A Multi-stage Algorithm for Solving Multi-objective Optimization Problems with Multi-constraints

Ruiqing Sun [1], ✉Juan Zou [1], *Member, IEEE,* Yuan Liu, Shengxiang Yang, *Senior Member, IEEE,* and Jinhua Zheng, *Member, IEEE,*

***Abstract*—There are usually multiple constraints in constrained multi-objective optimization. Those constraints reduce the feasible area of the constrained multi-objective optimization problems (CMOPs) and make it difficult for current multi-objective optimization algorithms (CMOEAs) to obtain satisfactory feasible solutions. In order to solve this problem, this paper studies the relationship between constraints, then obtains the priority between constraints according to the relationship between the Pareto Front (PF) of the single constraint and their common PF. Meanwhile, this paper proposes a multi-stage CMOEA and applies this priority, which can save computing resources while helping the algorithm converge. The proposed algorithm completely abandons the feasibility in the early stage to better explore the objective space, and obtains the priority of constraints according to the relationship; Then the algorithm evaluates a single constraint in the medium stage to further explore the objective space according to this priority, and abandons the evaluation of some less-important constraints according to the relationship to save the evaluation times; At the end stage of the algorithm, the feasibility will be fully considered to improve the quality of the solutions obtained in the first two stages, and finally get the solutions with good convergence, feasibility, and diversity. The results on five CMOP suites and three real-world CMOPs show that the algorithm proposed in this paper can have strong competitiveness in existing constrained multi-objective optimization.**

***Index Terms*—Constrained multi-objective optimization, Evolutionary algorithm, Constraint-handling priority.**

I. INTRODUCTION

Practical optimization problems in the real world include multiple conflicting objective functions that need to be optimized simultaneously [1]. Such optimization problems are usually called multi-objective optimization problems (MOPs) [2]. Most real-world multi-objective optimization problems contain one or more constraints on their objective values or decision variables. Such problems are called constrained multi-objective optimization problems (CMOPs), such as pressure vessel design [3], vibrating platform design [3], crash energy management for high-speed train [4], and power distribution system planning [5]. A CMOP can be defined as follows [6]:

$$\begin{cases} min\ F(X) = (f_1(X), f_2(X), \dots, f_m(X)), \\ subject\ to\ X \in \Omega, \\ \qquad g_i(X) \le 0, i = 1, \dots, p \\ \qquad h_j(X) = 0, j = 1, \dots, q \end{cases} \tag{1}$$

where $X = (x_1, x_2, \dots, x_n)$ is an n-dimensional decision variable vector from the decision space Ω; $F(X)$ is an objective function vector composed of m conflicting objective functions; $g_i(X)$ are q inequality constraints and $h_i(X)$ are p equality constraints.

For any two solutions $X_a, X_b \in \Omega$, the solution X_a Pareto dominates X_b if and only if $f_i(X_a) \le f_i(X_b)$ for each $i \in \{1, \dots, m\}$ and $f_j(X_a) < f_j(X_b)$ for at least one $j \in \{1, \dots, m\}$, denoted as $X_a \prec X_b$ [7]. If there are no mutually dominating solutions in a set, then the set is called Pareto optimal set [8]. The mapping of all Pareto optimal sets in the objective space called unconstrained Pareto Front (PF), the mapping of all Pareto optimal sets that satisfy the constraints in the objective space is called constrained PF.

The solution of each CMOP has the degree of violation of each constraint. For the degree of violation of a solutuion, $c_j(X)$ can be used to define the j-th constraint:

$$c_j(X) = \begin{cases} max(0, g_j(X)), & j = 1, \dots, p \\ max(0, |h_j(X) - \delta|), & j = p+1, \dots, p+q \end{cases} \tag{2}$$

where δ is a small enough relaxation factor (usually taken as 1e-6) to relax equality constraints [9]. The solution of a CMOP usually adds the violation values of each constraint by $CV(X)$ to determine the overall violation of all constraints [10].

$$CV(X) = \sum_{j=1}^{p+q} c_j(X), \tag{3}$$

Unlike MOPs, solving a CMOP considers not only convergence and diversity but also feasibility. The challenges brought by constraints to CMOP is a major difficulty in constrained multi-objective optimization. Generally speaking, the difficulty of solving a CMOP is positively related to the number constraints. The intersection of the feasible regions mapped by multiple constraints in the objective space forms

1 These authors have contributed equally to this work.

Ruiqing Sun, Juan Zou (Corresponding author, email: zoujuan@xtu.edu.cn), Yuan Liu, and Jinhua Zheng are with Hunan Engineering Research Center of Intelligent System Optimization and Security, Key Laboratory of Intelligent Computing and Information Processing, Ministry of Education of China, and Key Laboratory of Hunan Province for Internet of Things and Information Security, Xiangtan University, Xiangtan, 411105, Hunan Province, China.

Shengxiang Yang is with the School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, Leicester, U.K.

This work was supported in part by the National Natural Science Foundation of China (Grant No. 62276224, Grant No. 62176228 and Grant No. 61876164), in part by the Research Foundation of Education Bureau of Hunan Province, China (Grant No. 21A0444), in part by the Natural Science Foundation of Hunan Province, China (Grant No. 2022JJ40452) in part by the Natural Science Foundation of Hunan Province, China (Grant No. 2020JJ4590), and in part by the General Project of Hunan Education Department (Grant No. 21C0077)

Source code available at:
https://github.com/KFC-Grandpa/MOEA/blob/main/C3M.zip

the final feasible region. In the same CMOP, the feasible region mapped by a single constraint in the objective space is greater than or equal to the feasible region shared by this constraint with other constraints, and the former is easier to find the feasible solution than the latter. In addition, we also find that some constraints are more critical than others in CMOPs according to the relationship between the PF of constraints. Based on this, this paper proposes a multi-stage constrained multi-objective optimization evolutionary algorithm (CMOEA) to solve CMOPs. The CMOEA proposed in this paper will evaluate the impact of constraints on MOP in an easy to difficult way, and in order to obtain the solution as efficiently as possible, some less-important constraints will not be evaluated. The main contributions of this paper are as follows:

1) The relationship between constraints is discussed. According to this relationship, a constraint-handling priority and a method to determine the less-important constraint(s) are proposed.
2) A multi-stage CMOEA called C3M (Constraint, Multi-objective, Multi-stage, Multi-constraints) is proposed. The proposed algorithm evaluates MOP itself for exploration in the first stage. In the second stage, the algorithm will evaluate the impact of a single constraint on MOP to reduce the overall difficulty of CMOP. In this stage, some less-important constraint(s) will not be evaluated to save evaluation times. In the last stage, the algorithm will evaluate the impact of all constraints on MOP to obtain a final feasible solution with good diversity and convergence.

The rest of this article is organized as follows: in Section II, we review the existing constraint handling technologies and explain the motivation of this paper. In Section III, we introduce the C3M algorithm proposed in this paper and its details. Section IV analyzes the performance of our C3M in the CMOP test suites and real-world CMOPs and compares the existing multi-stage CMOEAs. Section V summarizes this paper.

II. Related Works And Motivation

A. Constraint Handling Techniques

After about 20 years of development, many constraint handling techniques (CHTs) have been proposed [11].

The feasibility based CHT is the most widely used. The most commonly used is Constraint Domination Principle (CDP), which is defined as follows: if $X_1 \prec X_2$, it is satisfied with:

$$\begin{cases} CV(X_1)=0, CV(X_2)>0 \\ CV(X_1)=0, CV(X_2)=0, and\ X_1 \prec X_2 \\ CV(X_1)>0, CV(X_2)>0, and\ CV(X_1)<CV(X_2) \end{cases} \tag{4}$$

The CDP was first proposed by Deb et al. in [12], It is simple, does not need any parameters, and can be easily combined with other CMOEAs. Though this method is simple and relatively effective, it ignores the other two elements to solve CMOP - convergence and diversity. Finally, it leads the population fall into local optimization when facing complex CMOPs.

In order to overcome this, researchers try to find a balance among these three elements. The methods adopted by researchers can be roughly divided into four categories: The first is the ϵ based method [13] [14]. The solution with a constraint violation value less than epsilon is regarded as a fake feasible solution. Fan et al. improved the ϵ method in [15], which will help the population overcome the large infeasible region. The second category is the self-adaptive penalty function method, which brings the constraint violation value into the objective function. The key of this method is the setting of penalty factors, which are often related to CMOPs [16]. Therefore, researchers have designed a series of adaptive penalty factors [17]. The multi-objective optimization-based method is the third category, which optimizes the constraints as a new objective function. A representative method is [18] and [19]. The last category is the stochastic ranking method, which was first proposed by Yao et al. [20]. This method selects individuals with better objective values with a larger probability, and selects individuals with better constraints violation with a smaller probability. However, not all solutions have a dominant relationship. Researchers have improved this problem in many ways [21] [22]. These four categories have the problems of parameter setting and increasing complexity, although they are improved compared with CDP, they still have difficulties in facing complex CMOPs.

To better balance feasibility, convergence, and diversity, the most cutting-edge method is to use the hybrid method. Researchers achieve it in two ways: the first way is to adopt different strategies in different populations. In CCMO [11], Tian et al. used a completely feasibility-based strategy in $Pop1$ to ensure feasibility while using a completely non-feasibility-based strategy in $Pop2$ to ensure convergence and diversity. This method has advantages in dealing with problems with relatively close constrained PF and unconstrained PF, but it is less helpful when they are far. Therefore, Zou et al. [23] and Ming et al. [24] have proposed improvements to face the situation. Li et al. also used two populations in CTAEA [25], in which the convergence-oriented archive CA aims to optimize constraints and objectives and push the population to the Pareto front, while the diversity-oriented archive DA explores underutilized by CA, including infeasible areas. Since DA prefers solutions with better diversity rather than infeasible solutions with better objective value, it is difficult to cross the infeasible area. The second way is to adopt different strategies in different stages. Tian et al. proposed a two-stage CMOEA in CMOEA_MS [26], when the proportion of feasible solutions is less than λ, the population will give the same priority to constraints and objective values; on the contrary, the population gives priority to constraints. The value of lambda will affect the performance, the fixed lambda value is difficult to face different CMOPs. In the ToP [27], only a single objective and all constraints are considered in the first stage; then, in the second stage, all constraints and objectives are considered to obtain the final solution. A single reference vector will cause the gathering of solutions at the end of stage 1, which is not conducive to exploring the feasible region in stage 2. Fan et al. proposed the push and pull search (PPS) [28] based on MOEA/D [29] framework, which helps the population explore the objective space by using different ϵ value in different stages. This method focuses on the information in the infeasible solution, thus ignoring in feasible region. Fei et al. proposed a similar method in [30], which uses the information in the infeasible region to improve the final quality of solutions. Zhu et al. [31] also used the ϵ method to help the population escape from the local optimization caused by constraints, which is essentially the utilization of information in infeasible reigon. Ma et al. proposed a constraint-handling priority different from this paper and applied to MSCMO [32]. In MSCMO, constraints are added one after the other and handled in different stages of evolution. This method may retain some infeasible solutions when the number of evaluations is insufficient .

