A Novel Two-phase Evolutionary Algorithm for Solving Constrained Multi-objective Optimization Problems⋆

Yanping Wang[a,b], Yuan Liu[a,b,*], Juan Zou[a,b], Jinhua Zheng[a,b,c] and Shengxiang Yang[d]

[a] *Key Laboratory of Hunan Province for Internet of Things and Information Security, School of Computer Science, Xiangtan University, Xiangtan 411105, Hunan, China*

[b] *Key Laboratory of Intelligent Computing and Information Processing, Ministry of Education, School of Computer Science, Xiangtan University, Xiangtan 411105, Hunan, China*

[c] *Hunan Provincial Key Laboratory of Intelligent Information Processing and Application, Hengyang 421002, China*

[d] *School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, UK*

ARTICLE INFO

Keywords:
Coevolution
Constraints
Evolutionary algorithms
Optimization algorithms

ABSTRACT

It is challenging to balance convergence and diversity in constrained multi-objective optimization problems (CMOPs) since the complex constraints will disperse the feasible regions into many diverse, small parts of the entire search region. Although there has been some research on CMOPs, existing evolutionary algorithms still cannot cause the evolutionary population to converge a diversified feasible Pareto-optimal front. In order to solve this problem, we propose a novel two-phase evolutionary algorithm for solving CMOPs, named DTAEA. DTAEA divides the population's coevolutionary process into two phases. In the first phase, the dual population weak coevolution is combined with the complementary environmental selection strategy to improve the algorithm's exploration under constraints, which makes the evolutionary population quickly traverse the infeasible regions and search for all of the feasible regions. When the proportion of feasible solutions in the population reaches a certain threshold or the convergence of feasible solutions reaches a certain level, the population's evolutionary process enters the second phase, that is, the progressive phase. In the second phase, a feasibility-oriented method guides a single population to distribute itself widely in the feasible regions explored in the first phase. Comparative experiments show that the DTAEA is more competitive than other algorithms on CMOP benchmarks.

1. Introduction

There are many optimization problems in practical engineering applications that need to optimize multiple objectives simultaneously under various constraints. Such problems include vehicle path problems [1], energy storage system optimization scheduling [2], robust analog circuit design [3], and multi-objective test resource allocation problem for reliability training [4]. These kinds of problems are called constrained multi-objective optimization problems (CMOPs) in the optimization community, and the formal definition is as follows:

$$\begin{cases} \text{minimize} & \mathbf{F}(\mathbf{x}) = \left(f_1(\mathbf{x}), \dots, f_m(\mathbf{x})\right)^T \\ \text{subject to} & g_j(\mathbf{x}) \geq 0, j = 1, \dots, p \\ & h_j(\mathbf{x}) = 0, j = p+1, \dots, q \\ & \mathbf{x} \in S \end{cases}, \tag{1}$$

where m is the number of objective functions; $\mathbf{x} = (x_1, \dots, x_n) \in S$ is a solution consisting of n decision variables; T is an n-dimensional decision vector, and $S \subseteq R_n$ represents the search space. CMOPs usually contain two types of constraints, where $g_j(\mathbf{x})$ represents one of the inequality constraints; $h_j(\mathbf{x})$ represents one of the equality constraints; for equational constraints in problems; p is the number of inequality constraints, and $q - p$ is the number of equation constraints.

Multi-objective functions of CMOPs generally conflict with one another, which means the optimal set of solutions for CMOPs is usually a set of compromise solutions. Although these compromise solutions cannot be optimal on all objectives simultaneously, there is no better way to achieve solutions on all objectives simultaneously. Solution $\mathbf{x}$ is

liu3yuan@xtu.edu.cn (Y. Liu)
ORCID(s):

called a feasible solution if it satisfies the constraints and an infeasible solution if it does not. For an infeasible solution, the degree to which the solution violates the constraint is generally described by the constraint violation (CV). For a solution $\mathbf{x}$, the smaller the CV value, the better the solution.The CV value of a feasible solution is zero, while the CV value of an infeasible solution is greater than zero. The CV of $\mathbf{x}$ at the jth constraint can be expressed as follows:

$$cv_j(\mathbf{x}) = \begin{cases} \max\left(0, g_j(\mathbf{x})\right), & j = 1, \dots, p \\ \max\left(0, \left|h_j(\mathbf{x})\right| - \epsilon\right), & j = p+1, \dots, q \end{cases}, \tag{2}$$

where ϵ is a sufficiently small positive number (e.g., 10^{-6}).The overall constraint violation of $\mathbf{x}$ can be calculated as:

$$CV(\mathbf{x}) = \sum_{j=1}^{q} cv_j(\mathbf{x}). \tag{3}$$

In the past, many evolutionary algorithms (EAs) have been proposed to solve multi-objective optimization problems (MOPs). Multi-objective optimization evolutionary algorithms (MOEAs) can be divided into three types. The first type includes Pareto-based algorithms where the solution selection consists of two parts: the Pareto dominance criteria to ensure the convergence of the population and the diversity maintenance strategy to improve the distribution of the population. Typical representatives are NSGA-II [5], PMEA-MA[6], SPEA2 [7], and GrEA [8]. The second type are the decomposition-based algorithms. These algorithms use a set of predefined uniformly distributed weights to decompose a MOP into multiple scalar optimization sub-problems and then use the scalar optimization strategy to optimize these sub-problems simultaneously. The performance of the decomposition-based algorithms depends on the setting of the weights. Examples of these algorithms include: MOEA/D [9], MOEA/DD [10], and DMEA-WUA [11]. Indicator-based MOEAs belong to the third type. In order to compare the performance of MOEAs, performance indicators are used to quantitatively evaluate the quality of the populations obtained by the algorithms. Indicator-based algorithms directly use these performance indicators to evaluate the quality of solutions and sequentially select the best for the next generation's population. Examples of these algorithms include I-DBEA [12], AR-MOEA [13], and HypE [14].

Coevolutionary mechanisms have been widely used in MOEAs [15, 16, 17]. Based on the general evolutionary algorithm, the coevolutionary algorithm considers the interaction between populations in the evolutionary process to ensure an appropriate balance between the diversity and convergence of the population. Coevolutionary algorithms can significantly improve the performance of EAs, and their divide-and-conquer strategy [15] [16] can effectively solve complex MOPs. This shows that the evolutionary advantages exhibited by coevolution are consistent with the research on MOEAs. An example is the DECAL [17] algorithm which is a decomposition-based coevolutionary algorithm that attempts to enhance the search capability of decomposition-based methods by introducing coevolutionary mechanisms.

MOEAs have been developed for many years and have proven to be an excellent solution for MOPs. Unlike general MOPs, CMOPs need to consider both objective functions and constraints. Therefore, it is necessary to add constraint processing techniques (CHTs) into the existing MOEA to solve such problems. CHTs help MOEAs to distinguish between feasible and infeasible regions, and they need the EA to assist in the search for optimal solutions. Balancing objective functions and constraints is a key issue in solving CMOPs. For complex CMOPs, it is challenging to get ideal solutions from traditional mathematical programming techniques. Liu et al. [18] proposed an indicator-based CMOEA framework that can combine MOEAs with different CHTs. As a result, many new CHTs have been proposed. There are four main types of methods for dealing with constraint optimization: 1) methods based on penalty function, 2) methods based on constrained objective separation, 3) methods based on multi-objective optimization, and 4) hybrid methods.

Methods based on penalty function convert the model of the problem into an unconstrained optimization model by means of a penalty factor, and use the idea of solving unconstrained optimization problems to deal with constrained optimization problems. Coit et al. [19] proposed a global optimization algorithm based on the adaptive difference evolutionary algorithm. The advantage of the penalty function method is simple and easy to operate. But the size of the penalty coefficient is difficult to determine.

In methods based on constraint-objective separation, constraints and objective functions are treated separately, use the objective functions or the constraint violation degree value as the fitness value. The feasibility rule is one of the most popular constraint-objective separation methods. The most representative CHT is the constraint dominance principle (CDP) proposed by Deb [5], which first compares the CV degree of the solutions until it meets certain conditions, then compare the objective function value, this method often ignores some infeasible solutions that carry useful information. To overcome the drawback of CHT, Ming et al. [20] proposed a new CHT based on the CDP, in which the fitness function of each solution is defined as a weighted sum of two rankings: one based on the CDP and the other on the Pareto-dominated solution.

Methods based on multi-objective optimization convert the CMOP into a MOP and solve it by using the MOP technique, which commonly transforms the constraint into another objective function to make the problem into a two-objective unconstrained optimization problem. The advantage of this method is that it can retain a certain amount of infeasible solutions in the solving process by comparing non-inferior solutions, which assists the algorithm to find the optimal global solution. The disadvantage is that multi-objective optimization consumes a large amount of computational resources. Deb [21] proposed a dual-objective constrained optimization algorithm that combines the minimized CV value as an additional objective with the penalty function method.

Since each method cannot be perfect on all kinds of CMOPs, the hybrid method is designed to combine the advantages of two or more algorithms to solve constrained optimization problems. Datta et al. [22] combine a MOP with the penalty function method to solve constrained optimization problems.

Researchers have made many efforts to balance the feasibility, convergence, and diversity in solving CMOPs [23]. However, most CMOEAs prioritize feasibility, so feasible solutions have higher priority than infeasible solutions, which means the algorithm is not able to cross the infeasible region quickly and converge to the final Pareto front (PF). However, infeasible solutions can provide directional information of other feasible regions for search, and become a link between different feasible regions. Allowing them to participate in evolution helps to expand the search scope, which is not only conducive to dealing with the problems of small or disconnected feasible regions, but also conducive to the exploration of the boundary of feasible regions. For this reason, researchers have proposed that different populations assign different priorities for objective functions and constraints, such as in C-TAEA [24], where the convergence-oriented archive (CA) considers constraints, while the diversity-oriented archive (DA) does not. In addition, there are different priorities for objective functions and constraints in different evolutionary stages. For example, PPS [25] does not consider constraints in the push stage and considers all objective functions and constraints in the pull stage. Although these methods help the populations to cross the infeasible regions, the search for infeasible solutions with good objectives is rather limited. Because forward exploration is not powerful enough, the search of infeasible regions may be hindered.

In light of the many challenges facing the constrained multi-objective domain in dealing with CMOPs, we try to investigate the rationality between feasible and infeasible solutions, and make the following research contributions:

1. A novel two-phase evolutionary algorithm is proposed to solve CMOPs. This algorithm aims to dynamically adjust the balance of the objective functions and the constraints through a two-stage strategy. The first phase combines dual-population weak coevolution and complementary environmental selection to improve the algorithm's exploration under constraints. The second phase adapts a feasibility-oriented method to ensure the diversity of the population in the feasibility regions.
2. Systematic experiments verify that our proposed algorithm provides better all-around performance on several constrained multi-objective benchmarks.

The main contents of the rest of this paper are as follows. Section 2 introduces the existing MOEAs in detail and analyzes its advantages and disadvantages. The proposed DTAEA algorithm is introduced in section 3, and the experimental results are presented in detail in section 4. Section 5 presents a comparison of real-world problems. Section 6 is a summary and outlook on the future work.

2. Related work

This section focuses on some basic definitions of CMOPs and existing methods for solving CMOPs. Since our work uses a phased MOEA approach, existing multi-stage optimization methods are mainly reviewed.

2.1. Basic definition

- *Feasible solution*: If the degree of CV value of the decision variable $\mathbf{x}$ is $\mathrm{CV}(\mathbf{x}) = 0$, the decision variable $\mathbf{x}$ is a feasible solution; otherwise, it is an infeasible solution.

