



A coevolution algorithm based on two-staged strategy for constrained multi-objective problems

Chaodong Fan^{1,2,3,4} · Jiawei Wang^{1,2} · Leyi Xiao^{3,4,5} · Fanyong Cheng³ · Zhaoyang Ai⁶ · Zhenhuan Zeng¹

Accepted: 19 February 2022 / Published online: 7 April 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

Constrained Multiobjective Problem (CMOP) is widely used in engineering applications, but the current constrained Multiobjective Optimization algorithms (CMOEA) often fails to effectively balance convergence and diversity. For this purpose, a two-stage co-evolution constrained multi-objective optimization evolutionary algorithm (TSC-CMOEA) is presented to solve constrained multi-objective optimization problems. This method divides the search process into two phases: in the first stage, the synchronous co-evolution is used, and the population corresponding to the help problem and the population corresponding to the raw problem cooperate with each other and share the offspring to produce better solutions, so as to quickly cross the infeasible region and approach the Pareto front; The second stage discards the help problem when it fails and maintains only the evolution of the main population to save computing resources and enhance convergence. The combination of synchronous co-evolution and staged strategy allows the population to traverse infeasible regions more efficiently and converge quickly to feasible and non-dominant regions. The test results on benchmark CMOPs show that the convergence and population distribution of TSC-CMOEA is significantly better than those of NSGA-II, NSGA-III, C-MOEA/D, PPS, ToP and CCMO.

Keywords Constraint handling technique · Multiobjective optimization · Coevolution · Constrained multi-objective evolutionary algorithms

1 Introduction

In scientific research and engineering practice, many practical optimization problems ultimately boil down to solving constrained multi-objective optimization problems, such as WEB server placement problem [1], microgrid optimization scheduling problem [2], and near-space communication deployment problem [3], etc. This type of constrained multi-objective problems (CMOP) can be defined as the following form:

$$\begin{cases} \text{minimize } \mathbf{F}(\mathbf{x}) & = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))^T \\ \text{subject to } g_i(\mathbf{x}) & \geq 0, i = 1, \dots, q \\ & h_j(\mathbf{x}) = 0, j = 1, \dots, p \\ & \mathbf{x} \in \mathbb{R}^n \end{cases} \quad (1)$$

$F(\mathbf{x})$ represents m objective functions that need to be optimized at the same time. $f_i(\mathbf{x})$ represents the i – th objective function, $g_i(\mathbf{x})$ and $h_j(\mathbf{x})$ represent inequality constraints and equality constraints, respectively, and q and p represent the number of inequality constraints and equality constraints, respectively. Due to the complexity of CMOP, traditional mathematical methods are difficult to obtain ideal results. The Evolutionary Algorithm (EA), as a type of optimization method developed by simulating biological evolution theory has many advantages in solving CMOP problems. Therefore, constrained multi-objective optimization algorithms have received widespread attention.

In the past two decades, multi-objective evolutionary algorithms (MOEAs) have achieved good results in solving various multi-objective problems [4]. When using MOEAs to solve CMOP, it is necessary to adopt appropriate constraint processing techniques to solve equality constraints and inequality constraints. Almost all constraint handling methods are to define an overall constraints violation function and minimize it [5]. Early MOEAs usually treated constraints and goals equally. For example: Penalty function method [6], which adds the constraint to the objective

✉ Leyi Xiao
xiaolytkx@163.com

function by multiplying the constraint with a penalty factor, thereby transforming the constrained optimization problem into a special unconstrained problem; The dynamic penalty method [7] is to construct the penalty factor as a function that changes with the number of iterations or iteration time, so as to dynamically adjust the problem. If the penalty factor changes according to the feedback information during the search process, it is called an adaptive penalty method [8]. However, it is difficult to achieve a good balance between constraints and goals by adjusting the penalty factor [9]. In order to solve the deficiencies in the penalty function constraint processing method, T.P. Runarsson et al. proposed a random sorting constraint processing method [10], which use a process similar to bubble sorting to deal with constrained optimization problems. Based on the random parameter θ , we can judge whether the current solution is based on the objective or the constraint. When θ is 0, the objective based sorting rule is used for random sorting; When θ is 1, constraint based sorting rules are used for random sorting. Decomposition based MOEA usually does not consider the Pareto domination, but it also uses a similar idea to solve the problem of judging infeasible solution in the aspect of constraint processing [11]. Specifically, when updating the solution of the weight vector, consider the degree of constraint violation of the solution, and prefer the solution with less constraint violation [12]. In the algorithm, when the angle between two feasible solutions is large, the relationship between the two solutions is considered as non-dominant, and the algorithm in [13] considers the minimization of objective values and the minimization of constraint violations as a two-objective optimization problem to be optimized.

However, there are two problems with the above methods: 1) There may be many non-zero local minima in the overall constraint violation function, which may lead to infeasible regions being searched. 2) The possible domains in the search space may be discontinuous, making it difficult to move from one area to another [14, 15]. In summary, existing multi-objective evolutionary algorithms (MOEAs) have encountered difficulties in solving CMOP. To solve these difficulties, many advanced constrained multi-objective optimization algorithms (CMOEAs) have been proposed. For example, the two archive coevolution algorithm (CTAEA) [16], which saves both convergence and diversity archives, improve the convergence of the population while maintaining the distribution of the population, and ultimately search for a non-dominant feasible region. PPS designs and uses a push-pull search method, which first pushes the population to the unconstrained Pareto front (PF), and then pulls the population into the feasible region. However, both the CTAEA's diversity archive throughout the search process and the PPS locates the push phase of

unconstrained PF has done a lot of useless work in the infeasible regions [17, 18].

In this regard, this paper proposes a staged coevolution algorithm to solve the above two dilemmas and improve search efficiency. The main contributions of this paper are as follows:

1. Aiming at the problem that the population is trapped in the local optimal feasible region, a new coevolution method is proposed. This method keeps the relative synchronization between the help population and the main population, makes the auxiliary population become more efficient auxiliary main population, and helps the population find the ideal non-dominated feasible region.
2. Aiming at the problem of waste of computing resources, a two-stage optimization strategy is designed. When the help population plays a positive guiding role, the coevolution stage is maintained; when the help population cannot play a role, the stage is changed in time so that the algorithm can search more efficiently.
3. A new algorithm, TSC-CMOEA, is obtained by combining these new coevolution methods with two-stage optimization strategies. Tests on multiple test sets show that TSC-CMOEA has great potential in solving CMOPs.

The main content of the rest of this paper is organized as follows: Section 2 is the basic knowledge of CMOPs, and introduces the shortcomings of algorithms such as CTAEA and CCMO in solving CMOP problems. In order to overcome the shortcomings of existing algorithms, an improved algorithm, TSC-CMOEA, is proposed in Section 3. The validity of TSC-CMOEA and its essential reasons is analysed in depth. Section 4 is the experimental part, which compares and analyses TSC-CMOEA with several CMOEAs which currently perform well, such as PPS, ToP and CTAEA. Section 5 is the conclusion of this article.

2 Related work

When solving most CMOPs, the feasibility of the solution is often considered before the convergence, so some MOEAs give priority to the feasibility of the solution in the dominance relationship. In NSGA-II, Deb et al. proposed a dominance-based constraint processing method called CDP [11]. This method embeds the feasibility criterion into Pareto dominance, in which feasible solutions dominate infeasible solutions, and solutions with lower constraint violations dominate solutions with higher constraint violations. Specifically, first, the overall

constraint violation degree of each solution is obtained by calculating formula (2).

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

Where $g_i(x)$ and $h_j(x)$ are the i -th inequality constraint and the j -th equality constraint of CMOP. If any of the following conditions is met, the solution x is said to dominate another solution y : 1) If $CV(x) = 0$ and $CV(y) = 0$, any $f_i(x) \leq f_i(y)$, and there is at least one dimension $f_i(x) < f_i(y)$; 2) $CV(x) < CV(y)$. CDP constraint processing technology can be used in other MOEAs based on the advantages of Pareto.

ϵ Constraint processing technology [12] relaxes the definition of feasibility on the basis of CDP constraint processing technology, and sets the condition for judging feasible and infeasible solutions, that is, the overall constraint violation degree to a value can be dynamically adjusted ϵ . When the overall constraint violation degree of a solution is less than ϵ , the current solution is considered to be a feasible solution [12].

The most recent CMOEAs can be divided into four categories according to the constraint processing technology:

The first type of CMOEAs is derived from the rigor of constrained single-objective optimization, which mainly relies on penalty functions, such as MOSOS [19] and $I_{\epsilon+}$ NSGA-II [8]. This idea is simple and easy to implement, but its performance is sensitive to the setting of the penalty factor with the corresponding penalty function. Too large a penalty factor can lead to indifference to an infeasible solution, while too small a penalty factor can lead to acceptance of an infeasible solution.

The second type of CMOEAs, they mainly use feasibility information. For example, NSGA-II [11] proposes a constraint dominance principle (CDP) in NSGA-II. When comparing a pair of feasible solutions, this is exactly the same as the classical Pareto advantage, while infeasible solutions are compared according to their constraint conflicts. CDP is one of the most popular constraint processing technologies in EMO due to its simplicity and efficiency. In addition, Fan et al. proposed an angle-based CDP algorithm (MOEA/D-ACDP) [13], which utilizes classical Pareto advantage to compare infeasible solutions within a given Angle. However, the constraint information used by these methods is relatively simple, and they often need to be used together with other methods to achieve better results.

The third type of CMOEAs, such as CM2M2 [20], is designed to increase selection pressure with useful infeasible solutions. CM2M2 proposed a CMOP evolutionary

algorithm based on two groups of weight vectors. In particular, it maintains a set of “infeasible” weight directions to facilitate constrained multi-objective optimization.

The fourth class is based on ϵ functions, which use ϵ as a tolerance threshold for constraint conflicts containing infeasible solutions (e.g. ϵ de-HP [12]). However, the setting of ϵ can affect convergence to a large extent, where a large ϵ value exerts a large selection pressure on convergence, while a small ϵ value fine-tunes the local area.

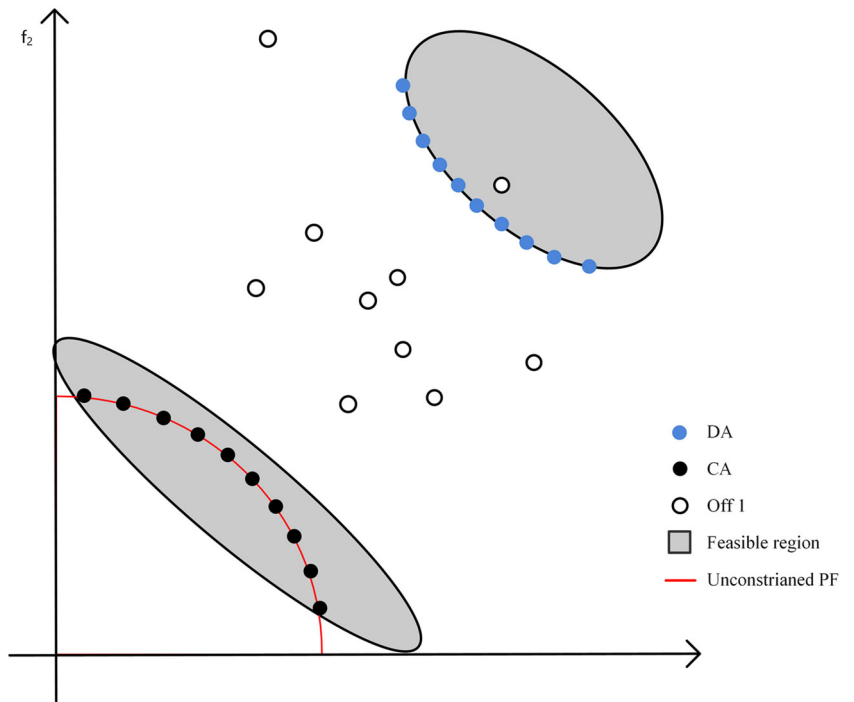
Coevolution technology has achieved good results in solving many difficult optimization problems such as multi-objective optimization problems [21], dynamic optimization problems [22], and large-scale optimization problems [23]. But in terms of handling constraints, coevolution technology is still in its infancy. [17] In [24], the coevolutionary genetic algorithm and differential evolution algorithm for solving constrained optimization problems are respectively proposed [25]. These two algorithms evolve into multiple populations at the same time, and each population is assigned an independent penalty factor to balance constraints and objectives [26]. In [27], the algorithm decomposes the constraints and assigns a population to each constraint, where each population first tries to satisfy its assigned constraints, and then evolves another constraint from other populations.

