A Coevolutionary Framework for Constrained Multiobjective Optimization Problems

Ye Tian, *Member, IEEE*, Tao Zhang, Jianhua Xiao, Xingyi Zhang, *Senior Member, IEEE*, and Yaochu Jin, *Fellow, IEEE*

***Abstract*—Constrained multiobjective optimization problems (CMOPs) are challenging because of the difficulty in handling both multiple objectives and constraints. While some evolutionary algorithms have demonstrated high performance on most CMOPs, they exhibit bad convergence or diversity performance on CMOPs with small feasible regions. To remedy this issue, this article proposes a coevolutionary framework for constrained multiobjective optimization, which solves a complex CMOP assisted by a simple helper problem. The proposed framework evolves one population to solve the original CMOP and evolves another population to solve a helper problem derived from the original one. While the two populations are evolved by the same optimizer separately, the assistance in solving the original CMOP is achieved by sharing useful information between the two populations. In the experiments, the proposed framework is compared to several state-of-the-art algorithms tailored for CMOPs. High competitiveness of the proposed framework is demonstrated by applying it to 47 benchmark CMOPs and the vehicle routing problem with time windows.**

***Index Terms*—Coevolution, constrained multiobjective optimization, evolutionary algorithm, vehicle routing problem.**

Manuscript received November 6, 2019; revised February 1, 2020, April 4, 2020, and May 5, 2020; accepted June 18, 2020. Date of publication June 22, 2020; date of current version January 29, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0100100; in part by the National Natural Science Foundation of China under Grant 61672033, Grant 61822301, Grant 61876123, Grant 61906001, Grant 61590922, and Grant U1804262; in part by the Hong Kong Scholars Program under Grant XJ2019035; in part by the Anhui Provincial Natural Science Foundation under Grant 1808085J06 and Grant 1908085QF271; in part by the State Key Laboratory of Synthetical Automation for Process Industries under Grant PAL-N201805; in part by the Royal Society International Exchanges Program under Grant IEC\NSFC\170279; and in part by the Fundamental Research Funds for the Central Universities under Grant 63192616. *(Corresponding author: Xingyi Zhang.)*

Ye Tian is with the Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, Institutes of Physical Science and Information Technology, Anhui University, Hefei 230601, China (e-mail: field910921@gmail.com).

Tao Zhang and Xingyi Zhang are with the Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, Anhui University, Hefei 230601, China (e-mail: taozhangemail@sina.com; xyzhanghust@gmail.com).

Jianhua Xiao is with the Research Center of Logistics, Nankai University, Tianjin 300071, China (e-mail: jhxiao@nankai.edu.cn).

Yaochu Jin is with the Department of Computer Science, University of Surrey, Guildford GU2 7XH, U.K., and also with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: yaochu.jin@surrey.ac.uk).

This article has supplementary downloadable material available at https://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/TEVC.2020.3004012

I. Introduction

CONSTRAINED multiobjective optimization problems (CMOPs) widely exist in many real-world applications, such as vehicle routing [1], robot gripper optimization [2], and water distribution system design [3]. A CMOP can be mathematically defined as

$$\begin{aligned}
\min_{\mathbf{x}} \quad & \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_M(\mathbf{x})) \\
\text{s.t.} \quad & \mathbf{x} \in \Omega \\
& g_i(\mathbf{x}) \le 0, \quad i = 1, \ldots, p \\
& h_j(\mathbf{x}) = 0, \quad j = 1, \ldots, q
\end{aligned} \tag{1}$$

where $\mathbf{x} = (x_1, \ldots, x_D) \in \Omega$ is a solution consisting of D decision variables; $\Omega \subseteq \mathbb{R}^D$ is the decision space; $\mathbf{f} : \Omega \to \mathbb{R}^M$ consists of M objectives; $g_i(\mathbf{x})$ are p inequality constraints; and $h_j(\mathbf{x})$ are q equality constraints. To solve a CMOP, the solutions should not only minimize the objectives $\mathbf{f}(\mathbf{x})$ as much as possible but also satisfy all the constraints $g_i(\mathbf{x})$ and $h_j(\mathbf{x})$. For example, the vehicle routing problem with time windows (VRPTWs) [4] aims to find the solutions minimizing both the number of vehicles and the total traveled distance, in which the solutions should also satisfy the time windows of customers and capacities of vehicles. Due to the strict constraints of time windows and capacities, it is difficult to find many feasible solutions for the problem, hence, the optimization of number of vehicles and total traveled distance becomes very challenging [5]. In short, existing multiobjective evolutionary algorithms (MOEAs) encounter difficulties on CMOPs [6].

Having been developed for more than two decades, MOEAs have shown high performance in solving various multiobjective optimization problems [7]. While attention has been drawn toward many-objective optimization [8] and large-scale multiobjective optimization [9] in recent years, more research efforts on constrained multiobjective optimization are needed [6], [10], [11]. CMOPs are not the extension of general multiobjective optimization problems with more objectives or decision variables, since the constraints and objectives should be separately handled and balanced [12]. To this end, various constraint handling techniques have been suggested, including the constrained dominance relation of NSGA-II [13], the two-archive collaborative framework of C-TAEA [14], and the biphasic search process of PPS [15].

Although constrained multiobjective optimization has been studied for two decades, there exist some limitations in the state-of-the-art MOEAs [16]. More specifically, existing MOEAs may be incapable of balancing constraints and objectives on CMOPs with small feasible regions [e.g., the feasible region is discrete or far from the unconstrained Pareto front (PF)], which leads to a bad convergence or diversity of the population [6]. Unfortunately, many real-world CMOPs are with small feasible regions (e.g., VRPTW [17]), posing stiff challenges to existing MOEAs. Inspired by the success of coevolutionary algorithms [18], this article proposes a coevolutionary framework for solving CMOPs. The main new contributions of this work are as follows.

1) A coevolutionary constrained multiobjective optimization (CCMO) framework is proposed for solving CMOPs, which aims to solve a CMOP with the assistance of solving a simple helper problem. The proposed CCMO evolves two populations with the same optimizer separately, where the first population is to solve the original CMOP and the second population is to solve a helper problem derived from the original one. The novelty of CCMO mainly lies in the new paradigm of coevolution, in which the cooperation between two populations is much weaker than the cooperation in existing coevolutionary algorithms. Case studies and experimental results demonstrate that the weak cooperation in CCMO is more effective than the strong cooperation in existing MOEAs for solving CMOPs.
2) Based on the proposed framework, an MOEA is proposed by adopting NSGA-II [13] as the optimizer for evolving the two populations. The proposed MOEA is tested on a set of benchmark CMOPs to verify its effectiveness. Moreover, the proposed MOEA is equipped with three local search strategies to solve the VRPTW problem. According to the experimental results, the proposed MOEA outperforms several state-of-the-art MOEAs on both the benchmark CMOPs and the VRPTW problem.

In the remainder of this article, we first introduce the existing MOEAs for CMOPs in Section II. Then, we elaborate on the proposed framework CCMO and examine its performance in comparison to existing MOEAs in Section III. Afterward, the experimental results are detailed in Sections IV and V. Finally, conclusions are drawn and future work is outlined in Section VI.

II. Related Work

In this section, the existing MOEAs for solving CMOPs are introduced. Since this article focuses on a coevolutionary framework, existing coevolutionary constraint handling techniques are also reviewed.

A. Existing MOEAs With Constraint Handling Techniques

Since both constraints and objectives are functions to be minimized, early MOEAs usually treat constraints and objectives equally. For example, in [19], the objective vector of each solution was extended by adding the violation of each constraint, then the nondominated sorting was performed on the extended objective vectors. In [20], the objective values of each solution were modified by considering its constraint violation. In [21], a constrained nondominated rank was defined by integrating the original nondominated rank and a constraint rank.

Although the above constraint handling techniques are straightforward and effective for some simple constraints, it is difficult to make a good balance between constraints and objectives by tuning the penalty factors [15]. Since the feasibility of solutions takes precedence over convergence in CMOPs, some other MOEAs make constraints prior to objectives in dominance relation. NSGA-II [13] embeds feasibility in Pareto dominance, where feasible solutions dominate infeasible solutions and a solution with lower constraint violation dominates another solution with higher constraint violation. To be specific, the constraint violation of each solution $\mathbf{x}$ is first calculated by

$$CV(\mathbf{x}) = \sum_{i=1}^{p} \max\{g_i(\mathbf{x}), 0\} + \sum_{j=1}^{q} \left|h_j(\mathbf{x})\right| \tag{2}$$

then a solution $\mathbf{x}$ is said to dominate another solution $\mathbf{y}$ if the following conditions hold:

1) if $CV(\mathbf{x}) = 0$ and $CV(\mathbf{y}) = 0$ $\forall i \in \{1, \ldots, M\}$ such that $f_i(\mathbf{x}) \leq f_i(\mathbf{y})$ and $\exists j \in \{1, \ldots, M\}$ such that $f_j(\mathbf{x}) < f_j(\mathbf{y})$;
2) otherwise, $CV(\mathbf{x}) < CV(\mathbf{y})$.

This constraint handling technique can be used in other MOEAs based on Pareto dominance. In addition, the number of violated constraints [22], the dominance relation based on constraints [23], and the normalized constraint violation [24] have also been considered.

For decomposition-based MOEAs ignoring Pareto dominance, a similar idea has been widely adopted to give priority to feasible solutions [25]. Specifically, when updating the solution of a weight vector, the solution with the lowest constraint violation is preferred; if there exist multiple feasible solutions, they compete with each other based on the aggregation function values on the weight vector. To relax the definition of feasibility, the algorithm in [26] regards solutions with small constraint violations as feasible ones; the algorithm in [27] treats two infeasible solutions as nondominated if the angle between them is large; the algorithm in [28] preserves the infeasible solutions in isolated regions; and the algorithm in [29] regards the objective value and constrain violation as a biobjective optimization problem to be optimized.

More recently, some MOEAs with multiple stages or populations have been developed to solve CMOPs, which can dynamically adjust the balance between constraints and objectives. The PPS framework [15] divides the search process into a push stage and a pull stage. In the push stage, the population is evolved without considering any constraints; while in the pull stage, the population is evolved by considering all the constraints and objectives. The ToP framework [10] also suggests a two-stage search process, where all the constraints and a single objective are considered in the first stage, and all the constraints and objectives are considered in the second stage. For C-TAEA [14], a convergence-oriented archive (CA) is evolved by optimizing both the constraints and objectives, and a diversity-oriented archive (DA) is evolved by optimizing only the objectives.

B. Coevolutionary Constraint Handling Techniques

Coevolutionary algorithms have shown effectiveness on many challenging problems, including large-scale optimization problems [30], dynamic optimization problems [31], many-objective optimization problems [32], and so on, but the development of coevolution for constraint handling is still in the infancy [18].

The coevolutionary constraint handling technique was first used in solving constrained single-objective optimization problems. In [33] and [34], a coevolutionary genetic algorithm and a differential evolution algorithm were proposed for constrained optimization, respectively. These two algorithms evolve multiple populations simultaneously, where each population is assigned an independent penalty factor for balancing constraints and objective. In the memetic coevolutionary differential evolution algorithm proposed in [35], a population is to minimize the objective regardless of constraints, and another population is to minimize the violation of constraints regardless of the objective. In [36], the algorithm decomposes the constraints and evolves one population for each constraint, where each population first tries to satisfy its assigned constraint and then the other constraints from other populations.

