



A collaborative competition multitasking framework for constrained multi-objective optimization

Xinyu Feng^{a,b} , Qianlong Dang^{a,b,*}, Xiaochuan Gao^{a,b} , Yanghui Wu^{a,b}, Lifei Zheng^{a,b}

^a College of Science, Northwest A & F University, Yangling 712100, China

^b Joint Laboratory of Algorithm and Simulation for Low Altitude Aircraft, Northwest A & F University, Yangling 712100, China

HIGHLIGHTS

- A CMOEA based on tri-stage collaborative competition multitasking (TCCMT) is proposed.
- During the process of evolution, there is always an optimal auxiliary task to help the main task evolve, which greatly improves the utilization of computing resources.
- The auxiliary tasks use the constrained adaptive regression strategy and double angle enhancement strategy.

ARTICLE INFO

Keywords:

Collaborative competition multitasking
Constrained adaptive regression
Double angle constraint relaxation
Constrained multi-objective optimization

ABSTRACT

Constrained multi-objective optimization problems (CMOPs) are common in the real world. Constrained multi-objective evolutionary algorithms (CMOEAs) based on evolutionary multi-tasking show excellent performance in solving CMOPs. However, not all tasks can find useful information during the process of evolution, which inevitably results in a waste of computing resources. In this paper, a CMOEA based on collaborative competition multitasking (TCCMT) is proposed, in which two auxiliary tasks are constructed to co-evolve with the main task in a collaborative competition manner. During the process of evolution, only the dominant auxiliary task is selected to help the main task evolve, which reduces the resource consumption to evolve the invalid tasks. Meanwhile, the evolutionary process is divided into three stages in order to balance exploration and exploitation. The auxiliary tasks customize the constrained adaptive regression strategy and double angle enhancement strategy respectively to improve the ability to solve different problems. Compared with the nine most advanced CMOEAs on 33 benchmark problems and 7 real-world engineering problems, the Friedman test results show that TCCMT achieves the best rank on all test problems and exhibits a statistically significant difference.

1. Introduction

Constrained multi-objective optimization is an important research direction in the field of evolutionary computation, and has many practical applications [1]. Taking the power distribution system planning problem [2] as an example, decision makers not only hope to minimize the total cost of construction and operation, but also need to minimize the loss of power outages caused by faults. However, the two goals are essentially conflicting: the pursuit of high reliability usually requires more capital to build robust facilities. Meanwhile, any power grid design schemes must ensure that the voltage of users and the load of lines and substations are stable within a safe range. All the power grids need to maintain a specific structure to guarantee the safety of the system. This problem

of simultaneously optimizing multiple conflicting objectives and strictly satisfying constraints is called constrained multi-objective optimization problem (CMOP). In general, a CMOP can be mathematically described as follows [3]:

Minimize $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x}))$,

$$\text{s.t.} \quad \begin{cases} \mathbf{x} \in \mathcal{D} \\ g_i(\mathbf{x}) \leq 0, i = 1, \dots, p \\ h_j(\mathbf{x}) = 0, j = p + 1, \dots, q \end{cases}, \quad (1)$$

where $\mathbf{F}(\mathbf{x})$ refers to the set of objective functions to be optimized, which have m objective functions. $\mathbf{x}$ is a decision vector with n -dimensions in

* Corresponding author at: College of Science, Northwest A & F University, Yangling 712100, China.

Email addresses: fengxinyu@nwfau.edu.cn (X. Feng), xidianqldang@163.com (Q. Dang), gaoxiaochuan@nwfau.edu.cn (X. Gao), sxxawyh08@nwsuaf.edu.cn (Y. Wu), zhenglifei@nwfau.edu.cn (L. Zheng).

the decision space $\mathcal{D}$. $g_i(\mathbf{x})$ and $h_j(\mathbf{x})$ represent i -th inequality constraint and j -th equality constraint, respectively.

The k -th constraint violation degree of $\mathbf{x}$ is calculated as follows:

$$CV_k(\mathbf{x}) = \begin{cases} \max(0, g_k(\mathbf{x})), & k = 1, \dots, p \\ \max(0, |h_k(\mathbf{x})| - \delta), & k = p + 1, \dots, q \end{cases}, \quad (2)$$

where δ is a very small positive number used to relax equality constraints into inequality constraints. The overall constraint violation (CV) degree of $\mathbf{x}$ is calculated as follows:

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

where $\mathbf{x}$ is considered to be a feasible solution when $CV(\mathbf{x}) = 0$.

Different from multi-objective optimization problems (MOPs), not only the conflict between objectives but also constraint satisfaction needs to be considered when solving CMOPs [4]. Finding the balance between multiple objectives that cannot be met by a single solution during optimization. Therefore, the key is to find a set of feasible Pareto optimal solutions, where the feasible Pareto optimal solution mapped to the objective space is called the feasible Pareto front or the constrained Pareto front (CPF) [5–7]. The traditional optimization method is easy to fall into local optimum, and it is difficult to converge when encountering complex problems. Evolutionary algorithms are suitable for solving various optimization problems due to their population-based search mechanism and independence from gradient information, which can output a set of high-quality solutions in one iteration [8–10]. Therefore, researchers have proposed many types of constrained multi-objective evolutionary algorithms (CMOEAs).

In order to deal with the conflict between constraints and objectives, researchers have proposed various constraint handling techniques (CHTs), such as penalty function method [11,12], multi-objective method [13], constraint dominance principle (CDP) [14], and random sorting method (SR) [15]. However, researchers have gradually found that a single CHT cannot balance the feasibility, convergence, and diversity of the solution. Specifically, if the algorithm pays too much attention to feasibility, it may stay in the current feasible region and fall into local optima. If the algorithm prefers more diversity, the number of infeasible solutions increases, which may lead to difficulty in convergence. Therefore, some new enhanced CHTs are applied to CMOEAs, such as multi-stage based CMOEAs [16,17], hybrid methods based CMOEAs [18], and multiple CHTs based CMOEAs [19,20].

Although the method based on enhanced CHT has made some achievements in the performance of the algorithm, it is still powerless in the face of complex problems. In recent years, a number of CMOEAs based on evolutionary multitasking (EMT) have been proposed. The concept of EMT was first proposed in [21], and multiple different tasks are optimized simultaneously. Each task corresponds to a population, and the population evolves independently. Through knowledge transfer between tasks, the target task achieves optimal performance. EMT enables a more comprehensive exploration of the search space using knowledge transfer between tasks, reducing the risk of falling into local optima. It is more flexible when solving different problems. After these advantages of EMT were discovered, researchers constructed a series of auxiliary populations to assist the main population. These auxiliary populations have different purposes from the main population. A typical algorithm is EMCMO [21], which does not consider constraints in its auxiliary population and helps improve the convergence performance of the main population. In addition, the auxiliary population of MTCMO [22] uses constraint relaxation to utilize high-quality infeasible solutions and improve the diversity of the main population.

However, EMT will inevitably have some problems. A single auxiliary population can not cope with all problems, while a large number of auxiliary populations will waste computing resources, resulting in

reduced efficiency. Although some methods have been designed to terminate the auxiliary population that does not help the main population much in the later stage of evolution [23], additional parameters need to be introduced. In response to this challenge, the concept of competitive multitasking (CMT) was first proposed in [24]. Unlike EMT, all tasks in CMT are equal, and their objective functions have the same performance indicator. The target of each task is comparable, so all tasks in CMT can form a competitive relationship. Thus, it is reasonable to introduce the concept of CMT into the auxiliary population of EMT to establish the priority of the auxiliary population and make reasonable resource allocation. Chu *et al.* [25] have introduced CMT into solving CMOPs and applied it to a decomposition-based framework to design a new CMOEA. This shows that using CMT to solve CMOPs is promising. However, there are some shortcomings in their designed approach, where all tasks in the evolutionary process are comparable and computational resources are allocated only for the selected main task. Therefore, it is necessary to design a more effective framework to apply CMT to CMOEAs.

Based on the above considerations, a CMOEA based on tri-stage collaborative competition multitasking (TCCMT) is proposed. The main contributions are as follows:

1. **Application of the TCCMT framework:** In the proposed TCCMT, the main task is used to solve the original CMOPs, and the two auxiliary tasks compete in a collaborative way to assist the main task. During the evolution, auxiliary tasks only select the dominant one to co-evolve with the main task and guide the evolution of each other through parent mixing. The evolution process is divided into three stages to implement different strategies. The first two stages are used to determine the competitive relationship between the auxiliary tasks, which can balance the exploration and exploitation. The strategies of auxiliary tasks switch adaptively to achieve better assistance in the third stage.
2. **Constrained adaptive regression task:** An improved unconstrained auxiliary task is constructed. Compared with the traditional unconstrained auxiliary task, this task can dynamically determine whether the unconstrained population considers constraints according to the ratio of feasible solutions in the later stage of evolution. In this way, when there are enough feasible solutions in the unconstrained population, the population is guided to explore more regions to search for potentially feasible regions. When there are few feasible solutions, constraints are considered to pull the population from the infeasible side back to explore and drive convergence.
3. **Double angle constraint relaxation task:** In order to enhance the exploration based on ϵ constraint relaxation technology, the angle-based dominance principle and angle-based environmental selection are designed for the constraint relaxation population. The superposition of the two angle information is conducive to selecting more high-quality infeasible solutions and enhancing global diversity. The angle-based dominance principle is accurate than the previous CDP, which can balance the objectives and constraints according to the angle between the two solutions. Angle-based environmental selection can select solutions with better diversity and provide effective information for the main population.
4. **Comparison of experimental results:** Compared with nine most advanced CMOEAs on 33 test problems on three test suites and 7 real-world engineering problems, the experimental results demonstrate the superior performance of the proposed TCCMT and the potential to solve practical problems.

The rest of this paper is structured as follows. Section 2 introduces the related work in the existing literature. Section 3 presents the motivation of this paper and describes the proposed TCCMT in detail. The

experimental results are shown in Section 4. Finally, Section 5 provides the conclusion and promising directions for future research.

2. Related work

2.1. CMOEAs based on constraint handling techniques

Researchers have designed many effective CHTs and applied them to CMOEAs to balance objectives and constraints, so as to achieve a balance between convergence, feasibility, and diversity [26]. The penalty function method constructs a penalty term based on the constraint violation degree, and adds it to the objective function to convert CMOPs into MOPs. The method is divided into static penalty function method [27], dynamic penalty function method [28], and adaptive penalty function method [29] according to the selection of parameters. The multi-objective method treats the objective function and constraints as objectives equally, thus avoiding the conflict between constraints and objectives. CDP is a typical CHT proposed by Deb *et al.* [30], and it separates constraints from objectives by defining dominance relationship. When comparing two individuals, constraints are given priority, followed by objectives. Random sorting method (SR) [15] is a probability-based CHT. It determines the priority of constraints and objectives according to the probability randomly generated by the function, which avoids the problem of premature convergence of the algorithm. CDP is improved in [31] to obtain the angle-based CDP (ACDP), which redefines the dominance relationship based on the ratio of angle information and feasible solutions. In [32], a dynamic parameter ϵ is introduced to enhance CDP. This method can treat infeasible solutions with constraint violation degree less than ϵ as feasible solutions, and the use of high-quality infeasible solutions can enhance diversity.

2.2. CMOEAs based on evolutionary multitasking

Qiao *et al.* [21] first tried to use EMT to solve CMOPs. In EMCMO, two related tasks are evolved simultaneously: the main task is for the original CMOPs, and the auxiliary task is for the MOP. An efficient knowledge transfer method is designed between the two tasks to improve the efficiency of the task. Similar to EMCMO, URCMO designed by Liang *et al.* [33] constructed two populations to approach the constrained Pareto front (CPF) and unconstrained Pareto front (UPF), respectively. Then, different evolution strategies are determined according to the learned relationship between CPF and UPF. Tian *et al.* [34] designed a weak co-evolutionary paradigm CCMO to alleviate the problem of the negative knowledge transfer in the later evolution. In [35], a bidirectional co-evolutionary CMOEAs (BiCo) was proposed. The main population and the auxiliary population approach the CPF from the feasible side and the infeasible side respectively to improve the performance of the algorithm. In addition, they design an angle-based selection strategy to retain good infeasible solutions and enhance diversity. To better utilize the valuable infeasible solutions, Qiao *et al.* [22] constructed a dynamic auxiliary task in MTCMO where the constraint boundary will gradually shrink to assist the main task. Subsequently, they proposed a more effective search algorithm and embedded it into MTCMO [36] to obtain IMTCMO. In the cDPEA proposed in [37], the auxiliary population explores the region between UPF and CPF, and uses the CHT of the adaptive penalty function to find promising infeasible solutions for the main population. In the DPCPRA proposed by Qiao *et al.* [38], a dynamic constraint handling mechanism is designed to enhance the exploration ability by gradually increasing the constraints to take advantage of the infeasible solutions. In addition, the dynamic resource allocation scheme can allocate computing resources for the main population and the auxiliary population. To improve the diversity of population, Wang *et al.* [39] proposed a tri-stage framework for diversity enhancement. At different stages of evolution process, the angle-domination strategy and the minimum neighborhood-based domination strategy are applied to retain the population with good convergence and diversity, respectively.

Due to the uncertainty of the positional relationship between CPF and UPF, the cooperation of the two populations in dealing with complex CMOPs has been unable to meet the needs of the algorithm. Based on the advantages of EMT, researchers have developed a series of CMOEAs with multiple auxiliary populations. Ming *et al.* [40] proposed a co-evolutionary framework based on three populations. One auxiliary population does not consider constraints, and the other auxiliary population uses constraint relaxation techniques. Moreover, the main population and the auxiliary population co-evolve in a weak co-evolution relationship. Qiao *et al.* [23] set up two auxiliary populations in CMEGL. The global auxiliary population and the local auxiliary population are used to enhance the global and local diversity, respectively. In addition, they design a global auxiliary population adaptive stop update strategy to stop using UPF in the later stage of evolution, which saves computing resources. Zhao *et al.* [41] presented a multi-population CMOEA based on knowledge transfer (C-MTEA). In the environmental selection stage, the sharing strategy among the populations is used to generate offspring, which achieves more effective knowledge transfer. Liu *et al.* [42] introduced an adaptive CEMT framework (ACEMT), which designs two adaptive auxiliary tasks. One of the auxiliary tasks dynamically reduces the constraint boundary, and the other auxiliary task adaptively selects constraints for the population, which improves the efficiency of solving CMOPs. Zhong *et al.* [43] divided the evolutionary stage into forward and backward stages in CEMTFB, each stage using different auxiliary populations. The auxiliary population in the forward stage ignores the constraints to promote the convergence of the population, and in the backward stage uses the reverse transportation strategy to approach the CPF from two complementary directions with the main population. Ming *et al.* [19] proposed a three-task framework (CMOEMT) based on different CHTs in order to balance feasibility, convergence, and diversity. The main population and the auxiliary populations adopt the constraint dominance principle, the Pareto dominance principle, and the constraint relaxation principle, respectively. Yu *et al.* [44] developed an evolutionary multitasking method, which uses the constraint subset to construct the auxiliary tasks. The tasks search their own CPF and then help the main population search for CPF. Qiao *et al.* [45] used the promising single constraint to construct auxiliary tasks to help solve CMOP. The most promising auxiliary task is preserved based on the proposed auxiliary task priority method to save computing resources.

2.3. CMOEAs based on competitive multitasking

In the optimization process of EMT, it is necessary to find the optimal solution for each task, but tasks can not always provide useful information at all stages of evolution. Therefore, EMT will inevitably waste computing resources. CMT can dynamically select the optimal task for optimization during the evolution process, which solves this problem well. Li *et al.* [24] proposed the competitive multitasking optimization problems (CMTOPs) for the first time. Compared with EMT, the objectives of all its tasks are comparable, and its optimal solution is the best in the optimal solutions of each task. At the same time, they developed an algorithm with online resource allocation strategy and adaptive information transmission mechanism to solve CMTOPs. Chu *et al.* [25] presented a competitive multitasking framework and integrated it into a decomposition-based framework (MOEAD-CMT). In addition, they designed a reward mechanism to allocate computing resources reasonably.