Literature [28] proposed a multi-objective particle swarm optimization algorithm. The algorithm uses one population to store available particles and another population to store unavailable particles. Among them, available particles evolve into Pareto optimal, and unavailable particles evolve into available particles. In addition, once the infeasible solution becomes feasible, it can be transferred to the feasible solution group. In [17], the differential coevolution framework is used to solve constrained multi-objective problems. In this paper, the constrained multi-objective problem with m constraints is transformed into m constrained single-objective optimization problems, and m populations used to solve the constrained single-objective optimization problem are evolved to solve the original CMOP.

For more advanced coevolutionary algorithms, they put more emphasis on the coordination of multiple populations. Both CTAEA and CCMO are among the most advanced coevolutionary algorithms.

CTAEA [28] is an evolutionary algorithm that evolves two populations. In CTAEA, the convergence-oriented archiving (CA) evolves into optimization constraints and objectives, and the diversity-oriented archiving (DA) evolves into optimization objectives only. At the same time, the two populations cooperate with each other in mating selection and environmental selection. Usually CTAEA selects one parent from CA and another parent from DA to produce offspring. Therefore, most offspring are located

Fig. 1 Population distribution when problems occur in CTAEA



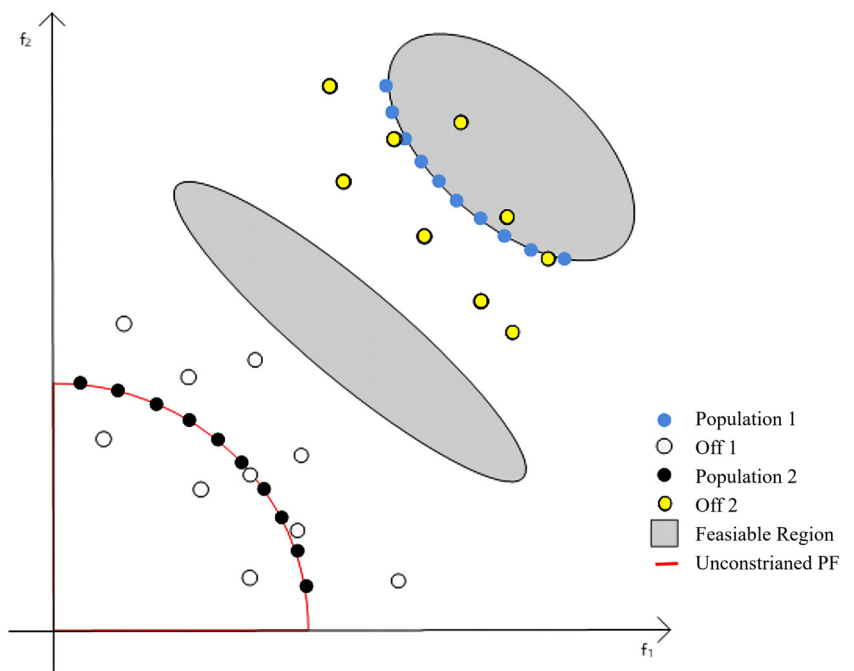
between CA and DA. The offsprings produced by this method not only have the diversity of DA, but also have the convergence of CA to a certain extent.

However, this method also has some problems as shown in Fig. 1: (1) From the distribution of the white dots Off1 population in Fig. 1, it can be seen that most individuals of Off1 are located in the infeasible area, and these offspring do not have good diversity or convergence. (2) Combining

the black CA population and the white DA population in Fig. 1, it can be seen that Off1 is mainly distributed in the middle area between the CA and the DA, and cannot effectively update the CA to have better diversity. Therefore, CTAEA ultimately cannot find a sufficient number of feasible and well-convergent solutions [10].

CCMO [17] is a coevolutionary algorithm that simultaneously calculates two objectives problems-the original prob-

Fig. 2 Population distribution when problems occur in CCMO



lem and the auxiliary problem. Usually, the original problem in CCMO is solved by Population 1. The auxiliary problem is defined as the original problem that removes the constraints and is solved by Population 2. The two populations have no other associations except for sharing offspring, which is called weak association. Specifically, Population 1 of CCMO generates its own offspring 1 through a predefined method, and Population 2 also generates its own offspring 2 through a predefined method. Two progeny populations are mixed with select individuals through their respective environments to finally solve the problem. This method is effective in the early stage of the algorithm, and the offspring produced by the auxiliary population can often help the population quickly cross the infeasible region [29].

But in the later stage of the algorithm, this method often has some problems, as showed in Fig. 2. (1) The offspring populations produced by Population 1 and Population 2 are near their respective populations. Offspring 1 and Offspring 2 seem to have nothing to do with each other and cannot cooperate well. (2) The calculation of Population 2 is not helpful to the main population, which is almost a waste of computing resources. As some recent studies have revealed, most existing MOEAs have encountered difficulties in obtaining a set of feasible solutions to a set of discrete and widely distributed CMOPs. In particular, the limitations of current coevolutionary algorithms are mainly due to the lack of interaction and waste of computing resources in the synergy between populations [18, 30, 31].

3 The proposed algorithm

3.1 Algorithm framework

In order to alleviate the current CMOEA's difficulty in effectively balancing convergence and diversity and waste of computing resources when processing complex CMOPs, this paper proposes a constrained multi-objective optimization algorithm (TSC-CMOEA) based on two-stage coevolution. As described in Algorithm 1, the TSC-CMOEA proposed in this paper is divided into two stages. Lines 6-21 (marked by '▲') in Algorithm 1 are the first stage of TSC-CMOEA. In this stage, the coevolution constraint processing technology is used to make the population quickly approach the real PF. The main purpose of this stage is to quickly cross a large number of discontinuous infeasible regions with the help of help problems; When the help problem can no longer help the population evolution well, TSC-CMOEA switches to the second stage (as shown in lines 23-28 marked by '■' in Algorithm 1). This stage uses CDP constraint processing technology to better complete the search. This stage is mainly to increase the

diversity of the population while reducing unnecessary function calculation requests.

Algorithm 1 Procedure of the proposed TSC-CMOEA.

Require: f_{raw} (raw CMOP), f_{help} (helper problem), N (population size)
Ensure: Population 1 (final population)
 Population 1 \leftarrow RandomInitialization(N)
 Population 2 \leftarrow RandomInitialization(N)
Evaluate Population1 by f_{raw}
Evaluate Population1 by f_{help}
while termination conditions are not met **do**
 if TSC-CMOEA is in its first phase & ($\text{rk} < \delta$) **then**
 ▲Parent1 \leftarrow Select $N/2$ parents from Population1 by the mating selection of MOEA
 ▲Parent2 \leftarrow Select $N/2$ parents from Population2 by the mating selection of MOEA
 ▲Off1 \leftarrow Generate $N/2$ offsprings based on Parent1 by the operators of MOEA;
 ▲Off2 \leftarrow Generate $N/2$ offsprings based on Parent2 by the operators of MOEA;
 ▲Population1 \leftarrow Population1 \cup Off1 \cup Off2
 if Problem interaction condition is satisfied **then**
 ▲Population2 \leftarrow Population2 \cup Off1 \cup Off2
 else
 ▲Population2 \leftarrow Population2 \cup Off2
 end if
 ▲Evaluate Population1 by f_{raw}
 ▲Evaluate Population1 by f_{help}
 ▲Population1 \leftarrow Select N solutions from Population1 by the environmental selection of MOEA
 ▲Population2 \leftarrow Select N solutions from Population2 by the environmental selection of MOEA
 else
 ■ ★Parent1 \leftarrow Select $N/2$ parents from Population1 by the mating selection of MOEA
 ■ Off1 \leftarrow Generate $N/2$ offsprings based on Parent1 by the operators of MOEA;
 ■ Population1 \leftarrow Population1 \cup Off1
 ■ Evaluate Population1 by f_{raw}
 ■ Population1 \leftarrow Select N solutions from Population1 by the environmental selection of MOEA
 end if
end while

3.2 The first stage

This section will specifically describe the evolutionary process of the first stage. It can be seen from Algorithm 1: In the first stage, the used dominance relationship and crowding distance are used as mating selection strategies, and two parent sets, Parent 1 and Parent 2, are selected from Population 1 and Population 2, respectively. Then, the two parent sets are generated by simulated binary crossover and polynomial mutation operators to generate offspring populations, Offspring 1 and Offspring 2. After that, Population 1 combined with the offspring population

Offspring 1 used an environmental selection strategy based on fitness to select individuals. Population 2 is combined with Offspring 1 every 10 iterations, and with Offspring 2 in other iterations, and then individuals are selected using an environment selection strategy based on fitness.

Such a population combination strategy can make the main population and the auxiliary population relatively synchronized. It is worth noting that the individual selection strategy used in TSC-CMOEA is different from the commonly used selection strategy used in NSGA-II. Specifically, the environment selection strategy used in TSC-CMOEA is similar to that proposed in SPEA2. First, determine the solution set storing all the solutions dominated by x and the solution set S_x storing all the solutions dominated by x , and find the $\lfloor \sqrt{2n} \rfloor$ nearest neighbor of x in the population. Then calculate an $fit(x)$ for each individual, $fit(x)$ is defined as following

$$fit(\mathbf{x}) = \sum_{y \in S_x} \|R_y\| + \frac{1}{\text{dist}(\mathbf{x}, \mathbf{x}') + 2} \quad (3)$$

Where the first component of $fit(x)$ calculates the total number of solutions that dominate x , and the second component calculates the reciprocal of the Euclidean distance from x to its $\lfloor \sqrt{2n} \rfloor$ -th nearest neighbor. The smaller the value of $fit(x)$, the better the quality of the solution. This individual selection strategy considers the distribution of the population more. Population 2 uses a special offspring population combination strategy to keep the original problem and the auxiliary problem as synchronous as possible. This operation can improve the utilization efficiency of the auxiliary problem. It should be noted that the solutions in Population 1 are all calculated from the raw problem f_{raw} , and Population 1 is also the final output result of the first stage.

The solution in Population 2 is calculated from the help problem f_{help} derived from the raw problem. Usually the help problem f_{help} is selected as the original problem f_{raw} without considering constraints.

3.3 The second stage

From the 23th line (marked by '★') of Algorithm 1, we can see that the input of the second stage is the output of the first stage, Population 1. At this time, Population 1 is very close to the real PF, but due to the coevolution of some characteristics, the diversity is slightly insufficient. As can be seen from lines 23-28 (marked by '▲') of the code in Algorithm 1, the processing of the entire problem in the second stage is similar to the processing of the main population in the first stage. Two Parent sets, Parent 1 and Parent 2, are selected from Population 1 by using dominant relationship and crowding distance as mating selection strategies. Then, the two parent sets

are used to generate progeny populations, Offspring 1, through the simulated binary crossover and polynomial mutation operators, respectively. After that, Population 1 combined with the offspring population Offspring 1 uses the environmental selection strategy to generate a new Population 1. In the second stage, because the help problem can no longer support the main population well, we do not calculate Population 2 in the second stage to save the calculation.

The main steps of TSC-CMOEA include two stages, and it is particularly critical to completing the stage conversion at the right time. The strategy for the switching phase is as follows: $r_k < \delta$, where r_k represents the rate of change of the lowest point in the population at the lowest l offspring, and δ is a minimum value set by the user (set $\delta = 1e - 3$ in this article). Note that δ is an empirical value. r_k is the change rate of the ideal point in the population. Through a large number of experiments, we found that when the δ value is set to a low value ($1e - 3$), the switching algorithm stage is effective, but the δ value does not tend to zero. We believe that when the population change rate is less than a certain threshold, is the best time to switch algorithm. If you let $\delta = 0$, you don't switch between the two phases, and the second phase doesn't work. The rate of change of the lowest point in the last l generation is defined as follows:

$$r_k = \frac{\|z^k - z^{k-l}\|}{\max\{\|z^{k-l}\|, 1e - 6\}} \quad (4)$$

Where z^k is the ideal point of generation k and z^{k-l} is the ideal point of generation $(k - l)$. In (4) $1e - 6$ is to ensure that the denominator is not zero, so its value can be closer to 0.

In general, the proposed TSC-CMOEA evolves Population 1 in the first stage to solve the raw problem, and at the same time evolves Population 2 to solve the help problem. The offspring produced by the two populations are shared conditionally. When the phase transition conditions are met, the proposed TSC-CMOEA enters the second phase. The second stage is based on Population 1 evolved in the first stage. CDP constraint processing technology cooperates with NSGA-II to evolve Population 1 to solve CMOPs.