B. Motivation

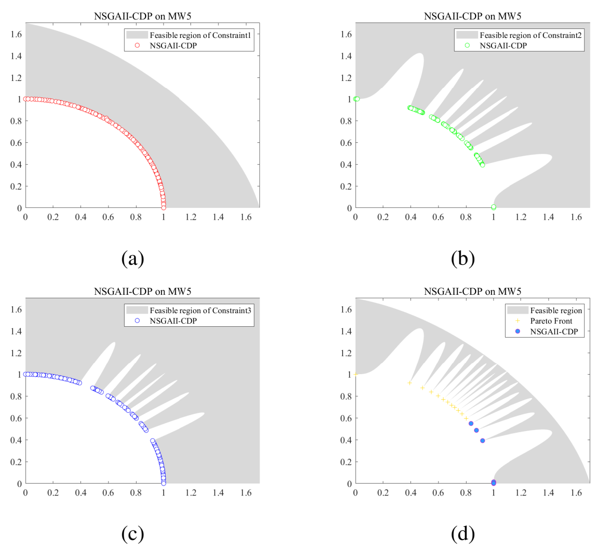


Fig. 1. (a)-(c): Solutions obtained by NSGAII-CDP on constraint1-3 constraint on MW5 respectively. (d) Solutions obtained by NSGAII-CDP on all constraints on MW5.

Real-world CMOPs usually contain multiple constraints. Generally speaking, whether a CMOP is easy to solve depends mainly on the number and difficulty of constraints. On the premise that the difficulty of each constraint is the same, the CMOPs with more constraints are more challenging to solve, while CMOPs with only one constraint is relatively easy to solve. The reason is that a single constraint is easier to meet than multiple constraints.

For example, MW5 [33] is a CMOP with three constraints. Figure 1(a) - 1(c) shows the solutions obtained by NSGAII-CDP only handle a single constraint, and figure 1(d) shows the results obtained by NSGAII-CDP handled all constraints. We can see that the feasible region of MW5 is narrow since the existence of multiple constraints, while a single constraint is easy to satisfy. As a result, all optimal results are obtained while NSGAII-CDP handles a single constraint, but only partial optimal solutions are obtained when handling all constraints.

Therefore, We can handle a single constraint to reduce the difficulty when exploring. However, do we need to handle all constraints? We might as well discuss the relationship between constraints. Suppose constraintA and constraintB are two different constraints in the same CMOP. From the PF between constraints, there are three types.

1) **Type A:** The PF of constraintA partially (all) satisfies constraintB, and the PF of constraintB partially (all) satisfies constraintA. The final PF consists of both. In this case, we can consider that the two constraints have the same priority. For example, figure 2(a) shows type A relationship.
2) **Type B:** The PF of constraintA partially (all) satisfies constraintB. In contrast, the PF of constraintB completely does not satisfy constraintA, and its final PF is all composed of the optimal solution set of constraintA. In this case, only considering constraintA can obtain partially (all) final feasible Pareto optimal solutions. Therefore, in this case, constraintA is more critical than constraintB. For example, figure 2(b) shows type B relationship.
3) **Type C:** The PF of constraintA does not satisfy constraintB, the PF of constraintB does not satisfy constraintA, and the final PF is not composed of the Pareto optimal solution sets of constraintA and constraintB. In this case, the priority of constraintA and constraintB

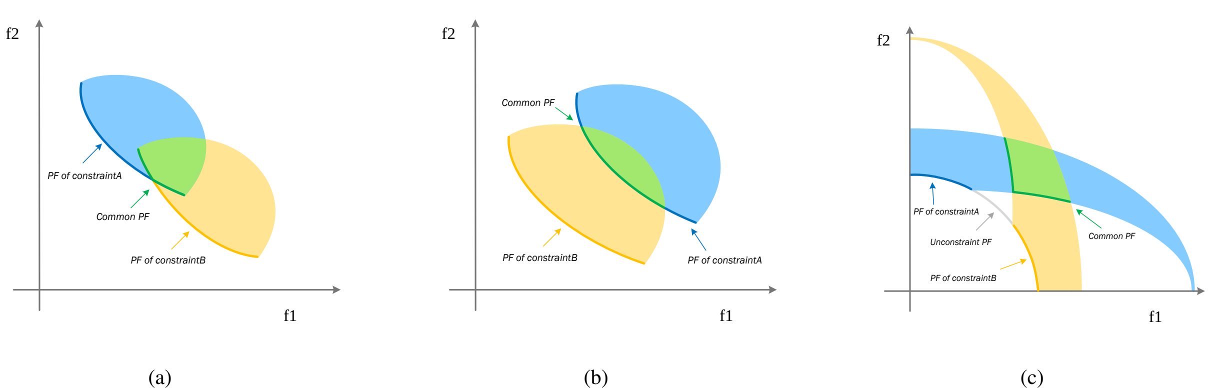


Fig. 2. Three relationships between constraints. (a) Type A. (b) Type B. (c) Type C. The blue region represents the feasible region of constraintA; The Yellow region represents the feasible region of constraintB; The green region represents the common feasible region of constraintA and constraintB.

is the same. For example, figure 2(c) shows type C relationship.

Based on the above discussion, we use the following strategies to deal with CMOPs:

1) First, we explore objective space in a simple-to-complex way. The feasibility is completely abandoned in the preliminary exploration, and only one constraint is evaluated at a time in further exploration.
2) Second, some constraints will not be evaluated to save the evaluation times according to the relationship between constraints.

In this way, CMOPs with multiple constraints can be solved more easily. The following section will introduce the flow and details of the C3M proposed in this paper.

III. PROPOSED ALGORITHM

A. Procedure of C3M

Algorithm 1: Procedure of the proposed C3M

```
Input: N (population size), β (latest time to stage 3)
Output: Popualtion (final popualtion)
1  Now = 0;
2  arch ← An external archive set;
3  P ← RandomInitialization;
4  totalcon ← Number of constraints in CMOP;
5  for i ← 1to totalcon do
6      Pop[i] = P;
7      E[i] = 0;
8  while termination criterion not fulfilled do
9      P′ ← Select N parents from P ;
10     O ← Generate N offsprings basic on P′;
11     Obj_k ← Sum of absolute values of objective values of k
         generation in P;
12     if Now == 0 then
13         P ← Environmental_Selection(P ∪ O, No_Con);
14         Pop[i] ← Environmental_Selection(Pop[i] ∪ O, i);
15         arch ← Update external archive;
16         if is_stable(Obj,P,N,M) then
17             P ← RandomInitialization;
18             S ← Get priority according to Algorithm 3;
19             Now = S[1];
20     else if 0 <Now ≤ totalcon then
21         P ← Environmental_Selection(P ∪ O, Now);
22         Pop[i] ← Environmental_Selection(Pop[i] ∪ O, i);
23         arch ← Update external archive;
24         if is_stable(Obj,P,N,M) then
25             E[Now] = 1;
26             E ← Update less-important constraints according to
                 Algorithm 4;
27             if Have constraint(s) need to be evaluated then
28                 P ← RandomInitialization;
29                 Now ← Next unevaluated constraint in S;
30             else
31                 P ← Environmental_Selection(P ∪ arch ∪
                     Pop, All_Con);
32                 Now = totalcon + 1
33     else
34         P ← Environmental_Selection(P ∪ O, All_Con);
35     if evaluated ≥ evaluation * β then
36         Now = totalcon +1;
37 return P;
```

The general flow of the proposed C3M is presented in Figure 3 and its pseudo-code is shown in Algorithm 1. For ease of reading, we will introduce different parts of Algorithm 1 in different paragraphs.

Initialization: Lines 1-7 are the initialization of the algorithm. Firstly, the algorithm will initialize the population randomly and maintain a population array Pop for each constraint. In order to make it easier for readers to understand, Pop in the following refers to the entire population array, while the ith element in the Pop contains N individuals (i.e., Pop[i] for ith constraint). These N individuals are the optimal individuals obtained by environmental selection considering only ith constraint. Pop only exists in the first two stages and will not participate in reproduction and update themselves only through environmental selection. Pop exists to determine the priority of constraint and less-important constraint(s). The purpose of the E array is whether it is necessary to evaluate the ith constraint. If E[i] is positive, the algorithm will know that the ith constraint is unnecessary to process. Now represents the processing constraint of the current evaluation. If Now is equal to 0, the algorithm is in stage 1, and all the constraints are not considered during evaluation; When Now is between 1-$totalcon$, the algorithm will process the Nowth constraint. At this time, the algorithm is in stage 2. When Now equals $totalcon$+1, the algorithm is in stage 3, and all constraints are considered in evaluating the population.

Reproduction: Lines 9-11 show the operation of population reproduction. The mating selection used in this paper is binary tournament selection, and the offspring generation method used in this paper is shown in section IV-B. In line 19, the sum of the absolute value of objective values of k-th generation of the population is stored into Obj_k to judge whether the stage will convert.

Stage1: Lines 12-19 describe the operation in stage 1. In this stage, the P using environmental selection without considering any constraints updates Pop[i] through environmental selection only considering ith constraint and updating the external archive. The environment selection used in this paper is similar to that used in NSGAII-CDP. They differ in how $CV(X)$ is calculated: When constraints are not considered, $CV(X)$ is always equal to 0; When the ith constraint is considered, the $c_i(X)$ is equals to $CV(X)$; When all constraints are considered, $CV(X)$ is calculated by equation 3. In lines 16-19, the Algorithm 2 judges whether the algorithm can enter stage 2. If the stage transition conditions are met. Then the algorithm will reinitialize the P, decide the priority of the constraints, put it into S, and set Now as the first element in S. Then the algorithm will enter stage 2.

Stage2: Lines 20-32 describe the operations in stage 2. In this stage, the P only considers the Nowth constraint when making environment selection, updates Pop[i] through environmental selection only considering ith constraint and update external archive. In lines 24-32, the Algorithm 2 judges whether the algorithm can enter the next stage (processing the following constraint or entering stage 3). The algorithm first updates E and then judges whether there have constraint(s) to evaluate according to E. If there are still constraint(s) that have not been evaluated, the algorithm will reinitialize the P,

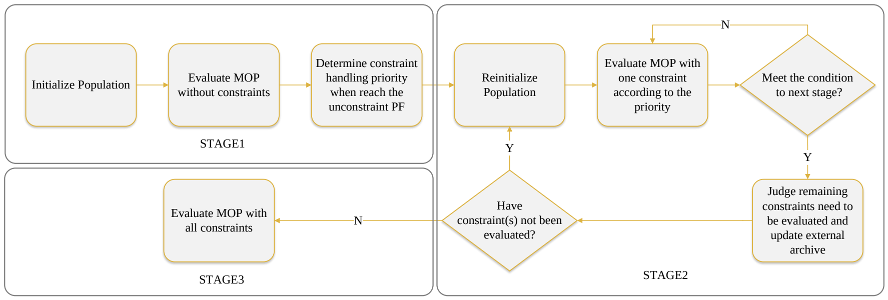


Fig. 3. Flowchart of the proposed C3M

then continue to evaluate the remaining constraints according to the priority in S; otherwise, it will enter stage 3.