In terms of constrained multiobjective optimization, a multiobjective particle swarm optimization algorithm was proposed in [37], which uses one population to store feasible particles and the other population to store infeasible particles, where feasible particles are updated toward Pareto optimality and infeasible particles are updated toward feasible particles. Besides, an infeasible solution can migrate to the feasible population once it becomes feasible. In [38], the differential evolution framework cooperatively evolves M populations for solving M-constrained single-objective optimization problems and evolves a population for solving the constrained M-objective optimization problem. C-TAEA [14] is also an evolutionary algorithm evolving two populations. In C-TAEA, the CA is evolved to optimize the constraints and objectives, and the DA is evolved to optimize only the objectives. Besides, the two populations cooperate with each other in the mating selection and environmental selection.

As revealed in some recent studies [6], [10], most existing MOEAs encounter difficulties in obtaining a set of well-converged and well-distributed feasible solutions for CMOPs. In particular, the limitation of coevolutionary MOEAs is mainly due to the strong cooperation between populations. In contrast, the proposed coevolutionary framework holds a weak cooperation between populations. The proposed framework is described in Section III-A, and the superiority of the weak cooperation over strong cooperation in solving CMOPs is analyzed in Section III-B in detail.

III. Proposed Framework

A. Procedure of CCMO

As presented in Algorithm 1, the proposed CCMO starts with the random initialization of two populations *Population*1 and *Population*2 with size N. In each generation, two-parent sets *Parent*1 and *Parent*2 are selected from *Population*1 and *Population*2 by the mating selection strategy of the employed MOEA, respectively. Then, each of the two-parent sets is used to generate an offspring population by the operators of the employed MOEA. Afterward, both *Population*1 and *Population*2 are combined with the two offspring populations, and further truncated by the environmental selection strategy of the employed MOEA. Finally, *Population*1 is returned as the final output. Note that the solutions in *Population*1 are always evaluated by the original CMOP $\mathbf{f}_{\text{origin}}$, and the solutions in *Population*2 are always evaluated by a helper problem $\mathbf{f}_{\text{help}}$ derived from $\mathbf{f}_{\text{origin}}$. In general, $\mathbf{f}_{\text{help}}$ consists of part of the objectives and constraints in $\mathbf{f}_{\text{origin}}$, so the calculation of $\mathbf{f}_{\text{help}}$ does not introduce additional function evaluation.

In short, the proposed CCMO evolves *Population*1 to solve the original problem $\mathbf{f}_{\text{origin}}$ and evolves *Population*2 to solve a helper problem $\mathbf{f}_{\text{helper}}$, where the assistance in solving $\mathbf{f}_{\text{origin}}$ is achieved by sharing the offsprings generated by the two populations. Since $\mathbf{f}_{\text{help}}$ is easier than $\mathbf{f}_{\text{origin}}$, the solutions in *Population*2 usually have better convergence and diversity than those in *Population*1, so the offsprings generated by *Population*2 can possibly improve the convergence of *Population*1. Besides, *Population*1 may get trapped into a narrow feasible region, while the offsprings in *Population*2 can help it jump out of local optimums. On the other hand, the offsprings generated by *Population*1 can also enhance the convergence speed of *Population*2 to some extent.

Algorithm 1: Procedure of the Proposed CCMO

Input: $\mathbf{f}_{origin}$ (original CMOP), $\mathbf{f}_{help}$ (helper problem), *MOEA* (the employed MOEA), N (population size)
Output: P (final population)

*Population*1 ← *RandomInitialization*(N);
*Population*2 ← *RandomInitialization*(N);
Evaluate *Population*1 by $\mathbf{f}_{origin}$;
Evaluate *Population*2 by $\mathbf{f}_{help}$;
while *termination criterion not fulfilled* **do**
    *Parent*1 ← Select $N/2$ parents from *Population*1 by the mating selection of *MOEA*;
    *Parent*2 ← Select $N/2$ parents from *Population*2 by the mating selection of *MOEA*;
    *Off*1 ← Generate $N/2$ offsprings based on *Parent*1 by the operators of *MOEA*;
    *Off*2 ← Generate $N/2$ offsprings based on *Parent*2 by the operators of *MOEA*;
    *Population*1 ← *Population*1 ∪ *Off*1 ∪ *Off*2;
    *Population*2 ← *Population*2 ∪ *Off*1 ∪ *Off*2;
    Evaluate *Population*1 by $\mathbf{f}_{origin}$;
    Evaluate *Population*2 by $\mathbf{f}_{help}$;
    *Population*1 ← Select N solutions from *Population*1 by the environmental selection of *MOEA*;
    *Population*2 ← Select N solutions from *Population*2 by the environmental selection of *MOEA*;
return *Population*1;

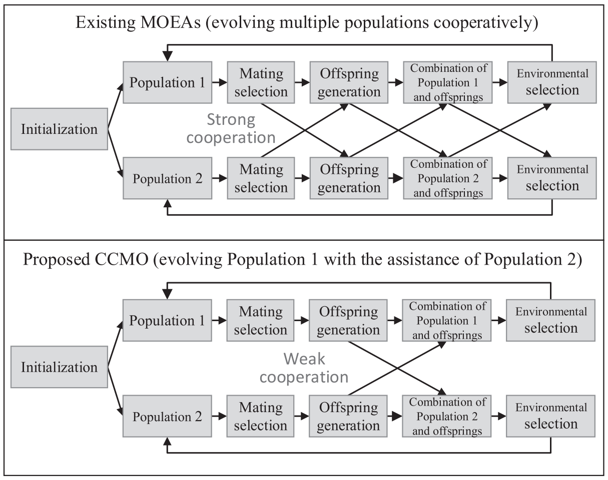

Fig. 1. Procedures of many existing coevolution-based MOEAs and the proposed CCMO.

It is worth noting that although there exist a few coevolutionary MOEAs tailored for CMOPs, the purpose of using coevolution in the proposed framework is totally different. Existing coevolutionary MOEAs evolve multiple populations to better balance convergence and diversity, where one population is to converge to the global PF and the other population is to explore the undeveloped areas for better diversity [14], [39]. In contrast, the proposed coevolutionary framework aims to solve a CMOP with the assistance of solving a helper problem. The difference between the ideas of CCMO and existing MOEAs is reflected in the difference between the procedures of evolution. As illustrated in Fig. 1, existing MOEAs make populations cooperate with each other in the mating selection, offspring generation, and environmental selection; in contrast, the populations in CCMO are evolved separately, only sharing all the offsprings in each generation. In other words, existing MOEAs use a strong cooperation to evolve the populations, whereas CCMO uses a weak cooperation to give each population the freedom to evolve toward the optimal PF of its own problem, and shares the offsprings generated by all the populations to assist in solving the original CMOP. Now, a key question may arise—is such a weak cooperation more effective than a strong cooperation for solving CMOPs? In the next section, the answer is cleared by several empirical studies.

B. Analysis of CCMO

To analyze the effectiveness and understand the mechanism of the proposed CCMO, it is combined with the constrained NSGA-II to tackle challenging CMOPs. More specifically, the constrained dominance relation and crowding distance are adopted as the criteria for selecting parents (lines 6 and 7 of Algorithm 1) and truncating populations (lines 14 and 15 of Algorithm 1), and the simulated binary crossover [40] and polynomial mutation [41] are adopted for generating offsprings (lines 8 and 9 of Algorithm 1). Here, the original CMOP without any constraint is regarded as the helper problem $\mathbf{f}_{\text{help}}$, which is the same as those in some existing MOEAs, such as PPS and C-TAEA.

The proposed CCMO is compared to NSGA-II and C-TAEA, where NSGA-II is adopted as the basic optimizer in CCMO and C-TAEA is a state-of-the-art MOEA evolving two populations (i.e., CA and DA) cooperatively. Fig. 2 plots the populations in the early, middle, and last generations of the compared MOEAs on 2-objective MW11, where the parameter settings are the same as those described in Section IV-A. MW11 has three small feasible regions as shown in the figure, which pose challenges to MOEAs in terms of diversity. As shown in the first column of Fig. 2, the population of NSGA-II can only converge to a single feasible region in the early generations and cannot spread to the other feasible regions at last, though the population distributes uniformly in one of the three feasible regions. This is because NSGA-II is driven by the constrained dominance relation, which always prefers feasible solutions and has trouble in jumping over infeasible regions. For the populations of C-TAEA, CA has a much better spread than the population of NSGA-II in the early generations, due to the diversity enhancement provided by DA. In the middle generations, CA locates in all the feasible regions since it considers both the constraints and objectives, while the population DA locates on the unconstrained PF since it considers only the objectives. However, C-TAEA usually selects a parent from CA and a parent from DA to generate an offspring, so most offsprings locate between CA and DA. In other words, the generated offsprings do not have good feasibility or good convergence, and are unable to update CA for better diversity. Therefore, C-TAEA cannot find a sufficient number of feasible and well-converged solutions at last. As for the populations of CCMO, *Population*1 can also have a good spread due to the assistance of *Population*2. But in contrast to C-TAEA, the parents in CCMO are selected from *Population*1 and *Population*2 separately, which makes some offsprings around *Population*1 and some others around *Population*2, and the diversity of *Population*1 can be enhanced by the offsprings around it. Hence, *Population*1 has better diversity than CA at last.

Fig. 3 depicts the populations in the early, middle, and last generations of the compared MOEAs on 2-objective C1-DTLZ3, which has a highly multimodal landscape and a band of the infeasible region, posing challenges to MOEAs in terms of convergence. As shown in the first column of Fig. 2, the population of NSGA-II can converge to the boundary $x^2 + y^2 = 6$ of a feasible region in the middle generations, but it has trouble in jumping over the infeasible band since the constrained dominance relation prefers feasible solutions, even though the feasible solutions have much worse convergence than some infeasible solutions. Once a feasible offspring under the infeasible band is found, the whole population can quickly jump over the infeasible band and converge to the global PF. However, this scenario rarely happens since the infeasible band is wide. For the populations of C-TAEA, the solutions in CA spread along the boundary of a feasible region, and most solutions in DA have worse convergence than those in CA in the middle generations. This is because C-TAEA updates DA by selecting the solutions having different directions from those in CA. While most solutions are feasible, the well-converged ones are put into

