



# A novel three-stage multi-population evolutionary algorithm for constrained multi-objective optimization problems

Chenli Shi<sup>1,2</sup> · Ziqi Wang<sup>1,2</sup> · Xiaohang Jin<sup>3,4</sup> · Zhengguo Xu<sup>1,2</sup> · Zhangsheng Wang<sup>5</sup> · Peng Shen<sup>6</sup>

Received: 3 November 2022 / Accepted: 7 July 2023 / Published online: 31 July 2023  
© The Author(s) 2023

## Abstract

Lots of real-world optimization problems are inherently constrained multi-objective optimization problems (CMOPs), but the existing constrained multi-objective optimization evolutionary algorithms (CMOEA) often fail to balance convergence and diversity effectively. Therefore, a novel constrained multi-objective optimization evolutionary algorithm based on three-stage multi-population coevolution (CMOEA-TMC) for complex CMOPs is proposed. CMOEA-TMC contains two populations, called *mainPop* and *helpPop*, which evolve with and without consideration of constraints, respectively. The proposed algorithm divides the search process into three stages. In the first stage, fast convergence is achieved by transforming the original multi-objective problems into multiple single-objective problems. Coarse-grained parallel evolution of subpopulations in *mainPop* and guidance information provided by *helpPop* can facilitate *mainPop* to quickly approach the Pareto front. In the second stage, the objective function of *mainPop* changes to the original problem. Coevolution of *mainPop* and *helpPop* by sharing offsprings can produce solutions with better diversity. In the third stage, the mining of the global optimal solutions is performed, discarding *helpPop* to save computational resources. For CMOEA-TMC, the combination of parallel evolution, coevolution, and staging strategy makes it easier for *mainPop* to converge and maintain good diversity. Experimental results on 33 benchmark CMOPs and a real-world boiler combustion optimization case show that CMOEA-TMC is more competitive than the other five advanced CMOEAs.

**Keywords** Constrained multi-objective optimization problems (CMOPs) · Evolutionary algorithms · Coevolution · Parallel algorithm · Staging strategy

## Introduction

In real-world applications, many optimization problems contain several conflicting objectives and multiple complicated constraints, such as the robot gripper optimization problem [26], the optimal scheduling in microgrids [15], the green coal production problem [5], and the structure optimization of a blended-wing-body underwater glider [22], etc. They

✉ Zhengguo Xu  
xzg@zju.edu.cn

Chenli Shi  
scl99@zju.edu.cn

Ziqi Wang  
wangziqincepu@163.com

Xiaohang Jin  
xhjin@zjut.edu.cn

Zhangsheng Wang  
wangzhangsheng@tongji.edu.cn

Peng Shen  
shensimpeng@163.com

<sup>1</sup> State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, Zhejiang, China

<sup>2</sup> State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, Zhejiang, China

<sup>3</sup> College of Mechanical Engineering, Zhejiang University of Technology, Hangzhou 310023, Zhejiang, China

<sup>4</sup> Key Laboratory of Special Purpose Equipment and Advanced Processing Technology, Ministry of Education and Zhejiang Province, Zhejiang University of Technology, Hangzhou 310023, Zhejiang, China

<sup>5</sup> School of Mechanical Engineering, Tongji University, Shanghai 201804, China

<sup>6</sup> Jurong Power Generation Branch, Huadian Jiangsu Energy Co., Ltd., Zhenjiang 212000, Jiangsu, China

can be defined as constrained multi-objective optimization problems (CMOPs) and expressed as follows:

$$\begin{cases} \min & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x}))^T \in \mathbb{F} \\ \text{s.t.} & \mathbf{x} \in \mathbb{S} \\ & g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, p \\ & h_i(\mathbf{x}) = 0, \quad i = p + 1, \dots, q, \end{cases} \quad (1)$$

where  $\mathbf{x} = (x_1, \dots, x_D)^T$  is a  $D$ -dimensional decision vector in the decision space  $\mathbb{S}$ ,  $\mathbf{F}(\mathbf{x})$  represents  $M$  objective functions that need to be optimized simultaneously,  $\mathbb{F}$  is the objective space,  $p$  is the number of inequality constraints,  $g_i(\mathbf{x})$  represents the  $i$ th inequality constraints,  $(q - p)$  is the number of equality constraints, and  $h_i(\mathbf{x})$  represents the  $(i - p)$ th equality constraints. The degree of constraint violation of  $\mathbf{x}$  at the  $i$ th constraint is calculated as

$$CV_i(\mathbf{x}) = \begin{cases} \max(0, g_i(\mathbf{x})), & \text{if } i \leq p \\ \max(0, |h_i(\mathbf{x}) - \eta|), & \text{otherwise} \end{cases}, \quad i = 1, 2, \dots, q, \quad (2)$$

where  $\eta$  is a sufficiently small positive value (e.g.,  $\eta = 10^{-4}$ ) for relaxing the equality constraints to inequality constraints. The overall constraint violation value of  $\mathbf{x}$  is calculated as

$$CV(\mathbf{x}) = \sum_{i=1}^q CV_i(\mathbf{x}), \quad (3)$$

then  $\mathbf{x}$  is feasible if  $CV(\mathbf{x}) = 0$ ; otherwise, it is infeasible. Suppose two feasible decision vectors  $\mathbf{x}_u$  and  $\mathbf{x}_v$ , if any  $f_i(\mathbf{x}_u) \leq f_i(\mathbf{x}_v)$ , and there is at least one dimension  $f_i(\mathbf{x}_u) < f_i(\mathbf{x}_v)$ , then  $\mathbf{x}_u$  is said to Pareto-dominate  $\mathbf{x}_v$ , denoted as  $\mathbf{x}_u < \mathbf{x}_v$ .

A feasible solution is called Pareto-optimal when no other feasible solution dominates it. The set of all Pareto-optimal solutions in the search space is the Pareto set (PS). The image of all Pareto-optimal solutions in the objective space is the Pareto front (PF).

Compared with unconstrained MOPs, CMOPs are more challenging and have received more attention in recent years. The critical to solve CMOPs is to effectively balance feasibility, convergence, and diversity [19]. As a result, a number of novel algorithms have recently emerged. A common feature of such algorithms is the transformation of the original CMOPs to make better use of the potential information in the search space and to avoid the difficulties encountered in solving the original CMOPs directly [19]. Some scientists are motivated to solve CMOPs using multi-stage evolution methods. They tend to promote different performance at different stages of the algorithm, generally achieving fast convergence before considering feasibility and diversity, proposing algorithms, such as ToP [20], PPS [13], and CMOEA-MS [31]. Some scientists try to preserve infeasible solutions in the

search process and extract useful genetic information from the infeasible ones to generate more diversified solutions. Along this line of research, many CMOEAs with multi-population cooperation methods have been proposed; for example, C-TAEA [18], CCMO [30], and Bico [21].

However, these algorithms also have certain drawbacks. It is difficult to design an effective transformation technique to guarantee the equivalence with the original problem. [14, 19] If the problem transformation mechanism is not reasonable, the algorithm performance will degrade dramatically due to the deviation of the transformed problem from the original one. The specific indication in multi-stage algorithms is that it may cause populations stuck in local optimums, especially when feasible regions are narrow or poorly distributed in the search space. In the case of multi-population algorithms, it is that too much attention to the transformed problem may lead to the waste of computational resources and poor feasibility when the PF of the transformed problem differs significantly from that of the original problem.

In this paper, to fully and efficiently explore the solution space and to better balance convergence and diversity, we combine multi-stage strategy and multi-population strategy, and propose a novel constrained multi-objective optimization evolutionary algorithm based on three-stage multi-population coevolution (CMOEA-TMC). The main contributions of CMOEA-TMC are listed as follows:

1. To address the problem that the single population is prone to fall into the local optimum, we propose a multi-population method that includes both parallel evolution and coevolution. (1) The main population, called *mainPop*, is divided into several subpopulations that evolve using the coarse-grained parallel algorithm to speed up convergence and perform global searches and (2) the helper population, called *helpPop*, is to better maintain diversity by providing information of infeasible solutions with *mainPop* during offspring generation and cross-evolution.
2. A novel three-stage strategy is designed to improve search efficiency and ensure solution feasibility. In the first stage, it focuses on fast convergence and global search by dividing subpopulations. In the second stage, it focuses on the independent evolution of subpopulations, which can maintain the diversity of solution sets. In the third stage, extreme value mining is performed to find global optimal solutions.
3. A novel algorithm named CMOEA-TMC is proposed. The combination of multi-population and multi-stage strategies allows the transformed problem to be as consistent as possible with the original ones and results in better overall performance of the algorithm. Systematic experiments on three sets of benchmark functions

and a real-world case demonstrate the effectiveness of CMOEA-TMC.

The rest of this article is organized as follows. The section “Related work” overviews the state-of-the-art evolutionary approaches developed for CMOPs. The section “The proposed algorithms: CMOEA-TMC” describes the details of CMOEA-TMC. Afterward, in the section “Experimental research”, we compare CMOEA-TMC with five advanced CMOEAs on 33 benchmark problems and a real-world boiler combustion optimization problem, showing the competitiveness of CMOEA-TMC. Finally, the section “Conclusion” concludes this article.

## Related work

In the early days, scientists first focus on multi-objective optimization problems (MOPs) without constraints. The optimization algorithms for MOPs can be classified into two categories: traditional algorithms and intelligent algorithms [10]. Traditional algorithms include the weighting method, the constraint method, and the linear programming method [10]. For example, the weighting method refers to assigning appropriate weighting factors to the different objectives in a multi-objective problem according to their importance, and then summed up as a new objective function. Traditional algorithms always transform a multi-objective problem into a single-objective problem, in which only one solution can be obtained in each optimization, resulting in poor diversity [3]. Intelligent algorithms mainly refer to evolutionary algorithms (EAs), which iterate in the unit of population and can handle large, complex search spaces. These methods produce a set of solutions in a single run and are better suited to multi-objective optimization needs [4]. Considering the high robustness and broad applicability of EAs, it has been recognized as an efficient approach to solving MOPs [32]. Over the past 2 decades, many EAs, such as genetic algorithm (GA) [6], differential evolution (DE) [27], and particle swarm optimization (PSO), [11] have been applied, and multi-objective evolutionary algorithms (MOEAs) have performed well in solving various MOPs. Many scientists have studied MOEAs and made some improvements, such as domination-based EA [9, 36], decomposition-based EA [16, 37], indicator-based EA [35, 39], etc.

Over the last 2 decades, many constraint handling techniques (CHTs) have been proposed for constrained problems. They are constrained dominance principle (CDP) [9], penalty function methods [34], and  $\varepsilon$ -constrained methods [28]. Of these, CDP is the most widely used one because of its efficiency and simplicity. These CHTs are usually utilized to solve single-objective optimization problems and can be used to solve CMOPs by extension or modification. For

example, Deb et al. [9] proposed the famous NSGA-II-CDP, which incorporates CDP into NSGA-II for environment selection. Fan et al. [12] proposed an angle-based CDP algorithm (MOEA/D-ACDP), which takes advantage of classical Pareto to compare infeasible solutions within a given angle. However, these methods still have some limitations, because they use relatively simple constraint information. They also have difficulty in achieving good results when dealing with types of problems, such as low feasible domains, strong constraints, and discontinuous PF. Therefore, there is a need to propose improved algorithms that are more applicable to CMOPs.

In recent years, to solve CMOPs more efficiently, many researchers try to transform CMOPs into other problems, such as multi-stage optimization problems or collaborative optimization problems. This can assist the population to discover some potential information during the evolutionary process and to converge better.

Multi-stage optimization generally involves dividing the optimization process into multiple stages, with different CHTs or different objects being optimized in different stages. Back in 2013, Miyakawa et al. [25] proposed a new CMOEA based on a two-stage non-dominated solution ranking. In the first stage, the degree of constraint violation for each constraint is considered as an objective function, and the whole population is classified into several fronts by non-dominated ranking based on the constraint violation values. In the second stage, each obtained front is reclassified by an undominated ranking based on the objective function value, and the parent population is selected from it. Miyakawa et al. claimed that feasible solutions with better objective functions can be found using this two-stage non-dominated ranking. Liu and Wang [20] proposed a two-stage framework called ToP for dealing with CMOPs with complex constraints. ToP is divided into two stages, the uniqueness of which is its first stage, where the CMOP is transformed into a constrained single-objective optimization problem using a weighted summation method. CDP is used to deal with the constraints, aiming at finding promising feasible regions and fast convergence, while the second stage is solved using common CMOEAs, such as NSGA-II and CMOEA/D. Fan et al. [13] embed the push–pull search (PPS) framework into CMOEA/D. PPS is also divided into two stages: in the push stage, only the objective function is considered for the search to approximate the unconstrained PF; in the pull stage, an improved CMOEA is used to pull the infeasible individuals obtained in the push stage to the feasible non-dominated regions. Tian et al. [31] devised a two-stage evolutionary algorithm, called CMOEA-MS, in which one stage helps the population to reach the feasible regions and the other stage allows the population to spread along the feasible boundaries. Furthermore, according to the status of the population, the algorithm can adaptively switch between the two stages.