Stage3: Line 34 are the operations in stage 3. At this time, the C3M considers all constraints. It is worth mentioning that to ensure that the population can enter stage 3 and obtain the final feasible solution since stage 1 & 2 is only an exploration of CMOPs, lines 35 and 36 specify the latest time to enter stage 3, β is the proportion of entering stage 3 at the latest (β = 0.7 in this paper,). In the following subsection, we will analyze and introduce more detail on C3M.

B. Analysis of proposed C3M

In this section, we show the detail of proposed C3M and how it works.

Algorithm 2: is_stable(Obj,P,N,M)

Input: Obj (sum of objective values of each generation), P (population), N (number of Population), M (number of objectives)
Output: $result$ (whether the population meets the conditions for entering the next stage)

```
result = 0;
FrontNo ← NDSort(P);
NC ← number of nondominant individual;
if NC == N then
    λ ← accroding to 5;
    if |Obj_k - Obj_{k-1}| ≤ λ then
        result = 1
return result;
```

1) Stage transition conditions: The C3M proposed in this paper contains multiple stages, including three major stages and many minor stages in stage 2, but their conversion conditions are the same. As shown in algorithm 2, when all the solutions in P are nondominant solutions, and the absolute value of the objective value of the two generations of population obj_k and obj_{k-1} is less or equal to λ, the population can enter the next stage.

$$\lambda = 10^{M-5}\frac{obj_k}{MN} \tag{5}$$

As shown in 5, the λ used in this paper is self-adaptive. It will adjust itself according to the population size N, problem scale, and the objective number M of the CMOP, $obj_k/(MN)$ is the average objective value of all individuals in the current population in each dimension, 10^{m-5} represents the magnification of this average value. When the dimension of the problem rises, this magnification will also increase. Since in an M-objective problem, the set of optimal solutions is in an $(M-1)$-dimensional hyperplane. However, the solution is always a point in the M-objective problem. When M is larger, the more possible positions of this solution in the $(M-1)$-dimensional hyperplane will be. The fixed λ may make it challenging to meet the conditions since the scales, and M between real-world CMOPs are usually huge.

In stage 1, the population will be in a stable state after reaching the unconstrained PF, while in stage 2, the population will be in a stable state after reaching a feasible region boundary of single constraint, which means the P does not need additional optimization, and the P has the conditions to enter the next stage.

Algorithm 3: Get_priority(Pop,$totalcon$)

Input: Pop (population of each constraint), $totalcon$ (number of constarints)
Output: S (The priority of processing constraints)

```
AllPop ← All individuals in Pop;
FrontNo ← NDSort(AllPop);
for i ← 1 to totalcon do
    min[i] ← The minimum dominance level of
      individuals in Pop[i] in FrontNo
S ← The index after the elements in min are arranged
  in descending order.
return S;
```

2) Priority of constraints: Deciding the priority of constraint is a key component of the C3M. The constraint priority decides through the Pop[i] maintained in stage 1, as shown in algorithm 3. The algorithm will get all nondominated levels of all solutions in Pop, obtain the best non-dominated level for each Pop[i], arrange these individuals in descending order, and output the index as the constraint priority.

As shown in the left side of figure 4, LIRCMOP7 comprises three constraints. After the nondominated sorting of all the populations in Pop[1], Pop[2] and Pop[3], the minimum non-dominated level of all solutions in Pop[2] and Pop[3] are 1, and the minimum non-dominated level in Pop[1] population is greater than 1. Therefore, 1st constraint is more important than 2nd constraint and 3rd constraint. The final priority is Constraint 1 >Constraint 2 = Constraint 3 The reason for this is straightforward: for a feasible solution, it must meet all constraints, so it must also meet the constraints farthest from the ideal point. In this example, for LIRCMOP7, there is no better solution than Pop[1]. In the region between the ideal point and the Pop[1], there is no feasible solution for the 1st constraint, but the final feasible solution must meet the 1st constraint, so the 1st constraint should be evaluated first.

Algorithm 4: Unnec(Pop,E,$totalcon$,Now)

Input: Pop (population of each constraint), E (constraints need to be processed), $totalcon$ (number of constarints), Now (the constraint consider in this stage)

Output: E (constraints need to be processed)

$AllPop \leftarrow$ All individuals in Pop[i];
$FrontNo \leftarrow$ NDSort($AllPop$);
$Min\_domin \leftarrow$ The minimum dominance level of individuals in $Pop[Now]$ in $FrontNo$;
for $i \leftarrow 1$ **to** *totalcon* **do**
  if i != Now **then**
    $Max\_domin \leftarrow$ The maximum dominance level of individuals in Pop[i] in $FrontNo$;
    if $Max\_domin \leq Min\_domin$ **then**
      E[i] = 1;
return E;

3) Deciding the unnecessary constraints: When algorithm 1 goes through stage 2, whenever a constraint is processed, it will determine whether the remaining constraints need to be evaluated during stage transition to save unnecessary evaluation times. The deciding of unnecessary constraints through the Pop[i] maintained stage 2. Its pseudo-code is shown in algorithm 4. Algorithm 4 first takes all the solutions in Pop[i] for non-dominated sorting and finds the smallest non-dominated level in the $Pop[Now]$ (i.e., the level with the best convergence in $Pop[Now]$), and find the level with the largest non-dominated level in the Pop[i] to which other constraints belong (i.e., the level with the worst convergence in Pop[i]). If the worst level in Pop[i] is better than the best individuals in the $Pop[Now]$, then according to Section II-B, We can think that it is in the Type B relationship of constraints, so the constraint with better PF does not need to be evaluated.

As shown on the right side of Figure 4, MW12 is a CMOP with two constraints. According to constraint-handling priority, the first constraint is more priority than the second constraint. Figure 4(b) shows the $Pop[1]$ when the first constraint is evaluated and the stage transition conditions are met, and figure 4(d) shows the $Pop[2]$. At this time, the $Pop[1]$ is dominated by $Pop[2]$ and meets the relationship of Type B. Therefore, the second constraint is not necessary to be evaluated. Besides, we can also see from figure 4(f) that the existence of the second constraint makes the final feasible region narrow and difficult to obtain the feasible solution, and our strategy considering only a single constraint will not be affected by this, which proves the superiority of our motivation.

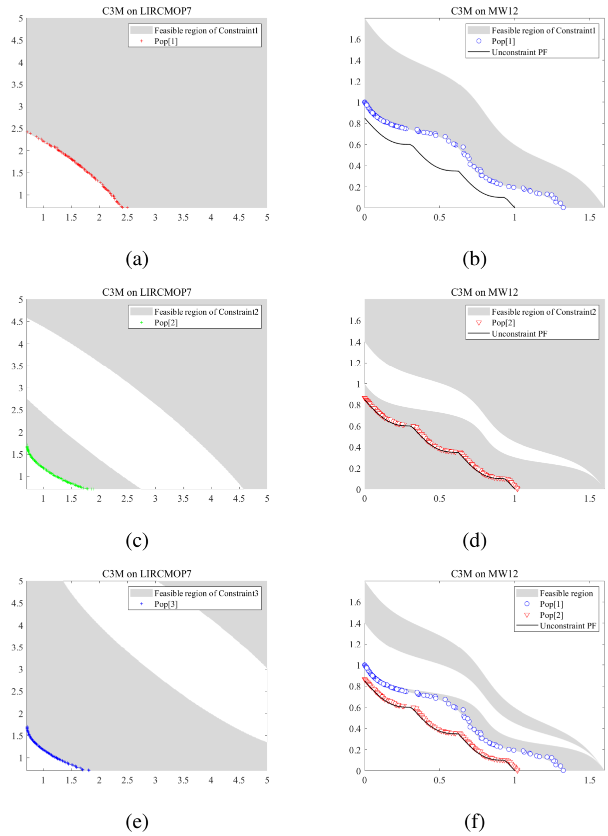


Fig. 4. Left side: Feasible region of each constraint in LIRCMOP7 and its Pop[i] after stage1. (a) First constraint of LIRCMOP7 (c) Second constraint of LIRCMOP7 (e) Thrid constraint of LIRCMOP7.
Right side: Feasible region of each constraint in MW12 and its Pop[i] after evaluating the first constraint, and its final feasible region. (b) First constraint of MW12 (d) Second constraint of MW12 (f) Final feasible region of MW12.

4) Reinitialize: Reinitialization is also a crucial part of C3M. It is mainly used at the beginning of stage 2 and each substage in stage 2. In the type C relationship, the final PF of the two constraints is not related to their respective PF but in the middle of the feasible region of the two constraints. Therefore, reinitialization can provide feasible solutions for the population when it passes through the middle of these crossed feasible regions. Reinitialization is also helpful for single constraint CMOPs, which can help the population obtain better diversity solutions and jump out of the local optimum.

5) Computational Complexity of C3M: Suppose N is the number of populations, C is the number of constraints, D is the number of decision vectors, and M is the objective numbers. In this paper, we use the Pareto-based reproduction method, in which the complexity of mating selection is $O(N)$, the complexity of genetic operators is $O(ND)$, and the complexity of environment selection is $O(MN^2)$. Although C3M is divided into three stages, C3M is the same as the applied reproduction method's mating selection and genetic operators. Therefore, C3M is the same complex as the reproduction methods applied to C3M in these two operations. As the environmental selection, C3M will make

$C + 2$ environmental selections in each generation in the first stage, and in the second stage, each generation will make less than $C + 2$ environmental selections. Note that this value gradually decreases with the optimization process. In the final stage, each generation will make one environmental selection as the selected reproduction method. Therefore, the complexity of the final environmental choice is $O(CMN^2)$. Therefore, the complexity of the final environmental choice is $O(CMN^2)$. Therefore, the algorithm proposed in this paper is slightly worse than the applied reproduction method, which mainly depends on the number of constraints C.

IV. Experimental Studies

All the experiments in this paper are based on PlatEMO2.9 [34]. The default parameters in the platform are used where there is no particular explanation.

A. Benchmark Suites and real-world CMOPs

We used CDTLZ [35] test suite, MW [33] test suite, DASCMOP [36] test suite, DOC [27] test suite and LIRCMOP [15] test suite to test C3M. Their Populaiton size N, objective number M, decision vector D and evaluation times FEs are shown in table I.

TABLE I
Parameter settings for MW, LIRCMOP, DOC, DASCMOP and CDTLZ.