- Pareto dominates: There are two solutions $\mathbf{x}$ and $\mathbf{y}$. The necessary and sufficient condition for $\mathbf{x}$ to dominate $\mathbf{y}$ is that for any $i(i = 1, 2, ..., m)$, there is $f_i(\mathbf{x}) \leq f_i(\mathbf{y})$, and for any $i(i = 1, 2, 3...m)$, there is $f_i(\mathbf{x}) < f_i(\mathbf{y})$, denoted as $\mathbf{x} \prec \mathbf{y}$.

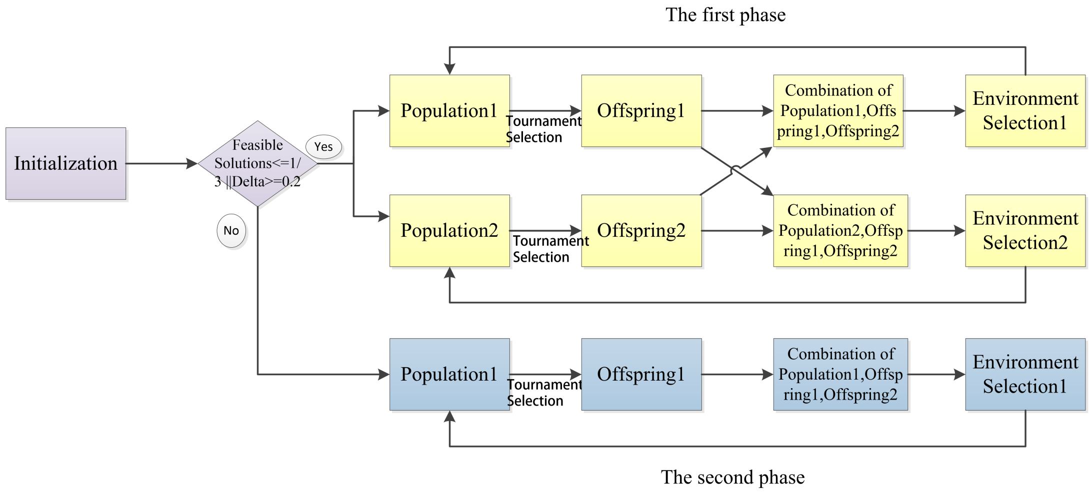

Figure 1: Procedure of the proposed DTAEA.

- Pareto optimal solution: If $\exists \mathbf{x} \in S$, no solution in S that dominates $\mathbf{x}$, $\mathbf{x}$ is said to be a Pareto optimal or non-inferior solution of the multi-objective optimization problem.
- Pareto set: If any two solutions in a set of solutions cannot dominate each other, then the set is called a Pareto set (PS).
- Pareto front: In PS, the set of objective function values corresponding to each solution is called the Pareto front.

2.2. Existing CMOEAs

Constraint optimization requires the solutions have high quality and satisfaction with the constraints. In some cases, if too much emphasis is placed on satisfying the constraints, the search may deviate from the region of good performance; conversely, if too much emphasis is placed on the solution's quality, the search space may deviate from the feasible region. Therefore, it is important to effectively reconcile the imbalance between constraints and objectives based on the problem's a priori information and the knowledge obtained from the search process. There are many CMOEAs for solving CMOPs. These can be divided into the following two categories according to how they balance objective functions and the constraints.

The first category is a static balance of objectives and constraints, where the algorithm is more inclined to feasible solutions, prioritizing constraints and then considering the performance of the objectives. For example, the NSGA-II-CDP [5] algorithm embeds the constraint dominance principle (CDP) in the NSGA-II algorithm, which first selects elite solutions from the feasible population and then from the infeasible population until the population size reaches the upper limit. As an extended version of MOEA/D, CMOEAD [26] embeds the feasibility dominance principle in each sub-region defined by the weight vector, causing each weight to prefer the feasible solutions. MOEA/D-IEpsilon [27] dynamically adjusts the ϵ level according to the ratio of feasible solutions to total solutions in the current population, and all solutions with constraint violation values less than ϵ are considered feasible, then the feasible solutions are always better than infeasible solutions. I-DBEA [28] also gives priority to feasible solutions. It broadens the definition of feasible solutions, regarded as feasible solutions, if solutions with small constraint violation values. In CAEAD [29], population 1 chooses the CDP method for selecting solutions and population 2 uses the epsilon method, both of which are feasible solutions first. This strategy preserves feasible solutions well but loses good quality infeasible solutions and the ability to cross infeasible regions quickly.

The second category is a dynamic balance of objectives and constraints. MOEA/D-DMepsilon [30] uses a new epsilon method to balance the exploration between the feasible regions and infeasible regions dynamically. DD-CMOEA [31] considers no constraints in the exploration stage. But in the exploitation stage, the main population considers constraints, and no constraint is considered in the auxiliary population. TOP [32] considers an objective and

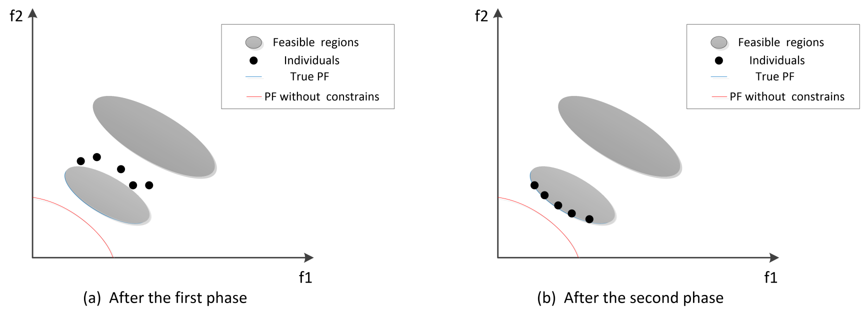

Figure 2: The population distribution of the algorithm in two stages.

all constraints in the first stage, while all objectives and constraints are considered in the second stage. CMOEA-MS [33] uses a two-stage method for solving CMOPs with complex feasible regions, where the solutions in stage A are mostly infeasible. The objectives have the same priority to constrains. In contrast, the solutions are mostly feasible in stage B, and the objectives have lower priority than the constraints for converging the solutions to the feasible region. The selection preference of each individual in DSPCMDE [34] is determined by the weighted sum of objective functions and constraints dynamically, and combines two DE operators with different properties as search algorithms. BiCo [35] is divided into the main population and archive population, the main population considers all constraints, but the archive population does not consider any constraints. Two constraint frameworks [36] [37] transform the original constraint optimization framework into dynamic constraint multi-objective optimization problems. This strategy can handle the balance of objective function and constraints well, but how to grasp this balance needs to be carefully measured.

In addition, coevolutionary frameworks are widely used in constrained multi-objective optimization algorithms, and coevolutionary frameworks perform well in dealing with complex constrained problems. Tian et al. [38] proposed a dual population-based coevolutionary framework (CCMO), in which one population solves constrained MOPs, and another population solves unconstrained MOPs. The collaboration between the two populations in CCMO is much weaker compared to existing coevolutionary algorithms. PDTP-MDE[39] divides the entire evolutionary process into two successive phases, achieving a balance of convergence and diversity in the first phase, and maintaining the viability and diversity of the population using feasible and promising infeasible solutions in the second phase, allowing the two phases to complement each other. In the C-TAEA [24] algorithm, the CA is oriented towards convergence; the DA is oriented towards diversity, and the two populations evolve in a complementary manner. CMOEAs have also been well developed on large-scale problems, such as [40] and [41].

2.3. MOTIVATION

In constrained multi-objective optimization, it is not easy to use the advantages of both feasible and infeasible solutions to optimize the offspring solutions. Using only feasible solutions, or infeasible candidate solutions will result in non-convergent feasible regions or convergent but infeasible offspring solutions.

Problems with small and discontinuous feasible regions usually only consider feasible solutions, while infeasible solutions are ignored. It is difficult for the population to pass through the infeasible region and converge to the final PF. Such is the case with C-TAEA [24], where the main population CA prefers feasible solutions more than infeasible solutions, leading to a final population that does not converge well to the optimal PF. For CCMO [38], the population completely ignores constraints throughout the evolutionary process. Therefore, CCMO may excel in finding solutions close to the unconstrained PF; such a solution may help to enhance the convergence of CCMO on certain problems. However, for problems where the PF lies on the constraint boundary, infeasible solutions close to the unconstrained PF may not be helpful.

To address the limitations of the existing CMOEA, we propose the DTAEA algorithm. In the first stage, the excellent infeasible solutions are well retained by treating the infeasible solutions of the two populations differently. When the proportion of feasible solutions in the population reaches a certain threshold or the convergence of feasible solutions reaches a certain level, the final PF is obtained by using a single-population constrained evolution, allowing the population to converge to the constrained PF, which we refer to as the progressive strategy. The first stage of the DTAEA algorithm ensures enough infeasible solutions to help the population traverse the infeasible region quickly. The second stage allows the final population to converge well on the constrained PF. In the next section, the DTAEA algorithm is described in detail.

3. The proposed algorithm

3.1. Procedure of DTAEA

The general procedure of our proposed DTAEA is shown in Figure 1. When the judgment condition is met, enter the first stage of two-population evolution; when the judgment condition is not satisfied, go to the second stage of single-population development that converges to the final PF. Our desired two-stage algorithm can traverse the infeasible region and converge to the final PF, just like Figure 2. As shown in Algorithm 1, the DTAEA is a two-stage algorithm and starts from the random initialization of two populations: Population1 and Population2 of size N. In the first stage, the parent sets Parent1 and Parent2 are selected from Population1 and Population2, respectively, by the adopted mating selection strategy of the MOEA. Afterward, Population1 and Population2 are combined with two offspring populations and are further truncated by different environmental selections. Only Population1 returns as the evolved population in the second stage and Populaiton1 as the final output.

Algorithm 1 first inputs the population size N and the maximum generation number T, initializes a count t to 0 (line 1), randomly generates Population1 and Population2 (lines 2-3), and calculates the CV value of each solution by formula (2) (line 4). The individual whose CV value is zero is a feasible solution. The proportion pf (line 5) of feasible solutions in Population1 is calculated. Generate two matingpools by tournament selection (lines 8-9). By using simulated binary crossover [42] and polynomial mutation [43] to generate offspring (lines 10-11). Two populations and two offsprings combine to generate new populations and evolve through different environmental selection strategies (lines 12-15). The feasible solution in Population1 is updated and judged on whether the population meets the second stage (line 16). In the second stage, Population2 is discarded, and the final solution set is obtained from the evolution of Population1 alone (lines 18-21).

3.2. The first step

In the first stage, Population1 and Population2 deal with infeasible solutions in different ways. In Population1, by using feasibility-oriented methods, feasible solutions have a higher priority than infeasible solutions. We borrow the idea from MOEA/D-M2M [44] to divide the objective space into N subregions, each of which is represented by a unique weight vector on canonical simplex. The definition of the ω_k is:

$$\omega_{\mathrm{k}} = \{F(\mathbf{x}) \in R^n | \langle F(\mathbf{x}), w_i \rangle \leq \langle F(\mathbf{x}), w_j \rangle\}, \tag{4}$$

where $i, j \in \{1, 2, \dots, N\}$ and $\langle F(\mathbf{x}), w \rangle$ is the acute angle between $F(\mathbf{x})$ and w.

