Egyptian Informatics Journal 25 (2024) 100422

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

A niche-based evolutionary algorithm with dual cooperative archive for solving constrained multi-objective optimization problems

Fengyu Guo [a], Hecheng Li [b,*]

[a] School of Computer Science and Technology, Qinghai Normal University, Xining, 810016, China
[b] School of Mathematics and Statistics, Qinghai Normal University, Xining, 810016, China

ARTICLE INFO

Keywords:
Constrained multi-objective optimization
Dual stage algorithm
Infeasible individual utilization
Niche-based selection method

ABSTRACT

Constrained multi-objective optimization problems (CMOPs) are commonly encountered in engineering practice. The key to effectively solving these problems lies in achieving a timely balance between convergence, diversity, and feasibility during iterations. Furthermore, the appropriate utilization of infeasible solutions is crucial for identifying potential feasible regions. In order to accomplish this comprehensive objective, we propose a novel dual-stage constrained multi-objective evolutionary algorithm (CMOEA) called NACMOEA in this paper. It can be characterized by the following features: 1) Introducing a novel niche-based individual selection and infeasible solution utilization strategy to enhance convergence, diversity, and feasibility. 2) Presenting a cooperative search strategy assisted by dual archives to approximate the constrained Pareto front (CPF) from both feasible and infeasible perspectives, thereby improving the efficiency of obtaining the complete CPF. 3) Designing a new stage switch method based on non-dominant coverage rate to ensure proper completion of search stage switching. Extensive experiments demonstrate that NACMOEA exhibits competitive comprehensive performance when compared with other advanced CMOEAs.

1. Introduction

CMOPs are widespread in scientific and engineering practice, such as front rail design [6], pressure vessel design [12], etc. To solve such problems normally requires CMOEAs to optimize multiple conflicting objectives with constraint restrictions. A general mathematical expression of CMOPs can be formulated as follows:

$$\begin{aligned} \text{minimize} \quad & F(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))^T \\ \text{subject to} \quad & g_j(\mathbf{x}) \le 0, j = 1, \ldots, p \\ & h_j(\mathbf{x}) = 0, j = p+1, \ldots, q \\ & \mathbf{x} \in \Omega \end{aligned} \tag{1}$$

where $\mathbf{x}$ is a decision variable vector in the search space Ω, $F(\mathbf{x})$ is an objective vector with m objective function values. $g_j(\mathbf{x})(h_j(\mathbf{x}))$ are the jth inequality (equality constraint). The constraint violation of $\mathbf{x}$ for the jth constraint can be calculated as follows:

$$CV_j(\mathbf{x}) = \begin{cases} \max\left\{0, g_j(\mathbf{x})\right\}, j = 1, \ldots, p, \\ \max\left\{0, \left|h_j(\mathbf{x})\right|\right\}, j = p+1, \ldots, q \end{cases} \tag{2}$$

the overall constraint violation is defined as:

$$\Phi(\mathbf{x}) = \sum_{j=1}^{q} CV_j(\mathbf{x}) \tag{3}$$

a solution is called a feasible solution when its total constraint violation is 0; otherwise, it is considered infeasible. To compare the performance of solutions for CMOPs, we introduce the concept of Pareto dominance. Given feasible solutions $\mathbf{x}$ and $\mathbf{y}$, $\mathbf{x}$ is said to Pareto dominate $\mathbf{y}$ (denoted as $\mathbf{x} \prec \mathbf{y}$) if $f_k(\mathbf{x}) \le f_k(\mathbf{y})$ for all $k \in (1, \ldots, m)$ and $f_k(\mathbf{x}) < f_k(\mathbf{y})$ for at least one k. A solution is considered Pareto optimal when no other feasible solution dominates it. The set of all Pareto optimal solutions in the search space is known as the Pareto optimal set (PS). The representation of PS in the objective space forms the Pareto optimal front (PF).

When traditional optimization algorithms are adopted to handle CMOPs, they face computational challenges that the algorithms have to identify the feasible region and keep diversity of individuals for the purpose of convergence. In fact, in most of CMOPs, constraints can always cause the search space infeasible, especially, some tight constraints can make the feasible region overly narrow and even isolated. As a result, the algorithm has difficulty striking a balance between feasibility and

* Corresponding author.
E-mail addresses: fengyu_guo1990@163.com (F. Guo), lihecheng@qhnu.edu.cn (H. Li).

https://doi.org/10.1016/j.eij.2023.100422
Received 30 September 2023; Received in revised form 9 November 2023; Accepted 20 November 2023
Available online 28 November 2023

convergence, easily getting stuck in a local optimum and failing to locate the full CPF.

A well-established technique for global optimization with great robustness and broad applicability is the multi-objective evolutionary algorithm (MOEA). MOEAs offer a collection of non-dominated approximately optimal solutions for multiple conflicting goals. Hence, they can handle complicated situations that are challenging for classical optimization procedures. In all popular categories of MOEAs, there are three representatives. Dominance-based MOEAs are categorized as the first category, such as NSGA-II [5] and SPEA2 [41]. In the second category of MOEAs, an objective decomposition scheme is adopted by using Chebyshev's formula, such as MOEA/D [37] and MOEA/DD [13]. And the third is indicator-based MOEAs such as IBEA [40] and HypE [3]. These MOEAs can effectively handle unconstrained multi-objective optimization problems, but they have some restrictions when applied to CMOPs, some or even all of the optimal solutions found by these algorithms lie in infeasible regions that are invalid for CMOPs. In order to overcome the shortcomings, the research community has developed several different constraint handling techniques (CHTs) to assist MOEAs of this kind to search for optimal feasible solutions. CHTs can be viewed as a selection strategy for handling constraints, which is used to select some infeasible solutions with low constraint violation. These infeasible solutions enable MOEAs to have the ability of crossing infeasible regions and eventually converge to the CPF.

The research on CMOEAs has attracted wide attention in the academic community and has made significant progress in recent years. However, there are still some issues in this field that need to be addressed. The first issue worth considering is how to better balance the algorithm's convergence, diversity, and feasibility. Due to the influence of constraints, achieving a balance in these characteristics is challenging for CMOEAs. Consequently, the algorithm is prone to getting stuck in a local feasible or optimal region and failing to discover all feasible regions, which hinders its convergence towards the CPF or only allows it to find a portion of it. The second problem is how to effectively utilize information from infeasible solutions. Specifically, identifying infeasible solutions located near undiscovered feasible regions can enhance the algorithm's exploration ability. Additionally, utilizing high-convergence infeasible solutions located near the feasible region can expedite the convergence of feasible individuals.

To address the aforementioned issue, this paper proposes a niche-based two-stage evolutionary algorithm called NACMOEA. This algorithm is supported by dual collaborative archives and incorporates a novel strategy for utilizing infeasible solutions. The key characteristics of the proposed approach are outlined below.

1. The search process of the proposed NACMOEA is divided into two stages, during which a new dual archive collaboration strategy is developed.
2. A niche-based individual selection and infeasible individual utilization strategy is proposed to better balance the convergence, diversity, and feasibility of NACMOEA.
3. A novel non-dominant coverage rate metric is designed for switching the search stage.
4. The proposed algorithm exhibits competitive performance compared with other advanced algorithms on a large number of complex CMOP test problems.

The remainder of the article is organized as follows. In Section 2, we briefly review the recent literature of CMOEAs and introduce the common CHTs. The motivation for this paper is subsequently elicited. The specific mechanisms of the proposed algorithm are described in detail in Section 3, followed by a description of the experimental studies in Section 4. Finally, the conclusions and future work are presented in Section 5.

2. Background

This section provides a review of existing CMOEAs, with a focus on analyzing CHTs in CMOEAs, comparing the advantages and weaknesses of several common CHTs, and presenting the motivation for this paper.

2.1. Literature review

Significant progress has been made in the academic community's research on CMOP, with various approaches proposed. These include those based on multiple populations, utilization of infeasible solutions, multi-stage strategies, and the balancing of objectives and constraints. The subsequent section presents an overview of representative literature concerning these methodologies.

2.1.1. CMOEAs based on multi-population/archive

The CHT based on a dual-archive set was first presented by C-TAEA [14]. In this approach, the two archive sets have complementary roles to ensure population diversity while promoting population convergence and feasibility. In CCMO [30] framework, a dual-population co-evolutionary strategy based on a weak cooperative relationship is provided. In this strategy, two sub-populations evolve independently and share their offspring in their respective environmental selection stages. The dual-population-based evolutionary algorithm proposed in c-DPEA [24] uses an adaptive penalty function to assist evolution using information from infeasible solutions and creates a new objective function to balance population convergence and diversity. In BiCo [15], a bidirectional co-evolutionary strategy is introduced to approximate the CPF from both the feasible and infeasible sides. The CMOCSO [22] framework solves CMOPs through a competition and cooperation mechanism between two swarm optimizers, effectively improving convergence. For constrained many-objective optimization problems, the MOEA/D-DAE [39] framework offers a detection and escape mechanism that uses the overall constraint violation rate of change to determine whether the population is trapped in a local optimum. In MFO [10], the constraints of the original CMOP are relaxed to create a simpler auxiliary problem; transferring experience gained from solving this auxiliary problem can be helpful in solving the original problem.

2.1.2. CMOEAs using infeasible solutions

In C-AnEA [33], an angle-based constraint dominance rule and density estimation method are provided, which give the infeasible solutions with excellent convergence a chance to survive. This helps the population cross the infeasible regions. In ShiP [19], a shift-based penalty is imposed on infeasible solutions, retaining those that do not fall into the region dominated by Locally Feasible Nadir after the shift. Inspired by multi-task evolution, the MTCMO [26] framework uses a highly related dynamic auxiliary task for the complex original problem and designs an improved constraint relaxation method to leverage high-quality infeasible solutions.