Problem	N	M	D	FEs	Problem	N	M	D	FEs
MW1	100	2	15	200000	LIRCMOP1	100	2	10	300000
MW2	100	2	15	200000	LIRCMOP2	100	2	10	300000
MW3	100	2	15	200000	LIRCMOP3	100	2	10	300000
MW4	100	3	15	200000	LIRCMOP4	100	2	10	300000
MW5	100	2	15	200000	LIRCMOP5	100	2	10	300000
MW6	100	2	15	200000	LIRCMOP6	100	2	10	300000
MW7	100	2	15	200000	LIRCMOP7	100	2	10	300000
MW8	100	3	15	200000	LIRCMOP8	100	2	10	300000
MW9	100	2	15	200000	LIRCMOP9	100	2	10	300000
MW10	100	2	15	200000	LIRCMOP10	100	2	10	300000
MW11	100	2	15	200000	LIRCMOP11	100	2	10	300000
MW12	100	2	15	200000	LIRCMOP12	100	2	10	300000
MW13	100	2	15	200000	LIRCMOP13	100	3	10	300000
MW14	100	3	15	200000	LIRCMOP14	100	3	10	300000
DOC1	100	2	6	300000	DASCMOP1	100	2	15	300000
DOC2	100	2	16	300000	DASCMOP2	100	2	15	300000
DOC3	100	2	10	300000	DASCMOP3	100	2	15	300000
DOC4	100	2	8	300000	DASCMOP4	100	2	15	300000
DOC5	100	2	8	300000	DASCMOP5	100	2	15	300000
DOC6	100	2	11	300000	DASCMOP6	100	2	15	300000
DOC7	100	2	11	300000	DASCMOP7	100	3	15	300000
DOC8	100	3	10	300000	DASCMOP8	100	3	15	300000
DOC9	100	3	11	300000	DASCMOP9	100	3	15	300000
C1-DTLZ1	100	3	7	200000	DC1-DTLZ3	100	3	12	200000
C1-DTLZ3	100	3	12	200000	DC2-DTLZ1	100	3	7	200000
C2-DTLZ2	100	3	12	200000	DC2-DTLZ3	100	3	12	200000
C3-DTLZ4	100	3	12	200000	DC3-DTLZ1	100	3	7	200000
DC1-DTLZ1	100	3	7	200000	DC3-DTLZ3	100	3	12	200000

For real-world CMOPs, we use three real-world CMOPs to test C3M and comparison algorithm. They are bulk carrier design [37], pressure vessel design [3], and reactor network design [38]; their parameter settings are shown in Table II. The bulk carrier design problem is a CMOP with nine inequality constraints. It aims to design a bulk carrier with the smallest transportation consumption, the lightest hull, and the largest cargo capacity. Pressure vessel design has two inequality constraints and two equality constraints. It is to design a cylindrical pressure vessel with both endsby semicircular spherical heads, minimize the total cost (material cost, forming and welding cost) and maximize the capacity. Reactor network design is a CMOP with one inequality constraint and four equality constraints. Because most of its constraints are inequality constraints, it is the most difficult one of the three real-world CMOPs in this paper. Reactor network design aims to design a network containing two CSTR reactors. The volume of these two CSTR reactors can be as small as possible, and the concentration of products flowing out of the second container can be as high as possible.

TABLE II
Parameter settings for real-world CMOPs.

Problems	N	M	D	FEs
Bulk Carrier Design	100	3	6	300000
Pressure Vessel Design	100	2	2	300000
Reactor Network Design	100	2	6	300000

B. CMOEAs used for comparisons

We used five multi-stage CMOEAs and a classic CMOEA to compair our C3M. The five multi-stage CMOEA is MSCMO [32], PPS [28], CTAEA [25], ToP [27] and TiGE_2 [39]; The classic CMOEA is NSGAII-CDP [12] since the C3M in this paper is also based on Pareto domination. The differential evolution (DE) [40] and polynomial mutation (PM) [41] are used to reproduce for LIRCMOP, DOC, DASCMOP and real-world CMOPs, while simulated binary crossover (SBX) [42] and polynomial mutation (PM) are used to reproduce for the rest problems. This is because using different operators will perform differently on the same problem. [43] Therefore, we have comprehensively selected the operator that performs best on this problem. When the binary crossover of probability simulation was set to 1, the probability of polynomial mutation was set to 1 / D (D represents the number of decision variables). The distribution index of crossover and mutation was set to 20; CR differential evolution and F parameter were set to 1 and 0.5, respectively. The parameter settings in algorithms are the same as those in the original paper.

C. Performance Metrics

In order to evaluate the performance of each CMOEAs, we use the Inverted General Distance (IGD) [44] to evaluate the performance of each algorithm on the CMOP test suite since the PF of the CMOP test suite is known. When we evaluate the performance of each algorithm on the real-world CMOP, we use the hypervolume(HV) [45] to evaluate them since the PF of the real-world CMOP is unknown.

IGD is improved based on General Distance(GD) [46]. On the contrary to GD, IGD evaluates the diversity and convergence of the algorithm by calculating the average value of the nearest distance from the reference point of PF to the solution in the obtained population. The smaller the value of IGD, the better the diversity and convergence of the algorithm.

$$IGD(P, Q) = \frac{\sum_{v \in P} d(v, Q)}{|P|}, \tag{6}$$

where P is the point set uniformly distributed on the real PF, and $|P|$ is the number of individuals of the point set distributed on the real PF. Q is the optimal Pareto optimal solution set obtained by the algorithm. $D(v, Q)$ is the minimum Euclidean distance from individual v to population Q in P.

Suppose $R = (r_1, r_2, ..., r_m)$ is the vector in the objective that can dominate the constraint PF (called the reference point). HV measures the supervolume of the objective space bounded by R and dominated by P:

$$HV = vol(\bigcup_{i \in P} [f_1(i), r_1] \times \cdots [f_m(i), r_m]), \tag{7}$$

where $vol(\cdots)$ represents a Lebesgue metric of Supervolume. Similar to IGD, the HV index can also measure the diversity and convergence of the constrained PF approximation. The higher the HV value, the higher the approximation of P.

D. Experimental Results

Table III, Table IV and Table V shows the comparison of IGD mean and standard deviation of 30 independent operations on CDTLZ&MW, DASCMOP&DOC and LIRCMOP by our C3M respectively. The IGD value of each problem was calculated according to the method recommended in [34], which is based on about 10,000 reference points sampled on the problem PF. We also used the Wilcoxon rank test [47] with a significance level of 0.05 to analyze the results. Among them '+', '−' and '≈' respectively showed that the results of CMOEA were significantly better, significantly worse, and statistically similar to the results of the proposed C3M.

1) Comparison on CDTLZ and MW test suite: Figure 5 shows the convergence profiles of IGD on MW12 of the feasible solutions obtained by each algorithm. C3M can quickly converge in the early stage, then obtain the solutions closer to PF.

Table III lists the average and standard deviation of IGD obtained by the proposed C3M and its comparison algorithm on 30 independent CDTLZ test suites and MW test suites. The average IGD value of the C3M proposed in this paper is in the leading position on 22 problems, and we can find that the C3M is significantly better than MSCMO, PPS, CTAEA, NSGAII-CDP, ToP and TiGE_2 on 19, 24, 20, 22, 22 and 24 problems respectively.

Not only in multi-constraint MOPs, C3M perform well in single-constrained MOPs. The reason is that reinitialization helps the population to improve diversity, so the population is more likely to jump out of the local optimum caused by both MOP and constraints.

Figure 7(c) shows the results with median IGD of each algorithm in DC3_DTLZ1; we can see that our C3M and MSCMO find the final PF of three segments and have good convergence and diversity. Although PPS and CTAEA also find the final PF of three segments, their diversity on PF is poor; NSGAII-CDP, ToP, and TiGE_2 have poor convergence

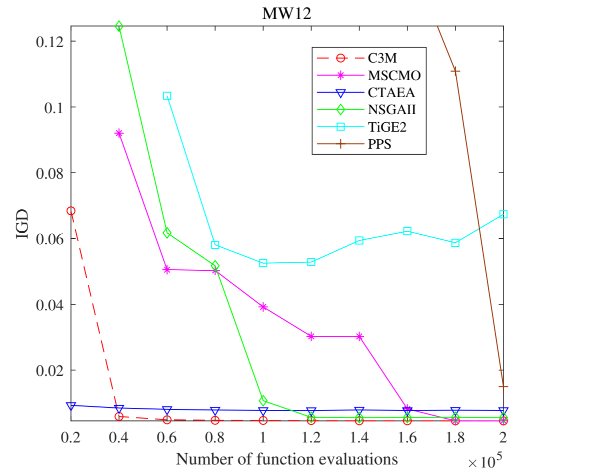


Fig. 5. Convergence profiles of feasible solutions of IGD values obtained by C3M, MSCMO, CTAEA, NSGAII-CDP, TiGE2, and PPS on MW12, averaged over 30 runs.

TABLE III
MEAN AND STANDARD DEVIATION OF IGD VALUES ON CDTLA AND MW PROBLEMS. 'NAN' INDICATES THAT NO FEASIBLE SOLUTION WAS FOUND. '+', '−', AND '≈' INDICATE THAT THE RESULT IS SIGNIFICANTLY BETTER, SIGNIFICANTLY WORSE, AND STATISTICALLY SIMILAR TO THAT OBTAINED BY C3M, RESPECTIVELY.