Coevolution refers to the evolutionary technology in which multiple objects carry out the collaborative search through certain mechanisms and strategies, such as multi-population, multi-algorithm, and multi-strategy integrated evolution. When applied to CMOPs, multi-population and multi-constraint processing is mostly used. Li et al. [18] proposed a two-archive evolutionary algorithm (C-TAEA). It maintains two archives simultaneously: (1) a convergence-oriented archive (CA), which optimizes constraints and objectives, and (2) a diversity-oriented archive (DA), which optimizes only objectives. The two populations cooperate with each other in the mating selection and environmental selection. CCMO [30] is a coevolutionary algorithm that simultaneously computes two objective problems, the original problem and the helper problem. Typically, the original problem in CCMO is solved by population 1. The helper problem is defined as the original problem with constraints removed and is solved by population 2. Coevolution is achieved by sharing offspring between the two populations. Liu et al. [21] designed a novel bidirectional coevolutionary algorithm using both the main population and archive population. Specifically, the main population maintains the feasibility of the solution set and moves from the feasible side to the PF, while the archive population uses the angle information to maintain the diversity of the solution set and approach the PF from the infeasible side.

In view of the effectiveness of multi-stage algorithms and coevolutionary algorithms in dealing with CMOPs, some optimization algorithms that combine the two strategies have been produced recently. Fan et al. [14] improved CCMO by proposing a two-stage coevolutionary constrained multi-objective optimization evolutionary algorithm (TSC-CMOEA), which switches to the second stage when the rate of population change is too small during coevolution, discarding the helper problem and keeping only the main population evolving, thus saving computational resources and enhancing the convergence of the population. Ming et al. [24] proposed a new method with dual stages and dual populations, called DD-CMOEA. Specifically, the dual populations are *mainPop* and *auxPop*, which evolve with and without considering constraints, respectively. The dual stages are exploration and exploitation, focusing on extensive search for solutions with good objective values in the exploration stage and convergence to the true PF in the exploitation stage.

## The proposed algorithms: CMOEA-TMC

### Main framework of CMOEA-TMC

To overcome the current CMOEAs' difficulty in effectively balancing feasibility, convergence, and diversity, this paper

proposed a novel algorithm—CMOEA-TMC. As outlined in Fig. 1, the proposed CMOEA-TMC has three stages: fast convergence stage, diversity maintenance stage, and extremum mining stage. It is worth noting that the proposed algorithm contains two archives: the main population (called *mainPop*) and the help population (called *helpPop*). *mainPop* is used to search for the constrained PF throughout the process, and *helpPop* searches for the unconstrained PF in the first and second stages.

---

### Algorithm 1 Procedure of CMOEA – TMC

---

**Input:** a CMOP, the number of optimization goals  $M$ , the subpopulation size  $N$ , the current number of function evaluations  $FE$ , and the maximum number of function evaluations  $maxFE$

**Output:** the final *mainPop*

```

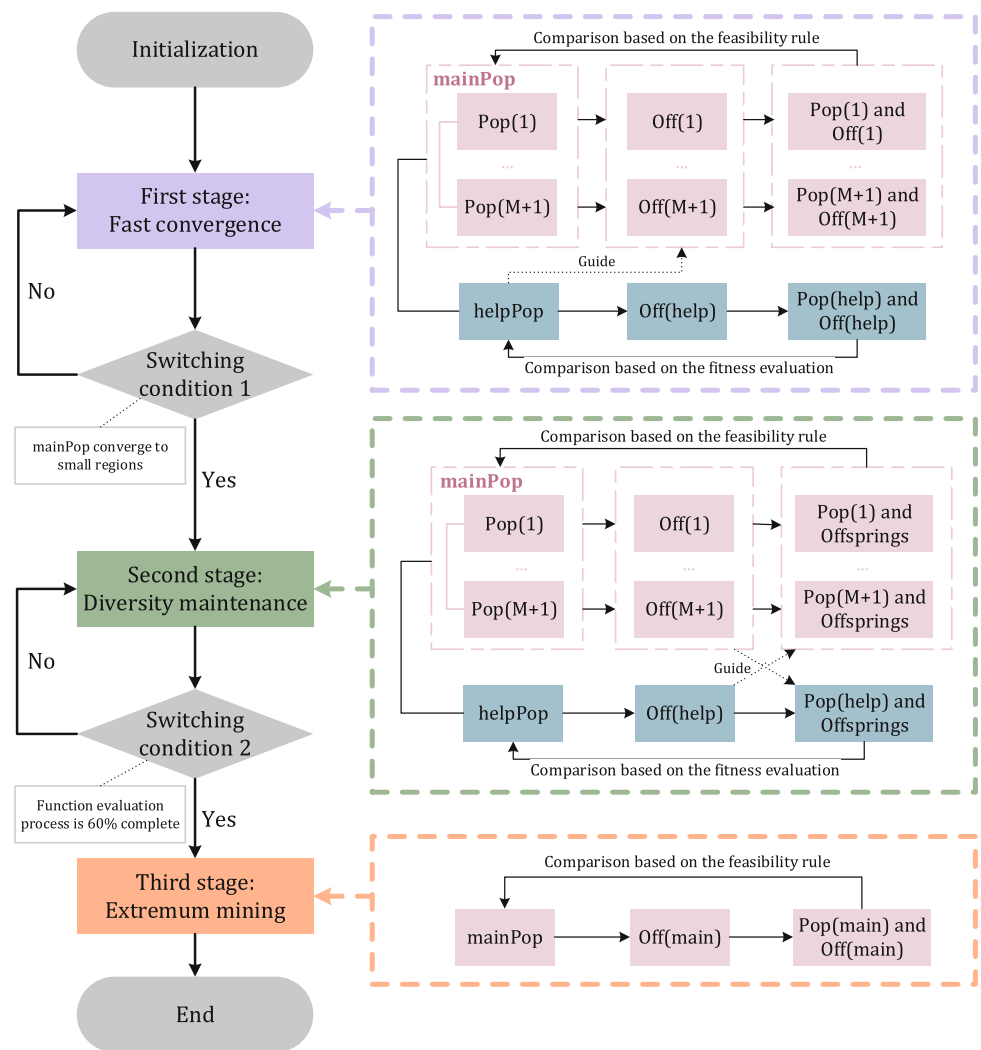
1:  $Pop(k) \leftarrow RandomInitialization(N)$ 
2:  $mainPop \leftarrow Pop(1) \cup \dots \cup Pop(M) \cup Pop(M + 1)$ 
3:  $helpPop \leftarrow RandomInitialization(N)$ 
4: while  $FE < maxFE$  do
5:   if CMOEA – TMC is in the first stage then
6:     Problem  $\leftarrow$  Transform a CMOP into  $(M+1)$  single-objective
       optimization problems according to Eq. (4)
7:     for  $k = 1$  to  $M + 1$  do
8:       Produce  $Off(k)$  from  $Pop(k)$  and  $helpPop$  using DE
       operators
9:       Update  $Pop(k)$  using the constrained dominance principle
10:    end for
11:     $mainPop \leftarrow Pop(1) \cup \dots \cup Pop(M) \cup Pop(M + 1)$ 
12:    Produce  $Off(help)$  from  $helpPop$  using GA operators
13:    Update  $helpPop$  using the fitness evaluation strategy
14:  else if CMOEA – TMC is in the second stage then
15:    Problem  $\leftarrow$  Use the original CMOP
16:    for  $k = 1$  to  $M + 1$  do
17:      Produce  $Off(k)$  from  $mainPop$  and  $helpPop$  using GA
       operators
18:      Update  $Pop(k)$  by discovering the non-dominated feasible
       solutions
19:    end for
20:     $mainPop \leftarrow Pop(1) \cup \dots \cup Pop(M) \cup Pop(M + 1)$ 
21:    Produce  $Off(help)$  from  $helpPop$  and  $mainPop$  using GA
       operators
22:    Update  $helpPop$  using the fitness evaluation strategy
23:  else
24:    Produce  $Off(main)$  from  $mainPop$  using GA operators
25:    Update  $mainPop$  by discovering the non-dominated feasible
       solutions
26:  end if
27: end while
28: return  $mainPop$ 

```

---

Algorithm 1 gives the pseudocode of CMOEA-TMC. In the first stage, the algorithm transforms the CMOP into multiple single-objective problems by setting different weight vectors, so as to achieve task partitioning of the complex problem. The detailed transformation process is described in the section “The first stage: fast convergence”. *mainPop* is also divided into multiple subpopulations and evolves using the coarse-grained parallel algorithm which has strong global search capability and achieves fast convergence. In addition,

**Fig. 1** The framework of CMOEA-TMC



the guidance information provided by *helpPop* to *mainPop* ensures that the population can maintain good diversity. In the second stage, *mainPop* changes to use the original CMOP as the objective function, which is beneficial to obtaining more high-quality feasible solutions. Meanwhile, the subpopulations in *mainPop* maintain independent evolution, so that all candidate solutions obtained in the first stage continue to converge to more promising feasible domains, maintaining the diversity of the solution set. In the third stage, the individuals of all subpopulations in *mainPop* are aggregated together to optimize for the global optimal solution. And, *helpPop* no longer provides additional information, as we consider it more important to focus on the constrained PF in this stage. These three stages are further explained in the sections “The first stage: fast convergence”–“The third stage: extremum mining”.

It is worth emphasizing that to compare the performance of different solutions, CDP is applied to the environmental selection. The definition of CDP is as follows. Given two

decision vectors  $\mathbf{x}_u$  and  $\mathbf{x}_v$ ,  $\mathbf{x}_u$  is said to constraint-dominate  $\mathbf{x}_v$  if one of the following conditions is satisfied:

- (1) both  $\mathbf{x}_u$  and  $\mathbf{x}_v$  are infeasible, and  $CV(\mathbf{x}_u) \leq CV(\mathbf{x}_v)$ ;
- (2)  $\mathbf{x}_u$  is feasible, yet  $\mathbf{x}_v$  is infeasible;
- (3) both  $\mathbf{x}_u$  and  $\mathbf{x}_v$  are feasible, and  $\mathbf{x}_u < \mathbf{x}_v$ .

In general, CDP can motivate the population to approach or enter the feasible region promptly, because feasible solutions are always selected before infeasible solutions in environmental selection.

### The first stage: fast convergence

Considering the fact that CMOEAs need to balance multiple objective functions, the convergence speed is inevitably slow, which presents a major challenge for solving CMOPs with a wide objective space. Therefore, this paper considers transforming a CMOP into multiple constrained single-objective

optimization problems by weighted summation, and solves it using the coarse-grained parallel evolutionary algorithm, that is, a subpopulation in *mainPop* focuses on one of the single-objective problems. Handling with a single-objective optimization problem with constraints is much easier than handling with a multi-objective optimization problem with constraints. Although each subpopulation can only approach some local solutions in the PF of the original problem, the global search ability and convergence performance of the population are significantly improved. Setting appropriate weight vectors and ending conditions can keep the algorithm with good convergence and diversity, and provide high-quality initial solutions for subsequent optimization.

Thus, a CMOP with  $M$  objective functions is transformed into  $(M + 1)$  constrained single-objective optimization problems

$$\begin{cases} \min & \begin{cases} f'_k(\mathbf{x}) = \frac{f_1(\mathbf{x}) + \dots + (\omega + 1)f_k(\mathbf{x}) + \dots + f_M(\mathbf{x})}{M + \omega}, & k = 1, \dots, M \\ f'_k(\mathbf{x}) = \frac{f_1(\mathbf{x}) + \dots + f_M(\mathbf{x})}{M}, & k = M + 1 \end{cases} \\ \text{s.t.} & \mathbf{x} \in \mathbb{S} \\ & g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, p \\ & h_i(\mathbf{x}) = 0, \quad i = p + 1, \dots, q, \end{cases} \quad (4)$$

where  $\omega$  is set to 2 based on the sensitivity analysis in the section “Sensitivity analysis of parameters in CMOEA-TMC”. Since we only change the objective function and keep the constraints unchanged, Eqs. (1) and (4) share the same feasible regions.

The benefits of multi-population strategy are explained below. As shown in Fig. 2, candidate solutions of single population are likely to be distributed in the center of the true PF, which leads it to potentially ignore solutions at the edges of the PF. As shown in Fig. 2, candidate solutions of multi-population may be evenly distributed across different locations of the true PF, which makes the expansion of the population distribution easier.

Next, the implementation details of the first stage are explained in two parts: (1) update procedure and (2) switching condition.

**(1) Update procedure** In the first stage, for *mainPop*, we need it to converge quickly to the true PF, so differential evolution (DE) is considered as the search engine. The pseudocode is presented in Algorithm 2.