3. Proposed TCCMT

3.1. Motivation

The existing CMOEAs usually use different constraint processing techniques [16,19] or construct different auxiliary populations [21,22] to assist the main population in order to achieve a balance between convergence, feasibility, and diversity when solving CMOPs. Although these

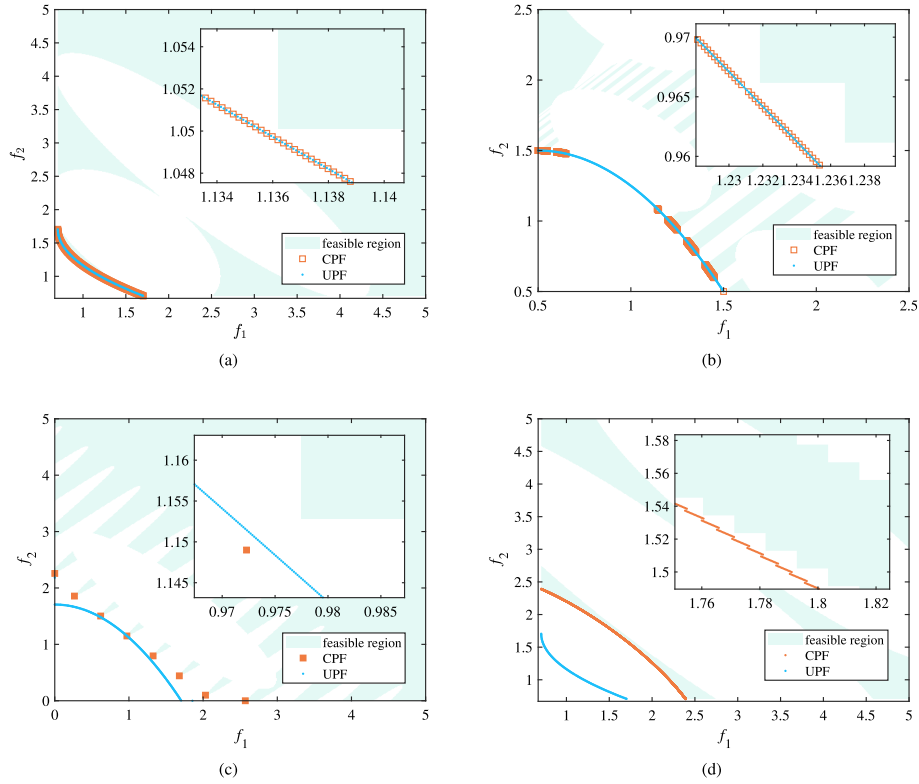


Fig. 1. Four positional relationships between CPF and UPF. Light green is the feasible region, and orange and blue lines represent CPF and UPF, respectively.

methods alleviate the drawback that it is difficult to achieve balance in a single population, multitasking parallelism will cause some unnecessary waste of computing resources. The dynamic resource allocation strategy [38] can reasonably allocate limited computing resources to each population according to performance feedback, but this strategy relies too much on prior knowledge and parameter settings. All the tasks set in MOEAD-CMT [25] are compared, and none of the tasks is used as the main target of algorithm evolution. At this time, there is only one task playing a role, and it is relatively difficult to maintain convergence, feasibility, and diversity. Therefore, a more comprehensive framework needs to be designed to integrate the advantages of EMT and CMT. In addition, the construction of auxiliary tasks is particularly important. Excellent auxiliary tasks can help the main population to quickly cross the infeasible regions, and can search for potential feasible regions to maintain diversity. Traditional auxiliary tasks often have an unconstrained auxiliary task, which can help the population converge quickly. However, if the unconstrained population searches for the UPF and the CPF and the UPF are far away from each other, the unconstrained auxiliary task will lose its effect and even have a negative impact [23,33]. As shown in Fig. 1, there are four cases of CPF and UPF in general: CPF and UPF are the same, CPF is part of UPF, CPF and UPF partially overlap and CPF, and UPF do not overlap, which correspond to (a), (b), (c), and (d), respectively. The relationship between CPF and UPF in the test problem is diverse, and the cases where CPF and UPF overlap are even rarer. This requires designing more effective unconstrained auxiliary tasks to solve this problem. There is a commonly used auxiliary task using constraint relaxation technology [22,37,46], which uses some high-quality infeasible solutions by relaxing the constraint boundary to maintain diversity. However, many of these auxiliary tasks directly adopt the same constraint handling technology and environmental selection strategy as the main task. This approach is obviously unreasonable. Because some strategies of the main population prefer feasibility, which leads to the deletion of some solutions with good diversity during the

evolution of the auxiliary population [39]. Therefore, some strategies suitable for auxiliary populations need to be introduced to maintain diversity.

In order to cope with these challenges, this paper proposes a tri-stage collaborative competition multitasking framework (TCCMT). Unlike EMT, only the dominant one of the auxiliary tasks co-evolves with the main task during the evolution. In different evolutionary stages, auxiliary task that plays the specific role assists the main task, which avoids the waste of resources caused by ineffective auxiliary tasks. The dominant auxiliary task is determined based on its contribution to the main population, thus avoiding the parameter setting problem in dynamic resource allocation. Meanwhile, in order to prevent the defect that all tasks in MOEAD-CMT maintain a competitive relationship, the two auxiliary populations in TCCMT guide the evolution of each other in the way of parent mixing during the evolution process, which achieves a collaborative effect at the same time. In addition, the two auxiliary populations customize the constrained adaptive regression strategy and double angle enhancement strategy, respectively, in order to solve the limitations of traditional auxiliary tasks. The constraint adaptive regression strategy can dynamically control the constraint state of the unconstrained auxiliary population by the proportion of feasible solutions in the population. Compared with the CMOEAs based on unconstrained population assistance, such as CCMO and CMEGL, even if the auxiliary population explores UPF, it can still guide the population to reverse explore from the infeasible side. This not only reduces the risk of negative knowledge transfer, but also enhances the exploration ability of the auxiliary population. In order to effectively utilize the high-quality infeasible solutions retained by the constraint relaxation technique, the double angle enhancement strategy adopts the angle-based dominance principle and angle-based environmental selection. Based on the traditional constraint relaxation technique to identify high-quality infeasible solutions, the proposed strategy can better preserve these solutions in the evolution process, which achieves the maintenance of diversity. Furthermore, the

Algorithm 1 General framework of TCCMT.

Input: N (population size), $MaxFE$ (maximum number of function evaluations), CP (parameter controlling the degree of parent cooperation), η (parameter controlling the evolutionary stage)

Output: P_1 (main population)

```

1: Initialize the populations  $P_1, HP_1, HP_2$ ;
2:  $FE = 3N$ ;
3: while  $FE < MaxFE$  do
4:   Calculate fitness of  $P_1$  on the CMOPs;
5:   Calculate fitness of  $HP_1$  using Algorithm 3;
6:   Calculate fitness of  $HP_2$  using Algorithm 4;
7:   if  $FE < \eta * MaxFE$  then
8:      $Sp_1 = Sp_2 = 0.5$ ; /* The two auxiliary populations have the
       same selection probability. */
9:   else
10:    Update the selection probabilities using Eqs. (4) and (5); /* The
      contribution value is mapped to the selection probability. */
11:   end if
12:   Use roulette wheel method to determine  $t = 1$  or  $2$  based on
       $Sp_1$  and  $Sp_2$ ; /* The auxiliary population with high selection
      probability is selected to execute the offspring generation. */
13:    $O_1 \leftarrow$  Generate  $N/2$  offspring by  $P_1$ ;
14:   for  $i = 1$  to  $N/2$  do
15:     if  $rand < CP$  then
16:        $Matingpool \leftarrow$  Select parents from  $HP_1$ ; /* The  $i$ -th parent
        comes from the dominant auxiliary population. */
17:     else
18:        $Matingpool \leftarrow$  Select parents from  $HP_i$  and the other  $HP$ ; /*
        The  $i$ -th parent comes from the two auxiliary populations. */
19:     end if
20:   end for
21:    $O_2 \leftarrow$  Generate  $N/2$  offspring from  $Matingpool$ ;
22:    $P_1 \leftarrow P_1 \cup O_1 \cup O_2$ ;
23:    $P_1 \leftarrow$  Select  $N$  individuals from  $P_1$  based on the strategy of SPEA2;

24: for  $j = 1$  to  $2$  do
25:    $HP_j \leftarrow HP_j \cup O_1 \cup O_2$ ;
26:   if  $j == 1$  then
27:      $HP_j \leftarrow$  Select  $N$  individuals from  $HP_j$  based on the strategy
      of unconstrained environmental selection;
28:   else
29:      $HP_j \leftarrow$  Select  $N$  individuals from  $HP_j$  based on the strategy
      of angle environmental selection;
30:   end if
31: end for
32:  $P_1 \leftarrow$  Select  $N$  individuals from  $P_1 \cup O_2$  using Algorithm 2;
33:  $FE = FE + N$ ;
34: end while

```

algorithm is divided into three stages to implement different strategies, which can adapt to the exploration and exploitation needs of different periods. At different stages, the strategy of the auxiliary task is switched, which enhances the ability of the algorithm to deal with changing problems.

3.2. General framework of TCCMT

Algorithm 1 presents the general framework of TCCMT. At the same time, the process of TCCMT is visually shown in the Fig. 2. Firstly, three populations P_1 , HP_1 , and HP_2 are initialized, which correspond to the main task, the constraint adaptive regression task, and the double angle constraint relaxation task, respectively (line 1). Three function evaluations are used for initialization, so the number of function evaluations (FE) is initialized to $3N$ (line 2). In each subsequent iteration, N offspring are generated. That means N function evaluations are consumed. When the current evaluation number of the population reaches the maximum function evaluation number ($MaxFE$), the cycle process ends. The fitness of each individual in the population is calculated at the beginning of each iteration (line 4). The fitness calculation methods of the two auxiliary populations are presented in Algorithms 3 and 4, respectively (lines 5–6). The first stage of the evolutionary process is conducted, when $FE < \eta * MaxFE$. Two auxiliary populations have a selection probability of 0.5, and the chances of selection for the dominant auxiliary task and the main population co-evolving are equal (line 8). Otherwise, it goes to the second stage, where the two auxiliary tasks determine the dominant auxiliary task based on how much it contributes to the main population. The selection probability is calculated as follows:

$$Sp'_t = \frac{Sp_{\min}}{2} + (1 - Sp_{\min}) \frac{C_t}{\sum_{i=1}^2 C_i}, \quad (4)$$

$$Sp_t = \frac{Sp'_t}{\sum_{i=1}^2 Sp'_i}, \quad (5)$$

where $Sp_{\min}$ is the minimum selection probability of auxiliary tasks, and is set to 0.1. It is used to prevent the case that Sp_t is calculated as 0. C_t represents the contribution generated in the previous iteration process. Eq. (5) performs a normalization operation to control the selection probability between 0 and 1 (line 10). At the same time, the roulette selection method is used to fairly select the dominant auxiliary tasks according to the selection probability (line 12). The auxiliary population with high selection probability is possible to be selected. After that, the main population uses genetic operators to generate $N/2$ offspring O_1 (line 13). The dominant auxiliary population first generates a mating pool of size $N/2$ based on the degree of parent collaboration (CP). The source of parents in the mating pool is determined by CP , where parents can be selected from the dominant population with some probability from another auxiliary population (lines 14–20). That is to say,

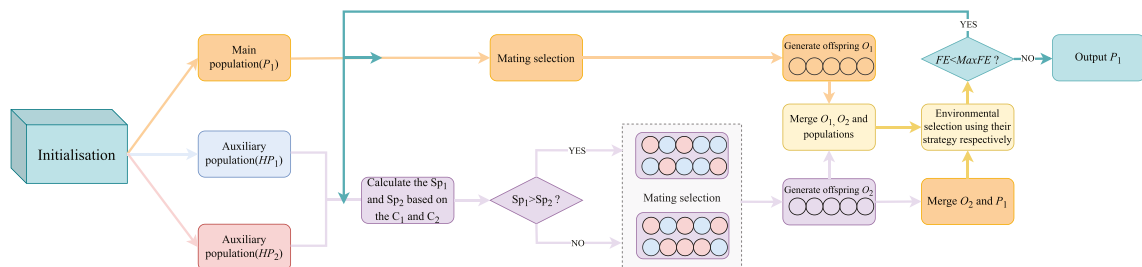


Fig. 2. An overview flowchart of TCCMT.

Algorithm 2 Archive update.

Input: P_1 (main population), O_2 (offspring generated by the dominant auxiliary population), N (population size)

Output: P_1 (updated main population), M_t (number of updated offspring)

```

1:  $P \leftarrow P_1 \cup O_2$ ;
2:  $F_i \leftarrow$  Calculate the fitness for each solution in  $P$ ;
3:  $S \leftarrow$  Initialize the candidate solution set where  $F_i < 1$ ;
4:  $L \leftarrow$  Count the number of candidate solutions;
5: if  $L < N$  then
6:   Sort all solutions by  $F_i$ ;
7:    $P_1 \leftarrow$  Select top  $N$  solutions as next population;
8: else if  $L > N$  then
9:    $K \leftarrow$  Count  $L - N$  solutions to be removed;
10:   $D_i \leftarrow$  Compute the Euclidean distance for each solution in  $S$ ;
11:  while  $K > 0$  do
12:    Sort all solutions by  $D_i$ ;
13:    Remove the most crowded solution;
14:     $K \leftarrow K - 1$ ;
15:    Update  $S$  and  $D_i$ ;
16:  end while
17:   $P_1 \leftarrow S$ ;
18: end if
19:  $M_t \leftarrow$  Count the solutions retained by  $O_2$  in  $P_1$ ;

```

$CP * N/2$ parents in the mating pool are from the dominant auxiliary population. Parents in the mating pool similarly generate $N/2$ offspring O_2 using genetic operators (line 21). The offspring generated by the two populations are merged with P_1 to form a new P_1 (line 22), and the environmental selection strategy of SPEA2 [47] is used to select N optimal individuals to be retained to the next generation (line 23). Similarly, all the offspring are merged with HP_1 and HP_2 to form new HP_1 and HP_2 , respectively (line 25). The two auxiliary populations select the optimal N individuals in the population according to the unconstrained environmental selection strategy and the angle-based environmental strategy to retain them to the next generation (lines 26–30). Subsequently, P_1 and O_2 are merged to select the N optimal individuals again using archive update strategy in Algorithm 2 to calculate the contribution of the offspring produced by the dominant auxiliary population to the main population (lines 32–33). Eventually, at the end of the iteration of the algorithm, the output main population P_1 will be used as the final solution set.

The contribution of the auxiliary population is calculated by the Algorithm 2. Firstly, the main population P_1 and the offspring generated by the auxiliary population O_2 are merged into archive P (line 1). Then, the fitness F_i of all solutions is calculated, and solutions with $F_i < 1$ are considered as candidate solution set S (lines 2–3). If the number of candidate solutions L is less than the population size N , the solutions in P are sorted according to F_i and the optimal N are selected as the next population (lines 4–7). While $L > N$, the crowding degree D_i is calculated according to the Euclidean distance for selecting (lines 9–10). All solutions in S are sorted according to D_i , before each iteration. The most crowded solution is filtered out, and S and D_i are updated until the number of S satisfies the population size (lines 11–16). The final S is regarded as the next population (line 17). Finally, the number of solutions retained by O_2 in the next population is counted as M_t , and the contribution value of the auxiliary population is calculated according to Eq. (6) (line 19):

$$C_t = C_t + M_t / N, \quad (6)$$

where M_t represents the number of individuals retained by O_2 after the environmental selection.

3.3. Constrained adaptive regression task

Traditional unconstrained auxiliary tasks can be described as follows:

$$\text{Minimize } \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})). \quad (7)$$

The task only involves the optimization of the objective and will directly converge to the UPF. It has two main drawbacks. Firstly, the positional relationship between UPF and CPF is diverse. In most cases, the search for UPF in the late evolution can not provide effective information for the main population. In addition, the solution searched by the task without considering constraints is obviously invalid. Secondly, unconstrained auxiliary tasks will converge faster than the main task since constraints are not considered. This makes it possible for the unconstrained population to miss some small feasible regions during the search for feasible solutions, causing local optima of the algorithm.

In order to avoid these effects, the constraint adaptive regression task is set as follows:

$$\begin{aligned} &\text{Minimize } \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})), \\ &\text{s.t. } \begin{cases} d * g_i(\mathbf{x}) \leq 0, i = 1, \dots, p \\ d * h_j(\mathbf{x}) = 0, j = p + 1, \dots, q \end{cases}, \end{aligned} \quad (8)$$

where d is dynamically determined by the proportion of feasible solutions and γ , which is set to 0 or 1. The feasible solutions are determined by the Eq. (3). A solution $\mathbf{x}$ is considered to be a feasible solution if and only if $CV(\mathbf{x}) = 0$. The proportion of feasible solutions is strictly calculated. Therefore, the constraint adaptive regression strategy is effective across all problem types, as the test problems all have the feasible regions. The specific implementation will be carried out through the calculation of fitness. Algorithm 3 presents the pseudo-code for calculating the fitness of the constrained adaptive regression population. Firstly, the constraint violation degree of each individual in the population is calculated (lines 1–2). Then, the number of feasible solutions in the current population is counted to obtain the proportion of feasible solutions (lines 3–4). The parameter γ is used to control the third stage of evolution. When $FE < \gamma * MaxFE$, d is set to 0. The dominance relationship between the two solutions is only determined by the objective value, and the constraint is ignored (line 6). When the evolutionary stage reaches the third stage, the dominance relationship between solutions is dynamically determined by the proportion of feasible solutions in the current population. When the proportion of the feasible solution is small, d is set to 1. Thus, the CDP is used to determine the dominance relationship

Algorithm 3 Calculate the fitness of HP_1 .