Problem	C	C3M	MSCMO	PPS	CTAEA	NSGAII	ToP	TiGE_2
C1_DTLZ1	1	1.9966e-2 (1.21e-4)	2.0100e-2 (1.79e-4) −	2.6331e-2 (8.07e-4) −	2.3334e-2 (1.68e-4) −	2.7325e-2 (1.34e-3) −	1.6567e-1 (1.74e-1) −	3.5892e-1 (8.11e-2) −
C1_DTLZ3	1	5.3271e-2 (3.79e-4)	5.3313e-2 (5.43e-4) ≈	1.1343e+0 (2.75e+0) −	6.7755e-2 (8.15e-3) −	4.3116e+0 (4.04e+0) −	1.6936e+0 (3.23e+0) −	5.8344e+0 (3.42e+0) −
C2_DTLZ2	1	4.2446e-2 (4.50e-4)	4.2510e-2 (4.97e-4) ≈	5.5456e-2 (1.78e-3) −	5.6709e-2 (1.46e-3) −	5.7047e-2 (1.73e-3) −	5.8888e-2 (3.09e-3) −	1.7819e-1 (1.48e-1) −
C3_DTLZ4	3	9.4516e-2 (1.48e-3)	9.5497e-2 (1.85e-3) ≈	1.2589e-1 (3.99e-3) −	1.1231e-1 (1.89e-3) −	1.2751e-1 (4.60e-3) −	1.4247e-1 (6.64e-3) −	1.9113e-1 (1.58e-2) −
DC1_DTLZ1	1	1.1368e-2 (7.01e-5)	1.1439e-2 (7.47e-5) −	2.0315e-2 (1.03e-3) −	1.5395e-2 (2.84e-4) −	1.4795e-2 (5.64e-4) −	1.9045e-2 (4.37e-3) −	4.2994e-1 (8.33e-2) −
DC1_DTLZ3	1	1.1343e-1 (6.91e-4)	1.1424e-1 (9.70e-4) −	1.4627e-1 (5.93e-2) −	1.2992e-1 (4.06e-3) −	1.2834e-1 (3.01e-3) −	6.0315e-1 (8.49e-1) −	1.2109e+0 (3.84e-1) −
DC2_DTLZ1	2	1.9982e-2 (1.04e-4)	2.0138e-2 (1.25e-4) −	2.8219e-2 (7.21e-4) −	2.3390e-2 (1.95e-4) −	NaN (NaN)	NaN (NaN)	3.6846e-1 (9.12e-2) −
DC2_DTLZ3	2	1.8831e-1 (2.29e-1)	5.3049e-2 (4.45e-4) ≈	1.9821e-1 (2.19e-1) −	6.3377e-2 (3.85e-4) +	NaN (NaN)	NaN (NaN)	1.0363e+0 (1.01e-2) −
DC3_DTLZ1	3	6.7842e-3 (4.48e-5)	1.2863e-2 (2.97e-2) −	2.7908e-2 (6.14e-2) −	9.6593e-3 (5.88e-4) −	1.2381e-1 (8.92e-2) −	7.4386e-1 (8.35e-1) −	9.1489e-1 (8.30e-1) −
DC3_DTLZ3	3	2.4634e-1 (1.80e-1)	3.0303e-1 (1.87e-1) −	6.4174e-1 (5.37e-1) −	1.7246e-1 (3.95e-3) +	1.4953e+0 (5.31e-1) −	5.8871e+0 (2.92e+0) −	2.8750e+0 (7.16e-1) −
MW1	1	1.6036e-3 (1.21e-5)	1.7986e-3 (9.91e-4) −	2.6461e-3 (9.48e-5) −	2.0162e-3 (8.74e-5) −	2.1389e-3 (1.00e-3) −	1.2064e-1 (1.15e-1) −	1.2755e-2 (3.00e-3) −
MW2	1	1.5447e-2 (7.87e-3)	2.1313e-2 (9.02e-3) −	1.4608e-1 (9.26e-2) −	1.8561e-2 (7.07e-3) ≈	2.4831e-2 (9.72e-3) −	1.4421e-1 (1.06e-1) −	6.4951e-1 (1.03e-1) −
MW3	2	4.4785e-3 (1.70e-4)	4.6700e-3 (1.52e-4) −	6.4131e-3 (5.49e-4) −	4.7189e-3 (2.35e-4) −	5.7574e-3 (2.75e-4) −	5.2863e-1 (4.56e-1) −	2.3021e-2 (3.26e-3) −
MW4	1	4.0214e-2 (3.14e-4)	4.1002e-2 (1.85e-3) −	5.4640e-2 (1.88e-3) −	4.6731e-2 (3.34e-4) −	5.4905e-2 (2.51e-3) −	4.5109e-1 (3.14e-1) −	1.0526e-1 (3.45e-2) −
MW5	3	6.4878e-4 (8.49e-4)	1.5515e-3 (3.28e-3) −	3.0190e-1 (3.65e-1) −	8.9400e-3 (2.25e-3) −	3.9697e-1 (3.57e-1) −	5.3904e-1 (3.63e-1) −	3.5542e-2 (7.25e-3) −
MW6	1	9.1395e-3 (4.98e-3)	5.2849e-2 (1.41e-1) −	5.1663e-1 (3.53e-1) −	9.8187e-3 (6.42e-3) ≈	3.2016e-2 (2.75e-2) −	6.5359e-1 (3.43e-1) −	1.7329e-1 (2.74e-1) −
MW7	2	4.0176e-3 (2.00e-4)	4.0427e-3 (2.20e-4) ≈	5.7884e-3 (7.33e-4) −	6.1762e-3 (5.40e-4) −	6.4411e-2 (1.55e-1) −	2.9893e-2 (7.93e-2) −	4.6730e-2 (2.08e-2) −
MW8	1	4.3863e-2 (1.34e-3)	4.6848e-2 (6.69e-3) −	1.3875e-1 (5.23e-2) −	5.5336e-2 (3.65e-3) −	6.4375e-2 (2.10e-2) −	2.8223e-1 (2.95e-1) −	6.9381e-1 (8.04e-2) −
MW9	1	4.2245e-3 (3.00e-4)	4.6480e-3 (5.18e-4) −	1.0569e-2 (5.13e-3) −	8.1438e-3 (7.42e-4) −	6.7222e-3 (1.78e-3) −	6.6566e-1 (2.26e-1) −	1.9260e-1 (2.44e-1) −
MW10	3	8.8451e-3 (7.59e-3)	7.2197e-2 (1.49e-1) −	3.4383e-1 (2.15e-1) −	1.9931e-2 (2.61e-2) −	1.4188e-1 (1.54e-1) −	3.1792e-1 (2.24e-1) −	9.2009e-2 (1.09e-1) −
MW11	4	5.8504e-3 (7.77e-5)	3.2193e-2 (4.14e-2) −	7.5853e-3 (4.93e-4) −	1.2319e-2 (1.21e-3) −	1.9974e-1 (3.02e-1) −	3.2881e-1 (3.08e-1) −	3.5692e-2 (6.38e-3) −
MW12	2	4.5454e-3 (5.77e-5)	4.5798e-3 (7.43e-5) −	7.4448e-3 (1.70e-3) −	7.7027e-3 (5.16e-4) −	5.5968e-3 (2.31e-4) −	7.0290e-1 (3.07e-1) −	4.8015e-2 (5.63e-2) −
MW13	2	1.6751e-2 (1.28e-2)	8.3797e-2 (4.20e-2) −	4.8714e-1 (4.04e-1) −	3.9365e-2 (2.25e-2) −	1.8793e-1 (3.37e-1) −	4.1059e-1 (4.19e-1) −	1.4749e+0 (7.35e-1) −
MW14	1	9.6227e-2 (1.37e-3)	2.0938e-1 (3.66e-1) −	1.3709e-1 (2.28e-2) −	1.1121e-1 (4.03e-3) −	1.2293e-1 (6.18e-3) −	2.6084e-1 (1.63e-1) −	1.5865e-1 (9.73e-3) −
+/ − / ≈			0/19/5	0/24/0	2/20/2	0/22/0	0/22/0	0/24/0

and do not approach the final PF.

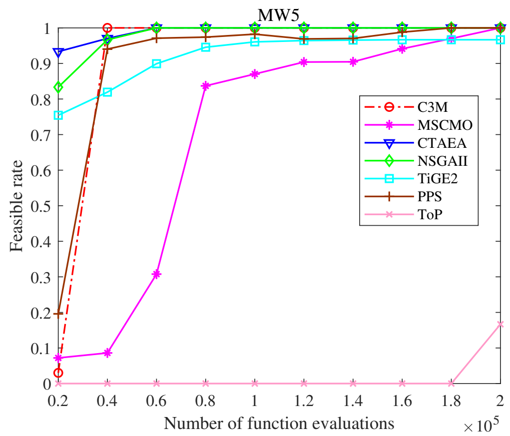


Fig. 6. The profiles of feasible rates obtained by C3M, MSCMO, CTAEA, NSGAII-CDP, TiGE2, ToP, and PPS on MW5, averaged over 30 runs.

The PF of MW5 is discontinuous and only consists of multiple points. Therefore, it is not easy to obtain a feasible solution while ensuring convergence. As shown in Figure 6, the proportion of feasible solutions of our algorithm and the comparison algorithm in the operation process. Our algorithm, NSGAII-CDP, and CTAEA can obtain a feasible solution in early stage, while MSCMO and PPS fully bring the feasible solutions at the last time. The solutions of ToP and TiGE_2 are not completely feasible.

Fig. 7(a) shows the performance of the result with median IGD in each algorithm on MW13; MW13 has the characteristics of a slender and narrow feasible region and easy to fall into local optimum. From the diversity perspective, we can see that C3M has wholly obtained all PF, while MSCMO, CTAEA, and NSGAII-CDP have obtained three discontinuous PF, but their final solutions are not entirely distributed on PF, while PPS, ToP, and TiGE_2 have only obtained part of PF. From the perspective of convergence, only C3M and CTAEA have relatively good convergence, relatively close to PF.

2) Comparison on DASCMOP&DOCs: DASCMOP consists of three types of constraints that have customizable difficulty. Users can manually specify parameters according to the required feasibility, convergence, and diversity difficulty. In DASCMOP, We find that constraint 2 has relation B to all other constraints, so C3M can quickly obtain a satisfactory feasible solution when dealing with DASCMOP.

DOC is a challenging constraint test suite with constraints in both the objective and decision space. There are three main difficulties in this problem. First, the proportion of the final feasible region is small, and it is difficult to find the approximate range of the feasible region related to the constrained PF. Second, many evaluation times are required to obtain solutions approaching the constrained PF after finding the approximate feasible region. Third, some problems are prone to local optima (such as DOC3). The relaxation of constraints in the first two stages enables C3M to cope with the situation where the feasible region is small, and reinitialization can also cope with local optimization to a certain extent. In addition, many constraints in the DOC problem have relationship B, which C3M can utilize to save evaluation times in stage 2. Therefore, C3M performs well in this test suite.

Table IV shows the performance of our C3M and other compared algorithms on DASCMOP and DOC test suites. We can see that C3M leads the average IGD value on 12 problems and is significantly better than MSCMO, PPS, CTAEA, NSGAII-CDP, ToP, and TiGE_2 on 8, 14, 14, 15, 16 and 18 problems respectively.

As shown in Figure 7(d), We can see that C3M not only approximates PF but also has the best diversity; MSCMO also approximates PF, but its distribution is slightly inferior to C3M; PPS, CTAEA, and NSGAII-CDP are approximate to PF, but their diversity on PF is poor; ToP and TiGE_2 only a few individuals approach PF, and the diversity is the worst.

Fig. 7(b) shows the performance of each algorithm on DOC1. It can be seen that C3M, MSCMO, and top are close to PF and have good diversity; PPS approximates PF, but not wholly approximates PF; CTAEA, NSGAII-CDP, and TiGE_2 did not approach PF.

3) Comparison on LIRCMOPs: The LIRCMOP test suite is mainly used to test the performance of algorithms in the face of large infeasible regions. It contains 2-3 constraints. Because the first stage of C3M does not consider constraints,

TABLE IV
MEAN AND STANDARD DEVIATION OF IGD VALUES ON DASCMOP, DOC PROBLEMS. 'NAN' INDICATES THAT NO FEASIBLE SOLUTION WAS FOUND. '+', '−', AND '≈' INDICATE THAT THE RESULT IS SIGNIFICANTLY BETTER, SIGNIFICANTLY WORSE, AND STATISTICALLY SIMILAR TO THAT OBTAINED BY C3M, RESPECTIVELY.