We employ three popular generation strategies to generate offspring:

(a) DE/current-to-rand/1:

$$\mathbf{u}_i = \mathbf{x}_i + F * (\mathbf{x}_{r1} - \mathbf{x}_i) + F * (\mathbf{x}_{r2} - \mathbf{x}_{r3}). \quad (5)$$

(b) DE/rand-to-best/1/bin:

$$\mathbf{v}_i = \mathbf{x}_{r1} + F * (\mathbf{x}_{\text{best}} - \mathbf{x}_{r1}) + F * (\mathbf{x}_{r2} - \mathbf{x}_{r3}) \quad (6)$$

**Algorithm 2** Evolution of subpopulation of *mainPop* in the first stage

---

```

Input:  $Pop(k, t) = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)$ ,  $helpPop(t)$ 
Output:  $Pop(k, t + 1)$ 
1:  $Pop(k, t + 1) = \emptyset$ 
2: for  $i = 1 : N$  do
3:   if  $rand < 0.4$  then
4:     Generate the trial vector  $\mathbf{u}_i$  according to DE/current-to-rand/1
5:   else if  $rand < 0.8$  then
6:     Generate the trial vector  $\mathbf{u}_i$  according to DE/rand-to-best/1/bin
7:   else
8:     Generate the trial vector  $\mathbf{u}_i$  according to DE/current-to-best/1
9:   end if
10:  Employ the feasibility rule to compare  $\mathbf{u}_i$  and  $\mathbf{x}_i$ , and store the better one into  $Pop(k, t + 1)$ 
11: end for
12: return  $Pop(k, t + 1)$ 

```

---

$$\mathbf{u}_{i,j} = \begin{cases} \mathbf{v}_{i,j} & \text{if } rand_j < CR \text{ or } j = j_{\text{rand}} \\ \mathbf{x}_{i,j} & \text{else.} \end{cases} \quad (7)$$

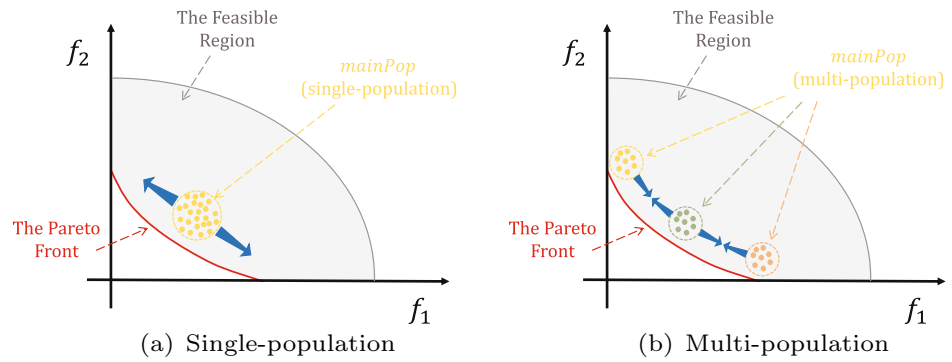
(c) DE/current-to-best/1:

$$\mathbf{u}_i = \mathbf{x}_i + F * (\mathbf{x}_{\text{best}}^{\text{help}} - \mathbf{x}_i) + F * (\mathbf{x}_{r1} - \mathbf{x}_{r2}), \quad (8)$$

where  $\mathbf{v}_i$  is the  $i$ th mutant vector,  $\mathbf{u}_i$  is the  $i$ th trial vector,  $r_1, r_2$ , and  $r_3$  are three mutually distinct integers chosen at random from  $[1, N]$ ,  $\mathbf{x}_{\text{best}}$  denotes the individual with the smallest objective function value in the current subpopulation,  $\mathbf{x}_{\text{best}}^{\text{help}}$  denotes the best individual of *helpPop*,  $rand_j$  is a random number in  $[0, 1]$ ,  $j_{\text{rand}}$  is a random integer in  $[1, D]$ ,  $F$  is the scaling factor, and CR is the crossover control parameter. In addition, as suggested by [20] and [33],  $F$  and CR are randomly chosen from the scaling factor pool (i.e.,  $F_{\text{pool}} = [0.6, 0.8, 1.0]$ ) and the crossover control parameter pool (i.e.,  $CR_{\text{pool}} = [0.1, 0.2, 1.0]$ ), respectively.

In DE/current-to-rand/1,  $\mathbf{x}_i$  learns information from other randomly selected individuals, which facilitates the expansion of the global search. In DE/rand-to-best/1/bin, information about the best individual in the current subpopulation is used. Note that  $\mathbf{x}_{\text{best}}$  in Eq. (6) is determined based on the transformed objective function  $f'_k$ . Before the population enters the feasible region,  $\mathbf{x}_{\text{best}}$  is similar to the randomly selected individual. After the population enters the feasible region, the population can be guided toward the optimal solution by  $\mathbf{x}_{\text{best}}$ . In DE/current-to-best/1, information about the optimal individual in *helpPop* is used. In general, *helpPop* converges more quickly to the unconstrained PF, and if there is an infeasible region between  $\mathbf{x}_{\text{best}}^{\text{help}}$  and *mainPop*, *mainPop* has the opportunity to cross the infeasible region and be led to a more convergent feasible region. Overall, these three variants make good use of information about local and global optimum, allowing the evolution of populations to better balance convergence and diversity.

**Fig. 2** Extending the distribution of *mainPop* using different population settings



For *helpPop*, we use the GA operator for optimization and environmental selection based on the fitness evaluation strategy proposed in SPEA2 [41].

**(2) Switching condition** If we perform a large number of fitness evaluations in the first stage, all individuals in *mainPop* are likely to converge to the optimal solution of the transformed single-objective problems, which would result in the lack of diversity. To obtain high-quality feasible solutions that are close to the PF of the original CMOP but maintain good diversity, we have designed the following three conditions to ensure that the evolution of subpopulation can be terminated early.

- (a) Feasibility condition: Following the suggestion in [20], the feasibility proportion of the current subpopulation is larger than 1/3.
- (b) Convergence condition: Suppose that  $f_{\max,j}$  and  $f_{\min,j}$  represent the maximum and minimum values of the  $j$ th objective function among all feasible solutions found in the current subpopulation, respectively. Then, the  $j$ th objective function of each individual is normalized as

$$\overline{f}_j(i) = \frac{f_j - f_{\min,j}}{f_{\max,j} - f_{\min,j}} \tag{9}$$

Subsequently, we add up all the normalized objective function values

$$\overline{f}'(i) = \sum_{j=1}^M \overline{f}_j(i) \tag{10}$$

Finally, we rank the feasible solutions from smallest to largest according to  $\overline{f}'(i)$ . The convergence index ( $\delta_{\text{con}}$ ) is defined as the maximum difference of  $\overline{f}'(i)$  among the first 1/2 feasible solutions

$$\delta_{\text{con}} = \overline{f}'(\text{median}) - \overline{f}'(1) \tag{11}$$

If  $\delta_{\text{con}}$  is less than 0.2, the convergence condition is regarded to be satisfied.

- (c) Diversity condition: Suppose that  $f_{\text{median},j}$  and  $f_{\text{min},j}$  represent the median and minimum values of the  $j$ th objective function among all feasible solutions found in the current subpopulation during evolution, respectively. The diversity index  $\delta_{\text{div}}$  is defined as

$$\delta_{\text{div}} = \min \left( \frac{f_{\text{median},j} - f_{\text{min},j}}{f_{\text{min},j} + 1e^{-5}} \right) \tag{12}$$

If  $\delta_{\text{div}}$  is less than 0.1, the diversity condition is regarded to be satisfied.

The purpose of the feasibility condition is to ensure that some feasible individuals have been obtained. The convergence condition indicates that some feasible solutions gradually converge to a small region, and the diversity condition indicates that some feasible solutions are relatively dispersed, rather than having converged to a small point. Therefore, for each subpopulation in *mainPop*, the first stage should terminate when the feasibility condition is satisfied and the convergence or diversity condition is met, thus ensuring convergence of the feasible solution and preventing loss of diversity.

When all subpopulations satisfy the above switching condition, the first stage of CMOEA-TMC ends and the second stage starts.

**The second stage: diversity maintenance**

Although we have found promising feasible solutions in the first stage, the performance of the population still needs to be improved. On the one hand, since we only use part of the feasible solutions in subpopulations of *mainPop* to measure the termination conditions in the first stage, it is possible that some individuals are far from the Pareto-optimal solution. On the other hand, due to the lack of an explicit diversity maintenance mechanism in single-objective optimization, the population is slightly less diverse across the PF. As shown in Fig. 3, the set of candidate solutions in *Pop*(2) is far away from the constrained PF, and some solutions in

$Pop(2)$  are dominated by solution A in  $Pop(1)$  and solution B in  $Pop(3)$ . Without diversity maintenance, these solutions may enter the region where  $Pop(1)$  or  $Pop(3)$  is located, as circled in Fig. 3. We expect the diversity and convergence of  $mainPop$  to be further enhanced in the second stage, with a distribution as shown in Fig. 3 in the ideal state.

Therefore, in this stage, we consider using the original CMOP to further optimize for the high-quality candidate solutions generated based on the first stage, to obtain a series of feasible solutions that are well distributed and converge well. Drawing on the ideas of CCMO [30], the coevolution of  $mainPop$  and  $helpPop$  is carried out in a weak cooperation by sharing the offsprings, to enable them have the opportunity to jump out of the infeasible region and evolve toward the global PF. The pseudocode for the second stage is presented in Algorithm 3.

---

**Algorithm 3** Evolution of  $mainPop$  and  $helpPop$  in the second stage

---

**Input:**  $mainPop(t)$ ,  $helpPop(t)$

**Output:**  $mainPop(t + 1)$ ,  $helpPop(t + 1)$

```

1:  $mainPop(t + 1) = \emptyset$ 
2:  $helpPop(t + 1) = \emptyset$ 
3: for  $i = 1 : M + 1$  do
4:    $Parent1 \leftarrow$  Select  $3N/5$  parents from  $Pop(i, t)$ 
5:    $Parent2 \leftarrow$  Select  $N/5$  parents from the remaining  $mainPop(t)$ 
   except  $Pop(i, t)$ 
6:    $Parent3 \leftarrow$  Select  $N/5$  parents from  $helpPop(t)$ 
7:    $Off1 \leftarrow$  Generate  $3N/5$  offsprings based on  $Parent1$  using
   GA operators
8:    $Off2 \leftarrow$  Generate  $N/5$  offsprings based on  $Parent2$  using GA
   operators
9:    $Off3 \leftarrow$  Generate  $N/5$  offsprings based on  $Parent3$  using GA
   operators
10:   $Population1 \leftarrow Pop(i, t) \cup Off1 \cup Off2 \cup Off3$ 
11:   $Pop(i, t + 1) \leftarrow$  Select  $N$  solutions from  $Population1$  by the
   environmental selection
12: end for
13:  $mainPop(t + 1) \leftarrow Pop(1, t + 1) \cup Pop(2, t + 1) \cup \dots \cup Pop(M + 1, t + 1)$ 
14:  $Parent4 \leftarrow$  Select  $4N/5$  parents from  $helpPop(t)$ 
15:  $Parent5 \leftarrow$  Select  $N/5$  parents from  $mainPop(t)$ 
16:  $Off4 \leftarrow$  Generate  $4N/5$  offsprings based on  $Parent4$  using GA
   operators
17:  $Off5 \leftarrow$  Generate  $N/5$  offsprings based on  $Parent5$  using GA
   operators
18:  $Population2 \leftarrow helpPop(t) \cup Off4 \cup Off5$ 
19:  $helpPop(t + 1) \leftarrow$  Select  $N$  solutions from  $Population2$  by the
   environmental selection
20: return  $mainPop(t + 1)$ ,  $helpPop(t + 1)$ 

```

---

In addition, we also set a switching condition for the second stage. When the number of function evaluations reaches 60% of the set maxFE, the second stage of CMOEA-TMC ends and the third stage starts.

## The third stage: extremum mining

After the second stage, we have obtained a series of well-distributed and convergent feasible solutions, but the independent evolution of multiple subpopulations has led to the possibility of some dominant solutions among the generated feasible solutions. Therefore, in the third stage, all subpopulations of  $mainPop$  are combined and compete together for extremum mining. It is worth mentioning that because the helper problem can no longer support  $mainPop$  well, we discard  $helpPop$  in the third stage to save the computational resources.

## Dynamic $\eta$ strategy for equation constraints

For equation constraints in CMOPs, we introduce a very small positive value  $\eta$  to relax equation constraints to inequality constraints. The value of  $\eta$  has a large impact on the convergence of the algorithm. If  $\eta$  takes a larger value, the convergence is usually better, but it also leads to a large difference between the optimized solution and the true solution. If  $\eta$  takes a small value, some infeasible solutions that are partially close to the feasible region may be discarded, because the equation constraints are not satisfied, thus affecting the convergence. Therefore, a dynamic  $\eta$  strategy is used, with  $\eta$  taking a slightly larger value in the early evolution, relaxing the stringency of the constraints on equations in the expectation of finding more promising solution sets, and gradually decreasing to an accepted value in the late evolution.

The specific dynamic adjustment formula for  $\eta$  is as

$$\eta = \begin{cases} 0.001 - (\log \frac{1}{5} \max FE)^{FE} \\ \times (0.001 - 0.0001), & FE \leq \frac{1}{5} \max FE \\ 0.0001, & FE > \frac{1}{5} \max FE, \end{cases} \quad (13)$$

where FE represents the current number of function evaluations and maxFE represents the maximum number of function evaluations set by the user.

## Effectiveness analysis of CMOEA-TMC

To analyze the effectiveness and mechanism of CMOEA-TMC, we show the early, middle, and last population distribution of CMOEA-TMC on 2-objective MW11 in Fig. 4. The gray surfaces denote the feasible regions of the problem, and the black and red lines denote the unconstrained PF and constrained PF, respectively.