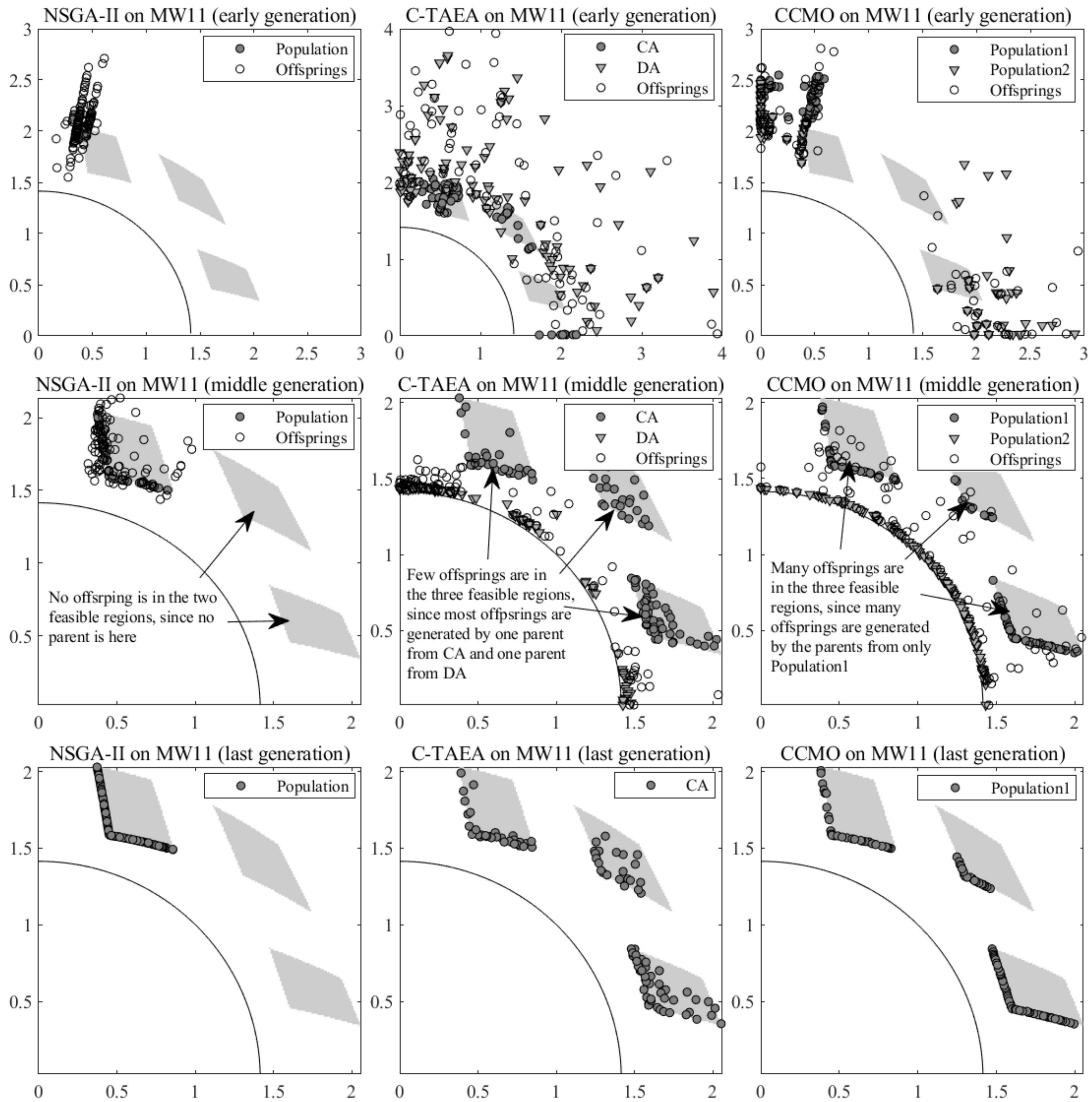

Fig. 2. Populations in the early, middle, and last generations of NSGA-II, C-TAEA, and CCMO on 2-objective MW11. The black line denotes the unconstrained PF of the problem, and the gray surfaces denote the feasible regions of the problem.

CA, and those with different directions must have worse convergence and are further put into DA. Since the generated offsprings locate between CA and DA, they are unlikely to have significantly better convergence than the solutions in CA. That is, the offsprings can hardly jump over the infeasible band to enable CA to evolve toward the global PF. For the populations of CCMO, *Population*1 is similar to CA since both of them are evolved without considering the constraints. In contrast, *Population*2 is different from DA since CCMO updates *Population*2 without considering the directions of solutions in the other population. Hence, most solutions in *Population*2 are in the infeasible region. Since some offsprings are generated around *Population*2, it is likely to generate an offspring that can jump out of the infeasible region, and such an offspring can help *Population*1 jump over the infeasible band and evolve toward the global PF. As a consequence, CCMO has better convergence performance than NSGA-II and C-TAEA since it can jump over infeasible regions more easily.

CCMO shows better diversity and convergence performance than existing MOEAs on MW11 and C1-DTLZ3, respectively. The superiority of CCMO on MW11 is due to the fact that CCMO generates offsprings based on the parents separately selected from *Population*1 and *Population*2; in other words, the two populations do not cooperate with each other in the mating selection of CCMO. Besides, the superiority of CCMO on C1-DTLZ3 is due to the fact that CCMO selects *Population*2 without considering the solutions in *Population*1, which means that the two populations do not cooperate with each other in the environmental selection of CCMO. Therefore, it can be confirmed that the weak cooperation in CCMO is more promising than the strong cooperation in C-TAEA for solving CMOPs. As further illustrations, the proposed CCMO is compared to its two variants with stronger cooperations. The first variant enables populations to cooperate with each other in the mating selection, i.e., each offspring is generated based on one parent from *Population*1 and one parent from *Population*2. The second variant enables populations to cooperate with each

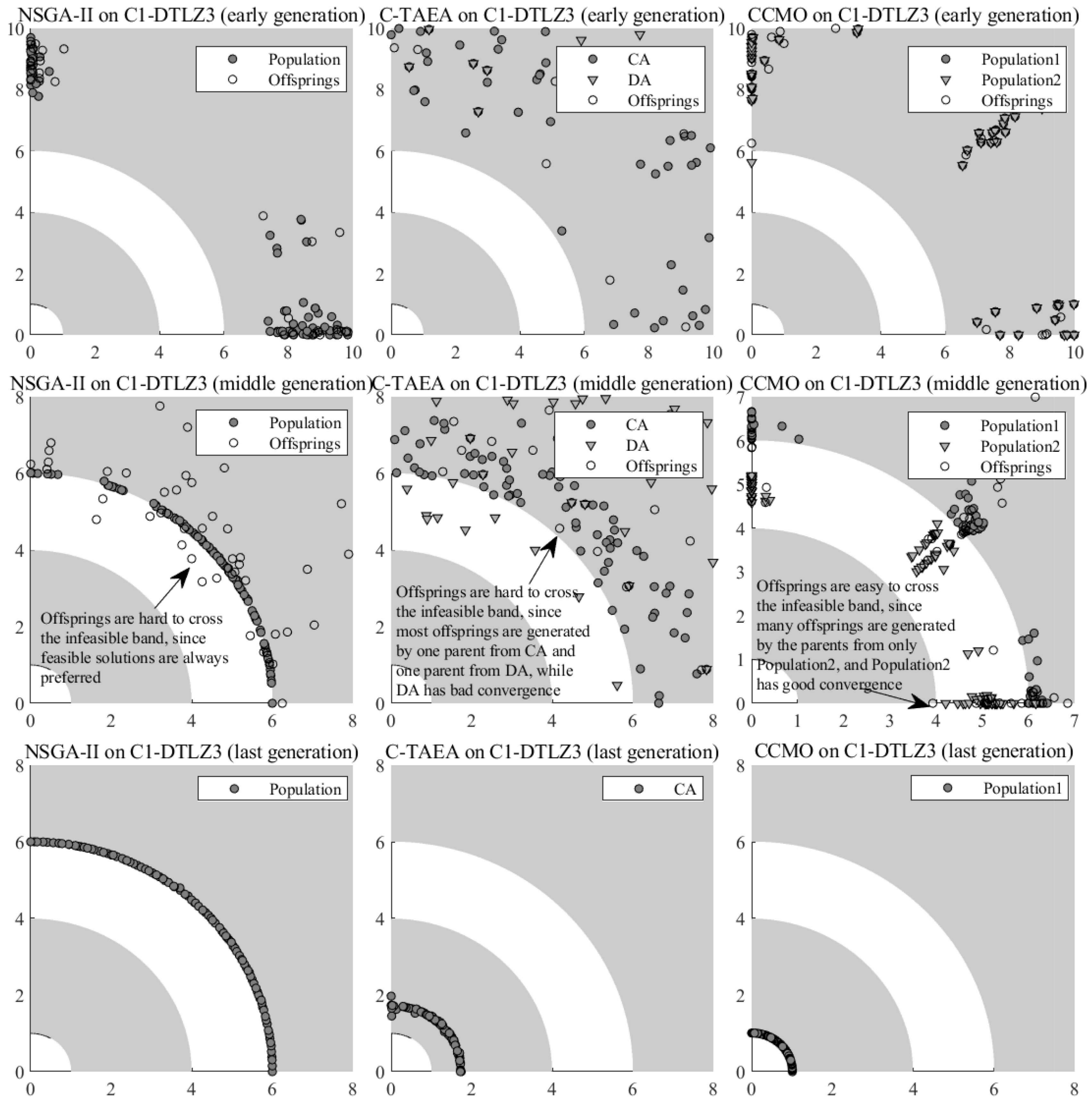

Fig. 3. Populations in the early, middle, and last generations of NSGA-II, C-TAEA, and CCMO on 2-objective C1-DTLZ3. The black line denotes the unconstrained PF of the problem, and the gray surfaces denote the feasible regions of the problem.

other in the environmental selection, i.e., the solutions having different directions from those in *Population*1 are put into *Population*2. According to the convergence profiles of IGD values shown in Fig. 4, it can be observed that the original CCMO converges faster than its two variants on MW11 and C1-DTLZ3. Therefore, the superiority of the weak cooperation in CCMO over stronger cooperations can be further confirmed.

C. Computational Complexity of CCMO

Since the proposed CCMO does not suggest any specific selection strategy as shown in Algorithm 1, the time complexity of the proposed CCMO is mainly determined by the employed MOEA (e.g., NSGA-II). Assuming that N is the population size, D is the number of decision variables, and M is the number of objectives, the worst case time complexities of the mating selection, genetic operators, and environmental selection of NSGA-II are $O(N)$, $O(ND)$, and $O(MN^2)$, respectively [13]. Since CCMO evolves two populations with the same strategies, the worst case time complexities of the mating selection, genetic operators, and environmental selection of CCMO with NSGA-II are $2 \times O(N/2) = O(N)$, $2 \times O([N/2]D) = O(ND)$, and $2 \times O(MN^2) = O(MN^2)$. As a consequence, the proposed CCMO has the same worst case time complexity as the employed MOEA, but it is in fact slower than the employed MOEA since it performs each search strategy twice a generation.

D. Remarks

Although the idea of evolving multiple populations has been adopted in several existing MOEAs for solving CMOPs, the coevolutionary framework of CCMO can exhibit better performance as illustrated above. To summarize, the advantages of CCMO mainly lie in the following three aspects.

1) The core idea of CCMO is to solve a difficult CMOP with the assistance of solving a helper problem, but not to evolve multiple populations cooperatively. To this end, CCMO enables each population to concentrate on

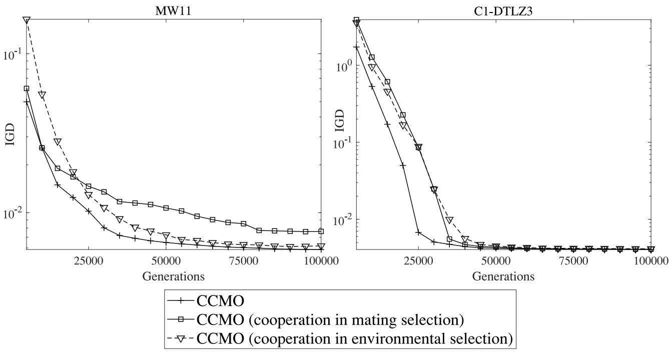

Fig. 4. Convergence profiles of IGD values obtained by CCMO, CCMO with cooperation in the mating selection (i.e., each offspring is generated based on two parents from different populations), and CCMO with cooperation in environmental selection (i.e., solutions having different directions from those in *Population*1 are put into *Population*2) on 2-objective MW11 and C1-DTLZ3, averaged over 30 runs.

the optimization of its own problem, without the disturbance of the other population solving the other problem. Therefore, the helper problem can be specially handled by a population, and this population can generate some well-converged and well-distributed offsprings to assist in solving the difficult CMOP. It is worth mentioning that a similar idea has been adopted in multifactorial evolution [42], which employs a unified representation to simultaneously solve different problems with the assistance of the implicit transfer of useful genetic material between problems [43].