Input: N (population size), HP_1 (auxiliary population), γ (parameter controlling the evolutionary stage)

Output: $Fitness_2$

```

1:  $CV_i \leftarrow$  Calculate constraint violation degree for each individual;
2:  $CV_i = \max(0, CV_i)$ ;
3:  $num \leftarrow$  Count individuals with  $CV == 0$ ;
4:  $p_f = num / N$ ;
5: if  $rand < p_f$  or  $FE < \gamma * MaxFE$  then
6:   The dominance relationship between the two solutions is determined according to the objective value;
7: else
8:   The dominance relationship between the two solutions is determined according to the CDP;
9: end if
10:  $R_i \leftarrow$  Calculate the number of other solutions dominated by each solution;
11:  $D_i \leftarrow$  Calculate the crowding distances of each solution;
12:  $Fitness_2 = R_i + D_i$ ;

```

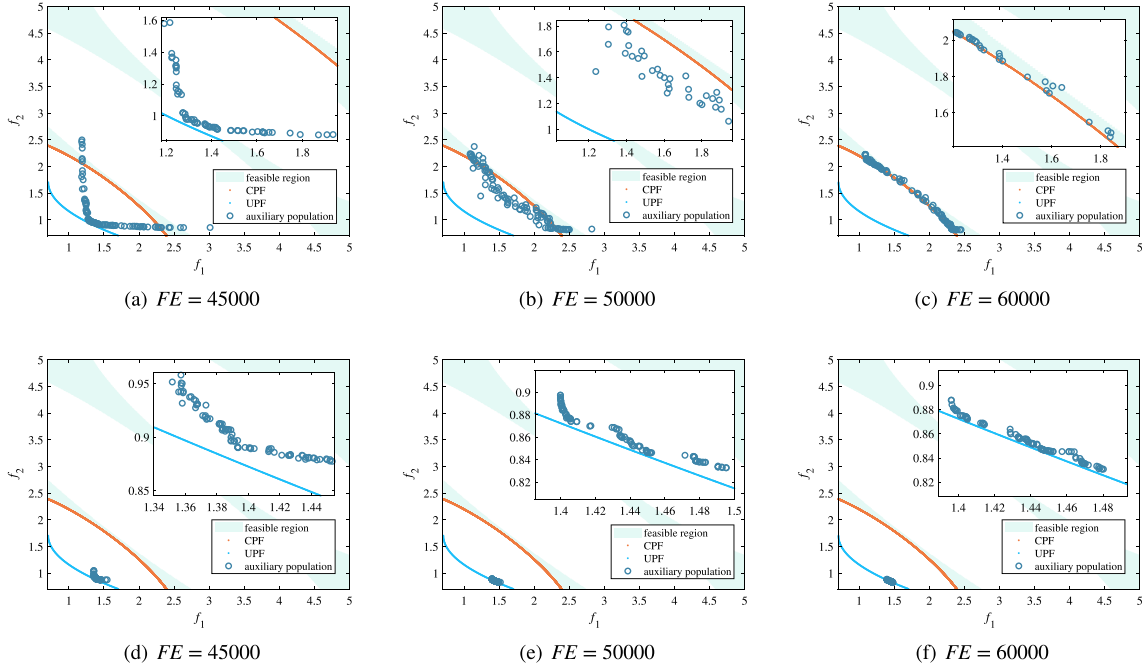


Fig. 3. The distribution of unconstrained auxiliary population in the evolutionary process. The above three images correspond to TCCMT, and the following three represent EMCMO.

(line 8). When the proportion of feasible solutions is high, the dominance relationship is the same as the first two stages. Finally, the fitness is obtained by calculating the number of other solutions dominated by each solution and the crowding distance (lines 10–12). Through such adaptive adjustment in the later stage, the use value of the auxiliary population can be effectively judged, and the strategy can be adjusted in time. In this way, the auxiliary population explores UPF. When the distance between CPF and UPF is far or the overlap is low, the auxiliary population can still explore back from the infeasible side again. Fig. 3 shows the distribution of the unconstrained auxiliary populations of TCCMT and EMCMO in different evolutionary periods on the LIRCMOP7 test problem. It can be seen that this traditional unconstrained auxiliary population of EMCMO will stagnate in the region after exploring to UPF. However, the proposed constrained adaptive regression population will explore back according to the current state of the population after arriving at UPF. This can enhance the exploration ability of the auxiliary population and provide valid information for the main population to explore CPF. At the same time, due to the competitive relationship between the two auxiliary tasks under the framework of collaborative competition, the auxiliary population will not converge quickly so that it can fully explore the search space, thus avoiding the risk of premature convergence.

3.4. Double angle constraint relaxation task

The constraint relaxation auxiliary task can be described as follows:

$$\begin{aligned} &\text{Minimize } \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})), \\ &\text{s.t. } CV(\mathbf{x}) \leq \varepsilon_G, \end{aligned} \quad (9)$$

where ε_G is updated by Ref. [48]:

$$\varepsilon_G = \varepsilon_0 \cdot \left(1 - \frac{G}{G_{max}}\right)^{\frac{-\log(\varepsilon_0) - 6}{\log(1-\rho)}}, \quad (10)$$

where G denotes the current generation, G_{max} represents the maximum number of generations, ε_0 is the initial maximum constraint violation degree, and ρ is used to control the decreasing speed of ε_G , which is

set to 0.5. The auxiliary task adjusts the degree of constraint relaxation by dynamically adjusting the size of ε_G . When the constraint violation degree is less than ε_G , the solution can be regarded as a feasible solution. This allows the constrained relaxation population to explore a wider region and retain some valuable infeasible solutions. Most of these potentially valuable infeasible solutions are located around the feasible region, which can guide the main population to gradually converge to the feasible region. In addition, the retained infeasible solutions can continue to explore new regions in the search space which increases the diversity. Many researchers focus on how to control the change of ε_G , but ignore how to use these infeasible solutions. Their approach is often to use the same strategy as the main population to select the solution after searching for a promising infeasible solution. However, the strategy of the main population is mostly feasibility priority, which is bound to have a negative impact on the diversity of the population.

In order to alleviate this shortcoming, the proposed algorithm uses an angle-based dominance relationship in the auxiliary population with relaxed constraints to balance objectives and constraints. Similarly, the strategy is implemented in the calculation of fitness. The specific pseudocode is shown in Algorithm 4. Firstly, the values of ε_G and θ_G are updated using the Eqs. (10) and (11), respectively (lines 1–2):

$$\theta_G = \begin{cases} \theta_0 \left(1 + \frac{k}{G_{max}}\right)^\lambda, & 1 \leq k \leq G \\ \frac{\pi}{2}, & G < k \leq G_{max} \end{cases}, \quad (11)$$

where θ_0 is an initial threshold, $G = \gamma * G_{max}$, λ is initialized to $\frac{\ln(\pi/(2\theta_0))}{\ln(1+\gamma)}$ to ensure $\theta_G = \frac{\pi}{2}$, when the evolution stage goes to the third stage. The angle between each two solutions is calculated as follows [42] (line 3):

$$A(\mathbf{x}_1, \mathbf{x}_2, \mathbf{Z}^*) = \arccos \left(\frac{(\mathbf{F}(\mathbf{x}_1) - \mathbf{Z}^*)^T \cdot (\mathbf{F}(\mathbf{x}_2) - \mathbf{Z}^*)}{\|\mathbf{F}(\mathbf{x}_1) - \mathbf{Z}^*\| \cdot \|\mathbf{F}(\mathbf{x}_2) - \mathbf{Z}^*\|} \right), \quad (12)$$

where $\mathbf{Z}^* = (\mathbf{Z}_1^*, \mathbf{Z}_2^*, \dots, \mathbf{Z}_m^*)$ is an ideal point, and $\mathbf{Z}_i^* = \min\{f_i(\mathbf{x}) | \mathbf{x} \in D\}$. In the two-dimensional objective space, the angle between any two

Algorithm 4 Calculate the fitness of HP_2 .**Input:** N (population size), HP_2 (auxiliary population)**Output:** $Fitness_3$

```

1:  $\varepsilon_G \leftarrow$  Update using Eq. (10);
2:  $\theta_G \leftarrow$  Update using Eq. (11);
3:  $A \leftarrow$  Calculate the angle between two individuals using Eq. (12);
4:  $CV_i \leftarrow$  Calculate constraint violation degree for each individual;
5:  $CV_i = \max(\varepsilon_G, CV_i)$ ;
6: for  $i = 1$  to  $N - 1$  do
7:   for  $j = i + 1$  to  $N$  do
8:     if  $(CV_i = CV_j) == 0$  then
9:       The dominance relationship between the two solutions is
       determined according to the objective value;
10:    else
11:      if  $A < \theta_G$  then
12:        The dominance relationship between the two solutions is
        determined according to constraint violation degree;
13:      else
14:        The dominance relationship between the two solutions is
        determined according to the objective value;
15:      end if
16:    end if
17:  end for
18: end for
19:  $R_i \leftarrow$  Calculate the number of other solutions dominated by each
    solution;
20:  $D_i \leftarrow$  Calculate the crowding distances of each solution;
21:  $Fitness_3 = R_i + D_i$ ;

```

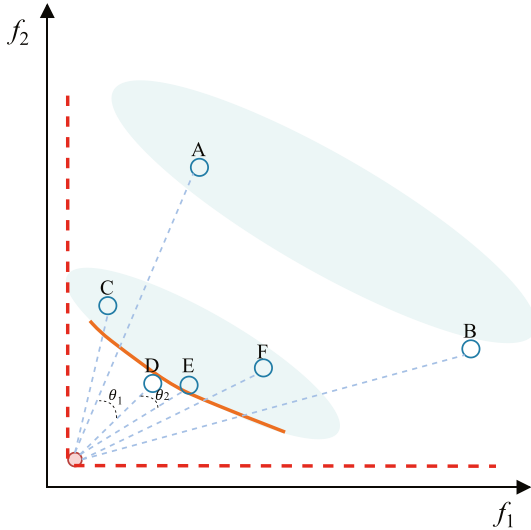


Fig. 4. Illustration of the double angle enhanced strategy. Light green is the feasible region, dark blue circles are the solutions, light red circle denotes the ideal point Z^* , and the orange line represents CPF.

solutions is calculated by Eq. (12). Similarly, the angle between any two solutions can be calculated in the three-dimensional or higher-dimensional objective space. Therefore, the angle-based dominance relationship can work in any case.

The value of ε_G will gradually decrease with the increase of evolutionary generations, making the degree of constraint relaxation smaller. The value of θ will gradually increase and be equal to $\frac{\pi}{2}$ after the third stage. Afterwards, the constraint violation degree is calculated for each individual and the solutions with constraint violation degree less than

ε_G are considered as feasible solutions (lines 4–5). For any two solutions in the population, if both are feasible solutions, the dominance relation is determined based on the objective value (line 9). Otherwise, the priority of the constraint and the objective is determined according to the angle between the two solutions. When the angle between the two solutions is less than θ , the dominance relationship between the two solutions is determined by the constraint violation degree (line 12). On the contrary, the dominance relationship is determined by the objective value (line 14). As shown in the Fig. 4, when the angle θ_2 between D and E is less than θ , D is dominated by E. But when the angle θ_1 between A and D is greater than θ , D dominates A. Since the angle between any two solutions is never greater than $\frac{\pi}{2}$, the angle-based dominance relationship is transformed into a CDP in the third stage. Therefore, in the later stage of evolution, the constrained relaxation population can provide sufficient convergence pressure for the main population. Finally, the number of other solutions dominated by each solution and the crowding distance are calculated to obtain the fitness (lines 19–21). In addition, we use an angle-based environmental selection strategy to update the constrained relaxation population during evolution. In the process of angle-based environmental selection, the angle between each two solutions is calculated. When selecting the individuals that need to be deleted, the algorithm will start from the two solutions of the minimum sharing angle to compare and delete the solutions with poor fitness. Finally, the algorithm will continue to find the next minimum angle and repeat the process until all individuals are selected. This strategy of combining two kinds of angle information effectively guarantees the diversity of solutions provided by the constrained relaxation auxiliary population.

3.5. Computational complexity analysis

The computational complexity of TCCMT mainly includes the mating selection, the offspring generation, the environmental selection in Algorithm 1 and the fitness calculation in Algorithms 3 and 4. The computational complexity of mating selection is generated when the main population and auxiliary population select the matingpool, and the runtime is $2 * O(N/2) = O(N)$. The part of offspring generation is that $N/2$ offspring are generated by the main population and the dominant auxiliary population, and the runtime is $2 * O(n * N/2) = O(nN)$. The environmental selection in TCCMT is mainly determined by the SPEA2 used. All the population perform environmental selection operation in each iteration. Thus, the runtime is $3 * O(mN^2)$. The fitness calculation in Algorithms 3 and 4 involves the Pareto dominance operation and constrained Pareto dominance operation, and their runtime are $O(mN^2)$ and $O((m+1)N^2)$, respectively. Therefore, the computational complexity of the proposed TCCMT is $O(N) + O(nN) + 3 * O(mN^2) + O(mN^2) + O((m+1)N^2) = O(mN^2)$, which is the same as many existing CMOEAs [19,49].

4. Experiment

4.1. Experiment setting

4.1.1. Compared algorithms

In order to verify the performance of TCCMT, nine excellent algorithms are selected for comparison: CCMO [34], cDPEA [37], BiCo [35], EMC MO [21], CMEGL [23], MTCMO [22], DEST [39], SC-EMT [44], and CSEMT [45]. CCMO, cDPEA, BiCo, EMC MO, and MTCMO all construct two populations to collaborate. The unconstrained auxiliary population and the main population in CCMO adopt a weak co-evolution paradigm. The auxiliary population in cDPEA applies the CHT of the adaptive penalty function to use the promising infeasible solutions between UPF and CPF to promote convergence. The two populations in BiCo approach the CPF from the feasible and infeasible sides of the search space, respectively. EMC MO uses an evolutionary multitasking framework to simultaneously evolve two populations, and designs a tentative method

between the two populations to discover useful knowledge for transfer. MTCMO constructs an auxiliary task whose constraint boundary will gradually shrink, so that it maintains a high correlation with the main task and continuously provides a supplementary evolutionary direction. CMEGL uses three populations to co-evolve, and two auxiliary populations are used to enhance global diversity and local diversity, respectively. Both SC_EMT and CSEMT construct multiple auxiliary tasks. SC_EMT develops a constraint subsets-based evolutionary multitasking method. Each auxiliary task has different constraint subsets. CSEMT designs an auxiliary task priority method to identify promising single constraint and use them to help solve CMOPs. DEST is a CMOEA based on the decomposition framework, which develops a tri-stage framework with three different evolutionary stages to enhance individual diversity.

These algorithms are excellent methods mentioned in the previous literature in Section 2. CCMO, cDPEA, BiCo, EMCMO and MTCMO are CMOEAs based on EMT with two populations. CMEGL, SC_EMT and CSEMT are CMOEAs based on EMT with multiple populations. By comparing with these algorithms based on the EMT framework, it can demonstrate the rational use of computing resources by the collaborative competition framework. CCMO and EMCMO are CMOEAs based on unconstrained population assistance, which verify the enhancement effect of the constrained adaptive regression strategy on population exploration ability. cDPEA, CMEGL and MTCMO use constraint relaxation techniques to exploit high-quality infeasible solutions, and DEST uses the diversity-enhanced tri-stage framework to improve diversity. They can compare with the proposed double angle constraint relaxation strategy to confirm the advantages in maintaining population diversity. In addition, the ablation experiment is carried out to prove the effectiveness of the proposed methods in Section 4.4.

4.1.2. Benchmark problems

Three typical test suites CF [50], LIRCMOP [51], and DASCOP [52] are selected to comprehensively test TCCMT. CF test suite has a total of 10 test problems. They all have the characteristics of large feasible region, but the CPF of most problems is located at the constraint boundary. This poses a great challenge to the convergence of the algorithm. In addition, CF8-10 contains three objective functions, making it more difficult to find CPF. LIRCMOP test suite contains 14 test problems. The constraint function in each test problem is jointly controlled by the distance function and the shape function. The distance function can adjust the difficulty of convergence, and the shape function is used to control the concave and convex shape of CPF. The superposition of shape and distance functions not only increases the complexity of the test problem, but also enhances its scalability, making the test problem more flexible. DASCOP test suite consists of 9 test problems with extensible objective functions, and the difficulty of each problem is controlled by three tuples. Each parameter of the tuple corresponds to a difficulty level, and constraints of multiple difficulty levels can be generated by combining different constraint functions. LIRCMOP and DASCOP test suites are easy to fall into local optima because of the difficulty of test problems and the large infeasible region. Therefore, these three test suites can deeply verify the performance of the proposed algorithm.

4.1.3. Experimental environment

The experiments are carried out on the PlatEMO v4.7 platform [53] in MATLAB (2022 b). The experimental running environment is Intel (R) Core (TM) i7-10700 CPU @ 2.90 GHz processor, 16 GB RAM, Windows 11 Professional operating system. For a fair comparison, the overall size (N) of all algorithms is set to 100, and the maximum number of evaluations ($MaxFE$) of the function is set to 80,000. We set the number of runs of each algorithm to 24 times. In addition, the parameters of all

algorithms are the same as those suggested in the original text. The parameters in the proposed algorithm are $\eta = 0.3$, $CP = 0.6$, $\gamma = 0.8$, and the specific analysis will be presented in Section 4.5.

4.1.4. Performance indicators

In order to better evaluate the advantages and disadvantages of the ten comparison algorithms, three performance indicators are selected: inverse generation distance plus (IGD+), hypervolume (HV) and feasible rate (FR). IGD+ is an indicator based on IGD improvement, which reflects the quality of the solution set by measuring the Euclidean distance between the solution set found by an algorithm and the true Pareto front (PF). The IGD+ is calculated as follows [54]:

$$IGD+(P^*, S) = \frac{1}{|P^*|} \sum_{y^* \in P^*} \min_{s \in S} d^+(y^*, s), \quad (13)$$

where P^* represents the true PF. S denotes the optimal solution set obtained by the algorithm. $d^+(y^*, s)$ is calculated as follows:

$$d^+(y^*, s) = \sqrt{\sum_{i=1}^m (\max\{s_i - y_i^*, 0\})^2}, \quad (14)$$

where m is the dimension of the objective space. It can be seen that the calculation of IGD+ depends on the choice of the true PF. In some cases, IGD+ may not be able to effectively distinguish similar solutions near the true PF, and thus can not reflect the actual diversity of the solution set. Therefore, HV is used to fill this gap. HV evaluates the performance of the algorithm by calculating the volume of the hypercube surrounded by the solutions in the PF and the reference points. The HV is calculated as follows [55]:

$$HV(S) = \Lambda(\{q \in \mathbb{R}^m \mid \exists s \in S : s \leq q \text{ and } q \leq r\}), \quad (15)$$

where $\Lambda(\cdot)$ denotes the Lebesgue measure and r is a fixed reference point defined in the objective space $\mathbb{R}^m$. It is evident that a lower value of IGD+ corresponds to improved algorithm performance, and a higher value of HV indicates an enhanced quality of the solution set searched by the algorithm. Therefore, these two indicators can be used to comprehensively evaluate the performance of the algorithms. Furthermore, FR is introduced to evaluate the feasibility of the solution set obtained by the algorithm. It represents the proportion of solutions satisfying all constraints in the population, which is calculated as follows:

$$FR(S) = \frac{1}{N} \sum_{i=1}^N \delta_i, \quad (16)$$

where N represents the size of population, and δ_i is calculated as follows:

$$\delta_i = \begin{cases} 1, & CV(x_i) \leq 0 \\ 0, & \text{otherwise} \end{cases}, \quad (17)$$

where $CV(x_i)$ denotes the constraint violation degree of the i -th individual in the population. Thus, the closer FR is to 1, the more feasible solutions in the population.