As shown in the first column of Fig. 4, the population of ToP converges quickly but can only converge to a single feasible region in the early generations and cannot spread to the other feasible regions at last. For MW11 which has three small feasible regions, ToP with only one population has a



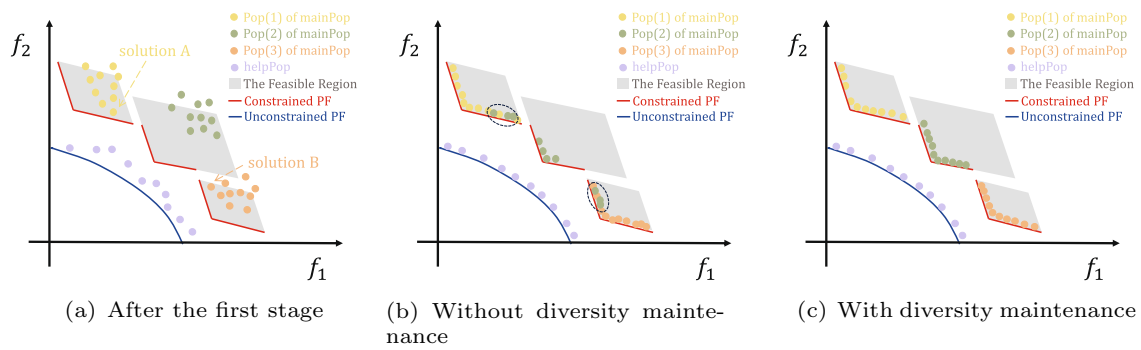


Fig. 3 Comparison of population distribution without and with diversity maintenance

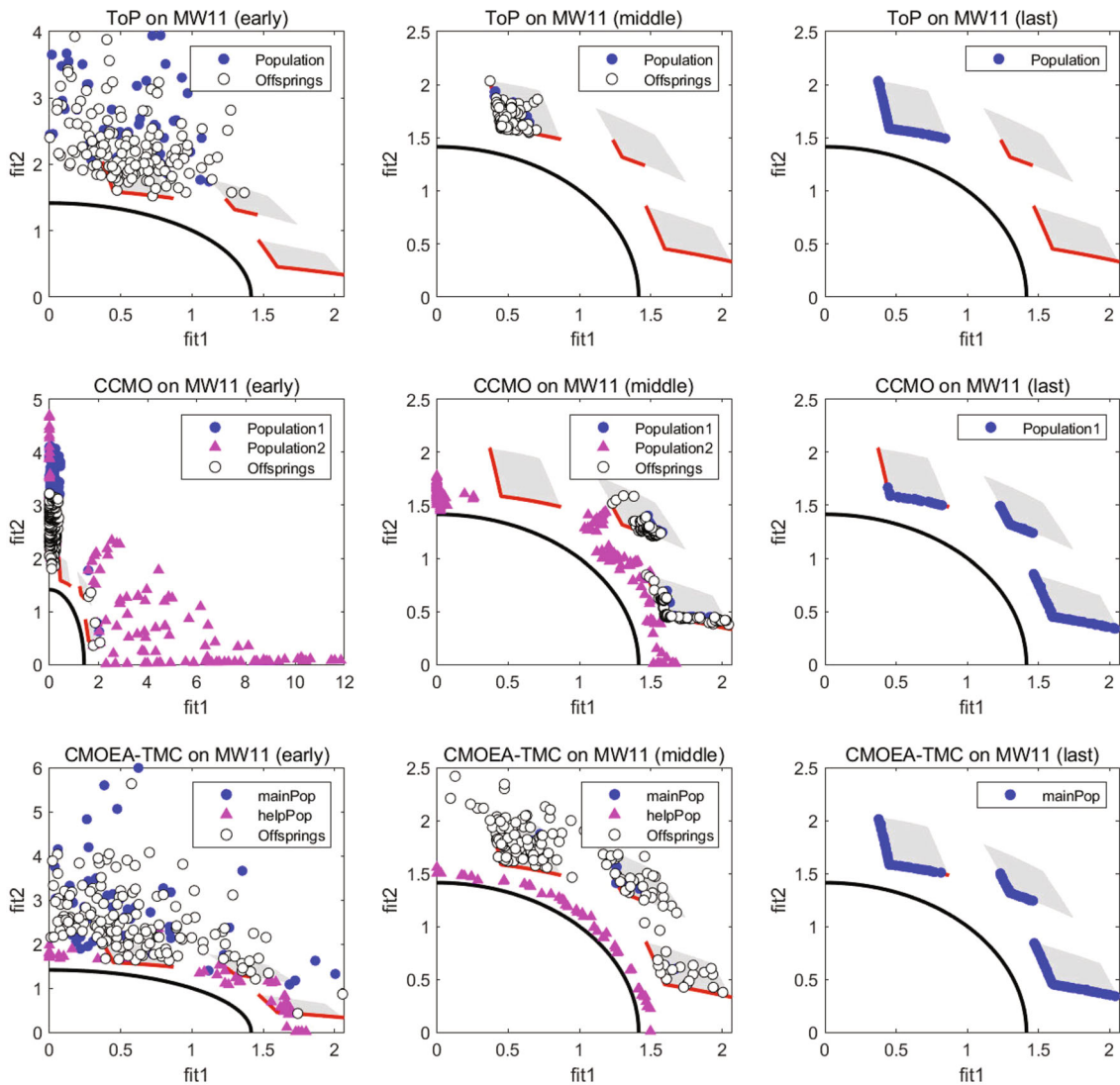


Fig. 4 Populations in the early, middle, and last generations of ToP, CCMO, and CMOEA-TMC on 2-objective MW11

lack of diversity maintenance. CMOEA-TMC and CCMO behave similarly in the early generations. They both have two populations, one considering constraints and the other not considering constraints. However, *helpPop* of CMOEA-

TMC converges faster in the early stage, which greatly helps *mainPop* to perform global search and thus maintain diversity. *Population2* of CCMO converges slower than *Population1*, which may lead to the inability to pro-

vide useful information to *Population1* during coevolution, and therefore, offsprings are mainly generated based on *Population1*.

In the middle generation, CMOEA-TMC evolves independently in three small feasible regions while generating offsprings located in infeasible regions. This indicates that *mainPop* and *helpPop* collaborate effectively and that CMOEA-TMC has the opportunity to spread across infeasible regions into feasible ones in the presence of dispersed feasible regions. In contrast, although CCMO also carries out cooperation between the two populations, the offsprings are mainly generated based on *Population1*, and thus, it has difficulty in accessing the previously missed feasible regions.

Throughout the evolutionary process, CMOEA-TMC explores the search space more fully. Finally, CMOEA-TMC obtains well-diversified final solutions and shows its strong performance.

### Computational complexity of CMOEA-TMC

For CMOEA-TMC, there are three stages: the first stage is DE evolution of subpopulations, the second stage is GA evolution of subpopulations, and the third stage is overall GA evolution. Let us denote the subpopulation size, the number of subpopulations, the population size, the number of objectives, and the dimension of decision vectors by  $N$ ,  $M + 1$ ,  $Q$ ,  $M$ , and  $D$ , where  $Q = N \times (M + 1)$ .

**(1) Time complexity** In the first stage, for each subpopulation, the time complexities of evolutionary variation, crossover, and environmental selection are  $O(ND)$ ,  $O(N)$ , and  $O(N)$ , respectively. Thus, the overall time complexity is  $(M + 1) \times (O(ND) + O(N) + O(N)) = O(MND)$ . In the second stage, for each subpopulation, the time complexities of mating selection, genetic operators, and environmental selection are  $O(N)$ ,  $O(ND)$ , and  $O(MN^2)$ , respectively. In general,  $O(MN^2) > O(ND)$ , so the overall time complexity is  $(M + 1) \times O(MN^2) = O(M^2N^2)$ . In the third stage, the entire population  $Q$  undergoes GA evolution. The time complexities of mating selection, genetic operators, and environmental selection are  $O(Q)$ ,  $O(QD)$ , and  $O(MQ^2)$ , respectively, where  $O(MQ^2)$  is usually the largest. Thus, the time complexity of the third stage is  $O(MQ^2) = O(M^3N^2)$ .

In summary, the time complexity of three stages of CMOEA-TMC is  $O(MND)$ ,  $O(M^2N^2)$ , and  $O(M^3N^2)$ , respectively. Some comparison algorithms that use GA operators, such as CCMO and C-TAEA, usually have a time complexity of  $O(M^3N^2)$ , so the proposed algorithm has an advantage in the first and second stages and can result in a smaller time cost.

**(2) Space complexity** The memory consumption of CMOEAs is concentrated in the storage of the parent and offspring populations. CMOEA-TMC needs to store *mainpop* and offspring population, each of which takes up  $O(DQ)$  space. In the first stage, the transformed single-objective problem is used for performance comparison, so the space used to store the objective values and constraint violation values is  $O(Q)$  and  $O(Q)$ , respectively. In the second and third stages, the space used for storing the objective values and constraint violation values is  $O(MQ)$  and  $O(Q)$ , respectively. Considering the different stages together, the overall space complexity of CMOEA-TMC is  $O(DQ) + O(MQ) + O(Q)$ . The space complexity and the population size are closely correlated, because the different algorithms mainly store the parent and offspring population information in each iteration. When the population size is fixed, the differences in space complexity among different algorithms are small. For CMOEA-TMC, no additional storage content is introduced, so its space complexity is approximate to that of the comparison algorithms.

## Experimental research

### Test sets and parameters setting

To systematically evaluate the performance of CMOEA-TMC, three benchmark problem test suites are employed to compare CMOEA-TMC with other four CMOEAs. Test suites are DOC, MW, and CF. To be specific, DOC is a recently proposed test suite that contains both decision and objective constraints, and it also contains both inequality and equality constraints. MW is also a recently proposed test suite that covers various features, such as small feasible regions, high dimensional decision space, and so on. As a classic test

**Table 1** Parameter settings of the test sets

Test set	Objective number	Population size	maxFE	
DOC	DOC1–DOC7	2	150	300,000
	DOC8, DOC9	3	200	300,000
MW	MW1–3, 5–7, 9–13	2	150	300,000
	MW4,8,14	3	200	300,000
CF	CF1–CF7	2	150	100,000
	CF8–CF10	3	200	100,000

suit, the CF test suit has nonlinear, discontinuous PF. The detailed description of these three test suits can be found in their original paper [20, 23, 38]. Table 1 shows parameter settings of the test sets used.

For CMOEA-TMC, we set  $\omega$ ,  $\delta_{con}$ , and  $\delta_{div}$  to 2, 0.2, and 0.1 based on parameter sensitivity experiments in the section “Sensitivity analysis of parameters in CMOEA-TMC”, and these values showed the best performance. For performance comparison, we consider five advanced CMOEAs, which are ToP, PPS, CCMO, RVEA, and C-TAEA. The parameters of all the compared algorithms are set as suggested in their original papers [2, 13, 18, 20, 30]. 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 and the difference  $\delta$  is less than 0.2. 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  $l = 20$ . Among these algorithms, ToP, PPS, and CCMO use DE to generate offspring solutions, while C-TAEA and RVEA employs the simulated binary crossover [7] and polynomial mutation [8] for generating offspring solutions. Besides, all the experiments in this paper were conducted on the PlatEMO proposed in [29].

**Performance indicators**

To measure the performance of different algorithms, two widely used metrics were employed in our experiments, which can measure both convergence and diversity.

- (1) Inverted Generational Distance (IGD) [1]: Assume that  $P$  is the set of feasible solutions obtained from a CMOEA, and  $P^*$  is the set of points uniformly sampled along the real PF, IGD is defined as

$$IGD(P, P^*) = \frac{1}{|P^*|} \sum_{z^* \in P^*} \text{distance}(z^*, P), \tag{14}$$

where  $\text{distance}(z^*, P)$  is the Euclidean distance between  $z^*$  and the nearest point in  $P$ , and  $|P^*|$  is the total number of points in  $P$ . It can be seen that calculating IGD needs to know the real PF, and the smaller the IGD value, the better the performance of a CMOEA.

- (2) Hypervolume (HV) [40]: HV measures the volume enclosed by  $P$  and a specified reference point in the objective space, and it is defined as

$$HV(P) = \text{VOL} \left( \bigcup_{x \in S} [f_1(x), z_1^r] \times \dots \times [f_M(x), z_M^r] \right), \tag{15}$$

where VOL indicates the Lebesgue measure, and  $z^r = (z_1^r, \dots, z_M^r)^T$  is a user-defined reference point. In this

**Table 2** Mean and standard deviation of the IGD values obtained by TOP, PPS, CCMO, C-TAEA, RVEA, and CMOEA-TMC for 30 independent runs on the DOC test suite

Problem	ToP	PPS	CCMO	C-TAEA	RVEA	CMOEA-TMC
DOC1	4.6096e-3(3.19e-4)≈	6.2519e-2(2.83e-2)–	5.3043e-3(5.71e-4)–	5.0599e+2(2.22e+2)–	1.8480e+0(1.41e+0)–	<b>4.5688e-3(2.52e-4)</b>
DOC2	NaN(NaN)	9.6971e-2(1.22e-1)–	8.0586e-2(1.64e-1)–	NaN(NaN)	NaN(NaN)	<b>4.5574e-2(3.50e-2)</b>
DOC3	3.5952e+1(7.60e+1)–	6.2604e-1(3.78e-1)–	4.5965e+2(2.96e+2)–	NaN(NaN)	5.9039e+2(1.50e+2)–	<b>2.4830e-1(2.41e-1)</b>
DOC4	6.7212e-2(5.79e-2)≈	2.5332e-1(4.18e-2)–	2.6142e-2(3.02e-3)≈	2.4408e+2(4.80e+2)–	1.1700e+0(4.78e-1)–	<b>2.5907e-2(6.53e-3)</b>
DOC5	1.4047e+1(3.97e+1)–	1.2918e+1(3.83e+1)–	3.3503e+1(5.65e+1)≈	NaN(NaN)	NaN(NaN)	<b>1.5739e-1(1.55e-1)</b>
DOC6	4.1585e+0(1.43e+0)–	5.1047e-1(4.36e-2)–	3.9353e-1(2.58e-1)–	3.2767e+1(2.99e+1)–	1.6149e+0(1.88e+0)–	<b>2.1758e-3(4.55e-4)</b>
DOC7	4.8121e-1(1.78e-1)–	2.7610e-1(1.37e-1)–	<b>1.9610e-3(1.21e-4)+</b>	NaN(NaN)	7.1870e+0(2.25e+0)–	6.2402e-2(1.28e-1)
DOC8	4.9400e+1(2.00e+1)–	1.2585e+2(3.64e+1)–	2.6182e-1(8.28e-2)–	4.3720e+2(1.53e+2)–	2.1950e+2(5.75e+1)–	<b>1.7235e-1(1.73e-1)</b>
DOC9	2.0245e-1(4.57e-2)–	2.7110e-1(1.71e-2)–	7.8181e-2(1.18e-2)–	7.8976e-1(1.74e-1)–	1.2820e+0(1.93e-1)–	<b>5.0322e-2(1.07e-2)</b>
+/-≈	0/6/2	0/9/0	1/6/2	0/5/0	0/7/0	

The best results are highlighted in bold

paper, the reference point is set to 1.1 times the lowest point of the PF, so that it can be dominated by all the feasible solutions in PF. It is worth noting that the larger the HV value, the better the performance of a CMOEA.

