Information Sciences 560 (2021) 68–91

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A multi-stage evolutionary algorithm for multi-objective optimization with complex constraints

Haiping Ma [a], Haoyu Wei [b], Ye Tian [a], Ran Cheng [c], Xingyi Zhang [b,]*

[a] *Key Laboratory of Intelligent Computing and Signal Processing of the Ministry of Education, Institutes of Physical Science and Information Technology, Anhui University, Hefei 230601, Anhui, China*
[b] *Key Laboratory of Intelligent Computing and Signal Processing of the Ministry of Education, School of Computer Science and Technology, Anhui University, Hefei 230601, Anhui, China*
[c] *Guangdong Provincial Key Laboratory of Brain-Inspired Intelligent Computation, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China*

A R T I C L E I N F O

Article history:
Received 30 August 2020
Received in revised form 7 January 2021
Accepted 11 January 2021
Available online 2 February 2021

Keywords:
Evolutionary algorithm
Constraint-handling priority
Constrained multi-objective optimization

A B S T R A C T

Constrained multi-objective optimization problems (CMOPs) are difficult to handle because objectives and constraints need to be considered simultaneously, especially when the constraints are extremely complex. Some recent algorithms work well when dealing with CMOPs with a simple feasible region; however, the effectiveness of most algorithms degrades considerably for CMOPs with complex feasible regions. To address this issue, this paper proposes a multi-stage evolutionary algorithm, where constraints are added one after the other and handled in different stages of evolution. Specifically, in the early stages, the algorithm only considers a small number of constraints, which can make the population efficiently converge to the potential feasible region with good diversity. As the algorithm moves to the later stages, more constraints are considered to search the optimal solutions based on the solutions obtained in the previous stages. Furthermore, a strategy for sorting the constraint-handling priority according to the impact on the unconstrained Pareto front is proposed, which can accelerate the convergence of the algorithm. Experimental results on five benchmark suites and three real-world applications showed that the proposed algorithm outperforms several state-of-the-art constraint multi-objective evolutionary algorithms when dealing with CMOPs, especially for problems with complex constraints.

1. Introduction

Constrained multi-objective optimization problems (CMOPs) widely exist in many real-world applications, such as optimal software product selection [44], knapsack problems [3], and capacitated arc routing problems [1]. All these problems need to optimize some conflicting objectives and meet a set of constraints simultaneously. Without loss of generality, a CMOP can be defined as

* Corresponding author.
E-mail addresses: hpma@ahu.edu.cn (H. Ma), xyzhanghust@gmail.com (X. Zhang).

https://doi.org/10.1016/j.ins.2021.01.029
0020-0255/

$$\begin{aligned} \text{Minimize} \quad & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_m(\mathbf{x})) \\ \text{subject to} \quad & \mathbf{x} \in \Omega \\ & g_i(\mathbf{x}) \leqslant 0, \quad i = 1, \cdots, p \\ & h_j(\mathbf{x}) = 0, \quad j = 1, \cdots, q \end{aligned} \tag{1}$$

where $\mathbf{x} = (x_1, \cdots, x_d) \in \Omega$ is a solution consisting of d decision variables, $\Omega \subseteq \mathbb{R}^d$ is the decision space, $\mathbf{F} : \Omega \rightarrow \mathbb{R}^m$ consists of m objectives, $g_i(\mathbf{x}) \leqslant 0$ is the i-th inequality constraint, and $h_j(\mathbf{x}) = 0$ is the j-th equality constraint. The overall constraint violation degree of a solution $\mathbf{x}$ adopted in this paper is defined as

$$C(\mathbf{x}) = \sum_{k=1}^{p} \max\{0, g_k(\mathbf{x})\} + \sum_{\ell=1}^{q} |h_\ell(\mathbf{x})| \tag{2}$$

A solution $\mathbf{x}$ is feasible if $C(\mathbf{x}) = 0$; otherwise, it is infeasible. The feasible region is composed of all solutions that satisfy all the constraints.

Over the last two decades, a variety of promising evolutionary algorithms have been developed, especially for solving multi-objective optimization problems [6]. These multi-objective evolutionary algorithms (MOEAs) can be roughly divided into four categories, namely evolutionary algorithms based on Pareto dominance, such as NSGA-II [11] and SPEA2 [50]; evolutionary algorithms based on decomposition, such as MOEA/D [46] and MOEA/D-AWA [34]; indicator-based evolutionary algorithms, such as IBEA [49] and AR-MOEA [36]; and mixed evolutionary algorithms, such as SRA [24] and Two_Arch [40]. Based on these four categories, many algorithms have been developed to solve various multi-objective optimization problems, i.e., large-scale multi-objective optimization problems [4,17] and many-objective optimization problems [10,36]. In recent years, constrained multi-objective optimization has attracted wide attention from researchers. An increasing number of MOEAs have been developed to solve such problems (MOEAs for solving CMOPs are thus named CMOEAs), such as NSGA-II-CDP [11] based on the constrained dominance principle, C-TAEA [25] based on the two-archive strategy, and ToP [27] based on the two-stage search process. Although these MOEAs achieve competitive performance in solving CMOPs, they suffer from considerable performance deterioration on some CMOPs [27,12], where the constrained landscape is significantly complex, such as the discrete feasible region with a huge infeasible barrier (e.g., LIR-CMOP10 [12]) and only several narrow feasible regions (e.g., MW11 [28]), as shown in Fig. 1. Generally, these complex constrained landscapes are composed of more than one constraint. The CMOPs will become simpler and easier to be handled if we consider these constraints one by one and handle them in different stages of evolution since the constrained landscape with a small number of constraints will not be so complicated.

Following this idea, this paper proposes a multi-stage evolutionary algorithm for solving CMOPs. In contrast to most existing algorithms that regard all constraints as a whole and deal with them together, the proposed algorithm divides the constraint-handling process into multiple stages and deals with the constraints one stage by stage. The main contributions of this work are summarized as follows:

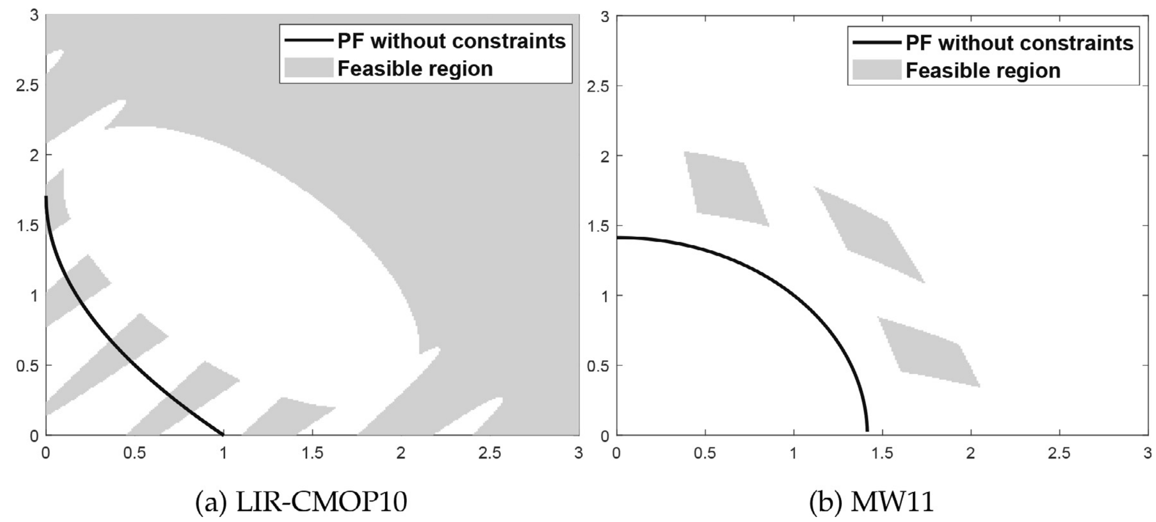

Fig. 1. Feasible regions of LIR-CMOP10 and MW11.

1) A multi-stage CMOEA (MSCMO) is proposed for solving CMOPs with complex constraints. In the proposed algorithm, the constraints are added one by one and handled in different stages of evolution. In the early stages, the proposed algorithm only deals with a small number of constraints, which can make the population efficiently converge to the potential feasible region with good diversity. With each stage, more constraints are considered and can be handled more easily based on the solutions obtained in the previous stages.
2) A strategy for sorting the constraint-handling priority according to the impact on the unconstrained Pareto front is suggested in the proposed MSCMO, which is used to determine the constraints to be handled at each stage. Experimental results on benchmark CMOPs showed that the proposed constraint-handling priority can accelerate the convergence of the algorithm.
3) The results of a series of experiments on benchmark CMOPs and real-world applications show that our algorithm is very competitive in comparison with some state-of-the-art CMOEAs, especially on the CMOPs with complex constraints.

The remainder of this paper is organized as follows. In Section 2, we first introduce existing CMOEAs with different constraint-handling techniques for CMOPs, and then we discuss the motivation for this work. In Section 3, the proposed algorithm is explained in detail. Experimental details on the benchmark CMOPs and real-world applications are given in Section 4. Finally, we present the conclusions and future work in Section 5.

2. Related works and motivation

2.1. Existing CMOEAs

In general, existing CMOEAs can be classified into four categories according to the constraint-handling technology that they adopt.

The first category is the penalty function approach. The main idea is to construct a penalty term based on the degree of individual constraint violation. A penalty fitness function $fitness(\mathbf{x})$ is first constructed by adding a penalty term $p(\mathbf{x})$ to the objective function $f(\mathbf{x})$. Then, $fitness(\mathbf{x})$ is used to evaluate the individuals. The penalty function method can be classified as follows: death penalty function [30], static penalty function [18], dynamic penalty function [15], and adaptive penalty function [26,35,19]. Among them, the adaptive penalty function has better performance because it can use the feedback information in the search process to dynamically adjust the parameters. In [26], the penalty function was combined with the rough set theory, where the penalty coefficient was adjusted adaptively according to the individual's constraint violation. In [35], a novel fuzzy rule-based penalty function approach was proposed for solving constrained optimization problems that use a fuzzy inference system to adaptively determine the penalty value. In [19], preferences were reformulated into constraints for evolutionary many-objective optimization, and a constrained dominance principle based on the penalty function approach was used for handling constraints.

The second category separates objectives and constraints. Specifically, this category can be divided into three groups: feasibility rule, stochastic ranking method, and ϵ-constraint method. For the feasibility rule, feasible solutions are superior to infeasible solutions. The feasibility rule CDP proposed by Deb in [7] is one of the most popular constraint-handling techniques, where one solution $\mathbf{x}$ is regarded as superior to another solution $\mathbf{y}$ under the following three conditions:

- $\mathbf{x}$ is feasible, whereas $\mathbf{y}$ is infeasible.
- both $\mathbf{x}$ and $\mathbf{y}$ are feasible, but $\mathbf{x}$ has a better objective value than $\mathbf{y}$.
- both $\mathbf{x}$ and $\mathbf{y}$ are infeasible, but the degree of constraint violation of $\mathbf{x}$ is smaller than that of $\mathbf{y}$.

In [29], a new fitness function of each solution is defined as the weighted sum of two rankings: one is the solution's ranking based on CDP and the other is the solution's ranking based on Pareto dominance. In [45], the probability parameter *pf* in the stochastic ranking method could change dynamically as the population evolves in the differential evolution (DE) framework. The ϵ-constraint method provides some relaxation to infeasible solutions. For an individual whose constraint violation degree is smaller than ϵ, the solution is regarded as feasible. On the contrary, for an individual whose constraint violation degree is larger than ϵ, the solution is regarded as infeasible. The key to the ϵ-constraint method is the way in which the value of ϵ is set. In [12], the ϵ value was adjusted dynamically according to the ratio of the feasible solutions to the total solutions in the current population, which works well for CMOPs with large infeasible regions. In [23], the original constrained multi/many-objective problem was transformed to a dynamic constrained multi/many-objective problem by setting different ϵ values at different time in order to balance convergence, diversity and feasibility. In [48], the ϵ value gradually decreased if the feasible proportion of the population was smaller than the threshold α. Otherwise, it was set to the maximum constraint violation degree among the individuals of the population, and a detect-and-escape strategy was proposed to help the population escape from the local optimum.

The third category is the multi-objectivization approach. The idea is to treat the constraints as additional objectives and to transform the constrained optimization problem into a multi-objective optimization problem. In [2], a feasible-ratio control technique was proposed and combined with multi-objective method, which makes a good balance in searching feasible space and infeasible space. In [47], a tri-goal evolution framework was proposed that designs three indicators for convergence, diversity, and feasibility to solve constrained many-objective problems.

The last category is the hybridization approach. This approach combines a variety of different constraint-handling techniques to deal with the constrained problem. In [42], an adaptive trade-off model was proposed in which the population evolution process is divided into three situations, namely only feasible solutions in the population, only infeasible solutions in the population, and both feasible and infeasible solutions in the population. In [42], different individual comparison and selection criteria were designed for different evolutionary situations. In [22], a new type of selection evolution strategy was designed for environmental selection. First, individuals with large constraint violations are removed from the entire parent and offspring populations, and all non-inferior individuals are selected to obtain the next generation. Then, if the number of non-inferior individuals is smaller than the population size, the penalty function method is used to select the remaining individuals.

In recent years, some multiple-stage or multi-population CMOEAs have also been developed to solve CMOPs. In [31], a two-stage collaborative evolution framework was proposed for solving CMOPs with deceptive constraints. The first phase adopts two subpopulations: one to explore feasible regions and the other to explore the entire space. The second stage uses the two subpopulations as the initial population to find Pareto-optimal solutions. The two stages are switchable based on the information found in the evolutionary process. In [14], the proposed PPS framework divides the search process into two different stages: push and pull search stages. In the push stage, a multi-objective evolutionary algorithm is used to explore the search space without considering any constraints, which can help to get across infeasible regions very quickly and to approach the unconstrained Pareto front. In the pull stage, the constraints are considered to help the population pull back to the real Pareto front. In [27], a two-stage framework called ToP was also proposed. In ToP, the first phase only considers a single objective and all constraints, and thus the original CMOPs are transformed into constrained single-objective optimization problems to find the promising feasible area. Then, in the second phase, all the constraints and objectives are considered to obtain the final solutions. In [39], a two-stage evolutionary algorithm, which adjusts the fitness evaluation strategies during the evolutionary process to adaptively balance objective optimization and constraint satisfaction was proposed. In the first stage, the objectives are given the same priority to constraints for reaching all the feasible regions. In the second stage, the objectives are given a lower priority than the constraints for diversifying the solutions along the feasible boundaries. In [41], a cooperative DE framework, CCMODE, was proposed for constrained multi-objective optimization, including M subpopulations for constrained single-objective optimization and an archive population for constrained M-objective optimization. In [25], a two-archive evolutionary algorithm, C-TAEA, was proposed for constrained multi-objective optimization. The convergence-oriented archive (CA) is evolved to optimize the constraints and objectives, and aims to push the population toward the Pareto front, whereas the diversity-oriented archive (DA) is evolved to optimize only the objectives and aims to explore areas underexploited by the CA, including the infeasible regions. CCMO [38] is also an evolutionary algorithm evolving two populations. The proposed framework evolves one population to solve the original CMOP to maintain feasibility as well as convergence, and evolves the other population to solve a helper problem derived from the original one, aiming to maintain diversity.

The detailed grouping of these algorithms and their representative algorithms are summarized in Table 1.