2.1.3. CMOEA based on a multi-stage strategy

In PPS [8], the unconstrained search is defined as the push stage, while the ϵ-constrained technique-based search is defined as the pull stage. The PPTA [27] framework presents a push-pull search strategy based on the assistance of a dual archive, which effectively improves the performance of PPS. The DTAEA [34] framework suggests a two-phase approach, where the first phase conducts a dual population search based on weak co-evolution and the second phase conducts a feasibility-oriented single population search that drives the population to converge to the final CPF. In DD-CMOEA [25], a dual-stage and dual-population strategy is employed to enhance the search capability of the algorithm by implementing different divisions of labor and collaboration strategies for each population in different stages. In FNDS [36], a population partitioning method based on feasible non-dominated solution sets is proposed, which dynamically divides the entire population into three

mutually exclusive subsets and performs different tasks for these subsets to guide them towards approaching CPF.

2.1.4. CMOEA based on objective and constraint balancing strategies

In CMOEA-MS [31], a new strategy for evaluating objective functions is provided to achieve a better balance between objectives and constraints by assigning different priorities to them. The TOR [20] framework rewrites the original fitness function as a sum of two rankings based on CDP and Pareto dominance, controlling the weights through feasible rate to achieve a balance between objectives and constraints. The CMME [23] framework effectively addresses the issue of decreasing selection pressure on populations as the number of objectives increases by employing two ranking strategies. DPSEA [32] dynamically adjusts the trade-off between objectives and constraints through population size.

2.2. Common CHTs

The MOEA research community has recently attempted to efficiently solve CMOPs by combining different MOEAs with multiple types of CHTs. CHTs, which are a crucial technique for handling constraints, play a particularly critical role in improving the performance of CMOEAs. Some representative CHTs are introduced as follows.

2.2.1. Constraint domination principle (CDP)

CDP is the most representative CHT employed in CMOEAs, which was originally proposed by Deb et al. in NSGA-II [5]. It classifies the population into three categories and assigns a higher selection priority to solutions with high feasibility and excellent convergence. The benefit of CDP is that its driving concept is straightforward and simple to apply. Assuming that $\mathbf{x}$ and $\mathbf{y}$ are two solutions, we say that $\mathbf{x}$ constrained dominates $\mathbf{y}$ (denoted by $\mathbf{x} \prec_{CDP} \mathbf{y}$) when one of the following conditions is satisfied.

- $\Phi(\mathbf{x}) = 0, \Phi(\mathbf{y}) \neq 0$;
- $\Phi(\mathbf{x}) = 0, \Phi(\mathbf{y}) = 0,$ *and* $\mathbf{x} \prec \mathbf{y}$;
- $\Phi(\mathbf{x}) \neq 0, \Phi(\mathbf{y}) \neq 0,$ *and* $\Phi(\mathbf{x}) < \Phi(\mathbf{y})$.

CDP is widely used in CMOEAs, such as NSGA-II [5] and C-NSGA-III [9]. However, due to its preference for feasible solutions, the population can easily get trapped in the local optimum region, thereby ignoring the benefits of infeasible solutions.

2.2.2. ϵ-constrained technique

The ϵ-constrained technique is another typical CHT that stems from a partial improvement of CDP, which introduces a constraint relaxation factor ϵ. For an infeasible solution $\mathbf{x}$, if its overall constraint violations $\Phi(\mathbf{x})$ are less than ϵ, the solution can still survive, and then these infeasible solutions are thereby incorporated into the evolutionary process. For two solutions $\mathbf{x}$ and $\mathbf{y}$, we say that $\mathbf{x}$ ϵ-constrained dominates $\mathbf{y}$ (denoted by $\mathbf{x} \prec_{\epsilon} \mathbf{y}$) when one of the following conditions is satisfied.

- $\Phi(\mathbf{x}) < \epsilon, \Phi(\mathbf{y}) < \epsilon,$ *and* $\mathbf{x} \prec \mathbf{y}$;
- $\Phi(\mathbf{x}) = \Phi(\mathbf{y}) = \epsilon,$ *and* $\mathbf{x} \prec \mathbf{y}$;
- $\Phi(\mathbf{x}) > \epsilon, \Phi(\mathbf{y}) > \epsilon,$ *and* $\Phi(\mathbf{x}) < \Phi(\mathbf{y})$.

An improved strategy to adaptively control ϵ is designed in [8], which is formulated as:

$$\varepsilon(k) = \begin{cases} (1-\tau)\varepsilon(k-1), & \text{if } r_k < \alpha \\ \varepsilon(0)\left(1 - \frac{k}{T_c}\right)^{cp}, & \text{if } r_k \geq \alpha \end{cases} \tag{4}$$

where k represents the kth generation, r_k is the feasible ratio, τ and α are two control parameters which are usually assigned 0.05 and 0.95, respectively. cp, usually assigned 2, is used to control the speed of reducing ϵ value in the case of $r_k \geq \alpha$, and $\epsilon(0)$ is the maximal constraint violation. The ϵ-constrained technique overcomes the shortcomings of CDP to a certain extent, enhancing the exploration ability of CMOEA, and is widely used in MOEA/D-DAE [39] and I-DBEA [2]. Noting that the parameter ϵ plays a key role in the search process of CMOEA, too large ϵ can cause infeasible solutions far from the feasible region to be selected, and too small value of ϵ possibly causes similar problems as CDP.

2.2.3. Two-rank fitness method

A new fitness is originally created in TOR [20], where two ranks $R_p(\mathbf{x})$ and $R_c(\mathbf{x})$ of solution $\mathbf{x}$ are calculated based on Pareto dominance and CDP, respectively. Then, a new fitness $\hat{F(\mathbf{x})}$ is calculated by taking the weighted sum of these two ranks.

$$\begin{aligned} \min \quad & \hat{F}(\mathbf{x}) = \alpha R_c(\mathbf{x}) + (1-\alpha) R_p(\mathbf{x}) \\ & \alpha = 0.5 + 0.5 P_f \end{aligned} \tag{5}$$

The weight is adaptively controlled by the feasible rate P_f. However, when the proportion of feasible solutions is very low, this approach functions similarly to CDP. Consequently, the algorithm can easily become trapped in a local optimum.

The academic community has also recommended some additional CHTs besides the previously mentioned ones. The C3M [28] framework determines the priority of constraints by analyzing the relationship between multiple constraints, and deals with constraints according to their priority. The NRC [16] framework proposed a multi-objective method based CHT, which uses three procedures of non-dominated sorting, reversed non-dominated sorting, and constrained crowding distance sorting to move the population toward to CPF. In MOEA/D-DPF [21], a dynamic penalty function approach is developed to solve CMOPs that is integrated into the MOEA/D framework in which the parameters gradually change with generation number. In TiGE-2 [38], with the aim of achieving the balance of performances, three distinct indicators are developed while considering the convergence, diversity, and feasibility. These indicators are then co-optimized using multi-objective techniques.

2.3. Motivation

From the literature survey conducted above, it is evident that current CMOEAs only focus on either the algorithm's balance scheme or the utilization of infeasible solutions when dealing with CMOPs. However, they fail to effectively combine these two strategies. Furthermore, existing CMOEAs lack the capability to explore extensively within the infeasible region and obtain a comprehensive CPF. Infeasible solutions play a vital role in discovering unexplored feasible regions and enhancing algorithm convergence; however, their potential advantage remains underutilized. Based on the above discussions, we propose employing the following strategies to address the limitations of existing CMOEAs:

1. A dual archive procedure, consisting of a convergence archive (CA) and a feasibility archive (FA), is established for assignment division and collaboration purposes. The CA retains the UPF discovered by the algorithm, which is utilized to create angle-based niches. These niches are then employed to partition the objective space and determine the spatial relationship between individuals. As for FA, it functions as the output population responsible for storing elite solutions near the CPF. It employs a niche-based individual updating strategy to strike a balance among convergence, feasibility, and diversity objectives.
2. The precise utilization of infeasible solutions facilitates the exploration of unknown feasible regions and enhances the convergence of feasible solutions. We strategically pair infeasible solutions with optimal feasibility and feasible solutions with optimal convergence within the same niche, followed by executing a crossover operation for these matched individuals. The resulting offspring exhibit

promising potential to simultaneously improve both solution feasibility and convergence.

The algorithm effectively achieves a balance between convergence, diversity, and feasibility while efficiently utilizing infeasible solutions. The subsequent section presents a comprehensive account of the proposed NACMOEA algorithm.

Algorithm 1: Framework of NACMOEA.

Input: population size: N; maximum generation: G_m; non-dominant coverage threshold: C_t; generation gap: G_p;
Output: FA

$t \leftarrow 1$, $stage \leftarrow 1$;
$P \leftarrow$ Random initialize $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$;
$CA, FA \leftarrow P$;
while $t \leq G_m$ **do**
  if $t > G_p$ **then**
    $C \leftarrow$ nondominant coverage of P by Equation (6) ;
  if $C \geq C_t \& stage == 1$ **then**
    $stage \leftarrow 2$;
  if $stage == 1$ **then**
    $O_p \leftarrow$ Select mating parents from P and generate N offspring ;
    $P \leftarrow$ EnvironmentalSelection($[P, O_p], N, stage$) ;
    $CA \leftarrow$ UpdateCA($[CA, P], N$) ;
    $FA \leftarrow$ UpdateFA($[FA, P], N, stage$) ;
    $t \leftarrow t + 1$;
  if $stage == 2$ **then**
    $O_p \leftarrow$ Select mating parents from P and generate N offspring ;
    $O_{fa} \leftarrow$ Select mating parents from FA and generate $N/2$ offspring ;
    $O_r \leftarrow$ using InfeasibleUtilization(CA, FA, P, N) Select mating parents from P and FA and generate $N/2$ offspring ;
    $O_t \leftarrow O_p \cup O_{fa} \cup O_r$;
    $P \leftarrow$ EnvironmentalSelection($[P, O_p], N, stage$) ;
    $FA \leftarrow$ UpdateFA($CA, [FA, O_t], N, stage$) ;
    $t \leftarrow t + 2$;
return FA ;

3. Our approach

This section provides a detailed introduction to the framework and various algorithm components of NACMOEA.