### Experimental results of DOC problems

On the DOC problem test set, Table 2 shows the average and standard deviation of the IGD values for 30 independent runs of ToP, PPS, CCMO, C-TAEA, RVEA, and CMOEA-TMC on the DOC problem test set. Table 3 shows the average and standard deviation of the HV values. In those two tables, Wilcoxon’s rank-sum two-sided comparison [17] at the 0.05 significance level is performed to test the statistical significance of the experimental results between two algorithms. The two-sided Wilcoxon rank-sum test tests the null hypothesis that the data from two samples are continuously distributed samples with equal medians, against the alternative that they are not. For convenience, “+”, “-”, and “≈” denote that a peer CMOEA performs better than, worse than, and similar to CMOEA-TMC, respectively. NaN indicates that the algorithm failed to find feasible solutions.

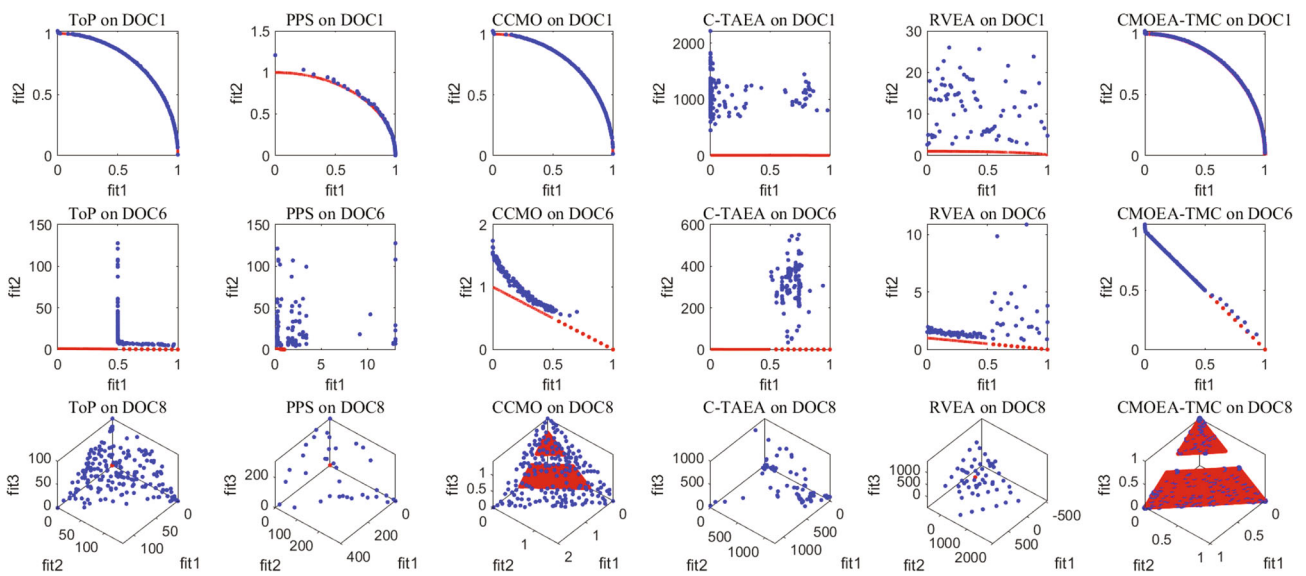
As can be seen from Tables 2 and 3, CMOEA-TMC clearly outperforms the other four compared algorithms on the DOC test set, achieving the best results on eight of the nine test problems, followed by CCMO, which achieves one best result on DOC7. It is found that DOC7 contains fewer constraints, and the helper population in CCMO can play a greater role. The DOC test set contains decision space constraints and objective space constraints, and the initial distribution of the objective function is more divergent, so it is more complex and difficult to solve. CMOEA-TMC shows better convergence and diversity on this series of problems, suggesting that the algorithm’s three-stage strategy and multiple subpopulations’ coevolutionary design are effective.

Figure 5 plots the performance of the six MOEAs on the three test problems DOC1, DOC6, and DOC8, where DOC1 and DOC6 are two-objective problems with continuous and discontinuous feasible domains, respectively, and DOC8 is a three-objective problem with a discontinuous feasible domain. In Fig. 5, the red line represents the true PF and the blue points are the final solutions obtained by the algorithm. It is clear that CMOEA-TMC exhibits the better convergence in all three problems, while the other CMOEAs still retain a large number of individuals far from the true PF. The fast convergence rate of CMOEA-TMC is mainly attributed to the single-objective evolution in the first stage and the coevolution with *helpPop*, which makes it easier for *mainPop* to skip the infeasible region and converge quickly. For DOC6 and DOC8 with discontinuous feasible regions, CMOEA-TMC shows better diversity than other CMOEAs, which is attributed to the strategy of independent parallel evolution of multiple subpopulations.

**Table 3** Mean and standard deviation of the HV values obtained by TOP, PPS, CCMO, C-TAEA, RVEA, and CMOEA-TMC for 30 independent runs on the DOC test suite

Problem	ToP	PPS	CCMO	C-TAEA	RVEA	CMOEA-TMC
DOC1	<b>3.4633e-1(3.44e-4)+</b>	2.8577e-1(2.03e-2)-	3.4546e-1(7.10e-4)≈	0.0000e+0(0.00e+0)-	1.3597e-2(3.10e-2)-	3.4541e-1(1.09e-3)
DOC2	NaN(NaN)	5.0934e-1(8.78e-2)-	5.6919e-1(1.04e-1)-	NaN(NaN)	NaN(NaN)	<b>5.7272e-1(3.41e-2)</b>
DOC3	0.0000e+0(0.00e+0)-	7.4665e-2(9.03e-2)-	2.0902e-2(7.96e-2)-	NaN(NaN)	0.0000e+0(0.00e+0)-	<b>2.1779e-1(9.07e-2)</b>
DOC4	4.8334e-1(6.93e-2)≈	2.7284e-1(3.55e-2)-	5.3253e-1(3.78e-3)≈	0.0000e+0(0.00e+0)-	6.9933e-3(1.24e-2)-	<b>5.3452e-1(7.17e-3)</b>
DOC5	2.9736e-1(1.45e-1)-	2.0431e-1(1.55e-1)-	3.5159e-1(2.16e-1)≈	NaN(NaN)	NaN(NaN)	<b>3.9958e-1(9.83e-2)</b>
DOC6	0.0000e+0(0.00e+0)-	1.9622e-1(2.68e-2)-	1.3986e-1(1.72e-1)-	0.0000e+0(0.00e+0)-	4.6437e-2(8.09e-2)-	<b>5.3397e-1(2.28e-2)</b>
DOC7	8.5318e-2(6.80e-2)-	1.9518e-1(9.45e-2)-	<b>5.2629e-1(8.46e-3)+</b>	NaN(NaN)	0.0000e+0(0.00e+0)-	4.0514e-1(1.12e-1)
DOC8	0.0000e+0(0.00e+0)-	0.0000e+0(0.00e+0)-	5.5404e-1(1.05e-1)-	0.0000e+0(0.00e+0)-	0.0000e+0(0.00e+0)-	<b>6.6768e-1(1.86e-1)</b>
DOC9	2.0412e-1(3.26e-2)-	1.1291e-1(1.32e-2)-	3.0681e-1(6.99e-3)-	6.1681e-4(1.07e-3)-	0.0000e+0(0.00e+0)-	<b>3.3621e-1(5.74e-3)</b>
+/-≈	1/6/1	0/9/0	1/5/3	0/5/0	0/7/0	

The best results are highlighted in bold



**Fig. 5** True PF and final solutions achieved by each algorithm on DOC1, DOC6, and DOC8

Meanwhile, to verify the effectiveness of the dynamic  $\eta$  strategy, the static  $\eta$  strategy and the dynamic  $\eta$  strategy run 30 times independently for the three problems of DOC3, DOC5, and DOC7 with equation constraints, respectively. The mean and standard deviation of the obtained IGD values are calculated, and the final results using the static  $\eta$  strategy and the dynamic  $\eta$  strategy are presented in Table 4. The bolded data in Table 4 indicate the solution with the smaller IGD value obtained under the same algorithm for the same test problem with equation constraints, in both cases with static  $\eta$  strategy and the dynamic  $\eta$  strategy. It can be seen that IGD values of compared algorithm decreased with the dynamic  $\eta$  strategy, except for the case where no feasible solution is obtained resulting in an NaN IGD value. After performing the two-sided Wilcoxon rank-sum test on the data, the IGD values of the static  $\eta$  strategy are usually worse than or similar to the IGD values of the dynamic  $\eta$  strategy, which means that the convergence and diversity of algorithms are improved after adopting the dynamic  $\eta$  strategy.

Based on the above analysis, we can see that the novel multiple subpopulation parallel evolution method and three-stage strategy of CMOEA-TMC have good results on the DOC test set, and that the proposed dynamic  $\eta$  adjustment strategy for equation constraints has some generality and can improve the performance of CMOEA for CMOPs with equation constraints.

### Experimental results of MW and CF problems

To further verify the validity of CMOEA-TMC, we used MW and CF test suites for performance testing. Table 5 shows the average and standard deviation of the IGD values for 30

independent runs of ToP, PPS, CCMO, C-TAEA, RVEA, and CMOEA-TMC on the MW and CF test problem sets, and the HV results are presented in Table 6.

As shown in Tables 5 and 6, CMOEA-TMC achieves the best results on 11 problems with 6 for C-TAEA, 4 for PPS, 2 for RVEA, and 1 for CCMO. For the MW test set, CMOEA-TMC and C-TAEA perform better, probably because most of the MW functions have scattered and narrow feasible domains, so algorithms that are more concerned with diversity have better results. CMOEA-TMC maintains the uniform distribution of solution sets around the objective space through parallel evolution of subpopulations, and C-TAEA uses a diversity-oriented weighted vector method to keep the diversity in each direction, both of them focusing more on the maintenance of diversity in different directions. For the CF test set, the algorithms with DE operators (PPS and CMOEA-TMC) outperform other methods, because the CF functions have low-dimensional decision variables and discrete feasible regions, and DE may help to maintain good diversity.

Figure 6 plots the performance of the six CMOEAs on the three test problems MW5, MW12, and CF6. The red points are the true PF, the blue points are the final solutions obtained by each algorithm, and the gray areas represent the feasible regions in the target space. For MW5 with discontinuous and small feasible regions, CMOEA-TMC shows better diversity performance than other CMOEAs. For MW12 with feasible regions separated by infeasible regions, CMOEA-TMC crosses the local optimum and evolves toward the true PF, shows better convergence performance than other CMOEAs. For CF6 with true PF in segments, CMOEA-TMC can fully explore the target space and exhibits powerful performance.

**Table 4** Mean and standard deviation of the IGD values before and after 30 independent runs of adding dynamic  $\eta$  strategy on the DOC3, DOC5, and DOC7

Algorithm	DOC3		DOC5		DOC7	
	Static $\eta$	Dynamic $\eta$	Static $\eta$	Dynamic $\eta$	Static $\eta$	Dynamic $\eta$
ToP	1.8171e+2(2.31e+2)–	<b>3.5952e+1(7.60e+1)</b>	2.7751e+1(5.14e+1)≈	<b>1.4047e+1(3.97e+1)</b>	8.2471e–1(4.74e–1)–	<b>4.8121e–1(1.78e–1)</b>
PPS	1.0851e+0(1.10e+0)≈	<b>6.2604e–1(3.78e–1)</b>	5.3718e+1(6.54e+1)–	<b>1.2918e+1(3.83e+1)</b>	3.9156e–1(1.18e–1)–	<b>2.7610e–1(1.37e–1)</b>
CCMO	7.1600e+2(4.57e+2)–	<b>4.5965e+2(2.96e+2)</b>	3.8690e+1(6.00e+1)–	<b>3.3503e+1(5.65e+1)</b>	2.0435e–3(1.50e–4)–	<b>1.9610e–3(1.21e–4)</b>
C-TAEA	NaN(NaN)	NaN(NaN)	NaN(NaN)	NaN(NaN)	NaN(NaN)	NaN(NaN)
RVEA	6.6408e+2(2.29e+2)≈	<b>5.9039e+2(1.50e+2)</b>	NaN(NaN)	NaN(NaN)	8.1398e+0(2.71e+0)≈	<b>7.1870e+0(2.25e+0)</b>
CMOEA-TMC	6.8325e–1(4.68e–1)–	<b>2.4830e–1(2.41e–1)</b>	9.0523e–1(3.90e+0)≈	<b>1.5739e–1(1.55e–1)</b>	1.4447e–1(1.89e–1)–	<b>6.2402e–2(1.28e–1)</b>
+/-≈	0/3/2		0/2/2		0/4/1	

The best results are highlighted in bold

Therefore, it can be concluded that the proposed CMOEA-TMC has better overall performance than compared CMOEAs when solving the benchmark CMOP, being able to better balance convergence and diversity, precisely due to the coevolution method and the three-stage algorithm design.