2.2. Motivation of this work

Although many constrained multi-objective evolutionary algorithms have been developed to solve CMOPs and have achieved good effectiveness in solving some problems, most of them face difficulties in handling CMOPs with complex constraints. Usually, most of the existing constrained optimization problems have more than one constraint [27,12], and these different constraints collaboratively form a complex constraint landscape. For example, the feasible region can be narrow and discrete, or there can be a huge infeasible barrier.

Table 1
Constrained evolutionary algorithms.

Constrained evolutionary algorithms	
Category	Representative Algorithms
Penalty function approach	RPGA, fpenalty
Feasibility rule	NSGA-II-CDP, TOR-NSGA-II
Stochastic sorting method	DSS-MDE
ϵ Constraint method	MOEA/D-DAE, MOEA/D-IEpsilon
	DCNSGA-III,
Multi-objectivization approach	FRC–CEA, TiGE-2
Hybridization approach	ATM, NSES
Multiple-stage or multi-population method	IDW-M2M-CDP, CCMODE
	CCMO, C-TAEA
	ToP, PPS, CMOEA-MS

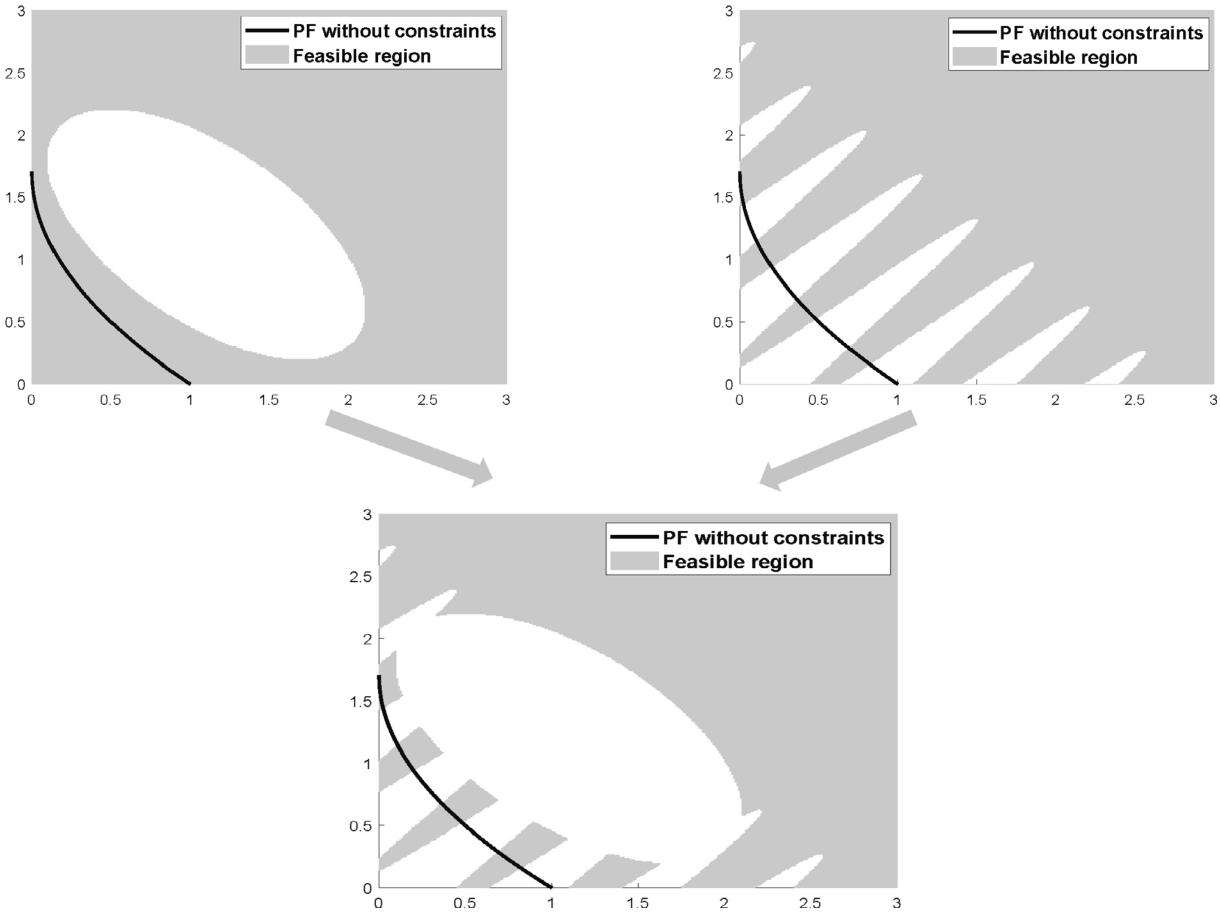

Fig. 2. Feasible region formed by the first constraint and the second constraint of LIR-CMOP10.

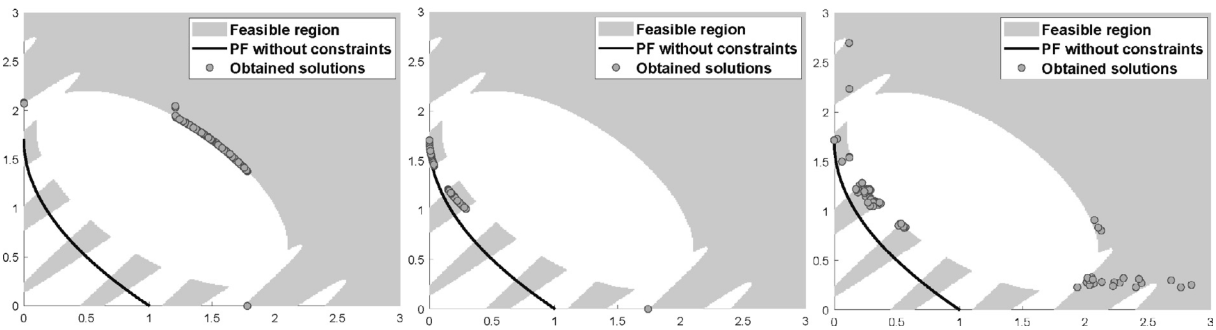

Fig. 3. The final populations obtained by NSGA-II-CDP, C-TAEA, and ToP on LIR-CMOP10.

Taking the problem LIR-CMOP10 [12] in Fig. 2 as an illustrative example, we can see that the first constraint of LIR-CMOP10 forms a huge infeasible barrier in front of the unconstrained Pareto front. The second constraint forms many discrete strip-infeasible regions that split the unconstrained Pareto front into many small segments. When the two constraints are considered separately, the feasible region is relatively easy to handle. However, when they are considered as a whole, these two constraints together form a particularly complex constrained landscape, which is a stiff challenge to most existing

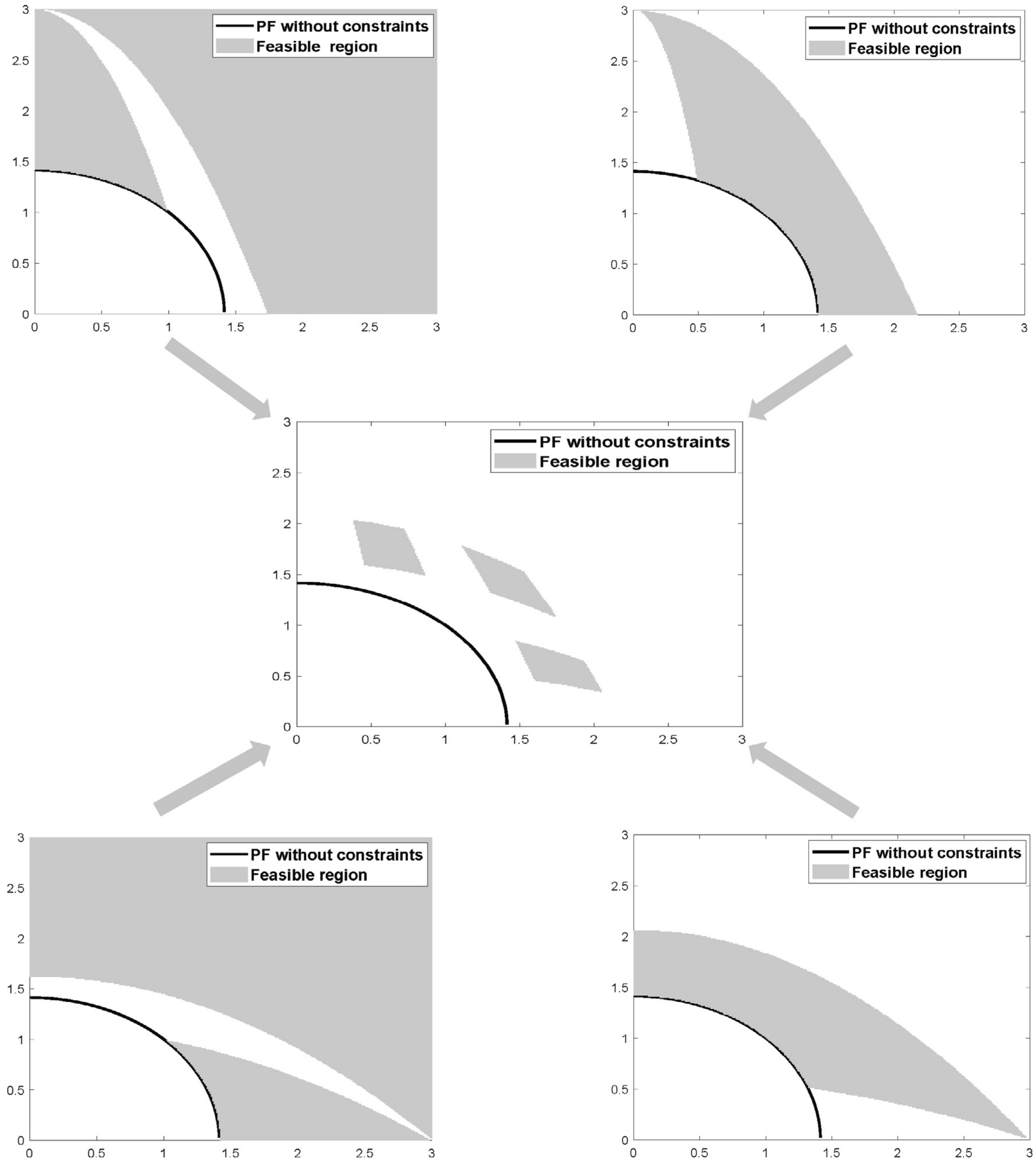

Fig. 4. Feasible region formed by the four constraints of MW11.

CMOEAs. The final populations obtained by NSGA-II-CDP [11], C-TAEA [25], and ToP [27] on the LIR-CMOP10 problem are shown in Fig. 3. We can see from the figure that the solutions obtained by NSGA-II-CDP are completely stuck at the boundary of the infeasible barrier, whereas the solutions obtained by C-TAEA and ToP can only cover a small part of the Pareto front of LIR-CMOP10 because of the complex feasible region landscape.

Taking the problem MW11 [28] in Fig. 4 as another illustrative example, we can see that the four constraints can be divided into two groups, where each group contains two constraints. The feasible region formed by the two constraints in each group is symmetrical to each other, and the four constraints form three narrow and discrete diamond-shaped feasible regions. The final solutions obtained by NSGA-II-CDP, C-TAEA, and ToP on MW11 are shown in Fig. 5. It can be seen from the figure that both NSGA-II-CDP and ToP can converge only to one part of the feasible region. Although C-TAEA can converge to all feasible regions, the convergence and diversity can be further improved.

Based on the above results, we have two important observations. First, a narrow feasible region will cause difficulties in finding feasible solutions. Second, a discrete feasible region can lead to local optima. However, if the complex constraints mentioned previously are treated separately, they are no longer considered to be complex. Based on this idea, this paper considers the constraints one by one and divides the process of solving CMOPs into multiple stages, as shown in Figs. 6 and 7. Specifically, the proposed algorithm starts with dealing with only one constraint and then considers more constraints gradually with each stage. In this way, the original complex CMOPs can be easily solved. In the next section, the procedure for the proposed MSCMO is described.

3. The proposed algorithm

3.1. Overview of the proposed MSCMO

Algorithm 1. Procedure of the proposed MSCMO

Input: n (population size), λ (stage conversion threshold parameter), and $Archive$ (external archive for saving solutions)
Output: P (final population)

//**Determination of theConstraint Handling Priority**
$CHP = Constraint\_Priority\_Determinate(n, \lambda)$;//Algorithm 2
//**Initialization**
$P \leftarrow RandomInitialization(n)$;
$currentCons \leftarrow \emptyset$;//$currentCons$ saves the set of constraints handled at the current stage
$F \leftarrow$ Calculate the fitness of the solutions in P;
while *termination criterion not fulfilled* **do**
  //**Reproduction**
  $P' \leftarrow$ Use binary tournament selection to select n parents from P based on the fitness;
  $Offspring \leftarrow Variation(P')$;
  $P \leftarrow P \bigcup Offspring$;
  //**Update external archive**
  if $|currentConstraint| > 0$ **then**
    $Archive = updateArchive(Offspring \bigcup Archive, n, currentCons)$;
  $F \leftarrow$ Calculate the fitness of population P;
  //**Environmental selection**
  $P \leftarrow$ Use the fitness and truncation method to select n solutions from P;
  //**Determine whether to enter the next stage**
  if $|currentConstraint| = 0$ **then**
    $ifr \leftarrow$ The infeasible proportion of the population on the constraint with the highest priority;
    if $ifr > 0$ **then**
      //Enter the next stage and add the constraint
      $Out\_Queue(CHP, constraint)$;
      $currentCons = currentCons \bigcup constraint$;
      $F \leftarrow$ Recalculate the fitness of population P ;
  else
    $maxChange \leftarrow$ Calculate the population change at each objective value and take the largest value;
    if $maxChange \leq \lambda$ **then**
      //Enter the next stage and add the constraint
      $Out\_Queue(CHP, constraint)$;
      $currentCons = currentCons \bigcup constraint$;
      $F \leftarrow$ Recalculate the fitness of population P ;
      //Construct the initial population of the next stage
      if $|Archive| = n$ **then**
        $P = Archive$;
      else
        $P = Pop\_Supplement(P \bigcup Archive, n)$;//Algorithm **??**
return P;

Algorithm 2, Constraint_Priority_Determinate

Input: n (population size), λ (stage conversion threshold parameter)

Output: CHP (constraint-handling priority)

$P_1 \leftarrow RandomInitialization(n)$;
$CHP \leftarrow Init\_Queue()$;
$F_1 \leftarrow$ Calculate the fitness of the solutions in P_1;
while *termination criterion not fulfilled* **do**
  $P_1' \leftarrow$ Use binary tournament selection to select n parents from P_1 based on the fitness;
  $P_1 \leftarrow P1 \bigcup Variation(P1')$;
  $F_1 \leftarrow$ Calculate the fitness of the solutions in P_1;
  $P_1 \leftarrow$ Use the fitness and truncation method to select n solutions from P_1;
  $maxChange \leftarrow$ Calculate the population change at each objective value and take the largest value;
  if $maxChange \leq \lambda$ **then**
    $ifr \leftarrow$ Calculate the infeasible rate of the population on each constraint;
    $CHP \leftarrow$ Sort the constraints according to the descending order of the infeasible rate;`//Save the constraint priority sequence into` CHP;
    break;
return CHP;

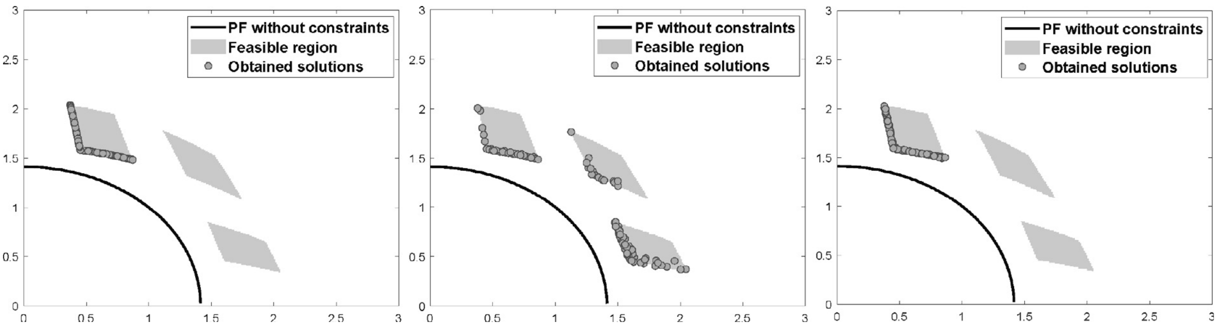

Fig. 5. The final populations obtained by NSGA-II-CDP, C-TAEA, and ToP on MW11.