3.1. Framework of NACMOEA

The framework of NACMOEA is shown in Algorithm 1. Initially, the generation counter t and search stage are set to 1. The population P is randomly initialized with N individuals. Subsequently, CA and FA are initialized by P. The algorithm starts from stage one, and the subsequent steps are iterated until the termination condition is met. Initially, to determine if the algorithm reaches a stable state, we compute the non-dominant coverage rate C of the current population using Equation (6). If this metric surpasses a predefined threshold C_t, the algorithm proceeds to stage two. In stage one, NACMOEA undertakes a convergence-oriented search to derive the UPF. A tournament selection is performed on P to select elite parents, and the simulated binary crossover operator and polynomial mutation operator are applied on selected parents to generate N offspring O_p. The next generation population is then generated by performing convergence-oriented environmental selection on combined P and O_p. The CA and FA employ the convergence-oriented and feasibility-oriented update strategy, respectively, to collect solutions that are close to the UPF and solutions near the optimal feasible region. In stage two, NACMOEA aims to approximate the complete CPF from both feasible and infeasible sides. By selecting parents from P, N Offspring O_p are generated. Since individuals in FA are located inside or near the optimal feasible region, their generated offspring can further explore the feasible region, so $N/2$ offspring O_{FA} are generated by the elite parents from FA. Another $N/2$ offspring are generated using the infeasible solution utilization strategy, which is described later. The ϵ-constrained technique-based environmental selection is implemented on the combined offspring population to form the next generation P. With the aim of discovering the complete CPF, a niche-based environmental selection strategy is employed for updating FA. In each iteration of stage two, since a total of $2N$ individuals are generated and evaluated, resulting in an increase of 2 for t in each iteration. At the end of iteration, FA is outputted as the final CPF found by NACMOEA.

Note that our proposed framework employs a unique two-stage search strategy, distinct from existing approaches. In the convergence-oriented search stage, constraints are not entirely ignored but rather assigned a small weight. Consequently, feasible solutions with good convergence have the potential to be retained, facilitating the exploration of the optimal feasible region for the problem. Similarly, in the feasibility-oriented search stage, convergence is not completely neglected but also given less emphasis. At this juncture, infeasible solutions with good convergence and low constraint violation may be preserved. Since these solutions are located near the feasible region and closer to the ideal point, they have the potential to improve the convergence of feasible solutions.

3.2. Decision of search stage

Switching between search stages is crucial because the proposed NACMOEA employs different search strategies during each stage. As the main population approaches the UPF and reaches a steady state, NACMOEA switches its focus to finding the CPF. Previous algorithms used metrics based on representative points like ideal and nadir points to reflect evolutionary state, but this strategy has a disadvantage in that nadir points are unstable and can change with each generation, leading to inaccurate estimations of evolutionary state. In this paper, we introduce a non-dominant coverage rate-based metric that overcomes these shortcomings.

The main idea of our proposed metric is that, according to the definition of Pareto dominance, when the population evolves into a steady state, the majority of individuals will become non-dominated and evenly distributed on the PF. Therefore, by comparing the dominance relationship between the current generation (t) population and its previous generation ($t - G_p$), if a larger proportion of individuals in t are non-dominated compared to those in $t - G_p$, it indicates that the t generation population has already converged to the PF and achieved a steady state. Conversely, if there still exist some dominance relationships between individuals in both populations, it suggests that further search should be continued as the evolution of the population has not yet stagnated at this stage. Our proposed metric has better stability compared to the metric based on representative points, but it also incurs a relatively higher computational cost due to the introduction of non-dominated sorting. The metric is shown as follows:

$$C = \frac{\left|\{a \in P^t \mid \forall b \in P^{t-G_p}, a \not\succ b\}\right|}{|P^t|} \tag{6}$$

where P^t is the current tth generation population, P^{t-G_p} is the $(t - G_p)$th historical generation population. When this metric is higher than the predefined threshold C_t, it indicates that more than C_t% individuals in the current population are non-dominated with respect to the historical populations, implying that the tth generation population has been improved slightly when compared to the $(t - G_p)$th generation. Hence, a stage switch should be executed.

3.3. Environmental selection

In this paper, environmental selection is employed to select the next generation population for the main population P. The fitness function used in this paper is defined as fellow, which is the same as that in SPEA2 [41].

$$Fit(\mathbf{x}) = \sum_{\mathbf{y}\in S_{\mathbf{x}}} |R_{\mathbf{y}}| + \frac{1}{dist(\mathbf{x},\mathbf{x}')+2} \tag{7}$$

where $R_{\mathbf{x}}$ is the set of solutions dominated by the solution $\mathbf{x}$ and $S_{\mathbf{x}}$ is the set of solutions dominating the solution $\mathbf{x}$. $\mathbf{x}'$ denotes the $\lfloor\sqrt{2n}\rfloor$ nearest neighbor to $\mathbf{x}$. The first part of Equation (7) denotes the total number of individuals dominated by the individual dominating $\mathbf{x}$ in the solution set, and the second part is the inverse of the Euclidean distance between $\mathbf{x}$ and its nearest $\lfloor\sqrt{2n}\rfloor$ neighbor $\mathbf{x}'$. A smaller $Fit(\mathbf{x})$ indicates the better quality of $\mathbf{x}$, and when $Fit(\mathbf{x}) < 1$, it means $\mathbf{x}$ is nondominant.

According to the above discussion, $Fit(\mathbf{x})$ is determined by the dominance relationship between $\mathbf{x}$ and other individuals in the population, so the $Fit(\mathbf{x})$ changes with different dominance rules. Since the search goal of P changes with search stage, the fitness function needs to be reformulated to realize the switch of search goal. Inspired by TOR [20], the switch of search goal can be achieved by assigning different weight to the rank-based fitness. Specifically, the dominance relationship of population is determined by Pareto dominance and CDP, respectively, and then the fitness is calculated by Equation (7), respectively. Sorting these two fitness values yields the rank-based fitness $R_p(\mathbf{x})$ and $R_c(\mathbf{x})$, where $R_p(\mathbf{x})$ represents the rank of $\mathbf{x}$ in population based on Pareto dominance. Similarly, $R_c(\mathbf{x})$ represents the rank of $\mathbf{x}$ based on CDP. Finally, by assigning different weight to $R_p(\mathbf{x})$ and $R_c(\mathbf{x})$, the final fitness $Fit_r(\mathbf{x})$ is given below:

$$Fit_r(\mathbf{x}) = \alpha * R_p(\mathbf{x}) + (1-\alpha) * R_c(\mathbf{x}) \tag{8}$$

As shown in Equation (8), $Fit_r(\mathbf{x})$ is calculated by the sum of $R_c(\mathbf{x})$ and $R_p(\mathbf{x})$ with different weights, respectively. In stage one, α is set to 0.9 and $Fit_r(\mathbf{x})$ mainly consists of $R_p(\mathbf{x})$. The algorithm executes convergence-oriented search with a weight of 0.1 for $R_c(\mathbf{x})$, while feasible solutions with good convergence are still likely to be retained, these feasible solutions can help for finding the optimal feasible regions. In stage two, α is set to 0.1 and $Fit_r(\mathbf{x})$ mainly consists of $R_c(\mathbf{x})$. The algorithm executes feasibility-oriented search with a weight of 0.1 for $R_p(\mathbf{x})$, during which infeasible solutions with low constraint violation and good convergence may be selected, which can help improve the convergence of the population. The pseudo code of environmental selection is shown in Algorithm 2.

Algorithm 2: EnvironmentalSelection.

Input: temp population: P_t; pop size: N; search stage: $stage$
Output: next generation population: P

```
Fit_p ← Calculate the Fitness of P_t by Equation (7) based on Pareto dominance ;
Fit_c ← Calculate the Fitness of P_t by Equation (7) based on CDP ;
R_p ← sort Fit_p ;
R_c ← sort Fit_c ;
if stage == 1 then
    α = 0.9 ;
else
    α = 0.1 ;
Fit_r ← calculate the rank-based Fitness by Equation (8) ;
R ← sort Fit_r ;
P ← select solutions of R(1 : N) from P_t;
return P ;
```

3.4. Update strategy of CA

In stage one, the UPF found by the main population is preserved via CA, while in stage two, CA is not updated, and is used to generate niches. The update strategy for CA is shown in Algorithm 3. Initially, the original CA is merged with the current population P to build a temporary population P_t and its fitness Fit_t is calculated. When the number of solutions with the $Fit_t < 1$ is less than N, the first N solutions are selected in descending order according to their fitness. When the number of solutions with the $Fit_t < 1$ is greater than N, the $|S_c| - N$ solutions are truncated according to the crowding degree. When the number of solutions with the $Fit < 1$ is equal to N, these solutions are directly selected for updating CA.

Algorithm 3: UpdateCA.

Input: archive: CA; current population: P; population size: N
Output: updated archive: CA

```
1  P_t ← CA ∪ P ;
2  Fit_t ← Fitness of P_t by Equation (7) based on Pareto dominance ;
3  S_c ← index of solutions which Fit_t < 1 ;
4  if |S_c| < N then
5      R ← sort Fit_t ;
6      S_c ← R(1 : N) ;
7  if |S_c| > N then
8      S_c ← use truncation method to select N solutions from S_c ;
9  CA ← P_t(S_c) ;
10 return CA;
```

3.5. Update strategy of FA

Algorithm 4: UpdateFA: stage 1.