2) Due to the weak cooperation between populations, the proposed CCMO is flexible and light. On the one hand, CCMO can be easily embedded with most existing MOEAs, since the two populations in CCMO are evolved by the same MOEA separately. On the other hand, CCMO does not suggest any novel selection strategy or introduce any new parameter, which is easy to implement without a significant increase in computational complexity.

3) The proposed CCMO evolves the second population *Population*2 to optimize a helper problem derived from the original CMOP, where the helper problem is not necessarily an unconstrained problem as those in existing MOEAs (e.g., PPS and C-TAEA). That is, when using CCMO to solve a specific CMOP, the helper problem can be empirically adjusted for better performance. In fact, for the experiments in the next two sections, the helper problem is set to the original problem without any constraint when solving the benchmark CMOPs, while it is set to the original problem with fewer constraints when solving the VRPTW problem.

IV. Comparisons on Benchmark Problems

This section verifies the performance of the proposed CCMO by comparing it to the constrained NSGA-II [13], PPS [15], C-TAEA [14], and ToP [10] on the constrained DTLZ test suite [14], [25], the MW test suite [6], the LIR-CMOP test suite [44], and the DOC test suite [10]. The experiments are implemented on the evolutionary multiobjective optimization platform [45].

A. Parameter Settings

1) Problems: The number of objectives M and the number of decision variables D of each benchmark problem are set as follows. For the ten constrained DTLZ problems, $M = 3$, $D = 7$ for C1-DTLZ1, DC1-DTLZ1, DC2-DTLZ1, and DC3-DTLZ1, and $D = 12$ for the remaining problems. For the 14 MW problems, $M = 3$ for MW4, MW8, and MW14, $M = 2$ for the remaining problems, and $D = 15$. For the 14 LIR-CMOP problems, $M = 3$ for LIR-CMOP13 and LIR-CMOP14, $M = 2$ for the remaining problems, and $D = 10$. For thenine DOC problems, $M = 3$ for DOC8 and DOC9, $M = 2$ for the remaining problems, and D is fixed to different values for different problems [10].

2) Algorithms: The parameters of all the compared algorithms are set as suggested in their original papers, which have demonstrated the high performance of these parameter settings. For the PPS framework, it is embedded with the constrained MOEA/D, where the parameter settings are $\alpha = 0.95$, $\tau = 0.1$, $cp = 2$, and $l = 20$. For the ToP framework, it is embedded with the constrained NSGA-II, where the first phase ends when the feasibility proportion P_f is larger than 1/3 or the difference δ is less than 0.2. For the proposed CCMO, it is also embedded with the constrained NSGA-II; to further enhance the population diversity, the truncation strategy in SPEA2 [46] is adopted in the environmental selection instead of crowding distance. Besides, the helper problem in CCMO is set to the original problem without any constraint.

3) Genetic Operators: NSGA-II, C-TAEA, and CCMO adopt the simulated binary crossover [40] and the polynomial mutation [41] to generate offsprings, while PPS and ToP adopt the differential evolution [47] and the polynomial mutation to generate offsprings. The probability of simulated binary crossover is set to 1, the probability of polynomial mutation is set to $1/D$ (D denotes the number of decision variables), the distribution index of both crossover and mutation is set to 20, and the parameters CR and F in differential evolution are set to 1 and 0.5, respectively.

4) Population Size and the Number of Function Evaluations: For a fair comparison, the population size is set to 100 on problems with two objectives and 105 on problems with three objectives for all the compared MOEAs. The total number of function evaluations of all populations is adopted as the termination criterion for all the compared MOEAs, which is set to a sufficiently large value to enable each MOEA to converge. Specifically, the number of function evaluations is set to 100 000 for the constrained DTLZ and MW problems, and set to 300 000 for the LIR-CMOP and DOC problems.

B. Experimental Results on Constrained DTLZ and MW Problems

Table I presents the mean value and standard deviation of the IGD values obtained by NSGA-II, PPS, C-TAEA, ToP, and CCMO on the constrained DTLZ test suite and the MW

TABLE I
IGD VALUE OF NSGA-II, PPS, C-TAEA, TOP, AND CCMO ON CONSTRAINED DTLZ AND MW PROBLEMS. THE BEST RESULT IN EACH ROW IS HIGHLIGHTED. "N/A" INDICATES THAT NO FEASIBLE SOLUTION IS FOUND. "+," "−," AND "≈" INDICATE THAT THE RESULT IS SIGNIFICANTLY BETTER, SIGNIFICANTLY WORSE, AND STATISTICALLY SIMILAR TO THAT OBTAINED BY CCMO, RESPECTIVELY

Problem	NSGA-II	PPS	C-TAEA	ToP	CCMO
C1-DTLZ1	2.6661e-2 (1.42e-3) −	2.6860e-2 (8.10e-4) −	2.3314e-2 (2.82e-4) ≈	3.7269e-1 (0.00e+0) −	1.9944e-2 (1.48e-4)
C2-DTLZ2	5.6562e-2 (2.71e-3) −	5.6032e-2 (1.80e-3) ≈	5.6306e-2 (1.07e-3) ≈	6.1116e-2 (5.93e-3) −	4.2735e-2 (4.96e-4)
C1-DTLZ3	7.0281e+0 (2.81e+0) −	1.0681e+0 (2.81e+0) −	1.8588e-1 (3.23e-1) ≈	2.2301e-1 (2.29e-1) −	5.3304e-2 (5.22e-4)
C3-DTLZ4	1.2577e-1 (3.40e-3) −	1.7316e-1 (9.09e-2) −	1.1292e-1 (2.58e-3) ≈	1.4333e-1 (5.55e-3) −	9.4618e-2 (1.28e-3)
DC1-DTLZ1	1.4887e-2 (8.86e-4) ≈	2.0029e-2 (1.47e-3) −	1.5168e-2 (2.08e-4) ≈	2.5308e-2 (6.93e-3) −	1.1373e-2 (1.38e-4)
DC1-DTLZ3	1.2984e-1 (3.05e-3) ≈	1.6964e-1 (1.03e-1) ≈	1.3094e-1 (2.39e-3) −	6.3444e-1 (9.37e-1) −	1.1407e-1 (7.71e-4)
DC2-DTLZ1	N/A	2.8587e-2 (4.07e-4) ≈	2.3267e-2 (1.84e-4) ≈	N/A	2.0127e-2 (1.29e-4)
DC2-DTLZ3	N/A	1.5810e-1 (2.02e-1) ≈	2.2512e-1 (2.33e-1) ≈	N/A	5.2954e-2 (3.17e-4)
DC3-DTLZ1	1.5541e-1 (1.14e-1) −	1.3250e-2 (7.78e-4) −	9.2340e-3 (2.26e-4) ≈	2.9054e+0 (3.85e+0) −	6.8172e-3 (2.73e-5)
DC3-DTLZ3	1.6939e+0 (3.23e-1) −	9.7563e-1 (9.01e-1) ≈	1.7093e-1 (3.38e-3) ≈	7.3461e+0 (4.65e+0) −	1.5811e-1 (1.16e-3)
MW1	2.0033e-3 (8.06e-5) ≈	3.1064e-3 (1.94e-4) −	2.0164e-3 (7.32e-5) ≈	N/A	1.6141e-3 (1.02e-5)
MW2	4.0008e-2 (2.35e-2) ≈	1.4630e-1 (1.03e-1) ≈	1.1948e-2 (6.49e-3) ≈	1.4281e-1 (1.21e-1) ≈	3.0184e-2 (2.22e-2)
MW3	5.9557e-3 (2.83e-4) ≈	6.3657e-3 (4.13e-4) −	4.9220e-3 (1.89e-4) ≈	5.2656e-1 (4.36e-1) −	4.6882e-3 (1.31e-4)
MW4	5.5794e-2 (2.07e-3) ≈	6.2161e-2 (6.45e-3) −	4.6642e-2 (3.42e-4) ≈	N/A	4.0786e-2 (4.14e-4)
MW5	3.5915e-1 (3.42e-1) −	3.7649e-1 (3.86e-1) ≈	1.2110e-2 (3.65e-3) ≈	N/A	5.2249e-4 (1.09e-4)
MW6	2.4507e-2 (1.32e-2) ≈	5.4190e-1 (2.65e-1) ≈	1.1619e-2 (8.73e-3) ≈	9.5561e-1 (3.60e-1) −	2.3629e-2 (8.73e-3)
MW7	5.0207e-3 (2.11e-4) ≈	5.7817e-3 (4.86e-4) ≈	6.4789e-3 (7.23e-4) −	5.5588e-2 (8.17e-2) −	4.7869e-3 (2.23e-4)
MW8	6.5857e-2 (6.09e-3) ≈	1.5099e-1 (5.18e-2) −	5.4147e-2 (1.63e-3) ≈	6.0557e-1 (3.62e-1) −	4.3866e-2 (2.38e-3)
MW9	5.2281e-3 (2.83e-4) ≈	9.7001e-1 (6.45e-1) −	1.0287e-2 (4.61e-4) −	2.2709e-1 (5.11e-1) −	4.3755e-3 (1.54e-4)
MW10	1.0414e-1 (3.36e-2) ≈	4.8307e-1 (2.09e-1) ≈	1.0789e-2 (1.04e-2) ≈	6.8925e-1 (0.00e+0) −	5.1263e-2 (4.11e-2)
MW11	2.2297e-1 (3.15e-1) −	7.5458e-3 (3.35e-4) ≈	1.4141e-2 (1.81e-3) −	6.8934e-1 (1.46e-1) −	6.0966e-3 (2.27e-4)
MW12	5.4987e-3 (1.66e-4) ≈	1.1671e-2 (9.36e-3) −	7.7417e-3 (7.88e-4) −	8.5904e-1 (7.09e-2) −	4.7738e-3 (1.07e-4)
MW13	2.9604e-1 (4.74e-1) ≈	5.2455e-1 (3.81e-1) −	2.5484e-2 (1.19e-2) ≈	6.6615e-1 (4.12e-1) −	6.3957e-2 (3.51e-2)
MW14	1.2066e-1 (4.00e-3) −	1.3052e-1 (7.47e-3) ≈	1.1030e-1 (4.13e-3) ≈	3.5506e-1 (3.87e-1) ≈	9.8086e-2 (9.20e-4)
+/ − / ≈	0/9/13	0/12/12	0/5/19	0/17/2	

test suite for 30 independent runs. The IGD values on each problem are calculated according to approximately 10 000 reference points sampled on the PF of the problem by the methods suggested in [48]. As shown in the table, the proposed CCMO obtains the best results on 20 problems, which is followed by C-TAEA achieving four best results. Besides, NSGA-II, PPS, and ToP perform the best on none of the 24 problems. Table I also gives the statistical results obtained by the Friedman test with Bonferroni correction at a significance level of 0.05 [49]. It can be found that CCMO significantly outperforms NSGA-II, PPS, C-TAEA, and ToP on 9, 12, 5, and 17 problems, respectively.