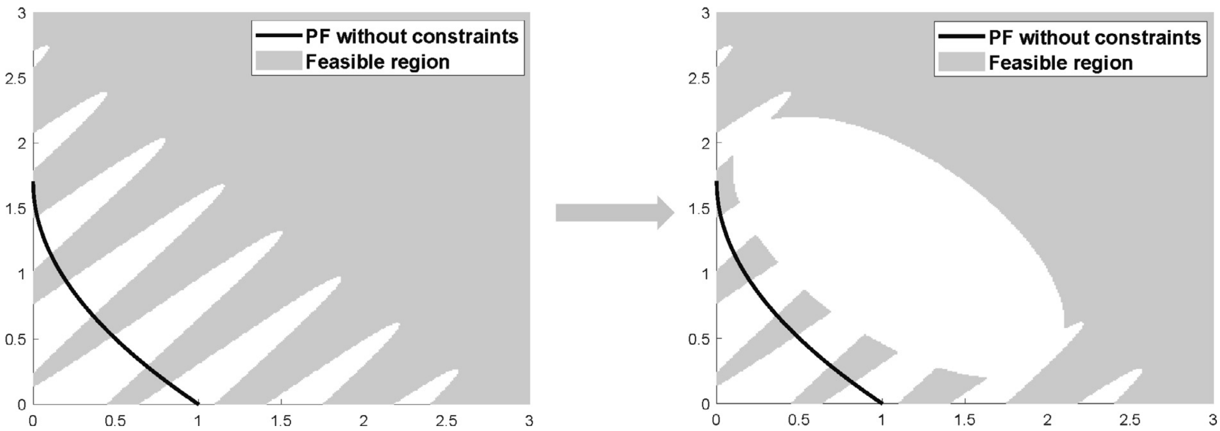

Fig. 6. An example illustrating the procedure where constraints are added one at a time and handled in different stages of the proposed MSCMO on LIR-CMOP10.

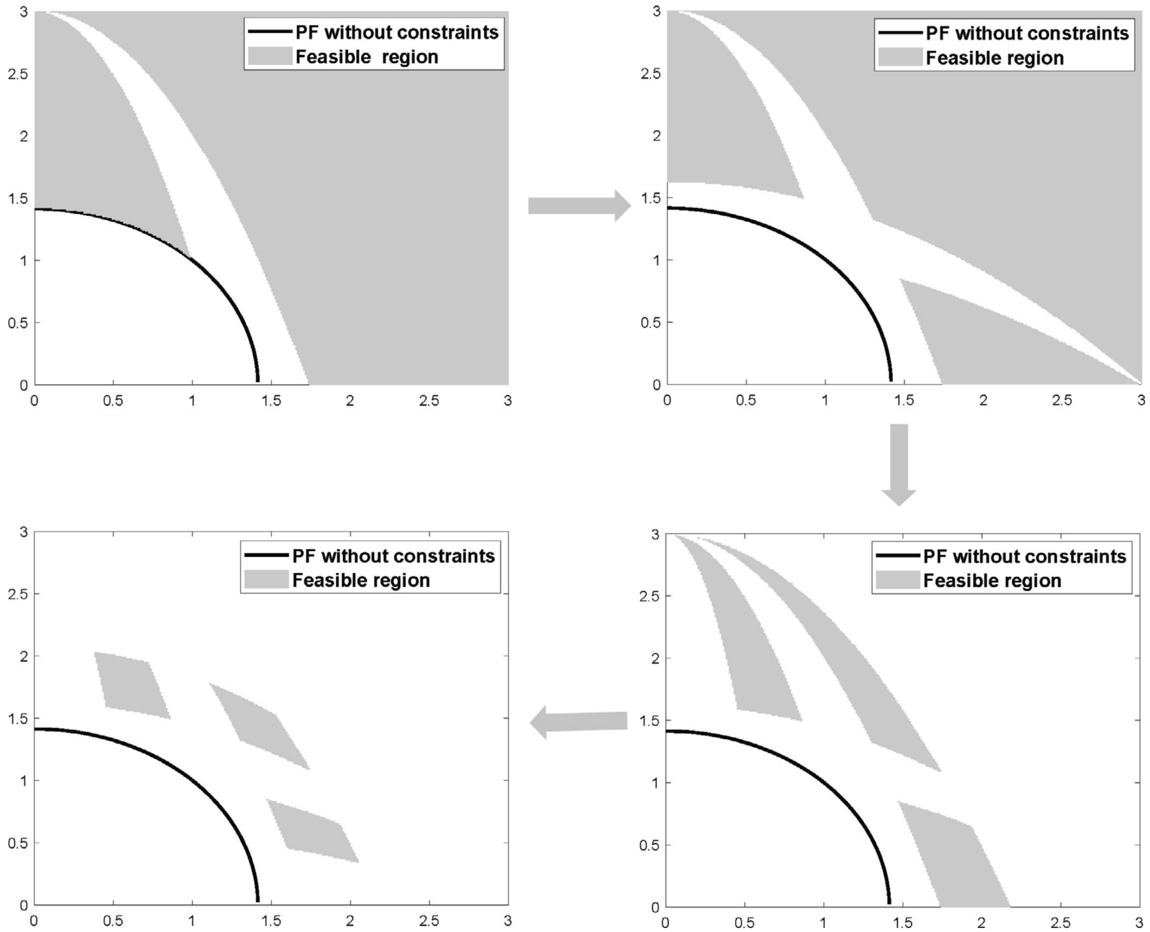

Fig. 7. An example illustrating the procedure where constraints are added one at a time and handled in different stages of the proposed MSCMO on MW11.

The general flow of the proposed MSCMO is presented in Fig. 8. First, the constraint-handling priority is determined, which is used to decide the constraints to be handled at each stage. Then, the algorithm initializes a population *P* with *n* individuals randomly, which is evolved generation by generation until the maximum evaluations are reached. At each generation, the algorithm repeats the following operations: (1) reproduction, (2) external archive update, (3) environmental selection, and (4) checking of the conversion condition from one stage to the next. This multi-stage constraint-handling technology suggested in the proposed MSCMO can considerably reduce the complexity of feasible regions, thus significantly improving the performance of MOEAs in solving CMOPs with complex feasible regions. Algorithm 1 provides the pseudocode of the proposed MSCMO. In lines 1–4, the constraint-handling priority *CHP* is first determined. Then, a population *P* with size *n* is initialized randomly, and the fitness *F* of the population *P* is calculated successively. The main loop of the proposed MSCMO is shown in lines 5–28. Especially, for each generation, the reproduction is shown in lines 6–8, which uses the binary

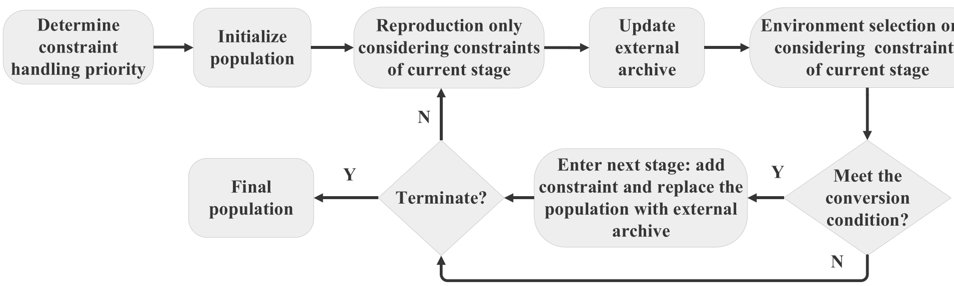

Fig. 8. Flowchart of the proposed MSCMO.

tournament selection to select n parents and generates n offspring. The external archive update is shown in lines 9–10, where the non-dominated feasible solutions that satisfy the constraints of the current stage are saved in the external archive. Then, the fitness and the truncation method are used to select n solutions in the environment selection shown in line 12 for the next generation. The determination of the conversion condition from one stage to the next stage and the corresponding operations are shown in lines 13–28. Specifically, if the conversion condition is satisfied, the algorithm enters the next stage in which new constraints are added according to the constraint-handling priority, and the population is replaced by the solutions in the external archive obtained in the previous stage. Otherwise, the evolution will remain in the current stage. The loop is executed until the maximum evaluations are reached, and then the population P is returned as the final result, as shown in line 29.

To determine the constraint-handling priority, Algorithm 2 describes the procedure in detail. In constrained multi-objective optimization, the true Pareto front of the CMOPs is determined by the unconstrained Pareto front and the feasible regions formed by the constraints together. According to this idea, we first randomly initialize a population P_1 and let the population P_1 evolve to the approximate unconstrained Pareto front. Then, we calculate the infeasible rate of P_1 on each constraint and sort the constraints according to their infeasible rates in descending order, which implies that the constraint having more impact on the unconstrained Pareto front is given higher priority. More specifically, to improve the efficiency of constraint handling, we combine these constraints whose infeasible rate is 0 and handle them together in one stage. To better understand the determination procedure of the constraint priority, we provide an illustrative example in Fig. 9. It can be seen from the figure that, after the operations are sorted and combined, the constraints are handled in the order of $\{5\}, \{2\}, \{1\}, \{4\}, \{3,6\}$, where i denotes the i-th constraint of the CMOPs.

3.2. Key components in multi-stage constraint-handling technology

In this subsection, we present three critical components in the suggested multi-stage constraint-handling technology: population reinitialization at each stage, fitness evaluation at each stage, and conversion conditions from the current stage to the next stage.

Population reinitialization at each stage. In the process of multi-stage evolution, we need to reinitialize the population P at the beginning of the current stage based on the solutions stored in the archive, which are obtained in the previous stages. Note that, at each stage, the non-dominated solutions that satisfy the constraints at that stage are stored in the *Archive*, and a truncation operation in SPEA2 [50] is carried out if the number of feasible non-dominated solutions is larger than n, where n is the size of the archive. Based on the *Archive*, the population P at each stage is reinitialized as follows: If the number of solutions in the *Archive* is equal to n, the population P is reinitialized by the solutions stored in the *Archive* directly. Otherwise, the operation *Pop_Supplement* is executed on the *Archive*, and the population P at the end of the previous stage is used to construct a new population *pop* with n individuals, and then *pop* is used to reinitialize the population P. The operation *Pop_Supplement* is similar to the update operation of DA in C-TAEA. More details can be seen in the original paper [25].

Fitness evaluation at each stage. We adopted the method in SPEA2 [50] to calculate the fitness at each stage in the proposed MSCMO because the fitness calculation method in SPEA2 considers the dominance and distance of individuals, which is beneficial to both convergence and diversity. Moreover, to make the algorithm achieve better performance, we used different constraint-handling mechanisms at different stages. To be specific, in the early stages of evolution, a multi-objective optimization-based constraint-handling mechanism was used to deal with the constraints to obtain solutions with a good diversity, where the constraint violation degree at the current stage was used as an extra objective function, and the method in SPEA2 was used to calculate the fitness value. In the last stage, considering all the constraints, we used the feasibility rule-based constraint-handling mechanism to accelerate the convergence.

Conversion conditions from the current stage to the next stage. In the proposed MSCMO, if the current stage is the first stage, the conversion condition from the current stage moving to the next stage is set to the infeasible rate of population P on the first constraint if the constraint-handling priority is larger than 0. Otherwise, the evolution from the current stage moving to the next stage needs to satisfy the following condition: The change in the average value of population P on each objective is less than a given threshold λ within g generations. The average value of population P on the i-th objective was calculated as follows. First, we used Formula (3) to normalize the objective value of each individual in population P. Then,

Fig. 9. An example of determining the constraint priority.

we used Formula (4) to calculate the average value of population P on each objective. In this paper, the threshold λ was set to 0.01 and, g was set to 100.

$$f'_i(\mathbf{x}) = \frac{f_i(\mathbf{x}) - f_i^{min}}{f_i^{max} - f_i^{min}}, \tag{3}$$

$$\bar{f}'_i = \frac{\sum_{\mathbf{x}\in P} f'_i(\mathbf{x})}{|P|} \tag{4}$$

where $\mathbf{x}$ is a solution of population P; $f_i(\mathbf{x})$ is the original value of the i-th objective of solution $\mathbf{x}$; $f'_i(\mathbf{x})$ is the i-th objective value of solution $\mathbf{x}$ after normalization; f_i^{max} and f_i^{min} are the maximum and minimum values, respectively, of the i-th objective in population P; $\bar{f}'_i$ is the average value of population P on the i-th objective; and $|P|$ is the number of individuals in population P.

3.3. Analysis of the proposed MSCMO

To show the superiority of the proposed MSCMO in dealing with complex constraint problems, we tested its performance on the LIR-CMOP10 and MW11 problems, which were introduced in subSection 2.2. The constraint-handling priority is {1}, {3}, {2}, and {4} for MW11 and {2} and {1} for LIR-CMOP10. For the MW11 problem, constraints 1 and 3 (same for constraints 2 and 4) are symmetrical; thus, their orders in the constraint-handling priority can be interchanged. The final solutions obtained at each stage for the two problems are shown in Figs. 10 and 11. Specifically, the first stage ends when some solutions violate the first constraint of the constraint-handling priority, as shown in Figs. 10(a) and 11(a). Then, the handled constraints are added one by one, and the final feasible non-dominated solutions at each stage are shown in Fig. 10(b)–(e) for MW11 and in Fig. 11(b)–(c) for LIR-CMOP10. Finally, the population begins to strengthen the convergence by the feasibility rule in the last stage, and the final solutions are obtained, as shown in Figs. 10(f) and 11(d). It can be seen from the figure that the solutions can find all the current feasible regions with good diversity at the end of each stage and converge to the global optimum in the last stage. The proposed MSCMO achieved excellent performance and showed stronger competitiveness than those of the three state-of-the-art methods a, namely NSGA-II-CDP, C-TAEA, and ToP, on the two CMOPs with complex constraints.