It is worth noting that in the first stage, Population 1 may often fall into a narrow feasible area, and sharing the offspring of Population 2 can help it jump out of the local optimum. At the same time, because f_{help} is usually simpler than f_{raw} , the solution in Population 2 usually has better convergence than the solution in Population 1. Therefore, descendants generated by Population 2 are likely to improve the convergence of Population 1 after the sharing operation.

The coevolution of the two populations can greatly improve the speed and accuracy of the algorithm's convergence, but it also comes with some minor problems.

For example, because there is a mapping relationship between the PF of the help problem and the real PF, and this relationship can sometimes only lead the population to find a part of the PF, so that the diversity of the population is greatly lost. Therefore, when the coevolution constraint processing technology cannot complete the task well, it is very necessary to switch to an effective constraint processing technology in time. After Population 1 ends the first stage, the auxiliary problem cannot help the original problem in the second stage; by entering the second stage, the main population makes full use of some infeasible solutions to expand the diversity of the population.

3.4 Analysis of TSC-CMOEA

In the first stage, we deal with both populations simultaneously, strengthening the tie between the populations by mixing the two offspring populations. Population 1 is a constrained population, and feasible solutions are fully considered to ensure the feasibility of the algorithm. Population 2 is an unconstrained population, which usually approaches the unconstrained PF preferentially than population 1, and the offspring generated by population 2 has higher convergence. Therefore, the introduction of two-population strategy not only retains feasible solutions well, but also achieves higher convergence, making the performance of the two populations complementary.

In the second stage, the algorithm to the end, there is some distance between the unconstrained PF and the constrained PF, this time to maintain double population strategy is not cost-effective, because of the population are near the real PF for local search, population 2 unconstrained PF for population 1 help has been very small. So, in the second stage, we removed population 2 and maintained only one population, with the main purpose of conducting a more detailed local search on the results of the first stage. This operation directly reduces the evaluation times of the function and indirectly improves the performance of the algorithm.

Also, at the end of the algorithm, the population 1 with high convergence obtained in the first stage is used to conduct local search with a method similar to dominance, which directly improves the performance of the algorithm. Therefore, the performance of the algorithm is improved by the two strategies proposed by us.

In order to analyze the effectiveness and mechanism of TSC-CMOEA, we show the early, middle and last population distribution of TSC-CMOEA on the MW9 in Fig. 3, and compare the distribution results with CTAEA and CCMO.

As showed in Fig. 3(a c), Population 1 of CCMO is responsible for the convergence of early populations

relatively concentrated, and Population 2 auxiliary problems are widely distributed. Some individuals in the final population cannot skip the infeasible region ahead.

It can be seen from Fig. 3(d f) that for CTAEA, DA is a population that maintains diversity, and CA is a population that improves convergence. In the early stage, due to the influence of DA, the diversity of CA performed well, but the convergence was insufficient, and the performance of the offspring was also the same. In the middle stage, since the DA population only considers the objectives and ignores constraints, the DA population is distributed on the unconstrained PF. However, CTAEA usually selects a parent from CA, and then selects a parent from DA to produce offspring, so most offspring are located between CA and DA. Due to this offspring generation strategy, the offspring produced do not have good feasibility or good convergence, and the CA cannot be updated to achieve better diversity. Therefore, in the end, CTAEA cannot perfectly find a sufficient number of good enough solutions.

Figure 3(g i) shows that TSC-CMOEA and CCMO behave similarly in the early and middle stage, because TSC-CMOEA originated from CCMO. For the early days of TSC-CMOEA, Population 1 considers constraints, and Population 2 does not consider constraints. Thanks to Population 2, the distribution and convergence of the overall population are very good. It can be seen from the figure that the two populations are synchronized with each other, and the parents of TSC-CMOEA are selected from Population 1 and Population 2, which makes some offspring around Population 1 and Population 2, and the surrounding offspring can enhance the diversity of Population 1. When TSC-CMOEA enters the second stage, it pays more attention to convergence. Therefore, the final TSC-CMOEA results are relatively better than CTAEA and CCMO.

4 Experimental research

4.1 Test sets

In order to evaluate the performance of TSC-CMOEA, we selected five test sets of C-DTLZ, DC-DTLZ, MW, LIRCMOP and DOC as the test objects. C-DTLZ and DC-DTLZ are traditional constraint test problems, MW is a test problem based on constraint construction method and distance function design, LIRCMOP is a large-scale infeasible region test problem, and DOC is a test problem that includes decision constraints and objectives constraints. A detailed description of these test sets, including mathematical definitions and properties, can be found in their original papers [30–33].

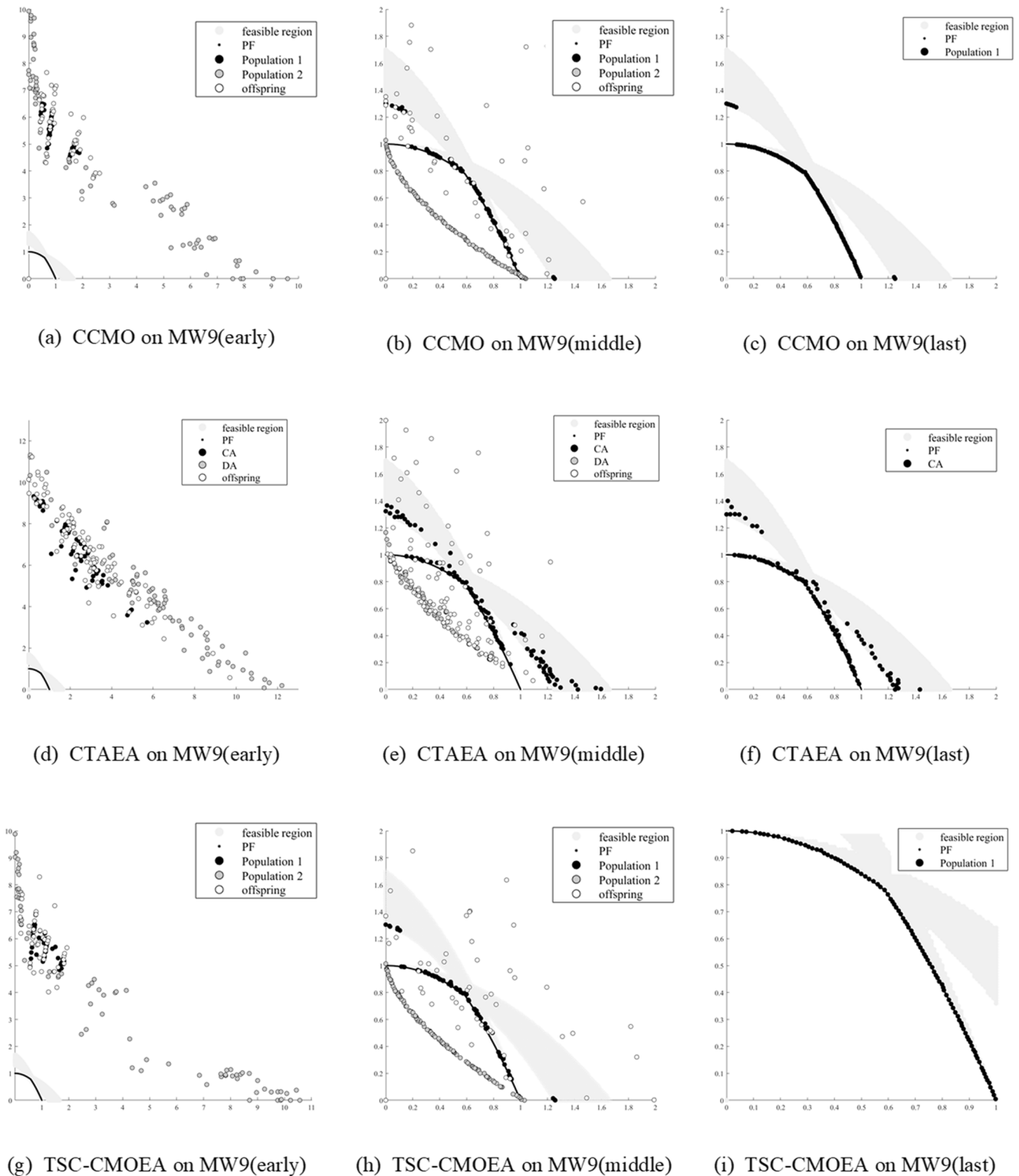


Fig. 3 The population distribution of CCMO, CTAEA and TSC-CMOEA in the early, middle and late stages on the MW9

4.2 Parameter setting

All the algorithms participating in the comparison have some common parameter settings. Table 1 shows the

parameter settings of the proposed algorithm and the comparison algorithm. For the two-objective problem, the population size is set to 100, and for the three objective problem, the population size is set to 105. Termination

Table 1 Parameters set of CCMO, NSGA-II, NSGA-III, C-MOEA/D, PPS, TOP, and TSC-CMOEA

Algorithms	Test set	Mutation probability	Distribution index	Operator	CR	F	Other parameters	Function evaluations
TSC-CMOEA	C-DTLZ, MW			SBX	0.9	-	-	100000
	LIRCMOP,DOC			DE	1.0	0.5	-	300000
CCMO	C-DTLZ,MW			SBX	0.9	-	-	100000
	LIRCMOP,DOC			DE	1.0	0.5	-	300000
NSGA-II	C-DTLZ,MW			SBX	0.9	-	-	100000
	LIRCMOP,DOC	1/D						300000
NSGA-III	C-DTLZ,MW	(D denotes the	20	SBX	0.9	-	-	100000
	LIRCMOP,DOC	number of decision variables)						300000
CMOEA/D	C-DTLZ,MW			SBX	0.9	-	-	100000
	LIRCMOP,DOC							300000
PPS	C-DTLZ,MW			DE	1.0	0.5	$\alpha = 0.95, \tau = 0.1,$	100000
	LIRCMOP,DOC						$cp = 2, l = 20$	300000
ToP	C-DTLZ,MW			DE	1.0	0.5	$P_f = 1/3,$	100000
	LIRCMOP,DOC						$\delta = 0.2$	300000

conditions of all MOEAs participating in the comparison are set so that the total number of evaluations of all population functions reaches the specified number, and the setting is large enough to ensure that each MOEAs can converge. For constrained DTLZ and MW problems, the number of function evaluations is set to 100,000, while for LIR-CMOP and DOC problems, the number of function evaluations is set to 300,000. Since the DE operator has excellent performance in solving complex problems, for LIRCMOP and DOC problems, the DE operator and polynomial mutation are used to generate progeny, and for the remaining problems, the SBX operator and polynomial mutation are used to generate progeny. For the SBX operator, the crossover probability and distribution index are set to 0.9 and 20, respectively. For the DE operator, the crossover rate and scale factor are set to 1.0 and 0.5, respectively. For the polynomial mutation operator, the mutation probability and distribution index are set to $P_m = 1/n$ (n is the number of decision variables) and 20, respectively. The parameters of all comparison algorithms are set according to the suggestions in the original paper [4, 9, 11, 17, 23, 30].

4.3 Performance indicators

The goal of CMOPS is to find a set of uniformly distributed feasible solution sets that are as close to the real PF as possible. In order to quantitatively compare the performance of each algorithm, we used two popular metrics in

the experiment-Inverse Generation Distance (IGD). Their definitions are as following:

$$\begin{cases} IGD(P^*, A) = \frac{y_{y^* \in P^*} \sum(y^*, A)}{\|P^*\|} \\ d(y^*, A) = \min_{y \in A} \left\{ \sqrt{\sum_{i=1}^m (y_i^* - y_i)^2} \right\} \end{cases} \quad (5)$$

Where P^* represents a set of representative solutions in the real PF, and A is the approximate PF realized by CMOEA. m represents the number of objectives. y is a solution in A , y_i is the i -th objective value of y , y^* is the solution with the closest Euclidean distance to y in P^* , and y_i^* is the y -th objective function value.

HV reflects the proximity of the undominated solution set implemented by CMOEA to the true PF, a larger HV means that the corresponding undominated set is closer to the true PF.

$$HV(S) = VOL \left(\bigcup_{x \in S} [f_1(x), z_1^r] \times \dots [f_m(x), z_m^r] \right) \quad (6)$$

where $VOL(-)$ is the Leberger measure, m denotes the number of objectives, and $z^r = (z_1^r, \dots, z_m^r)^T$ is the user-defined reference point in the objective space. For each CMOP, the reference point is 1.2 times the distance from the lowest point of the true PF. It is worth noting that larger values of high pressure may indicate better performance in terms of diversity and/or convergence.