4.2. Comparative results

Tables 1 and 2 show the average and standard deviation of the IGD+ and HV values obtained by the proposed TCCMT and the nine comparison algorithms. Table 3 shows the FR of the final solution set obtained by all algorithms. The Wilcoxon rank sum test [56] with a significance level of 0.05 is used to analyze these results. All algorithms run independently 24 times on each test problem. Therefore, the test can determine whether the results in multiple independent runs are from the same distribution. The results are based on the rank comparison of independent samples to show whether there are significant differences between the

Table 1
IGD+ results of ten comparison algorithms on CF, LIRCMOP, and DASCMP test suites.

Problem	m	n	CCMO	cDPEA	BiCo	EMCMO	CMEGL	MTCMO	DEST	SC_EMT	CSEMT	TCCMT
CF1	2	10	7.2303e-3 (7.43e-4)	9.0720e-3 (1.13e-3)	2.0566e-2 (2.91e-3)	7.3163e-3 (8.67e-4)	5.8216e-3 (5.27e-4)	1.4671e-2 (2.38e-3)	8.5231e-3 (9.23e-4)	7.7291e-3 (9.39e-4)	1.4547e-2 (1.73e-3)	4.9671e-3 (5.36e-4)
CF2	2	10	3.1056e-2 (1.14e-2)	3.0477e-2 (1.13e-2)	4.4579e-2 (1.47e-2)	3.4956e-2 (1.06e-2)	3.5730e-2 (1.61e-2)	4.1565e-2 (2.35e-2)	4.8457e-2 (2.60e-2)	3.2437e-2 (9.05e-3)	3.9530e-2 (2.74e-2)	1.2030e-2 (3.87e-3)
CF3	2	10	2.1912e-1 (3.21e-2)	2.1541e-1 (4.41e-2)	2.2239e-1 (6.75e-2)	2.1205e-1 (5.46e-2)	2.0127e-1 (5.16e-2)	2.2581e-1 (8.06e-2)	2.3541e-1 (5.61e-2)	2.3310e-1 (3.85e-2)	2.0994e-1 (6.95e-2)	1.6000e-1 (6.61e-2)
CF4	2	10	7.7620e-2 (2.15e-2)	7.7201e-2 (2.09e-2)	7.2291e-2 (1.60e-2)	8.0372e-2 (3.05e-2)	8.5897e-2 (3.59e-2)	8.0993e-2 (3.63e-2)	8.7997e-2 (2.99e-2)	8.2895e-2 (2.34e-2)	8.6072e-2 (3.70e-2)	4.1801e-2 (1.06e-2)
CF5	2	10	2.2150e-1 (9.19e-2)	2.2207e-1 (8.95e-2)	2.0846e-1 (9.31e-2)	2.3205e-1 (8.74e-2)	2.0226e-1 (8.66e-2)	2.1860e-1 (7.43e-2)	1.9728e-1 (7.11e-2)	2.1489e-1 (8.17e-2)	1.7441e-1 (7.10e-2)	1.8897e-1 (8.62e-2)
CF6	2	10	3.9112e-2 (1.36e-2)	3.5157e-2 (1.45e-2)	6.6192e-2 (2.74e-2)	4.0264e-2 (1.27e-2)	3.9429e-2 (1.32e-2)	3.5701e-2 (9.69e-3)	3.4032e-2 (1.04e-2)	3.7120e-2 (1.12e-2)	3.5878e-2 (1.13e-2)	3.1123e-2 (1.57e-2)
CF7	2	10	2.1112e-1 (1.34e-1)	2.3092e-1 (1.01e-1)	2.2255e-1 (1.18e-1)	1.7610e-1 (1.01e-1)	2.0494e-1 (1.21e-1)	1.7993e-1 (6.44e-2)	1.8691e-1 (6.51e-2)	1.7201e-1 (6.14e-2)	2.0806e-1 (9.73e-2)	1.6791e-1 (7.58e-2)
CF8	3	10	1.4305e-1 (4.47e-2)	1.7093e-1 (6.53e-2)	1.5738e-1 (4.06e-2)	1.8003e-1 (7.46e-2)	1.5509e-1 (3.45e-2)	1.9234e-1 (8.31e-2)	1.4606e-1 (4.46e-2)	1.9425e-1 (6.97e-2)	1.2623e-1 (9.91e-2)	1.2944e-1 (1.39e-2)
CF9	3	10	6.9963e-2 (2.17e-2)	6.8675e-2 (1.72e-2)	6.8013e-2 (1.74e-2)	7.2992e-2 (1.10e-2)	7.1697e-2 (1.65e-2)	6.6801e-2 (1.39e-2)	7.5816e-2 (1.88e-2)	7.2818e-2 (1.23e-2)	6.9680e-2 (8.49e-3)	6.1967e-2 (4.34e-3)
CF10	3	10	2.5914e-1 (1.04e-1)	3.4883e-1 (1.23e-1)	4.8240e-1 (1.10e-1)	2.5895e-1 (8.95e-2)	3.0712e-1 (1.23e-1)	2.5477e-1 (1.27e-1)	3.3417e-1 (8.07e-2)	3.3417e-1 (1.02e-1)	2.5636e-1 (8.45e-2)	2.5636e-1 (8.16e-2)
LIRCMOP1	2	30	2.2603e-1 (4.94e-2)	1.5367e-1 (5.28e-2)	1.9012e-1 (1.32e-2)	2.2247e-1 (3.94e-2)	1.0967e-1 (2.43e-2)	1.3402e-1 (1.51e-2)	6.0673e-2 (2.70e-2)	1.0471e-1 (2.83e-2)	1.1552e-1 (3.32e-2)	5.5372e-2 (5.52e-2)
LIRCMOP2	2	30	1.5551e-1 (2.83e-2)	8.6958e-2 (1.87e-2)	1.2033e-1 (1.32e-2)	1.4468e-1 (1.95e-2)	7.3656e-2 (1.04e-2)	7.9136e-2 (8.44e-3)	5.1345e-2 (1.19e-2)	7.7632e-2 (1.37e-2)	9.1892e-2 (2.56e-2)	2.2742e-2 (8.01e-3)
LIRCMOP3	2	30	2.3827e-1 (2.96e-2)	1.4166e-1 (3.92e-2)	1.8528e-1 (1.55e-2)	2.3382e-1 (4.71e-2)	1.4693e-1 (3.01e-2)	1.3402e-1 (1.79e-2)	8.8950e-2 (2.78e-2)	1.3750e-1 (2.31e-2)	1.6759e-1 (3.94e-2)	4.8176e-2 (2.64e-2)
LIRCMOP4	2	30	1.7092e-1 (2.81e-2)	1.0579e-1 (1.86e-2)	1.2991e-1 (1.70e-2)	1.6472e-1 (3.26e-2)	1.0831e-1 (2.42e-2)	9.5497e-2 (1.51e-2)	8.8877e-2 (1.77e-2)	1.0409e-1 (2.16e-2)	1.1327e-1 (2.86e-2)	4.2762e-2 (1.84e-2)
LIRCMOP5	2	30	2.1657e-1 (5.69e-2)	2.3622e-1 (5.15e-2)	1.2252e+0 (6.70e-3)	2.2190e-1 (3.91e-2)	3.5058e-1 (3.32e-1)	9.6729e-1 (4.39e-1)	2.1589e-1 (3.42e-2)	2.9871e-1 (2.82e-1)	4.2398e-1 (4.09e-1)	3.1208e-2 (1.38e-2)
LIRCMOP6	2	30	2.1876e-1 (6.19e-2)	1.3453e-1 (4.22e-2)	2.5243e-1 (2.58e-4)	2.5243e-1 (4.32e-2)	1.0563e+0 (3.59e-1)	4.0942e-1 (4.61e-1)	4.0942e-1 (5.52e-2)	3.8581e-1 (3.66e-1)	2.0939e-2 (3.00e-1)	2.0939e-2 (9.19e-3)
LIRCMOP7	2	30	9.5719e-2 (2.66e-2)	1.0963e-1 (2.92e-2)	2.5809e-1 (4.44e-1)	8.8673e-2 (2.01e-2)	9.9750e-2 (2.15e-2)	1.1013e-1 (2.18e-2)	9.7532e-2 (2.47e-2)	9.7532e-2 (1.89e-2)	9.3114e-2 (1.84e-2)	4.0067e-2 (3.57e-2)
LIRCMOP8	2	30	1.3800e-1 (4.44e-2)	1.6522e-1 (3.34e-2)	1.2505e+0 (6.86e-1)	1.6434e-1 (4.22e-2)	1.6286e-1 (4.50e-2)	1.6977e-1 (4.08e-2)	1.7626e-1 (4.50e-2)	1.4902e-1 (3.82e-2)	1.5406e-1 (5.22e-2)	3.4253e-2 (3.34e-2)
LIRCMOP9	2	30	3.9343e-1 (1.18e-1)	3.4965e-1 (9.51e-2)	9.4865e-1 (1.37e-1)	3.7999e-1 (1.38e-1)	4.0253e-1 (1.39e-1)	8.4857e-1 (1.69e-1)	2.5338e-1 (1.14e-1)	4.9752e-1 (1.36e-1)	4.2288e-1 (1.55e-1)	1.0841e-1 (4.93e-2)
LIRCMOP10	2	30	1.5862e-1 (6.60e-2)	2.2275e-1 (7.44e-2)	9.3927e-1 (9.62e-2)	1.6645e-1 (7.09e-2)	2.3701e-1 (1.48e-1)	7.2236e-1 (2.17e-1)	1.7962e-1 (7.87e-2)	2.9862e-1 (1.17e-1)	3.7880e-1 (1.65e-1)	2.4233e-2 (1.80e-2)
LIRCMOP11	2	30	6.1170e-2 (2.84e-2)	8.4563e-2 (4.95e-2)	7.1965e-1 (2.52e-1)	7.1238e-2 (4.10e-2)	6.5369e-2 (3.39e-2)	5.6233e-1 (2.07e-1)	9.9937e-2 (5.47e-2)	1.4686e-1 (1.10e-1)	1.3855e-1 (1.36e-1)	9.7301e-3 (1.44e-2)
LIRCMOP12	2	30	1.8970e-1 (7.86e-2)	1.8968e-1 (6.76e-2)	6.7292e-1 (2.67e-1)	1.8266e-1 (5.62e-2)	1.6452e-1 (7.88e-2)	4.0291e-1 (1.19e-1)	1.5920e-1 (9.41e-2)	2.0926e-1 (8.67e-2)	2.4225e-1 (9.85e-2)	6.6613e-2 (4.79e-2)
LIRCMOP13	3	30	4.5622e-2 (1.80e-3)	4.4658e-2 (1.31e-3)	4.0661e-2 (1.63e-3)	4.0661e-2 (9.37e-4)	4.0661e-2 (1.59e-3)	4.0661e-2 (1.68e-3)	4.3128e-2 (1.15e-3)	4.3128e-2 (9.62e-4)	4.0518e-2 (1.30e-3)	5.6007e-2 (2.39e-3)
LIRCMOP14	3	30	4.6773e-2 (1.56e-3)	4.5643e-2 (9.29e-4)	4.5593e-2 (1.83e-3)	4.5593e-2 (9.04e-4)	4.5706e-2 (9.88e-4)	1.2704e+0 (1.78e-3)	4.4321e-2 (1.10e-3)	4.0120e-2 (8.31e-4)	4.1836e-2 (1.48e-3)	4.9519e-2 (1.59e-3)
DASCMP1	2	30	6.7833e-1 (8.95e-2)	6.6090e-1 (9.20e-2)	7.2878e-1 (2.63e-2)	7.1365e-1 (5.49e-2)	6.7839e-1 (6.81e-2)	7.1046e-1 (5.23e-2)	6.1694e-1 (1.07e-1)	7.0733e-1 (5.51e-2)	6.9821e-1 (6.88e-2)	2.5989e-3 (1.92e-4)
DASCMP2	2	30	1.4111e-1 (7.81e-3)	1.3688e-1 (1.18e-2)	1.4893e-1 (8.68e-3)	1.3978e-1 (6.98e-3)	1.3339e-1 (1.20e-2)	1.3588e-1 (8.25e-3)	1.2269e-1 (7.57e-3)	1.3824e-1 (7.11e-3)	1.3471e-1 (8.88e-3)	4.2249e-3 (2.07e-4)
DASCMP3	2	30	1.8204e-1 (2.41e-2)	1.8309e-1 (1.65e-2)	1.8821e-1 (1.13e-2)	1.9044e-1 (2.40e-2)	1.6922e-1 (2.17e-2)	1.7044e-1 (2.18e-2)	1.6672e-1 (2.18e-2)	1.6672e-1 (1.83e-2)	1.4091e-1 (2.13e-2)	2.8903e-2 (5.89e-2)
DASCMP4	2	30	1.5523e-3 (9.23e-4)	9.5619e-4 (6.70e-4)	2.0654e-1 (1.40e-1)	1.3061e-3 (8.35e-4)	8.9362e-4 (5.16e-4)	2.3244e-2 (6.82e-2)	3.3695e-2 (1.11e-3)	3.3695e-2 (1.03e-1)	3.3695e-2 (4.54e-4)	8.4316e-4 (3.71e-4)
DASCMP5	2	30	1.9738e-3 (1.42e-4)	2.0112e-3 (1.22e-4)	8.5539e-2 (2.05e-1)	2.6725e-3 (1.74e-3)	2.0094e-3 (1.19e-4)	2.1714e-2 (8.56e-2)	2.2994e-3 (9.88e-4)	2.2994e-3 (1.29e-3)	2.7527e-3 (3.56e-4)	2.0806e-3 (1.04e-4)
DASCMP6	2	30	1.2575e-2 (1.59e-2)	7.1366e-3 (4.67e-3)	1.6163e-1 (1.87e-1)	2.9859e-2 (3.76e-2)	9.6285e-3 (1.40e-2)	2.6248e-1 (2.03e-1)	2.6747e-2 (3.76e-2)	2.0343e-1 (1.79e-1)	1.4427e-2 (1.55e-2)	5.4178e-3 (1.06e-4)
DASCMP7	3	30	2.3931e-2 (1.09e-3)	2.4804e-2 (5.46e-3)	2.7081e-2 (1.02e-3)	2.3846e-2 (1.16e-3)	2.4058e-2 (1.09e-3)	2.3498e-2 (9.67e-4)	2.7119e-2 (2.08e-3)	2.3500e-2 (1.26e-3)	2.4130e-2 (1.48e-3)	2.3490e-2 (8.71e-4)
DASCMP8	3	30	1.8577e-2 (8.10e-4)	1.9864e-2 (9.87e-4)	2.1595e-2 (6.79e-4)	1.8538e-2 (9.81e-4)	1.8362e-2 (7.98e-4)	1.8607e-2 (7.48e-4)	2.0749e-2 (1.01e-3)	1.8737e-2 (6.81e-4)	1.8602e-2 (9.18e-4)	1.8621e-2 (6.23e-4)
DASCMP9	3	30	2.5230e-1 (4.66e-2)	1.7609e-1 (5.09e-2)	2.5515e-1 (4.14e-2)	2.5309e-1 (3.11e-2)	1.9785e-1 (4.33e-2)	2.2041e-1 (4.46e-2)	3.0965e-2 (7.33e-3)	2.3019e-1 (2.88e-2)	2.1485e-1 (1.96e-2)	1.9873e-2 (8.19e-4)
+/-/=			3/22/8	2/28/3	0/30/3	2/25/6	3/25/5	0/27/6	2/27/4	2/26/5	2/26/5	

algorithms. The symbols “+” and “-” in the table represent that the performance of the comparison algorithm is significantly better than and worse than TCCMT. The symbol “=” denotes no significant difference between the comparison algorithms. The best results for each test problem have been highlighted in bold and blue in Tables 1 and 2. Table 3 highlights the results where the feasible solutions are not completely found in bold and gray. It can be seen from Table 1 that TCCMT is superior to CCMO, cDPEA, BiCo, EMCMO, CMEGL, MTCMO, DEST, SC_EMT, and CSEMT in the IGD+ indicator on 22, 28, 30, 25, 27, 27, 26, and 26 test problems, respectively. Similarly, Table 2 shows that TCCMT outperforms these comparison algorithms on 22, 27, 30, 25, 24, 27, 27, 27, 25, and 27 test problems on the HV indicator, respectively. Table 3 indicates that TCCMT finds feasible solution set for all problems. In general, the proposed TCCMT is far superior to

other algorithms in both IGD+ and HV indicators and exhibits perfect feasibility.

4.2.1. Results for CF

In the CF test suite, TCCMT achieves the best IGD+ results on 9 test problems and the best HV results on 8 test problems. It is approximately equal to the dominant algorithm on other problems that have not achieved the best results, which shows the excellent performance of TCCMT on the CF test suite. As shown in Fig. 5, it illustrates the distribution of the solutions obtained by the ten comparison algorithms on CF2. It is easy to see that the proposed TCCMT converges well and is most evenly distributed on the CPF. Due to the introduction of a highly complex variable coupling relationship in the CF test suite, it brings great challenges to the convergence of the algorithm.

Table 2

HV results of ten comparison algorithms on CF, LIRCMOP, and DASCMP test suites.