Problem	C	C3M	MSCMO	PPS	CTAEA	NSGAII	ToP	TiGE_2
DASCMOP1	11	2.3892e-3 (5.10e-4)	3.0216e-3 (1.42e-3) ≈	2.3292e-3 (9.20e-4) ≈	6.3068e-2 (8.47e-2) −	4.1985e-1 (2.18e-1) −	6.4864e-2 (1.62e-1) ≈	3.4212e-2 (2.74e-2) −
DASCMOP2	11	2.9398e-3 (6.95e-5)	2.9823e-3 (7.27e-5) −	3.9394e-3 (1.23e-4) −	1.0151e-2 (5.68e-3) −	1.3585e-1 (3.53e-2) −	2.7327e-2 (5.41e-2) −	3.0903e-2 (1.58e-2) −
DASCMOP3	11	1.8440e-2 (2.38e-3)	1.9169e-2 (1.27e-3) ≈	1.9283e-2 (9.20e-5) ≈	3.1075e-2 (9.05e-3) −	2.7457e-1 (5.83e-2) −	2.1113e-1 (1.46e-1) −	5.6374e-2 (5.45e-2) −
DASCMOP4	11	1.2557e-3 (1.07e-4)	1.1921e-3 (9.47e-5) +	8.4760e-3 (3.68e-2) −	1.0968e-2 (2.13e-3) −	5.5425e-2 (1.10e-1) −	8.2824e-1 (2.15e-1) −	9.1731e-2 (7.41e-2) −
DASCMOP5	11	3.0343e-3 (1.55e-4)	3.0769e-3 (1.72e-4) ≈	4.0588e-3 (1.49e-4) −	7.2231e-3 (4.99e-4) −	3.5638e-3 (9.36e-5) −	7.2443e-1 (9.81e-2) −	9.6106e-2 (8.28e-2) −
DASCMOP6	11	1.7647e-2 (2.99e-3)	2.0611e-2 (6.01e-3) −	2.2590e-2 (2.26e-2) ≈	2.3311e-2 (3.41e-3) −	5.7046e-2 (9.79e-2) ≈	7.9451e-1 (1.32e-1) −	1.4234e-1 (1.09e-1) −
DASCMOP7	7	3.8939e-2 (3.23e-3)	4.9352e-2 (1.60e-2) −	6.2540e-2 (7.01e-3) −	5.7441e-2 (9.40e-3) −	4.8783e-2 (2.98e-3) −	8.3722e-1 (2.62e-1) −	2.7788e-1 (2.11e-1) −
DASCMOP8	7	4.8767e-2 (8.11e-3)	6.2360e-2 (2.30e-2) −	8.8851e-2 (1.09e-2) −	9.3247e-2 (1.27e-2) −	6.0896e-2 (2.85e-3) −	8.3965e-1 (2.44e-1) −	3.9314e-1 (1.81e-1) −
DASCMOP9	7	4.0725e-2 (1.01e-3)	4.0795e-2 (1.08e-3) ≈	8.2867e-2 (4.38e-2) −	9.5401e-2 (6.55e-3) −	6.7934e-2 (4.63e-3) −	6.2105e-2 (3.60e-3) −	2.0189e-1 (7.35e-2) −
DOC1	7	5.1586e-3 (3.43e-4)	5.1666e-3 (4.13e-4) ≈	6.1366e-2 (4.04e-2) −	5.0113e+2 (2.56e+2) −	2.2824e+0 (1.46e+0) −	5.8070e-3 (3.37e-4) −	2.9317e+0 (1.53e+0) −
DOC2	7	1.2979e-2 (1.14e-2)	6.3690e-3 (4.38e-3) +	4.2341e-1 (1.67e-1) −	NaN (NaN)	NaN (NaN)	4.7837e-1 (5.08e-2) −	1.4414e-2 (3.40e-3) −
DOC3	10	5.9689e+2 (3.93e+2)	1.9371e+2 (1.97e+2) +	2.2031e+2 (1.75e+2) +	NaN (NaN)	7.2358e+2 (1.99e+2) −	6.8810e+1 (1.07e+2) +	7.6407e+2 (4.84e+2) −
DOC4	6	2.1242e-2 (5.38e-3)	2.5188e-2 (6.47e-3) −	3.0231e-1 (7.75e-2) −	2.2190e+2 (2.72e+2) −	1.2371e+0 (1.72e+0) −	7.3291e-2 (4.98e-2) −	9.0698e+0 (4.36e+0) −
DOC5	9	3.8962e-2 (6.20e-2)	8.7489e+0 (3.37e+1) −	7.2733e+1 (7.87e+1) −	NaN (NaN)	NaN (NaN)	3.8859e+1 (5.93e+1) −	4.4514e+1 (6.05e+1) −
DOC6	10	2.7418e-3 (1.54e-4)	3.1536e-3 (9.37e-4) −	4.8207e-1 (7.66e-2) −	5.7547e+1 (1.00e+2) −	2.4392e+0 (2.68e+0) −	2.8619e+0 (1.59e+0) −	8.9932e+0 (2.82e+0) −
DOC7	6	2.8044e-3 (7.60e-4)	2.3580e-3 (1.26e-4) +	5.2854e-1 (1.10e-1) −	NaN (NaN)	6.1653e+0 (2.46e+0) −	2.2880e-1 (2.40e-1) −	1.6298e+0 (1.25e+0) −
DOC8	7	6.1969e-2 (3.07e-3)	6.2679e-2 (3.72e-3) ≈	7.7593e+1 (3.13e+1) −	3.6186e+2 (1.55e+2) −	5.8966e+1 (5.99e+1) −	1.3858e+1 (7.77e+0) −	8.1622e+1 (2.14e+1) −
DOC9	14	7.5354e-2 (1.22e-2)	8.4572e-2 (1.24e-2) −	2.7260e-1 (2.02e-2) −	9.8499e-1 (3.47e-1) −	1.7119e-1 (7.73e-2) −	1.7368e-1 (3.36e-2) −	8.7195e-2 (2.13e-2) −
+/ − / ≈			4/8/6	1/14/3	0/14/0	0/15/1	1/16/1	0/18/0

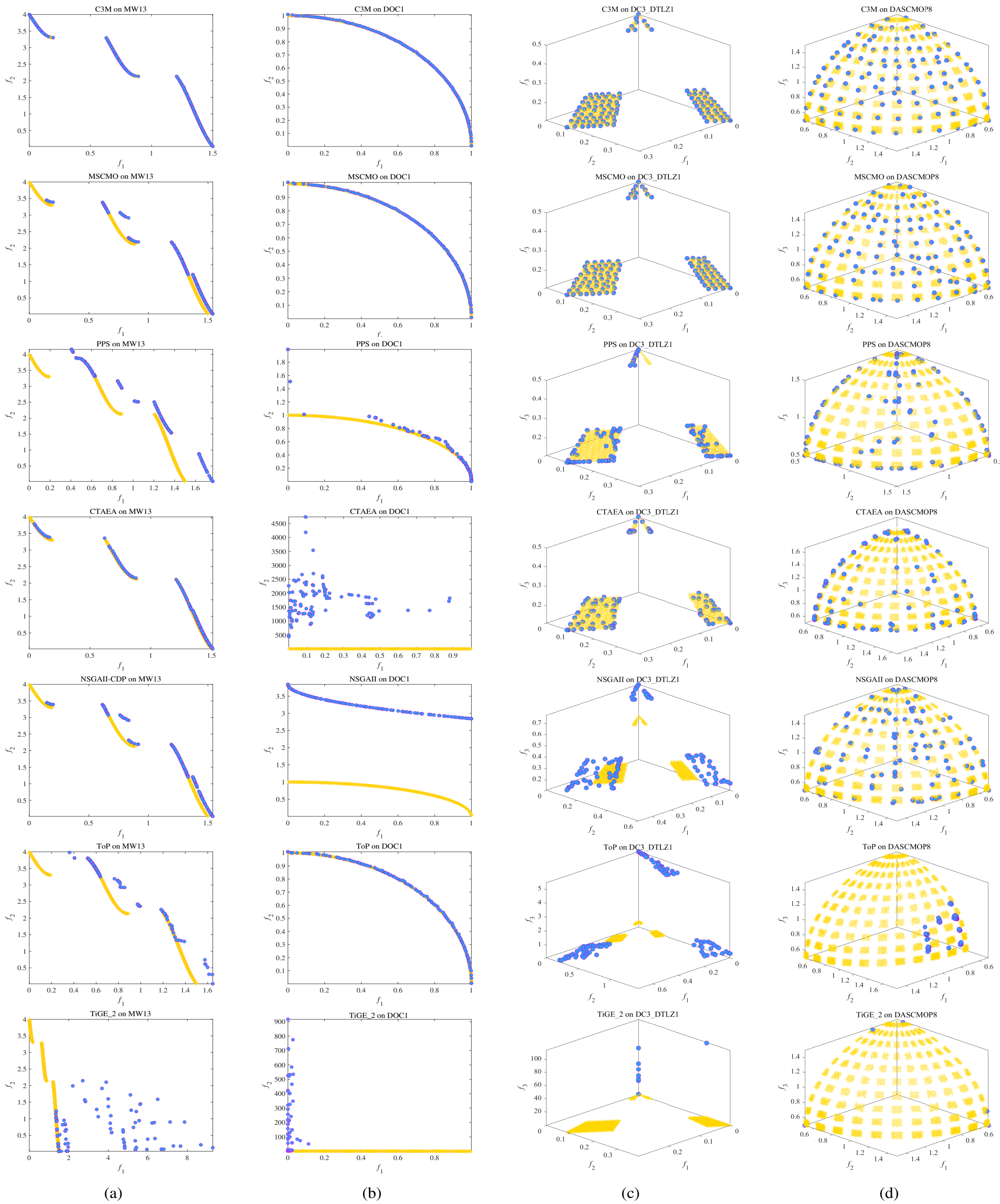


Fig. 7. Solutions with median IGD value among 30 runs obtained by C3M, MSCMO, PPS, CTAEA, NSGAII-CDP, ToP, and TiGE_2 on MW13, DOC1, DC3_DTLZ1, and DASCMOP8, respectively. Where, each column identified by (a), (b), (c), and (d) represents the graph of the nondominated solutions of the comparison experiment between C3M and the other algorithm. Blue circles represent the obtained solutions, and orange cross represent constraint PF.

TABLE V
MEAN AND STANDARD DEVIATION OF IGD VALUES ON LIRCMOP PROBLEMS AND REAL-WORD CMOPS. 'NAN' INDICATES THAT NO FEASIBLE SOLUTION WAS FOUND. '+', '−', AND '≈' INDICATE THAT THE RESULT IS SIGNIFICANTLY BETTER, SIGNIFICANTLY WORSE, AND STATISTICALLY SIMILAR TO THAT OBTAINED BY C3M, RESPECTIVELY.