Table 2 Mean and standard deviation of the IGD values obtained by NSGA-II, NSGA-III, C-MOEA/D, PPS, TOP, and TSC-CMOEA for 30 independent runs on the constrained DTLZ test suite and the MW test suite

Problem	CTAEA	CCMO	TSC-CMOEA2	TSC-CMOEA
C1_DTLZ1	2.3178e-2 (2.40e-4) –	2.0041e-2 (1.32e-4) –	1.9978e-2 (1.30e-4) –	1.9895e-2 (1.32e-4)
C1_DTLZ3	3.4215e-1 (1.47e+0) –	5.3559e-2 (5.87e-4) ≈	5.3521e-2 (4.70e-4) ≈	5.3279e-2 (5.53e-4)
C2_DTLZ2	5.6577e-2 (1.05e-3) –	4.2770e-2 (6.88e-4) –	4.2892e-2 (5.96e-4) –	4.2367e-2 (5.11e-4)
C3_DTLZ4	1.1276e-1 (2.18e-3) +	1.1965e-1 (1.36e-1) ≈	1.1962e-1 (1.36e-1) ≈	1.9448e-1 (2.58e-1)
DC1_DTLZ1	1.5226e-2 (3.04e-4) –	1.1389e-2 (7.15e-5) ≈	1.1419e-2 (8.95e-5) ≈	1.1378e-2 (9.71e-5)
DC1_DTLZ3	1.3026e-1 (3.56e-3) –	1.1391e-1 (1.03e-3) ≈	1.1422e-1 (1.13e-3) –	1.1352e-1 (7.70e-4)
DC2_DTLZ1	2.3257e-2 (1.43e-4) –	2.0147e-2 (1.77e-4) –	2.0150e-2 (1.14e-4) –	2.0030e-2 (1.48e-4)
DC2_DTLZ3	1.7087e-1 (2.00e-1) –	5.3203e-2 (4.19e-4) –	5.3231e-2 (4.49e-4) –	5.2774e-2 (3.65e-4)
DC3_DTLZ1	9.3515e-3 (2.26e-4) –	6.8641e-3 (8.62e-5) ≈	6.8787e-3 (7.46e-5) ≈	6.8583e-3 (6.86e-5)
DC3_DTLZ3	1.7122e-1 (3.33e-3) –	1.5887e-1 (1.68e-3) ≈	1.5884e-1 (1.52e-3) ≈	1.5877e-1 (1.59e-3)
MW1	2.0250e-3 (7.04e-5) –	1.6169e-3 (1.38e-5) –	1.6138e-3 (1.28e-5) –	1.5980e-3 (9.19e-6)
MW2	1.8517e-2 (1.04e-2) ≈	1.8705e-2 (6.65e-3) ≈	1.7319e-2 (6.39e-3) ≈	1.8739e-2 (6.79e-3)
MW3	4.9647e-3 (2.93e-4) ≈	4.8251e-3 (2.69e-4) ≈	4.8752e-3 (2.40e-4) ≈	4.9514e-3 (5.17e-4)
MW4	4.6500e-2 (2.66e-4) –	4.0926e-2 (4.18e-4) –	4.0924e-2 (4.01e-4) –	4.0440e-2 (3.68e-4)
MW5	1.3777e-2 (3.03e-3) –	1.4200e-3 (1.39e-3) ≈	9.9185e-4 (1.16e-3) ≈	9.6387e-4 (1.35e-3)
MW6	9.9780e-3 (5.69e-3) +	3.5039e-2 (8.29e-2) ≈	1.7975e-2 (1.07e-2) ≈	1.9129e-2 (9.97e-3)
MW7	6.9005e-3 (8.27e-4) –	4.6105e-3 (3.26e-4) –	4.6303e-3 (3.69e-4) –	4.4270e-3 (3.06e-4)
MW8	5.4844e-2 (3.07e-3) –	5.1784e-2 (2.35e-2) ≈	4.5744e-2 (2.89e-3) ≈	4.5570e-2 (2.67e-3)
MW9	8.6074e-3 (4.97e-4) –	2.8012e-2 (1.29e-1) ≈	5.7915e-3 (7.36e-3) ≈	4.3940e-3 (2.19e-4)
MW10	1.2437e-2 (7.91e-3) +	3.9047e-2 (3.52e-2) ≈	3.0207e-2 (3.31e-2) ≈	3.0653e-2 (2.85e-2)
MW11	1.4449e-2 (2.66e-3) –	6.0670e-3 (2.22e-4) –	6.0381e-3 (2.40e-4) –	5.9255e-3 (2.47e-4)
MW12	7.9017e-3 (8.15e-4) –	4.9671e-2 (1.71e-1) –	2.6509e-2 (1.18e-1) ≈	4.7267e-3 (1.03e-4)
MW13	3.0520e-2 (1.61e-2) +	6.6541e-2 (3.70e-2) ≈	6.8959e-2 (3.08e-2) ≈	6.3608e-2 (3.59e-2)
MW14	1.1140e-1 (3.75e-3) –	9.7689e-2 (1.94e-3) –	9.8669e-2 (2.01e-3) –	9.7083e-2 (2.47e-3)
LIRCPOP1	1.9526e-1 (9.02e-2) –	1.2150e-1 (7.60e-2) ≈	1.1855e-1 (6.40e-2) ≈	9.6198e-2 (6.14e-2)
LIRCPOP2	1.3644e-1 (3.84e-2) ≈	9.8489e-2 (6.20e-2) ≈	9.4549e-2 (6.64e-2) ≈	1.0334e-1 (5.74e-2)
LIRCPOP3	2.2606e-1 (8.08e-2) –	1.0615e-1 (6.39e-2) ≈	1.1203e-1 (6.69e-2) ≈	9.8033e-2 (5.55e-2)
LIRCPOP4	2.2487e-1 (8.76e-2) –	1.2923e-2 (7.43e-3) +	1.2917e-1 (5.32e-2) ≈	1.3318e-1 (6.38e-2)
LIRCPOP5	1.1615e+0 (2.35e-1) –	2.0833e-2 (1.67e-2) –	7.8872e-3 (6.12e-4) –	7.4428e-3 (5.41e-4)
LIRCPOP6	1.2520e+0 (2.86e-1) –	6.4187e-3 (3.85e-4) +	7.0491e-3 (5.13e-4) ≈	6.9969e-3 (1.10e-3)
LIRCPOP7	1.3239e-1 (2.87e-2) –	7.4545e-3 (5.21e-4) ≈	7.3281e-3 (2.98e-4) ≈	7.3071e-3 (2.66e-4)
LIRCPOP8	7.4634e-1 (6.43e-1) –	7.2858e-3 (2.45e-4) ≈	7.2412e-3 (1.79e-4) ≈	7.2908e-3 (2.29e-4)
LIRCPOP9	3.5913e-1 (7.55e-2) –	8.3791e-2 (2.65e-2) –	1.0852e-2 (3.12e-3) ≈	1.7788e-2 (2.10e-2)
LIRCPOP10	2.6609e-1 (1.30e-1) –	6.9414e-3 (1.02e-3) –	6.8818e-3 (5.33e-4) –	6.4508e-3 (3.41e-4)
LIRCPOP11	1.5440e-1 (2.23e-2) –	4.7263e-3 (1.10e-3) –	2.9089e-3 (1.71e-4) –	2.7702e-3 (1.05e-4)
LIRCPOP12	9.7824e-2 (2.13e-2) –	1.1425e-2 (7.01e-3) –	4.1169e-3 (3.32e-4) –	3.8654e-3 (4.88e-4)
LIRCPOP13	5.4893e-2 (7.64e-4) +	7.8422e-2 (1.35e-3) +	1.1952e-1 (1.75e-3) –	1.1794e-1 (1.83e-3)
LIRCPOP14	5.5616e-2 (1.05e-3) +	6.0956e-2 (6.90e-4) +	1.0395e-1 (1.91e-3) –	1.0272e-1 (1.56e-3)
DOC1	5.0625e+2 (2.70e+2) –	5.8383e-3 (6.42e-4) –	5.8157e-3 (6.29e-4) ≈	5.4714e-3 (3.43e-4)
DOC2	NaN (NaN)	2.3586e-2 (6.11e-2) –	1.8901e-2 (4.01e-2) –	4.9571e-3 (2.90e-3)
DOC3	NaN (NaN)	4.8586e+2 (4.45e+2) +	5.7524e+2 (4.68e+2) ≈	7.7070e+2 (4.75e+2)
DOC4	2.3515e+2 (2.44e+2) –	2.4100e-2 (3.34e-3) ≈	2.3959e-2 (7.28e-3) ≈	2.3486e-2 (7.91e-3)
DOC5	NaN (NaN)	8.7488e+0 (3.37e+1) ≈	2.8028e+1 (5.56e+1) ≈	1.5403e+1 (4.33e+1)
DOC6	2.6728e+1 (1.67e+1) –	2.6107e-2 (8.30e-2) –	3.2305e-2 (1.30e-1) –	3.2305e-3 (6.67e-4)
DOC7	NaN (NaN)	2.5550e-2 (1.26e-1) –	8.6632e-3 (3.36e-2) –	2.4263e-3 (9.72e-5)
DOC8	4.8122e+2 (1.89e+2) –	7.2006e-2 (5.82e-3) –	7.1983e-2 (5.29e-3) –	6.4513e-2 (3.18e-3)
DOC9	9.0351e-1 (4.19e-1) –	8.2003e-2 (1.06e-2) –	8.1497e-2 (8.87e-3) –	7.4340e-2 (8.71e-3)
+ / - / ≈	6/34/3	5/21/21	0/21/26	

Table 3 Mean and standard deviation of the IGD values obtained by NSGA-II, NSGA-III, C-MOEA/D, PPS, TOP, and TSC-CMOEA for 30 independent runs on the constrained DTLZ test suite and the MW test suite