Problem	m	n	CCMO	cDPEA	BiCo	EMCMO	CMEGL	MTCMO	DEST	SC_EMT	CSEMT	TCCMT
CF1	2	10	5.5704e-1 (8.98e-4)	5.5489e-1 (1.38e-3)	5.4019e-1 (3.69e-3)	5.5686e-1 (1.08e-3)	5.5875e-1 (7.33e-4)	5.4744e-1 (3.04e-3)	5.5545e-1 (1.19e-3)	5.5615e-1 (1.31e-3)	5.4779e-1 (2.25e-3)	5.6006e-1 (6.49e-4)
CF2	2	10	6.2996e-1 (1.85e-2)	6.3253e-1 (2.26e-2)	5.9406e-1 (2.62e-2)	6.2380e-1 (1.91e-2)	6.2840e-1 (2.15e-2)	6.1132e-1 (2.71e-2)	6.0576e-1 (3.85e-2)	6.2314e-1 (1.95e-2)	6.2307e-1 (3.04e-2)	6.6262e-1 (5.84e-3)
CF3	2	10	1.7440e-1 (3.95e-2)	1.7615e-1 (3.59e-2)	1.8150e-1 (4.22e-2)	1.8051e-1 (3.67e-2)	1.8007e-1 (4.13e-2)	1.8058e-1 (4.48e-2)	1.5091e-1 (3.89e-2)	1.5619e-1 (4.42e-2)	1.8742e-1 (5.00e-2)	2.2187e-1 (5.90e-2)
CF4	2	10	4.1641e-1 (3.10e-2)	4.1740e-1 (2.75e-2)	4.2364e-1 (2.16e-2)	4.1112e-1 (4.50e-2)	4.0305e-1 (5.21e-2)	4.1266e-1 (4.84e-2)	4.0063e-1 (4.20e-2)	4.0809e-1 (3.28e-2)	4.0329e-1 (5.52e-2)	4.7014e-1 (1.57e-2)
CF5	2	10	2.7456e-1 (7.11e-2)	2.7591e-1 (6.75e-2)	2.9047e-1 (7.50e-2)	2.6430e-1 (6.93e-2)	2.8679e-1 (6.58e-2)	2.7365e-1 (5.38e-2)	2.8570e-1 (6.05e-2)	2.8429e-1 (6.15e-2)	3.0860e-1 (5.96e-2)	3.1364e-1 (7.11e-2)
CF6	2	10	6.4980e-1 (1.45e-2)	6.5456e-1 (1.45e-2)	6.2227e-1 (2.62e-2)	6.4891e-1 (1.30e-2)	6.4877e-1 (1.34e-2)	6.5232e-1 (1.00e-2)	6.5398e-1 (1.12e-2)	6.5185e-1 (1.12e-2)	6.5353e-1 (1.19e-2)	6.6108e-1 (1.68e-2)
CF7	2	10	4.2219e-1 (1.20e-1)	4.1614e-1 (8.33e-2)	4.1409e-1 (1.08e-1)	4.6160e-1 (8.64e-2)	4.2302e-1 (1.22e-1)	4.2020e-1 (6.56e-2)	4.3225e-1 (6.52e-2)	4.5618e-1 (6.07e-2)	4.3564e-1 (8.02e-2)	4.6080e-1 (8.82e-2)
CF8	3	10	3.1549e-1 (6.46e-2)	2.6917e-1 (8.82e-2)	3.0560e-1 (5.36e-2)	2.6367e-1 (8.44e-2)	2.8883e-1 (5.50e-2)	2.6987e-1 (7.30e-2)	3.2217e-1 (6.97e-2)	2.5059e-1 (5.38e-2)	2.6532e-1 (3.56e-2)	3.3831e-1 (2.64e-2)
CF9	3	10	4.0879e-1 (5.26e-2)	4.1303e-1 (3.45e-2)	4.0726e-1 (4.03e-2)	4.1014e-1 (1.68e-2)	4.0952e-1 (3.77e-2)	4.1515e-1 (3.33e-2)	4.0032e-1 (4.07e-2)	4.0714e-1 (2.47e-2)	4.1335e-1 (1.46e-2)	4.2804e-1 (8.62e-3)
CF10	3	10	1.7348e-1 (3.00e-2)	1.2809e-1 (3.74e-2)	1.1637e-1 (6.02e-3)	1.5484e-1 (4.08e-2)	1.5375e-1 (4.03e-2)	1.3749e-1 (4.94e-2)	1.6035e-1 (3.10e-2)	1.4614e-1 (4.21e-2)	1.5850e-1 (3.80e-2)	1.4850e-1 (4.79e-2)
LIRCMOP1	2	30	1.2074e-1 (1.81e-2)	1.5397e-1 (1.96e-2)	1.3618e-1 (6.54e-3)	1.2383e-1 (1.71e-2)	1.7169e-1 (1.36e-2)	1.5959e-1 (9.39e-3)	1.9468e-1 (1.69e-2)	1.7320e-1 (1.60e-2)	1.6748e-1 (1.42e-2)	2.0173e-1 (2.68e-2)
LIRCMOP2	2	30	2.3336e-1 (1.85e-2)	2.8100e-1 (1.77e-2)	2.5467e-1 (9.05e-3)	2.3977e-1 (1.34e-2)	2.9012e-1 (9.67e-3)	2.8386e-1 (7.50e-3)	3.1358e-1 (1.25e-2)	2.8536e-1 (1.01e-2)	2.7536e-1 (1.83e-2)	3.4279e-1 (7.64e-3)
LIRCMOP3	2	30	1.0540e-1 (1.26e-2)	1.4304e-1 (1.51e-2)	1.2628e-1 (6.70e-3)	1.0860e-1 (1.73e-2)	1.3883e-1 (1.14e-2)	1.4590e-1 (8.42e-3)	1.6571e-1 (1.11e-2)	1.4476e-1 (1.28e-2)	1.3116e-1 (1.60e-2)	1.8002e-1 (1.17e-2)
LIRCMOP4	2	30	2.0120e-1 (1.63e-2)	2.3916e-1 (1.26e-2)	2.2420e-1 (1.08e-2)	2.0503e-1 (1.95e-2)	2.3841e-1 (1.59e-2)	2.4582e-1 (1.10e-2)	2.5451e-1 (1.32e-2)	2.3994e-1 (1.45e-2)	2.3522e-1 (1.86e-2)	2.8550e-1 (1.33e-2)
LIRCMOP5	2	30	1.5960e-1 (2.57e-2)	1.5078e-1 (2.27e-2)	0.0000e+0 (0.00e+0)	1.5607e-1 (1.68e-2)	1.3391e-1 (5.40e-2)	3.8658e-2 (6.85e-2)	1.5873e-1 (1.55e-2)	1.4599e-1 (4.87e-2)	1.2505e-1 (6.89e-2)	2.7604e-1 (1.12e-2)
LIRCMOP6	2	30	1.2232e-1 (1.54e-2)	9.9969e-2 (7.39e-3)	0.0000e+0 (0.00e+0)	1.1187e-1 (1.21e-2)	8.9961e-2 (3.70e-2)	2.7167e-2 (4.33e-2)	1.1081e-1 (1.43e-2)	9.2800e-2 (3.82e-2)	9.4019e-2 (3.07e-2)	1.8930e-1 (4.16e-3)
LIRCMOP7	2	30	2.5045e-1 (1.10e-2)	2.4469e-1 (1.15e-2)	2.2022e-1 (7.03e-2)	2.5288e-1 (7.78e-3)	2.4837e-1 (8.36e-3)	2.4432e-1 (8.54e-3)	2.4788e-1 (9.81e-3)	2.4973e-1 (7.45e-3)	2.5112e-1 (7.44e-3)	2.7635e-1 (1.77e-2)
LIRCMOP8	2	30	2.4035e-1 (1.32e-2)	2.3258e-1 (8.94e-3)	6.4482e-2 (1.03e-1)	2.3384e-1 (1.14e-2)	2.3266e-1 (1.20e-2)	2.3127e-1 (1.01e-2)	2.3084e-1 (9.19e-3)	2.3679e-1 (9.79e-3)	2.3841e-1 (1.43e-2)	2.7931e-1 (1.67e-2)
LIRCMOP9	2	30	3.4516e-1 (5.93e-2)	3.6692e-1 (7.92e-2)	1.1473e-1 (3.01e-2)	3.5629e-1 (6.68e-2)	3.4028e-1 (7.61e-2)	1.6560e-1 (6.97e-2)	4.1508e-1 (6.18e-2)	2.8805e-1 (7.55e-2)	3.2809e-1 (7.76e-2)	5.0024e-1 (4.29e-2)
LIRCMOP10	2	30	6.0054e-1 (4.27e-2)	5.5446e-1 (5.33e-2)	7.0893e-2 (3.17e-2)	5.8894e-1 (6.52e-2)	5.3045e-1 (1.23e-1)	1.6290e-1 (1.54e-1)	5.7755e-1 (7.59e-2)	4.8258e-1 (1.03e-1)	4.1839e-1 (1.42e-1)	6.9342e-1 (1.92e-3)
LIRCMOP11	2	30	6.5676e-1 (2.13e-2)	6.3733e-1 (4.18e-2)	2.6430e-1 (1.12e-1)	6.4735e-1 (3.68e-2)	6.5287e-1 (2.08e-2)	3.2705e-1 (1.28e-1)	6.2449e-1 (4.29e-2)	5.9596e-1 (8.11e-2)	6.0410e-1 (8.72e-2)	6.8740e-1 (9.62e-3)
LIRCMOP12	2	30	5.0639e-1 (4.88e-2)	5.0880e-1 (3.61e-2)	3.0789e-1 (1.16e-1)	5.0944e-1 (3.28e-2)	5.2242e-1 (4.43e-2)	4.3373e-1 (3.77e-2)	5.2746e-1 (4.88e-2)	5.0042e-1 (4.18e-2)	4.9164e-1 (3.65e-2)	5.8136e-1 (2.98e-2)
LIRCMOP13	3	30	5.5397e-1 (1.68e-3)	5.5454e-1 (1.57e-3)	9.5097e-5 (1.03e-4)	5.5912e-1 (9.82e-4)	5.5444e-1 (1.54e-3)	1.1244e-4 (1.19e-4)	5.5467e-1 (1.50e-3)	5.6070e-1 (8.86e-4)	5.5939e-1 (1.12e-3)	5.4296e-1 (2.34e-3)
LIRCMOP14	3	30	5.5389e-1 (1.32e-3)	5.5475e-1 (1.05e-3)	4.0479e-4 (2.54e-4)	5.5507e-1 (9.22e-4)	5.5491e-1 (9.26e-4)	4.8235e-4 (3.55e-4)	5.5483e-1 (1.30e-3)	5.6088e-1 (7.35e-4)	5.5900e-1 (1.58e-3)	5.5097e-1 (1.45e-3)
DASCMP1	2	30	1.8161e-2 (2.16e-2)	2.0227e-2 (2.25e-2)	7.4230e-3 (4.72e-3)	9.8732e-3 (1.71e-2)	1.6768e-2 (1.65e-2)	1.0477e-2 (1.25e-2)	3.2184e-2 (2.68e-2)	1.1339e-2 (1.30e-2)	1.2777e-2 (1.68e-2)	2.1108e-1 (9.27e-4)
DASCMP2	2	30	2.5723e-1 (2.78e-3)	2.6070e-1 (3.86e-3)	2.4826e-1 (3.72e-3)	2.5803e-1 (3.87e-3)	2.6549e-1 (3.39e-3)	2.6459e-1 (3.13e-3)	2.7159e-1 (3.33e-3)	2.5964e-1 (2.68e-3)	2.6244e-1 (3.58e-3)	3.5444e-1 (1.58e-4)
DASCMP3	2	30	2.1427e-1 (1.42e-2)	2.1235e-1 (9.66e-3)	2.1037e-1 (3.44e-3)	2.1159e-1 (1.15e-2)	2.2059e-1 (1.27e-2)	2.2090e-1 (1.30e-2)	2.2159e-1 (1.07e-2)	2.2159e-1 (1.25e-2)	2.2370e-1 (1.50e-2)	2.9949e-1 (3.01e-2)
DASCMP4	2	30	1.9999e-1 (4.32e-3)	2.0362e-1 (1.32e-3)	1.1942e-1 (5.32e-2)	2.0099e-1 (4.11e-3)	2.0315e-1 (2.34e-3)	1.9322e-1 (2.84e-2)	1.9996e-1 (3.42e-3)	1.8855e-1 (3.60e-2)	2.0269e-1 (1.78e-3)	2.0343e-1 (1.75e-3)
DASCMP5	2	30	3.5125e-1 (1.80e-4)	3.5120e-1 (1.79e-4)	3.0670e-1 (1.01e-1)	3.5085e-1 (1.20e-3)	3.5124e-1 (1.84e-4)	3.3906e-1 (5.07e-2)	3.4941e-1 (7.33e-4)	3.5110e-1 (1.17e-3)	3.5041e-1 (3.90e-4)	3.5112e-1 (2.19e-4)
DASCMP6	2	30	3.0622e-1 (1.11e-2)	3.0973e-1 (6.42e-3)	2.2743e-1 (9.97e-2)	2.9760e-1 (1.96e-2)	3.0834e-1 (1.04e-2)	1.7434e-1 (1.06e-1)	2.9914e-1 (2.02e-2)	2.0342e-1 (9.61e-2)	3.0775e-1 (6.79e-3)	3.1204e-1 (2.40e-4)
DASCMP7	3	30	2.8806e-1 (6.19e-4)	2.8753e-1 (4.44e-3)	2.8543e-1 (4.16e-3)	2.8804e-1 (7.37e-4)	2.8784e-1 (3.58e-4)	2.8827e-1 (3.05e-4)	2.8560e-1 (7.34e-4)	2.8833e-1 (5.84e-4)	2.8760e-1 (6.37e-4)	2.8813e-1 (4.08e-4)
DASCMP8	3	30	2.0712e-1 (4.27e-4)	2.0604e-1 (4.07e-4)	2.0514e-1 (3.98e-3)	2.0711e-1 (5.90e-4)	2.0684e-1 (4.21e-4)	2.0728e-1 (3.93e-4)	2.0542e-1 (4.96e-4)	2.0735e-1 (4.15e-4)	2.0679e-1 (5.49e-4)	2.0670e-1 (4.84e-4)
DASCMP9	3	30	1.2532e-1 (1.34e-2)	1.5065e-1 (1.71e-2)	1.2429e-1 (1.23e-2)	1.2485e-1 (8.94e-3)	1.4424e-1 (1.41e-2)	1.3646e-1 (1.44e-2)	1.9840e-1 (4.56e-3)	1.3257e-1 (9.32e-3)	1.3742e-1 (7.65e-3)	2.0561e-1 (2.99e-4)
+/-/=			3/22/8	3/27/3	0/30/3	3/25/5	3/24/6	1/27/5	2/27/4	4/25/4	2/27/4	

In addition, the CPFs of most test problems in the CF test suite are located at the constraint boundary (CF4-7) or part of the UPF (CF1-3 and CF8-10). However, many of the comparison algorithms construct auxiliary tasks to approach UPF (CCMO, EMCMO, and CMEGL) or use the potential infeasible solutions between UPF and CPF (cDPEA, BiCo, MTCMO, and SC_EMT), which obviously does not help the convergence of the algorithm. The two auxiliary populations in the TCCMT explore the diversity sufficiently in the first two stages by means of cooperative competition. Meantime, the similarity among the two auxiliary tasks and the main task is high in the later stage of evolution, which can provide a strong and effective convergence pressure for the algorithm. Therefore, TCCMT achieves a remarkable advantage on the CF test suite.

4.2.2. Results for LIRCMOP

On the 14 test problems of LIRCMOP, TCCMT only performs poorly on the two problems of LIRCMOP13 and LIRCMOP14, but it is significantly better than the comparison algorithms on all other problems. In order to better illustrate the good performance of TCCMT, LIRCMOP6 is selected to show the distribution of the solutions obtained by the ten comparison algorithms. It can be seen from the Fig. 6 that only the solution obtained by TCCMT searches for CPF and is well distributed. On the LIRCMOP1-4 test problems, due to the constraints, their feasible regions are relatively narrow and difficult to search. Therefore, this type of test problem requires the algorithm to have a certain ability to explore diversity. With the diversity-enhanced framework, DEST shows some advantages on LIRCMOP1-4. cDPEA, MTCMO and other algorithms obtain

Table 3

FR results of ten comparison algorithms on CF, LIRCMOP, and DASCMP test suites.