Input: archive: FA; current population: P; population size: N;
Output: updated archive: FA;

```
1  P_t ← FA ∪ P ;
2  Fit_t ← Calculate the Fitness of P_t by Equation (7) based on CDP ;
3  S_c ← index of solutions which Fit_t < 1 ;
4  if |S_c| < N then
5      R ← sort Fit_t ;
6      S_c ← P_t(R(1 : N)) ;
7  if |S_c| > N then
8      S_c ← use truncation method select N solutions from S_c;
9  FA ← P_t(S_c) ;
10 return FA ;
```

In the stage one, FA collects highly feasible solutions obtained during the evolutionary process. Subsequently, in the stage two, FA conducts a localized search around these solutions discovered in the initial stage, progressively approaching the CPF from the feasible side. The update strategy of FA in stage one is similar to that of Algorithm 3 with the only difference being step 2 (line 2), where the fitness of P_t is calculated using CDP instead of Pareto dominance.

In stage two, FA employs a niche-based update strategy designed to generate an angle-based niche and select individuals within it to preserve adequate diversity and sufficient selection pressure. Only one solution is chosen in each niche. To establish the angle-based niche, we first need to compute the angle between solutions situated on the UPF, which is expressed as follows:

$$Angle(\mathbf{x},\mathbf{y}) = arccos\left(\frac{F(\mathbf{x}) \bullet F(\mathbf{y})}{\|F(\mathbf{x})\|_2 \cdot \|F(\mathbf{y})\|_2}\right) \tag{9}$$

Equation (9) calculates the angle between two solutions $\mathbf{x}$ and $\mathbf{y}$ in the objective space, where $F(\mathbf{x}) \bullet F(\mathbf{y})$ represents the inner product of solutions $\mathbf{x}$ and $\mathbf{y}$. The niche size determines the number and location of solutions falling into it, which has a significant impact on the selection pressure and diversity. In order to provide sufficient selection pressure for evolution and ensure diversity, the niche size A_n is adaptively set to the maximum angle between individuals in CA, which is shown as follows:

$$A_n = \{\max_{\mathbf{x},\mathbf{y}\in CA, \mathbf{x}\neq\mathbf{y}} Angle(\mathbf{x},\mathbf{y})\} \tag{10}$$

when the angle between a candidate solution $\mathbf{y}$ and a given solution $\mathbf{x}$ on UPF is smaller than the A_n, $\mathbf{y}$ is considered to be located within the niche of $\mathbf{x}$. $\theta_{\mathbf{cp}}(t)$ represents the set of solutions P located within the niche of tth solution in CA, i.e., $\mathbf{x}_t^c$. The equation as shown below:

$$\theta_{cp}(t) = \left\{\mathbf{y} \in P \mid \underset{\mathbf{x}_t^c\in CA}{Angle}(\mathbf{x}_t^c,\mathbf{y}) < A_n\right\} \tag{11}$$

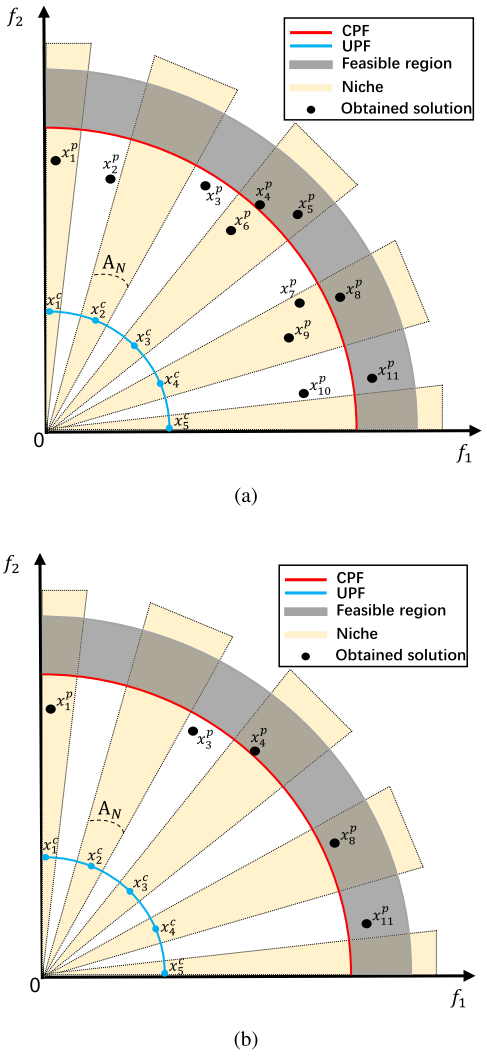


Fig. 1. Illustration of the update strategy of FA in stage 2.

To provide further clarification on the update strategy of FA in stage two, we present the main concept in Fig. 1. As depicted in Fig. 1(a), $\mathbf{x}_t^c$ represents the UPF identified by CA during stage one, which is utilized for niche creation. The angle between two solutions in CA is computed using Equation (9), with the maximum value being assigned as the niche size A_n. Subsequently, based on Equation (11), $\mathbf{x}_i^p$ denotes candidate solutions that are allocated into distinct niches. For instance, the angle between $\mathbf{x}_1^c$ and $\mathbf{x}_1^p$ is smaller than A_n, indicating that $\mathbf{x}_1^p$ is positioned within the niche of $\mathbf{x}_1^c$. Consequently, we define $\theta_{cp}(1)$ as a set containing only $\mathbf{x}_1^p$. Similarly, for $\theta_{cp}(2)$, it remains empty. However, for $\theta_{cp}(3)$, it comprises of the elements: $\{\mathbf{x}_4^p, \mathbf{x}_5^p, \mathbf{x}_6^p\}$ and so forth. The selection is conducted on each niche to update FA, considering the number of individuals located within the same niche, there are three cases: In case one, where only one solution is located within the niche (e.g., $\theta_{cp}(1) = \{\mathbf{x}_1^p\}$. Regardless of its feasibility, it is chosen for updating FA. This is because $\mathbf{x}_1^p$ retains diversity information as an isolated individual, which may contribute to enhancing the integrity of the CPF. In case two, when multiple candidate solutions exist within a single niche (e.g., $\theta_{cp}(3) = \{\mathbf{x}_4^p, \mathbf{x}_5^p, \mathbf{x}_6^p\}$), a feasibility priority selection is employed. The best individual within the niche is selected for updating FA. Therefore, x_4^p is selected. For case three, there exist no solutions within the niche, such as when $\theta_{cp}(2) = \emptyset$. As the FA serves as a feasibility-orientated archive, we choose two individuals with angles closest to $\mathbf{x}_{c2}$, denoted as $\mathbf{x}_2^p$ and $\mathbf{x}_3^p$. Among them, we select the one with the best feasibility, namely $\mathbf{x}_3^p$, to ensure that the algorithm explores the feasible region in this approximation direction. The selected individuals in this iteration are illustrated in Fig. 1(b). The pseudo code for updating strategy of FA in stage two is presented in Algorithm 5.

Algorithm 5: UpdateFA: stage 2.

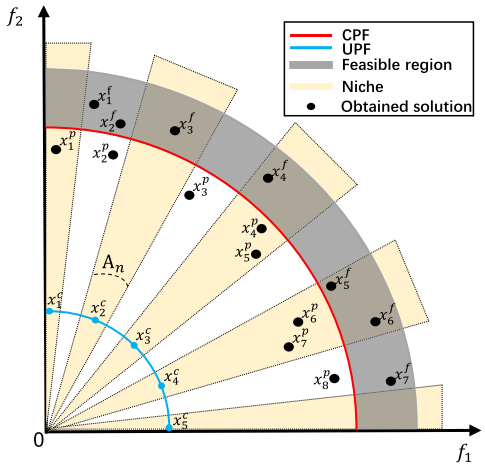
Input: archive: CA, FA; offspring: O; population size: N; search stage: $stage$
Output: updated archive: FA;

$A_{ca} \leftarrow \underset{\mathbf{x},\mathbf{y}\in CA,\mathbf{x}\neq\mathbf{y}}{Angle}(\mathbf{x},\mathbf{y})$;
$A_n \leftarrow$ the maximum element of A_{ca} by Equation (10);
$P \leftarrow FA \cup O_t$;
$A_{cp} \leftarrow \underset{\mathbf{x}\in CA,\mathbf{y}\in P}{Angle}(\mathbf{x},\mathbf{y})$;
$\theta_{cp} \leftarrow$ solutions in P located within the niche of CA by Equation (11) ;
foreach $i = 1, 2, \ldots, N$ **do**
  if $|\theta_{cp}(i)| = 1$ **then**
    $FA(i) \leftarrow \theta_{cp}(i)$;
  if $|\theta_{cp}(i)| > 1$ **then**
    $FA(i) \leftarrow \underset{\mathbf{x}\in\theta_{cp}(i)}{argmin}\, Fit_c(\mathbf{x})$;
  if $|\theta_{cp}(i)| = 0$ **then**
    $\mathbf{c}_1, \mathbf{c}_2 \leftarrow \underset{\mathbf{c}\in P}{argmin}\, Angle(\mathbf{x}_i \in CA \mid \mathbf{x}_i, \mathbf{c})$;
    $FA(i) \leftarrow \underset{\mathbf{x}\in\{\mathbf{c}1,\mathbf{c}2\}}{argmin}\, Fit_c(\mathbf{x})$
return FA ;

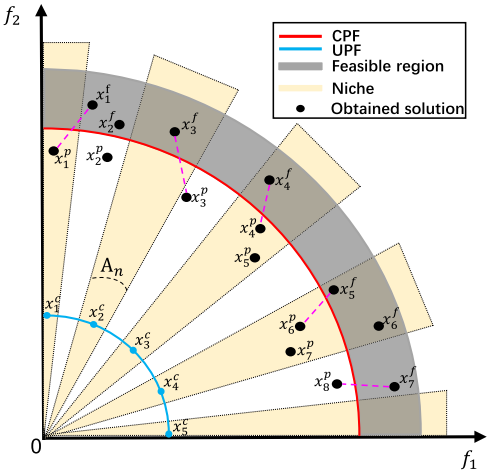
3.6. Infeasible solution utilization

In order to explore more viable regions and enhance the convergence of existing feasible solutions, a niche-based strategy for utilizing infeasible solutions is adopted in stage two. In this strategy, the main population P undergoes ϵ-constrained environment selection, gradually approaching CPF from the infeasible side. Meanwhile, FA updates itself based on feasibility priority, approaching CPF from the feasible side. Niches are employed to determine the positional relationship between feasible and infeasible solutions within the same approximation direction. This allows for matching a feasible solution with optimal convergence to an infeasible solution with the lowest CV within the same niche. Subsequently, an offspring generation operator is applied to these matched solutions, generating offspring that have potential for exploring unknown feasible regions and improving the convergence of feasible solutions.