Fig. 5 plots the feasible and nondominated solutions with median IGD value among 30 runs obtained by the five MOEAs on C1-DTLZ3, MW5, and MW8. For C1-DTLZ3 with a highly multimodal landscape, it is obvious that CCMO has better convergence performance than the other MOEAs. As further evidenced by Fig. 6, CCMO has much faster convergence speed than the other MOEAs on C1-DTLZ3. As illustrated in Fig. 3, the fast convergence speed of CCMO is mainly attributed to the offsprings generated by the population for the helper problem, which enables the population for the original CMOP to jump over infeasible regions more easily. For MW5 and MW8 with discontinuous feasible regions, CCMO exhibits better diversity performance than the other MOEAs, which is owed to the offsprings generated by the population for the original CMOP as illustrated in Fig. 2. In short, the weak cooperation between *Population*1 for the original CMOP and *Population*2 for the helper problem can effectively improve the convergence and diversity of *Population*1. On the one hand, since the constrained DTLZ problems have multimodal landscape and wide infeasible band, the offsprings generated by *Population*2 can help *Population*1 jump over infeasible regions for better convergence. On the other hand, since the MW problems have small and discontinuous feasible regions, the offsprings generated by *Population*1 can improve the diversity of *Population*1. In contrast, these benefits cannot be achieved in a strong cooperation, since most offsprings are generated between the two populations.

Furthermore, Table II lists the proportion of feasible and nondominated solutions in the final population obtained by the five MOEAs. For NSGA-II with the constrained dominance relation, the obtained solutions are all feasible and nondominated on 19 out of the 24 problems. For PPS with a multistage framework, the obtained solutions are all feasible and nondominated on 23 problems. C-TAEA with a coevolutionary framework obtains fewer feasible and nondominated solutions than NSGA-II and PPS. ToP can only obtain a few feasible and nondominated solutions on the 24 problems, since the strategies in ToP are tailored for solving the CMOPs with constraints in both the decision and objective spaces [10]. Besides, the proposed CCMO is able to obtain a sufficient number of feasible and nondominated solutions on all the 24 problems except for MW5, where the PF of MW5 contains only several isolated Pareto-optimal solutions.

C. Experimental Results on LIR-CMOP and DOC Problems

The proposed CCMO is further challenged on more difficult test suites, namely, LIR-CMOP and DOC. LIR-CMOP contains 14 CMOPs with small feasible regions and complicated linkages between position and distance variables, and

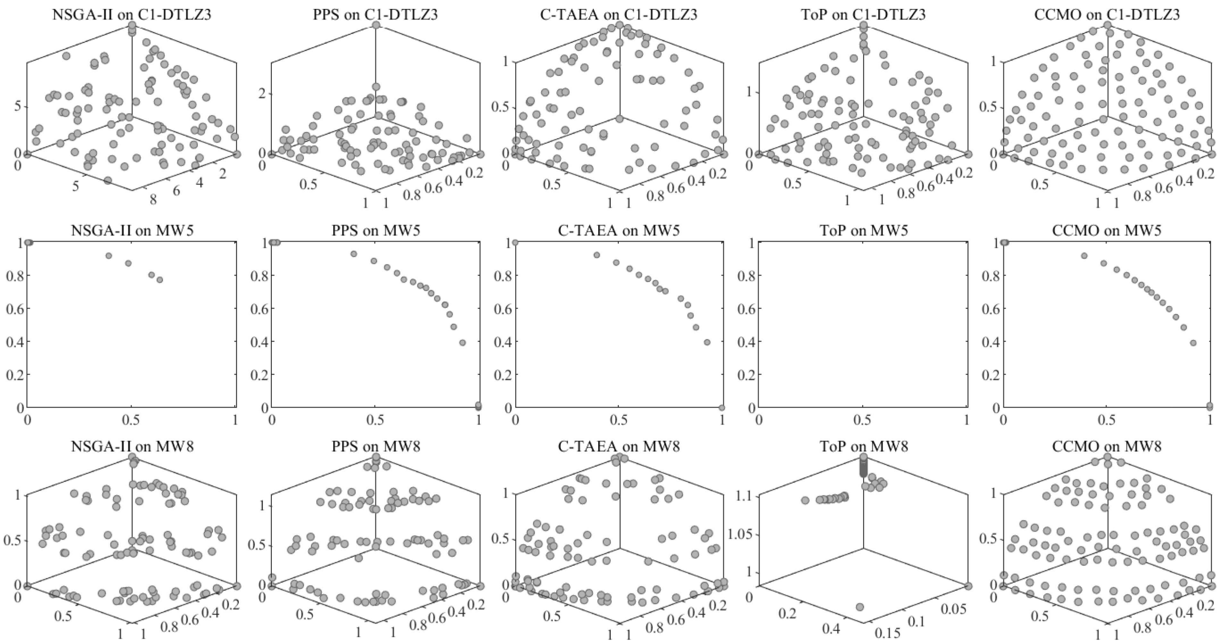

Fig. 5. Feasible and nondominated solutions with median IGD value among 30 runs obtained by NSGA-II, PPS, C-TAEA, ToP, and CCMO on C1-DTLZ3, MW5, and MW8.

TABLE II
PROPORTION OF FEASIBLE AND NONDOMINATED SOLUTIONS IN THE POPULATION OBTAINED BY NSGA-II, PPS, C-TAEA, TOP, AND CCMO ON CONSTRAINED DTLZ AND MW PROBLEMS. THE BEST RESULTS IN EACH ROW ARE HIGHLIGHTED. "+," "−," AND "≈" INDICATE THAT THE RESULT IS SIGNIFICANTLY BETTER, SIGNIFICANTLY WORSE, AND STATISTICALLY SIMILAR TO THAT OBTAINED BY CCMO, RESPECTIVELY

Problem	NSGA-II	PPS	C-TAEA	ToP	CCMO
C1-DTLZ1	1.0000e+0 (0.00e+0) ≈	9.9725e-1 (5.09e-3) ≈	1.0000e+0 (0.00e+0) ≈	1.0875e-1 (4.42e-2) −	1.0000e+0 (0.00e+0)
C2-DTLZ2	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0)
C1-DTLZ3	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0)
C3-DTLZ4	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0)
DC1-DTLZ1	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0)
DC1-DTLZ3	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0)
DC2-DTLZ1	3.5375e-1 (5.93e-2) −	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	1.6125e-1 (3.64e-2) −	1.0000e+0 (0.00e+0)
DC2-DTLZ3	4.3750e-1 (4.18e-1) −	1.0000e+0 (0.00e+0) ≈	9.4918e-1 (1.44e-1) ≈	1.4000e-1 (4.63e-2) −	1.0000e+0 (0.00e+0)
DC3-DTLZ1	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	9.4500e-1 (1.56e-1) ≈	1.0000e+0 (0.00e+0)
DC3-DTLZ3	9.9750e-1 (7.07e-3) ≈	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	9.8250e-1 (3.62e-2) ≈	1.0000e+0 (0.00e+0)
MW1	8.8125e-1 (3.36e-1) ≈	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	3.7500e-2 (1.75e-2) −	1.0000e+0 (0.00e+0)
MW2	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	6.0500e-1 (3.92e-1) −	1.0000e+0 (0.00e+0)
MW3	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	4.2125e-1 (4.79e-1) −	1.0000e+0 (0.00e+0)
MW4	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	6.3125e-1 (1.94e-1) −	1.0000e+0 (0.00e+0)
MW5	9.9875e-1 (3.54e-3) ≈	1.0000e+0 (0.00e+0) ≈	1.5875e-1 (9.91e-3) −	6.6250e-2 (2.97e-2) −	8.5375e-1 (1.55e-1)
MW6	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	4.3250e-1 (2.60e-1) −	1.0000e+0 (0.00e+0)
MW7	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	7.3375e-1 (4.93e-2) −	9.0625e-1 (1.98e-1) ≈	1.0000e+0 (0.00e+0)
MW8	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	6.8750e-1 (4.29e-1) −	1.0000e+0 (0.00e+0)
MW9	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	5.3125e-1 (3.09e-2) −	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0)
MW10	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	1.0375e-1 (1.23e-1) −	1.0000e+0 (0.00e+0)
MW11	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	5.7875e-1 (5.00e-2) −	8.5375e-1 (1.09e-1) ≈	1.0000e+0 (0.00e+0)
MW12	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	8.1750e-1 (5.18e-2) −	3.6625e-1 (2.27e-1) −	1.0000e+0 (0.00e+0)
MW13	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	7.7375e-1 (1.85e-1) −	1.0000e+0 (0.00e+0)
MW14	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0) ≈	1.0000e+0 (0.00e+0)
+/ − / ≈	0/2/22	0/0/24	0/5/19	0/13/11	

DOC contains nine CMOPs with complex constraints in both decision and objective spaces. According to the experimental results shown in Table III, the numbers of best results obtained by NSGA-II, PPS, C-TAEA, ToP, and CCMO are 0, 5, 0, 1, and 17, respectively. Therefore, it can be concluded that the proposed CCMO has better overall performance than some state-of-the-art MOEAs for solving benchmark CMOPs.

Moreover, Table IV lists the mean value of the time consumption of the five MOEAs on each test suite, averaged over 30 runs. It can be found that NSGA-II consumes the least time and C-TAEA consumes the most time. Besides, it is worth noting that the proposed CCMO is less efficient than NSGA-II though they use the same search strategies, since CCMO performs selection strategies twice a generation and employs the truncation strategy instead of crowding distance,

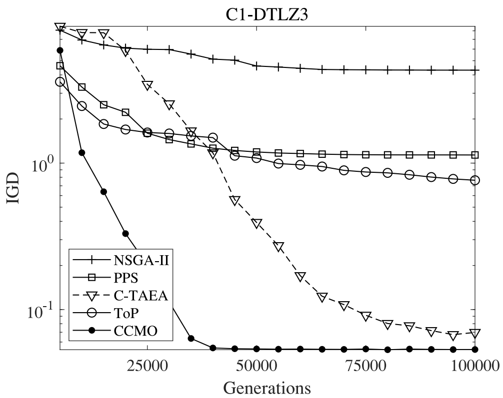

Fig. 6. Convergence profiles of IGD values obtained by NSGA-II, PPS, C-TAEA, ToP, and CCMO on C1-DTLZ3, averaged over 30 runs.