In addition, to highlight the advantages of the proposed multi-stage strategy, we compared it with two existing two-stage CMOEAs, i.e., PPS [14] and ToP [27]. The MW5 [28] problem was used as the test problem, where the optimal solution is only

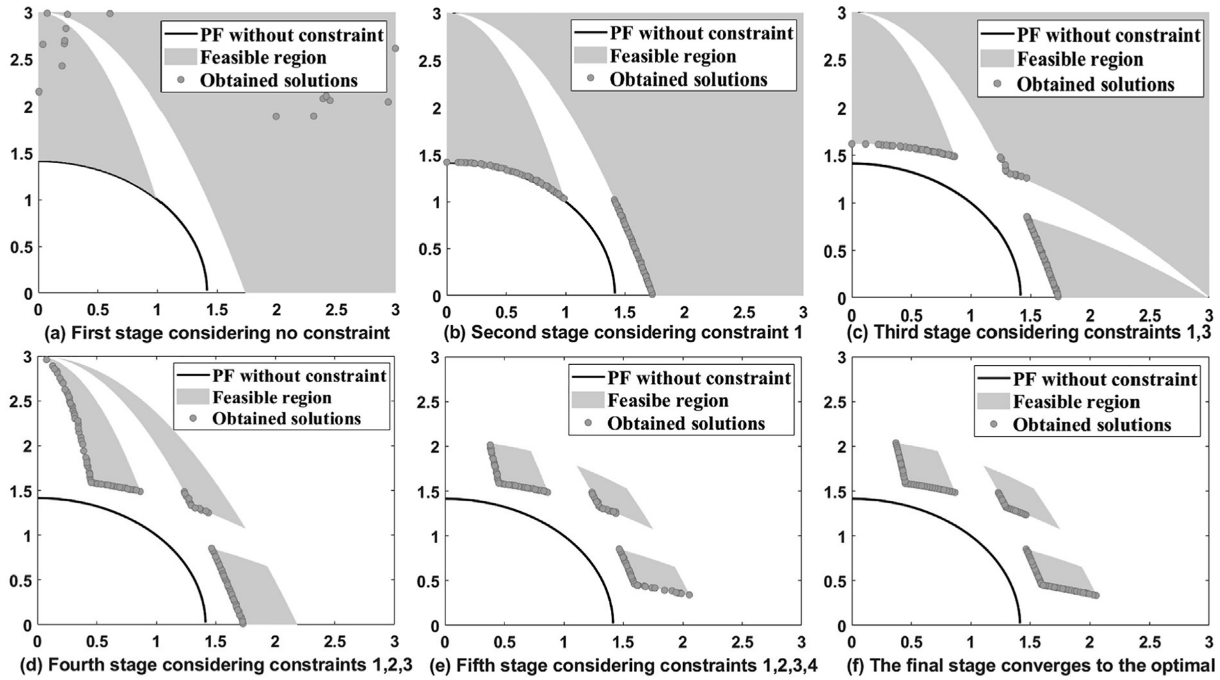

Fig. 10. Populations obtained by the proposed MSCMO at different stages on MW11.

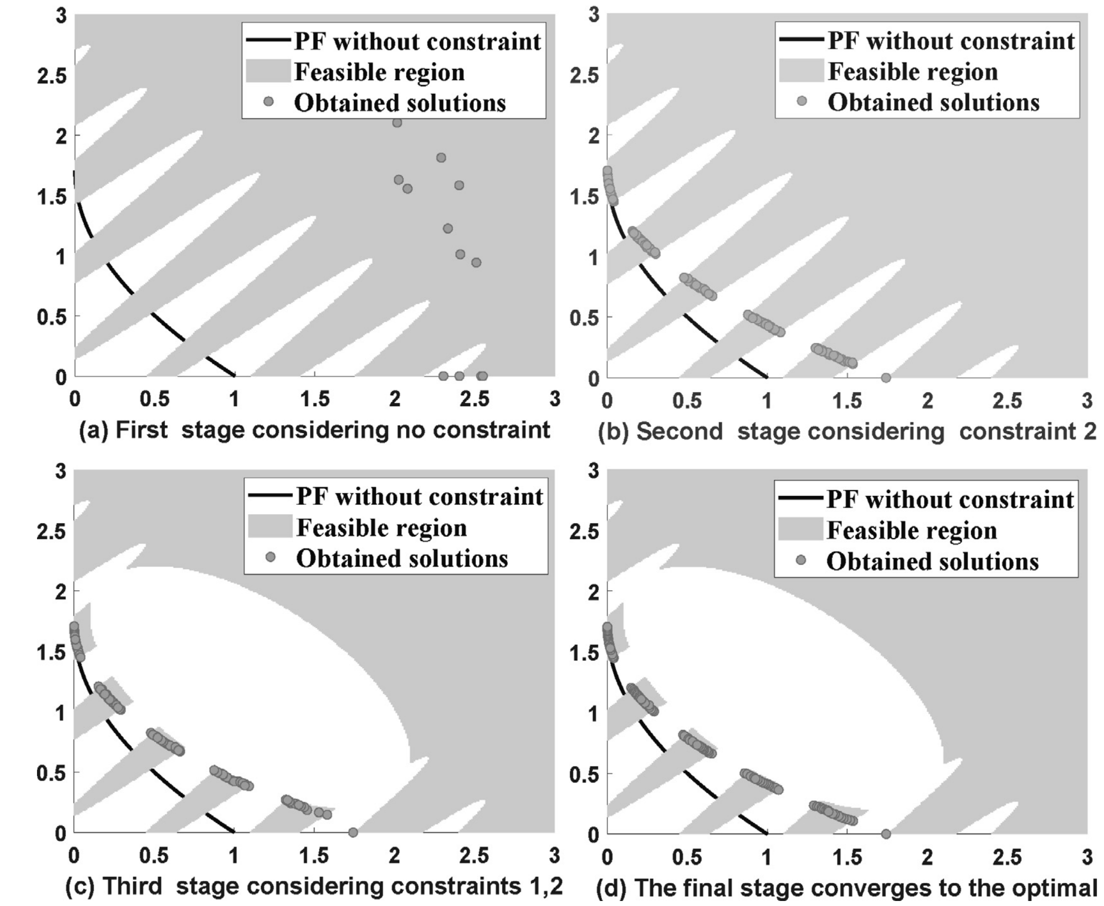

Fig. 11. Populations obtained by the proposed MSCMO at different stages on LIR-CMOP10.

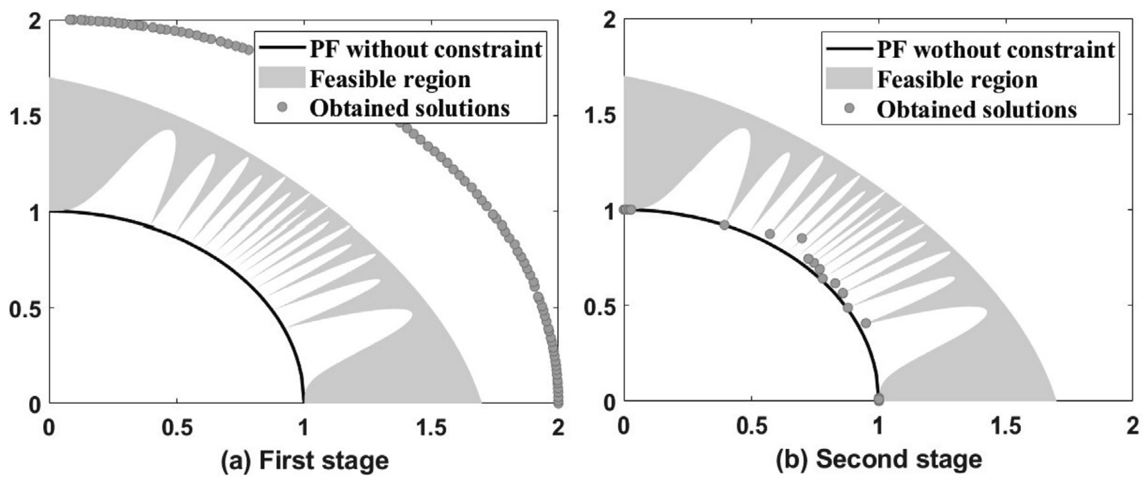

Fig. 12. Populations obtained by PPS at different stages on MW5.

a few points on the unconstrained Pareto front, and it is very difficult to find the global optimum. The solutions obtained by the three algorithms at different stages are shown in Figs. 12–14. PPS pushes the population to the approximate unconstrained Pareto front at the first stage and pulls the population to the true Pareto front at the second stage. The solutions obtained by PPS at the first stage are shown in Fig. 12(a). Then, all constraints are considered to evolve the population at the second stage, and the solutions are obtained as shown in Fig. 12(b). From Fig. 12(b), we can observe that PPS only finds

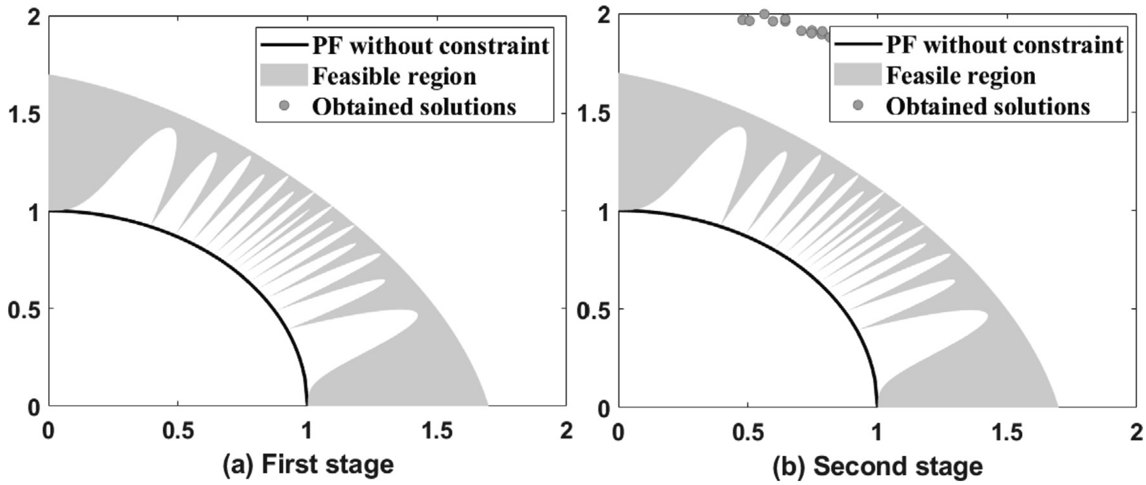

Fig. 13. Populations obtained by ToP at different stages on MW5.

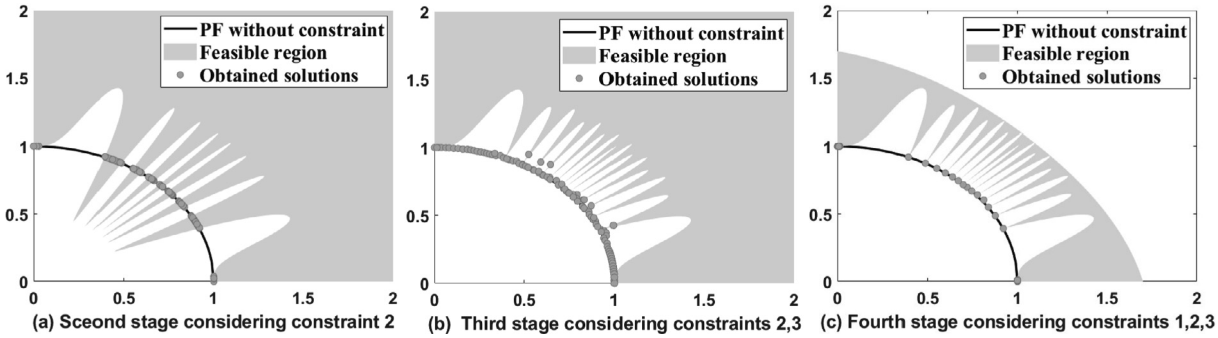

Fig. 14. Populations obtained by the proposed MSCMO at different stages on MW5.

part of the optimal feasible solutions. For ToP, it solves a single objective with all constraint problems at the first stage and then solves all the objectives and constraints at the second stage. Owing to the complexity of the constraints, the feasible region cannot be found at the first stage of ToP shown in Fig. 13(a). Obviously, Top cannot obtain a feasible solution at the second stage either, as shown in Fig. 13(b). The proposed MSCMO divides the evolution process into multiple stages

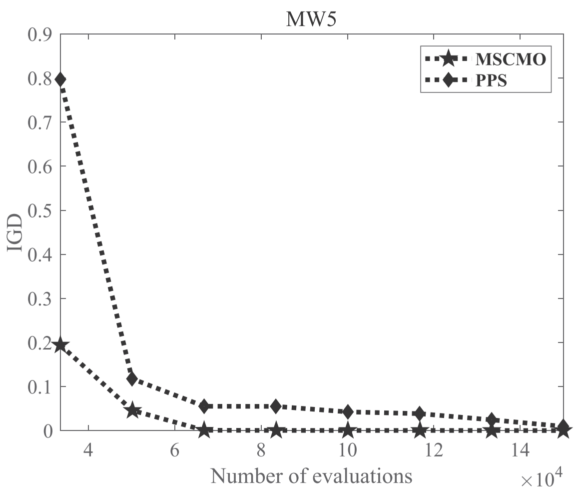

Fig. 15. IGD value curves of PPS and the proposed MSCMO on MW5.

according to the constraint priority. First, as shown in Fig. 14(a), only one constraint is considered, the feasible region is larger, and the proposed MSCMO can find many individuals that meet the constraints of the current stage. Then, in the later stage shown in Fig. 14(b), owing to another constraint being added, the feasible region becomes extremely complex, and the optimal feasible region becomes several points on the unconstrained Pareto front. However, the proposed MSCMO can still find all feasible regions because it is based on the solutions with a great diversity obtained in the previous stage. Finally, in the last stage, considering all the constraints, the convergence is strengthened through the feasibility rule, and all solutions can converge to the global optimum, as shown in Fig. 14(c). The changes in the inverted generational distance (IGD) values of the PPS and the proposed MSCMO during the evolution process on the MW5 problem are shown in Fig. 15. Note that there is no IGD curve for ToP because no feasible solution can be found during the entire evolution of ToP. We found that the proposed MSCMO can always maintain a low IGD value throughout the evolution process compared with PPS. In conclusion, both PPS and ToP consider all constraints simultaneously. If the feasible region of the problem is extremely complicated, it is difficult to find the global optimum, or even no feasible solutions can be found. However, for the proposed MSCMO, the multi-stage constraint-handling technology can considerably reduce the complexity of feasible regions, and thus it can significantly improve the performance of MOEAs in solving CMOPs with complex feasible regions.

4. Experimental study

To verify the performance of the proposed algorithm, we performed a series of experiments, the results of which are discussed in this section. First, the proposed MSCMO was compared with five state-of-the-art CMOEAs on five benchmark test suites. Then, we verified the effectiveness of the constraint-handling priority strategy suggested in the proposed MSCMO. Finally, to further verify the performance of the proposed MSCMO, we tested several CMOPs from real-world applications. All our experiments were executed on the evolutionary multi-objective optimization platform PlatEMO [37].