Problem	<i>m</i>	<i>n</i>	CCMO	cDPEA	BiCo	EMCMO	CMEGL	MTCMO	DEST	SC_EMT	CSEMT	TCCMT
CF1	2	10	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
CF2	2	10	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
CF3	2	10	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
CF4	2	10	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
CF5	2	10	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
CF6	2	10	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
CF7	2	10	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
CF8	3	10	100.00%	100.00%	95.83%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
CF9	3	10	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
CF10	3	10	100.00%	100.00%	4.17%	100.00%	100.00%	87.50%	100.00%	100.00%	100.00%	100.00%
LIRCMOP1	2	30	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
LIRCMOP2	2	30	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
LIRCMOP3	2	30	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
LIRCMOP4	2	30	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
LIRCMOP5	2	30	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
LIRCMOP6	2	30	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
LIRCMOP7	2	30	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
LIRCMOP8	2	30	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
LIRCMOP9	2	30	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
LIRCMOP10	2	30	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
LIRCMOP11	2	30	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
LIRCMOP12	2	30	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
LIRCMOP13	3	30	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
LIRCMOP14	3	30	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
DASCMP1	2	30	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
DASCMP2	2	30	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
DASCMP3	2	30	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
DASCMP4	2	30	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
DASCMP5	2	30	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
DASCMP6	2	30	100.00%	100.00%	95.83%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
DASCMP7	3	30	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
DASCMP8	3	30	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
DASCMP9	3	30	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%

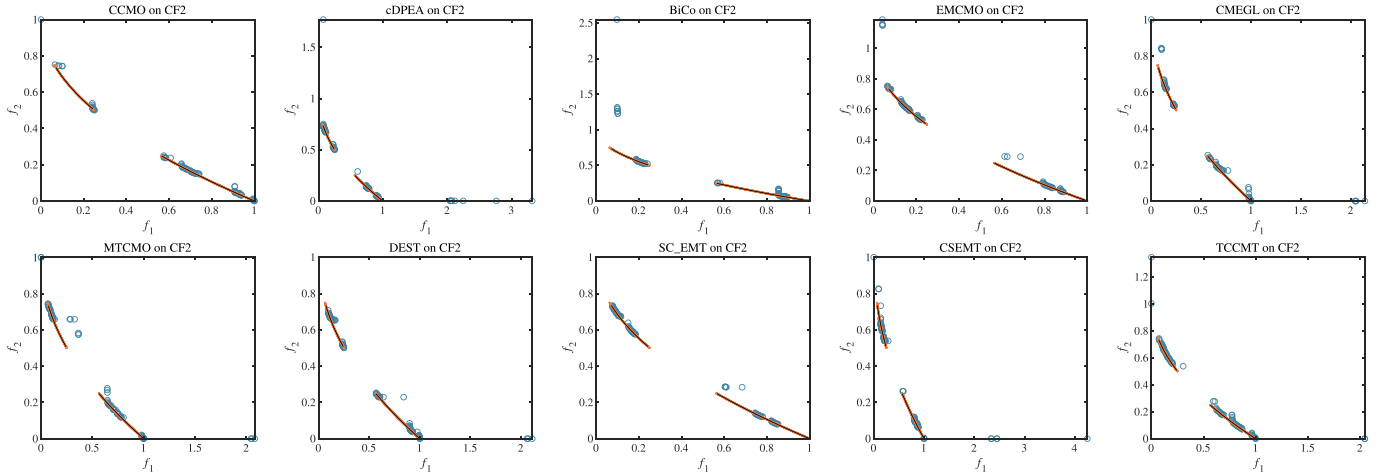


Fig. 5. The distribution of the main population formed by ten comparison algorithms on CF2. Light green is the feasible region, and orange and blue lines represent CPF and UPF, respectively. The teal blue circles refer to the population.

better results by using the information of infeasible solutions between UPF and CPF to maintain a certain diversity. However, TCCMT performs best in maintaining diversity, which is fundamentally due to the important role played by the double angle constraint relaxation auxiliary task. The UPF and CPF of LIRCMOP5 and LIRCMOP6 are overlapped, so the algorithms using UPF information such as CCMO and EMCMO have achieved good results. However, these algorithms are not fully exploited for diversity, resulting in incomplete CPF search. Because of the competitive relationship between the two auxiliary tasks, TCCMT can use another more valuable auxiliary task after searching for UPF to better find CPF. The UPF and CPF of LIRCMOP9-10 are partially overlapped, which is similar to the case of LIRCMOP5-6, but requires stronger exploration ability to identify CPF. In LIRCMOP7-8, CPF is blocked by a large number of infeasible regions, and the distance between CPF and UPF is far. The feasible regions of CPF in LIRCMOP11-12 are isolated from each other, and CPF is located at the constraint boundary. Both of

these two types of test problems need to cross large infeasible regions and achieve balance in diversity and convergence. Many comparison algorithms can only achieve one point, so they are obviously insufficient compared with TCCMT. LIRCMOP13-14 are all three-objective high-dimensional complex problems. The proposed TCCMT on these two problems is not dominant, but SC_EMT and CSEMT perform well. This is mainly because their auxiliary tasks only consider a single constraint or partial constraint when considering constraints, which can quickly converge.

4.2.3. Results for DASCMP

On the DASCMP test suite, TCCMT achieves the best results on 7 test problems. Especially on DASCMP1-3, the proposed TCCMT is significantly better than all the comparison algorithms. The difficulty tuples of the three problems of DASCMP1-3 are the same, and the feasible

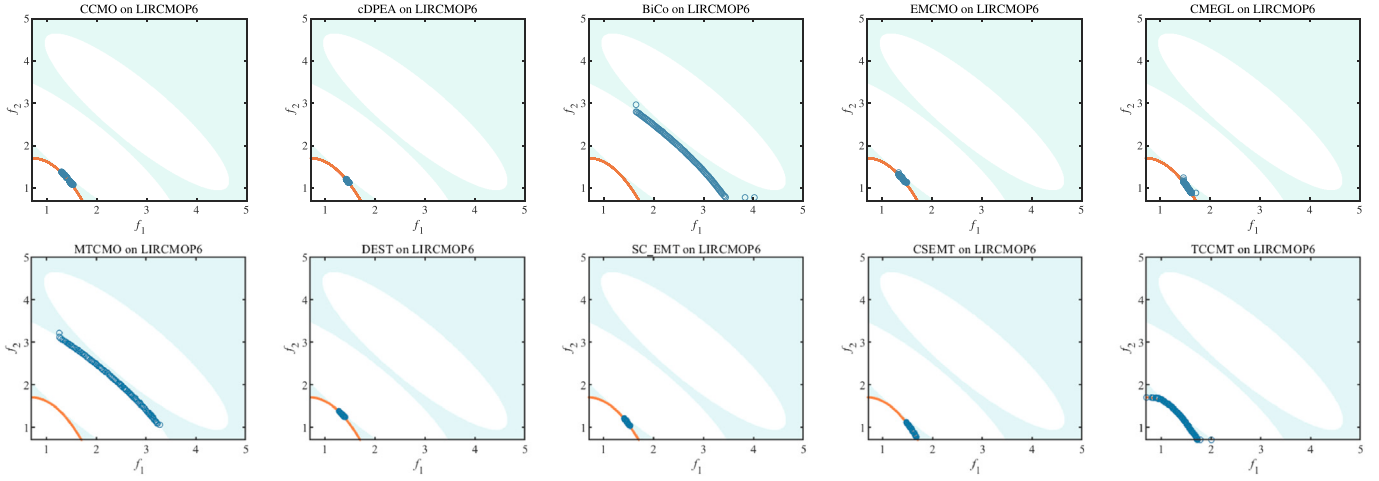


Fig. 6. The distribution of the main population formed by ten comparison algorithms on LIRCOP6. Light green is the feasible region, and orange and blue lines represent CPF and UPF, respectively. The teal blue circles refer to the population.

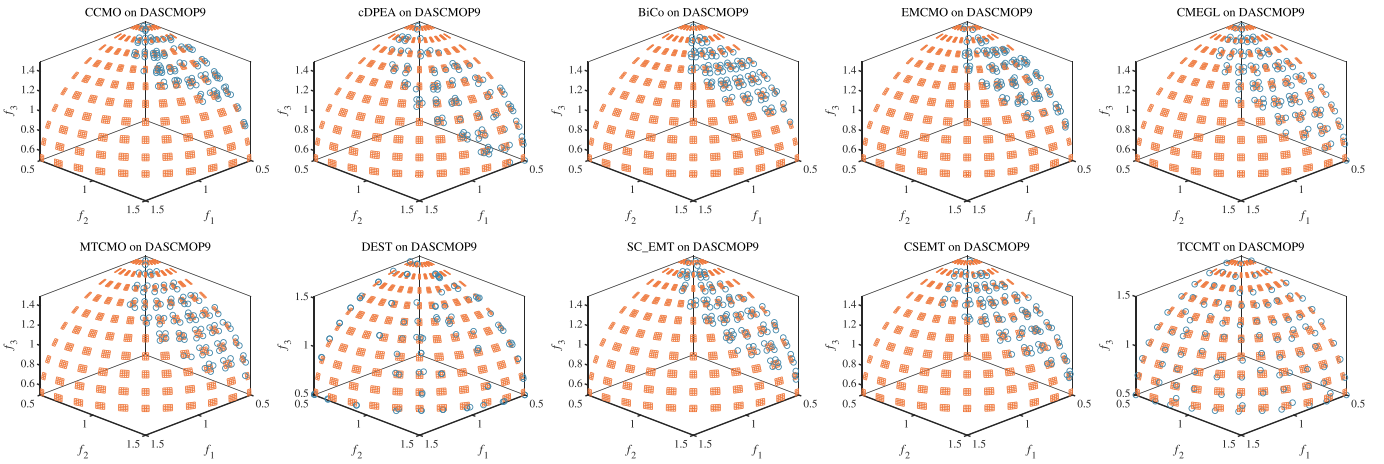


Fig. 7. The distribution of the main population formed by ten comparison algorithms on DASCOP9. Light green is the feasible region, and orange and blue lines represent CPF and UPF, respectively. The teal blue circles refer to the population.

region is divided into shapes of different sizes by large infeasible regions. Unconstrained auxiliary tasks constructed by many algorithms will quickly approach UPF, thus missing the exploration of some small feasible regions. This has led to local optima in many comparison algorithms. The proposed TCCMT will select the best auxiliary task to co-evolve according to the degree of contribution in the process of evolution, and the constrained adaptive regression task in the third stage will adaptively select the constraint.

Therefore, the population can fully explore the feasible region and reduce the risk of local optima. DASCOP4-6 uses the same constraint function, and many algorithms can easily find CPF with the help of unconstrained auxiliary population. However, compared with other algorithms, the double angle constraint relaxation auxiliary task can help the main population obtain better distribution on CPF in the later stage. DASCOP7-9 has three objectives and the same constraint function as well. TCCMT is still ahead of many comparison algorithms on these three issues. DASCOP9 is selected to show the distribution of the solutions obtained by the ten comparison algorithms. It can be clearly seen from the Fig. 7 that the solution distribution obtained by TCCMT is the best, which intuitively shows the good diversity of the algorithm.

4.3. Statistics results

In order to show the advantages of the TCCMT, the data of Tables 1 and 2 are imported into KEEL [57] software for the Wilcoxon signed-ranks test and Friedman aligned test. Wilcoxon signed-ranks test is used to test the statistically significant difference of the performance indicators obtained by the TCCMT and each comparison algorithm on the same test problem. Tables 4 and 5 show the results of the Wilcoxon test on IGD+ and HV indicators, respectively. A larger difference between R^+ and R^- in the table represents a more pronounced distinction between the proposed TCCMT and nine comparison algorithms. It can be seen from the table that the difference between every comparison algorithm and TCCMT is significantly large. At the same time, the exact p -value in the comparison results is less than 0.05. These fully demonstrate the remarkable advantages of TCCMT. Friedman aligned test uses ranks to evaluate whether there is a statistically significant difference in the overall performance of the algorithms on all test suites. It can be seen from the Figs. 8 and 9 that TCCMT has achieved the best rank in both IGD+ and HV results. This shows that the performance of the algorithm is significantly better than the comparison algorithms. In conclusion, the Wilcoxon signed-ranks test confirms the general significant advantages

Table 4
IGD+ results obtained by the Wilcoxon test for TCCMT.

TCCMT VS	R^+	R^-	Exact p -value	Asymptotic p -value
CCMO	512.0	16.0	7.8700e-8	0.000003
cDPEA	544.0	17.0	4.8200e-8	0.000002
BiCo	561.0	0.0	2.328e-10	0.000001
EMCMO	542.0	19.0	7.1480e-8	0.000003
CMEGL	540.0	21.0	1.0408e-7	0.000003
MTCMO	559.5	1.5	5.820e-10	0.000001
DEST	543.0	18.0	5.8900e-8	0.000003
SC_EMT	514.0	14.0	5.1220e-8	0.000003
CSEMT	503.0	25.0	4.2100e-7	0.000008

Table 5
HV results obtained by the Wilcoxon test for TCCMT.

TCCMT VS	R^+	R^-	Exact p -value	Asymptotic p -value
CCMO	506.0	22.0	2.4960e-7	0.000004
cDPEA	541.5	19.5	7.8930e-8	0.000003
BiCo	561.0	0.0	2.328e-10	0.000001
EMCMO	507.0	21.0	2.0820e-7	0.000005
CMEGL	540.5	20.5	9.5230e-8	0.000003
MTCMO	559.5	1.5	5.820e-10	0.000001
DEST	543.0	18.0	5.8900e-8	0.000002
SC_EMT	511.5	16.5	8.7550e-8	0.000004
CSEMT	508.5	19.5	1.5786e-7	0.000004

of the proposed algorithm in each test problem. The Friedman aligned test establishes the leading position of TCCMT in the overall comparison of all algorithms. Therefore, the two statistical test results are consistent and fully confirm the statistically significant advantages of the proposed TCCMT.

4.4. Ablation study

The proposed TCCMT is studied to verify the effectiveness of each part. We mainly test the stage control, the collaborative competition framework, and the improvement of the auxiliary task. Seven variants are constructed: TCCMT_b1, TCCMT_b2, TCCMT_b3, TCCMT_b4, TCCMT_b5, TCCMT_b6, and TCCMT_b7. These variants are compared with TCCMT using the same test suite. The summary of the experimental results and the Wilcoxon test is presented in Tables 6 and 7. In order to show the results intuitively, 6 test problems are selected to draw the IGD+ convergence curve, which are presented in Fig. 10. TCCMT_b1, TCCMT_b2, and TCCMT_b3 are the verification of the effectiveness of the stage control in the evolutionary process. TCCMT_b1 does not have the first stage during the evolution. TCCMT_b2 and TCCMT_b3 do not have the second and third stages, respectively. The results show that the lack of any kind of stage control will lead to a decline in the performance of the algorithm. In the process of evolution, the population will focus on different adaptive characteristics and optimization objectives. Therefore, it is necessary to control the evolution stage of the algorithm

to implement different strategies. In the next section, we will analyze the parameters of stage control. TCCMT_b4 and TCCMT_b5 explore the effectiveness of the proposed collaborative competition framework. In TCCMT_b4, there is only competition between two auxiliary tasks. This means the two auxiliary tasks do not collaborate through parent mixing. However, this makes TCCMT_b4 perform poorly on some test problems, which confirms the value of collaborative competition. The algorithm in TCCMT_b5 does not use the framework of collaborative competition, and simultaneously evolves three populations similar to EMT. But from the results, the effect decreased significantly. The variants TCCMT_b6 and TCCMT_b7 mainly verify the strategy of the auxiliary task. In TCCMT_b6, the auxiliary task HP_1 does not adopt the strategy of constrained adaptive regression. While the auxiliary task HP_2 in TCCMT_b7 does not use the double angle enhancement strategy, the results show that the proposed TCCMT can adapt to more problems and make breakthroughs in some difficult problems by using the improved auxiliary tasks.

In the EMT framework, all auxiliary tasks perform knowledge transfer with the main task. This not only carries the risk of negative knowledge transfer, but also causes a waste of computing resources. In the proposed collaborative competition multitasking framework, only the dominant auxiliary task can be selected to co-evolve with the main task. The judgment of the dominant auxiliary task is based on its contribution to the main population. It is measured by the number of solutions generated by the auxiliary task remaining in the main population after being filtered by Algorithm 2. During the archive update, the poor fitness and the most crowded solutions are deleted. Therefore, the retained knowledge has good convergence and diversity, which is beneficial to the main population. Eventually, the auxiliary population containing more useful knowledge is selected to co-evolve with the main population, which effectively avoids the negative knowledge transfer. In addition, the adaptive adjustment strategy of the auxiliary tasks in the later stage of evolution can maintain a high degree of similarity with the main task. The generated solution avoids the goal conflict with the main task from the source, which can reduce the possibility of negative knowledge transfer. The auxiliary task has stronger exploration ability, which can provide more useful information. The retained high-quality solutions can help the main population maintain diversity and avoid falling into local optimum. Therefore, TCCMT performs better than the variants.

Although the proposed framework improves the performance of the algorithm and reduces the negative knowledge transfer, the dynamic switching between auxiliary tasks may lead to the risk of unstable knowledge transfer. When the auxiliary task over-optimizes the objectives unrelated to the main task, the generated solutions may interfere with the evolution direction of the main population. Especially in the problem of fragmented feasible region, the infeasible solutions with superior objective value are easy to mislead the main population to cross the feasible boundary. In addition, the mechanism of evaluating the advantage of auxiliary tasks only based on contribution is slightly single.

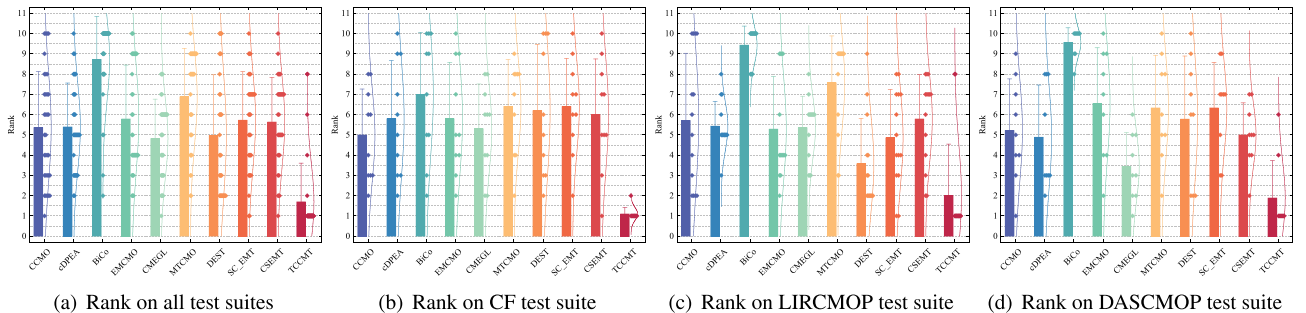


Fig. 8. The box plots of average IGD+ rank obtained by the Friedman test.