TABLE III
IGD VALUE OF NSGA-II, PPS, C-TAEA, TOP, AND CCMO ON LIR-CMOP AND DOC PROBLEMS. THE BEST RESULT IN EACH ROW IS HIGHLIGHTED. "N/A" INDICATES THAT NO FEASIBLE SOLUTION IS FOUND. "+," "−," AND "≈" INDICATE THAT THE RESULT IS SIGNIFICANTLY BETTER, SIGNIFICANTLY WORSE, AND STATISTICALLY SIMILAR TO THAT OBTAINED BY CCMO, RESPECTIVELY

Problem	NSGA-II	PPS	C-TAEA	ToP	CCMO
LIR-CMOP1	2.210e-1 −	7.305e-3 ≈	2.503e-1 −	8.460e-2 ≈	3.913e-2
LIR-CMOP2	1.383e-1 ≈	5.885e-3 +	4.582e-2 +	9.245e-2 ≈	8.478e-2
LIR-CMOP3	2.786e-1 −	6.754e-3 +	1.674e-1 ≈	3.306e-1 −	1.544e-1
LIR-CMOP4	2.422e-1 −	3.431e-3 +	1.223e-1 ≈	3.594e-1 −	1.239e-1
LIR-CMOP5	7.401e-1 −	7.292e-3 ≈	5.977e-2 ≈	2.973e-1 ≈	5.192e-3
LIR-CMOP6	3.254e-1 −	7.899e-3 ≈	1.377e-1 ≈	6.341e-3 ≈	5.223e-3
LIR-CMOP7	9.137e-3 ≈	1.058e-2 ≈	2.000e-2 −	8.768e-3 ≈	7.239e-3
LIR-CMOP8	1.230e-2 ≈	1.048e-2 ≈	1.528e-2 −	8.718e-3 ≈	7.250e-3
LIR-CMOP9	4.338e-1 −	3.227e-3 ≈	5.157e-2 ≈	3.678e-1 −	2.680e-3
LIR-CMOP10	2.825e-1 −	5.458e-3 ≈	1.099e-1 −	5.394e-3 ≈	4.617e-3
LIR-CMOP11	9.023e-2 −	2.396e-3 ≈	1.471e-1 −	1.432e-1 −	2.391e-3
LIR-CMOP12	1.109e-1 −	2.954e-3 ≈	1.245e-2 ≈	5.931e-2 ≈	2.817e-3
LIR-CMOP13	1.188e-1 −	1.247e-1 −	1.078e-1 ≈	1.274e-1 −	1.077e-1
LIR-CMOP14	1.208e-1 −	1.170e-1 −	1.110e-1 −	1.184e-1 −	9.966e-2
DOC1	1.788e+0 ≈	5.062e-2 ≈	4.290e+2 −	5.975e-3 ≈	5.751e-3
DOC2	N/A	4.859e-1 −	N/A	4.099e-1 −	6.517e-2
DOC3	6.859e+2 ≈	1.219e+2 ≈	N/A	3.064e+2 ≈	4.620e+2
DOC4	1.046e+0 ≈	2.690e-1 ≈	2.265e+2 −	4.125e-2 ≈	2.248e-2
DOC5	N/A	8.298e+1 ≈	N/A	1.247e-1 ≈	2.185e+1
DOC6	2.001e+0 ≈	4.975e-1 ≈	3.405e+1 −	1.911e+0 ≈	4.365e-3
DOC7	4.438e+0 −	4.985e-1 ≈	N/A	3.186e-1 ≈	2.524e-3
DOC8	7.634e+1 ≈	8.791e+1 −	3.657e+2 −	1.393e+1 ≈	7.460e-2
DOC9	1.694e-1 −	2.731e-1 −	7.035e-1 −	1.704e-1 −	7.505e-2
+/ − / ≈	0/13/8	3/5/15	1/11/7	0/8/15	

where the truncation strategy is much more time consuming than calculating crowding distance [50].

V. COMPARISONS ON VEHICLE ROUTING PROBLEM WITH TIME WINDOWS

This section verifies the performance of CCMO in solving the VRPTW problem, which is an extensively studied combinational optimization problem with complicated decision space and strict constraints [4]. VRPTW has been tackled by many exact, heuristic, and metaheuristic methods [51]. In recent years, a number of MOEAs have also been employed for solving VRPTW, including NSGA [52], MOPSO [53], and MOEA/D [17].

A VRPTW considers a central depot and a number of customers, where each customer has its own location, demand, and service time window. The goal of solving the problem is to minimize the total traveled distance of multiple vehicles for serving all the customers, with the satisfaction of the capacities of vehicles and the service time windows of customers. The objectives of a VRPTW can be defined as [17]

$$\begin{aligned} \min_{\mathbf{x}} \ & \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x})) \\ & f_1(\mathbf{x}) = K \\ & f_2(\mathbf{x}) = \sum_{i=0}^{N}\sum_{j=0}^{N}\sum_{k=1}^{K} d_{ij}a_{ijk} \end{aligned} \tag{3}$$

where $\mathbf{x}$ is a solution denoting the routes of multiple vehicles; N denotes the number of customers; K denotes the number of vehicles; d_{ij} denotes the distance between customers c_i and c_j (c_0 denotes the central depot); and a_{ijk} is set to 1 if arc $\langle c_i, c_j \rangle$ is traversed by the kth vehicle and 0 otherwise. In short, the problem aims to minimize both the number of vehicles f_1 and the total traveled distance f_2. The constraints of a VRPTW can be defined as

$$\sum_{i=1}^{N} a_{i0k} = \sum_{j=1}^{N} a_{0jk} = 1 \text{ for } k \in \{1, \ldots, K\} \tag{4}$$

$$\sum_{j=0, j\neq i}^{N} a_{ijk} = \sum_{j=0, j\neq i}^{N} a_{jik} \leq 1 \quad \text{for } i \in \{1, \ldots, N\} \text{ and } k \in \{1, \ldots, K\} \tag{5}$$

$$\sum_{k=1}^{K}\sum_{i=0, i\neq j}^{N} a_{ijk} = 1 \text{ for } j \in \{1, \ldots, N\} \tag{6}$$

$$\sum_{k=1}^{K}\sum_{j=0, j\neq i}^{N} a_{ijk} = 1 \text{ for } i \in \{1, \ldots, N\} \tag{7}$$

$$\sum_{i=0}^{N} q_i \sum_{j=0, j\neq i}^{N} a_{ijk} \leq Q \text{ for } k \in \{1, \ldots, K\} \tag{8}$$

$$t_j = t_i + w_i + s_i + t_{ij} \text{ for } i, j \in \{1, \ldots, N\}, \quad i \neq j \tag{9}$$

$$e_i \leq t_i + w_i \leq l_i \text{ for } i \in \{0, \ldots, N\} \tag{10}$$

where Q denotes the capacity of each vehicle; q_i denotes the demand of customer c_i; t_i denotes the time when the vehicle arrives at c_i; w_i denotes the waiting time at c_i (in case the vehicle arrives before the service time window); s_i denotes the service time required by c_i; t_{ij} denotes the traveling time between c_i and c_j; and e_i and l_i determine the service time window of c_i. In short, (4) and (5) ensure that each vehicle always starts from the depot, visits customers in sequence, and finally returns to the depot, (6) and (7) ensure that each customer is visited only once, (8) gives the constraints of capacities of vehicles, and (9) and (10) give the constraints of time windows of customers.

A. Genetic Operators for Solving VRPTW

As shown in Fig. 7, a solution for VRPTW is usually represented by a permutation of customers (i.e., $1, \ldots, N$) and the depot (i.e., 0), where the routes of different vehicles are

TABLE IV
AVERAGE TIME CONSUMPTION (IN SECOND) OF NSGA-II, PPS, C-TAEA, TOP, AND CCMO ON CONSTRAINED DTLZ, MW, LIR-CMOP, AND DOC PROBLEMS. THE BEST RESULT IN EACH ROW IS HIGHLIGHTED

Problem	NSGA-II	PPS	C-TAEA	ToP	CCMO
DTLZ problems	3.9241e+0	3.5348e+1	2.4955e+2	1.0940e+1	5.3379e+1
MW problems	4.1752e+0	4.1412e+1	2.6718e+2	2.4737e+1	4.9596e+1
LIR-CMOP problems	1.1564e+1	1.2950e+2	7.1397e+2	2.1835e+1	1.5344e+2
DOC problems	1.1703e+1	1.0128e+2	5.3677e+2	5.8446e+1	1.1980e+2

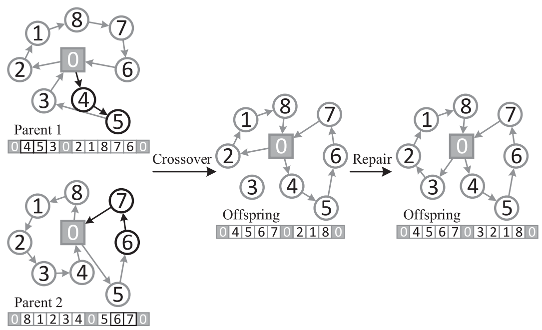

Fig. 7. Procedure of sequence-based crossover.

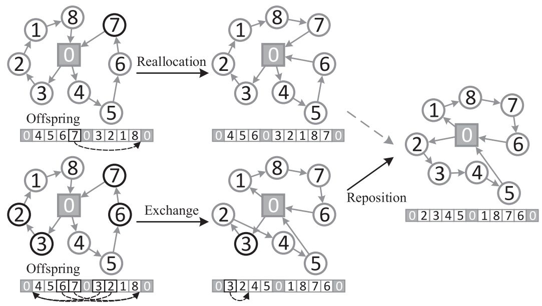

Fig. 8. Procedure of remove and reinsert mutation.

separated by the value 0. Accordingly, the sequence-based crossover [54] and the remove and reinsert mutation [55] are adopted to generate offsprings. Given two parents as shown in Fig. 7, the crossover operator randomly selects a route from the first parent (e.g., 4-5-3) and a route from the second parent (e.g., 5-6-7), and splits each route into two sequences. Then, the offspring is set to the same as the first parent, while a new route is generated by combining the first sequence of the first route and the second sequence of the second route (e.g., 4-5-6-7). If some customers are missed in the offspring (e.g., 3), the offspring will be repaired by randomly inserting the missed customers into a feasible position.

According to Fig. 8, the mutation operator selects two routes from the offspring, then performs the reallocation operator if the two routes are the same and the exchange operator otherwise. The reallocation operator selects a sequence (e.g., 5-6-7) from the route, and checks whether each customer in the sequence can be inserted into other routes for a feasible solution with the shorter distance. The exchange operator selects a sequence (e.g., 3-2 and 6-7) from each route and checks

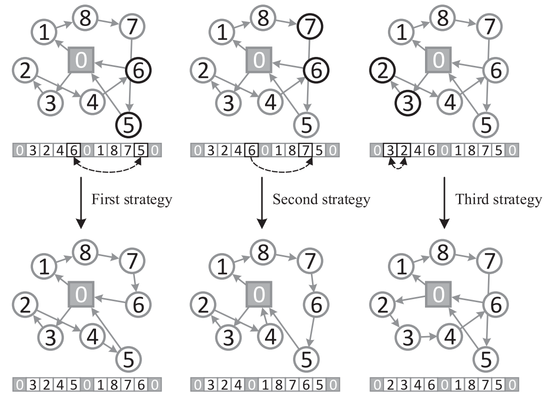

Fig. 9. Procedure of the local search strategies in CCMO.

whether each customer in each sequence can be inserted into the other route for a feasible solution with the shorter distance. Finally, the reposition operator is performed, which selects a customer (e.g., 3) and checks whether it can be inserted into other positions in the same route for a shorter distance.

B. Local Search Strategies for Solving VRPTW

To further enhance the performance of MOEAs for solving VRPTW, three local search strategies are suggested to fine-tune the population at the end of each ten generations. For each solution in the population, one of the three local search strategies is randomly selected and performed on it, and it can be replaced by the fine-tuned solution if the latter has a shorter distance.