The mechanism of the infeasible solution utilization is illustrated in Fig. 2. As shown in Fig. 2(a), since FA is a feasibility-oriented archive, solutions within FA, denoted as x_i^f, are mostly located within feasible region. Conversely, solutions within main population P, denoted as x_i^p are located in infeasible region, approaching CPF from infeasible side. Similar to the update strategy of FA in stage two, the solutions $\mathbf{x}_i^c$ in CA are used to create angle-based niches, with a maximum angle between $\mathbf{x}_i^c$ set as niche size A_n. Afterwards, $\mathbf{x}_i^p$ have been assigned to distinct niches based on their angular relationships with respect to $\mathbf{x}_i^c$. We denote $\theta_{cp}(t)$ as the collection of $\mathbf{x}_i^p$ within the niche of $\mathbf{x}_t^c$, where for instance, if the angle between $\mathbf{x}_3^c$ and both $\mathbf{x}_4^p$ and $\mathbf{x}_5^p$ is smaller than A_n, then we have $\theta_{cp}(3) = \{\mathbf{x}_4^p, \mathbf{x}_5^p\}$. Similarly, we define $\theta_{cf}(t)$ as the collection of vectors from FA that fall within the niche of vector $\mathbf{x}_t^{(c)}$, thereby characterizing the set of feasible solutions for a given context. Given our objective to select suitable paired parents from both P and FA for offspring generation, there are two cases in which a parent can be selected from FA. For case one, multiple solutions exist within the niche of x_t^c. Given that FA prioritizes feasibility, enhancing convergence becomes crucial as a majority of its individuals are feasible. The parent is selected based on the minimal sum of objective vectors since it can roughly indicate the convergence for feasible solutions in the same niche. For instance, consider the niche of x_4^c, where $\theta_{cf}(4) = \{x_5^f, x_6^f\}$. As the sum of objective vectors for x_5^f is lower than that of x_6^f, FA selects x_5^f as the parent in this niche. In case two, no solution from FA exists within the niche of x_t^c, and thus, the individual in FA with the



(a)



(b)

Fig. 2. Illustration of the infeasible solution utilization strategy.

smallest angle to x_i^c is chosen. This operation is designed to ensure the exploration of CPF in the approximate direction of x_i^c. Taking the niche of x_1^c as an illustrative example, it can be observed that there exists no solution of FA within the niche. Consequently, the individual closest to this niche, denoted as x_1^f, is selected.

The parent selection strategy for individuals in P follows a similar concept, with the only distinction being that when multiple individuals of P exist within the same niche. Since P evolves from the UPF and approaches the CPF from an infeasible side, it primarily comprises infeasible individuals with excellent convergence. Consequently, it is preferable to choose an individual with lower constraint violation as a mating parent to generate offspring with improved feasibility. For instance, considering the niche of $\mathbf{x}_4^c$, where $\theta_{cp}(4) = \{x_6^p, x_7^p\}$ and x_6^p exhibits a lower constraint violation compared to x_7^p, we select x_6^p. The matched parents from FA and P are illustrated in Fig. 2(b). Please refer to Algorithm 6 for the corresponding pseudo code.

3.7. Further discussion

In this article, the niche technique is introduced to provide valuable insights into population evolution. By employing angle-based niches, we can identify small regions in the objective space that share similar directions, thereby facilitating the partitioning of the population based on these regions. Individuals assigned to the same niche can be considered as exploring similar directions in the objective space. To strike a balance between algorithm performance and exploration of more feasible regions, we propose a method of niche-based environment selection and utilization of infeasible solutions. Specifically, the functionality of niche-based environment selection and infeasible solution utilization can be summarized as follows:

1. In the stage 2 update strategy of FA, a niche-based environmental selection strategy is employed to choose candidate feasible individuals. This approach aims at selecting the most feasible individual within a niche where candidate individuals congregate, thereby enhancing the overall feasibility of the population. In instances where niches are sparsely populated with candidates, this strategy ensures exploration in an approximate direction by selecting the individual closest to that trajectory, thus augmenting population diversity.
2. In the utilization strategy of infeasible solutions, angle-based niches are employed to provide directional information. Feasible and infeasible solutions that lie approximately in the same direction are paired together for generating offspring individuals. These offspring individuals possess the potential to explore the CPF along this approximate direction, thereby enhancing both convergence and diversity of feasible solutions and facilitating the attainment of a complete CPF.

As mentioned above, angle-based niche can preserve diversity information and provide reasonable selection pressure for the environmental selection, which is crucial for balancing the convergence, diversity, and feasibility of the population. In addition, it can also provide positional information for utilizing infeasible solutions and guide the population to explore unknown CPF, which is particularly important for improving the overall performance of NAC-MOEA.

Algorithm 6: InfeasibleUtilization.

Input: archive: CA; archive: FA; main population: P; population size: N
Output: paired solutions: NoF, NoP

$A_{ca} \leftarrow \underset{\mathbf{x},\mathbf{y}\in CA,\mathbf{x}\neq\mathbf{y}}{Angle}(\mathbf{x},\mathbf{y})$;
$A_n \leftarrow$ the Nth element of A_{ca} by Equation (10);
$A_{cp} \leftarrow \underset{\mathbf{x}\in CA,\mathbf{y}\in P}{Angle}(\mathbf{x},\mathbf{y})$;
$A_{cf} \leftarrow \underset{\mathbf{x}\in CA,\mathbf{y}\in FA}{Angle}(\mathbf{x},\mathbf{y})$;
$\theta_{cp} \leftarrow$ solutions of P located within the niche of CA by Equation (11);
$\theta_{cf} \leftarrow$ solutions of FA located within the niche of CA by Equation (11);
foreach $i = 1, 2, \ldots, N$ **do**
  if $|\theta_{cf}(i)| \geq 1$ **then**
    $NoF(i) \leftarrow \underset{\mathbf{x}\in\theta_{cf}(i)}{argmin} \sum_{i=1}^{m} f_i(\mathbf{x})$;
  if $|\theta_{cf}(i)| = 0$ **then**
    $NoF(i) \leftarrow \underset{\mathbf{x}\in FA}{argmin}\, Angle(\mathbf{x}_i^c, \mathbf{x})$;
  if $|\theta_{cp}(i)| \geq 1$ **then**
    $NoP(i) \leftarrow \underset{\mathbf{x}\in\theta_{cp}(i)}{argmin}\, \Phi(\mathbf{x})$;
  if $|\theta_{cp}(i)| = 0$ **then**
    $NoP(i) \leftarrow \underset{\mathbf{x}\in P}{argmin}\, Angle(\mathbf{x}_i^c, \mathbf{x})$;
return NoF, NoP

4. Experimental setup

In this section, we initially provide an overview of the experiment's various components, including benchmark problems, performance metrics, parameter settings, and advanced CMOEAs for comparison. Subsequently, we employ the proposed algorithm to address distinct CMOPs and contrast the statistical results with other advanced CMOEAs to assess the performance of our proposed method.

4.1. Benchmark suits and CMOEAs for comparison

Two well-known benchmark suits MW [18] and LIR-CMOP [7] were chosen as test problem for the empirical study. MW notable for its close approximation to the real CMOPs and extensive coverage of different problem types, while LIR-CMOPs stand out due to their large infeasible region and significant difficulty in solving. To evaluate the performance of NACOMEA, two typical CMOEAs, C-NSGAII-CDP [5] and C-MOEA/D [9], and five advanced CMOEAs, CMOEA-MS [31], C-TAEA [14], ToP [17], DC-NSGAIII [11], TiGE-2 [38] were selected for peer comparison. All experiments in this paper were conducted using PlatEMO [35].

4.2. Performance metrics

In this section, two comprehensive performance evaluation metrics, namely inverted generational distance (IGD) [4] and hypervolume (HV) [42], are introduced to compare the performance of the proposed algorithm with that of existing advanced algorithms.

1. IGD mainly evaluates the convergence and diversity performance of the algorithm by calculating the average of the minimum distance sum between each point on the real Pareto front and the solution set obtained by the algorithm. Suppose PF is the real Pareto front of the problem and PF^* is the Pareto front found by the algorithm, the IGD of proposed algorithm is calculated as follows:
$$IGD = \frac{\sum_{\mathbf{x}\in PF} d_{ed}(\mathbf{x}, PF^*)}{|\,PF\,|} \tag{12}$$
where $d_{ed}(\mathbf{x}, PF^*)$ represents the minimum Euclidean distance between $\mathbf{x}$ and PF^*. The smaller IGD value indicates the better performance of the algorithm.
2. HV assesses the comprehensive performance of the algorithm by determining the hypervolume value of the space enclosed by the non-dominated solution set and the reference point. The HV of a solution set can be calculated as follows:
$$HV = \lambda\left(\bigcup_{i=1}^{|S|} v_i\right) \tag{13}$$
where λ represents the Lebesgue measure, v_i denotes the hypervolume formed by the reference point and the non-dominated solution, and S is the non-dominated solution set. The larger HV indicates the better comprehensive performance of the algorithm.