In the proposed feasibility-oriented method, for any unsolvable $\mathbf{x}_i$, the objective vector of the infeasible solution is modified by formula (4), where F^{max} represents the maximum value of each objective of all solutions in the current population, and w_f is the weight vector in the subregion associated with F^{max}. In this way, the infeasible solution linearly converges and is located in the same subregion where F^{max} is located. $\psi'(\mathbf{x}_i)$ is the normalization constraint violation of $\mathbf{x}_i$, divided by $\psi(\mathbf{x}_i)$. It is calculated as the maximum CV value in the population.

$$F'(\mathbf{x}_i) = F^{max} + \psi'(\mathbf{x}_i) \cdot \frac{w_f}{|w_f|}, \tag{5}$$

where

$$\psi'(\mathbf{x}_i) = \frac{\psi(\mathbf{x}_i)}{\Psi_{max}}. \tag{6}$$

This approach is feasibility-oriented. After calculation and modification, all infeasible solutions are dominated by feasible solutions. Therefore, the feasible solutions in Population1 have higher priority than the infeasible solutions. Although the beneficial infeasible solutions may be lost, the well-performing feasible solutions are fully preserved to ensure the feasibility of the population.

Algorithm 1: Procedure of the Proposed DTAEA

```
Input: N(Population size), T(max generation).
Output: Population1
Initialization of the generation counter: t ← 0;
Population1 ← RandomInitialization(N);
Population2 ← RandomInitialization(N);
Calculate constraint violation CV(x);
Calculate the ratio of feasible solutions of Population1 pf;
while t < T do
    if DTAEA is in the first phase then
        Generate MatingPool1 of size N by Tournament Selection;
        Generate MatingPool2 of size N by Tournament Selection;
        Off1 ← MatingPool1(Population1);
        Off2 ← MatingPool2(Population2);
        Population1 ← Population1 ∪ Off1 ∪ Off2 ;
        Population2 ← Population2 ∪ Off1 ∪ Off2 ;
        Use Environmental Selection1 to select the best N solutions into Population1;
        Use Environmental Selection2 to select the best N solutions into Population2;
        Update the ratio of feasible solutions of Population1 pf;
    else
        Generate MatingPool1 of size N by Tournament Selection;
        Off1 ← MatingPool1(Population1);
        Population1 ← Population1 ∪ Off1 ;
        Use Environmental Selection1 to select the best N solutions into Population1;
    end
    t ← t + 1;
end
```

The method to deal with infeasible solutions in Population2 is an adaptive penalty function method, borrowed from [20], which adaptively adjusts the feasible solutions and infeasible solutions to have the same priority by a given index α, which can effectively retain infeasible solutions with advantages, help the population to pass through the infeasible region quickly, and prevent the population from converging in partially feasible regions. Meanwhile, Population1 fully considers the feasibility of solutions. In Population2, infeasible solutions are fully considered. Thus, complementarity is formed by the simultaneous co-evolution of Population1 and Population2. It is important to note that the coordination of Population1 and Population2 differs from most co-evolutionary algorithms. They update populations with the only intersection between offspring, which is a way to weaken cooperation, and the two populations are sufficiently independent but interrelated.

For the adaptive penalty function method, the objective vectors of the infeasible solution are modified by the following formula:

$$F'(\mathbf{x}_i) = F(\mathbf{x}_i) + \psi'(\mathbf{x}_i)^{\alpha} \cdot (F_i^{max} - F(\mathbf{x}_i)), \tag{7}$$

where

$$\alpha = \frac{e^{T/2-t}}{max\{\gamma, 10^{-6}\}}, e \approx 2.7, \tag{8}$$

where $F'(\mathbf{x}_i)$ is the modified objective vector, and $F(\mathbf{x}_i)$ is the original objective vector; $\psi'(\mathbf{x}_i)$ is the normalized CV value of the above formula (5); α is the index of the normalized CV value; F_i^{max} consists of the maximum value of each objective vector in the subregion associated with $F(\mathbf{x}_i)$; T refers to the maximum number of generations; and t refers to the current number of generations, and γ is the Population1 compared with Population2, and the number of unique solutions in Population1 is divided by N. When γ is equal to zero, it is replaced with a positive threshold, 10^{-6}, to avoid a zero denominator.

The objective vectors of the infeasible solution are modified by formula (6), the penalty factor of the adaptive penalty function is calculated by formula (7), and it uses the dynamic penalty function method. The number of current generations t will affect α accordingly. The larger the number of current generations t, the smaller α will be. Conversely, the smaller t is, the larger α will be, and the smaller the number of unique solutions in Population1. That is, the smaller the γ, the larger the α, so the $F'(\mathbf{x}_i)$ are closer to the $F(\mathbf{x}_i)$, and the size of $F'(\mathbf{x}_i)$ is between $F(\mathbf{x}_i)$ and F_i^{max}. By this method, the infeasible solution is biased. Some of the infeasible solutions may even dominate the feasible solution after being modified, which improves the competitiveness of the infeasible solution and allows good quality infeasible solutions to be retained.

The application of this method makes the penalty function of infeasible solutions adaptively adjust according to the evolutionary state and have the same competitiveness as feasible solutions. The infeasible solutions with better competitiveness have suitable objective vectors and small CV values, and their wide distribution takes the diversity of populations into full consideration. Through the two methods, Population1 uses a feasibility-oriented approach to mainly consider feasible solutions. In contrast, Population2 uses an adaptive penalty function approach to retain the infeasible solutions with advantages, and the two populations complement each other through cooperation between the offspring.

The most serious difficulty of two-stage evolution is when to enter the second stage. That is, the stopping criterion of the first stage needs to be fully considered. In the first stage, the infeasible solutions with great objective vectors and small CVs are fully preserved through the cooperation of the two populations. It is evident that it is not desirable to keep evolving in this way, and the first stage should be terminated after the feasible solutions and infeasible solutions in the population reach a certain level, essentially, when at the end of the first stage, we expect to obtain some high-quality feasible solutions and infeasible solutions but maintain good diversity. In this case, to obtain high-quality feasible solutions, we design the following two conditions:

Condition 1: The proportion of feasible solutions in the current Population1 (pf) is more significant than 1/3 of the current Population1.

Condition 2: The maximum objective function difference of the first 1/3 of feasible solutions in Population1 is less than 0.2; $f_{max,j}$ and $f_{min,j}$ represent the maximum and minimum values of the jth objective function of the feasible solution, and $\bar{F}_j(\mathbf{x}_i)$ represent each feasible solution in the population of the jth objective function of $\mathbf{x}_i$. Next, add up all the $\bar{F}_j(\mathbf{x}_i)$, $j\epsilon(1,\ldots,m)$ and obtain $\bar{F}(\mathbf{x}_i)$:

$$\bar{F}(\mathbf{x}_i) = \sum_{j=1}^{m} \frac{f_j(\mathbf{x}_i) - f_{\min,j}}{f_{\max,j} - f_{\min,j}}, \tag{9}$$

where $\bar{F}(\mathbf{x}_i)$ is the sum of the objective vectors after normalization. Then sort the feasible solutions according to $\bar{F}(\mathbf{x}_i)$, select the first 1/3 feasible solutions to calculate the biggest difference. If the difference is less than 0.2, the second condition is judged to be satisfied.

The purpose of condition 1 is to ensure that some feasible solutions are obtained. In addition, condition 2 indicates that some feasible solutions gradually converge to a small range. If only feasible solutions are considered, it may lead to poor population convergence, while if only convergence is considered, it may lead to few feasible solutions in the population. To avoid the population converging in the unconstrained PF or local feasible region, enter the second stage as soon as condition 1 or condition 2 is satisfied.

3.3. The second step

Furthermore, we observed that all dual-population algorithms end up with one of the populations as the final output, so we propose a new progressive strategy by considering the evolution of only a single population at a later stage of the algorithm. Due to the aforementioned advantages, the first stage has the potential to explore promising feasible regions and can quickly traverse infeasible regions. However, the ultimate goal is to find the Pareto optimal solution of the original CMOP, with the final population individuals distributed on the PF.

When the two populations run to the later stage in the manner of weak co-evolution, because the intersection of the two populations is only produced between the offspring, the independence of the two populations is very strong, and the possibility of mutual influence is slight. Therefore, we propose a progressive strategy, consider ignore the evolution of Population2, only Population1 evolution to converge to the final PF. As shown in Figure 3, the DOC1 test problem is compared with two situations. One is the dual population evolution converging to the final PF, and the other is the dual population evolution first and then the single population evolution converging to the final PF. It can be seen from

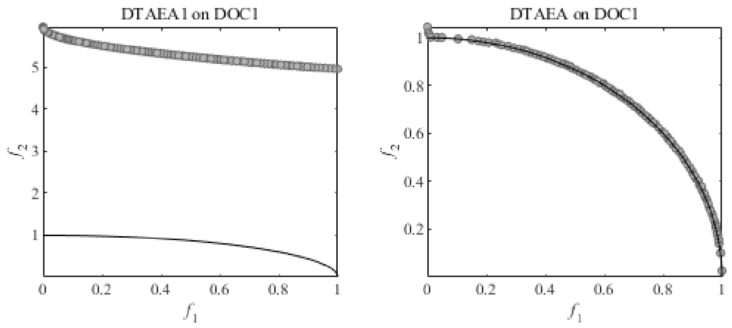

Figure 3: The final population of dual-population evolution and two-stage evolution on the DOC test problem.

the Figure 3 that the dual-population evolutionary method cannot converge to the correct PF. Obviously, the two-stage evolution method can converge very well. Therefore, we conclude the two-stage strategy is effective.

Since judgment condition 2 only considers the first 1/3 of feasible solutions in the second stage, some solutions may be far from the Pareto optimal solution. Therefore, the evolutionary method of the first stage Population1 is used to continue the evolutionary convergence of individual populations to the final PF without wasting redundant computational resources.

3.4. Analysis of DTAEA

The DTAEA algorithm uses a two-stage strategy that takes full consider of the balance of objective functions and constraints. In the first stage, different populations use different methods to balance objectives and constraints dynamically. The two populations are relatively independent, which is more conducive to the rapid convergence of the populations. In the second stage, as the populations mostly fall near the final PF after the first stage, the infeasible solution is of little help at this time, so only the priority of the constraint is considered to be greater than the objective function, that is, the priority of the feasible solution is greater than infeasible solution, which makes the populations converge better on the constrained PF. Thus, the principle of our proposed two-stage algorithm is close to the ideal state of Figure 2.

CCMO is a typical optimization algorithm that ignore constraints in the evolutionary process and can find solutions close to the unconstrained PF well. Despite this, the performance of CCMO is not performing well for test problems that separate unconstrained PF and constraint PF. At the same time, DTAEA not only fully considers retaining the advantageous infeasible solutions to help the population cross the infeasible region but also helps the population converge to the final PF through a progressive strategy.

Compared with the c-DPEA algorithm, the DTAEA algorithm uses a two-stage strategy to guarantee the priority of feasible solutions while fully considering infeasible solutions. Its collaborative framework and the use of a two-stage approach accelerate the convergence of the population. The balancing effect of decision variables and objective variables is significantly improved.

3.5. Computational Complexity of DTAEA

The worst-case complexity analysis of the main operations in DTAEA is as follows. In the first phase, first, generate two matingpools takes $2 \times O(N) = O(N)$. Next, offspring generation are $2 \times O(N/2) = O(N)$. For environmental selection, the time complexity takes $2 \times O(MN^2) = O(MN^2)$, M is the number of objectives. The second phase is faster than the first phase since it is developed by a single population. Generating one matingpool takes $O(N) = O(N)$. Next, offspring generation are $O(N/2) = O(N)$. For environmental selection, the time complexity takes $O(MN^2) = O(MN^2)$.