In the proposed search strategies, the neighborhood customers of each customer c_i are defined as those having the $\lceil 0.1N \rceil$ shortest distances to c_i. As shown in Fig. 9, the first strategy randomly selects two routes, then traverses each customer c_i in the first route to check whether c_i can be swapped with each customer c_j in the second route for a feasible solution with shorter distance; the customer c_j to be swapped with c_i should be a neighborhood customer of c_i. The second strategy randomly selects two routes, then traverses each customer c_i in the first route to check whether c_i can be inserted after each customer c_j in the second route for a feasible solution with shorter distance; similarly, the customer c_j in the second route should be a neighborhood customer of c_i. The third strategy randomly selects a route, then traverses each pair of customers in the route to check whether the customers between them can be reversed for a shorter distance. To summarize, these three strategies can reduce the objective values of each solution by different greedy strategies. Specifically, the first strategy aims to enhance the diversity by exchanging the customers in different routes, the second strategy aims to reduce the number of vehicles by moving customers from one route to another, and the third strategy aims to reduce the distance by reallocating the customers in the same route.

C. Parameter Settings

The proposed CCMO is compared to NSGA-II, PPS, C-TAEA, ToP, MACS-VRPTW [56], HMOMA [57], and M-MOEA/D [17] on R101–R112, where R101–R112 are

TABLE V
OBJECTIVE VALUES OF THE NONDOMINATED SOLUTIONS OBTAINED BY NSGA-II, PPS, C-TAEA, TOP, MACS-VRPTW, HMOMA, M-MOEA/D, AND CCMO ON R101–R112. THE BEST RESULTS IN EACH ROW ARE HIGHLIGHTED

Problem	NSGA-II		PPS		C-TAEA		ToP		MACS-VRPTW		HMOMA		M-MOEA/D		CCMO		CCMO with an unconstrained helper problem	
	f_1	f_2	f_1	f_2	f_1	f_2	f_1	f_2	f_1	f_2	f_1	f_2	f_1	f_2	f_1	f_2	f_1	f_2
R101	19	1790.17	19	1825.64	19	1864.28	19	1690.87	20	1684.63	19	1690.28	19	1652.17	**19**	**1650.80**	19	1663.86
	20	1689.70	20	1703.70	20	1758.15	20	1664.13	21	1656.39	20	1664.13	20	1644.70	**20**	**1643.79**	20	1649.84
R102	17	1564.12	17	1562.19	17	1583.64	17	1513.74	18	1527.18	17	1513.74	**17**	**1486.12**	17	1492.92	17	1499.37
	18	1504.73	18	1501.83	18	1528.35	18	1487.07	19	1496.42	18	1487.07	**18**	**1473.73**	18	1477.03	18	1489.61
R103	13	1473.25	14	1392.04	14	1402.76	13	1392.73	14	1287.00			**13**	**1354.22**	**13**	**1354.22**	13	1375.26
	14	1346.57	15	1289.30	15	1286.47	14	1247.31	15	1268.52	14	1237.05	**14**	**1213.62**	**14**	**1213.62**	14	1237.39
R104	10	1092.31	10	1064.74	10	1054.39	10	1020.87	10	1047.58	10	1020.87	10	999.31	**10**	**992.38**	10	1008.24
	11	1032.47	11	1022.58	11	1027.70	11	1010.24	11	1043.73	11	1010.24	**11**	**991.91**	**11**	**991.91**	11	1005.93
R105	14	1458.64	14	1446.15	14	1437.83	14	1415.13	15	1424.62	14	1415.13	14	1410.64	**14**	**1382.50**	14	1415.13
	15	1399.58	15	1392.06	15	1387.61	15	1390.12	16	1384.33	15	1390.12	15	1366.58	**15**	**1366.18**	15	1380.85
R106	12	1308.65	12	1305.31	12	1368.59	12	1284.82					12	1265.99	**12**	**1262.05**	12	1271.86
	13	1291.22	13	1283.57	13	1289.22	13	1254.22	13	1270.28	13	1254.22	**13**	**1249.22**	13	1253.35	13	1260.99
R107	10	1179.47	10	1146.62	10	1193.75	10	1147.93					10	1139.47	**10**	**1135.40**	10	1174.29
	11	1106.22	11	1102.31	11	1112.45	11	1100.52	11	1125.59	11	1100.52	11	1086.22	**11**	**1075.71**	11	1086.52
R108	10	987.52	10	980.26	10	985.08	10	975.34	10	971.91	10	975.34	10	965.52	**10**	**952.40**	10	986.48
R109	12	1174.44	12	1197.52	12	1195.04	12	1169.85	12	1224.67	12	1169.85	12	1157.44	**12**	**1153.89**	12	1174.83
	13	1168.38	13	1187.67	13	1183.93	13	1166.09	13	1215.06	13	1166.09	13	1155.38	**13**	**1151.84**	13	1155.52
R110	11	1181.26	11	1158.73	11	1178.53	11	1112.21			11	1112.21	**11**	**1110.68**	11	1110.68	11	1146.63
	12	1166.03	12	1149.91	12	1152.70	12	1108.42	12	1150.28			12	1106.03	**12**	**1101.20**	12	1107.97
R111	11	1109.82	11	1107.86	11	1104.03	11	1084.76	11	1135.61	11	1084.76	11	1073.82	**11**	**1064.73**	11	1100.26
			12	1086.45	12	1092.27	12	1079.82			12	1079.82			**12**	**1061.33**	12	1084.60
R112	10	981.43	10	996.73	10	991.56	10	976.99	11	1027.13	10	976.99	10	981.43	**10**	**969.48**	10	979.51

12 datasets taken from Solomon's benchmark [58], each of which contains 100 customers with various location, demand, and service time window. MACS-VRPTW, HMOMA, and M-MOEA/D are three effective evolutionary algorithms tailored for VRPTW, which are based on the ant colony system, Pareto dominance-based MOEA, and decomposition-based MOEA, respectively.

Since NSGA-II, PPS, C-TAEA, ToP, and the proposed CCMO are not tailored for VRPTW, all of them employ the genetic operators described in Section V-A and the local search strategies proposed in Section V-B. Besides, MACS-VRPTW, HMOMA, and M-MOEA/D use their own recombination operators, and they adopt different local search strategies. The parameters of PPS and ToP are set to the same as those in Section IV-A. For MACS-VRPTW, the parameter settings in the ant colony system are $\alpha = 1$, $q_0 = 0.9$, $\beta = 2$, and $\rho = 0.1$. For HMOMA, the archive size is set to 200, the depth of local search is set to 10, and the ratio of computing resources of two phases is set to 1/3. For M-MOEA/D, the neighborhood list size is set to 10 and the archive size is set to 100. For CCMO, the helper problem is set to the original problem without the constraints of the time window [i.e., (9) and (10)]. Besides, the population size and the number of function evaluations are set to 50 and 100 000 for all the compared MOEAs, respectively. A parameter sensitivity analysis of all the compared MOEAs is given in the Supplementary Material I.

D. Experimental Results on VRPTW

Table V lists the objective values of the results of the eight MOEAs on R101–R112, where each result includes the nondominated solutions among all the solutions obtained in 30 independent runs. It is clear that most solutions found by CCMO have the same number of vehicles but shorter traveled distance than those found by the other MOEAs. For NSGA-II, PPS, C-TAEA, and ToP, although they use the same genetic operators and local search strategies to CCMO, they exhibit bad performance since their search strategies cannot balance between constraints and objectives on VRPTW that has a highly discrete landscape and a small feasible region. For MACS-VRPTW and HMOMA, they are also worse than M-MOEA/D and CCMO since they always prefer feasible solutions and are easily trapped into local optima. For M-MOEA/D with a novel selection strategy, it obtains some better results than CCMO since the selection strategy is tailored for VRPTW that makes a balance between constraints and objectives. While for the proposed CCMO, it solves VRPTW with the assistance of solving the same problem without the constraints of the time window, having achieved the best solutions on 11 out of the 12 datasets.

To verify the rationality of the helper problem in CCMO, the last column of Table V also lists the objective values obtained by a variant of CCMO, which sets the helper problem as the original VRPTW without any constraint (i.e., a traveling salesman problem). It is obvious that CCMO obtains better solutions than the variant with an unconstrained problem. In fact, according to the table, the performance of CCMO becomes worse than M-MOEA/D on most datasets if an unconstrained problem is used as the helper problem. Therefore, it is confirmed that a simple problem without the constraints of time window is more promising to assist in solving VRPTW than a helper problem without all constraints. Besides, the proposed CCMO is compared to MACS-VRPTW, HMOMA, and M-MOEA/D with the same local search strategies in the Supplementary Material II. The experimental results show that CCMO still outperforms the three algorithms, which demonstrates that the better performance of CCMO is mainly due to the proposed framework but not the local search strategies.

VI. Conclusion and Future Work

This article has proposed a coevolutionary framework for solving CMOPs, which aims to solve a difficult CMOP with the assistance of solving a helper problem. To this end, the proposed framework evolves one population to solve the original CMOP and evolves another population to solve a helper problem derived from the original one. In contrast to existing coevolutionary MOEAs, the proposed framework holds a weak cooperation between populations to achieve the assistance in solving the original CMOP, rather than a strong cooperation to evolve multiple populations cooperatively. As verified in Section III-B, the novel paradigm of CCMO is promising for solving CMOPs.

In the experiments, the proposed framework has been compared to several state-of-the-art MOEAs tailored for CMOPs. According to the results in Sections IV-B and IV-C, the proposed framework has a better overall performance than the compared MOEAs on 47 benchmark CMOPs. Moreover, the proposed framework has been tested on the VRPTW problem, which has also shown better performance than existing MOEAs.

Since the proposed framework is light and flexible, it is highly desirable to extend it for better performance and wider applicability. First, more effective optimizers need to be developed and equipped in the proposed framework to better evolve the two populations. Second, the performance of the proposed framework on the VRPTW problem can be further enhanced by adopting some tailored strategies (e.g., the random key-based representation [59]). Third, it is necessary to acquire some useful experience in designing the helper problem for a given real-world CMOP.

References

[1] N. Jozefowiez, F. Semet, and E. G. Talbi, "Multi-objective vehicle routing problems," *Eur. J. Oper. Res.*, vol. 189, no. 2, pp. 293–309, 2008.

[2] R. Saravanan, S. Ramabalan, N. G. R. Ebenezer, and C. Dharmaraja, "Evolutionary multi criteria design optimization of robot grippers," *Appl. Soft Comput.*, vol. 9, no. 1, pp. 159–172, 2009.

[3] H. Mala-Jetmarova, N. Sultanova, and D. A. Savic, "Lost in optimisation of water distribution systems? A literature review of system operation," *Environ. Model. Softw.*, vol. 93, pp. 209–254, Jul. 2017.

[4] L. H. Lee, K. C. Tan, K. Ou, and Y. H. Chew, "Vehicle capacity planning system: A case study on vehicle routing problem with time windows," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 33, no. 2, pp. 169–178, Mar. 2003.

[5] A. Dixit, A. Mishra, and A. Shukla, "Vehicle routing problem with time windows using meta-heuristic algorithms: A survey," in *Harmony Search and Nature Inspired Optimization Algorithms* (Advances in Intelligent Systems and Computing), vol. 741. Singapore: Springer, 2019, pp. 539–546.

[6] 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.

[7] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 32–49, 2011.

[8] K. Li, R. Wang, T. Zhang, and H. Ishibuchi, "Evolutionary many-objective optimization: A comparative study of the state-of-the-art," *IEEE Access*, vol. 6, pp. 26194–26214, 2018.