It is necessary to adequately sample the true PF of test problem for calculating IGD. According to the approach of [29], about 10000 sample points are sampled on the real PF. To calculate HV, the reference point of the output population is set to $(1, \ldots, 1)$, and the point $(1.1 * z^{nad})$ is used to normalize the objective values.

4.3. Parameter setting

DE operator is used to generate offspring in PPS [8], while genetic operator is used in all other compared CMOEAs.

Simulated binary crossover (SBX) [1] and polynomial mutation (PM) [5] are used as the genetic operator for CMOEAs that use genetic operator as the evolutionary operator, with the following parameter settings:

1. Crossover probability $p_c = 1$ and distribution index $\eta_c = 20$.
2. Mutation probability $p_m = 1/n$ and distribution index $\eta_m = 20$.

For CMOEAs which adopt DE as the evolutionary operator, the parameters CR and F in DE operator are set to 1 and 0.5, respectively. The population size N is set to 100. The number of decision variable D in all CMOEAs is set to 10. For a fair comparison. The maximal number of function evaluations $MaxFE$ is set to 200000 for LIR-CMOP benchmark suite, and 80000 for MW benchmark suite. All CMOEAs apply these settings for a fair comparison, the other parameter settings of all CMOEAs in comparison are the same as suggested in their corresponding original articles.

The parameters for the ϵ-constrained technique in NACMOEA are as follows. τ is 0.05; α is 0.95; T_c is $(0.9 * MaxFE)/N$ and cp is 2.

The parameters for compute the search stage are set as follows. The generation gap G_p is 20; the coverage rate threshold C_t is 0.95.

The aforementioned parameter configurations remained constant throughout the experimental procedure.

Each algorithm runs 30 times independently on each test function, and the mean and standard deviation of IGD and HV values are presented in corresponding tables. In particular, the best results are highlighted in boldface. To have a statistically sound conclusion, we use the Wilcoxon's rank sum test at a significant level of 5% to validate the significance of the better performance achieved by the proposed NACMOEA with respect to the other compared algorithms.

4.4. Analysis of experimental results

In this section, the performance of NACMOEA is compared with other eight elite CMOEAs mentioned earlier on LIR-CMOP and MW benchmark CMOPs.

1) Comparisons on LIR-CMOP suite: There are a total of 14 CMOPs within the LIR-CMOP benchmark suite. The experimental results based on the IGD are presented in Table 1. The analysis reveals that NACMOEA demonstrates the most superior performance on 9 test problems. In comparison, CMOEA-MS and DCNSGAIII exhibit exceptional performance on 3 and 2 test problems, respectively. However, C-MOEA/D, C-NSGAII-CDP, C-TAEA, ToP, and TiGE2 failed to achieve the best results on any of the test problems. To be explicit, the proposed NACMOEA demonstrates best performance on LIR-CMOP1-LIR-CMOP6, LIR-CMOP8, and LIR-CMOP10-LIR-CMOP11, indicating that NACMOEA is effective in addressing scenarios where UPF and CPF are separated and the feasible region is narrow (LIR-CMOP1-LIR-CMOP4). In instance that the feasible region is obstructed by a large area of infeasible region (LIR-CMOP5-LIR-CMOP6, LIR-CMOP8), NACMOEA is also capable of crossing through a multitude of infeasible regions and reaching the CPF. Furthermore, NACMOEA can cope with cases where CPF and UPF partially overlap and the feasible region is narrow and isolated (LIR-CMOP10-LIR-CMOP11). Table 2 displays the HV result of eight compared CMOEAs on LIR-CMOP1-LIR-CMOP14. As illustrated, NACMOEA exhibits its superior performance on eight CMOPs and competitive against the other two CMOPs.

Fig. 3 illustrates the populations with the median IGD obtained from the eight CMOEAs on LIR-CMOP11, whose feasible region is isolated and narrow. It can be discerned that some solutions of C-MOEA/D, C-TAEA, and TiGE-2 fail to converge to the CPF, demonstrating insufficient convergence capability. While the populations of C-NSGAII-CDP, CMOEA-MS, and ToP fail to maintain good diversity, which results in the failure to obtain a complete CPF. The populations of DCNSGAIII and NACMOEA exhibit an optimal balance between convergence, diversity, and feasibility, with all solutions within these populations traversing the infeasible region, thus achieving the complete CPF.

2) MW benchmark suite comparisons: The MW benchmark suite contains a total of 14 CMOPs. Table 3 presents the experimental results of NACMOEA and other advanced CMOEAs on the MW based on IGD. The proposed NACMOEA demonstrates superior performance in terms of MW and attains the most favorable outcomes across 6 test problems. Meanwhile, C-TAEA excels in 5 test problem solutions. Both CMOEA-MS and DCNSGAIII exhibit equivalent efficiency on a single test problem. However, C-NSGAII-CDP, ToP, and TiGE-2 fail to achieve optimal results in any test problems.

Table 1
Statistical results of IGD obtained by C-MOEA/D, C-NSGAII-CDP, C-TAEA, CMOEA-MS, DCNSGAIII, TOP, TiGE2 and the proposed NACMOEA on the LIR-CMOP benchmark suite. Best result in each row is highlighted.

Problem	M	C-MOEA/D	C-NSGAII-CDP	C-TAEA	CMOEA-MS	DCNSGAIII	ToP	TiGE-2	NACMOEA
LIRCMOP1	2	1.6569e-1 (5.04e-2) -	2.3454e-1 (6.23e-2) -	2.2888e-1 (1.34e-1) -	3.9355e-1 (1.46e-1) -	3.7499e-2 (1.67e-2) -	1.2399e-1 (8.86e-2) -	9.5208e-2 (4.30e-2) -	1.8245e-2 (4.95e-3)
LIRCMOP2	2	1.2404e-1 (3.42e-2) -	1.6739e-1 (5.47e-2) -	8.3119e-2 (3.46e-2) -	3.2164e-1 (1.08e-1) -	4.3175e-2 (1.57e-2) -	1.6413e-1 (9.70e-2) -	1.0882e-1 (2.59e-2) -	2.2810e-2 (5.00e-3)
LIRCMOP3	2	1.7927e-1 (6.05e-2) -	2.3718e-1 (8.16e-2) -	2.9440e-1 (1.91e-1) -	3.6344e-1 (1.21e-1) -	4.8155e-2 (3.11e-2) -	3.6246e-1 (8.21e-2) -	8.7310e-2 (2.50e-2) -	2.6055e-2 (1.39e-2)
LIRCMOP4	2	1.7869e-1 (5.62e-2) -	2.3887e-1 (7.25e-2) -	1.7807e-1 (1.09e-1) -	2.8070e-1 (8.74e-2) -	3.8080e-2 (1.61e-2) -	3.1630e-1 (5.20e-2) -	1.0289e-1 (2.52e-2) -	2.7916e-2 (1.04e-2)
LIRCMOP5	2	7.1368e-1 (5.33e-1) -	6.5485e-1 (5.22e-1) -	9.2411e-2 (2.14e-2) -	1.9767e-2 (2.38e-2) ≈	2.6591e-2 (1.58e-2) -	1.4394e-1 (3.55e-1) ≈	3.2770e-1 (1.06e-1) -	1.4251e-2 (1.34e-2)
LIRCMOP6	2	8.2848e-1 (6.04e-1) -	5.6214e-1 (5.39e-1) -	1.3693e-1 (1.29e-1) -	1.4576e-2 (3.75e-2) ≈	6.3814e-2 (2.46e-1) -	4.3586e-2 (8.31e-2) ≈	4.4798e-1 (2.01e-1) -	1.1531e-2 (1.44e-2)
LIRCMOP7	2	7.0524e-2 (2.93e-2) -	2.2115e-2 (2.82e-2) -	2.0159e-2 (7.50e-3) -	7.0537e-3 (6.80e-4) +	1.2531e-2 (2.73e-3) -	8.6212e-3 (3.08e-4) +	1.5998e-1 (5.01e-2) -	9.1088e-3 (2.73e-3)
LIRCMOP8	2	8.1395e-2 (4.07e-2) -	2.6587e-2 (3.53e-2) -	2.7893e-2 (4.12e-2) -	1.4851e-2 (2.98e-2) -	1.3797e-2 (4.24e-3) -	2.0973e-2 (6.73e-2) -	3.1627e-1 (1.34e-1) -	8.8445e-3 (4.59e-3)
LIRCMOP9	2	4.0501e-1 (8.67e-2) -	4.3878e-1 (1.33e-1) -	6.8170e-2 (3.36e-2) ≈	2.5178e-1 (1.35e-1) -	1.3509e-2 (2.24e-2) ≈	3.3808e-1 (9.57e-2) -	7.1003e-1 (2.51e-1) -	7.4281e-2 (6.86e-2)
LIRCMOP10	2	1.9883e-1 (6.37e-2) -	3.0191e-1 (1.03e-1) -	7.8340e-2 (6.72e-2) -	8.0013e-2 (5.69e-2) -	9.1694e-3 (1.49e-2) -	5.5883e-3 (3.20e-4) -	4.8089e-1 (6.05e-2) -	4.8836e-3 (1.44e-4)
LIRCMOP11	2	2.1262e-1 (1.15e-1) -	1.9104e-1 (1.49e-1) -	1.3390e-1 (3.97e-2) -	8.0190e-2 (4.66e-2) -	5.3132e-3 (1.30e-2) -	9.8869e-2 (6.27e-2) -	5.0013e-1 (8.55e-2) -	2.4242e-3 (5.43e-5)
LIRCMOP12	2	1.5644e-1 (6.35e-2) -	1.2374e-1 (5.54e-2) -	1.9280e-2 (6.95e-3) -	6.0949e-2 (5.06e-2) -	3.6273e-3 (1.04e-3) +	1.8157e-2 (3.95e-2) -	5.0892e-1 (3.10e-1) -	1.0385e-2 (2.51e-2)
LIRCMOP13	3	9.2899e-2 (1.77e-7) +	1.5868e-1 (2.20e-1) -	1.0855e-1 (2.50e-3) -	9.2582e-2 (9.38e-4) +	9.2899e-2 (7.55e-7) +	1.2742e-1 (3.96e-3) -	3.5011e-1 (6.96e-2) -	9.5820e-2 (1.28e-3)
LIRCMOP14	3	9.5314e-2 (1.14e-6) +	1.2325e-1 (5.39e-3) -	1.1158e-1 (1.02e-3) -	9.4933e-2 (1.14e-3) +	9.5555e-2 (1.18e-4) +	1.1987e-1 (2.93e-3) -	5.8779e-1 (3.35e-1) -	9.7403e-2 (1.06e-3)
+/-/≈		2/12/0	0/14/0	0/13/1	3/9/2	3/10/1	1/11/2	0/14/0	

Table 2
Statistical results of HV obtained by C-MOEA/D, C-NSGAII-CDP, C-TAEA, CMOEA-MS, DCNSGAIII, TOP, TiGE2 and the proposed NACMOEA on the LIR-CMOP benchmark suite. Best result in each row is highlighted.