### Sensitivity analysis of parameters in CMOEA-TMC

In CMOEA-TMC, there are three parameters (i.e.,  $\omega$ ,  $\delta_{con}$  and  $\delta_{div}$ ) that have an impact on the performance of the algorithm.  $\omega$  determines the weight of different objective functions when converting a multi-objective problem into a single-objective problem. A large value of  $\omega$  may cause subpopulations to fall into local optimum, while a small value of  $\omega$  may cause overlapping search areas.  $\delta_{con}$  and  $\delta_{div}$  determine when to switch from the first stage to the second stage. A small value of  $\delta_{con}$  or  $\delta_{div}$  may result in the feasible solutions clustering in a very small region. However, with a large value of  $\delta_{con}$  or  $\delta_{div}$ , the solution set may be far from the true PF when the first stage ends. Therefore, we conducted the sensitivity analysis of above three parameters.

We tested the performance on six test instances: DOC1, DOC6, DOC8, MW5, MW12, and CF6. Their PFs have different characteristics and therefore can provide insight on various aspects. We chose five different  $\omega$  values: 1/2, 1, 2, 3, 4; four different  $\delta_{con}$  values: 0.1, 0.2, 0.3, 0.4; and four different  $\delta_{div}$  values: 0.05, 0.1, 0.15, 0.2. The average IGD values of the algorithm over ten independent runs were used as the evaluation criterion. The test results are shown in Figs. 7 and 8.

We can observe that, overall, CMOEA-TMC exhibits better performance with  $\omega = 2$ ,  $\delta_{con} = 0.2$  and  $\delta_{div} = 0.1$ . Therefore, in this paper,  $\omega$  is set to 2,  $\delta_{con}$  is set to 0.2 and  $\delta_{div}$  is set to 0.1.

### Experiments on computing performance

To verify the computing performance of CMOEA-TMC, the average computation time of five comparative algorithms was counted. All experiments were performed on the computer with an i7-10510U CPU, on the PlatEMO built in MATLAB R2021b. Table 7 shows the average computation time spent by each algorithm running on the test problems in each test suit.

As shown in Table 7, for DOC and MW test suits, CMOEA-TMC has the shortest computation time except for RVEA. For CF test suite, the average computation time of CMOEA-TMC is slightly larger than that of ToP and RVEA, but much smaller than the other three algorithms. Therefore, the proposed CMOEA-TMC is computationally efficient.

**Table 5** Mean and standard deviation of the IGD values obtained by TOP, PPS, CCMO, C-TAEA, RVEA, and CMOEA-TMC for 30 independent runs on the MW and CF test suites

Problem	ToP	PPS	CCMO	C-TAEA	RVEA	CMOEA-TMC
MW1	2.6445e-1(3.05e-1)-	2.2610e-2(1.05e-1)-	1.4728e-3(4.46e-5)-	1.2705e-3(4.17e-5)≈	3.1098e-3(7.65e-4)-	<b>1.2688e-3(2.36e-5)</b>
MW2	1.4422e-1(1.01e-1)-	1.5039e-1(9.01e-2)-	8.4399e-2(3.82e-2)-	<b>1.2126e-2(5.58e-3)≈</b>	1.9318e-2(1.68e-2)≈	1.5954e-2(5.05e-3)
MW3	2.3414e-1(3.93e-1)-	4.2520e-3(5.48e-4)-	3.4715e-3(9.43e-5)+	<b>3.2099e-3(1.54e-4)+</b>	6.8629e-3(1.32e-3)-	3.9634e-3(2.13e-4)
MW4	2.0863e-1(9.65e-2)-	3.9100e-2(1.11e-3)≈	4.0739e-2(7.18e-4)-	3.0210e-2(1.46e-4)+	<b>2.7467e-2(1.59e-5)+</b>	4.0063e-2(2.95e-3)
MW5	7.3047e-1(2.50e-2)-	3.9956e-1(3.73e-1)-	1.2293e-3(4.52e-4)-	7.5648e-3(2.39e-3)-	6.3404e-2(1.44e-2)-	<b>9.0055e-4(1.26e-3)</b>
MW6	4.4456e-1(4.10e-1)-	5.2883e-1(3.49e-1)-	4.2372e-1(1.86e-1)-	<b>4.7965e-3(2.66e-3)+</b>	1.0126e-2(3.53e-3)≈	1.5584e-2(8.77e-3)
MW7	2.6632e-2(8.02e-2)-	3.6291e-3(2.72e-4)-	3.6431e-3(2.53e-4)-	5.9388e-3(1.19e-3)-	5.8119e-3(7.41e-4)-	<b>3.3066e-3(1.61e-4)</b>
MW8	2.7452e-1(3.19e-1)-	1.3052e-1(5.32e-2)-	1.1317e-1(3.63e-2)-	3.4385e-2(9.01e-4)+	<b>3.3775e-2(6.72e-4)+</b>	4.0992e-2(4.49e-3)
MW9	3.5386e-1(3.33e-1)-	7.7339e-3(2.81e-3)-	3.7100e-3(1.04e-3)-	5.7940e-3(3.22e-4)-	9.7687e-3(2.88e-3)-	<b>3.3300e-3(1.26e-4)</b>
MW10	3.0739e-1(3.26e-1)-	4.5865e-1(2.35e-1)-	3.4583e-1(1.46e-1)-	<b>7.0742e-3(8.48e-3)≈</b>	8.7430e-3(4.55e-3)≈	1.1255e-2(8.15e-3)
MW11	1.9265e-1(2.61e-1)-	4.9088e-3(1.29e-4)-	4.5388e-3(1.23e-4)≈	8.8170e-3(7.83e-4)-	2.7505e-2(5.71e-3)-	<b>4.4921e-3(1.10e-4)</b>
MW12	4.9743e-1(3.24e-1)-	2.0128e-2(8.15e-2)-	2.6083e-2(1.23e-1)≈	5.3772e-3(4.48e-4)-	4.8729e-3(8.82e-4)-	<b>3.6284e-3(1.15e-4)</b>
MW13	6.3809e-1(4.93e-1)-	3.8820e-1(3.15e-1)-	2.0842e-1(6.37e-2)-	<b>2.1348e-2(1.17e-2)+</b>	2.2422e-2(1.47e-2)+	5.8758e-2(3.10e-2)
MW14	1.4012e-1(1.06e-1)-	9.2077e-2(5.02e-3)-	7.3492e-2(8.78e-4)+	<b>7.1294e-2(1.49e-3)+</b>	9.6096e-2(1.48e-3)-	8.6330e-2(3.53e-3)
CF1	5.2929e-3(1.27e-3)+	2.7365e-3(7.76e-4)+	<b>2.1262e-3(1.07e-3)+</b>	4.1975e-2(6.82e-3)-	8.0072e-2(6.55e-3)-	1.3895e-2(1.81e-3)
CF2	6.9696e-3(3.83e-3)+	<b>2.8709e-3(1.83e-4)+</b>	9.9034e-3(8.84e-3)+	3.5649e-2(1.54e-2)-	5.6236e-2(1.20e-2)-	2.7056e-2(1.07e-2)
CF3	3.3468e-1(6.88e-2)-	2.9225e-1(1.28e-1)-	3.7455e-1(5.99e-2)-	2.5505e-1(1.39e-1)≈	2.4811e-1(6.78e-2)-	<b>2.1817e-1(7.56e-2)</b>
CF4	5.0742e-2(1.18e-2)+	<b>3.0523e-2(1.65e-2)+</b>	5.6159e-2(1.30e-2)≈	8.1321e-2(2.30e-2)-	1.3174e-1(4.37e-2)-	6.0952e-2(1.47e-2)
CF5	3.0885e-1(7.45e-2)-	1.9793e-1(7.14e-2)-	4.1548e-1(1.53e-1)-	3.2812e-1(1.39e-1)-	3.2526e-1(1.47e-1)-	<b>1.5326e-1(4.47e-2)</b>
CF6	6.3406e-2(2.90e-2)-	5.0577e-2(2.18e-2)-	3.8062e-2(4.59e-3)-	9.0020e-2(4.22e-2)-	8.6162e-2(1.77e-2)-	<b>3.5772e-2(1.09e-2)</b>
CF7	2.3735e-1(1.17e-1)≈	2.2682e-1(8.82e-2)≈	1.8803e-1(6.49e-2)≈	2.9011e-1(1.83e-1)-	2.3359e-1(1.24e-1)≈	<b>1.8625e-1(7.42e-2)</b>
CF8	4.0756e-1(8.63e-2)-	<b>1.6061e-1(4.08e-2)+</b>	1.9616e-1(3.71e-2)≈	2.1998e-1(3.25e-2)-	3.1590e-1(8.63e-2)-	2.0620e-1(4.20e-2)
CF9	1.2308e-1(1.42e-2)-	<b>7.6050e-2(7.56e-3)+</b>	8.4188e-2(3.82e-3)+	9.9064e-2(1.22e-2)≈	1.9138e-1(5.85e-2)-	1.0262e-1(1.37e-2)
CF10	2.0478e+0(1.55e+0)-	3.3294e-1(9.69e-2)≈	9.7393e-1(2.32e-1)-	3.1804e-1(1.41e-1)≈	5.0603e-1(1.12e-1)-	<b>2.8450e-1(6.84e-2)</b>
+/-≈	3/201	5/16/3	5/14/5	6/12/6	3/17/4	

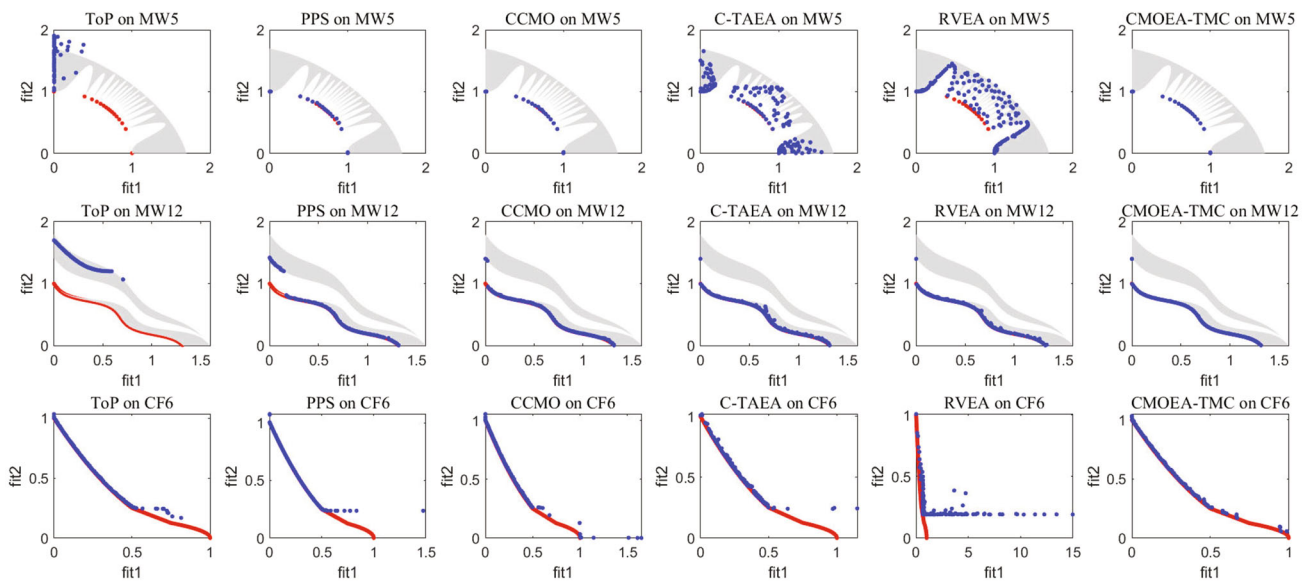
The best results are highlighted in bold

**Table 6** Mean and standard deviation of the HV values obtained by TOP, PPS, CCMO, C-TAEA, RVEA, and CMOEA-TMC for 30 independent runs on the MW and CF test suites