[9] Y. Tian, X. Zheng, X. Zhang, and Y. Jin, "Efficient large-scale multi-objective optimization based on a competitive swarm optimizer," *IEEE Trans. Cybern.*, early access, Apr. 3, 2019, doi: 10.1109/TCYB.2019.2906383.

[10] 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.

[11] Z. Fan *et al.*, "Difficulty adjustable and scalable constrained multi-objective test problem toolkit," *Evol. Comput.*, early access, May 23, 2019, doi: 10.1162/evco_a_00259.

[12] Y. Wang, J. Li, X. Xue, and B. Wang, "Utilizing the correlation between constraints and objective function for constrained evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 29–43, Feb. 2020.

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

[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] Z. Fan, W. Li, X. Cai, L. Hui, and E. D. Goodman, "Push and pull search for solving constrained multi-objective optimization problems," *Swarm Evol. Comput.*, vol. 44, pp. 665–679, Feb. 2019.

[16] 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.*, 2017, pp. 209–216.

[17] Y. Qi, Z. Hou, H. Li, J. Huang, and X. Li, "A decomposition based memetic algorithm for multi-objective vehicle routing problem with time windows," *Comput. Oper. Res.*, vol. 62, pp. 61–77, Oct. 2015.

[18] L. M. Antonio and C. A. Coello Coello, "Coevolutionary multiobjective evolutionary algorithms: Survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 22, no. 6, pp. 851–865, Dec. 2018.

[19] T. Ray, K. Tai, and K. C. Seow, "Multiobjective design optimization by an evolutionary algorithm," *Eng. Optim.*, vol. 33, no. 4, pp. 399–424, 2001.

[20] 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.

[21] W. Ning, B. Guo, Y. Yan, X. Wu, J. Wu, and D. Zhao, "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. 1645–1664, 2017.

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

[23] S. Kukkonen and J. Lampinen, "GDE3: The third evolution step of generalized differential evolution," in *Proc. IEEE Congr. Evol. Comput.*, vol. 1, 2005, pp. 443–450.

[24] Q. Chen, J. Ding, S. Yang, and T. Chai, "A novel evolutionary algorithm for dynamic constrained multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, early access, Dec. 6, 2019, doi: 10.1109/TEVC.2019.2958075.

[25] 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.

[26] M. Asafuddoula, T. Ray, and R. Sarker, "A decomposition based evolutionary algorithm for many objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 3, pp. 445–460, Jun. 2015.

[27] Z. Fan, W. Li, X. Cai, K. Hu, H. Lin, and H. Li, "Angle-based constrained dominance principle in MOEA/D for constrained multi-objective optimization problems," in *Proc. IEEE Congr. Evol. Comput.*, 2016, pp. 460–467.

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

[29] B. Wang, H. Li, Q. Zhang, and Y. Wang, "Decomposition-based multiobjective optimization for constrained evolutionary optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, Nov. 15, 2018, doi: 10.1109/TSMC.2018.2876335.

[30] Z. Yang, K. Tang, and X. Yao, "Large scale evolutionary optimization using cooperative coevolution," *Inf. Sci.*, vol. 178, no. 15, pp. 2985–2999, 2008.

[31] C. K. Goh and K. C. Tan, “A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization,” *IEEE Trans. Evol. Comput.*, vol. 13, no. 1, pp. 103–127, Feb. 2009.
[32] H. Wang, L. Jiao, and X. Yao, “Two_Arch2: An improved two-archive algorithm for many-objective optimization,” *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 524–541, Aug. 2015.
[33] C. A. Coello Coello, “Use of a self-adaptive penalty approach for engineering optimization problems,” *Comput. Ind.*, vol. 41, no. 2, pp. 113–127, 2000.
[34] F. Z. Huang, L. Wang, and Q. He, “An effective co-evolutionary differential evolution for constrained optimization,” *Appl. Math. Comput.*, vol. 186, no. 1, pp. 340–356, 2007.
[35] B. Liu, H. Ma, X. Zhang, and Y. Zhou, “A memetic co-evolutionary differential evolution algorithm for constrained optimization,” in *Proc. IEEE Congr. Evol. Comput.*, 2007, pp. 2996–3002.
[36] E. Kieffer, G. Danoy, P. Bouvry, and A. Nagih, “A new co-evolutionary algorithm based on constraint decomposition,” in *Proc. IEEE Int. Parallel Distrib. Process. Symp. Workshops*, 2017, pp. 492–500.
[37] 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.
[38] J. Wang, G. Liang, and J. Zhang, “Cooperative differential evolution framework for constrained multiobjective optimization,” *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2060–2072, Jun. 2019.
[39] M. Li, S. Yang, and X. Liu, “Pareto or non-Pareto: Bi-criterion evolution in multi-objective optimization,” *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 645–665, Oct. 2016.
[40] K. Deb and R. B. Agrawal, “Simulated binary crossover for continuous search space,” *Complex Syst.*, vol. 9, no. 4, pp. 115–148, 1995.
[41] K. Deb and M. Goyal, “A combined genetic adaptive search (GeneAS) for engineering design,” *Comput. Sci. Informat.*, vol. 26, no. 4, pp. 30–45, 1996.
[42] A. Gupta, Y. S. Ong, and L. Feng, “Multifactorial evolution: Toward evolutionary multitasking,” *IEEE Trans. Evol. Comput.*, vol. 20, no. 3, pp. 343–357, Jun. 2016.
[43] J. Zhong, L. Feng, W. Cai, and Y. S. Ong, “Multifactorial genetic programming for symbolic regression problems,” *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, Jul. 25, 2018, doi: 10.1109/TSMC.2018.2853719.
[44] 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.
[45] 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.
[46] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization,” in *Proc. 5th Conf. Evol. Methods Design Optim. Control Appl. Ind. Prob.*, 2001, pp. 95–100.
[47] H. Li and Q. Zhang, “Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II,” *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, Apr. 2009.
[48] Y. Tian, X. Xiang, X. Zhang, R. Cheng, and Y. Jin, “Sampling reference points on the Pareto fronts of benchmark multi-objective optimization problems,” in *Proc. IEEE Congr. Evol. Comput.*, 2018, pp. 1–6.
[49] J. Derrac, S. Garcia, D. Molina, and F. Herrera, “A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms,” *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, 2011.
[50] Y. Tian, C. He, R. Cheng, and X. Zhang, “A multi-stage evolutionary algorithm for better diversity preservation in multi-objective optimization,” *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, Dec. 20, 2019, doi: 10.1109/TSMC.2019.2956288.
[51] N. A. EI-Sherbeny, “Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods,” *J. King Saud Univ. Sci.*, vol. 22, no. 3, pp. 123–131, 2010.
[52] M. Rahoual, B. Kitoun, M. H. Mabed, V. Bachelet, and F. Benameur, “Multicriteria genetic algorithms for the vehicle routing problem with time windows,” in *Proc. 4th Metaheuristics Int. Conf.*, 2001, pp. 527–532.
[53] Y. J. Gong, J. Zhang, and O. Liu, “Optimizing the vehicle routing problem with time windows: A discrete particle swarm optimization approach,” *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 2, pp. 254–267, Mar. 2012.
[54] J. Y. Potvin and S. Bengio, “The vehicle routing problem with time windows part II: Genetic search,” *INFORMS J. Comput.*, vol. 8, no. 2, pp. 165–172, 1996.
[55] A. Garcia-Najera and J. A. Bullinaria, “An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows,” *Comput. Oper. Res.*, vol. 38, no. 1, pp. 287–300, 2011.
[56] L. M. Gambardella, E. Taillard, and G. Agazzi, “MACS-VRPTW: A multiple colony system for vehicle routing problems with time windows,” in *New Ideas in Optimization*. London, U.K.: McGraw-Hill, 1999, pp. 63–76.
[57] J. Wang, W. Ren, Z. Zhang, H. Huang, and Y. Zhou, “A hybrid multiobjective memetic algorithm for multiobjective periodic vehicle routing problem with time windows,” *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, Aug. 17, 2018, doi: 10.1109/TSMC.2018.2861879.
[58] M. M. Solomon, “Algorithms for the vehicle routing and scheduling problems with time window constraints,” *Oper. Res.*, vol. 35, no. 2, pp. 254–265, 1987.
[59] E. Ruiz, V. Soto-Mendoza, A. E. R. Barbosa, and R. Reyes, “Solving the open vehicle routing problem with capacity and distance constraints with a biased random key genetic algorithm,” *Comput. Ind. Eng.*, vol. 133, pp. 207–219, Jul. 2019.

Ye Tian (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees from Anhui University, Hefei, China, in 2012, 2015, and 2018, respectively.

He is currently an Associate Professor with the Institutes of Physical Science and Information Technology, Anhui University, and also a Postdoctoral Research Fellow with the Department of Computer Science, City University of Hong Kong, Hong Kong. His current research interests include multiobjective optimization methods and their applications.

Dr. Tian is the recipient of the 2018 IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Award and the 2020 *IEEE Computational Intelligence Magazine* Outstanding Paper Award.

Tao Zhang received the B.Eng. degree from Chuzhou University, Chuzhou, China, in 2017. He is currently pursuing the M.Sc. degree with the School of Computer Science and Technology, Anhui University, Hefei, China.

His current research interests include evolutionary algorithms and their applications in logistics.

Jianhua Xiao received the Ph.D. degree in system engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2008.

He is currently an Associate Professor with the Research Center of Logistics, Nankai University, Tianjin, China. His research interests include combinatorial optimization, bioinspired computation, and logistics optimization.

Xingyi Zhang (Senior Member, IEEE) received the B.Sc. degree from Fuyang Normal College, Fuyang, China, in 2003, and the M.Sc. and Ph.D. degrees from the Huazhong University of Science and Technology, Wuhan, China, in 2006 and 2009, respectively.

He is currently a Professor with the School of Computer Science and Technology, Anhui University, Hefei, China. His current research interests include unconventional models and algorithms of computation, multiobjective optimization, and membrane computing.

Prof. Zhang is the recipient of the 2018 IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Award and the 2020 *IEEE Computational Intelligence Magazine* Outstanding Paper Award.

Yaochu Jin (Fellow, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 1988, 1991, and 1996, respectively, and the Dr.-Ing. degree from Ruhr University Bochum, Bochum, Germany, in 2001.

He is currently a Distinguished Chair Professor of computational intelligence with the Department of Computer Science, University of Surrey, Guildford, U.K., where he heads the Nature Inspired Computing and Engineering Group. He was a Finland Distinguished Professor and a Changjiang Distinguished Visiting Professor. He has (co)authored over 300 peer-reviewed journal and conference papers and been granted eight patents on evolutionary optimization.

Dr. Jin was the recipient of the 2014 and 2016 *IEEE Computational Intelligence Magazine* Outstanding Paper Award, the 2018 IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Award, and the Best Paper Award of the 2010 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology. He has been named a Highly Cited Researcher for 2019 by the Web of Science group. He is the Editor-in-Chief of the IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS and the Co-Editor-in-Chief of *Complex & Intelligent Systems*. He is also an Associate Editor or an Editorial Board Member of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the IEEE TRANSACTIONS ON CYBERNETICS, *Evolutionary Computation*, and *Soft Computing*. He was an IEEE Distinguished Lecturer from 2017 to 2019.