4.1. Experiment settings

- Tested problems: The benchmark problems adopted in this paper included MW problems [28], LIR-CMOP problems [12], DAS-CMOP problems [13], constrained DTLZ problems [10,25], and DOC problems [27]. The number of objectives and number of decision variables, denoted as m and d, respectively, for these benchmark problems were set as follows. For the MW problems [28], LIR-CMOP problems [12], and DAS-CMOP problems [13], $m = 2$ and $d = 15$ for all problems except for MW-4, MW-8, MW-14, LIR-CMOP13, LIR-CMOP14, DAS-CMOP7, DAS-CMOP8, and DAS-CMOP9, for which m was set to 3. For the constrained DTLZ [10] [25], $m = 3$ and $d = 12$ for all the problems except for C1-DTLZ1, DC1-DTLZ1, DC2-DTLZ1, and DC3-DTLZ1, for where d was set to 7. For the DOC problems [27], m and d were set according to the values in the original paper [27]. In addition, we also adopted three real-world CMOPs to verify the performance of the proposed MSCMO: the speed reducer design problem [32], the disk brake design problem [16], and the car side impact problem [21]. The definitions of the three real-world CMOPs are as follows. The speed reducer design problem [32] aims to minimize the stress in one of the two gear shafts and the volume and is subject to a number of constraints imposed by the gear and shaft. The disk brake design problem [16] aims to minimize the mass of the brake and the stopping time and is subject to the constraints imposed by the distance between the radii, maximum length of the brake, pressure, temperature, and torque limitations. The car side impact problem [21] aims to minimize the weight of the car, public force of the passenger, and average velocity of the V-pillar and is subject to thelimiting values of 10 aspects. The m values for the CMOPs were $2, 2$, and 3, and the d values were $7, 4$, and 7, respectively. The number of function evaluations for MW, constrained DTLZ, LIR-CMOP, DAS-CMOP, DOC, and real-world CMOPs was set to $200,000, 250,000, 300,000, 300,000, 300,000$, and $300,000$, respectively. Note that the proposed MSMCO uses an extra population to determine the constraint-handling priority and needs a certain number of function evaluations. For fairness, these function evaluations are included in the total function evaluations.
- Compared CMOEAs: Five state-of-the-art CMOEAs, namely NSGA-II-CDP [11], C-TAEA [25], PPS [14], ToP [27], and MOEA/D-DAE [48], were adopted as baselines to verify the performance of the proposed algorithm. The parameters of all the compared algorithms were set as suggested in their original papers. For PPS, the parameter was embedded in the constrained MOEA/D [21], where the parameter settingswere $\alpha = 0.95, \tau = 0.1, cp = 2$, and $l = 20$. For ToP, it was embedded in the NSGA-II-CDP, where the first phase ended when the feasibility proportion P_f was larger than $1/3$ or the difference δ was less than 0.2. For MOEA/D-DAE, the parameter α was set to 0.95. In addition, the population size of all the comparison algorithms and the proposed algorithm in our experiment was set to 100.
- Genetic operators: NSGA-II-CDP, C-TAEA, MOEA/D-DAE, and MSCMO use simulated binary crossover [8] and polynomial mutation [9] to generate offspring solutions, whereas PPS and ToP use DE [33] and polynomial mutation. The crossover probability was set to 1, the mutation probability was set to $1/d$ (d is the number of decision variables), the distribution index of both the crossover and the mutation was set to 20, and the parameters CR and F in the DE were set to 1 and 0.5, respectively.
- Evaluation metrics: For the benchmark problems, the IGD [5] metric was employed to assess the performance of all the CMOEAs because their true Pareto fronts (PFs) are known. Approximately $10,000$ reference points sampled on the PF of the problem were used for the calculation of the IGD. For the real-world CMOPs, the hypervolume (HV)[43] metric was employed, where the reference point was set to $1.1 \times z^{nad}$ (z^{nad} denotes the nadir point of the non-dominated solutions

Table 2
Statistical results of the IGD obtained by the proposed MSCMO and all the compared CMOEAs on the DTLZ and MW problems. The best result in each row is highlighted. “N/A” indicates that no feasible solution was found.

Problem	NSGA-II-CDP	C-TAEA	PPS	ToP	MOEA/D-DAE	MSCMO
C1-DTLZ1	2.713e-2 (1.25e-3) –	2.338e-2 (1.55e-4) –	2.613e-2 (8.66e-4) –	7.937e-2 (9.25e-2) –	2.111e-2 (1.19e-4) –	2.010e-2 (1.68e-4)
C2-DTLZ2	6.167e+0 (3.42e+0) –	7.172e-2 (8.74e-3) –	1.929e+0 (3.42e+0) –	8.725e-1 (2.43e+0) –	6.419e-2 (5.24e-3) –	5.344e-2 (6.96e-4)
C1-DTLZ3	5.657e-2 (3.26e-3) –	5.679e-2 (1.43e-3) –	5.541e-2 (1.77e-3) –	5.906e-2 (3.03e-3) –	4.416e-2 (4.16e-4) –	4.257e-2 (4.58e-4)
C3-DTLZ4	1.289e-1 (5.85e-3) –	1.124e-1 (2.45e-3) –	1.233e-1 (3.66e-3) –	1.427e-1 (7.69e-3) –	5.046e-1 (3.20e-1) –	9.543e-2 (1.46e-3)
DC1-DTLZ1	1.473e-2 (4.73e-4) –	1.564e-2 (5.17e-4) –	2.063e-2 (3.22e-3) –	1.207e-2 (1.22e-4) –	5.046e-1 (2.57e-1) –	1.140e-2 (8.14e-5)
DC1-DTLZ3	1.288e-1 (3.06e-3) –	1.304e-1 (3.99e-3) –	1.677e-1 (1.00e-1) –	2.703e-1 (1.78e-1) –	1.154e-1 (1.83e-3) –	1.137e-1 (8.86e-4)
DC2-DTLZ1	N/A	2.340e-2 (1.71e-4) –	2.810e-2 (7.48e-4) –	N/A	2.875e-2 (3.30e-2) –	2.014e-2 (1.52e-4)
DC2-DTLZ3	N/A	6.349e-2 (6.96e-4) –	2.799e-1 (2.49e-1) –	9.345e-1 (0.00e+0) ≈	2.952e-1 (2.59e-1) –	5.275e-2 (3.22e-4)
DC3-DTLZ1	1.328e-1 (6.89e-2) –	9.645e-3 (8.16e-4) –	3.881e-2 (6.28e-2) –	8.445e-1 (2.12e+0) –	7.386e-3 (8.71e-5) –	6.854e-3 (6.35e-5)
DC3-DTLZ3	1.389e+0 (4.37e-1) –	1.722e-1 (4.27e-3) +	4.364e-1 (4.52e-1) –	5.574e+0 (2.40e+0) –	1.591e-1 (2.18e-3) ≈	1.958e-1 (1.14e-1)
MW1	1.969e-3 (5.66e-5) –	2.213e-3 (9.86e-4) –	2.653e-3 (9.39e-5) –	4.948e-1 (0.00e+0) –	1.833e-3 (3.66e-5) –	1.619e-3 (1.74e-5)
MW2	2.474e-2 (1.02e-2) ≈	1.798e-2 (8.20e-3) ≈	1.202e-1 (4.52e-1) –	1.128e-1 (8.97e-2) –	8.579e-2 (7.18e-2) –	2.678e-2 (2.30e-2)
MW3	5690e-3 (2.66e-4) –	4.732e-3 (2.17e-4) ≈	6.473e-3 (6.06e-4) –	4.708e-1 (4.58e-1) –	4.988e-3 (2.02e-4) –	4.655e-3 (1.41e-4)
MW4	5.534e-2 (2.26e-3) –	4.660e-2 (3.56e-4) –	5.552e-2 (1.93e-3) –	2.854e-1 (2.20e-1) –	4.293e-2 (3.69e-4) –	4.057e-2 (3.38e-4)
MW5	2.964e-1 (3.51e-1) –	7.977e-3 (2.47e-3) –	3.498e-1 (3.72e-1) –	4.917e-1 (3.34e-1) –	8.233e-5 (5.22e-5) +	9.217e-4 (7.75e-4)
MW6	4.137e-2 (8.85e-2) ≈	9.895e-3 (5.27e-3) +	4.623e-1 (2.64e-1) –	6.745e-1 (3.68e-1) –	3.758e-1 (2.09e-1) –	2.744e-2 (2.07e-2)
MW7	1.981e-2 (8.15e-2) –	6.277e-3 (6.42e-4) –	5.711e-3 (5.07e-4) –	1.460e-2 (4.40e-3) –	4.515e-3 (1.81e-4) –	4.033e-3 (1.70e-4)
MW8	5.870e-2 (5.92e-3) –	5.480e-2 (2.05e-3) –	1.562e-1 (6.05e-2) –	3.988e-1 (3.50e-1) –	1.054e-1 (4.84e-2) –	4.653e-2 (4.29e-3)
MW9	5.051e-3 (1.94e-4) –	1.0574e-2 (5.89e-4) –	3.928e-1 (5.21e-1) –	6.267e-3 (8.82e-4) –	4.837e-3 (2.68e-4) –	4.070e-3 (1.15e-4)
MW10	1.548e-1 (1.75e-1) –	1.344e-2 (9.93e-3) +	3.530e-1 (1.97e-1) –	4.641e-1 (2.91e-1) –	3.425e-1 (2.25e-1) –	3.8566e-2 (2.44e-2)
MW11	2.134e-1 (3.21e-1) –	1.236e-2 (1.50e-3) –	7.478e-3 (3.51e-4) –	2.577e-1 (2.73e-1) –	6.696e-3 (3.75e-4) –	5.877e-3 (9.52e-5)
MW12	5.575e-3 (1.78e-4) –	7.807e-3 (7.64e-4) –	7.438e-3 (7.79e-4) –	7.707e-1 (2.00e-1) –	5.210e-3 (1.74e-4) –	4.611e-3 (1.04e-4)
MW13	1.550e-1 (2.44e-1) –	3.975e-2 (2.39e-2) +	4.648e-1 (3.70e-1) –	5.745e-1 (4.45e-1) –	3.243e-1 (3.11e-1) –	6.331e-2 (3.34e-2)
MW14	1.241e-1 (9.04e-3) –	1.116e-1 (3.68e-3) –	1.335e-1 (1.08e-2) –	2.428e-1 (1.58e-1) –	1.117e-1 (3.84e-2) –	9.694e-2 (1.18e-3)
+/ – / ≈	0/20/2	4/18/2	0/24/0	0/21/2	1/22/1	

obtained by all the CMOEAs), as suggested in [20]. All tested problems were performed for 30 independent runs, and the mean value and standard deviation were calculated for comparing the proposed algorithm with the state-of-the-art algorithms.

4.2. Experimental results on the benchmark problems

The results for the DTLZ and MW problems are given in Table 2. From Table 2, the following observations can be made. (1) The proposed MSCMO achieves the best results on 18 problems, followed by C-TAEA, which achieves the best results on 4 problems, and MOEA/D-DAE, which achieves 2 best results, whereas NSGA-II-CDP, PPS, and ToP do not achieve the best results. (2) For some CMOPs with a simple feasible region, such as C1-DTLZ1, C2-DTLZ2, MW1, and MW12, which contain only one constraint, most CMOEAs can yield good results. To further illustrate this observation, Fig. 16 plots the populations obtained by the proposed MSCMO and all the compared CMOEAs on MW1 with the median value. We can observe from the figure that the proposed MSCMO and all the compared CMOEAs except for ToP can find the global optimal solutions. (3) However, for CMOPs with narrow, discrete, or irregular feasible regions, such as the DC1-DTLZ1, DC1-DTLZ3, DC3-DTLZ1, MW5, and MW11 problems, the proposed MSCMO shows a better performance. It is worth mentioning that MOEA/D-DAE has also achieved good results on these problems because its unique DAE mechanism can help the population escape from the local optimum. To further illustrate this observation, Fig. 17 plots the populations obtained by the proposed MSCMO and all the compared CMOEAs on DC3-DTLZ1 with the median value. It can be seen from the figure that both NSGA-II-CDP and ToP converge to the top of the optimal feasible region and cannot find feasible solutions with the optimal objective value. The

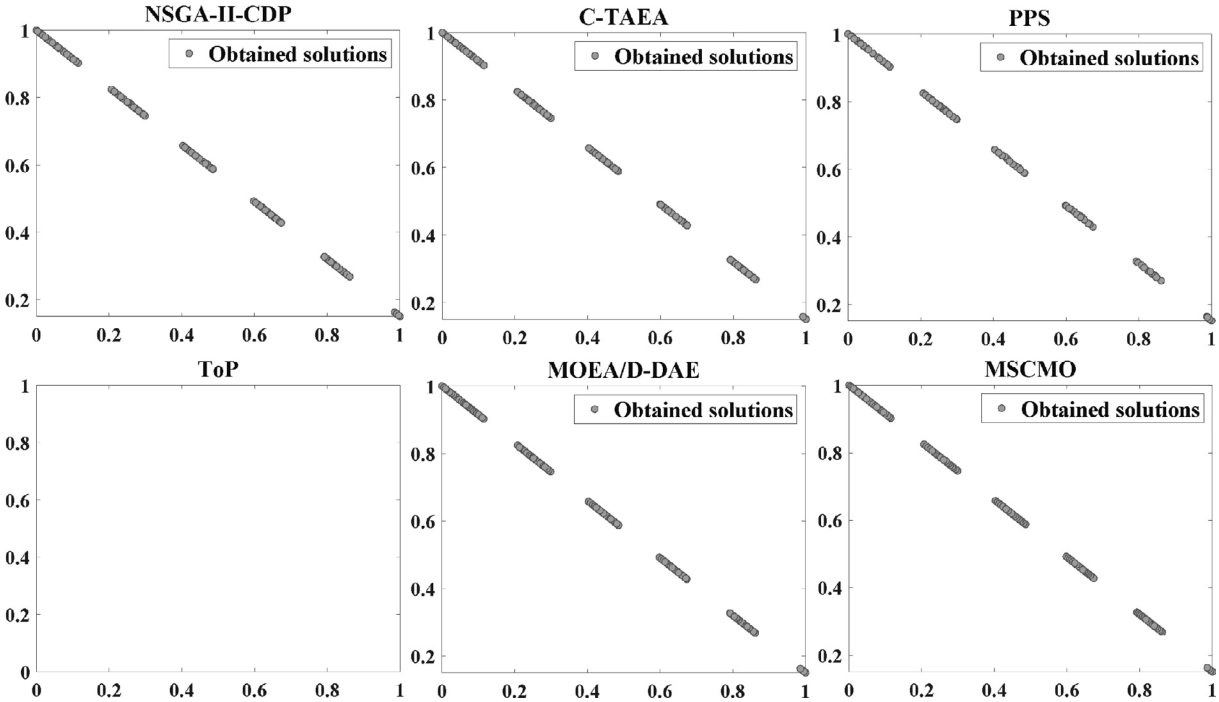

Fig. 16. Populations with the median IGD obtained by the proposed MSCMO and all the compared CMOEAs on MW1.