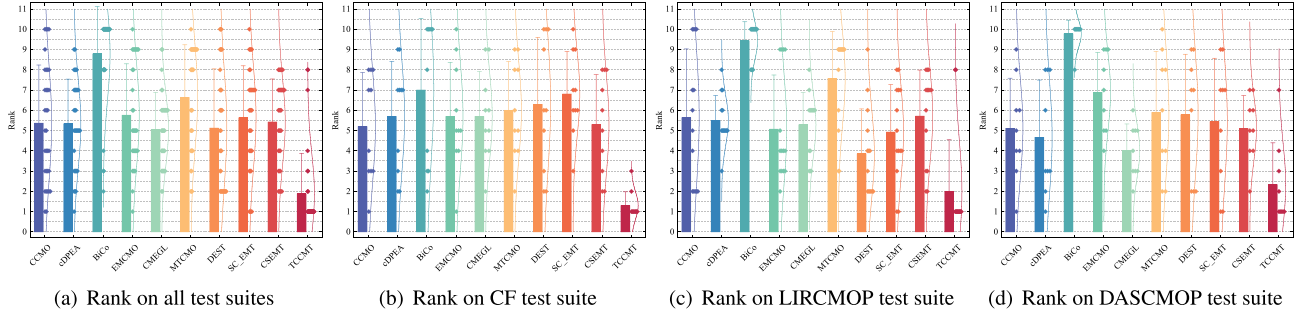


Fig. 9. The box plots of average HV rank obtained by the Friedman test.

Table 6

The summary of IGD+ results and the Wilcoxon test for ablation study.

TCCMT VS	+/-/=	R^+	R^-	Exact p -value	Asymptotic p -value
TCCMT_b1	0/4/29	333.5	227.5	≥ 0.2	0.339108
TCCMT_b2	0/11/22	512.0	16.0	$7.870e-8$	0.000003
TCCMT_b3	0/15/18	513.0	48.0	$5.628e-6$	0.000031
TCCMT_b4	0/9/24	432.0	129.0	$5.826e-3$	0.006609
TCCMT_b5	0/14/19	384.0	177.0	$6.522e-3$	0.063133
TCCMT_b6	0/8/25	438.0	90.0	$7.132e-4$	0.001073
TCCMT_b7	0/6/27	335.5	225.5	≥ 0.2	0.320279

Table 7

The summary of HV results and the Wilcoxon test for ablation study.

TCCMT VS	+/-/=	R^+	R^-	Exact p -value	Asymptotic p -value
TCCMT_b1	1/3/29	330.5	230.5	≥ 0.2	0.348276
TCCMT_b2	0/12/21	523.5	37.5	$1.4332e-6$	0.000002
TCCMT_b3	1/16/16	492.0	36.0	$2.3120e-6$	0.000014
TCCMT_b4	0/9/24	386.0	142.0	$2.1620e-2$	0.014984
TCCMT_b5	2/12/19	406.0	155.0	$2.4120e-2$	0.016072
TCCMT_b6	0/8/25	423.5	104.5	$2.1500e-3$	0.001485
TCCMT_b7	0/5/28	295.5	232.5	≥ 0.2	0.499192

Other potential values contained in auxiliary tasks have not been considered, such as high-quality infeasible solutions, convergence speed and exploration ability.

4.5. Parameter analysis

The proposed TCCMT has three hyperparameters: the degree of collaboration CP , stage control parameters η and γ . These three parameters do not affect each other, so they are analyzed separately. A total of fourteen variants are set up with different parameter settings. The experimental results are tested using Friedman to obtain the rank of each parameter, which is shown in Fig. 11. For CP , we construct five variants, $CP = \{0.5, 0.6, 0.7, 0.8, 0.9\}$. Obviously, a smaller value of CP is associated with a greater degree of collaboration. It can be seen from the results that with the increase of CP , the performance of the algorithm is significantly worse. When it increases to the maximum value of 1, it corresponds to the variant without collaboration in the previous section. When the CP is too small, the performance of the algorithm will decrease. This may be because the degree of cooperation is too high, which disturbs the evolutionary direction of the normal task. The phase control parameter η is mainly used to control the competitive state of two tasks in the evolutionary phase, and five variants $\eta = \{0.1, 0.2, 0.3, 0.4, 0.5\}$ are set for the compared study. The algorithm needs to fully explore the search space in the early evolution process to ensure diversity and avoid premature convergence. Therefore, when η is set too small, the performance of the algorithm is significantly reduced. Since the control parameters of the two stages jointly determine the entire evolutionary

stage, only four variants of γ are set, $\gamma = \{0.6, 0.7, 0.8, 0.9\}$. In the later stage of evolution, the algorithm needs to ensure sufficient convergence driving force to improve the quality of the solution and achieve the convergence of the global optimal solution. Therefore, when $\eta = 0.9$, the auxiliary task does not transform the strategy in time, and the effect of the algorithm is significantly worse. According to the results shown in Fig. 11, $CP = 0.6$, $\eta = 0.3$, and $\gamma = 0.8$ is a set of parameter settings that make the algorithm achieve the best results.

The above analysis shows that the tri-stage design can dynamically adjust the strategies to effectively balance exploration and exploitation, which improves the performance of TCCMT. However, the parameters of stage control are fixed and need to be preset in advance, which brings some limitations. Firstly, when the number of function evaluations is strictly limited, the allocation of fixed stage runtime may lead to the lack of resources in the critical stage. Due to the limitation of computing resources, it is difficult to achieve the optimal in each stage. If the allocation is insufficient in the early exploration stage, it is difficult for the population to fully explore the search space, which increases the risk of falling into local optimum. While the allocation of the later development stage is insufficient, the convergence of population can not be guaranteed, which decreases the quality of the solution set. Secondly, the structure of the feasible region and the conflict intensity of objective function in different test problems are significantly different, resulting in different evolution time required for each stage. Therefore, fixed parameters cannot adapt to all problems. More importantly, fixed parameters lack the ability to respond to the dynamic state of the population. When encountering complex problems with high constraints, if the proportion of feasible solutions plummets or the diversity declines sharply, the algorithm cannot adjust the stage strategy independently to cope with unexpected challenges. In addition, the fixed parameters require a large number of experiments in advance to determine the best parameter. The performance of the algorithm is highly sensitive to the parameters of the stage switching, and small parameter changes may lead to significant fluctuations in the quality of the solution. Therefore, it is difficult to select the appropriate parameters for the algorithm under the limitation of the application scenario of the real-world engineering problems. In summary, the tri-stage design can certainly improve the performance of the algorithm, but an adaptive stage transition mechanism is urgently needed to deal with these limitations.

4.6. Practical application

When solving real-world engineering problems, the algorithm needs more flexibility and robustness due to the increase in objectives and the difficulty of constraints. Therefore, the potential value of the algorithm is mainly reflected in its ability to solve complex real-world engineering problems. We have conducted experimental research with the comparison algorithms on 7 real-world engineering problems. The selected problems come from different fields of science and engineering, which are vibrating platform design [58], disc brake design [59],

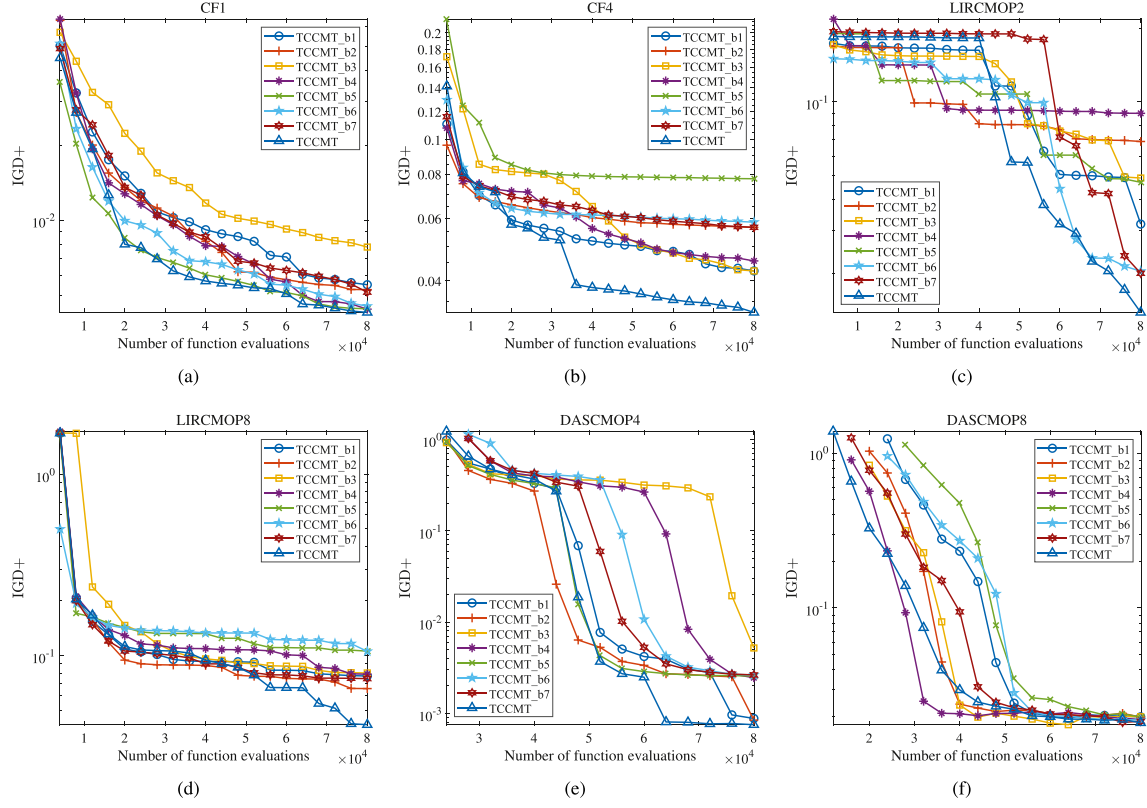
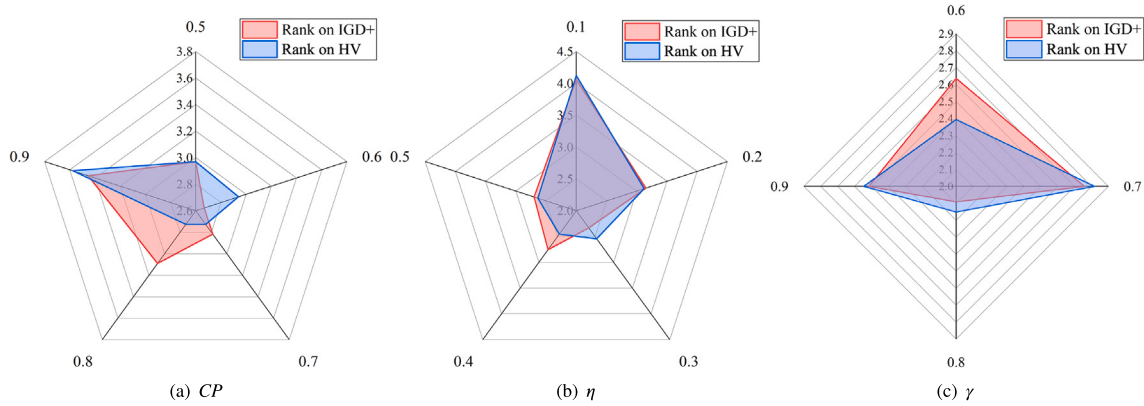


Fig. 10. The convergence curves of IGD+ for ablation study.

Fig. 11. IGD+ and HV rank of CP , η , and γ under different values obtained by Friedman test.

car side impact design [60], simply supported i-beam design [61], bulk carrier design [62], reactor network design [63], and synchronous optimal pulse-width modulation of 3-level inverters [64]. Since the IGD+ indicator measures the performance of the algorithm by calculating the Euclidean distance between the approximate PF and the true PF, the true PF is unknown in real-world engineering problems. Therefore, only the HV indicator is used. The results are shown in the Table 8. The Wilcoxon rank sum test with a significance level of 0.05 is used in the experiment. The best results of each problem are highlighted in bold and blue. It can be seen that TCCMT achieves the best result on four problems. In addition, the Friedman aligned test results of the experiment are shown in the Fig. 12. TCCMT achieves the best rank on all problems. Due to the characteristics of real-world engineering problems, algorithms need to obtain the optimal results under limited computing resources. Compared with EMC MO and CMEGL, the two CMOEAs based on EMT, TCCMT can

select the best auxiliary task to co-evolve through a collaborative competition mechanism. Therefore, TCCMT can reasonably use computing resources, which enables it to take advantage of these real-world engineering problems. Moreover, the constraints of real-world engineering problems are stricter, which makes the feasible regions in the search space relatively small. CMOEAs based on unconstrained population assistance, such as CCMO and EMC MO, may fall into local optimum due to insufficient guidance of feasible information in the auxiliary population. The proposed constraint adaptive regression strategy can dynamically decide whether to consider the constraint information, which can improve the exploration ability of the auxiliary population so as to avoid this problem. It is worth noting that different application scenarios often require different trade-offs. Therefore, decision makers usually need a set of Pareto optimal solution sets with good diversity to support dynamic needs. TCCMT can effectively maintain diversity with the help of

Table 8
HV results of ten comparison algorithms on 7 real-world engineering problems.

Problem	m	n	CCMO	cDPEA	BiCo	EMCMO	CMEGL	MTCMO	DEST	SC_EMT	CSEMT	TCCMT
Vibrating platform design	2	5	2.8871e-1 (1.43e-1) =	2.9128e-1 (1.38e-1) =	3.3655e-1 (8.00e-2) =	3.2960e-1 (9.66e-2) =	3.9257e-1 (8.41e-2) =	3.9277e-1 (5.00e-2) =	1.3706e-1 (9.97e-2) =	3.9293e-1 (1.47e-5) +	3.9172e-1 (3.10e-3) =	3.9251e-1 (9.83e-4) =
Disc brake design	2	4	4.3432e-1 (1.04e-3) =	4.3478e-1 (3.51e-4) =	4.3495e-1 (1.13e-4) =	4.3413e-1 (1.08e-4) =	4.3490e-1 (8.68e-5) =	4.3491e-1 (1.59e-3) =	4.3378e-1 (7.81e-4) =	4.3492e-1 (8.22e-4) =	4.3467e-1 (8.22e-4) =	4.3498e-1 (1.07e-4) =
Car side impact design	3	7	2.6011e-2 (5.20e-5) =	2.5999e-2 (5.35e-5) =	2.6141e-2 (4.18e-5) +	2.6028e-2 (3.92e-5) =	2.6063e-2 (4.71e-5) =	2.6027e-2 (4.71e-5) =	2.5752e-2 (4.71e-5) =	2.6098e-2 (3.40e-5) =	2.6021e-2 (4.49e-5) =	2.6093e-2 (3.70e-5) =
Simply supported I-beam design	2	4	5.5556e-1 (3.24e-3) =	5.5541e-1 (1.94e-3) =	5.5742e-1 (1.36e-3) +	5.5633e-1 (3.32e-3) =	5.5571e-1 (2.15e-3) =	5.5485e-1 (3.31e-3) =	5.5161e-1 (2.80e-3) =	5.5688e-1 (1.68e-3) =	5.5568e-1 (3.09e-3) =	5.5563e-1 (3.21e-3) =
Bulk carrier design	3	6	4.0229e-1 (1.23e-2) =	2.9888e-1 (8.35e-2) =	2.6925e-1 (3.84e-2) =	3.9760e-1 (1.58e-2) =	3.9107e-1 (2.43e-2) =	4.0603e-1 (6.78e-3) =	1.9190e-1 (5.60e-2) =	4.0002e-1 (1.68e-2) =	3.0895e-1 (4.56e-2) =	4.0652e-1 (3.27e-3) =
Reactor network design	2	6	3.1415e-1 (1.77e-1) =	6.3036e-1 (2.59e-1) =	2.5163e-1 (1.05e-1) =	3.6959e-1 (1.73e-1) =	4.2919e-1 (1.60e-1) =	3.3074e-1 (1.62e-1) =	NaN (NaN) =	2.7715e-1 (1.45e-1) =	5.3709e-1 (1.76e-1) =	7.5183e-1 (2.93e-1) =
Synchronous optimal pulse-width modulation of 3-level inverters	2	25	5.2303e-1 (1.25e-1) =	5.8414e-1 (8.30e-2) =	5.4331e-1 (1.13e-1) =	5.4186e-1 (9.70e-2) =	5.6372e-1 (1.07e-1) =	6.1707e-1 (9.63e-2) =	NaN (NaN) =	5.2398e-1 (1.15e-1) =	NaN (NaN) =	6.2761e-1 (1.49e-1) =
+/-=			0/4/3	0/4/3	2/2/3	0/5/2	0/3/4	0/3/4	0/7/0	1/2/4	0/3/4	

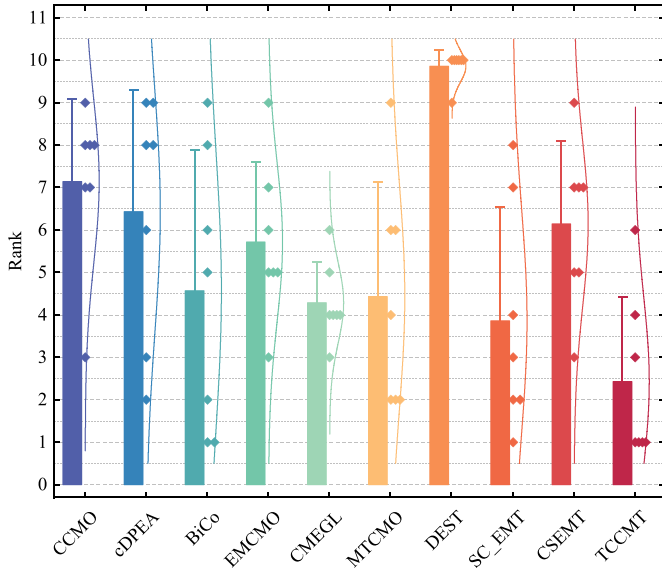


Fig. 12. The box plots of average HV rank obtained by the Friedman test on real-world engineering problems.

the double angle enhancement strategy, showing good practical application values. The above discussion shows that the proposed TCCMT obtains the powerful performance in solving real-world engineering problems.