4. EXPERIMENTAL STUDY

This section verifies the performance of the proposed DTAEA by comparing it to the C-TAEA [24], CCMO [38], DD-CMOEA [31], PPS [25], and TOP [32] on the C-DTLZ [26] [45], DC-DTLZ [24], MW [46], DOC [32], and DASCMOP [47]. The experimental results are in this section's discussion.

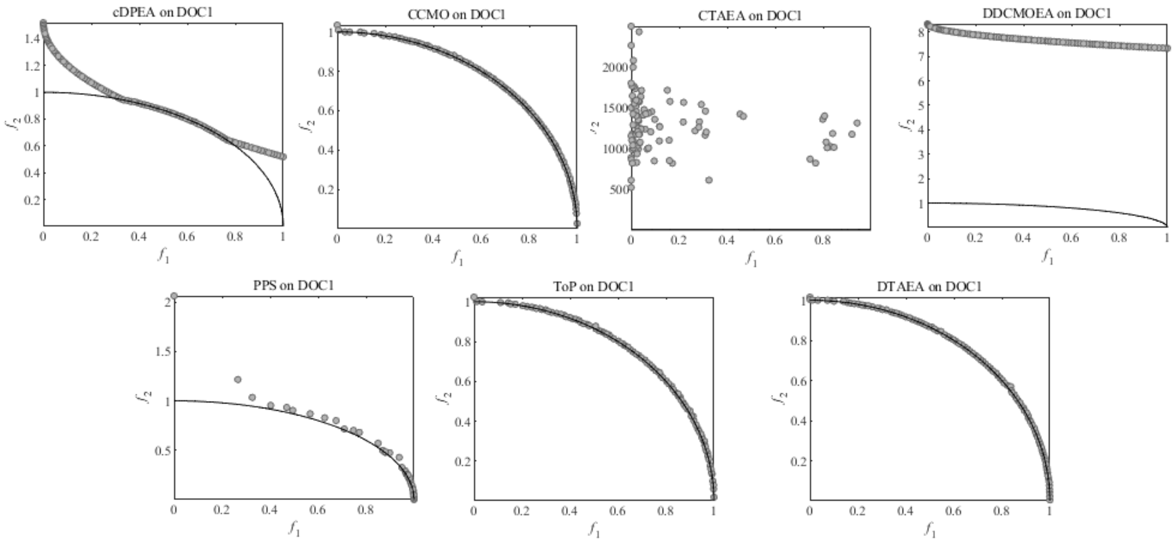

Figure 4: The final populations with median IGD value among 30 runs obtained by c-DPEA, CCMO, C-TAEA, DD-CMOEA, PPS, TOP and DTAEA on DOC1.

Table I.IGD value of c-DPEA, C-TAEA, CCMO, DD-CMOEA, PPS, TOP AND DTAEA on Constrained DTLZ, MW AND DOC Problems, Best Result IN Each ROW IS HIGHLIGHTED. "NaN" INDICATES THAT NO FEASIBLE SOLUTION IS FOUND, "+", "-" AND "≈" INDICATE THAT THE RESULT IS SIGNIFICANTLY BETTER, SIGNIFICANTLY WORSE, OR STATISTICALLY SIMILAR TO DTAEA.

Problem	c-DPEA	C-TAEA	CCMO	DD-CMOEA	PPS	ToP	DTAEA
C1_DTLZ1	2.0015e-2 (1.64e-4) ≈	2.3237e-2 (2.02e-4) −	1.9998e-2 (1.45e-4) ≈	2.1009e-2 (2.16e-4) −	2.5829e-2 (8.10e-4) −	NaN (NaN)	1.9936e-2 (1.42e-4)
C1_DTLZ3	5.3723e-2 (5.64e-4) +	7.2769e-2 (1.63e-2) +	5.3561e-2 (6.73e-4) +	5.6295e-2 (7.54e-4) +	2.7243e+0 (3.81e+0) ≈	1.0177e+0 (2.40e+0) ≈	4.3292e+0 (4.01e+0)
C2_DTLZ2	4.2895e-2 (6.09e-4) −	5.6315e-2 (9.22e-4) −	4.2797e-2 (4.82e-4) −	4.5133e-2 (5.24e-4) −	5.5199e-2 (1.45e-3) −	5.9919e-2 (4.56e-3) −	4.2275e-2 (5.65e-4)
C3_DTLZ4	9.6888e-2 (1.46e-3) +	1.1199e-1 (2.59e-3) +	1.4469e-1 (1.89e-1) ≈	1.9927e-1 (2.56e-1) −	1.6827e-1 (9.75e-2) −	1.4096e-1 (5.15e-3) −	1.2013e-1 (1.36e-1)
DC1_DTLZ1	1.1421e-2 (8.06e-5) −	1.5303e-2 (3.22e-4) −	1.1426e-2 (1.05e-4) −	1.2092e-2 (1.35e-4) −	3.2425e-2 (8.49e-3) −	2.0938e-2 (6.07e-3) −	1.1355e-2 (5.89e-5)
DC1_DTLZ3	3.3918e-2 (4.41e-4) ≈	4.3315e-2 (1.19e-3) −	3.3882e-2 (4.57e-4) ≈	3.5797e-2 (3.57e-4) −	3.4905e-1 (2.13e-1) −	9.9078e-1 (1.61e+0) −	3.3846e-2 (3.65e-4)
DC2_DTLZ1	2.0144e-2 (1.49e-4) +	2.4468e-1 (1.18e-1) −	2.0136e-2 (1.31e-4) +	2.1094e-2 (2.70e-4) ≈	4.2231e-2 (4.39e-2) ≈	NaN (NaN)	9.7154e-2 (7.33e-2)
DC2_DTLZ3	1.5514e-1 (2.08e-1) +	NaN (NaN)	7.2696e-1 (6.21e-2) −	5.5752e-2 (3.78e-4) ≈	5.0213e-1 (1.86e-1) −	NaN (NaN)	3.2791e-1 (2.57e-1)
DC3_DTLZ1	6.8715e-3 (6.71e-5) +	9.3850e-3 (2.68e-4) +	6.8179e-3 (5.18e-5) +	7.2598e-3 (7.51e-5) +	1.0326e+0 (1.98e+0) ≈	2.3341e+0 (2.09e+0) −	1.4867e-1 (8.54e-2)
DC3_DTLZ3	2.1703e-1 (2.63e-1) +	2.6301e-2 (6.24e-4) +	1.9996e-2 (2.57e-4) +	2.1198e-2 (3.53e-4) +	2.7060e+0 (2.13e+0) −	8.6514e+0 (3.74e+0) −	1.3072e+0 (4.42e-1)
MW1	1.6152e-3 (1.27e-5) +	2.1824e-3 (9.71e-4) −	1.6180e-3 (1.07e-5) +	1.6176e-3 (1.34e-5) +	3.3096e-3 (6.60e-4) −	6.2107e-1 (1.03e-1) −	1.7825e-3 (1.01e-3)
MW2	2.2387e-2 (1.55e-2) ≈	1.7020e-2 (8.02e-3) ≈	2.0396e-2 (1.54e-2) ≈	2.0203e-2 (6.44e-3) ≈	1.3139e-1 (7.93e-2) −	2.0222e-1 (2.16e-1) −	1.6853e-2 (7.35e-3)
MW3	4.9640e-3 (3.05e-4) ≈	5.0973e-3 (3.57e-4) ≈	9.5427e-3 (1.36e-3) −	4.5534e-3 (2.47e-4) +	6.3603e-3 (5.05e-4) −	5.5944e-1 (3.87e-1) −	4.9606e-3 (2.61e-4)
MW4	4.0614e-2 (3.16e-4) −	4.6580e-2 (3.22e-4) −	6.4457e-2 (9.68e-3) −	4.2809e-2 (3.72e-4) −	5.7345e-2 (1.71e-3) −	8.4074e-1 (0.00e+0) ≈	4.0404e-2 (3.63e-4)
MW5	1.0166e-3 (1.40e-3) ≈	9.4116e-2 (1.94e-2) −	2.6373e-2 (1.25e-2) −	7.4686e-4 (9.40e-4) ≈	4.7557e-1 (3.53e-1) −	8.8184e-1 (0.00e+0) ≈	1.0661e-3 (1.07e-3)
MW6	1.2726e-2 (7.23e-3) ≈	2.0580e-2 (5.57e-3) +	1.8121e-2 (9.06e-3) +	2.3640e-2 (1.21e-2) +	5.7783e-1 (2.84e-1) −	7.7279e-1 (3.41e-1) −	4.0993e-2 (1.02e-1)
MW7	4.3952e-3 (2.43e-4) +	1.3984e-2 (1.18e-3) −	1.3554e-2 (3.01e-3) −	4.2726e-3 (2.83e-4) +	5.5266e-3 (4.46e-4) −	8.1806e-2 (1.77e-1) −	4.2677e-3 (1.92e-4)
MW8	4.5437e-2 (4.61e-3) ≈	5.4469e-2 (3.48e-3) −	5.3654e-2 (1.90e-2) −	5.0550e-2 (4.77e-3) −	1.6300e-1 (1.02e-1) −	7.1353e-1 (3.45e-1) −	4.4982e-2 (2.96e-3)
MW9	2.7682e-2 (1.25e-1) −	3.1205e-2 (1.20e-2) −	1.6056e-2 (6.33e-2) −	2.7862e-2 (1.29e-1) −	8.8331e-2 (2.12e-1) −	6.5753e-1 (2.89e-1) −	8.8326e-3 (1.45e-3)
MW10	1.8788e-2 (1.36e-2) ≈	2.9337e-2 (1.17e-2) ≈	3.8929e-2 (3.06e-2) ≈	5.7810e-2 (1.08e-1) ≈	4.7948e-1 (2.26e-1) −	NaN (NaN)	3.2011e-2 (2.62e-2)
MW11	6.1951e-3 (1.60e-4) −	2.5172e-2 (3.40e-3) −	4.0308e-2 (6.71e-2) −	5.9643e-3 (1.43e-4) ≈	7.3800e-3 (4.37e-4) −	5.9076e-1 (1.99e-1) −	6.0244e-3 (9.68e-5)
MW12	4.8865e-3 (1.17e-4) −	2.8224e-2 (1.32e-2) −	4.9671e-2 (1.71e-1) −	4.7967e-3 (4.28e-4) −	9.8480e-2 (2.09e-1) −	7.7151e-1 (1.14e-1) −	4.7839e-3 (9.39e-5)
MW13	4.3567e-2 (2.82e-2) +	1.0036e-1 (1.92e-2) ≈	8.2437e-2 (3.71e-2) ≈	7.4030e-2 (3.31e-2) ≈	4.1489e-1 (2.74e-1) −	8.0331e-1 (4.04e-1) −	9.3628e-2 (5.67e-2)
MW14	9.8011e-2 (1.93e-3) −	8.9551e-1 (1.34e-1) −	9.7689e-2 (1.71e-1) −	1.0350e-1 (2.30e-3) −	1.4295e-1 (1.76e-2) −	3.9863e-1 (3.89e-1) −	9.6088e-2 (1.59e-3)
DOC1	2.7709e-1 (3.33e-1) −	5.4644e+2 (3.20e+2) −	5.7976e-3 (5.40e-4) −	6.0359e+0 (4.77e+0) −	7.7935e-2 (5.58e-2) −	6.6691e-3 (7.47e-4) −	4.9446e-3 (2.53e-4)
DOC2	NaN (NaN)	NaN (NaN)	3.9145e-2 (1.20e-1) −	NaN (NaN)	3.8793e-1 (2.23e-1) −	NaN (NaN)	5.9601e-3 (5.23e-3)
DOC3	5.8595e+2 (1.63e+2) −	NaN (NaN)	5.6912e+2 (4.26e+2) ≈	6.5190e+2 (2.36e+2) ≈	2.5563e+2 (1.62e+2) +	1.1981e+2 (2.17e+2) +	5.4049e+2 (3.60e+2)
DOC4	4.6912e-1 (1.65e-1) −	2.2334e+3 (6.93e+3) −	2.6218e-2 (7.99e-3) −	1.7214e+0 (1.62e+0) −	3.4714e-1 (9.50e-2) −	2.9264e-1 (2.90e-1) −	1.5523e-2 (2.41e-3)
DOC5	NaN (NaN)	NaN (NaN)	5.1593e-2 (5.50e-2) −	NaN (NaN)	NaN (NaN)	NaN (NaN)	2.0531e-2 (5.23e-3)
DOC6	2.6054e+0 (2.40e+0) −	3.2870e+2 (8.21e+2) −	7.5200e-2 (1.80e-1) −	2.9779e+0 (2.91e+0) −	5.2351e-1 (8.63e-2) −	1.5120e+1 (8.47e+0) −	2.5485e-3 (8.91e-5)
DOC7	7.7434e+0 (2.46e+0) −	NaN (NaN)	2.6121e-3 (3.03e-4) ≈	5.7572e+0 (2.66e+0) −	5.6368e-1 (2.31e-1) −	NaN (NaN)	5.2541e-1 (1.04e+0)
DOC8	6.6434e+1 (4.96e+1) −	NaN (NaN)	7.2819e-2 (5.66e-3) −	1.1123e+2 (1.06e+2) −	1.2738e+2 (5.00e+1) −	1.8145e+2 (9.46e+1) −	5.7538e-2 (2.95e-3)
DOC9	1.9510e-1 (1.15e-1) −	NaN (NaN)	8.0441e-2 (1.24e-2) ≈	1.9171e-1 (1.19e-1) ≈	3.4343e-1 (4.09e-2) −	3.3147e-1 (9.86e-2) −	9.5631e-2 (5.42e-2)
+/ − / ≈	9/14/8	5/17/4	6/18/9	7/15/9	1/28/3	1/22/3	