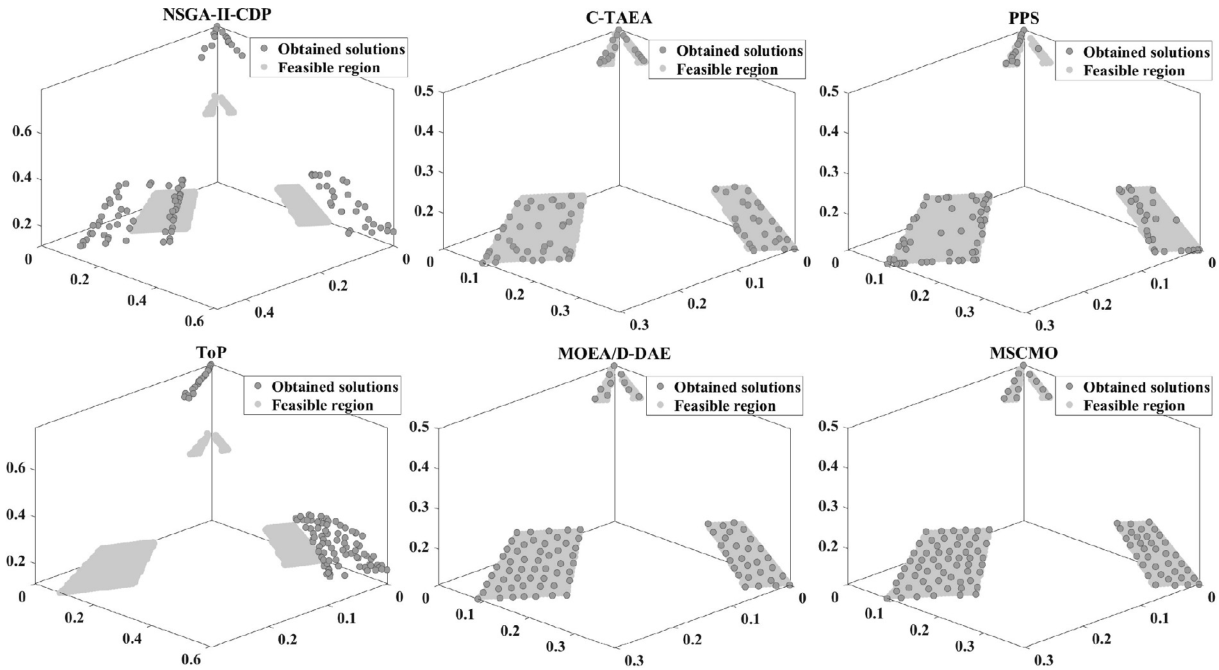

Fig. 17. Populations with the median IGD obtained by the proposed MSCMO and all the compared CMOEAs on DC3-DTLZ1.

proposed MSCMO and the remaining three compared CMOEAs can find the optimal feasible region; however, the populations obtained by C-TAEA and PPS cannot cover all feasible regions because of the complex feasible region. Because of the DAE mechanism, MOEA/D-DAE can escape from the local optimum and obtain good solutions containing all the feasible regions. The proposed MSCMO only considers part of the constraints in the early stages, and thus it can easily cross the local optimum and obtain solutions with good diversity. As the proposed MSCMO proceeds with more constraints being added stage by stage, the population that evolved based on the solutions with a good diversity can converge to excellent results.

Table 3 shows the results for the LIR-CMOP problems. From Table 3, we can make the following observations. (1) The proposed MSCMO achieves the best results on 7 problems, followed by MOEA/D-DAE, which achieves 5 best results, and PPS,

Table 3
Statistical results of the IGD obtained by the proposed MSMCO and all the compared CMOEAs on the LIR-CMOP problems. The best result in each row is highlighted.

Problem	NSGA-II-CDP	C-TAEA	PPS	ToP	MOEA/D-DAE	MSCMO
LIR-CMOP1	2.427e-1 (5.06e-2) −	1.122e-1 (5.42e-2) −	9.925e-3 (3.41e-3) +	2.193e-1 (1.29e-1) −	5.898e-3 (6.45e-4) +	1.495e-2 (7.14e-3)
LIR-CMOP2	1.956e-1 (4.34e-2) −	5.229e-2 (1.31e-2) −	6.080e-3 (3.43e-4) +	1.513e-1 (6.98e-2) −	5.407e-3 (2.73e-4) ≈	1.309e-2 (1.55e-2)
LIR-CMOP3	2.595e-1 (5.91e-2) −	1.922e-1 (1.15e-1) −	7.108e-3 (4.53e-3) +	3.334e-1 (2.84e-2) −	2.989e-1 (1.65e-1) −	6.373e-2 (6.06e-2)
LIR-CMOP4	2.555e-1 (4.61e-2) −	1.215e-1 (5.84e-2) −	3.584e-3 (4.06e-4) +	3.176e-1 (4.25e-2) −	1.804e-1 (1.04e-1) −	3.671e-2 (3.09e-2)
LIR-CMOP5	9.505e-1 (4.28e-1) −	2.240e-1 (1.88e-1) −	6.575e-3 (6.20e-4) −	2.751e-1 (4.90e-1) −	5.005e-3 (2.56e-4) +	5.241e-3 (2.25e-4)
LIR-CMOP6	1.153e+0 (3.94e-1) −	5.586e-1 (4.69e-1) −	7.566e-3 (1.00e-3) −	4.736e-1 (4.22e-1) −	5.940e-3 (2.56e-4) ≈	5.274e-3 (1.97e-4)
LIR-CMOP7	6.169e-2 (3.07e-2) −	7.097e-2 (2.96e-2) −	1.063e-2 (8.87e-4) −	6.450e-2 (3.06e-1) −	1.083e-2 (7.23e-4) −	7.132e-3 (1.99e-4)
LIR-CMOP8	2.767e-1 (4.83e-1) −	1.124e-1 (8.20e-2) −	1.046e-2 (8.45e-4) −	1.784e-1 (2.10e-1) −	9.251e-3 (6.57e-4) −	7.135e-3 (2.00e-4)
LIR-CMOP9	5.724e-1 (1.58e-1) −	1.387e-1 (3.73e-2) −	4.979e-2 (1.09e-1) −	3.791e-1 (3.54e-2) −	2.734e-3 (9.78e-5) ≈	2.729e-3 (7.53e-5)
LIR-CMOP10	4.315e-1 (1.17e-1) −	1.239e-1 (1.21e-1) −	5.263e-3 (2.14e-4) −	5.953e-3 (2.18e-4) −	4.428e-3 (1.55e-4) ≈	4.539e-3 (2.14e-4)
LIR-CMOP11	4.072e-1 (2.18e-1) −	1.703e-1 (2.51e-2) −	7.152e-3 (2.57e-2) −	1.804e-1 (7.01e-2) −	2.555e-3 (1.20e-4) −	2.392e-3 (4.30e-5)
LIR-CMOP12	2.497e-1 (9.49e-2) −	3.167e-2 (1.55e-2) −	3.026e-3 (9.51e-5) −	1.515e-1 (3.84e-2) −	3.013e-3 (1.28e-4) −	2.888e-3 (2.16e-4)
LIR-CMOP13	6.827e-1 (6.12e-1) −	1.085e-1 (1.55e-3) +	1.282e-1 (4.62e-3) −	1.317e-1 (3.61e-3) −	9.779e-2 (1.24e-3) +	1.115e-1 (1.45e-3)
LIR-CMOP14	6.630e-1 (5.89e-1) −	1.116e-1 (1.25e-3) −	1.193e-1 (2.98e-3) −	1.205e-1 (3.20e-3) −	1.010e-1 (6.94e-4) ≈	1.009e-1 (1.07e-3)
+/ − / ≈	0/14/0	1/13/0	4/10/0	0/14/0	3/6/5	

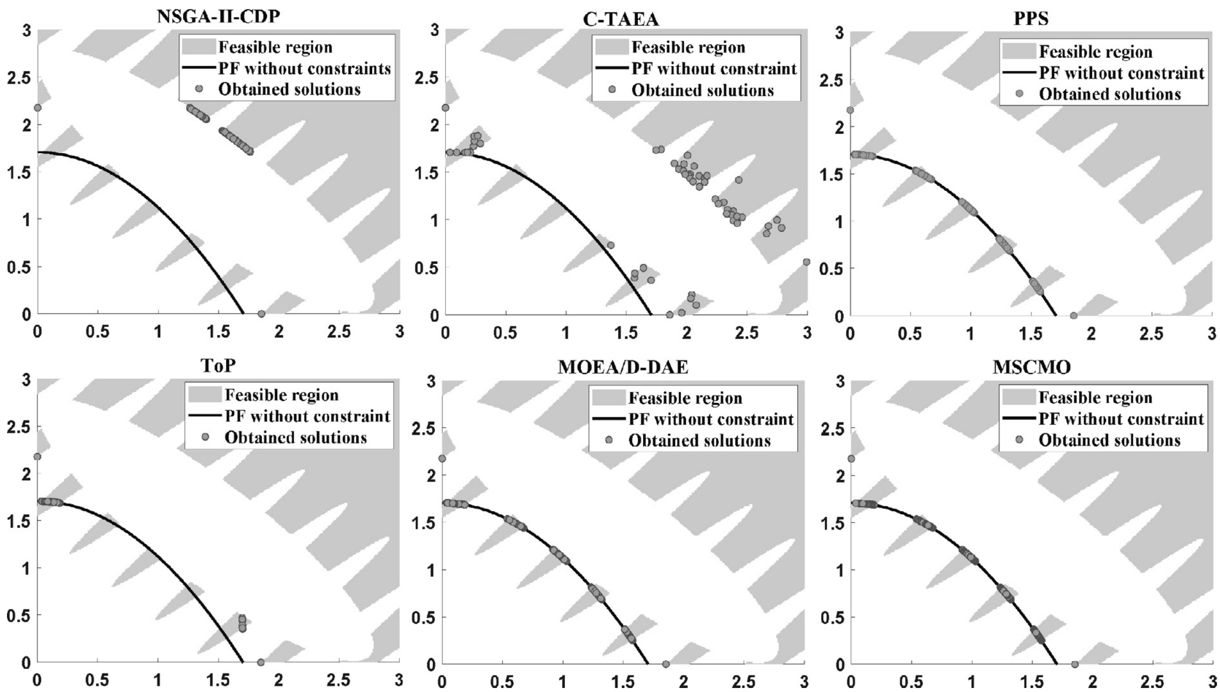

Fig. 18. Populations with the median IGD obtained by the proposed MSMCO and all the compared CMOEAs on LIR-CMOP9.

which achieves the best results on 2 problems, whereas the other three compared CMOEAs achieve no best results. (2) For the problems LIR-CMOP1 to LIR-CMOP4 whose true Pareto front is a curve and the upper and lower sides of the curve are surrounded by infeasible regions, the results of the proposed MSCMO are worse than that of PPS or MOEA/D-DAE. The main reason can be attributed to the fact that the proposed MSCMO has difficulty finding feasible solutions and maintaining a good diversity in the early stage on such problems, and thus the population will lose the diversity at the later stages and will only converge to the local optimum in the end. (3) For the remaining problems, there are mainly two types of constraints: one type constitutes a huge infeasible barrier, and the other type affects the unconstrained Pareto front. According to the constraint priority sorting strategy, the MSMCO will cross the infeasible barrier and deal with the constraints that affect the unconstrained Pareto front first, and thus it can achieve good performance. MOEA/D-DAE and PPS having aDAE mechanism and a push–pull search stage, respectively, can cross the infeasible barrier and maintain a good diversity, thus also achieving good performance. To further illustrate this observation, Fig. 18 plots the populations obtained by the proposed MSMCO and all the compared CMOEAs with the median IGD on the LIR-CMOP9 problem. The population of NSGA-II-CDP cannot cross the infeasible barrier. For C-TAEA, part of the population crosses the infeasible barrier but gets trapped in local optima, whereas the other part cannot cross the infeasible barrier. Although ToP can cross the infeasible barrier, it gets trapped in local optima. The MSMCO and all the other two CMOEAs can cross the infeasible barrier, achieve excellent performance, and find the global optimum in the end.

In addition, we compared the performance of the proposed MSCMO and all the compared algorithms on two more complex CMOPs, namely DAS-CMOP problems and DOC problems, which contain more than 10 constraints on average. Table 4 shows the statistical results of the IGD of all the CMOEAs. From Table 4, we can see that the proposed MSCMO shows significantly outstanding performance on such complex problems. Specifically, the proposed MSCMO obtains 16 best results in all 18 problems; both PPS and ToP achieve 1 best result, whereas the other CMOEAs achieve no best results. For DOC problems that not only contain objective space constraints but also have many decision space constraints, NSGA-II-CDP, C-TAEA, and MOEA/D-DAE cannot find feasible solutions in some problems. However, the two-stage CMOEAs, namely PPS, ToP, and the proposed MSCMO, can find feasible solutions in all problems, which confirms the superiority of handling constraints in stages. The populations of six CMOEAs on DOC2 with the median IGD are shown in Fig. 19. Among the six CMOEAs, only the proposed MSCMO finds the global optimal solutions, which further proves the superiority of the proposed MSCMO for handling CMOPs with complex constraints.

Table 4
Statistical results of the IGD obtained by the proposed MSMCO and all the compared CMOEAs on the DAS-CMOP and DOC problems. The best result in each row is highlighted. "N/A" indicates that no feasible solution was found.

Problem	NSGA-II-CDP	C-TAEA	PPS	ToP	MOEA/D-DAE	MSCMO
DAS-CMOP1	2.808e-1 (2.00e-1) –	5.014e-2 (7.10e-2) –	2.255e-3 (8.51e-4) ≈	5.554e-2 (1.55e-1) –	3.612e-3 (2.163-3) –	2.266e-3 (3.74e-4)
DAS-CMOP2	1.248e-1 (3.50e-2) –	8.413e-3 (2.30e-3) –	3.927e-3 (1.22e-4) –	2.962e-2 (5.53e-2) –	3.218e-3 (8.10e-5) –	3.000e-3 (8.67e-5)
DAS-CMOP3	2.464e-1 (5.78e-2) –	3.215e-2 (8.56e-3) –	1.927e-2 (7.74e-5) –	2.557e-1 (1.11e-1) –	3.522e-2 (6.58-2) –	1.916e-2 (1.26e-3)
DAS-CMOP4	5.542e-2 (1.10e-1) –	1.017e-2 (2.24e-3) –	1.639e-3 (6.32e-5) –	7.766e-1 (2.00e-1) –	1.576e-3 (4.14e-4) –	1.173e-3 (6.06e-5)
DAS-CMOP5	3.615e-3 (1.08e-4) –	7.358e-3 (4.94e-4) –	4.019e-3 (1.29e-4) –	7.190e-1 (1.15e-1) –	3.283e-3 (1.95e-4) –	3.148e-3 (1.64e-4)
DAS-CMOP6	4.596e-2 (8.09e-2) ≈	2.360e-2 (3.42e-3) –	1.956e-2 (5.24-3) ≈	7.783e-1 (9.71e-2) –	2.124e-2 (1.68e-3) –	1.902e-2 (2.41e-3)
DAS-CMOP7	4.857e-2 (2.32e-3) –	5.850e-2 (7.47e-3) –	6.291e-2 (6.66e-3) –	8.802e-1 (2.77e-1) –	5.350e-2 (1.05e-2) –	4.765e-2 (1.24e-2)
DAS-CMOP8	6.023e-2 (3.53e-3) –	9.428e-2 (1.20e-2) –	8.891e-2 (1.35e-2) –	8.320e-1 (2.10e-1) –	7.685e-2 (3.15e-2) –	5.350e-2 (1.39e-2)
DAS-CMOP9	6.812e-2 (4.16e-3) –	9.654e-2 (8.01e-3) –	7.203e-2 (1.13e-2) –	6.176e-2 (3.11e-3) –	5.953e-2 (6.31e-3) –	4.080e-2 (1.24e-3)
DOC1	2.058e+0 (1.38e+0) –	5.569e+2 (1.89e+2) –	6.344e-2 (4.24e-2) –	5.953e-3 (3.27e-4) –	1.070e-2 (7.11e-2) –	5.214e-3 (4.20e-4)
DOC2	N/A	N/A	3.440e-1 (2.42e-1) –	4.390e-1 (1.21e-1) –	N/A	5.360e-3 (3.53e-3)
DOC3	6.936e+2 (1.77e+2) –	N/A	3.242e+2 (3.63e+2) ≈	9.803e+1 (1.52e+2) ≈	3.244e+2 (3.34e+2) ≈	2.307e+2 (2.76e+2)
DOC4	6.025e-1 (3.49e-1) –	1.773e+2 (2.14e+2) –	2.786e-1 (5.04e-2) –	5.970e-2 (5.10e-2) ≈	5.846e-1 (3.03e-1) –	4.687e-2 (1.13e-1)
DOC5	N/A	N/A	1.502e+1 (3.62e+1) –	5.131e+1 (1.06e+2) –	N/A	7.752e+0 (3.16e+1)
DOC6	2.734e+0 (3.75e+0) –	9.859e+1 (3.24e+2) –	5.188e-1 (4.75e-2) –	2.574e+0 (1.35e+0) –	5.107e-1 (1.16e-1) –	2.977e-3 (2.04e-4)
DOC7	4.739e+0 (1.84e+0) –	N/A	4.344e-1 (1.72e-1) –	3.472e-1 (3.01e-1) –	9.237e-1 (3.65e-1) –	2.366e-3 (8.80e-5)
DOC8	7.148e+1 (6.21e+1) –	4.006e+2 (9.89e+1) –	9.415e+1 (3.48e+1) –	1.782e+1 (1.10e+1) –	7.720e+1 (8.51e+1) –	6.279e-2 (3.35e-3)
DOC9	1.767e-1 (9.04e-2) –	6.035e-1 (6.96e-2) –	2.7522e-1 (2.21e-2) –	1.745e-1 (3.95e-2) –	2.349e-1 (1.16e-1) –	8.534e-2 (1.07e-2)
+/ – / ≈	0/15/1	0/14/0	0/15/3	0/16/2	0/14/2	