Problem	NSGAI	NSGAIII	CMOEA/D	PPS	ToP	TSC-CMOEA
C1_DTLZ1	2.5006e-1 (1.33e-1) -	2.0455e-2 (9.52e-5) -	2.0517e-2 (3.79e-5) -	2.6432e-2 (8.42e-4) -	2.1081e-1 (0.00e+0) ≈	1.9895e-2 (1.32e-4)
C1_DTLZ3	6.6965e+0 (3.02e+0) -	5.2231e+0 (3.75e+0) -	2.4535e+0 (3.48e+0) -	6.0828e-1 (2.02e+0) -	9.9496e-1 (2.42e+0) -	5.3279e-2 (5.53e-4)
C2_DTLZ2	8.5235e-2 (1.17e-1) -	4.8126e-2 (3.34e-4) -	4.9310e-2 (2.91e-5) -	5.6323e-2 (2.14e-3) -	6.0195e-2 (5.14e-3) -	4.2367e-2 (5.11e-4)
C3_DTLZ4	1.2775e-1 (4.87e-3) +	1.4251e-1 (1.91e-1) +	9.1352e-2 (2.45e-5) +	1.5839e-1 (8.58e-2) +	1.4286e-1 (6.09e-3) +	1.9448e-1 (2.58e-1)
DC1_DTLZ1	1.5415e-2 (3.12e-3) -	1.4203e-2 (1.51e-4) -	1.8769e-2 (3.81e-3) -	3.2826e-2 (5.93e-2) -	3.1151e-2 (5.97e-2) -	1.1378e-2 (9.71e-5)
DC1_DTLZ3	1.4910e-1 (7.83e-2) -	1.2702e-1 (1.36e-3) -	1.2837e-1 (6.01e-2) -	2.1096e-1 (2.36e-1) -	1.5319e+0 (2.42e+0) -	1.1352e-1 (7.70e-4)
DC2_DTLZ1	NaN (NaN)	1.3862e-1 (5.60e-2) -	8.2812e-2 (7.24e-2) -	2.8177e-2 (8.03e-4) -	NaN (NaN)	2.0030e-2 (1.48e-4)
DC2_DTLZ3	NaN (NaN)	5.6074e-1 (0.00e+0) ≈	5.6943e-1 (2.43e-2) -	1.7013e-1 (2.01e-1) -	NaN (NaN)	5.2774e-2 (3.65e-4)
DC3_DTLZ1	1.5326e-1 (8.85e-2) -	1.1193e-1 (7.81e-2) -	7.6685e-2 (7.83e-2) -	1.1048e-1 (2.40e-1) -	2.5688e+0 (3.01e+0) -	6.8583e-3 (6.86e-5)
DC3_DTLZ3	1.8238e+0 (5.84e-1) -	1.5349e+0 (4.59e-1) -	1.1548e+0 (4.89e-1) -	7.8919e-1 (1.21e+0) -	6.4293e+0 (3.42e+0) -	1.5877e-1 (1.59e-3)
MW1	NaN (NaN)	8.0453e-3 (1.40e-2) -	4.3710e-3 (2.22e-3) -	3.1436e-3 (3.91e-4) -	5.2905e-1 (2.25e-1) -	1.5980e-3 (9.19e-6)
MW2	1.9385e-1 (2.46e-1) -	2.4371e-2 (9.11e-3) -	2.2002e-2 (6.98e-3) -	1.2186e-1 (8.32e-2) -	1.9179e-1 (1.26e-1) -	1.8739e-2 (6.79e-3)
MW3	4.4796e-1 (3.33e-1) -	4.2867e-2 (1.68e-1) -	8.5662e-3 (1.58e-2) -	6.3160e-3 (3.93e-4) -	5.0320e-1 (3.90e-1) -	4.9514e-3 (5.17e-4)
MW4	NaN (NaN)	4.1268e-2 (6.90e-5) -	4.2088e-2 (1.67e-3) -	6.1430e-2 (5.40e-3) -	6.5717e-1 (3.50e-2) -	4.0440e-2 (3.68e-4)
MW5	NaN (NaN)	2.2218e-1 (3.24e-1) -	2.5181e-3 (2.96e-3) -	4.2660e-1 (3.63e-1) -	NaN (NaN)	9.6387e-4 (1.35e-3)
MW6	7.6209e-1 (2.14e-1) -	4.2948e-2 (8.60e-2) -	1.6725e-2 (8.29e-3) ≈	4.6275e-1 (3.18e-1) -	7.1488e-1 (3.75e-1) -	1.9129e-2 (9.97e-3)
MW7	3.9128e-1 (2.65e-1) -	7.5276e-2 (1.61e-1) -	1.9604e-2 (8.16e-2) -	5.5560e-3 (4.86e-4) -	6.8661e-2 (1.12e-1) -	4.4270e-3 (3.06e-4)
MW8	6.6404e-1 (2.34e-1) -	6.8782e-2 (7.89e-2) -	5.0763e-2 (3.46e-3) -	1.6351e-1 (5.69e-2) -	6.8923e-1 (3.17e-1) -	4.5570e-2 (2.67e-3)
MW9	NaN (NaN)	6.3817e-2 (1.77e-1) -	8.6238e-3 (3.40e-3) -	4.9174e-2 (1.34e-1) -	5.5340e-1 (3.38e-1) -	4.3940e-3 (2.19e-4)
MW10	6.7046e-1 (7.83e-2) -	1.3045e-1 (1.22e-1) -	7.0199e-2 (1.07e-1) -	3.6310e-1 (2.00e-1) -	NaN (NaN)	3.0653e-2 (2.85e-2)
MW11	7.9797e-1 (1.72e-1) -	5.2288e-1 (2.89e-1) -	2.6197e-1 (3.36e-1) -	7.4551e-3 (3.67e-4) -	6.2993e-1 (1.92e-1) -	5.9255e-3 (2.47e-4)
MW12	NaN (NaN)	4.4915e-2 (1.13e-1) -	4.8064e-3 (1.58e-4) -	1.1189e-2 (8.69e-3) -	8.0373e-1 (1.74e-1) -	4.7267e-3 (1.03e-4)
MW13	6.7719e-1 (5.44e-1) -	3.3233e-1 (4.16e-1) -	1.4450e-1 (2.61e-1) -	5.5115e-1 (3.88e-1) -	8.9748e-1 (5.04e-1) -	6.3608e-2 (3.59e-2)
MW14	3.4956e-1 (4.89e-1) -	1.3180e-1 (2.90e-3) -	2.1239e-1 (1.28e-3) -	1.3202e-1 (8.73e-3) -	3.4831e-1 (3.76e-1) -	9.7083e-2 (2.47e-3)
+ / - / ≈	1/16/0	1/22/1	1/22/1	1/23/0	1/18/1	

Table 4 Mean and standard deviation of the HV values obtained by NSGA-II, NSGA-III, C-MOEA/D, PPS, TOP, and TSC-CMOEA for 30 independent runs on DTLZ test suite and MW test suite

Problem	NSGAII	NSGAIII	CMOEA/D	PPS	TOP	SCCMO
C1_DTLZ1	3.5119e-1 (2.11e-1) –	8.3985e-1 (2.15e-3) –	8.4111e-1 (6.66e-4) ≈	8.1761e-1 (4.97e-3) –	4.3117e-1 (0.00e+0) ≈	8.4092e-1 (2.19e-3)
C1_DTLZ3	8.9136e-2 (2.03e-1) –	1.5908e-1 (2.51e-1) –	3.0027e-1 (2.73e-1) –	4.7183e-1 (1.31e-1) –	3.5356e-1 (2.04e-1) –	5.6077e-1 (8.52e-4)
C2_DTLZ2	4.7101e-1 (7.21e-2) –	5.0910e-1 (2.35e-3) –	5.1528e-1 (8.59e-5) –	4.9980e-1 (3.93e-3) –	4.6870e-1 (1.52e-2) –	5.1791e-1 (1.56e-3)
C3_DTLZ4	7.6596e-1 (3.12e-3) +	7.7657e-1 (6.69e-2) +	7.9583e-1 (5.37e-5) +	7.6176e-1 (2.42e-2) +	7.5574e-1 (4.88e-3) –	7.5666e-1 (8.73e-2)
DC1_DTLZ1	6.1008e-1 (1.16e-2) –	6.2927e-1 (1.77e-3) –	6.2555e-1 (1.28e-2) –	5.8583e-1 (7.61e-2) –	5.7126e-1 (7.51e-2) –	6.3283e-1 (6.35e-4)
DC1_DTLZ3	4.7417e-1 (4.87e-2) –	4.9213e-1 (8.55e-4) –	4.8814e-1 (4.46e-2) –	4.1615e-1 (1.28e-1) –	2.0942e-1 (1.90e-1) –	4.9800e-1 (8.23e-4)
DC2_DTLZ1	NaN (NaN)	5.4288e-1 (1.41e-1) –	6.8409e-1 (1.83e-1) –	8.0656e-1 (5.35e-3) –	NaN (NaN)	8.4244e-1 (4.92e-4)
DC2_DTLZ3	NaN (NaN)	8.0944e-3 (0.00e+0) ≈	7.4677e-3 (1.50e-3) –	4.1961e-1 (2.04e-1) –	NaN (NaN)	5.6042e-1 (1.15e-3)
DC3_DTLZ1	1.7757e-1 (1.81e-1) –	2.4572e-1 (2.13e-1) –	3.6449e-1 (2.10e-1) –	4.1273e-1 (2.05e-1) –	9.4736e-3 (5.19e-2) –	5.3616e-1 (1.22e-3)
DC3_DTLZ3	0.0000e+0 (0.00e+0) –	0.0000e+0 (0.00e+0) –	0.0000e+0 (0.00e+0) –	2.3810e-1 (1.89e-1) –	0.0000e+0 (0.00e+0) –	4.1090e-1 (1.56e-3)
MW1	NaN (NaN)	4.8097e-1 (1.56e-2) –	4.8614e-1 (4.35e-3) –	4.8723e-1 (1.21e-3) –	6.4791e-2 (9.16e-2) –	4.9009e-1 (4.46e-5)
MW2	3.9831e-1 (1.60e-1) –	5.4731e-1 (1.38e-2) –	5.5054e-1 (1.09e-2) –	4.2370e-1 (9.66e-2) –	3.5093e-1 (1.18e-1) –	5.5622e-1 (1.08e-2)
MW3	2.4758e-1 (1.77e-1) –	5.2157e-1 (1.00e-1) ≈	5.4164e-1 (1.28e-2) ≈	5.4331e-1 (7.10e-4) –	2.1629e-1 (2.11e-1) –	5.4419e-1 (9.93e-4)
MW4	NaN (NaN)	8.4155e-1 (6.93e-5) –	8.4080e-1 (1.42e-3) –	8.0206e-1 (1.12e-2) –	1.9018e-1 (5.04e-2) –	8.4218e-1 (3.78e-4)
MW5	NaN (NaN)	2.4475e-1 (1.01e-1) –	3.2364e-1 (1.25e-3) –	1.8692e-1 (1.12e-1) –	NaN (NaN)	3.2422e-1 (6.36e-4)
MW6	5.8546e-2 (6.18e-2) –	2.8655e-1 (3.67e-2) –	3.0718e-1 (1.16e-2) ≈	1.3226e-1 (8.02e-2) –	7.2174e-2 (7.28e-2) –	3.0390e-1 (1.37e-2)
MW7	2.4528e-1 (1.05e-1) –	3.8318e-1 (6.27e-2) –	4.0563e-1 (3.04e-2) –	4.1196e-1 (3.01e-4) –	3.6517e-1 (4.90e-2) –	4.1248e-1 (5.06e-4)
MW8	1.2250e-1 (9.23e-2) –	5.0925e-1 (3.97e-2) –	5.3471e-1 (1.42e-2) ≈	3.0173e-1 (9.23e-2) –	1.2528e-1 (1.38e-1) –	5.3524e-1 (1.24e-2)
MW9	NaN (NaN)	3.5480e-1 (9.75e-2) –	3.8824e-1 (4.71e-3) –	3.5621e-1 (7.42e-2) –	8.3039e-2 (1.17e-1) –	3.9894e-1 (1.59e-3)
MW10	1.0497e-1 (3.94e-2) –	3.6123e-1 (6.52e-2) –	3.9850e-1 (5.67e-2) –	2.4182e-1 (9.48e-2) –	NaN (NaN)	4.2495e-1 (2.21e-2)
MW11	2.5596e-1 (3.91e-2) –	3.1420e-1 (7.20e-2) –	3.8277e-1 (8.53e-2) –	4.4749e-1 (1.36e-4) +	2.8406e-1 (4.03e-2) –	4.4741e-1 (1.37e-3)
MW12	NaN (NaN)	5.7465e-1 (7.65e-2) –	6.0513e-1 (4.26e-4) +	5.9560e-1 (1.24e-2) –	2.1514e-2 (4.81e-2) –	6.0497e-1 (1.50e-4)
MW13	2.6615e-1 (1.29e-1) –	3.8259e-1 (7.08e-2) –	4.2836e-1 (4.97e-2) –	2.3861e-1 (1.25e-1) –	2.0305e-1 (1.19e-1) –	4.4907e-1 (1.75e-2)
MW14	3.7832e-1 (1.55e-1) –	4.6594e-1 (1.91e-3) –	4.4409e-1 (1.42e-3) –	4.4860e-1 (4.32e-3) –	3.6159e-1 (1.49e-1) –	4.7368e-1 (1.93e-3)
+ / - / ≈	1/16/0	1/21/2	2/18/4	2/22/0	0/19/1	

4.4 Effective test of algorithm improvement

In order to verify the effectiveness of TSC-CMOEA's improvement strategy, we compared it with CTAEA and CCMO. Table 2 shows the average and standard deviation of the IGD for 30 independent runs of CTAEA, CCMO, TSC-CMOEA and TSC-CMOEA2 on the DTLZ, MW, LIRCMOP and DOC test problem sets. As can be seen from the table, TSC-CMOEA achieved the best results in 35 of the 47 test problems, followed by CTAEA which achieved 7 best results, and finally CCMO only achieved 5 best results. Table 2 also shows the statistical results of the Bonferroni-corrected Friedman test at the 0.05 significance level [34]. It can be seen that TSC-CMOEA is significantly better than CTAEA on 36 issues and significantly better than CCMO on 21 issues. TSC-CMOEA shows better convergence on these problems, which are caused by the two-stage design of the algorithm, and the increase in the second stage strengthens the convergence of the population. TSC-CMOEA2 is a variant of the TSC-CMOEA without two-phase strategy. As can be seen from IGD value data in Table 2, TSC-CMOEA2 has some advantages over CCMO, but it is not very obvious. However, by comparing TSC-CMOEA2 with TSC-CMOEA, it can be found that TSC-CMOEA has obvious advantages. This shows that the two stage strategy can improve the performance of the algorithm more greatly.