First, the proposed DTAEA was compared with the most advanced CMOEAs. Then, the DTAEA's effectiveness in dealing with CMOPs was verified. Finally, the performance of the algorithm was further analyzed. All experiments were conducted on the evolutionary multi-objective optimization platform PLATEMO[48].

4.1. EXPERIMENTAL SETTING

1. Test problems: The number of objectives and the number of decision variables for each test problem were set as follows: For the DTLZ test problems, C1-DTLZ1, DC1-DTLZ1, DC2-DTLZ1, DC3-DTLZ1 had $M = 3$, $D = 7$.

For the remaining problems, $D = 12$; for 14 MW problems, $M = 3$ for MW4, MW8 and MW14, and $M = 2, D = 15$ for the remaining problems; for the 9 DOC problems, $M = 3$ for DOC8 and DOC9, and for the others, $M = 2$. D was different for different problems.

2. Population size and the number of function evaluations: All population sizes were set to 100. The function evaluation number of all populations was used as the termination criterion for all comparisons of MOEA, and the criterion was set to a large enough value so that each MOEA could converge. Specifically, for restricted DTLZ problems, MW problems, and DASCMOP problems, the number of function evaluations was set to 100,000, and for DOC problems, the number of function evaluations was set to 300,000, and the type was 2.
3. Algorithm: The parameter settings of all comparison algorithms were consistent with those in the original text, so the algorithm would perform at its best. For the PPS algorithm, the $\alpha = 0.95$, $tau = 0.1$, $cp = 2$, and $l = 20$. For the TOP algorithm, the $pf = 1/3$, and the $\delta = 0.2$. PPS and TOP use differential evolution [49] and polynomial mutation to generate offspring solutions [42], but c-DPEA, C-TAEA, CCMO, and DD-CMOEA use simulated binary crossover [50] and polynomial mutation to generate offspring solutions. The crossover probability was set to 1; the mutation probability was set to $1/d$ (d is the number of decision variables), and the distribution index of crossover and mutation was set to 20. The parameters CR and F of differential evolution were set to 1 and 0.5, respectively.

4.2. Comparative experimental results

We usually use the calculated inverted generational distance (IGD) [51] value and hypervolume (HV) [52] to reflect the performance of different algorithms.

1. IGD can simultaneously reflect the convergence and diversity of the algorithm. The definition of IGD is as follows[53]:

$$\begin{cases} IGD(P^*, A) = \frac{\sum_{y^* \in P^*} d(y^*, A)}{|P^*|} \\ d(y^*, A) = \min_{y \in A} \left\{ \sqrt{\sum_{i=1}^{m} \left(y_i^* - y_i\right)^2} \right\} \end{cases}, \tag{10}$$

   where P^* represents a set of representative solutions in the actual PF; A is the approximate PF realized by CMOEA and m represents the number of objectives. The smaller the IGD value, the better the diversity and convergence. In contrast, the larger the IGD value, the worse the convergence and diversity.
2. Hypervolume index (HV): The volume of the area in the objective space enclosed by the nondominated solution set and the reference point obtained by the algorithm. HV reflects the closeness of the group of nondominated solutions achieved by a CMOEA to the real PF. The larger the HV value, the better the overall performance of the algorithm. The definition of HV is as follows:

$$HV(S) = \mathrm{VOL}\left(\bigcup_{x \in S} \left[f_1(x), z_1^r\right] \times \ldots \left[f_m(x), z_m^r\right]\right), \tag{11}$$

   where $\mathrm{VOL}(\cdot)$ is the Lebesgue measure; m denotes the number of objectives; $z^r = (z_1^r, \ldots, z_m^r)^T$ is a user-defined reference point in the objective space.

4.3. DISCUSSION

In this subsection, DTAEA is compared with four state-of-the-art dual-population algorithms and two state-of-the-art single-population algorithms. We quantitatively evaluate their performance using two widely used metrics, the inverse intergenerational distance (IGD) and the excess capacity (HV). The symbols '+', '-' or '≈' imply that the corresponding competitors outperform, underperform, or are comparable to DTAEA. In addition, the best metric for each issue is highlighted in gray in each table.

Table I shows that c-DPEA [20], C-TAEA [24], CCMO [38], DD-CMOEA [31], PPS [25], TOP [32], and DTAEA performed 30 independent runs on the constraint DTLZ problem, MW problem, and DOC problem. According to the method suggested in [54] for sampling on the PF of the problem, about 10,000 reference points were used to calculate the IGD value of each problem. The Table I shows that the proposed DTAEA algorithm performed well on most of the test problems. There were 10 DTLZ test problems, 14 MW test problems, and 9 DOC test problems. On these 33 test

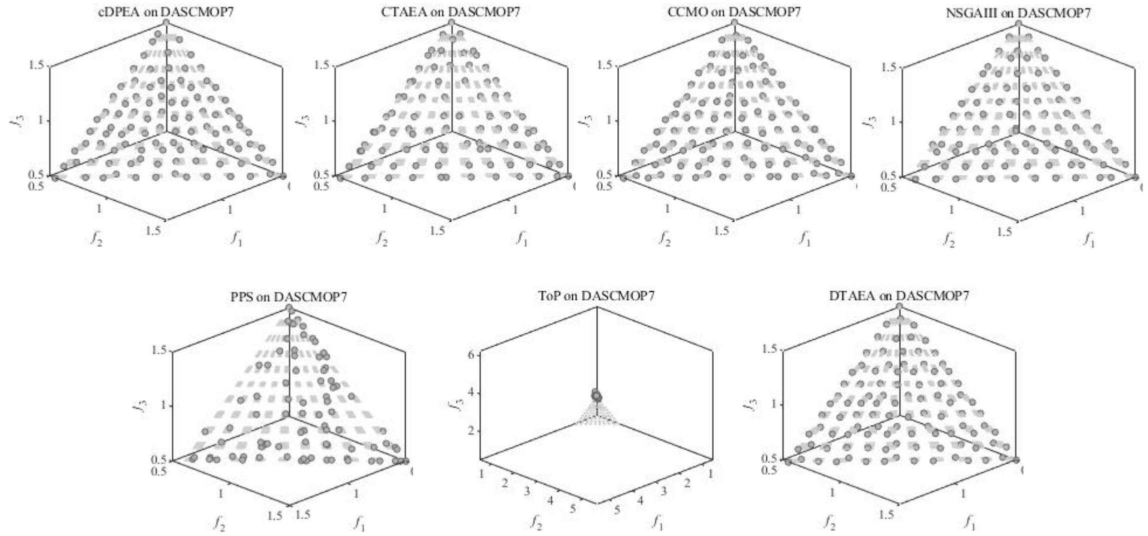

Figure 5: The final populations with median IGD value among 30 runs obtained by c-DPEA, C-TAEA, CCMO, DD-CMOEA, PPS, TOP and DTAEA on DASCMOP7.

Table II. HV value of c-DPEA,C-TAEA, CCMO, DD-CMOEA, PPS, TOP AND DTAEA on Constrained DTLZ, MW AND DOC Problems, Best Result IN Each ROW IS HIGHLIGHTED. "NaN"INDICATES THAT NO FEASIBLE SOLUTION IS FOUND, "+", "-" AND "≈" INDICATE THAT THE RESULT IS SIGNIFICANTLY BETTER, SIGNIFICANTLY WORSE, OR STATISTICALLY SIMILAR TO DTAEA.

