf. Sci.*, vol. 509, pp. 400–419, Jan. 2020.

[22] H. Jain and K. Deb, “An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, Part II: Handling constraints and extending to an adaptive approach,” *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 602–622, Aug. 2014.

[23] M. A. Jan and R. A. Khanum, “A study of two penalty-parameterless constraint handling techniques in the framework of MOEA/D,” *Appl. Soft Comput.*, vol. 13, no. 1, pp. 128–148, 2013.

[24] J. Wei and Y. Wang, “An infeasible elitist based particle swarm optimization for constrained multiobjective optimization and its convergence,” *Int. J. Pattern Recognit. Artif. Intell.*, vol. 24, no. 3, pp. 381–400, 2010.

[25] R. Storn and K. Price, “Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces,” *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[26] F. Jiménez, A. F. Gómez-Skarmeta, G. Sánchez, and K. Deb, “An evolutionary algorithm for constrained multi-objective optimization,” in *Proc. Congr. Evol. Comput. (CEC)*, vol. 2, 2002, pp. 1133–1138.

[27] M. Miyakawa, K. Takadama, and H. Sato, “Two-stage non-dominated sorting and directed mating for solving problems with multi-objectives

and constraints," in *Proc. 15th Annu. Conf. Genet. Evol. Comput.*, 2013, pp. 647–654.

[28] K. Deb, "An efficient constraint handling method for genetic algorithms," *Comput. Meth. Appl. Mech. Eng.*, vol. 186, no. 2, pp. 311–338, 2000.

[29] C. A. Coello Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art," *Comput. Meth. Appl. Mech. Eng.*, vol. 191, nos. 11–12, pp. 1245–1287, 2002.

[30] A. Oyama, K. Shimoyama, and K. Fujii, "New constraint-handling method for multi-objective and multi-constraint evolutionary optimization," *Trans. Jpn. Soc. Aeronautical Space Sci.*, vol. 50, no. 167, pp. 56–62, 2007.

[31] A. E. Sorkhabi, M. D. Amiri, and A. R. Khanteymoori, "Duality evolution: An efficient approach to constraint handling in multi-objective particle swarm optimization," *Soft Comput.*, vol. 21, no. 24, pp. 7251–7267, 2017.

[32] Y. G. Woldesenbet, G. G. Yen, and B. G. Tessema, "Constraint handling in multiobjective evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 514–525, Jun. 2009.

[33] M. A. Jan and Q. Zhang, "MOEA/D for constrained multiobjective optimization: Some preliminary experimental results," in *Proc. U.K. Workshop Comput. Intell. (UKCI)*, 2010, pp. 1–6.

[34] L. Jiao, J. Luo, R. Shang, and F. Liu, "A modified objective function method with feasible-guiding strategy to solve constrained multi-objective optimization problems," *Appl. Soft Comput.*, vol. 14, no. 1, pp. 363–380, 2014.

[35] N. Young, "Blended ranking to cross infeasible regions in constrained-multiobjective problems," in *Proc. Comput. Intell. Model. Control Autom. Int. Conf. Intell. Agents Web Technol. Internet Commerce Int. Conf.*, vol. 2, 2005, pp. 191–196.

[36] W. Ning, B. Guo, Y. Yan, X. Wu, J. Wu, and Z. Dan, "Constrained multi-objective optimization using constrained non-dominated sorting combined with an improved hybrid multi-objective evolutionary algorithm," *Eng. Optim.*, vol. 49, no. 10, pp. 1–20, 2017.

[37] H. L. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 450–455, Jun. 2014.

[38] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[39] M. Asafuddoula, T. Ray, R. Sarker, and K. Alam, "An adaptive constraint handling approach embedded MOEA/D," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2012, pp. 1–8.

[40] Z. Fan *et al.*, "An improved epsilon constraint handling method embedded in MOEA/D for constrained multi-objective optimization problems," in *Proc. IEEE Symp. Comput. Intell. (SSCI)*, 2016, pp. 1–8.

[41] B. Qu and P. N. Suganthan, "Constrained multi-objective optimization algorithm with an ensemble of constraint handling methods," *Eng. Optim.*, vol. 43, no. 4, pp. 403–416, 2011.

[42] Z. Fan, Y. Fang, W. Li, X. Cai, C. Wei, and E. Goodman, "MOEA/D with angle-based constrained dominance principle for constrained multi-objective optimization problems," *Appl. Soft Comput.*, vol. 74, pp. 621–633, Jan. 2019.