Problem	C	C3M	MSCMO	PPS	CTAEA	NSGAII	ToP	TiGE_2
LIRCMOP1	2	4.3986e-2 (4.18e-2)	1.8357e-2 (1.44e-2) +	7.5073e-3 (1.39e-3) +	1.9327e-1 (9.34e-2) −	2.0663e-1 (8.35e-2) −	1.2764e-1 (1.21e-1) −	4.0719e-2 (1.99e-2) ≈
LIRCMOP2	2	3.4528e-2 (2.82e-2)	1.1771e-2 (1.36e-2) +	5.9131e-3 (4.09e-4) +	5.5501e-2 (1.13e-2) −	1.5540e-1 (6.42e-2) −	7.6816e-2 (6.46e-2) −	3.2451e-2 (1.02e-2) ≈
LIRCMOP3	3	1.5573e-1 (5.56e-2)	5.5101e-2 (3.65e-2) +	7.8132e-3 (5.38e-3) +	1.9832e-1 (1.50e-1) ≈	2.3769e-1 (9.20e-2) −	3.6098e-1 (6.70e-2) −	2.8716e-2 (1.50e-2) +
LIRCMOP4	3	1.5166e-1 (7.05e-2)	6.6288e-2 (5.16e-2) +	3.3364e-3 (2.54e-4) +	1.2986e-1 (5.46e-2) ≈	2.2684e-1 (6.69e-2) −	3.1266e-1 (2.95e-2) −	3.6142e-2 (2.36e-2) +
LIRCMOP5	2	4.9316e-3 (1.92e-4)	4.9850e-3 (1.82e-4) ≈	6.5742e-3 (6.81e-4) −	7.0247e-2 (1.63e-2) −	6.4944e-1 (5.15e-1) −	6.7428e-3 (4.37e-4) −	5.7894e-1 (9.26e-2) −
LIRCMOP6	2	5.0152e-3 (2.04e-4)	5.0219e-3 (1.81e-4) ≈	7.8309e-3 (7.47e-4) −	1.4804e-1 (1.58e-1) −	3.4787e-1 (4.67e-1) −	1.6523e-2 (5.54e-2) −	6.0042e-1 (2.92e-2) −
LIRCMOP7	3	7.0314e-3 (2.20e-4)	7.0552e-3 (1.85e-4) ≈	1.0904e-2 (1.04e-3) −	1.9908e-2 (5.60e-3) −	9.9038e-3 (2.59e-3) −	8.5331e-3 (3.06e-4) −	7.4770e-1 (4.43e-1) −
LIRCMOP8	3	7.0300e-3 (2.23e-4)	7.0800e-3 (1.73e-4) ≈	1.0913e-2 (1.08e-3) −	1.6509e-2 (2.91e-3) −	2.1992e-2 (3.65e-2) −	2.3385e-2 (8.03e-2) −	1.0505e+0 (3.29e-1) −
LIRCMOP9	2	2.4608e-3 (3.98e-5)	2.4690e-3 (6.62e-5) ≈	3.1711e-3 (1.28e-4) −	5.2635e-2 (2.35e-2) −	4.1994e-1 (1.01e-1) −	2.9012e-1 (1.57e-1) −	6.2572e-1 (9.45e-2) −
LIRCMOP10	2	4.1612e-3 (1.01e-4)	4.2269e-3 (1.24e-4) −	5.1554e-3 (2.21e-4) −	7.4999e-2 (7.47e-2) −	2.8954e-1 (1.01e-1) −	5.5306e-3 (2.26e-4) −	6.2778e-1 (1.61e-1) −
LIRCMOP11	2	2.3629e-3 (4.57e-5)	2.3665e-3 (4.89e-5) ≈	2.4304e-3 (7.59e-5) −	1.1519e-1 (3.91e-2) −	1.2104e-1 (9.12e-2) −	1.4136e-1 (3.37e-2) −	4.3439e-1 (1.78e-1) −
LIRCMOP12	2	3.0026e-3 (1.13e-4)	3.0481e-3 (1.47e-4) ≈	3.1170e-3 (7.32e-5) −	1.4681e-2 (4.46e-3) −	1.0378e-1 (8.57e-2) −	3.0856e-2 (5.09e-2) −	7.6258e-1 (5.30e-1) −
LIRCMOP13	2	1.0632e-1 (1.74e-3)	1.0657e-1 (1.78e-3) ≈	1.2382e-1 (3.27e-3) −	1.0853e-1 (1.74e-3) −	1.1789e-1 (4.34e-3) −	1.2595e-1 (3.76e-3) −	8.1730e-1 (7.68e-2) −
LIRCMOP14	3	1.0016e-1 (1.14e-3)	1.0015e-1 (9.51e-4) ≈	1.1743e-1 (2.45e-3) −	1.1122e-1 (1.02e-3) −	1.6105e-1 (2.12e-1) −	1.1947e-1 (3.12e-3) −	9.1662e-1 (1.78e-1) −
Bulk Carrier Design	9	4.0871e-1 (1.40e-3)	3.2351e-1 (6.54e-2) −	1.6285e-1 (3.51e-2) −	2.4525e-1 (3.98e-2) −	2.7054e-1 (2.78e-3) −	2.5736e-1 (3.36e-2) −	2.7637e-1 (6.22e-2) −
Pressure Vessel Design	4	6.0838e-1 (2.96e-4)	6.0812e-1 (3.92e-4) −	5.9497e-1 (2.15e-3) −	6.0682e-1 (1.16e-3) −	6.0535e-1 (6.89e-4) −	6.0800e-1 (3.21e-4) −	5.3694e-1 (3.54e-2) −
Reactor Network Design	5	8.9091e-1 (9.13e-2)	NaN (NaN)	8.1685e-1 (3.97e-2) −	NaN (NaN)	2.7940e-1 (1.33e-1) −	4.3282e-1 (2.24e-1) −	5.2943e-1 (2.56e-1) −
+/ − / ≈			4/3/9	4/13/0	0/14/2	0/17/0	0/17/0	2/13/2

it can well deal with the problem of the same or partial same of unconstrained PF and constrained PF. When facing the problem of unconstrained PF and constrained PF not being the same, evaluating a single constraint can also help the algorithm quickly reach the approximate position of constrained PF. Stage 3 can further optimize the solutions obtained in the first two stages. PPS performs best in LIRCMOP1-4 since the feasible area of LIRCMOP1-4 is extremely narrow, and PPS makes high use of infeasible information near the constraint PF.

Table V shows the performance of C3M and its comparison algorithm on the LIRCMOP test suite. It can be seen that PPS has the best IGD on the LIRCMOP1-4 problem, and on the remaining problems, although C3M is dominant in the average IGD value, it is only slightly outperformance of MSCMO. Compared with MSCMO, PPS, CTAEA, NSGAII-CDP, ToP, and TiGE_2, our C3M has significant advantages on 1,10,12,14,14 and 12 problems, respectively.

4) Comparison on Real-World CMOPs: Finally, we test our C3M on the real-world test problem. The Table V show that the HV of our C3M is significant outperformance on three problems.

E. Further investigations of C3M

In this section, we use ablation experiments to verify the effectiveness of the part of the proposed algorithm, so we design three variants for the algorithm. The first variant is used to verify the effectiveness of constraint-handling priority; at the end of stage 1, the constraint-handling priority will not be determined, but a random sequence is used as the priority of constraints; The second variant is used to verify the importance of giving up less-important constraints. When this variant evaluates a single constraint, all constraints will be evaluated until it reaches the threshold to force it into stage 3 (i.e., Algorithm 4 will not be executed.); The third variant is used to verify the effectiveness of reinitialization, which does not reinitialization when algorithm 1 from stage 1 to stage 2 and switching evaluation constraints on stage 2, but directly uses its Pop[i].

Table VI shows the results of C3M and its three variants on DASCMOP test suite. We can see that C3M obtained the six best average values on DASCMOP. The average values of C3M1 and C3M2 are not optimal, which are significantly worse than C3M in four and seven problems respectively. C3M3 has the advantage of average value in three problems, but it is significantly worse than C3M in three problems proving that our strategy is effective.

TABLE VI
MEAN AND STANDARD DEVIATION OF IGD VALUES ON DASCMOP PROBLEMS. '+', '−', AND '≈' INDICATE THAT THE RESULT IS SIGNIFICANTLY BETTER, SIGNIFICANTLY WORSE, AND STATISTICALLY SIMILAR TO THAT OBTAINED BY C3M RESPECTIVELY.

Problem	C3M	C3M1	C3M2	C3M3
DASCMOP1	2.3892e-3 (5.10e-4)	2.6924e-3 (9.55e-4) ≈	3.4271e-3 (3.89e-3) ≈	4.0248e-3 (4.31e-3) −
DASCMOP2	2.9398e-3 (6.95e-5)	2.9901e-3 (6.95e-5) −	3.1774e-3 (7.86e-5) −	3.0057e-3 (1.06e-4) −
DASCMOP3	1.8440e-2 (2.38e-3)	1.8933e-2 (1.76e-3) ≈	1.9253e-2 (1.20e-3) −	1.9373e-2 (6.74e-5) ≈
DASCMOP4	1.2557e-3 (1.07e-4)	1.2690e-3 (2.19e-4) ≈	1.8841e-3 (9.50e-4) −	1.1583e-3 (2.26e-5) +
DASCMOP5	3.0343e-3 (1.55e-4)	3.7113e-3 (8.39e-4) −	3.4299e-3 (2.80e-4) −	2.9679e-3 (1.14e-4) ≈
DASCMOP6	1.7647e-2 (2.99e-3)	1.9352e-2 (1.05e-3) ≈	1.9305e-2 (1.22e-3) ≈	1.9363e-2 (1.22e-3) ≈
DASCMOP7	3.8939e-2 (3.23e-3)	4.3542e-2 (6.03e-3) −	5.9675e-2 (9.01e-2) −	4.1139e-2 (3.79e-3) −
DASCMOP8	4.8767e-2 (8.11e-3)	5.3652e-2 (8.15e-3) −	6.3037e-2 (5.58e-2) −	4.7481e-2 (3.42e-3) ≈
DASCMOP9	4.0725e-2 (1.01e-3)	4.1097e-2 (1.18e-3) ≈	4.4762e-2 (1.57e-2) −	4.0843e-2 (9.14e-4) ≈
+/ − / ≈		0/4/5	0/7/2	1/3/5

V. CONCLUSION

In this paper, we discuss the relationship between constraints and propose a constraint-handling priority strategy based on the relationship between constraints; An adaptive parameter is designed to detect whether the population reaches the edge of constraints or unconstrained PF. We designed a multi-stage CMOEA, which completely ignored the feasibility in the early stage, considered part of the feasibility in the middle stage, and all the feasibility in the last stage. The results on five constraint test suites and three real-world CMOPs show that our C3M is very competitive.

The C3M proposed in this paper is generally effective, but we hope to study it further. 1) There is a small possibility of miscalculating constraint-handling priority, so whether we can find a way to reduce this possibility or whether we can find a way to use existing prior knowledge to start the algorithm directly from stage 2. 2) Can we further study the relationship in motivation so that stage 2 does not evaluate a single constraint but evaluates similar constraints to save evaluation times further?

REFERENCES

[1] X. Yang, J. Zou, S. Yang, J. Zheng, and Y. Liu, "A fuzzy decision variables framework for large-scale multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2021.

[2] Z.-H. Zhan, J. Li, J. Cao, J. Zhang, H. S.-H. Chung, and Y.-H. Shi, "Multiple populations for multiple objectives: A coevolutionary technique for solving multiobjective optimization problems," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 445–463, 2013.