Problem	c-DPEA	C-TAEA	CCMO	DD-CMOEA	PPS	ToP	DTAEA
C1_DTLZ1	8.4114e-1 (1.72e-3) ≈	8.3701e-1 (1.74e-3) −	8.4084e-1 (3.80e-3) ≈	8.4022e-1 (9.20e-4) −	8.1784e-1 (4.68e-3) −	NaN (NaN)	8.4123e-1 (2.27e-3)
C1_DTLZ3	5.6026e-1 (1.38e-3) +	5.4062e-1 (1.26e-2) +	5.6075e-1 (1.02e-3) +	5.5793e-1 (1.23e-3) +	3.4514e-1 (2.48e-1) ≈	3.2181e-1 (2.12e-1) ≈	2.3421e-1 (2.74e-1)
C2_DTLZ2	5.1575e-1 (1.38e-3) −	5.0763e-1 (1.19e-3) −	5.1554e-1 (1.46e-3) −	5.1336e-1 (1.89e-3) −	4.9901e-1 (3.22e-3) −	4.6654e-1 (1.29e-2) −	5.1801e-1 (9.62e-4)
C3_DTLZ4	7.8848e-1 (1.15e-3) +	7.8522e-1 (1.26e-3) +	7.7318e-1 (6.40e-2) ≈	7.5427e-1 (8.65e-2) −	7.5776e-1 (2.74e-2) −	7.5648e-1 (4.00e-3) −	7.8164e-1 (4.60e-2)
DC1_DTLZ1	6.3279e-1 (7.26e-4) ≈	6.2728e-1 (1.24e-3) −	6.3263e-1 (6.65e-4) ≈	6.3116e-1 (1.20e-3) −	5.7095e-1 (2.39e-2) −	5.8471e-1 (2.26e-2) −	6.3293e-1 (5.06e-4)
DC1_DTLZ3	4.7440e-1 (6.76e-4) −	4.6233e-1 (1.99e-3) −	4.7424e-1 (7.49e-4) −	4.7330e-1 (1.02e-3) −	2.9228e-1 (1.38e-1) −	1.6510e-1 (1.61e-1) −	4.7479e-1 (1.00e-3)
DC2_DTLZ1	8.4233e-1 (4.05e-4) +	2.7506e-1 (2.40e-1) −	8.4238e-1 (4.17e-4) +	8.4077e-1 (5.73e-4) ≈	7.6987e-1 (1.16e-1) ≈	NaN (NaN)	6.4826e-1 (1.84e-1)
DC2_DTLZ3	4.5121e-1 (2.23e-1) +	NaN (NaN)	0.0000e+0 (0.00e+0) −	5.5799e-1 (1.39e-3) +	8.6296e-2 (1.87e-1) −	NaN (NaN)	2.6159e-1 (2.71e-1)
DC3_DTLZ1	5.3582e-1 (1.05e-3) +	5.2312e-1 (2.82e-3) +	5.3616e-1 (7.87e-4) +	5.3543e-1 (1.08e-3) +	2.4436e-1 (2.11e-1) ≈	1.0591e-2 (5.80e-2) −	1.8233e-1 (1.82e-1)
DC3_DTLZ3	2.3312e-1 (1.80e-1) +	3.6032e-1 (1.86e-3) +	3.6797e-1 (9.27e-4) +	3.6751e-1 (1.40e-3) +	9.8942e-3 (3.77e-2) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)
MW1	4.9009e-1 (3.45e-5) ≈	4.8849e-1 (1.98e-3) −	4.9005e-1 (7.60e-5) +	4.9002e-1 (9.99e-5) +	4.8652e-1 (1.73e-3) −	0.0000e+0 (0.00e+0) −	4.8972e-1 (2.04e-3)
MW2	5.5111e-1 (2.23e-2) ≈	5.5961e-1 (1.25e-2) ≈	5.5457e-1 (2.25e-2) ≈	5.5388e-1 (1.03e-2) ≈	4.1132e-1 (8.84e-2) −	3.6613e-1 (1.58e-1) −	5.5966e-1 (1.33e-2)
MW3	5.4420e-1 (6.33e-4) ≈	5.4452e-1 (6.06e-4) ≈	5.3729e-1 (2.12e-3) −	5.4481e-1 (4.79e-4) +	5.4279e-1 (8.20e-4) −	1.9039e-1 (2.04e-1) −	5.4440e-1 (6.53e-4)
MW4	8.4193e-1 (3.93e-4) −	8.3820e-1 (2.49e-4) −	8.0735e-1 (1.21e-2) −	8.4002e-1 (4.27e-4) −	8.1329e-1 (5.16e-3) −	8.1378e-2 (0.00e+0) ≈	8.4236e-1 (3.13e-4)
MW5	3.2423e-1 (5.62e-4) ≈	2.1436e-1 (2.42e-2) −	3.0104e-1 (9.27e-3) −	3.2426e-1 (5.06e-4) ≈	1.7202e-1 (1.08e-1) −	0.0000e+0 (0.00e+0) ≈	3.2427e-1 (3.62e-4)
MW6	3.1294e-1 (1.04e-2) ≈	3.0240e-1 (8.58e-3) −	3.0487e-1 (1.25e-2) +	2.9782e-1 (1.66e-2) −	9.1671e-2 (7.93e-2) −	6.6122e-2 (7.67e-2) −	3.0433e-1 (2.53e-2)
MW7	4.1225e-1 (3.55e-4) +	3.9844e-1 (2.40e-3) −	4.0636e-1 (1.51e-3) −	4.1213e-1 (6.45e-4) ≈	4.1178e-1 (4.60e-4) −	3.6009e-1 (7.47e-2) −	4.1248e-1 (3.68e-4)
MW8	5.3879e-1 (1.38e-2) ≈	5.1541e-1 (1.26e-2) −	5.1332e-1 (3.81e-2) −	5.2313e-1 (1.54e-2) −	3.2151e-1 (1.07e-1) −	1.2237e-1 (1.45e-1) −	5.3936e-1 (1.38e-2)
MW9	3.8513e-1 (7.28e-2) −	3.6027e-1 (1.28e-2) −	2.9006e-1 (1.63e-1) −	3.8640e-1 (7.30e-2) −	3.3698e-1 (1.16e-1) −	5.3934e-2 (1.21e-1) −	3.8921e-1 (2.70e-3)
MW10	4.3487e-1 (1.38e-2) ≈	4.2288e-1 (9.96e-3) ≈	4.1780e-1 (2.42e-2) ≈	4.0836e-1 (5.66e-2) ≈	1.8912e-1 (1.03e-1) −	NaN (NaN)	4.2333e-1 (2.08e-2)
MW11	4.4750e-1 (2.21e-4) −	4.3650e-1 (2.09e-3) −	4.3129e-1 (2.54e-2) −	4.4790e-1 (1.54e-4) ≈	4.4744e-1 (1.55e-4) −	2.8990e-1 (4.07e-2) −	4.4782e-1 (1.88e-4)
MW12	6.0472e-1 (1.47e-4) −	5.7243e-1 (1.72e-2) −	5.1789e-1 (2.07e-1) −	6.0482e-1 (7.52e-4) ≈	5.1440e-1 (1.86e-1) −	1.5347e-2 (3.07e-2) −	6.0495e-1 (1.74e-4)
MW13	4.5857e-1 (1.28e-2) +	4.3404e-1 (1.10e-2) ≈	4.3919e-1 (2.05e-2) ≈	4.4360e-1 (1.72e-2) ≈	2.6771e-1 (1.04e-1) −	2.0007e-1 (1.08e-1) −	4.3393e-1 (2.94e-2)
MW14	4.7293e-1 (1.81e-3) −	1.1424e-1 (3.35e-2) −	3.3088e-1 (7.56e-2) −	4.7151e-1 (1.94e-3) −	4.4624e-1 (6.55e-3) −	3.4057e-1 (1.56e-1) −	4.7462e-1 (1.34e-3)
DOC1	1.9803e-1 (1.21e-1) −	0.0000e+0 (0.00e+0) −	3.4459e-1 (6.22e-4) −	1.5031e-2 (4.77e-2) −	2.8250e-1 (3.33e-2) −	3.4389e-1 (5.11e-4) −	3.4551e-1 (3.27e-4)
DOC2	NaN (NaN)	NaN (NaN)	5.9210e-1 (9.33e-2) −	NaN (NaN)	2.7131e-1 (1.43e-1) −	NaN (NaN)	6.1846e-1 (7.33e-3)
DOC3	0.0000e+0 (0.00e+0) ≈	NaN (NaN)	2.3975e-2 (7.86e-2) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	1.2075e-2 (4.38e-2) ≈	9.4331e-3 (5.08e-2)
DOC4	1.4483e-1 (1.08e-1) −	0.0000e+0 (0.00e+0) −	5.3281e-1 (9.04e-3) −	5.9601e-2 (1.24e-1) −	2.1275e-1 (7.30e-2) −	3.0525e-1 (2.34e-1) −	5.4566e-1 (2.91e-3)
DOC5	NaN (NaN)	NaN (NaN)	4.6829e-1 (3.15e-2) −	NaN (NaN)	NaN (NaN)	NaN (NaN)	4.8926e-1 (4.78e-3)
DOC6	2.1786e-2 (9.21e-2) −	0.0000e+0 (0.00e+0) −	4.4018e-1 (1.58e-1) −	1.0435e-2 (5.72e-2) −	1.4809e-1 (4.77e-2) −	0.0000e+0 (0.00e+0) −	5.4207e-1 (6.23e-3)
DOC7	0.0000e+0 (0.00e+0) −	NaN (NaN)	5.2475e-1 (1.20e-2) ≈	0.0000e+0 (0.00e+0) −	7.5310e-2 (1.03e-1) −	NaN (NaN)	3.7490e-1 (2.49e-1)
DOC8	0.0000e+0 (0.00e+0) −	NaN (NaN)	7.8680e-1 (7.23e-3) −	0.0000e+0 (0.00e+0) −	0.0000e+0 (0.00e+0) −	0.0000e+0 (0.00e+0) −	8.0661e-1 (3.55e-3)
DOC9	NaN (NaN)	NaN (NaN)	NaN (NaN)	NaN (NaN)	0.0000e+0 (0.00e+0)	0.0000e+0 (0.00e+0)	NaN (NaN)
+/ − / ≈	8/12/10	4/18/4	6/18/8	6/15/9	0/26/5	0/20/5	

problems, the DTAEA algorithm showed the best performance on seventeen problems; followed by c-DPEA achieving nine best results; C-TAEA achiecing five best results; and CCMO, DD-CMOEA achieving six best results and seven best results respectively; PPS, TOP only achieved one best result on the 33 problems.

Figure 4 plots the final population distribution obtained by running the c-DPEA, CCMO, C-TAEA, DD-CMOEA, PPS, TOP and DTAEA on the DOC1 test problem for 30 times. C-TAEA and DD-CMOEA did not converge to the real PF on the DOC1 test problem. Only some populations of c-DPEA and PPS converged on PF. In CCMO and TOP, most of their populations converged on PF, a minimal number still fell on the edge of PF. DTAEA converged better

Table III. The IGD value of DTAEAc,DTAEAd and DTAEA on MW test problem.

Problem	DTAEAc	DTAEAd	DTAEA
MW1	1.6146e-3 (3.14e-5) +	1.7766e-3 (9.93e-4) ≈	1.7825e-3 (1.01e-3)
MW2	1.7612e-2 (6.70e-3) ≈	2.1462e-2 (1.54e-2) ≈	1.6853e-2 (7.35e-3)
MW3	3.5559e-2 (1.67e-1) ≈	5.0979e-3 (3.70e-4) ≈	4.9606e-3 (2.61e-4)
MW4	4.0703e-2 (4.00e-4) −	4.0304e-2 (2.94e-4) ≈	4.0404e-2 (3.63e-4)
MW5	5.2184e-3 (1.26e-2) ≈	1.8301e-3 (3.08e-3) ≈	1.0661e-3 (1.07e-3)
MW6	1.5704e-1 (1.96e-1) −	4.5257e-2 (1.03e-1) ≈	4.0993e-2 (1.02e-1)
MW7	4.5021e-3 (3.25e-4) ≈	4.4799e-3 (2.56e-4) ≈	4.2677e-3 (1.92e-4)
MW8	4.6551e-2 (4.74e-3) ≈	4.4926e-2 (3.43e-3) ≈	4.4982e-2 (2.96e-3)
MW9	9.7161e-3 (2.19e-3) ≈	1.0192e-2 (2.76e-3) −	8.8326e-3 (1.45e-3)
MW10	5.2996e-2 (1.04e-1) ≈	3.2347e-2 (2.41e-2) ≈	3.2011e-2 (2.62e-2)
MW11	6.0851e-3 (1.54e-4) ≈	6.0657e-3 (1.09e-4) ≈	6.0244e-3 (9.68e-5)
MW12	5.0942e-3 (2.01e-3) −	4.7546e-3 (1.38e-4) ≈	4.7839e-3 (9.39e-5)
MW13	1.3416e-1 (6.81e-2) −	1.0820e-1 (7.58e-2) −	9.3628e-2 (5.67e-2)
MW14	9.8003e-2 (2.31e-3) −	9.6678e-2 (1.37e-3) ≈	9.6088e-2 (1.59e-3)
+/ − / ≈	1/5/8	0/2/12	

than the other MOEAs. The performance was due to the fact that the DTAEA algorithm considers both feasible and infeasible solutions, and its two-stage evolutionary strategy accelerated the convergence speed very well, converging to the PF optimal surface across the infeasible region.