At the same time, the two stage strategy mainly improves the local search performance of the algorithm, so the new coevolution method is also crucial for the population to reach the ideal region more efficiently.

Combining Tables 2 and 3 and the above analysis, we can see that the new coevolution method and the phased strategy have good results.

4.5 Experimental results of DTLZ and MW problems

In order to verify the performance of TSC-CMOEA, we select NSGA-II, NSGA-III, C-MOEA/D, PPS and ToP and other current algorithms that perform well in solving constrained multi-objective optimization problems as the comparison objects. All algorithms constrain the average and standard deviation of the IGD values obtained during 30 independent runs on the DTLZ test suite and the MW test suite. The IGD value of each question is calculated according to the method suggested in [35] for approximately 10,000 reference points sampled on the PF of the question. As showed in Table 3, the proposed TSC-CMOEA achieved the best results on 18 issues, and secondly, CMOEA/D achieved the best results on two issues. Then we computed the HV values of TSC-CMOEA and the comparison algorithm on the two test sets, DTLZ and MW. Table 4 shows the results of the HV values. The computed results are

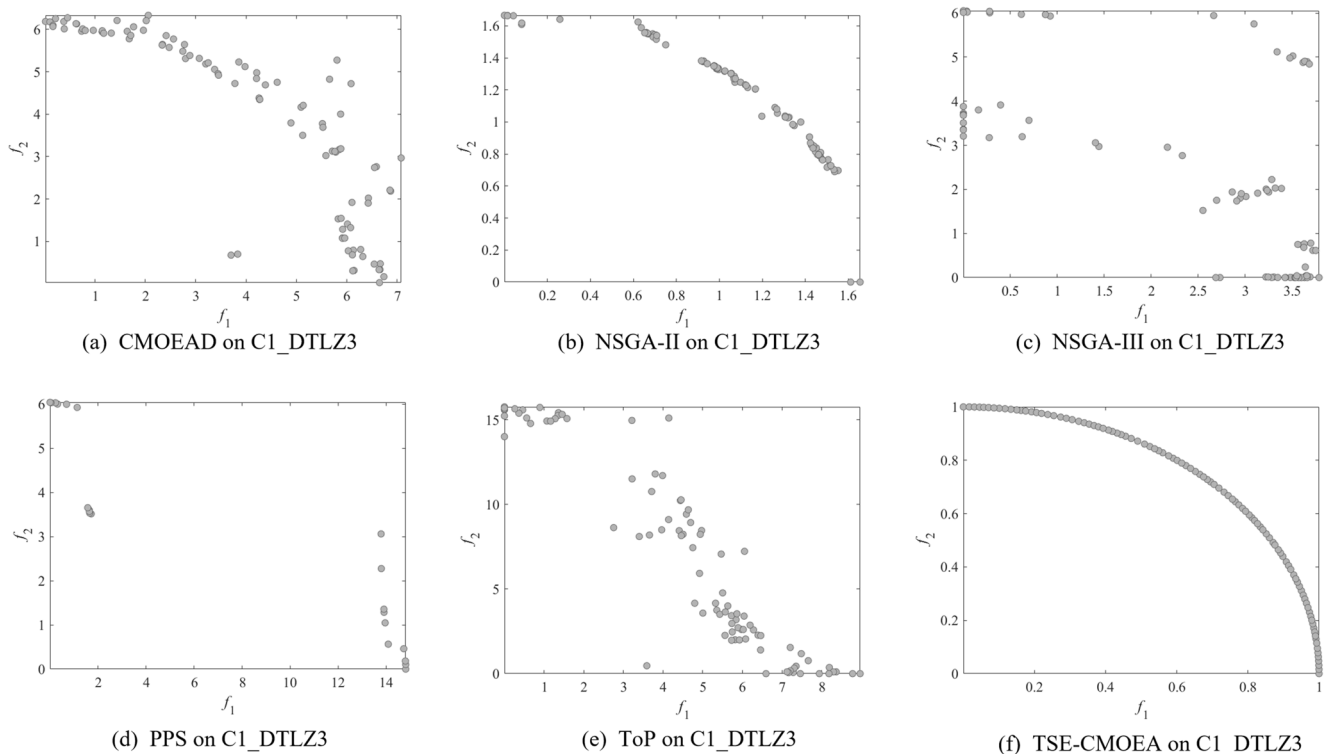


Fig. 4 The final populations of NSGA-II, NSGA-III, C-MOEA/D, PPS, ToP and TSC-CMOEA in C1_DTLZ3

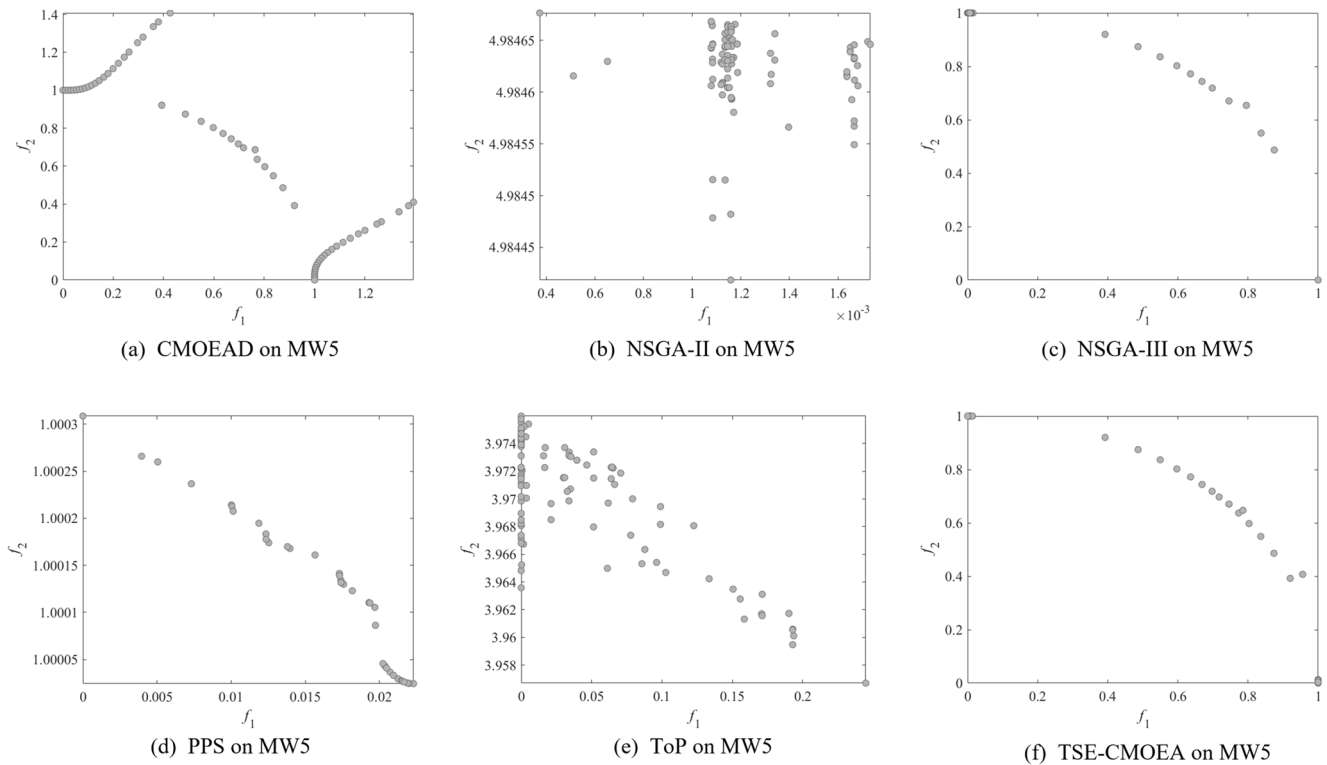


Fig. 5 The final populations of NSGA-II, NSGA-III, C-MOEA/D, PPS, ToP and TSC-CMOEA in MW5

shown in Tables 3 and 4, and it can be seen that our proposed algorithm achieves the lead on most of the problems.

Figures 4 and 5 plot NSGA-II, NSGA-III, C-MOEA/D, PPS, ToP, CTAEA, CCMO, and TSC-CMOEA of MW5 and C1_DTLZ3, and the feasible non-dominated solutions of the median IGD value in 30 runs. For MW5 and MW9 problems with discontinuous feasible regions, TSC-CMOEA shows better convergence and distribution than other MOEAs. This is because the help problem assists the raw problem. For Population1 which deals with the raw CMOP and Population2 which deal with help problems, the cooperative relationship between them effectively improves the convergence and diversity of Population1. It can be seen from Fig. 2 that the convergence of TSC-CMOEA is better than that of other MOEAs, and other MOEAs still have a large number of individuals in the feasible region far from the real PF. Rapid convergence rate of TSC-CMOEA is mainly attributed to the offspring produced by the help problem, which makes it easier for the raw CMOP to skip the infeasible region. In short, in the constrained DTLZ problem with multi-modal landscape and a wider infeasible region and the MW problem with a small and discontinuous feasible region, the offspring produced by Population 2 help Population 1 to skip the infeasible region and obtain better convergence and diversity.

4.6 Experimental results of LIR-CMOP and DOC problems

To further verify the performance of TSC-CMOEA, we used more difficult test suites (ie LIRCMOP and DOC) to perform supplementary tests. LIRCMOP contains 14 CMOPs with small feasible regions and complex relationships between position and distance variables. DOC contains 9 CMOPs with complex constraints in both decision space and target space. Table 5 shows the average and standard deviation of the IGD values obtained by NSGA-II, NSGA-III, C-MOEA/D, PPS, ToP, and TSC-CMOEA during 30 independent runs on the LIRCMOP test suite and the DOC test suite. TSC-CMOEA made on 15 test questions optimal, PPS is 5, ToP is 1. And we also computed HV values on these test sets and the results are shown in Table 6. The data in Table 6 shows that it performs similarly to the IGD values, and our proposed algorithm outperforms the comparison algorithm on most of the test problems, with only a few falling slightly back to the PPS. Therefore, it can be concluded that the proposed TSC-CMOEA has better overall performance than some of the most advanced MOEAs when solving the benchmark CMOP. This is precisely caused by the new coevolution method and the two-stage algorithm design.

Table 5 Mean and standard deviation of the IGD values obtained by NSGA-II, NSGA-III, C-MOEA/D, PPS, TOP, and TSC-CMOEA for 30 independent runs on LIRC-MOP test suite and DOC test suite