Problem	M	C-MOEA/D	C-TAEA	CMOEA-MS	DCNSGAIII	ToP	TiGE-2	C-NSGAII-CDP	NACMOEA
LIRCMOP1	2	1.6276e-1 (1.69e-2) -	1.3963e-1 (3.58e-2) -	1.1314e-1 (2.83e-2) -	2.2456e-1 (6.91e-3) -	1.8180e-1 (3.84e-2) -	1.9850e-1 (1.22e-2) -	1.3970e-1 (2.02e-2) -	2.3362e-1 (1.81e-3)
LIRCMOP2	2	2.8900e-1 (1.86e-2) -	3.1564e-1 (2.19e-2) -	2.0975e-1 (4.51e-2) -	3.4092e-1 (5.59e-3) -	2.8395e-1 (4.51e-2) -	3.1786e-1 (1.18e-2) -	2.6745e-1 (2.84e-2) -	3.5198e-1 (2.05e-3)
LIRCMOP3	2	1.3679e-1 (1.99e-2) -	1.1618e-1 (3.97e-2) -	9.7910e-2 (2.27e-2) -	1.9494e-1 (8.75e-3) -	9.3626e-2 (1.74e-2) -	1.7208e-1 (1.03e-2) -	1.2359e-1 (2.12e-2) -	2.0253e-1 (3.66e-3)
LIRCMOP4	2	2.3706e-1 (2.48e-2) -	2.2814e-1 (4.06e-2) -	1.9664e-1 (3.95e-2) -	3.0198e-1 (5.30e-3) -	1.8434e-1 (2.69e-2) -	2.7634e-1 (1.07e-2) -	2.1412e-1 (3.01e-2) -	3.0621e-1 (4.64e-3)
LIRCMOP5	2	1.0112e-1 (1.11e-1) -	2.5760e-1 (1.16e-2) -	2.8525e-1 (8.51e-3) ≈	2.8004e-1 (7.44e-3) -	2.5323e-1 (9.49e-2) -	1.7679e-1 (3.47e-2) -	1.1451e-1 (1.26e-1) -	2.8507e-1 (9.27e-3)
LIRCMOP6	2	6.9360e-2 (8.13e-2) -	1.4987e-1 (3.89e-2) -	1.9423e-1 (8.45e-3) ≈	1.8518e-1 (3.61e-2) -	1.8121e-1 (3.31e-2) -	8.4661e-2 (2.31e-2) -	9.7847e-2 (7.55e-2) -	1.9443e-1 (4.98e-3)
LIRCMOP7	2	2.6636e-1 (1.01e-2) -	2.8780e-1 (4.02e-3) -	2.9472e-1 (5.83e-4) +	2.9097e-1 (1.54e-3) -	2.9389e-1 (1.66e-4) ≈	2.3614e-1 (1.29e-2) -	2.8739e-1 (1.17e-2) -	2.9322e-1 (1.68e-3)
LIRCMOP8	2	2.6496e-1 (1.15e-2) -	2.8719e-1 (1.03e-2) -	2.9161e-1 (1.19e-2) -	2.9058e-1 (1.94e-3) -	2.9146e-1 (1.28e-2) -	1.8754e-1 (3.98e-2) -	2.8844e-1 (1.01e-2) -	2.9342e-1 (3.15e-3)
LIRCMOP9	2	4.2860e-1 (4.70e-2) -	5.2607e-1 (1.45e-2) -	4.7208e-1 (5.78e-2) -	5.6258e-1 (5.92e-3) ≈	4.6686e-1 (2.87e-2) -	3.0050e-1 (9.71e-2) -	4.0757e-1 (8.28e-2) -	5.4903e-1 (2.28e-2)
LIRCMOP10	2	5.9514e-1 (3.47e-2) -	6.6790e-1 (2.78e-2) -	6.7741e-1 (2.60e-2) -	7.0415e-1 (5.60e-3) -	7.0737e-1 (1.67e-4) -	4.2773e-1 (2.44e-2) -	5.5529e-1 (6.03e-2) -	7.0813e-1 (8.07e-5)
LIRCMOP11	2	6.2555e-1 (8.43e-2) -	6.4309e-1 (1.33e-2) -	6.5930e-1 (2.67e-2) -	6.9289e-1 (4.59e-3) -	6.3190e-1 (4.18e-2) -	4.1228e-1 (3.89e-2) -	5.7907e-1 (1.10e-1) -	6.9405e-1 (5.58e-6)
LIRCMOP12	2	5.8330e-1 (2.66e-2) -	6.1068e-1 (2.95e-3) -	5.9785e-1 (2.25e-2) -	6.2001e-1 (4.64e-4) +	6.1335e-1 (1.87e-2) -	4.1179e-1 (1.16e-1) -	5.6721e-1 (2.72e-2) -	6.1745e-1 (1.03e-2)
LIRCMOP13	3	5.5962e-1 (8.49e-7) +	5.4685e-1 (1.50e-3) -	5.5613e-1 (1.79e-3) +	5.5962e-1 (2.18e-6) +	5.1327e-1 (3.93e-3) -	4.5642e-1 (2.88e-2) -	5.1311e-1 (9.70e-2) -	5.5036e-1 (1.55e-3)
LIRCMOP14	3	5.5957e-1 (1.24e-5) +	5.4635e-1 (8.91e-4) -	5.5500e-1 (1.04e-3) +	5.5856e-1 (4.15e-4) +	5.2808e-1 (3.80e-3) -	3.6538e-1 (1.16e-1) -	5.3038e-1 (4.50e-3) -	5.5165e-1 (1.49e-3)
+/-/≈		2/12/0	0/14/0	3/9/2	3/10/1	0/13/1	0/14/0	0/14/0	

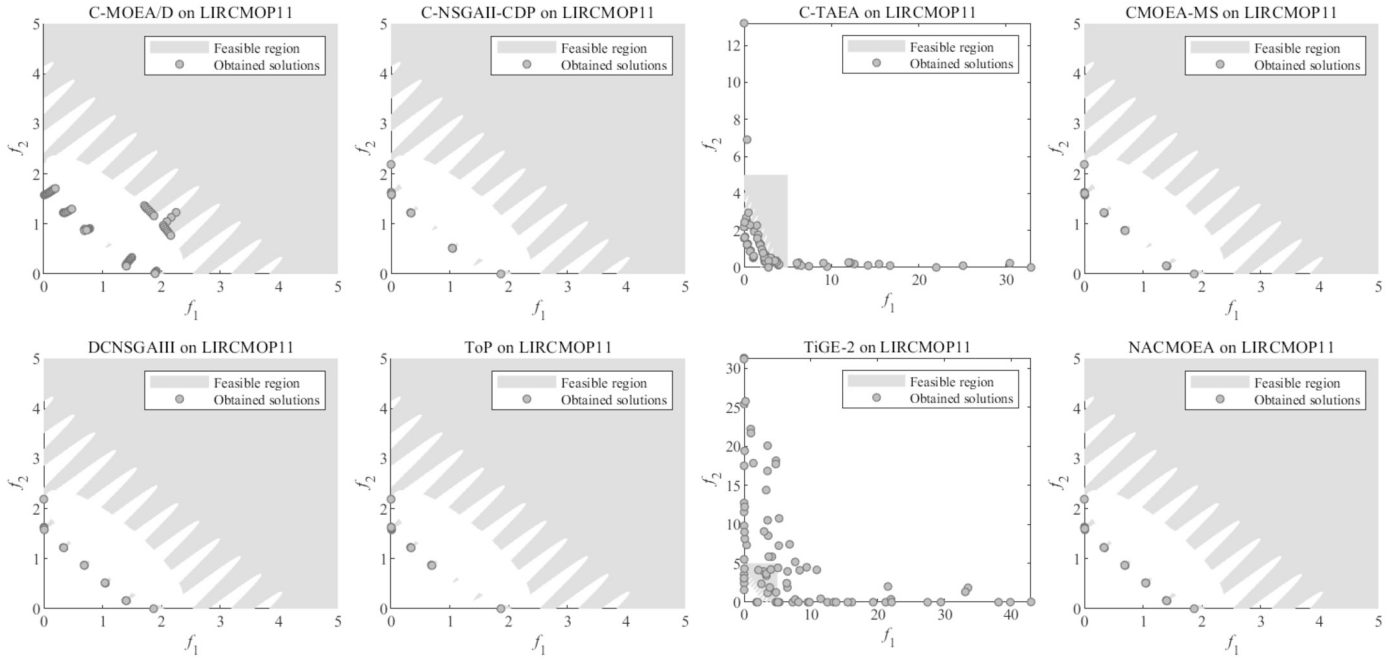


Fig. 3. Populations with median IGD obtained by C-MOEA/D, C-NSGAII-CDP, C-TAEA, CMOEA-MS, DCNSGAIII, MOEA/D-DAE, TOP, TiGE2 and the proposed NACMOEA on the LIR-CMOP11.