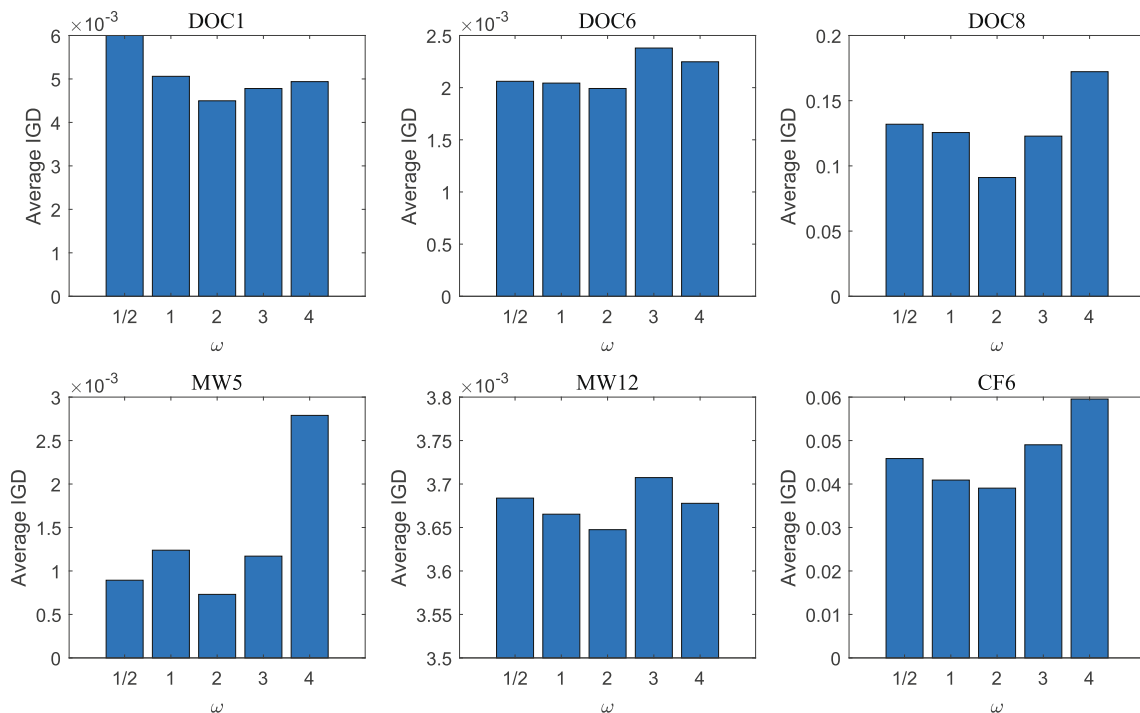
Problem	TOP	PPS	CCMO	C-TAEA	RVEA	CMOEA-TMC
MW1	2.6798e-1(1.94e-1)-	4.7072e-1(9.51e-2)-	4.8940e-1(9.92e-5)-	4.8966e-1(1.59e-4)-	4.8660e-1(9.70e-4)-	<b>4.9030e-1(1.42e-5)</b>
MW2	3.9983e-1(1.11e-1)-	3.9151e-1(9.76e-2)-	4.6609e-1(4.86e-2)-	<b>5.6796e-1(9.74e-3)≈</b>	5.5648e-1(2.62e-2)≈	5.6103e-1(8.91e-3)
MW3	3.9662e-1(2.28e-1)-	5.4563e-1(6.78e-4)+	5.4548e-1(1.55e-4)≈	<b>5.4653e-1(2.39e-4)+</b>	5.4297e-1(1.28e-3)-	5.4542e-1(3.63e-4)
MW4	5.7803e-1(1.28e-1)-	8.3530e-1(3.10e-3)-	8.3555e-1(9.00e-4)-	8.5139e-1(9.14e-5)+	<b>8.5309e-1(1.55e-5)+</b>	8.3899e-1(3.93e-3)
MW5	8.5150e-2(6.82e-3)-	1.9784e-1(1.16e-1)-	3.2383e-1(2.67e-4)-	3.1992e-1(1.41e-3)-	2.7727e-1(1.07e-2)-	<b>3.2432e-1(4.84e-4)</b>
MW6	1.4132e-1(1.02e-1)-	1.0652e-1(9.13e-2)-	1.3614e-1(6.05e-2)-	<b>3.2262e-1(5.05e-3)+</b>	3.1686e-1(6.43e-3)≈	3.0890e-1(1.21e-2)
MW7	3.9489e-1(2.90e-2)-	<b>4.1337e-1(8.74e-5)+</b>	4.1324e-1(9.64e-5)≈	4.1023e-1(9.07e-4)-	4.0858e-1(1.09e-3)-	4.1320e-1(2.96e-4)
MW8	3.2346e-1(1.66e-1)-	3.6144e-1(8.81e-2)-	3.9518e-1(5.97e-2)-	5.5215e-1(4.60e-3)+	<b>5.5867e-1(5.12e-3)+</b>	5.2977e-1(1.30e-2)
MW9	1.8480e-1(1.72e-1)-	3.8864e-1(3.72e-3)-	3.9795e-1(3.72e-3)-	3.9723e-1(1.27e-3)-	3.8720e-1(3.80e-3)-	<b>4.0045e-1(1.31e-3)</b>
MW10	2.7309e-1(1.56e-1)-	1.9949e-1(1.10e-1)-	2.4757e-1(6.70e-2)-	<b>4.4855e-1(9.97e-3)≈</b>	4.4556e-1(7.28e-3)≈	4.4261e-1(9.84e-3)
MW11	3.9268e-1(6.82e-2)-	4.4833e-1(8.59e-5)-	4.4785e-1(1.94e-4)-	4.4546e-1(5.51e-4)-	4.3997e-1(2.12e-3)-	<b>4.4842e-1(9.94e-5)</b>
MW12	1.9253e-1(2.32e-1)-	5.9319e-1(5.55e-2)-	5.8532e-1(1.08e-1)-	6.0367e-1(3.64e-4)-	6.0435e-1(1.10e-3)-	<b>6.0608e-1(1.31e-4)</b>
MW13	2.3605e-1(1.28e-1)-	2.8570e-1(1.05e-1)-	3.5552e-1(3.73e-2)-	4.6787e-1(6.81e-3)+	<b>4.6869e-1(8.52e-3)+</b>	4.5164e-1(1.38e-2)
MW14	4.4427e-1(4.77e-2)-	4.7336e-1(1.95e-3)+	4.7505e-1(1.02e-3)+	<b>4.8321e-1(1.18e-3)+</b>	4.7464e-1(1.61e-3)+	4.6979e-1(2.31e-3)
CF1	5.5966e-1(1.55e-3)+	5.6278e-1(9.45e-4)+	<b>5.6351e-1(1.33e-3)+</b>	5.1609e-1(6.98e-3)-	4.6256e-1(9.44e-3)-	5.4883e-1(2.10e-3)
CF2	6.6840e-1(7.48e-3)+	<b>6.7644e-1(2.12e-3)+</b>	6.6803e-1(9.24e-3)+	6.2483e-1(2.17e-2)-	5.9038e-1(2.51e-2)-	6.4316e-1(1.85e-2)
CF3	7.4213e-2(3.25e-2)-	1.6300e-1(4.52e-2)-	4.2375e-2(3.91e-2)-	1.7863e-1(4.33e-2)≈	1.5528e-1(4.24e-2)-	<b>1.8741e-1(3.81e-2)</b>
CF4	4.6643e-1(1.62e-2)+	<b>4.9445e-1(2.10e-2)+</b>	4.6070e-1(1.09e-2)≈	4.2496e-1(2.09e-2)-	3.6114e-1(5.03e-2)-	4.5619e-1(1.84e-2)
CF5	2.0785e-1(5.77e-2)-	3.5236e-1(4.26e-2)≈	1.0007e-1(9.94e-2)-	2.7460e-1(7.21e-2)-	2.6988e-1(7.11e-2)-	<b>3.6812e-1(4.45e-2)</b>
CF6	6.4930e-1(1.58e-2)-	6.5133e-1(1.43e-2)-	6.6560e-1(3.80e-3)≈	6.3667e-1(2.08e-2)-	5.8586e-1(2.12e-2)-	<b>6.6633e-1(4.82e-3)</b>
CF7	4.2398e-1(9.97e-2)-	4.5638e-1(6.20e-2)-	4.5806e-1(6.38e-2)-	4.3131e-1(1.40e-1)-	4.8774e-1(7.87e-2)≈	<b>5.1090e-1(5.27e-2)</b>
CF8	1.9612e-1(5.40e-2)-	<b>3.5378e-1(5.10e-2)+</b>	3.2817e-1(2.58e-2)+	2.7367e-1(3.77e-2)≈	1.6144e-1(2.93e-2)-	2.8074e-1(3.50e-2)
CF9	3.9361e-1(2.32e-2)≈	<b>4.6510e-1(1.36e-2)+</b>	4.3443e-1(9.82e-3)+	4.0528e-1(1.70e-2)≈	3.0915e-1(5.56e-2)-	4.0055e-1(1.49e-2)
CF10	0.0000e+0(0.00e+0)-	<b>1.7881e-1(3.71e-2)≈</b>	9.8982e-4(3.22e-3)-	1.7840e-1(6.76e-2)≈	8.8152e-2(2.05e-2)-	1.7015e-1(3.31e-2)
+/-≈	3/20/1	8/14/2	5/15/4	6/12/6	4/16/4	

The best results are highlighted in bold





**Fig. 6** True PF and final solutions achieved by each algorithm on MW5, MW12, and CF6

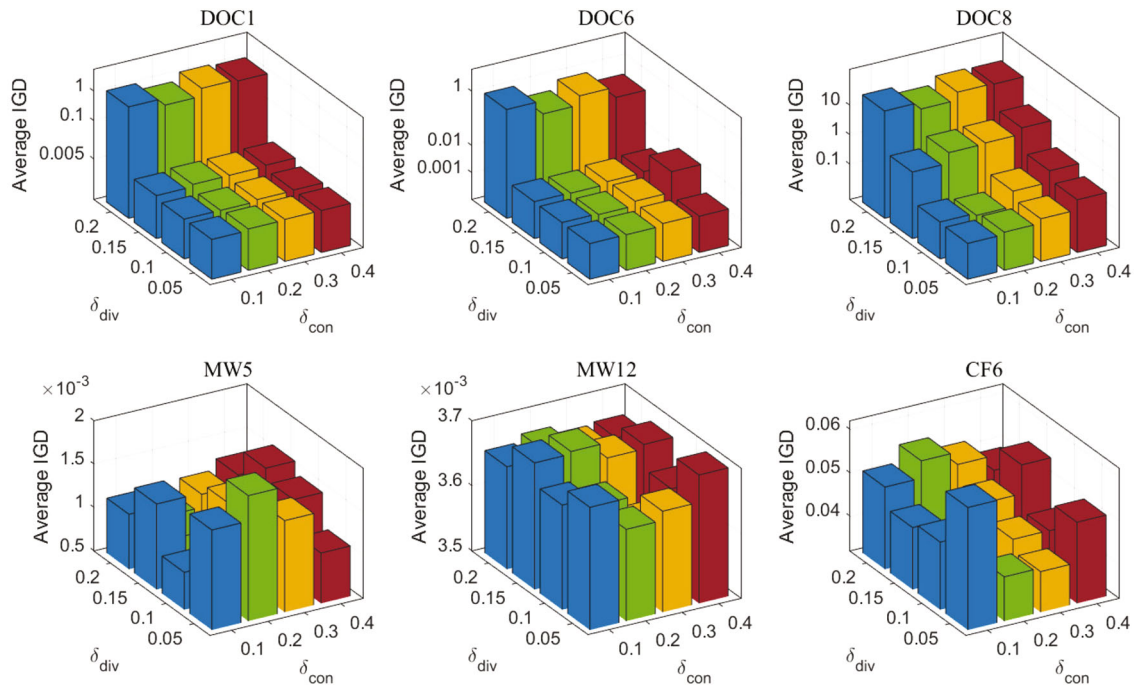


**Fig. 7** Average IGD values provided by CMOEA-TMC with different  $\omega$  on DOC1, DOC6, DOC8, MW5, MW12, and CF6

**Experimental results of real-world boiler combustion optimization problem**

After testing CMOEA-TMC’s performance in solving a series of benchmark problems, this subsection examines the performance of CMOEA-TMC and four comparison CMOEAs on a real-world boiler combustion optimization problem.

Power plant boilers have generated a lot of valuable historical data over many years of operation, which can be used to model the relationship between operating variables and performance indicators. The model can be optimized under specific constraints and the optimization results can provide guidance for adjustments to the boiler combustion process, thus improving the thermal efficiency and reducing the generation of pollutants.



**Fig. 8** Average IGD values provided by CMOEA-TMC with different combinations of  $\delta_{con}$  and  $\delta_{div}$  on DOC1, DOC6, DOC8, MW5, MW12, and CF6

**Table 7** Mean time consumption (in seconds) of CMOEA-TMC and the peer CMOEAs on DOC, MW, and CF test suits

Problem	ToP	PPS	CCMO	C-TAEA	RVEA	CMOEA-TMC
DOC test suit	67.40	105.42	162.86	611.80	8.09	38.60
MW test suit	44.76	74.95	159.88	446.87	8.53	25.20
CF test suit	5.47	26.67	25.69	131.33	3.41	9.63

In our experiment, the gradient boosted decision tree (GBDT) is used to model the operational data generated during boiler combustion. The inputs to the model are 19 non-adjustable operational variables and 16 adjustable operational variables, corresponding to variables, such as coal feed and damper opening in boiler combustion. The outputs of the model are four max–min normalized performance indicators corresponding to the outlet temperatures of air preheaters A and B and the  $NO_x$  content of the inlet flue A and B. For the training process of GBDT, we used 5466 offline historical samples with the training data set of  $5466 \times 19$ . Adequate historical data ensure the accuracy of GBDT, and an additional sample is used to test the performance of CMOEAs. The training and testing process is performed offline.