Problem	NSGAI	NSGAIII	C-MOEA/D	PPS	TOP	TSC-CMOEA
LIRC-MOP1	3.5130e-1 (1.82e-2) –	2.8160e-1 (2.61e-2) –	2.5696e-1 (3.53e-2) –	1.3350e-2 (8.35e-3) +	3.0420e-1 (1.81e-2) –	9.6198e-2 (6.14e-2)
LIRC-MOP2	2.9657e-1 (1.89e-2) –	2.4829e-1 (2.85e-2) –	2.1917e-1 (3.18e-2) –	7.3315e-3 (1.51e-3) +	2.7022e-1 (1.89e-2) –	1.0689e-1 (6.11e-2)
LIRC-MOP3	3.4301e-1 (1.88e-2) –	3.0861e-1 (3.20e-2) –	2.7898e-1 (4.16e-2) –	8.9704e-3 (5.00e-3) +	3.3141e-1 (1.90e-2) –	9.8033e-2 (5.55e-2)
LIRC-MOP4	3.1978e-1 (1.07e-2) –	2.9000e-1 (3.19e-2) –	2.6148e-1 (3.59e-2) –	3.9794e-3 (4.71e-4) +	3.1830e-1 (1.63e-2) –	1.3318e-1 (6.38e-2)
LIRC-MOP5	1.2148e+0 (7.04e-3) –	1.2231e+0 (5.88e-3) –	1.2145e+0 (7.62e-3) –	6.8458e-3 (7.50e-4) +	1.1186e+0 (1.97e-1) –	7.4428e-3 (5.41e-4)
LIRC-MOP6	1.3206e+0 (1.38e-1) –	1.3461e+0 (3.90e-4) –	1.3452e+0 (1.67e-4) –	7.8235e-3 (7.22e-4) –	1.1764e+0 (3.48e-1) –	6.9969e-3 (1.10e-3)
LIRC-MOP7	5.4435e-1 (6.98e-1) –	4.9799e-1 (6.66e-1) –	9.4158e-1 (7.62e-1) –	1.0254e-2 (8.14e-4) –	8.9817e-1 (8.00e-1) –	7.3071e-3 (2.66e-4)
LIRC-MOP8	7.0612e-1 (7.02e-1) –	9.5778e-1 (7.37e-1) –	1.5389e+0 (4.34e-1) –	1.0566e-2 (1.66e-3) –	1.1491e+0 (6.77e-1) –	7.2908e-3 (2.29e-4)
LIRC-MOP9	9.2122e-1 (1.01e-1) –	8.5359e-1 (1.53e-1) –	8.0741e-1 (1.79e-1) –	2.1287e-1 (1.70e-1) –	4.5662e-1 (1.13e-1) –	1.7788e-2 (2.10e-2)
LIRC-MOP10	8.6361e-1 (1.15e-1) –	7.8043e-1 (1.39e-1) –	5.1971e-1 (2.39e-1) –	1.2772e-2 (3.81e-2) –	3.9775e-1 (8.51e-2) –	6.4508e-3 (3.41e-4)
LIRC-MOP11	9.1643e-1 (1.25e-1) –	7.1929e-1 (9.68e-2) –	8.1343e-1 (1.28e-1) –	3.9146e-2 (8.13e-2) ≈	3.8119e-1 (1.00e-1) –	2.7702e-3 (1.05e-4)
LIRC-MOP12	6.9308e-1 (9.06e-2) –	6.2571e-1 (1.73e-1) –	5.4672e-1 (1.70e-1) –	8.1598e-2 (5.80e-2) –	2.8850e-1 (9.22e-2) –	3.8654e-3 (4.88e-4)
LIRC-MOP13	1.3242e+0 (2.55e-3) –	1.3030e+0 (2.95e-4) –	1.3020e+0 (1.07e-4) –	1.3333e-1 (4.31e-3) –	1.3217e+0 (8.44e-2) –	1.1794e-1 (1.83e-3)
LIRC-MOP14	1.2822e+0 (2.90e-3) –	1.2591e+0 (4.30e-4) –	1.2581e+0 (7.98e-5) –	1.1760e-1 (3.35e-3) –	1.2535e+0 (1.47e-1) –	1.0272e-1 (1.56e-3)
DOC1	2.0130e+0 (1.49e+0) –	2.4628e+0 (1.81e+0) –	2.1195e+0 (1.30e+0) –	7.5072e-2 (4.37e-2) –	5.9478e-3 (3.00e-4) –	5.4714e-3 (3.43e-4)
DOC2	NaN (NaN)	NaN (NaN)	NaN (NaN)	3.3606e-1 (1.77e-1) –	4.5227e-1 (1.27e-1) –	4.9571e-3 (2.90e-3)
DOC3	6.3154e+2 (1.11e+2) ≈	6.8392e+2 (1.48e+2) ≈	6.6845e+2 (2.45e+2) ≈	2.0849e+2 (1.73e+2) +	1.3630e+2 (1.93e+2) +	7.7070e+2 (4.75e+2)
DOC4	9.5114e-1 (6.19e-1) –	7.5673e-1 (6.85e-1) –	6.2930e-1 (3.05e-1) –	2.8308e-1 (5.15e-2) –	8.1964e-2 (6.72e-2) ≈	2.3486e-2 (7.91e-3)
DOC5	NaN (NaN)	NaN (NaN)	NaN (NaN)	3.5905e+1 (6.07e+1) –	2.9290e+1 (5.16e+1) ≈	3.0759e+1 (5.71e+1)
DOC6	3.3199e+0 (4.02e+0) –	2.2435e+0 (3.05e+0) –	1.1648e+0 (1.58e+0) –	5.1919e-1 (7.90e-2) –	2.5004e+0 (1.06e+0) –	3.2305e-3 (6.67e-4)
DOC7	8.1158e+0 (2.03e+0) –	6.0611e+0 (1.92e+0) –	5.5129e+0 (2.20e+0) –	4.1983e-1 (1.82e-1) –	3.7638e-1 (3.49e-1) –	2.4263e-3 (9.72e-5)
DOC8	1.5725e+2 (1.47e+2) –	6.1502e+1 (6.50e+1) –	7.9556e+1 (7.60e+1) –	9.4089e+1 (3.29e+1) –	1.6257e+1 (9.03e+0) –	6.4513e-2 (3.18e-3)
DOC9	2.4665e-1 (1.17e-1) –	1.5333e-1 (9.52e-2) –	4.1048e-1 (6.29e-2) –	2.7210e-1 (2.69e-2) –	1.7538e-1 (3.70e-2) –	7.4340e-2 (8.71e-3)
+ / - / ≈	0/20/1	0/20/1	0/20/1	6/16/1	1/20/2	

Table 6 Mean and standard deviation of the HV values obtained by NSGA-II, NSGA-III, C-MOEA/D, PPS, TOP, and TSC-CMOEA for 30 independent runs on LIRC-MOP test suite and DOC test suite

Problem	NSGAI	NSGAIII	CMOEA/D	PPS	ToP	SCCMO
LIRC-MOP1	9.8487e-2 (8.24e-3) –	1.1848e-1 (9.92e-3) –	1.2451e-1 (1.18e-2) –	2.3482e-1 (2.76e-3) +	1.1228e-1 (8.02e-3) –	1.8583e-1 (2.89e-2)
LIRC-MOP2	2.1450e-1 (1.47e-2) –	2.2942e-1 (1.40e-2) –	2.4650e-1 (1.71e-2) –	3.5911e-1 (6.62e-4) +	2.2091e-1 (1.15e-2) –	3.0130e-1 (3.26e-2)
LIRC-MOP3	9.2331e-2 (5.20e-3) –	1.0094e-1 (1.03e-2) –	1.0885e-1 (1.10e-2) –	2.0424e-1 (1.50e-3) +	9.4031e-2 (7.11e-3) –	1.6731e-1 (2.18e-2)
LIRC-MOP4	1.8119e-1 (1.05e-2) –	1.9109e-1 (1.48e-2) –	2.0237e-1 (1.60e-2) –	3.1506e-1 (5.49e-4) +	1.8090e-1 (1.32e-2) –	2.5817e-1 (2.78e-2)
LIRC-MOP5	0.0000e+0 (0.00e+0) –	0.0000e+0 (0.00e+0) –	0.0000e+0 (0.00e+0) –	2.9162e-1 (2.82e-4) +	1.1331e-2 (4.52e-2) –	2.9074e-1 (3.25e-4)
LIRC-MOP6	2.6621e-3 (1.46e-2) –	0.0000e+0 (0.00e+0) –	0.0000e+0 (0.00e+0) –	1.9697e-1 (1.98e-4) +	1.1507e-2 (2.34e-2) –	1.9616e-1 (5.80e-4)
LIRC-MOP7	1.7956e-1 (1.10e-1) –	1.8631e-1 (1.05e-1) –	1.1414e-1 (1.20e-1) –	2.9350e-1 (5.55e-4) –	1.2785e-1 (1.32e-1) –	2.9444e-1 (1.41e-4)
LIRC-MOP8	1.5035e-1 (1.08e-1) –	1.1084e-1 (1.13e-1) –	2.1824e-2 (6.66e-2) –	2.9373e-1 (3.42e-4) –	8.9355e-2 (1.13e-1) –	2.9440e-1 (1.11e-4)
LIRC-MOP9	1.6677e-1 (5.50e-2) –	1.6878e-1 (7.72e-2) –	2.0774e-1 (7.76e-2) –	5.0403e-1 (5.05e-2) –	3.8432e-1 (7.50e-2) –	5.6023e-1 (5.40e-3)
LIRC-MOP10	1.3809e-1 (7.42e-2) –	1.5800e-1 (1.17e-1) –	3.6271e-1 (1.71e-1) –	7.0415e-1 (1.76e-2) –	4.9414e-1 (6.88e-2) –	7.0578e-1 (2.68e-4)
LIRC-MOP11	2.0224e-1 (6.20e-2) –	2.4822e-1 (6.89e-2) –	2.3748e-1 (8.36e-2) –	6.7060e-1 (5.16e-2) –	4.3931e-1 (7.91e-2) –	6.9384e-1 (7.03e-5)
LIRC-MOP12	3.4099e-1 (7.75e-2) –	3.2068e-1 (9.73e-2) –	4.0897e-1 (6.82e-2) –	5.8599e-1 (2.56e-2) –	4.7550e-1 (4.75e-2) –	6.1996e-1 (2.11e-4)
LIRC-MOP13	7.2390e-5 (1.07e-4) –	4.4778e-4 (7.37e-6) –	4.4711e-4 (7.88e-6) –	5.1402e-1 (6.39e-3) +	4.2316e-3 (1.60e-2) –	5.1134e-1 (2.54e-3)
LIRC-MOP14	5.1608e-4 (4.56e-4) –	9.9602e-4 (1.15e-5) –	9.9966e-4 (8.77e-6) –	5.3558e-1 (3.41e-3) –	1.1468e-2 (3.49e-2) –	5.4004e-1 (2.23e-3)
DOC1	4.8557e-2 (9.97e-2) –	2.7629e-2 (5.82e-2) –	1.1494e-2 (3.10e-2) –	2.8252e-1 (2.70e-2) –	3.4444e-1 (4.44e-4) –	3.4483e-1 (6.00e-4)
DOC2	NaN (NaN)	NaN (NaN)	NaN (NaN)	2.6745e-1 (1.13e-1) –	2.3436e-1 (1.04e-1) –	6.1987e-1 (3.92e-3)
DOC3	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	7.7168e-4 (4.23e-3) ≈	1.0703e-2 (5.86e-2)
DOC4	7.1419e-2 (1.21e-1) –	1.1749e-1 (1.46e-1) –	7.0097e-2 (7.93e-2) –	2.5253e-1 (4.91e-2) –	4.7331e-1 (6.78e-2) ≈	5.3613e-1 (9.03e-3)
DOC5	NaN (NaN)	NaN (NaN)	NaN (NaN)	6.3887e-2 (1.16e-1) –	2.5979e-1 (1.79e-1) –	4.1540e-1 (1.59e-1)
DOC6	1.3851e-2 (6.03e-2) –	3.2531e-2 (9.43e-2) –	7.1467e-2 (1.06e-1) –	1.6065e-1 (5.62e-2) –	0.0000e+0 (0.00e+0) –	5.2033e-1 (1.01e-2)
DOC7	0.0000e+0 (0.00e+0) –	0.0000e+0 (0.00e+0) –	0.0000e+0 (0.00e+0) –	1.1564e-1 (1.04e-1) –	1.9275e-1 (1.71e-1) –	5.3753e-1 (7.53e-3)
DOC8	0.0000e+0 (0.00e+0) –	0.0000e+0 (0.00e+0) –	0.0000e+0 (0.00e+0) –	0.0000e+0 (0.00e+0) –	0.0000e+0 (0.00e+0) –	7.9735e-1 (4.41e-3)
DOC9	0.0000e+0 (0.00e+0)	NaN (NaN)	0.0000e+0 (0.00e+0)	NaN (NaN)	0.0000e+0 (0.00e+0)	NaN (NaN)
+ / - / ≈	0/19/1	0/19/1	0/19/1	7/14/1	0/20/2	

Table 7 HV Value of NSGAI I , NSGAI II I , CMOEAD , PPS , ToP and SCCMO on microgrid dispatch model, where the best result on each test instance is highlight

Problem	NSGAI I	NSGAI II I	CMOEAD	PPS	ToP	SCCMO
DM	0.0000e+0 (0.00e+0)	-1.1019e-3 (2.86e-3)	-6.0606e-3 (3.72e-3)	-1.6250e-2 (9.17e-3)	-0.0000e+0 (0.00e+0)	-2.1540e-2 (7.48e-3)
+ / - / ≈	0/1/0	0/1/0	0/1/0	0/1/0	0/1/0	

4.7 Experimental results of microgrid scheduling problem

Finally, we use the proposed TSC-CMOEA to optimize a microgrid dispatch model. The dispatch model of the microgrid is as follows.

$$\begin{aligned} \text{minimize } \mathbf{f}(\mathbf{x}) = & \begin{cases} f_1 = C_{ex} + C_{DG} + C_{em} + C_{lc} \\ f_2 = \max P_{ex,t} - \min P_{ex,t} = \Delta P_{ex} \end{cases} \\ \text{subject to } & g_j(\mathbf{x}) \leq 0, j = 1, 2, 3. \end{aligned} \quad (7)$$

Where C_{ex} is the transaction cost of the external grid, C_{DG} is the generation cost of DG, C_{em} is the operation and maintenance cost of EER and storage devices, C_{lc} is the compensation cost of flexible load and $P_{ex,t}$ is the power exchanges value between hybrid microgrid and external grid. The objective function is to minimize the economic cost of the microgrid and minimize the peak-to-valley difference between power supply and demand. The decision variables include microgrid dispatchable units such as energy storage ESS charging and discharging power and levelizable load transfer. The model also has complex constraints such as the equal power constraint of shiftable load shift in and out, the ESS equipped capacity constraint and the battery hourly power variation constraint. We take the example of a microgrid system in a building of a university in Hunan Province. Table 7 lists the HV values obtained for five comparative MOEAs, where each MOEA is run for 1,000 generations with a population size of 100 . As can be seen from the Table 7, TSC-CMOEA shows better overall performance than the comparative MOEA in the optimization of the microgrid dispatch model, obtaining the best HV values. Therefore, it can be confirmed that the proposed TSC-CMOEA is also effective in solving CMOPs in applications.