[43] Z. Fan *et al.*, "Push and pull search for solving constrained multi-objective optimization problems," *Swarm Evol. Comput.*, vol. 44, pp. 665–679, Feb. 2019.

[44] Z. Fan *et al.*, "Push and pull search embedded in an M2M framework for solving constrained multi-objective optimization problems," *Swarm Evol. Comput.*, vol. 54, May 2020, Art. no. 100651, doi: 10.1016/j.swevo.2020.100651.

[45] K. Li, S. Kwong, and K. Deb, "A dual-population paradigm for evolutionary multiobjective optimization," *Inf. Sci.*, vol. 309, pp. 50–72, Jul. 2015.

[46] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Proc. Evol. Methods Design Optim. Control Appl. Ind. Problems (EUROGEN)*, 2001, pp. 95–100.

[47] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, 2011.

[48] Z. Cai and Y. Wang, "A multiobjective optimization-based evolutionary algorithm for constrained optimization," *IEEE Trans. Evol. Comput.*, vol. 10, no. 6, pp. 658–675, Dec. 2006.

[49] P. D. Justesen, "Multi-objective optimization using evolutionary algorithms," Ph.D. dissertation, Dept. Comput. Sci., University of Aarhus, Aarhus, Denmark, 2009.

[50] T. Ray and K. M. Liew, "A swarm metaphor for multiobjective design optimization," *Eng. Optim.*, vol. 34, no. 2, pp. 141–153, 2002.

[51] A. Osyczka and S. Kundu, "A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm," *Struct. Optim.*, vol. 10, pp. 94–99, Oct. 1995.

[52] N. Srinivas and K. Deb, "Multiobjective function optimization using nondominated sorting genetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 2, no. 3, pp. 1301–1308, Sep. 1994.

[53] M. Tanaka, H. Watanabe, Y. Furukawa, and T. Tanino, "GA-based decision support system for multicriteria optimization," in *Proc. IEEE Int. Conf. Syst. Man Cybern. Intell. Syst. 21st Century*, vol. 2, Oct. 1995, pp. 1556–1561.

[54] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization," *IEEE Comput. Intell. Mag.*, vol. 12, no. 4, pp. 73–87, Nov. 2017.

[55] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 694–716, Oct. 2015.

[56] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "Performance of decomposition-based many-objective algorithms strongly depends on Pareto front shapes," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 169–190, Apr. 2017.

[57] Z. Fan, Y. Fang, W. Li, J. Lu, X. Cai, and C. Wei, "A comparative study of constrained multi-objective evolutionary algorithms on constrained multi-objective optimization problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2017, pp. 209–216.

[58] J. Alcalafdez *et al.*, "KEEL: A software tool to assess evolutionary algorithms for data mining problems," *Soft Comput.*, vol. 13, no. 3, pp. 307–318, 2008.

[59] W. Li, Z. Fan, X. Cai, H. Lin, S. Xie, and S. Wang, "Design optimization of MEMS using constrained multi-objective evolutionary algorithm," in *Proc. Companion Annu. Conf. Genet. Evol. Comput.*, 2014, pp. 1175–1180.

Zhi-Zhong Liu received the B.Eng. degree in automation and the Ph.D. degree in control science and engineering from Central South University, Changsha, China, in 2013 and 2019, respectively.

He is a Research Associate with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China. His current research interests include evolutionary computation, swarm intelligence, constrained multiobjective optimization, and evolutionary design.

Bing-Chuan Wang received the B.Eng. degree in automation and the M.S. degree in control science and engineering from Central South University, Changsha, China, in 2013 and 2016, respectively, and the Ph.D. degree in system engineering and engineering management from the City University of Hong Kong, Hong Kong, in 2019.

He is an Assistant Professor with the School of Automation, Central South University. His current research interests include evolutionary computation, constrained optimization, and intelligent modeling.

Ke Tang (Senior Member, IEEE) received the B.Eng. degree from the Huazhong University of Science and Technology, Wuhan, China, in 2002, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2007.

From 2007 to 2017, he was with the School of Computer Science and Technology, University of Science and Technology of China, Hefei, China, first as an Associate Professor from 2007 to 2011, and later, as a Professor from 2011 to 2017. He is currently a Professor with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China. He has over 9000 Google Scholar citations with an H-index of 45. His current research interests include evolutionary computation, machine learning, and their applications.

Dr. Tang was a recipient of the Royal Society Newton Advanced Fellowship in 2015 and the 2018 IEEE Computational Intelligence Society Outstanding Early Career Award. He was an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and served as a member of the editorial board for a few other journals.