The demand of the power plant is to reduce these four performance indicators as much as possible to improve the coal combustion efficiency. By analyzing the historical data, it was found that the dampers opening corresponding to the 2nd, 4th, 9th, 12th, 14th, and 16th operating variables fluctuated very little, so the constraint was set to limit their optimization range to within 1% above and below the current sample value.

The boiler combustion optimization model is as follows:

$$\begin{cases} \min & \begin{cases} f_1 = \text{GBDT}_1(\mathbf{v}, \mathbf{x}) \\ f_2 = \text{GBDT}_2(\mathbf{v}, \mathbf{x}) \\ f_3 = \text{GBDT}_3(\mathbf{v}, \mathbf{x}) \\ f_4 = \text{GBDT}_4(\mathbf{v}, \mathbf{x}) \end{cases} \\ \text{s.t.} & \mathbf{x} \in \mathbb{S} \\ & g_i(\mathbf{x}) \leq 0, \quad i = 2, 4, 9, 12, 16, \end{cases} \quad (16)$$

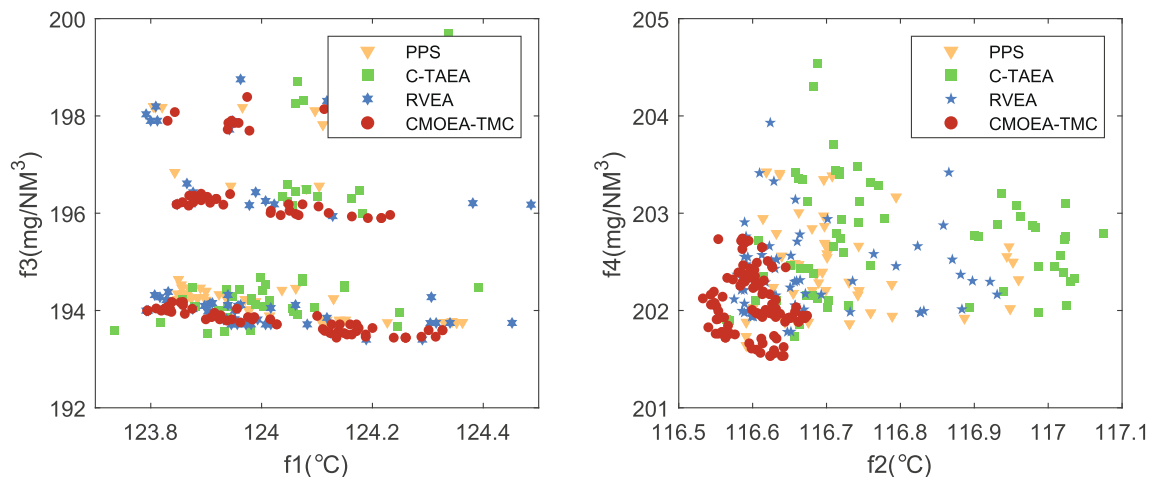
where  $\mathbf{v}$  is the non-adjustable operational variables, and  $\mathbf{x}$  is the adjustable operational variables.

Table 8 lists the HV values obtained for the six compared CMOEAs, where each CMOEA was run ten times independently. The population size was set to 100, maxFE was set to 10,000, and the reference point for calculating HV was set to [1,1,1,1]. From Table 8, it can be seen that CMOEA-TMC shows better overall performance than other CMOEAs in boiler combustion optimization, obtaining the best HV value. Figure 9 shows the results of some compared algorithms on this real-world problem, where four objective

**Table 8** Mean and standard deviation of the HV values obtained by TOP, PPS, CCMO, C-TAEA, RVEA, and CMOEA-TMC for ten independent runs on boiler combustion optimization

Problem	ToP	PPS	CCMO	C-TAEA	RVEA	CMOEA-TMC
Coal	1.0705e-1(1.44e-2)-	1.2200e-1(4.00e-3)-	1.2516e-1(6.17e-4)-	1.2593e-1(6.16e-3)≈	1.2394e-1(4.05e-3)-	<b>1.2899e-1(3.05e-3)</b>
	+ / - / ≈	0 / 1 / 0	0 / 1 / 0	0 / 0 / 1	0 / 1 / 0	

The best results are highlighted in bold

**Fig. 9** Final solutions achieved by compared algorithm on real-world boiler combustion optimization problem

functions are reverse normalized to their actual values. The solutions obtained by CMOEA-TMC converge better and have smaller objective function values, which is more evident for the objective functions  $f_2$  and  $f_4$ . Therefore, it can be concluded that the usefulness of CMOEA-TMC is also demonstrated in the real-world problem.

## Conclusion

In this paper, a novel algorithm named CMOEA-TMC is proposed for solving complex CMOPs. The combination of parallel evolution, coevolution, and staging strategies better balances the feasibility, convergence, and diversity of solution sets. In the first stage, the subpopulations in *mainPop* evolve independently to achieve fast convergence. When the subpopulations all converge to a small region, the algorithm switches to the second stage, which enhances the collaboration between *mainPop* and *helpPop* to maintain population diversity and promote convergence. When the number of function evaluations reaches 60% of the set maximum, the algorithm switches to the third stage to find the global optimal non-dominated solutions.

The broad effectiveness of CMOEA-TMC is verified by applying on the DOC, MW, and CF test suites as well as a real-world problem in comparison with five advanced algorithms, namely ToP, PPS, CCMO, RVEA, and C-TAEA.

Experimental results comparing with two-stage algorithms ToP and PPS show that the proposed three-stage strategy can help obtain solutions with better convergence. Experimental results comparing with multi-population algorithms CCMO and C-TAEA show that parallel evolution strategy and coevolution strategy proposed in this paper can well avoid populations from falling into local optimums. For CMOPs with complex constraints, such as the DOC test suit, the performance indicators of the proposed algorithm are significantly better than that of other algorithms, and the computation time is shorter. Therefore, CMOEA-TMC has good performance in CMOPs and can be applied to real-world problems. In addition, since good experimental results were obtained on the various test problems, the same population size specifications can be used for the new test problems. For test problems with more than three objectives, a larger number of population size and iterations are likely to be required, because more solutions are needed to cover the entire PF. As the efficiency of the offspring generation improves, the proposed algorithm will also have promising applications in many-objective optimization problems.

**Acknowledgements** This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61973269, in part by the National Key Research and Development Program of China under Grant 2019YFB1705502, in part by the Fundamental Research Funds for the Central Universities (Zhejiang University NGICS Platform), in part by the National Key Research and Development Program

of China under Grant 2022YFE0198900, and in part by the Ningbo Natural Science Foundation under Grant 2021J038.

## Declarations

**Conflict of interest** All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Bosman PA, Thierens D (2003) The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Trans Evol Comput* 7(2):174–188
- Cheng R, Jin Y, Olhofer M et al (2016) A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Trans Evol Comput* 20(5):773–791
- Coello CC (1999) An updated survey of evolutionary multiobjective optimization techniques: state of the art and future trends. In: Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406). IEEE, pp 3–13
- Coello CC (2006) Evolutionary multi-objective optimization: a historical view of the field. *IEEE Comput Intell Mag* 1(1):28–36
- Cui Z, Zhang J, Wu D et al (2020) Hybrid many-objective particle swarm optimization algorithm for green coal production problem. *Inf Sci* 518:256–271
- De Oliveira LL, Freitas AA, Tinós R (2018) Multi-objective genetic algorithms in the study of the genetic code's adaptability. *Inf Sci* 425:48–61
- Deb K, Agrawal RB et al (1995) Simulated binary crossover for continuous search space. *Complex Syst* 9(2):115–148
- Deb K, Goyal M et al (1996) A combined genetic adaptive search (GeneAS) for engineering design. *Comput Sci Inform* 26:30–45
- Deb K, Pratap A, Agarwal S et al (2002) A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6(2):182–197
- Deb K, Sindhya K, Hakanen J (2016) Multi-objective optimization. In: Decision sciences. CRC Press, Boca Raton, pp 161–200. <https://www.taylorfrancis.com/chapters/edit/10.1201/9781315183176-12/multi-objective-optimization-kalyanmoy-deb-karthik-sindhya-jussi-hakanen>
- Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: MHS'95. Proceedings of the sixth international symposium on micro machine and human science. IEEE, pp 39–43
- Fan Z, Li W, Cai X et al (2016) Angle-based constrained dominance principle in MOEA/D for constrained multi-objective optimization problems. In: 2016 IEEE congress on evolutionary computation (CEC), pp 460–467
- Fan Z, Li W, Cai X et al (2019) Push and pull search for solving constrained multi-objective optimization problems. *Swarm Evol Comput* 44:665–679
- Fan C, Wang J, Xiao L et al (2022) A coevolution algorithm based on two-staged strategy for constrained multi-objective problems. *Appl Intell*. <https://doi.org/10.1007/s10489-022-03421-7>
- Farzin H, Fotuhi-Firuzabad M, Moeini-Aghtaie M (2016) A stochastic multi-objective framework for optimal scheduling of energy storage systems in microgrids. *IEEE Trans Smart Grid* 8(1):117–127
- Geng H, Xu K, Zhang Y et al (2023) A classification tree and decomposition based multi-objective evolutionary algorithm with adaptive operator selection. *Complex Intell Syst* 9(1): 579–596
- Hollander M, Wolfe DA, Chicken E (2013) Nonparametric statistical methods. Wiley, Hoboken
- Li K, Chen R, Fu G et al (2018) Two-archive evolutionary algorithm for constrained multiobjective optimization. *IEEE Trans Evol Comput* 23(2):303–315
- Liang J, Ban X, Yu K et al (2022) A survey on evolutionary constrained multi-objective optimization. *IEEE Trans Evol Comput*. <https://doi.org/10.1109/TEVC.2022.3155533>
- Liu Z, Wang Y (2019) Handling constrained multiobjective optimization problems with constraints in both the decision and objective spaces. *IEEE Trans Evol Comput* 23(5):870–884
- Liu Z, Wang B, Tang K (2021) Handling constrained multiobjective optimization problems via bidirectional coevolution. *IEEE Trans Cybern*. <https://doi.org/10.1109/TCYB.2021.3056176>
- Long W, Dong H, Wang P et al (2023) A constrained multi-objective optimization algorithm using an efficient global diversity strategy. *Complex Intell Syst* 9(2): 1455–1478
- Ma Z, Wang Y (2019) Evolutionary constrained multiobjective optimization: test suite construction and performance comparisons. *IEEE Trans Evol Comput* 23(6):972–986
- Ming M, Wang R, Ishibuchi H et al (2021) A novel dual-stage dual-population evolutionary algorithm for constrained multi-objective optimization. *IEEE Trans Evol Comput*. <https://doi.org/10.1109/TEVC.2021.3131124>
- Miyakawa M, Takadama K, Sato H (2013) Two-stage non-dominated sorting and directed mating for solving problems with multi-objectives and constraints. In: Proceedings of the 15th annual conference on genetic and evolutionary computation, pp 647–654
- Saravanan R, Ramabalan S, Ebenezer NGR et al (2009) Evolutionary multi criteria design optimization of robot grippers. *Appl Soft Comput* 9(1):159–172
- Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Global Optim* 11(4):341–359
- Takahama T, Sakai S (2006) Constrained optimization by the  $\epsilon$  constrained differential evolution with gradient-based mutation and feasible elites. In: 2006 IEEE international conference on evolutionary computation, pp 1–8
- Tian Y, Cheng R, Zhang X et al (2017) PlatEMO: a MATLAB platform for evolutionary multi-objective optimization. *IEEE Comput Intell Mag* 12(4):73–87
- Tian Y, Zhang T, Xiao J et al (2020) A coevolutionary framework for constrained multiobjective optimization problems. *IEEE Trans Evol Comput* 25(1):102–116
- Tian Y, Zhang Y, Su Y et al (2021) Balancing objective optimization and constraint satisfaction in constrained evolutionary multiobjective optimization. *IEEE Trans Cybern* 52(9):9559–9572
- Vikhar PA (2016) Evolutionary algorithms: a critical review and its future prospects. In: 2016 International conference on global trends in signal processing, information computing and communication (ICGTSPICC). IEEE, pp 261–265

33. Wang Y, Wang BC, Li HX et al (2015) Incorporating objective function information into the feasibility rule for constrained evolutionary optimization. *IEEE Trans Cybern* 46(12):2938–2952
34. Xia Z, Liu Y, Lu J et al (2020) Penalty method for constrained distributed quaternion-variable optimization. *IEEE Trans Cybern* 51(11):5631–5636
35. Yang T, Zhang S, Li C (2021) A multi-objective hyper-heuristic algorithm based on adaptive epsilon-greedy selection. *Complex Intell Syst* 7(2):765–780
36. Yang F, Xu L, Chu X et al (2021) A new dominance relation based on convergence indicators and niching for many-objective optimization. *Appl Intell* 51(8):5525–5542
37. Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans Evol Comput* 11(6):712–731
38. Zhang Q, Zhou A, Zhao S et al (2008) Multiobjective optimization test instances for the CEC 2009 special session and competition. *University of Essex* 264:1–30
39. Zitzler E, Künzli S (2004) Indicator-based selection in multiobjective search. In: *International conference on parallel problem solving from nature*, pp 832–842
40. Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Trans Evol Comput* 3(4):257–271
41. Zitzler E, Laumanns M, Thiele L (2001) SPEA2: improving the strength Pareto evolutionary algorithm. *TIK-report* 103

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.