Furthermore, Figure 5 plots the feasible and nondominated solutions with median IGD value among 30 runs obtained by the seven CMOEAs on the DASCMOP7 test problem. The figure shows that the DASCMOP7 test problem is a discrete feasible region, and its feasible region is very small. In this test problem, the other six algorithms are not evenly distributed. C-TAEA, PPS, and TOP converge in the local feasibility region, and only a tiny part of the solution converged to the final PF. The DTAEA algorithm performs well on DASCMOP7. It converges more uniformly over multiple disconnected feasible regions in the second stage due to the priority of feasible solutions after rapidly crossing infeasible regions through the first stage.

Table II shows that c-DPEA [20], C-TAEA [24], CCMO [38], DD-CMOEA [31], PPS [25], TOP [32] and DTAEA performed 30 independent runs on the constraint DTLZ problem, MW problem, and DOC problem. According to the method suggested in [54] for sampling on the PF of the problem, about 10,000 reference points were used to calculate the HV value of each problem. The Table II shows that the DTAEA algorithm performed well on most of the test problems. There were 10 DTLZ test problems, 14 MW test problems, and 9 DOC test problems. On these 33 test problems, the DTAEA algorithm performed best on 18 problems; the c-DPEA achieved eight best results, the C-TAEA achieving four best results, CCMO and DD-CMOEA both achieving six best results. PPS and TOP was not able to obtain a better result on the 33 problems.

We compared the DTAEA algorithm with some of its variants on the MW test problem. The results are shown in Table III, where the first variant, DTAEAd, is an evolutionary method that only retains the first stage, dual-population weakly cooperative and complementary evolution. The second variant, DTAEAc, is a progressive strategy that retains the second stage of single population prioritization for feasibility. Since DTAEAd is the main evolution, it even outperforms the DTAEA on some test problems, while DTAEAc is the proposed progressive strategy that serves to make the population converge on the final constrained PF at the last stage of population evolution, and thus performs worse as a comparison algorithm alone, but the DTAEA algorithm by the combination of these two stages obtained better results on most test problems.

In summary, the effectiveness of the DTAEA's two-stage strategy is verified. Although it achieves general performance on the C-DTLZ and DC-DTLZ problems, which is caused by the insufficient ability of the second stage to cross the infeasible region, and for C1-DTLZ3 and DC3-DTLC3, they place an infeasible barrier in the reachable objective space, which prevents the population from converging to the true PF, however, the other test problems, DTAEA shows the competitiveness. The two-stage superiority of the DTAEA algorithm is demonstrated. In the high-dimensional case of the C-DTLZ test problem, our DTAEA algorithm obtains partially better performance, as shown in Table IV, which is slightly inferior to C-TAEA but still very competitive except for the C1-DTLZ3 test problem. Among the high-dimensional CMOPs, C-TAEA comes first due to its unique targeting to solve the C-DTLZ test

Table IV. The IGD value of c-DPEA,C-TAEA, CCMO, DD-CMOEA, PPS, TOP AND DTAEA on C-DTLZ test problem.

Problem	M	c-DPEA	C-TAEA	CCMO	DD-CMOEA	PPS	ToP	DTAEA
C1_DTLZ1	5	6.8854e-2 (6.75e-4) ≈	8.2664e-2 (1.18e-2) −	3.8129e-1 (7.49e-2) −	7.2923e-2 (8.87e-4) −	8.1450e-2 (1.85e-3) −	NaN (NaN)	6.8687e-2 (8.93e-4)
	8	1.2327e-1 (1.10e-3) +	1.2886e-1 (3.63e-3) −	4.3300e-1 (1.01e-1) −	1.3693e-1 (7.76e-3) −	1.3195e-1 (7.80e-3) −	NaN (NaN)	1.2398e-1 (1.33e-3)
C1_DTLZ3	5	1.2218e+0 (2.06e+0) +	1.0711e+1 (2.86e+0) ≈	6.9369e-1 (3.10e-1) +	9.2129e-1 (5.61e-1) +	2.0515e+0 (3.36e+0) +	7.5051e+0 (5.75e+0) ≈	8.9545e+0 (4.62e+0)
	8	1.4358e+2 (1.83e+1) ≈	1.1740e+1 (1.34e-2) +	1.5113e+2 (1.30e+1) −	1.5626e+2 (1.33e+1) −	3.5753e+0 (5.15e+0) +	2.1832e+1 (4.21e+0) +	1.3930e+2 (1.94e+1)
C2_DTLZ2	5	1.8033e-1 (2.67e-3) ≈	1.7990e-1 (2.24e-3) +	1.9001e-1 (5.12e-3) −	1.9020e-1 (4.22e-3) −	2.2615e-1 (1.02e-2) −	2.4208e-1 (7.44e-3) −	1.8143e-1 (2.01e-3)
	8	2.9959e-1 (7.16e-3) −	3.0514e-1 (3.77e-3) −	3.8410e-1 (1.31e-1) −	3.3663e-1 (1.05e-1) −	5.5350e-1 (7.44e-2) −	8.3004e-1 (1.84e-1) −	2.9299e-1 (7.17e-3)
C3_DTLZ4	5	4.5498e-1 (2.20e-2) −	3.6281e-1 (3.82e-3) +	4.0001e-1 (4.48e-2) +	4.2662e-1 (4.18e-2) +	5.4955e-1 (3.26e-2) −	5.3052e-1 (2.45e-2) −	4.3751e-1 (2.10e-2)
	8	2.1949e+0 (3.30e-2) ≈	5.8617e-1 (3.64e-3) +	2.1826e+0 (1.86e-2) ≈	2.2350e+0 (5.29e-2) −	8.5104e-1 (3.55e-2) +	1.2926e+0 (2.39e-1) +	2.1954e+0 (2.64e-2)
+/ − / ≈		2/2/4	4/3/1	2/5/1	2/6/0	3/5/0	2/3/1	

problem. Objectively, for high-dimensional CMOPs where the infeasible space division fragments the feasible region, the second-stage progressive strategy of the DTAEA algorithm leads to insufficient exploration power for spanning the infeasible region and therefore affects the convergence of the algorithm due to slow convergence.

The ability of the algorithm to traverse a fairly large infeasible region and converge to a small feasible region is a significant challenge for MW test problems with small feasible regions. In addition, many MW problems have discontinuous feasible regions or discontinuous PFs. It is critical for the algorithm to maintain diversity and convergence, and the DTAEA algorithm achieved significant advantages over existing algorithms for the MW problem. Particularly good performance has been achieved on the MW9, MW11 and MW12 problems with unconstrained PF and constrained PF separation.

The DOC testing problem considers both decision and objective variables, including equation constraints and inequality constraints. The PF of DOC has various properties such as continuous disconnectedness, convexity, concavity, linearity, mixture, degeneracy, and multimodality. Both the decision space and the objective space have many complex properties, which make it difficult for the CMOEA algorithm to find the final feasible solution set. However, due to the advantages of the two-stage strategy, the DTAEA algorithm performed significantly better than the other algorithms on the DOC problem.

5. Comparison in mechanical design applications

In this section, the performance of DTAEA is verified by comparing with six algorithms on real-world optimization problems, RWCMOPs test problem [55]. The mathematical definitions of these CMOPs can be found in their original literature.

5.1. Parameter settings

We selected five mechanical design applications from RWCMOPs. They are five different real-world constrained optimization test problems where M, D vary from 2 to 3, 2 to 7, respectively. M is the total number of objectives, D is the total number of decision variables of the problem. To be more specific, besides car side design with $M = 3$, others are $M = 2$. For the speed reducer design problem and car side design problem, $D = 7$. For the gear train design and four bar plane truss, $D = 4$, and $D = 2$ for two bar plane truss.

5.2. Experimental Results on real-world problems

Table V shows the HV values obtained for 30 independent runs of the seven CMOEAs on five real-world test problems. The best mean value of each problem is highlighted in boldface. It can be observed that the DTAEA algorithm significantly outperforms the other six algorithms on real-world test problems, especially on the speed reducer design test problem with the most inequality constraints. That is, the usefulness of the DTAEA is also demonstrated for real-world problems. DTAEA achieved the three best results. C-TAEA and TOP both achieved one best result.

6. Conclusion

This paper proposes a new two-stage evolutionary optimization algorithm to solve the problem that CMOEAs cannot balance convergence and diversity in complex CMOPs. The algorithm fully considers feasible and infeasible solutions in the first stage. The population can quickly traverse the infeasible region and effectively prevent the population from converging to the local feasible region. When the feasible solution or the difference of feasible solutions in the population reaches a specific value, it enters the second stage. In the second stage, use a progressive strategy helps the final populations converge to the constrained PF. The results show the effectiveness of the DTAEA compared

Table V. The HV value of c-DPEA,C-TAEA, CCMO, DD-CMOEA, PPS, TOP AND DTAEA on real-world test problems.

Problem	c-DPEA	C-TAEA	CCMO	DD-CMOEA	PPS	ToP	DTAEA
Speed Reducer Design	2.7693e-1 (5.24e-5)	2.4303e-1 (4.26e-2)	2.7700e-1 (4.15e-5)	2.7699e-1 (3.91e-5)	2.5820e-1 (4.46e-2)	2.7609e-1 (5.01e-4)	**2.7701e-1 (3.16e-5)**
Gear Train Design	4.8456e-1 (5.08e-5)	4.8391e-1 (3.02e-4)	4.8454e-1 (3.98e-5)	4.8458e-1 (4.44e-5)	4.8448e-1 (4.27e-5)	4.8444e-1 (5.42e-5)	**4.8462e-1 (4.54e-5)**
Car Side Design	2.5987e-2 (5.26e-5)	**2.6104e-2 (4.48e-5)**	2.6034e-2 (4.24e-5)	2.5927e-2 (6.12e-5)	2.4693e-2 (8.63e-4)	2.5734e-2 (9.50e-5)	2.5987e-2 (6.09e-5)
Four Bar Plane Truss	4.0945e-1 (1.70e-4)	4.0860e-1 (6.33e-4	4.0945e-1 (1.46e-4)	4.0960e-1 (1.38e-4)	3.8608e-1 (5.97e-3)	4.0935e-1 (1.07e-4)	**4.0962e-1 (1.26e-4)**
Two Bar Plane Truss	8.4104e-1 (2.72e-3)	8.4370e-1 (3.32e-3)	8.4117e-1 (2.11e-3)	8.4212e-1 (1.21e-3)	8.4700e-1 (1.91e-4)	**8.4735e-1 (1.66e-4)**	8.4188e-1 (1.11e-3)

to those algorithms of recent years. However, there are still shortcomings, such as whether there is a better time to move to the second stage if the distribution of conditions is too wide, which is an issue to consider in the future.

Acknowledgments

The authors wish to thank the support of the National Natural Science Foundation of China (Grant No. 61876164 and 62176228), the Natural Science Foundation of Hunan Province (Grant No. 2020JJ4590), the Education Department Major Project of Hunan Province (Grant No. 17A212), the MOEA Key Laboratory of Intelligent Computing and Information Processing, the Science and Technology Plan Project of Hunan Province (Grant No. 2016TP1020), the Provinces and Cities Joint Foundation Project (Grant No. 2017JJ4001), and the Hunan province science and technology project funds (2018TP1036).