5. Conclusion

In this paper, we propose a CMOEA based on a collaborative competition multitasking framework (TCCMT) to solve the unnecessary waste of computing resources. Three related optimization tasks and three evolutionary stages are constructed in TCCMT. During the evolution process, only the dominant one between the two auxiliary tasks is selected to co-evolve with the main task. The two auxiliary tasks determine the advantage and disadvantage in a collaborative competition manner. To be specific, while maintaining the competitive relationship, the auxiliary tasks can cooperate in a mixed parent manner to guide the evolution of each other. Meanwhile, different strategies are implemented to satisfy the exploration and exploitation in each stage. To enhance the ability of auxiliary tasks, they customize the constrained adaptive regression strategy and the double angle enhancement strategy, respectively. The constrained adaptive regression strategy can be dynamically adjusted according to the proportion of feasible solutions in the population, which improves the exploration ability. The double angle enhancement strategy makes effective use of high-quality infeasible solutions through the angle-based dominance principle and angle-based environmental selection, thereby maintaining the diversity of population. Compared with the nine most advanced CMOEAs on 33

test problems, the results show that TCCMT achieves the best IGD+ results on 28 test problems and the best HV results on 25 test problems. The Wilcoxon signed-ranks test results demonstrate that TCCMT is significantly different from other comparison algorithms. The Friedman aligned test results indicate that TCCMT achieves the best rank on all test problems. In addition, the experimental results on 7 real-world engineering problems show that TCCMT achieves the best HV result on 4 problems.

In the future, there are still some noteworthy research directions. Firstly, the selection of dominant auxiliary task is only based on the contribution of auxiliary tasks, but some potential values are not considered, such as high-quality infeasible solutions in the population. Thus, a more perfect competitive mechanism and task effectiveness evaluation indicator should be designed. It is expected to comprehensively evaluate the value of auxiliary tasks and realize the dynamic optimization of the selection. Furthermore, TCCMT has insufficient processing ability on some high-dimensional complex problems, which need specific auxiliary tasks to provide assistance. Therefore, more diversified auxiliary tasks should be nested in the competitive mechanism to provide a comprehensive auxiliary effect for different problem characteristics. Finally, TCCMT divides the evolution process into three stages, but the control of each stage depends on the preset parameters. Since different problems require different evolution times at each stage, selecting a suitable parameter for all problems is difficult. It is necessary to develop an adaptive stage transition mechanism to avoid the influence of parameters on the algorithm and enhance the adaptability.

CRedit authorship contribution statement

Xinyu Feng: Resources, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Qianlong Dang:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Data curation. **Xiaochuan Gao:** Writing – review & editing, Project administration, Methodology, Investigation. **Yanghui Wu:** Visualization, Software, Resources. **Lifei Zheng:** Resources, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported in part by the [National Nature Science Foundation of China](#) under Grant 12501711, in part by the [China Postdoctoral Science Foundation](#) under Grant 2024M762632, in part by the [Natural Science Foundation of Shaanxi](#) under Grant 2024JC-YBQN-0687, in part by the [Social Science Foundation of Shaanxi](#) under Grant 2024R027, and in part by the [Fundamental Research Funds of the Central Universities of China](#) under Grant 2452023013.

Data availability

No data were used for the research described in the article.

References

- [1] G. Huang, M. Hu, X. Yang, X. Wang, Y. Wang, F. Huang, A review of constrained multi-objective evolutionary algorithm-based unmanned aerial vehicle mission planning: key techniques and challenges, *Drones* 8 (7) (2024).
- [2] F. RivasDávalos, M.R. Irving, An approach based on the strength pareto evolutionary algorithm 2 for power distribution system planning, in: *Evolutionary Multi-Criterion Optimization*, Springer Berlin Heidelberg, 2005, pp. 707–720.
- [3] J. Liang, H. Lin, C. Yue, X. Ban, K. Yu, Evolutionary constrained multi-objective optimization: a review, *Vicinagearth* 1 (1) (2024) 5.
- [4] Q. Ye, W. Wang, G. Li, R. Dai, A self-organizing assisted multi-task algorithm for constrained multi-objective optimization problems, *Inf. Sci.* 664 (2024) 120339.
- [5] Y. Li, X. Feng, H. Yu, A constrained multiobjective evolutionary algorithm with the two-archive weak cooperation, *Inf. Sci.* 615 (2022) 415–430.
- [6] J. Zhou, Y. Zhang, P.N. Suganthan, Dual population approximate constrained pareto front for constrained multiobjective optimization, *Inf. Sci.* 648 (2023) 119591.
- [7] Y. Hua, Q. Liu, K. Hao, Y. Jin, A survey of evolutionary algorithms for multi-objective optimization problems with irregular pareto fronts, *IEEE/CAA J. Autom. Sin.* 8 (2) (2021) 303–318.
- [8] Y. Zhang, Y. Tian, H. Jiang, X. Zhang, Y. Jin, Design and analysis of helper-problem-assisted evolutionary algorithm for constrained multiobjective optimization, *Inf. Sci.* 648 (2023) 119547.
- [9] Bilal, M. Pant, H. Zaheer, L. Garcia-Hernandez, A. Abraham, Differential evolution: a review of more than two decades of research, *Eng. Appl. Artif. Intell.* 90 (2020) 103479.
- [10] A. Lambora, K. Gupta, K. Chopra, Genetic algorithm- a literature review, in: 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (comitcon), 2019, pp. 380–384.
- [11] Z. Xia, Y. Liu, J. Lu, J. Cao, L. Rutkowski, Penalty method for constrained distributed quaternion-variable optimization, *IEEE Trans. Cybern.* 51 (11) (2020) 5631–5636.
- [12] Z. Ma, Y. Wang, Shift-based penalty for evolutionary constrained multiobjective optimization and its application, *IEEE Trans. Cybern.* 53 (1) (2021) 18–30.
- [13] T. Ray, H.K. Singh, A. Isaacs, W. Smith, Infeasibility driven evolutionary algorithm for constrained optimization, *Constraint-Handling In Evolutionary Optimization* (2009) 145–165.
- [14] B. Wang, H. Li, J. Li, Y. Wang, Composite differential evolution for constrained evolutionary optimization, *IEEE Trans. Syst. Man Cybern. Syst.* 49 (7) (2018) 1482–1495.
- [15] Q. Gu, Q. Wang, N.N. Xiong, S. Jiang, L. Chen, Surrogate-assisted evolutionary algorithm for expensive constrained multi-objective discrete optimization problems, *Complex Intell. Syst.* 8 (4) (2022) 2699–2718.
- [16] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, E. Goodman, Push and pull search for solving constrained multi-objective optimization problems, *Swarm Evol. Comput.* 44 (2019) 665–679.
- [17] W. Luo, X. Yu, G.G. Yen, Dual-stage and dual-population cooperative evolutionary algorithm for solving constrained multiobjective problems, *Appl. Soft Comput.* 160 (2024) 111703.
- [18] O. Cuate, A. Ponsich, L. Uribe, S. ZapotecasMartínez, A. Lara, O. Schütze, A new hybrid evolutionary algorithm for the treatment of equality constrained mops, *Mathematics* 8 (1) (2020).
- [19] F. Ming, W. Gong, L. Wang, L. Gao, Constrained multiobjective optimization via multitasking and knowledge transfer, *IEEE Trans. Evol. Comput.* 28 (1) (2024) 77–89.
- [20] J. Zhou, Z. Liu, Y. Li, C. Lu, L. Gao, Voting enabled dynamic constraint handling portfolios for evolutionary multi-/many-objective optimization, *Inf. Sci.* 694 (2025) 121700.
- [21] K. Qiao, K. Yu, B. Qu, J. Liang, H. Song, C. Yue, An evolutionary multitasking optimization framework for constrained multiobjective optimization problems, *IEEE Trans. Evol. Comput.* 26 (2) (2022) 263–277.
- [22] K. Qiao, K. Yu, B. Qu, J. Liang, H. Song, C. Yue, H. Lin, K.C. Tan, Dynamic auxiliary task-based evolutionary multitasking for constrained multiobjective optimization, *IEEE Trans. Evol. Comput.* 27 (3) (2022) 642–656.
- [23] K. Qiao, J. Liang, Z. Liu, K. Yu, C. Yue, B. Qu, Evolutionary multitasking with global and local auxiliary tasks for constrained multi-objective optimization, *IEEE/CAA J. Autom. Sin.* 10 (10) (2023) 1951–1964.
- [24] G. Li, Q. Zhang, Z. Wang, Evolutionary competitive multitasking optimization, *IEEE Trans. Evol. Comput.* 26 (2) (2022) 278–289.
- [25] X. Chu, F. Ming, W. Gong, Competitive multitasking for computational resource allocation in evolutionary constrained multi-objective optimization, *IEEE Trans. Evol. Comput.* (2024).
- [26] J. Liang, X. Ban, K. Yu, B. Qu, K. Qiao, C. Yue, K. Chen, K.C. Tan, A survey on evolutionary constrained multiobjective optimization, *IEEE Trans. Evol. Comput.* 27 (2) (2023) 201–221.
- [27] A. Homaifar, C.X. Qi, S.H. Lai, Constrained optimization via genetic algorithms, *Simulation* 62 (4) (1994) 242–253.
- [28] J.A. Joines, C.R. Houck, On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GA's, in: *Proceedings of the First IEEE Conference on Evolutionary Computation*, vol. 2, 1994, pp. 579–584.
- [29] K. Rasheed, An adaptive penalty approach for constrained genetic-algorithm optimization, in: *Proceedings of the Third Annual Genetic Programming Conference*, Citeseer, 1998, pp. 584–590.
- [30] K. Deb, A. Pratap, S. Agarwal, T.A.M.T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Trans. Evol. Comput.* 6 (2) (2002) 182–197.
- [31] Z. Fan, W. Li, X. Cai, K. Hu, H. Lin, H. Li, Angle-based constrained dominance principle in MOEA/D for constrained multi-objective optimization problems, in: 2016 IEEE Congress on Evolutionary Computation (CEC), 2016, pp. 460–467.
- [32] T. Takahama, S. Sakai, Constrained optimization by the ϵ constrained differential evolution with gradient-based mutation and feasible elites, in: 2006 IEEE International Conference on Evolutionary Computation, 2006, pp. 1–8.
- [33] J. Liang, L. Zhang, K. Yu, B. Qu, F. Shang, K. Qiao, Interactive niching-based two-stage evolutionary algorithm for constrained multiobjective optimization, *Swarm Evol. Comput.* 83 (2023) 101402.
- [34] Y. Tian, T. Zhang, J. Xiao, X. Zhang, Y. Jin, A coevolutionary framework for constrained multiobjective optimization problems, *IEEE Trans. Evol. Comput.* 25 (1) (2020) 102–116.
- [35] Z. Liu, B. Wang, K. Tang, Handling constrained multiobjective optimization problems via bidirectional coevolution, *IEEE Trans. Cybern.* 52 (10) (2021) 10163–10176.
- [36] K. Qiao, J. Liang, K. Yu, C. Yue, H. Lin, D. Zhang, B. Qu, Evolutionary constrained multiobjective optimization: scalable high-dimensional constraint benchmarks and algorithm, *IEEE Trans. Evol. Comput.* (2023).
- [37] M. Ming, A. Trivedi, R. Wang, D. Srinivasan, T. Zhang, A dual-population-based evolutionary algorithm for constrained multiobjective optimization, *IEEE Trans. Evol. Comput.* 25 (4) (2021) 739–753.
- [38] K. Qiao, Z. Chen, B. Qu, K. Yu, C. Yue, K. Chen, J. Liang, A dual-population evolutionary algorithm based on dynamic constraint processing and resources allocation for constrained multi-objective optimization problems, *Expert Syst. Appl.* 238 (2024) 121707.
- [39] Y. Wang, C. Hu, F. Ming, Y. Li, W. Gong, L. Gao, A diversity-enhanced tri-stage framework for constrained multi-objective optimization, *IEEE Trans. Evol. Comput.* (2024) 1.
- [40] F. Ming, W. Gong, L. Wang, C. Lu, A tri-population based co-evolutionary framework for constrained multi-objective optimization problems, *Swarm Evol. Comput.* 70 (2022) 101055.
- [41] S. Zhao, X. Hao, L. Chen, Y. Feng, Multi-population constrained multi-objective evolutionary algorithm based on knowledge transfer, in: 2024 6th International Conference on Data-Driven Optimization of Complex Systems (DOCS), 2024, pp. 214–220.
- [42] S. Liu, Z. Wang, Q. Lin, J. Chen, Coevolutionary multitasking for constrained multiobjective optimization, *Swarm Evol. Comput.* 91 (2024) 101727.
- [43] X. Zhong, X. Yao, K. Qiao, D. Gong, A multitasking-based constrained multi-objective evolutionary algorithm with forward and backward stages, *IEEE Trans. Emerg. Top. Comput. Intell.* 8 (5) (2024) 3474–3488.
- [44] K. Yu, L. Wang, J. Liang, H. Wang, K. Qiao, T. Liang, Constraint subsets-based evolutionary multitasking for constrained multiobjective optimization, *Swarm Evol. Comput.* 86 (2024) 101531.
- [45] K. Qiao, J. Liang, K. Yu, X. Ban, C. Yue, B. Qu, P.N. Suganthan, Constraints separation based evolutionary multitasking for constrained multi-objective optimization problems, *IEEE/CAA J. Autom. Sin.* 11 (8) (2024) 1819–1835.
- [46] Z. He, H. Liu, A dual-population auxiliary multiobjective coevolutionary algorithm for constrained multiobjective optimization problems, *Appl. Soft Comput.* 163 (2024) 111827.
- [47] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: improving the strength pareto evolutionary algorithm for multiobjective optimization, in: *Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems. Proceedings of the EUROGEN'2001. Athens. Greece, September 19-21, 2001*.
- [48] B. Wang, H. Li, J. Li, Y. Wang, Composite differential evolution for constrained evolutionary optimization, *IEEE Trans. Syst. Man Cybern. Syst.* 49 (7) (2019) 1482–1495.
- [49] Q. Bao, M. Wang, G. Dai, X. Chen, Z. Song, S. Li, A dual-population based bidirectional coevolution algorithm for constrained multi-objective optimization problems, *Expert Syst. Appl.* 215 (2023) 119258.
- [50] Q. Zhang, A. Zhou, S. Zhao, P.N. Suganthan, S. Tiwari, Multiobjective optimization test instances for the CEC 2009 special session and competition, *Mech. Eng.* (2008).
- [51] Z. Fan, W. Li, X. Cai, H. Huang, Y. Fang, Y. You, J. Mo, C. Wei, E. Goodman, An improved epsilon constraint-handling method in MOEA/D for cmops with large infeasible regions, *Soft Comput.* (2017).
- [52] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, E. Goodman, Difficulty adjustable and scalable constrained multiobjective test problem toolkit, *Evol. Comput.* 28 (3) (2020) 339–378.
- [53] Y. Tian, R. Cheng, X. Zhang, Y. Jin, Platemo: a MATLAB platform for evolutionary multi-objective optimization [educational forum], *IEEE Comput. Intell. Mag.* 12 (4) (2017) 73–87.
- [54] Z. Yang, T. Yao, Y. Jiang, J. Zhang, X. Zhang, Indexes-based and partial restart-based constrained multiobjective optimization, *IEEE Trans. Evol. Comput.* (2024) 1.
- [55] A.P. Guerreiro, C.M. Fonseca, L. Paquete, The hypervolume indicator: computational problems and algorithms, *ACM Comput. Surv.* 54 (6) (2021).
- [56] E. Zitzler, J. Knowles, L. Thiele, Quality assessment of pareto set approximations, *Multiobjective Optimization: Interactive And Evolutionary Approaches* (2008) 373–404.
- [57] I. Triguero, S. González, J.M. Moyano, S. García, J. Alcaládez, J. Luengo, A. Fernández, M.J. Del Jesús, L. Sánchez, F. Herrera, Keel 3.0: an open source software for multi-stage analysis in data mining, *Int. J. Comput. Intell. Syst.* 10 (1) (2017) 1238–1249.

- [58] S. Narayanan, S. Azarm, On improving multiobjective genetic algorithms for design optimization, *Struct. Optim.* 18 (1999) 146–155.
- [59] A. Osyczka, S. Kundu, A genetic algorithm-based multicriteria optimization method, *Proc. 1st World Congr. Struct. Multidiscip. Optim.* (1995) 909–914.
- [60] H. Jain, 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.* 18 (4) (2014) 602–622.
- [61] H. Huang, Y. Gu, X. Du, An interactive fuzzy multi-objective optimization method for engineering design, *Eng. Appl. Artif. Intell.* 19 (5) (2006) 451–460.
- [62] M.G. Parsons, R.L. Scott, Formulation of multicriterion design optimization problems for solution with scalar numerical optimization methods, *J. Ship Res.* 48 (1) (2004) 61–76.
- [63] H.S. Ryoo, N.V. Sahinidis, Global optimization of nonconvex nlp and minlp with applications in process design, *Comput. Chem. Eng.* 19 (5) (1995) 551–566.
- [64] A.K. Rathore, J. Holtz, T. Boller, Synchronous optimal pulsewidth modulation for low-switching-frequency control of medium-voltage multilevel inverters, *IEEE Trans. Ind. Electron.* 57 (7) (2010) 2374–2381.