4.3. Effectiveness of the constraint-handling priority sorting strategy

To verify the effectiveness of the proposed constraint priority sorting strategy of the proposed MSCMO,we performed an ablation study on the multi-constraint test suite DAS-CMOP [13] with three variants of the proposed MSCMO. Although the constraint-handling priority is different in the proposed MSCMO, the other components of the three variants are the same as those of the proposed MSCMO. The first variant MSCMO1 sorts the constraints in descending order according to the infeasible rate; however, the constraints with an infeasible rate of 0 are not combined. The second variant MSCMO2 sorts the con-

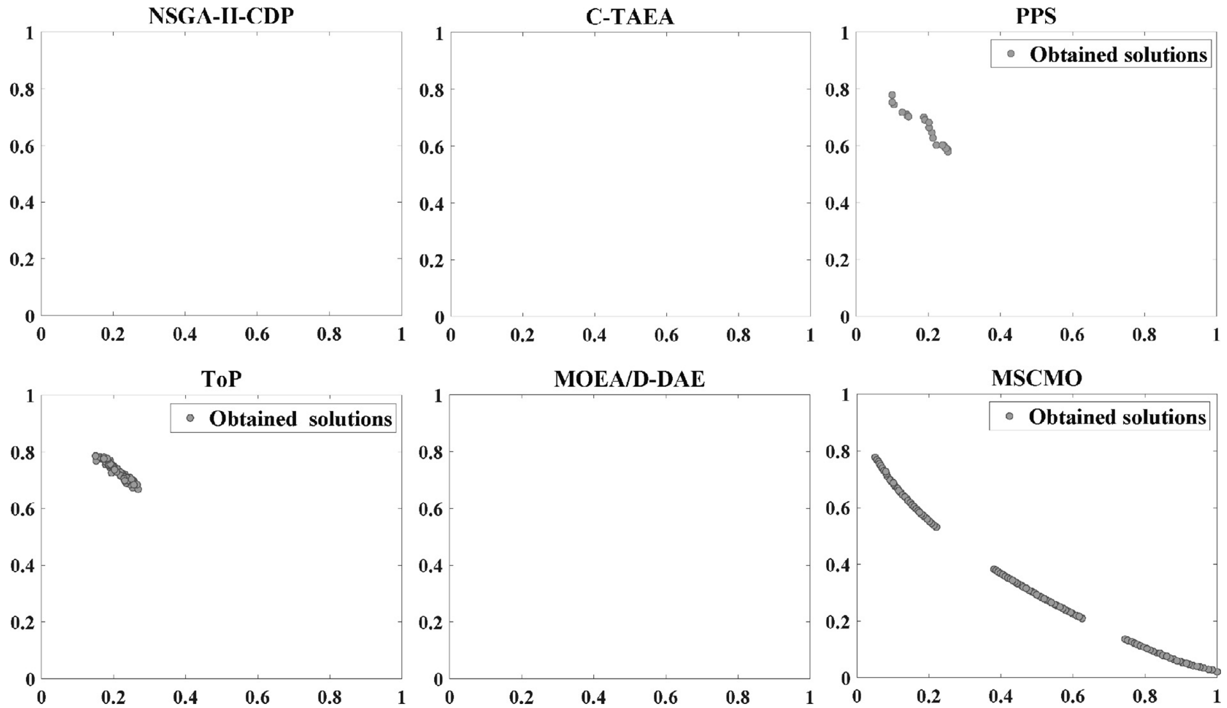

Fig. 19. Populations with the median IGD obtained by the proposed MSMCO and all the compared CMOEAs on DOC2.

Table 5
Statistical results of the IGD obtained by the proposed MSCMO and its three variants on the DAS-CMOP problems. The best result in each row is highlighted.

Problem	MSCMO1	MSCMO2	MSCMO3	MSCMO
DAS-CMOP1	2.576e-3 (7.99e-4) ≈	2.739e-3 (7.63e-4) −	2.586e-3 (7.91e-4) −	2.268e-3 (3.74e-4)
DAS-CMOP2	3.123e-3 (9.59e-5) −	3.194e-3 (8.35e-5) −	3.118e-3 (9.39e-5) −	3.000e-3 (8.67e-5)
DAS-CMOP3	1.854e-2 (2.35e-3)≈	1.920e-2 (1.32e-3) ≈	1.855e-2 (2.33e-3) ≈	1.916e-2 (1.26e-3)
DAS-CMOP4	1.654e-3 (1.12e-3) −	2.035e-3 (9.83e-3) −	1.301e-3 (4.06e-4) −	1.173e-3 (6.06e-5)
DAS-CMOP5	3.892e-3 (2.61e-3) −	3.631e-3 (5.99e-4) −	3.862e-3 (6.59e-4) −	3.148e-3 (1.64e-4)
DAS-CMOP6	2.287e-2 (1.37e-2) −	2.034e-2 (4.78e-3) ≈	1.955e-2 (2.19e-3) ≈	1.902e-2 (2.41e-3)
DAS-CMOP7	5.479e-2 (3.07e-2) ≈	4.306e-2 (4.17e-3) ≈	4.324e-2 (9.83e-3) ≈	4.765e-2 (2.14e-2)
DAS-CMOP8	5.973e-2 (3.01e-2) ≈	5.099e-2 (4.94e-3) ≈	5.872e-2 (2.77e-2) ≈	5.350e-2 (1.39e-2)
DAS-CMOP9	4.095e-2 (1.22e-3) ≈	4.084e-2 (1.13e-3) ≈	4.080e-2 (1.16e-3) ≈	4.080e-2 (1.24e-3)
+/ − / ≈	0/4/5	0/4/5	0/4/5	

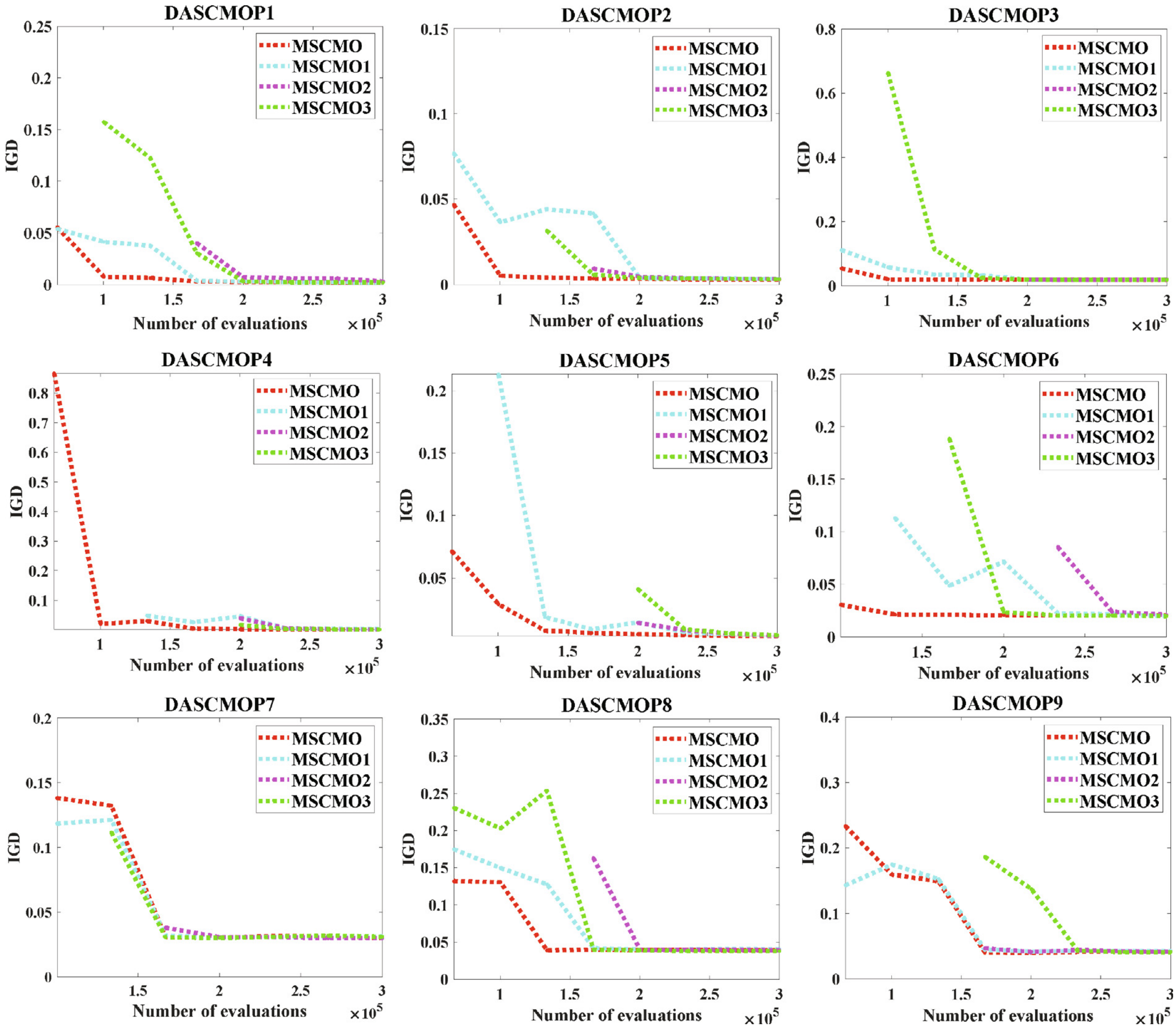

Fig. 20. Experimental results of the proposed MSCMO and its three variants on the DAS-CMOP problems in terms of the efficiency of finding feasible solutions and convergence speed.

straints in ascending order according to the infeasible rate, and the last variant MSCMO3 sorts the constraints ina random order according to the infeasible rate. The statistical results of the IGD of the proposed MSCMO and its three variants are shown in Table 5. The proposed MSCMO achieves the best results for most test instances. We can see that their IGD results are relatively close, which implies that no matter how the constraints are ordered this multi-stage constraint-handling method can achieve good results. To further demonstrate the effectiveness of the designed constraint-handling priority sorting strategy, we compared the proposed MSCMO and its three variants in terms of the efficiency of finding feasible solutions and convergence speed, as shown in Fig. 20. From the figure, we can see that both MSCMO and MSCMO1 can find feasible solutions earlier than the other two variants. MSCMO2, whose constraint priority order is completely in contrast to those of MSCMO and MSCMO1, always finds the feasible solution in the last stage. For MSCMO3, which handles constraint priority randomly, the time to find a feasible solution is uncertain (in most cases, earlier than MSCMO2, but later than MSCMO and MSCMO1). In terms of convergence speed, the proposed MSCMO is significantly faster than the other variants in almost all problems. Therefore, we can conclude that the proposed constraint priority sorting strategy can improve the efficiency of finding feasible solutions and increase the speed of convergence, especially for CMOPs with complex constraints.

Table 6
Statistical results of the IGD obtained by the proposed MSMCO and its three variants on the LIR-CMOP problems. The best result in each row is highlighted.

Problem	MSCMOi	MSCMOii	MSCMOiii	MSCMO
LIR-CMOP1	4.841e-1 (1.75e-1) –	7.586e-2 (7.01e-2) –	4.787e-1 (1.21e-1) –	1.495e-2 (7.14e-3)
LIR-CMOP2	6.593e-1 (2.34e-1) –	4.469e-2 (6.53e-2) –	4.386e-1 (1.21e-1) –	1.309e-2 (1.55e-2)
LIR-CMOP3	4.449e-1 (1.93e-1) –	1.578e-1 (1.46e-1) –	3.838e-1 (3.33e-2) –	6.373e-2 (6.06e-2)
LIR-CMOP4	3.832e-1 (6.87e-2) –	2.086e-1 (1.73e-1) –	4.135e-1 (1.63e-1) –	3.671e-2 (3.09e-2)
LIR-CMOP5	5.886e-3 (8.80e-4) –	5.121e-1 (5.76e-1) –	5.926e-3 (7.29e-4) –	5.241e-3 (2.25e-4)
LIR-CMOP6	5.405e-3 (2.23e-4) –	3.394e-1 (3.66e-1) –	5.700e-3 (8.81e-4) –	5.274e-3 (1.97e-4)
LIR-CMOP7	5.773e-2 (5.87e-3) –	7.150e-3 (1.84e-4) ≈	5.654e-2 (7.75e-3) –	7.132e-3 (1.99e-4)
LIR-CMOP8	5.365-2 (7.34e-3) –	7.097e-3 (2.37e-4) ≈	4.999e-2 (6.88e-3) –	7.135e-3 (2.00e-4)
LIR-CMOP9	4.334e-2 (3.29e-2) –	1.636e-1 (1.443-1) –	4.595e-2 (3.39e-2) –	2.729e-3 (7.53e-5)
LIR-CMOP10	1.180e-2 (7.92e-4) –	4.581e-3 (1.78e-4) ≈	1.178e-2 (6.16e-4) –	4.539e-3 (2.14e-4)
LIR-CMOP11	3.263e-2 (2.37e-2) –	3.763e-2 (5.78e-2) –	4.160e-2 (3.01e-2) –	2.392e-3 (4.30e-5)
LIR-CMOP12	4.793e-2 (3.11e-2) –	9.668e-2 (7.24e-2) –	4.402e-2 (3.07e-2) –	2.888e-3 (2.16e-4)
LIR-CMOP13	1.259e-1 (5.45e-3) –	1.109e-1 (1.58e-3) ≈	1.255e-1 (5.36e-3) –	1.115e-1 (1.45e-3)
LIR-CMOP14	1.487e-1 (1.14e-2) –	1.011e-1 (1.30e-3) ≈	1.481e-1 (7.40e-3) –	1.009e-1 (1.07e-3)
+/ – / ≈	0/14/0	0/9/5	0/14/0	

Table 7
Statistical results of HV obtained by the proposed MSMCO and all the compared CMOEAs on the three real-world CMOPs. The best result in each row is highlighted.