References

[1] Jiahai Wang, Ying Zhou, Yong Wang, Jun Zhang, CL Philip Chen, and Zibin Zheng. Multiobjective vehicle routing problems with simultaneous delivery and pickup and time windows: formulation, instances, and algorithms. *IEEE transactions on cybernetics*, 46(3):582–594, 2015.

[2] Hossein Farzin, Mahmud Fotuhi-Firuzabad, and Moein Moeini-Aghtaie. A stochastic multi-objective framework for optimal scheduling of energy storage systems in microgrids. *IEEE Transactions on Smart Grid*, 8(1):117–127, 2016.

[3] Giuseppe Nicosia, Salvatore Rinaudo, and Eva Sciacca. An evolutionary algorithm-based approach to robust analog circuit design using constrained multi-objective optimization. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 7–20. Springer, 2007.

[4] Zhaopin Su, Guofu Zhang, Feng Yue, Dezhi Zhan, Miqing Li, Bin Li, and Xin Yao. Enhanced constraint handling for reliability-constrained multiobjective testing resource allocation. *IEEE Transactions on Evolutionary Computation*, 25(3):537–551, 2021.

[5] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[6] Yuan Liu, Ningbo Zhu, and Miqing Li. Solving many-objective optimization problems by a pareto-based evolutionary algorithm with preprocessing and a penalty mechanism. *IEEE Transactions on Cybernetics*, 2020.

[7] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. Spea2: Improving the strength pareto evolutionary algorithm. *TIK-report*, 103, 2001.

[8] Shengxiang Yang, Miqing Li, Xiaohui Liu, and Jinhua Zheng. A grid-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 17(5):721–736, 2013.

[9] Qingfu Zhang and Hui Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6):712–731, 2007.

[10] Ke Li, Kalyanmoy Deb, Qingfu Zhang, and Sam Kwong. An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Transactions on Evolutionary Computation*, 19(5):694–716, 2014.

[11] Yuan Liu, Yikun Hu, Ningbo Zhu, Kenli Li, Juan Zou, and Miqing Li. A decomposition-based multiobjective evolutionary algorithm with weights updated adaptively. *Information Sciences*, 572:343–377, 2021.

[12] Eckart Zitzler and Simon Künzli. Indicator-based selection in multiobjective search. In *International conference on parallel problem solving from nature*, pages 832–842. Springer, 2004.

[13] Ye Tian, Ran Cheng, Xingyi Zhang, Fan Cheng, and Yaochu Jin. An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility. *IEEE Transactions on Evolutionary Computation*, 22(4):609–622, 2017.

[14] Johannes Bader and Eckart Zitzler. Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary computation*, 19(1):45–76, 2011.

[15] Yue-Jiao Gong, Wei-Neng Chen, Zhi-Hui Zhan, Jun Zhang, Yun Li, Qingfu Zhang, and Jing-Jing Li. Distributed evolutionary algorithms and their models: A survey of the state-of-the-art. *Applied Soft Computing*, 34:286–300, 2015.

[16] Mitchell A Potter and Kenneth A De Jong. Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary computation*, 8(1):1–29, 2000.
[17] Yu-Hui Zhang, Yue-Jiao Gong, Tian-Long Gu, Hua-Qiang Yuan, Wei Zhang, Sam Kwong, and Jun Zhang. Decal: Decomposition-based coevolutionary algorithm for many-objective optimization. *IEEE transactions on cybernetics*, 49(1):27–41, 2017.
[18] Zhi-Zhong Liu, Yong Wang, and Bing-Chuan Wang. Indicator-based constrained multiobjective evolutionary algorithms. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(9):5414–5426, 2021.
[19] D. W. Coit, A. E. Smith, and D. M. Tate. Adaptive penalty methods for genetic optimization of constrained combinatorial problems. *INFORMS Journal on Computing*, 8, 1996.
[20] Mengjun Ming, Anupam Trivedi, Rui Wang, Dipti Srinivasan, and Tao Zhang. A dual-population based evolutionary algorithm for constrained multi-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2021.
[21] Kalyanmoy Deb and Rituparna Datta. A bi-objective constrained optimization algorithm using a hybrid evolutionary and penalty function approach. *Engineering Optimization*, 45(5):503–527, 2013.
[22] R. Datta, Mfp Costa, K. Deb, and A. Gaspar-Cunha. An evolutionary algorithm based pattern search approach for constrained optimization. In *2013 IEEE Congress on Evolutionary Computation*, 2013.
[23] Jing Liang, Xuanxuan Ban, Kunjie Yu, Boyang Qu, Kangjia Qiao, Caitong Yue, Ke Chen, and Kay Chen Tan. A survey on evolutionary constrained multi-objective optimization. *IEEE Transactions on Evolutionary Computation*, pages 1–1, 2022.
[24] Ke Li, Renzhi Chen, Guangtao Fu, and Xin Yao. Two-archive evolutionary algorithm for constrained multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 23(2):303–315, 2018.
[25] Zhun Fan, Wenji Li, Xinye Cai, Hui Li, Caimin Wei, Qingfu Zhang, Kalyanmoy Deb, and Erik Goodman. Push and pull search for solving constrained multi-objective optimization problems. *Swarm and evolutionary computation*, 44:665–679, 2019.
[26] Himanshu Jain and Kalyanmoy 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 Transactions on evolutionary computation*, 18(4):602–622, 2013.
[27] Zhun Fan, Wenji Li, Xinye Cai, Han Huang, Yi Fang, Yugen You, Jiajie Mo, Caimin Wei, and Erik Goodman. An improved epsilon constraint-handling method in moea/d for cmops with large infeasible regions. *Soft Computing*, 23(23):12491–12510, 2019.
[28] M. Asafuddoula, T. Ray, and R. Sarker. A decomposition-based evolutionary algorithm for many objective optimization. *IEEE Transactions on Evolutionary Computation*, 19(3):445–460, 2015.
[29] B Jza, B Rsa, D Sya, and C Jzab. A dual-population algorithm based on alternative evolution and degeneration for solving constrained multi-objective optimization problems. *Information Sciences*, 2021.
[30] Jinlong Zhou, Juan Zou, Jinhua Zheng, Shengxiang Yang, Dunwei Gong, and Tingrui Pei. An infeasible solutions diversity maintenance epsilon constraint handling method for evolutionary constrained multiobjective optimization. *Soft Computing*, pages 1–12, 2021.
[31] Mengjun Ming, Rui Wang, Hisao Ishibuchi, and Tao Zhang. A novel dual-stage dual-population evolutionary algorithm for constrained multi-objective optimization. *IEEE Transactions on Evolutionary Computation*, pages 1–1, 2021.
[32] Zhi-Zhong Liu and Yong Wang. Handling constrained multiobjective optimization problems with constraints in both the decision and objective spaces. *IEEE Transactions on Evolutionary Computation*, 23(5):870–884, 2019.
[33] Ye Tian, Yajie Zhang, Yansen Su, Xingyi Zhang, Kay Chen Tan, and Yaochu Jin. Balancing objective optimization and constraint satisfaction in constrained evolutionary multiobjective optimization. *IEEE Transactions on Cybernetics*, 2021.
[34] Kunjie Yu, Jing Liang, Boyang Qu, Yong Luo, and Caitong Yue. Dynamic selection preference-assisted constrained multiobjective differential evolution. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pages 1–12, 2021.
[35] Zhi-Zhong Liu, Bing-Chuan Wang, and Ke Tang. Handling constrained multiobjective optimization problems via bidirectional coevolution. *IEEE Transactions on Cybernetics*, pages 1–14, 2021.
[36] S. Zeng, R. Jiao, C. Li, L. Xi, and J. S. Alkasassbeh. A general framework of dynamic constrained multiobjective evolutionary algorithms for constrained optimization. *IEEE Trans Cybern*, 47(9):2678–2688, 2017.
[37] S. Zeng, R. Jiao, C. Li, and R. Wang. Constrained optimisation by solving equivalent dynamic loosely-constrained multiobjective optimisation problem. *International Journal of Bio-Inspired Computation*, 13(2):86–, 2019.
[38] Ye Tian, Tao Zhang, Jianhua Xiao, Xingyi Zhang, and Yaochu Jin. A coevolutionary framework for constrained multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 25(1):102–116, 2020.
[39] Kunjie Yu, Jing Liang, Boyang Qu, and Caitong Yue. Purpose-directed two-phase multiobjective differential evolution for constrained multiobjective optimization. *Swarm and Evolutionary Computation*, 60:100799, 2021.
[40] C. He, R. Cheng, Y. Tian, X. Zhang, and Y. Jin. Paired offspring generation for constrained large-scale multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, PP(99):1–1, 2020.
[41] Y. Tian, X. Zheng, X. Zhang, and Y. Jin. Efficient large-scale multi-objective optimization based on a competitive swarm optimizer. *IEEE Transactions on Cybernetics*, pages 1–13, 2019.
[42] Kalyanmoy Deb, Ram Bhushan Agrawal, et al. Simulated binary crossover for continuous search space. *Complex systems*, 9(2):115–148, 1995.
[43] Kalyanmoy Deb, Mayank Goyal, et al. A combined genetic adaptive search (geneas) for engineering design. *Computer Science and informatics*, 26:30–45, 1996.
[44] Hai-Lin Liu, Fangqing Gu, and Qingfu Zhang. Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. *IEEE transactions on evolutionary computation*, 18(3):450–455, 2013.
[45] Md Asafuddoula, Tapabrata Ray, and Ruhul Sarker. A decomposition-based evolutionary algorithm for many objective optimization. *IEEE Transactions on Evolutionary Computation*, 19(3):445–460, 2014.
[46] Zhongwei Ma and Yong Wang. Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons. *IEEE Transactions on Evolutionary Computation*, 23(6):972–986, 2019.

[47] Zhun Fan, Wenji Li, Xinye Cai, Hui Li, Caimin Wei, Qingfu Zhang, Kalyanmoy Deb, and Erik Goodman. Difficulty adjustable and scalable constrained multiobjective test problem toolkit. *Evolutionary Computation*, 28(3):339–378, 2020.

[48] Ye Tian, Ran Cheng, Xingyi Zhang, and Yaochu Jin. Platemo: A matlab platform for evolutionary multi-objective optimization [educational forum]. *IEEE Computational Intelligence Magazine*, 12(4):73–87, 2017.

[49] Kenneth Price, Rainer M Storn, and Jouni A Lampinen. *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media, 2006.

[50] R. B. Agrawal, K. Deb, and R. B. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9(3):115–148, 1994.

[51] Peter AN Bosman and Dirk Thierens. The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE transactions on evolutionary computation*, 7(2):174–188, 2003.

[52] Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4):257–271, 1999.

[53] Ye Tian, Xiaoshu Xiang, Xingyi Zhang, Ran Cheng, and Yaochu Jin. Sampling reference points on the pareto fronts of benchmark multi-objective optimization problems. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–6, 2018.

[54] Miqing Li, Shengxiang Yang, and Xiaohui Liu. Pareto or non-pareto: Bi-criterion evolution in multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 20(5):645–665, 2015.

[55] Abhishek Kumar, Guohua Wu, Mostafa Z. Ali, Qizhang Luo, Rammohan Mallipeddi, Ponnuthurai Nagaratnam Suganthan, and Swagatam Das. A benchmark-suite of real-world constrained multi-objective optimization problems and some baseline results. *Swarm and Evolutionary Computation*, 67:100961, 2021.