5 Conclusion

In this paper, we propose a new coevolution method and staged coevolutionary algorithm for constrained multi-objective algorithms which are difficult to converge or have poor distribution when dealing with complex constraints. It is designed to solve a difficult CMOP problem by adding a help problem and an additional processing stage. In the algorithm, the first stage uses

the proposed new coevolution method to process CMOP, so that the population can quickly and efficiently cross the infeasible region. When the population change rate is too small during the evolution process, it will switch to the second stage; in the second stage, only the raw problem processing part is retained. Through further optimization of the raw problem, the expected solution set is finally obtained. The results of comparison with CTAEA and CCMO show that the coevolution and two-stage strategy in this paper are effective; comparative experiments with algorithms such as NSGA-II, NSGA-III, C-MOEA/D, PPS and ToP show that TSC-CMOEA can effectively avoid local optimization caused by constraints when solving multi-objective problem. The solution results of complex multi-objective optimization problems such as LIR-CMOP and DOC show that TSC-CMOEA has better convergence performance and distribution performance in solving complex CMOP problems. In addition, if the efficiency of the offspring generation strategy in this paper can be improved, it may improve the performance of the algorithm on large-scale problems.

Acknowledgments The authors are very grateful to the anonymous reviewers for their valuable comments on improving this article. Additionally, this work is supported by Hunan Provincial Natural Science Foundation of China (No. 2020JJ4587), Guangdong Basic and Applied Basic Research Foundation (No. 2019A1515110423), Open Fund Project of Key Laboratory of Advanced Perception and Intelligent Control of High-end Equipment of Ministry of Education (No. GDSC202020), and Open Fund Project of Fujian Provincial Key Laboratory of Data Intensive Computing (No. BD202004). Supported by the Open Research Fund of AnHui Key Laboratory of Detection Technology and Energy Saving Devices (No. JCKJ2021B05).

References

1. Tan B, Ma H, Mei Y, Zhang M (2018) Evolutionary multi-objective optimization for web service location allocation problem. *IEEE Trans Serv Comput*
2. Farzin H, Fotuhi-Firuzabad M, Moeini-Aghtaie M (2016) A stochastic multi-objective framework for optimal scheduling of energy storage systems in microgrids. *IEEE Trans Smart Grid* 8(1):117–127
3. Gong M, Wang Z, Zhu Z, Jiao L (2017) A similarity-based multiobjective evolutionary algorithm for deployment optimization of near space communication system. *IEEE Trans Evol Comput* 21(6):878–897
4. Deb K, Jain H (2014) An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting

- approach, part i: Solving problems with box constraints. *IEEE Trans Evol Comput* 18(4):577–601
5. Zhu Q, Zhang Q, Lin Q (2020) A constrained multiobjective evolutionary algorithm with detect-and-escape strategy. *IEEE Trans Evol Comput* 24(5):938–947
 6. Garg H (2016) A hybrid pso-ga algorithm for constrained optimization problems. *Appl Math Comput* 274:292–305
 7. Qiao J, Zhou H, Yang C, Yang S (2019) A decomposition-based multiobjective evolutionary algorithm with angle-based adaptive penalty. *Appl Soft Comput* 74:190–205
 8. Woldesenbet YG, Yen GG, Tessema BG (2009) Constraint handling in multiobjective evolutionary optimization. *IEEE Trans Evol Comput* 13(3):514–525
 9. Fan Z, Li W, Cai X, Li H, Wei C, Zhang Q, Deb K, Goodman E (2019) Push and pull search for solving constrained multi-objective optimization problems. *Swarm Evol Comput* 44:665–679
 10. Runarsson TP, Yao X (2000) Stochastic ranking for constrained evolutionary optimization. *IEEE Trans Evol Comput* 4(3):284–294
 11. Deb K, Pratap A, Agarwal S, Meyarivan TAMT (2002) A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Trans Evol Comput* 6(2):182–197
 12. Yi W, Gao L, Pei Z, Lu J, Chen Y (2021) ε constrained differential evolution using halfspace partition for optimization problems. *J Intell Manuf* 32(1):157–178
 13. Fan Z, Li W, Cai X, Hu K, Lin H, Li H (2016) Angle-based constrained dominance principle in moea/d for constrained multi-objective optimization problems. In: 2016 IEEE Congress on evolutionary computation (CEC), IEEE, pp 460–467
 14. Xia M, Dong M (2021) A novel two-archive evolutionary algorithm for constrained multi-objective optimization with small feasible regions. *Knowl-Based Syst.*, pp 107693
 15. Liu ZZ, Wang BC, Tang K (2021) Handling constrained multiobjective optimization problems via bidirectional coevolution. *IEEE Trans Cybern PP(99):1–14*
 16. Wang B-C, Li H-X, Zhang Q, Wang Y (2018) Decomposition-based multiobjective optimization for constrained evolutionary optimization. *IEEE Transactions on systems, man, and cybernetics: systems*
 17. Tian Y, Zhang T, Xiao J, Zhang X, Jin Y (2020) A coevolutionary framework for constrained multi-objective optimization problems. *IEEE Transactions on systems, man, and cybernetics: systems*
 18. Yu K, Liang J, Qu B, Yue C (2021) Purpose-directed two-phase multiobjective differential evolution for constrained multiobjective optimization. *Swarm Evol Comput* 60:100799
 19. Panda A, Pani S (2016) A symbiotic organisms search algorithm with adaptive penalty function to solve multi-objective constrained optimization problems. *Appl Soft Comput* 46(C):344–360
 20. Peng C, Liu HL, Gu F (2017) An evolutionary algorithm with directed weights for constrained multi-objective optimization. *Appl Soft Comput* 60:613–622. <https://doi.org/10.1016/j.asoc.2017.06.053>, <https://www.sciencedirect.com/science/article/pii/S1568494617303964>
 21. Liu R, Li J, Mu C, Jiao L et al (2017) A coevolutionary technique based on multi-swarm particle swarm optimization for dynamic multi-objective optimization. *Eur J Oper Res* 261(3):1028–1051
 22. Peng X, Liu K, Jin Y (2016) A dynamic optimization approach to the design of cooperative co-evolutionary algorithms. *Knowl-Based Syst* 109:174–186
 23. Zhang X-Y, Gong Y-J, Lin Y, Zhang J, Kwong S, Zhang J (2019) Dynamic cooperative coevolution for large scale optimization. *IEEE Trans Evol Comput* 23(6):935–948
 24. Mohamed AW (2018) A novel differential evolution algorithm for solving constrained engineering optimization problems. *J Intell Manuf* 29(3):659–692
 25. Zhou J, Zou J, Yang S, Zheng J, Pei T (2021) Niche-based and angle-based selection strategies for many-objective evolutionary optimization. *Information Sciences*
 26. Yang N, Liu HL (2021) Adaptively allocating constraint-handling techniques for constrained multi-objective optimization problems. *Int J Pattern Recognit Artif Intell*
 27. Wang J, Liang G, Zhang J (2018) Cooperative differential evolution framework for constrained multiobjective optimization. *IEEE Trans Cybern* 49(6):2060–2072
 28. Li K, Chen R, Fu G, Yao X (2018) Two-archive evolutionary algorithm for constrained multiobjective optimization. *IEEE Trans Evol Comput* 23(2):303–315
 29. Wang J, Li Y, Zhang Q, Zhang Z, Gao S (2021) Cooperative multiobjective evolutionary algorithm with propulsive population for constrained multiobjective optimization. *IEEE Trans Syst Man Cybern: Syst PP(99):1–16*
 30. Liu ZZ, Wang Y (2019) Handling constrained multiobjective optimization problems with constraints in both the decision and objective spaces. *IEEE Trans Evol Comput* 23(5):870–884
 31. Tian Y, Cheng R, Zhang X, Jin Y (2017) Platemo: A matlab platform for evolutionary multi-objective optimization [educational forum]. *IEEE Comput Intell Mag* 12(4):73–87
 32. Ma Z, Wang Y (2019) Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons. *IEEE Trans Evol Comput* 23(6):972–986
 33. Fan Z, Li W, Cai X, Huang H, Fang Y, You Y, Mo J, Wei C, Goodman E (2019) An improved epsilon constraint-handling method in moea/d for cmops with large infeasible regions. *Soft Comput* 23(23):12491–12510
 34. a JD, b SG, C DM, A FH (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol Comput* 1(1):3–18
 35. Tian Y, Xiang X, Zhang X, Cheng R, Jin Y (2018) Sampling reference points on the pareto fronts of benchmark multi-objective optimization problems. In: 2018 IEEE congress on evolutionary computation (CEC), pp 1–6 (2018). IEEE

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



Chaodong Fan received his master's degree and doctor's degree from Hunan University in 2011 and 2014 respectively. Mainly engaged in intelligent computing, pattern recognition and intelligent systems, smart grid related fields. Some new intelligent optimization algorithms and image processing technologies, such as molecular dynamics optimization algorithm, spatial section projection histogram and post-processing strategy based on threshold, are proposed. His

research interests include artificial intelligence, image processing and model recognition.



Fanyong Cheng received the Ph.D. degree in control science and engineering from the Hunan University, Changsha, China, in 2017. Currently, he is an associate professor with Anhui polytechnic University. He is the reviewer of many international top journals, such as IEEE Transactions on Industrial Electronics, IEEE Transactions on Industrial Informatics and so on. He is generally interested in machine learning, fault detection and diagnosis, image processing, together with their applications in various sectors.



Jiawei Wang was born in 1997 in Changsha, Hunan Province, China. He is studying in Xiangtan University for his master's degree. His main research interests are evolutionary algorithms.



Zhaoyang Ai is now with the Institute of Cognitive Control and Biophysics Linguistics, College of Foreign Languages, Hunan University. From 2009 to 2010 he was on a leave as a visiting scholar at Yale University with intervals of visit to Harvard University and MIT. He also visited Dresden University of Technology in 2018. Current research interests: Evolution Studies, Biolinguistic Engineering, Cognitive Control and Biophysics Linguistics,

Conductive Education, Translation Studies, and Lexicology etc.



Leyi Xiao was born in 1986 in Changde, Hunan, China. She graduated from Hunan University with a doctorate in engineering in 2020. Her research interest is in image processing and model recognition, which relate to both the field of computer vision and pattern recognition.



Zhenhuan Zeng was born in 1995 in Changsha, Hunan Province, China. He is studying in Xiangtan University for his master's degree. His main research interests are evolutionary algorithms, and computer vision.

Affiliations

Chaodong Fan^{1,2,3,4} · Jiawei Wang^{1,2} · Leyi Xiao^{3,4,5} · Fanyong Cheng³ · Zhaoyang Ai⁶ · Zhenhuan Zeng¹

- ¹ School of Computer Science, Xiangtan University, Xiangtan, 411105, Hunan, China
- ² Foshan Green Intelligent Manufacturing Research Institute of Xiangtan University, Foshan, 528000, Guangdong, China
- ³ Key Laboratory of Advanced Perception and Intelligent Control of High-end Equipment of Ministry of Education, Anhui Polytechnic University, Wuhu, 241000, Anhui, China
- ⁴ School of Information Technology and Management, Hunan University of Finance and Economics, Changsha, 410205, Hunan, China
- ⁵ AnHui Key Laboratory of Detection Technology and Energy Saving Devices, AnHui Polytechnic University, Wuhu, 241000, Anhui, China
- ⁶ College of Foreign Languages; Inter-disciplinary Research Center of Language Intelligence and Cultural Heritages, Hunan University, Changsha, 410082, Hunan, China