Problem (no. of variables / no. of objectives / no. of constraints)	NSGA-II-CDP	C-TAEA	PPS	ToP	MOEA/D-DAE	MSCMO
Speed reducer design problem (7/2/11)	3.5794e-1 (2.94e-4) –	2.7788e-1 (1.95e-2) –	2.7805e-1 (1.71e-2) –	3.5746e-1 (4.70e-4) –	2.7135e-1 (1.11e-2) –	3.6149e-1 (1.80e-4)
Disk brake design problem (4/2/5)	7.5940e-1 (2.21e-4) +	7.5751e-1 (8.25e-4) –	7.5930e-1 (3.17e-4) +	7.5862e-1 (2.29e-4) –	7.5867e-1 (8.10e-4) ≈	7.5882e-1 (2.49e-4)
Car side impact problem (7/3/10)	6.3998e-2 (1.233-4) +	6.40953-2 (1.77e-4) –	6.3480e-2 (1.53e-4) +	6.4058e-2 (1.67e-4) +	6.4545e-2 (1.25e-4) ≈	6.4709e-2 (6.79e-5)
+/ – / ≈	1/2/0	0/3/0	1/2/0	0/3/0	0/2/1	

4.4. Effectiveness of using different constraint-handling mechanisms at different stages

In the proposed MSCMO, different constraint-handling mechanisms at different stages are utilized. Specifically, before the last stage, the multi-objective optimization method is used to deal with constraints and aims to maintain a good diversity of the population. In the last stage, after all constraints are considered, the feasibility rule is used to strengthen the convergence of the population. Because the early stage focuses on diversity and the later stage focuses on convergence, the population is more likely to converge to the global optimum. To verify the effectiveness of this method, we performed an ablation study on the LIR-COMP problems, where MSCMOwas compared with its three variants. The first variant, MSCMOi, adopts multi-objective optimization to handle constraints at all stages, whereas the second variant, MSCMOii, adopts a feasibility rule to handle constraints at all stages. The third variant, MSMCOiii, swaps the order of using the two mechanisms (i.e., adopting the feasibility rule first and then the multi-objective optimization method). Table 6 presents the IGD results of MSCMO and its three variants on LIRCMOP1-14. It can be seen from the table that MSCMO is significantly better than its three variants on most problems; therefore, the effectiveness of using different constraint-handling mechanisms at different stages can be verified.

4.5. Experimental results on real-world CMOPs

Finally, to further verify the performance of the proposed MSCMO, we performed experiments on three real-world CMOPs, i.e., the speed reducer design problem [32], which has two objectives with 7 variables and 11 inequality constraints; the disk brake design problem [16], which has two objectives with 4 variables and 5 inequality constraints; and the car side impact problem [21],which contains three objectives with 7 variables and 10 inequality constraints. The definition of the objectives, variables, andconstraints on the three problems and the specific mathematical formulations can be found in their original papers [32,16,21]. The HV results of all the algorithms on the three real-world CMOPs are shown in Table 7. Among the three problems, the proposed MSCMO achieves the best results for the speed reducer design problem and car side impact problem. For the disk brake design problem, the proposed MSCMO obtains the third best result, which is slightly worse than those of NSGA-II-CDP and PPS. In short, the proposed MSCMO shows better performance than the five compared CMOEA on the three real-world CMOPs, which further indicates that it is also very competitive in solving practical problems.

5. Conclusions

In this paper, we proposed a multi-stage CMOEA for solving CMOPs with a relatively complex feasible region. Specifically, in the proposed algorithm, the constraints are added one by one and handled in different stages of evolution. In the early stages, only a small number of constraints are considered, which makes the population efficiently converge to the potential feasible region with good diversity. As the proposed algorithm enters the next stage, more constraints are considered, and the diversity of the population is maintained by using the solutions found in the previous stages. In this manner, the original complex problem can be easily solved. Furthermore, we also sort and make a combination of constraints to improve the efficiency of searching for feasible solutions and convergence speed. In the experimental part, the proposed algorithm was compared with several state-of-the-art CMOEAs on five benchmark suites and three real-world applications. The results showed that our proposed algorithm is very competitive, especially for CMOPs with complex constraints.

Because the proposed algorithm is very flexible and easy to implement, it is desirable to solve other types of CMOPs with more complex feasible regions. Furthermore, it is also interesting to design more effective sorting and combining strategies for the constraints to further improve the efficiency and effectiveness. For example, we can classify the types of constraints and handle the same types of constraints together.

CRediT authorship contribution statement

Haiping Ma: Investigation, Methodology, Writing - original draft. **Haoyu Wei:** Data curation, Methodology, Visualization. **Ye Tian:** Writing - review & editing. **Ran Cheng:** Writing - review & editing. **Xingyi Zhang:** Conceptualization, Investigation, Writing - review & editing.

Declaration of Competing Interest

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

Acknowledgments

This work was supported by the National Key Research and Development Project (2018AAA0100105), the National Natural Science Foundation of China (61822301, 61672033, 61906001, and U1804262), the Hong Kong Scholars Program

(XJ2019035), the Anhui Provincial Natural Science Foundation (1808085J06 and 1908085QF271), and the CCF-Tencent Open Research Fund (CCF-Tencent RAGR20200121).

References

[1] Y. Zhang, Y. Mei, B. Zhang, K. Jiang, Divide-and-conquer large scale capacitated arc routing problems with route cutting off decomposition, Information Sciences 553 (2021) 208–224. http://www.sciencedirect.com/science/article/pii/S0020025520310975, https://doi.org/10.1016/j.ins.2020.11.011.
[2] R. Jiao, S. Zeng, C. Li, A feasible-ratio control technique for constrained optimization, Information Sciences 502 (2019) 201–217.
[3] I.M. Ali, D. Essam, K. Kasmarik, Novel binary differential evolution algorithm for knapsack problems, Information Sciences 542 (2021) 177–194.
[4] Y. Yina, Y. Zhao, H. Li, X. Dong, Multi-objective evolutionary clustering for large-scale dynamic community detection, Information Sciences 549 (2020) 269–287.
[5] P.A.N. Bosman, D. Thierens, The balance between proximity and diversity in multiobjective evolutionary algorithms, IEEE Transactions on Evolutionary Computation 7 (2003) 174–188.
[6] C.A.C. Coello, S.G. Brambila, J.F. Gamboa, G.C.T. Ma, R.H. Gámez, Evolutionary multiobjective optimization: open research areas and some challenges lying ahead, Complex & Intelligent Systems 6 (2020) 221–236.
[7] K. Deb, An efficient constraint handling method for genetic algorithms, Computer Methods in Applied Mechanics and Engineering 186 (2000) 311–338.
[8] K. Deb, R.B. Agrawal, Simulated binary crossover for continuous search space, Complex Systems 9 (1995) 115–148.
[9] K. Deb, M. Goyal, A combined genetic adaptive search (GeneAS) for engineering design, Computer Science and Informatics 26 (1996) 30–45.
[10] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints, IEEE Transactions on Evolutionary Computation 18 (2014) 577–601.
[11] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2002) 182–197.
[12] Z. Fan, W. Li, X. Cai, H. Huang, Y. Fang, Y. You, J. Mo, C. Wei, E.D. Goodman, An improved epsilon constraint-handling method in moea/d for cmops with large infeasible regions, Soft Computing 23 (2019) 12491–12510..
[13] Z. Fan, W. Li, X. Cai, L. Hui, E.D. Goodman, in press. Difficulty adjustable and scalable constrained multi-objective test problem toolkit, Evolutionary Computation (2019).
[14] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, E.D. Goodman, Push and pull search for solving constrained multi-objective optimization problems, Swarm and Evolutionary Computation 44 (2019) 665–679.
[15] R. Farmani, J.A. Wright, Self-adaptive fitness formulation for constrained optimization, IEEE Transactions on Evolutionary Computation 7 (2003) 445–455, https://doi.org/10.1109/TEVC.2003.817236.
[16] A. Got, A. Moussaoui, D. Zouache, A guided population archive whale optimization algorithm for solving multiobjective optimization problems, Expert Systems With Applications 141 (2020) 112972.
[17] C. He, L. Li, Y. Tian, X. Zhang, R. Cheng, Y. Jin, X. Yao, Accelerating large-scale multiobjective optimization via problem reformulation, IEEE Transactions on Evolutionary Computation 23 (2019) 949–961.
[18] A. Homaifar, C. Qi, H.Y. Lai, Constrained optimization via genetic algorithms, Transactions of The Society for Modeling and Simulation International - SIMULATION 62 (1994) 242–253, https://doi.org/10.1177/003754979406200405.
[19] Z. Hou, C. He, R. Cheng, Reformulating preferences into constraints for evolutionary multi- and many-objective optimization, Information Sciences 541 (2020) 1–15..
[20] H. Ishibuchi, R. Imada, Y. Setoguchi, Y. Nojima, How to specify a reference point in hypervolume calculation for fair performance comparison, Evolutionary Computation 26 (2018) 411–440.
[21] H. Jain, K. Deb, An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach, IEEE Transactions on Evolutionary Computation 18 (2014) 602–622.
[22] L. Jiao, L. Li, R. Shang, F. Liu, R. Stolkin, A novel selection evolutionary strategy for constrained optimization, Information Sciences 239 (2013) 122–141.
[23] R. Jiao, S. Zeng, C. Li, S. Yang, Y.S. Ong, Handling constrained many-objective optimization problems via problem transformation, IEEE Transactions on Cybernetics (2020) 1–14, https://doi.org/10.1109/TCYB.2020.3031642.
[24] B. Li, K. Tang, J. Li, X. Yao, Stochastic ranking algorithm for many-objective optimization based on multiple indicators, IEEE Transactions on Evolutionary Computation 20 (2016) 924–938.
[25] K. Li, R. Chen, G. Fu, X. Yao, Two-archive evolutionary algorithm for constrained multi-objective optimization, IEEE Transactions on Evolutionary Computation 23 (2018) 303–315.
[26] Chih-Hao Lin, A rough penalty genetic algorithm for constrained optimization, Information Sciences 241 (2013) 119–137.
[27] Z. Liu, Y. Wang, Handling constrained multiobjective optimization problems with constraints in both the decision and objective spaces, IEEE Transactions on Evolutionary Computation 23 (2019) 870–884.
[28] Ma Z., Wang Y., Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons, IEEE Transactions on Evolutionary Computation, 2019 (in press)..
[29] Z. Ma, Y. Wang, W. Song, A new fitness function with two rankings for evolutionary constrained multiobjective optimization, IEEE Transactions on Systems, Man, and Cybernetics: Systems (2019) 1–12, https://doi.org/10.1109/TSMC.2019.2943973.
[30] J.R. McDonnell, R.G. Reynolds, D.B. Fogel, A Survey of Constraint Handling Techniques in Evolutionary Computation Methods, MIT Press, 1995.
[31] C. Peng, H. Liu, E.D. Goodman, A cooperative evolutionary framework based on an improved version of directed weight vectors for constrained multiobjective optimization with deceptive constraints, IEEE Transactions on Cybernetics (2020) 1–13.
[32] C. Peng, H. Liu, F. Gu, An evolutionary algorithm with directed weights for constrained multi-objective optimization, Applied Soft Computing 60 (2017) 613–622.
[33] K. Price, R.M. Storn, J.A. Lampinen, Differential Evolution: A Practical Approach to Global Optimization, Springer Science & Business Media, 2006.
[34] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, J. Wu, MOEA/D with adaptive weight adjustment, Evolutionary Computation 22 (2014) 231–264.
[35] C. Saha, S. Das, K. Pal, S. Mukherjee, A fuzzy rule-based penalty function approach for constrained evolutionary optimization, IEEE Transactions on Cybernetics 46 (2016) 2953–2965, https://doi.org/10.1109/TCYB.2014.2359985.
[36] Y. Tian, R. Cheng, X. Zhang, F. Cheng, Y. Jin, An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility, IEEE Transactions on Evolutionary Computation 22 (2018) 609–622.
[37] Y. Tian, R. Cheng, X. Zhang, Y. Jin, PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum], IEEE Computational Intelligence Magazine 12 (2017) 73–87, https://doi.org/10.1109/MCI.2017.2742868.
[38] Y. Tian, T. Zhang, J. Xiao, X. Zhang, Y. Jin, A coevolutionary framework for constrained multi-objective optimization problems, IEEE Transactions on Evolutionary Computation (2020) 1, https://doi.org/10.1109/TEVC.2020.3004012.
[39] Y. Tian, Y. Zhang, Y. Su, X. Zhang, K. Tan, Y. Jin, Balancing objective optimization and constraint satisfaction in constrained evolutionary multi-objective optimization, IEEE Transactions on Cybernetics (2020), https://doi.org/10.1109/TCYB.2020.3021138.
[40] H. Wang, L. Jiao, X. Yao, Two_arch2: An improved two-archive algorithm for many-objective optimization, IEEE Transactions on Evolutionary Computation 19 (2015) 524–541.
[41] J. Wang, G. Liang, J. Zhang, Cooperative differential evolution framework for constrained multiobjective optimization, IEEE Transactions on Cybernetics 49 (2019) 2060–2072, https://doi.org/10.1109/TCYB.2018.2819208.
[42] Y. Wang, Z. Cai, Y. Zhou, W. Zeng, An adaptive tradeoff model for constrained evolutionary optimization, IEEE Transactions on Evolutionary Computation 12 (2008) 80–92.

[43] L. While, P. Hingston, L. Barone, S. Huband, A faster algorithm for calculating hypervolume, IEEE Transactions on Evolutionary Computation 10 (2006) 29–38.
[44] Y. Xiang, X. Yang, Y. Zhou, H. Huang, Enhancing decomposition-based algorithms by estimation of distribution for constrained optimal software product selection, IEEE Transactions on Evolutionary Computation 24 (2020) 245–259.
[45] M. Zhang, W. Luo, X. Wang, Differential evolution with dynamic stochastic selection for constrained optimization, Information Sciences 178 (2008) 3043–3074.
[46] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Transactions on Evolutionary Computation 11 (2007) 712–731.
[47] Y. Zhou, M. Zhu, J. Wang, Z. Zhang, Y. Xiang, J. Zhang, Tri-goal evolution framework for constrained many-objective optimization, IEEE Transactions on Systems, Man, and Cybernetics: Systems 50 (2020) 3086–3099, https://doi.org/10.1109/TSMC.2018.2858843.
[48] Q. Zhu, Q. Zhang, Q. Lin, A constrained multiobjective evolutionary algorithm with detect-and-escape strategy, IEEE Transactions on Evolutionary Computation (2020) 1.
[49] E. Zitzler, S. Kunzli, Indicator-based selection in multiobjective, search (2004) 832–842..
[50] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization, in: Proceedings of the Fifth Conference on Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems, 2001, pp. 95–100.