[3] B. Kannan and S. Kramer, "An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design," *Journal of Mechanical Design - J MECH DESIGN*, vol. 116, 06 1994.

[4] A. Edpuganti and A. K. Rathore, "Optimal pulsewidth modulation for common-mode voltage elimination scheme of medium-voltage modular multilevel converter-fed open-end stator winding induction motor drives," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 1, pp. 848–856, 2017.

[5] F. Rivas-Davalos and M. Irving, "An approach based on the strength pareto evolutionary algorithm 2 for power distribution system planning," vol. 3410, pp. 707–720, 03 2005.

[6] K. Deb, *Constrained Multi-objective Evolutionary Algorithm*, 01 2019, pp. 85–118.

[7] H. Tang, F. Yu, J. Zou, S. Yang, and J. Zheng, "A constrained multi-objective evolutionary strategy based on population state detection," *Swarm Evol. Comput.*, vol. 68, p. 100978, 2022. [Online]. Available: https://doi.org/10.1016/j.swevo.2021.100978

[8] Q. Zhang, A. Zhou, and Y. Jin, "Rm-meda: A regularity model-based multiobjective estimation of distribution algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 12, pp. 41 – 63, 03 2008.

[9] Z. Cai and Y. Wang, "A multiobjective optimization-based evolutionary algorithm for constrained optimization," *IEEE Transactions on Evolutionary Computation*, vol. 10(6), pp. 658 – 675, 01 2007.

[10] J. Zhou, J. Zou, J. Zheng, S. Yang, D. Gong, and T. Pei, "An infeasible solutions diversity maintenance epsilon constraint handling method for evolutionary constrained multiobjective optimization," *Soft Comput.*, vol. 25, no. 13, pp. 8051–8062, 2021. [Online]. Available: https://doi.org/10.1007/s00500-021-05880-5

[11] Y. Tian, T. Zhang, J. Xiao, X. Zhang, and Y. Jin, "A coevolutionary framework for constrained multi-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. PP, 06 2020.

[12] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii, ieee trans. on evol," *IEEE Transactions on Evolutionary Computation - TEC*, vol. 6, 01 2002.

[13] T. Takahama and S. Sakai, "Constrained optimization by the epsilon constrained differential evolution with gradient-based mutation and feasible elites," *2006 IEEE Congress on Evolutionary Computation, CEC 2006*, pp. 1–8, 01 2006.

[14] S. Sakai and T. Takahama, "Efficient constrained optimization by the epsilon constrained rank-based differential evolution," *2012 IEEE Congress on Evolutionary Computation, CEC 2012*, pp. 1–8, 08 2010.

[15] Z. Fan, W. Li, X. Cai, H. Huang, Y. Fang, Y. Yugen, J. Mo, C. Wei, and E. Goodman, "An improved epsilon constraint-handling method in moea/d for cmops with large infeasible regions," *Soft Computing*, vol. 23, 12 2019.

[16] Z. Fan, W. Li, X. Cai, K. Hu, H. Lin, and H. Li, "Angle-based constrained dominance principle in moea/d for constrained multi-objective optimization problems," pp. 460–467, 2016.

[17] Y. Woldesenbet, B. Tessema, and G. Yen, "Constraint handling in multiobjective evolutionary optimization," pp. 3077–3084, 09 2007.

[18] T. Ray, H. K. Singh, A. Isaacs, and W. Smith, *Infeasibility Driven Evolutionary Algorithm for Constrained Optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[19] H.-L. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Transactions on Evolutionary Computation*, vol. 18, pp. 450–455, 06 2014.

[20] T. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 284–294, 2000.

[21] G. Huantong, M. Zhang, L. Huang, and X. Wang, "Infeasible elitists and stochastic ranking selection in constrained evolutionary multi-objective optimization," vol. 4247, pp. 336–344, 10 2006.

[22] M. A. Jan and R. Khanum, "A study of two penalty-parameterless constraint handling techniques in the framework of moea/d," *Applied Soft Computing*, vol. 13, p. 128–148, 01 2013.

[23] J. Zou, R. Sun, S. Yang, and J. Zheng, "A dual-population algorithm based on alternative evolution and degeneration for solving constrained multi-objective optimization problems," *Information Sciences*, vol. 579, pp. 89–102, 2021.

[24] M. Ming, A. Trivedi, R. Wang, D. Srinivasan, and T. Zhang, "A dual-population-based evolutionary algorithm for constrained multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 4, pp. 739–753, 2021.

[25] K. Li, R. Chen, G. Fu, and X. Yao, "Two-archive evolutionary algorithm for constrained multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 303–315, 2019.

[26] Y. Tian, Y. Zhang, Y. Su, X. Zhang, K. Tan, and Y. Jin, "Balancing objective optimization and constraint satisfaction in constrained evolutionary multi-objective optimization," *IEEE Transactions on Cybernetics*, 08 2020.

[27] Z. Liu and Y. Wang, "Handling constrained multiobjective optimization problems with constraints in both the decision and objective spaces," *IEEE Transactions on Evolutionary Computation*, vol. PP, pp. 1–1, 01 2019.

[28] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, and E. Goodman, "Push and pull search for solving constrained multi-objective optimization problems," *Swarm and Evolutionary Computation*, vol. 44, 09 2017.

[29] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, pp. 712 – 731, 01 2008.

[30] F. Ming, W. Gong, H. Zhen, S. Li, L. Wang, and Z. Liao, "A simple two-stage evolutionary algorithm for constrained multi-objective optimization," *Knowledge-Based Systems*, vol. 228, p. 107263, 2021.

[31] Q. Zhu, Q. Zhang, and Q. Lin, "A constrained multiobjective evolutionary algorithm with detect-and-escape strategy," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 5, pp. 938–947, 2020.

[32] H. Ma, H. Wei, Y. Tian, R. Cheng, and X. Zhang, "A multi-stage evolutionary algorithm for multi-objective optimization with complex constraints," *Information Sciences*, vol. 560, 01 2021.

[33] Z. Ma and Y. Wang, "Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons," *IEEE Transactions on Evolutionary Computation*, vol. PP, pp. 1–1, 02 2019.

[34] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "Platemo: A matlab platform for evolutionary multi-objective optimization," *IEEE Computational Intelligence Magazine*, vol. 12, pp. 73–87, 11 2017.

[35] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: Handling constraints and extending to an adaptive approach," *IEEE Transactions on Evolutionary Computation*, vol. 18, pp. 602–622, 08 2014.

[36] Z. Fan, W. Li, X. Cai, H. Li, Q. Zhang, K. Deb, and E. Goodman, "Difficulty adjustable and scalable constrained multi-objective test problem toolkit," *Evolutionary Computation*, vol. 28, pp. 1–28, 05 2019.

[37] M. Parsons and R. Scott, "Formulation of multicriterion design optimization problems for solution with scalar numerical optimization methods," *Journal of Ship Research*, vol. 48, pp. 61–76, 03 2004.

[38] A. Gujarathi, S. Purohit, and B. Srikanth, "Optimization of reactor network design problem using jumping gene adaptation of differential evolution," *Journal of Physics: Conference Series*, vol. 622, p. 012044, 06 2015.

[39] Y. Zhou, M. Zhu, J. Wang, Z. Zhang, Y. Xiang, and J. Zhang, "Tri-goal evolution framework for constrained many-objective optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 8, pp. 3086–3099, 2020.

[40] W.-J. Yu, M. Shen, W.-N. Chen, Z.-H. Zhan, Y.-J. Gong, Y. Lin, O. Liu, and J. Zhang, "Differential evolution with two-level parameter adaptation," *IEEE Transactions on Cybernetics*, vol. 44, no. 7, pp. 1080–1099, 2014.

[41] K. Deb, "An efficient constraint handling method for genetic algorithms," *Comput. Method Appl. M.*, vol. 186, pp. 311–338, 01 2000.

[42] R. Agrawal, K. Deb, and R. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, 06 2000.

[43] K. Qiao, K. Yu, B. Qu, J. Liang, C. Yue, and X. Ban, "Feature extraction for recommendation of constrained multi-objective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2022.

[44] P. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 174–188, 2003.

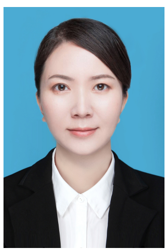
[45] L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume," *IEEE Transactions on Evolutionary Computation*, vol. 10, pp. 29 – 38, 03 2006.

[46] D. Veldhuizen and G. Lamont, "Evolutionary computation and convergence to a pareto front," *Late Breaking Papers at the Genetic Programming 1998 Conference*, 02 1999.

[47] E. Zitzler, J. Knowles, and L. Thiele, "Quality assessment of pareto set approximations," in *Multiobjective Optimization*. Springer, 2008, pp. 373–404.

Ruiqing Sun received a B.Sc. degree from Hunan University of Technology in network engineering in 2020. He is pursuing an M.Sc. degree in computer technology at Xiangtan University, Xiangtan, China.

His current research interests include multi-objective optimization and swarm intelligence.



Juan Zou received the B.Sc. degree in computer application technology, the M.Sc. degree in computer application technology, and the Ph.D. degree in applied mathematics from Xiangtan University, Xiangtan, China, in 2002, 2005, and 2014, respectively.

She is currently a Professor of Computational Intelligence at the School of Computer Science and School of Cyberspace Science, Xiangtan, China. She has over 50 publications and one monograph. Her current research interests include multiobjective optimization, swarm intelligence, artificial neural networks, and related applications.

Yuan Liu received the M.Sc. degree in computer science from the School of Computer Science and School of Cyberspace Science, Xiangtan University, Xiangtan, China, in 2017 and a Ph.D. degree from the College of Information Science and Engineering, Hunan University, Changsha, China, in 2021. Currently, he is a lecturer at the School of Computer Science and School of Cyberspace Science at Xiangtan University.

His research interests include evolutionary algorithms and machine learning.

Shangxiang Yang (M'00–SM'14) received the Ph.D. degree from Northeastern University, Shenyang, China, in 1999.

He is currently a Professor of Computational Intelligence and the Deputy Director of the Institute of Artificial Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester, U.K. He has over 370 publications with an H-index of 65 according to Google Scholar. His current research interests include evolutionary computation, swarm intelligence, artificial neural networks, data mining and data stream mining, and relevant real-world applications. Prof. Yang serves as an Associate Editor/Editorial Board Member for a number of international journals, such as the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION , IEEE TRANSACTIONS ON CYBERNETICS , Information Sciences, and CAAI Transactions on Intelligence Technology.

Jinhua Zheng received the B.Sc. degree in computer software and the M.Sc. degree in computer application from the Nanjing University of Science and Technology, Nanjing, China, in 1986 and 1989, respectively, and the Ph.D. degree in control theory and control engineering from Central South University, Changsha, China, in 2000. Since 2000, he has been a Professor at the School of Computer Science and School of Cyberspace Science, Xiangtan University, China.

He has over 140 research publications, including an academic monograph and four textbooks. His current research interests include evolutionary computation, multiobjective evolutionary optimization, robust optimization, and evolutionary algorithms in real-world applications.