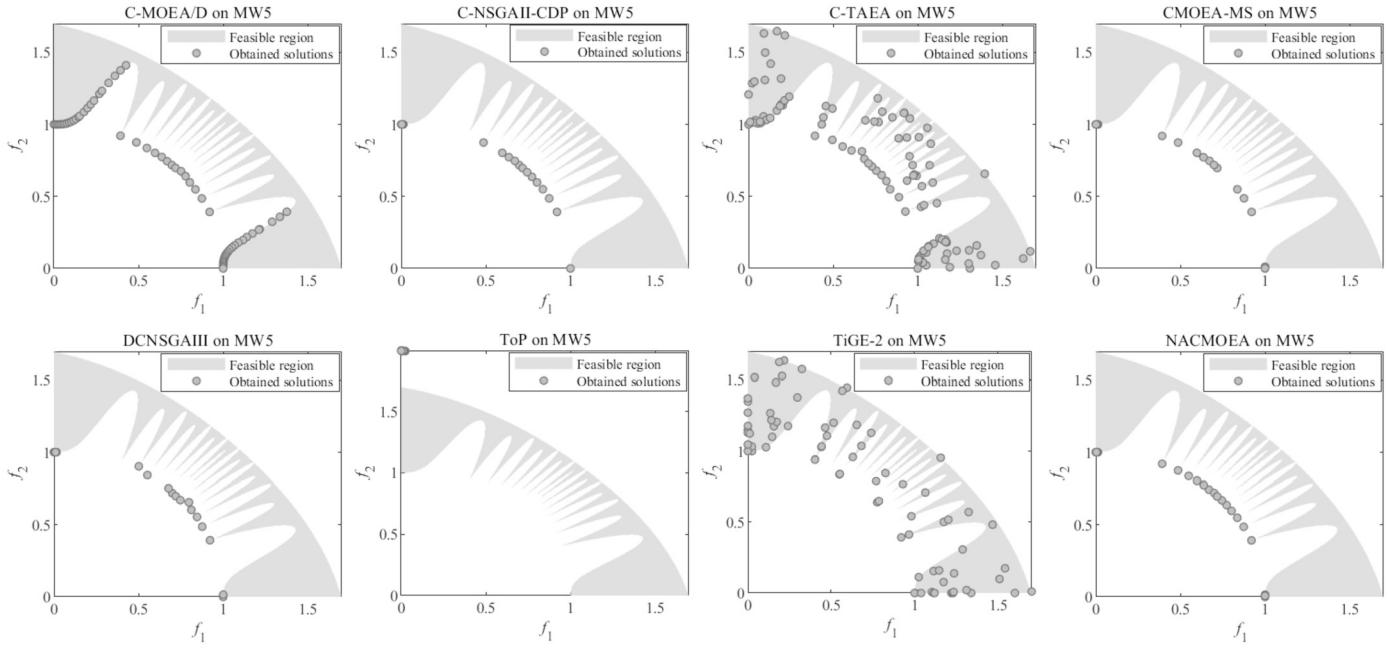


Fig. 4. Populations with median IGD obtained by C-MOEA/D, C-NSGAII-CDP, C-TAEA, CMOEA-MS, DCNSGAIII, MOEA/D-DAE, TOP, TiGE2 and the proposed NACMOEA on the MW5.

The rationale behind this phenomenon lies in the adoption of a feasibility-oriented search strategy by C-NSGAII-CDP and ToP, which excessively prioritizes constraint satisfaction at the expense of maintaining diversity during the search process. The convergence speed of TiGE-2 is relatively slow, with most individuals failing to converge near the CPF within an equivalent number of function evaluations. The feasible region of the MW test set is primarily characterized by disconnected or narrow traits, which makes it easy for CMOEA to get trapped into local feasible regions. NACMOEA exhibits superior performance on MW1, MW5, MW7, MW9, and MW11-MW12. Specifically, both MW1 and MW11 present a disconnected geometry of CPF. The feasible regions of MW5 and MW7 are connected; however, their CPFs are situated along narrow boundaries of the feasible region, posing a significant challenge to obtain. On the other hand, the CPF and UPF of both MW9 and MW12 are entirely separate. By effectively handling such complex CMOPs through the utilization of infeasible solutions, NACMOEA demonstrates its efficacy. Table 4 showcases the hypervolume (HV) results obtained by NACMOEA as well as eight other CMOEAs for comparison purposes on the aforementioned CMOPs within the context of this study. Overall, our proposed NACMOEA outperforms all others on six CMOPs while remaining competitive on the remaining four CMOPs.

Fig. 4 illustrates the experimental results of the eight CMOEAs on the MW5 problem, where the CPF of MW5 is discrete and located at the end

Table 3
Statistical results of IGD obtained by C-MOEA/D, C-NSGAII-CDP, C-TAEA, CMOEA-MS, DCNSGAIII, MOEA/D-DAE, TOP, TiGE2 and the proposed NACMOEA on the MW benchmark suite. Best result in each row is highlighted.

Problem	M	C-MOEA/D	C-NSGAII-CDP	C-TAEA	CMOEA-MS	DCNSGAIII	ToP	TiGE-2	NACMOEA
MW1	2	3.8561e-3 (1.75e-3) -	3.6776e-2 (4.47e-2) -	2.0128e-3 (7.42e-5) -	1.8543e-3 (9.90e-4) ≈	2.3465e-3 (6.75e-5) -	2.8692e-1 (1.90e-1) -	2.4913e-2 (3.19e-2) -	1.6795e-3 (2.09e-5)
MW2	2	1.2379e-2 (9.82e-3) ≈	1.6165e-2 (9.60e-3) ≈	8.9175e-3 (7.87e-3) +	1.5435e-2 (7.99e-3) ≈	8.6716e-3 (7.04e-3) +	9.2834e-2 (7.33e-2) -	7.9361e-2 (1.78e-1) -	1.4736e-2 (9.21e-3)
MW3	2	5.2770e-3 (4.16e-4) -	6.6536e-3 (4.99e-3) -	4.9632e-3 (2.17e-4) ≈	5.3178e-3 (3.12e-4) -	6.2302e-3 (7.46e-4) -	3.5980e-1 (4.10e-1) -	2.2675e-2 (3.60e-3) -	5.0381e-3 (1.74e-4)
MW4	3	4.1180e-2 (1.87e-4) +	5.5181e-2 (2.98e-3) -	4.6595e-2 (3.00e-4) -	4.1859e-2 (3.89e-4) +	4.1722e-2 (7.26e-4) +	2.7012e-1 (1.58e-1) -	9.2694e-2 (3.83e-2) -	4.2366e-2 (4.69e-4)
MW5	2	1.8168e-2 (7.31e-2) -	6.2432e-2 (1.86e-1) -	1.4708e-2 (2.70e-3) -	2.3237e-2 (2.50e-2) -	2.2877e-2 (8.89e-3) -	6.8731e-1 (2.35e-1) -	5.6428e-2 (1.36e-2) -	1.1153e-3 (7.44e-4)
MW6	2	1.2610e-2 (8.93e-3) ≈	1.6776e-2 (9.91e-3) -	5.9709e-3 (4.09e-3) ≈	1.1695e-2 (7.85e-3) ≈	1.0195e-2 (8.90e-3) ≈	3.4238e-1 (2.50e-1) -	3.9546e-2 (1.24e-2) -	1.2809e-2 (1.09e-2)
MW7	2	4.6394e-3 (1.01e-4) -	1.9810e-2 (8.16e-2) -	7.1631e-3 (5.87e-4) -	2.3438e-2 (2.33e-2) ≈	5.8954e-3 (7.36e-4) -	1.1128e-1 (1.94e-1) -	3.7463e-2 (1.91e-2) -	4.3507e-3 (2.77e-4)
MW8	3	5.0244e-2 (2.45e-3) -	5.5203e-2 (4.57e-3) -	5.3037e-2 (1.63e-3) -	4.4060e-2 (2.08e-3) ≈	4.8387e-2 (1.44e-3) -	3.3577e-1 (3.37e-1) -	6.8464e-1 (1.44e-1) -	4.4152e-2 (1.35e-3)
MW9	2	7.3367e-3 (1.62e-3) -	8.8554e-3 (2.50e-3) -	8.4538e-3 (6.24e-4) -	6.7675e-2 (1.42e-1) -	1.1275e-2 (2.76e-3) -	3.7694e-1 (3.21e-1) -	3.2327e-2 (4.76e-3) -	4.5016e-3 (2.28e-4)
MW10	2	3.8720e-2 (4.60e-2) -	5.8309e-2 (4.41e-2) -	7.9136e-3 (7.04e-3) ≈	2.4414e-2 (2.24e-2) ≈	1.7853e-2 (1.51e-2) ≈	1.5442e-1 (7.43e-2) -	4.2238e-2 (3.22e-2) -	2.7905e-2 (3.67e-2)
MW11	2	5.6499e-2 (1.74e-1) -	1.9457e-1 (3.08e-1) -	1.5161e-2 (2.53e-3) -	6.3523e-3 (7.84e-4) -	8.2962e-3 (1.19e-3) -	5.5260e-1 (2.76e-1) -	3.0114e-2 (4.59e-3) -	5.9915e-3 (1.40e-4)
MW12	2	4.8073e-3 (7.73e-5) -	5.6128e-3 (2.31e-4) -	7.6779e-3 (5.78e-4) -	5.1780e-3 (2.48e-4) -	6.3666e-3 (4.10e-4) -	6.6322e-1 (2.53e-1) -	2.8330e-2 (5.06e-3) -	4.7568e-3 (1.03e-4)
MW13	2	4.6458e-2 (3.07e-2) ≈	7.7992e-2 (7.63e-2) -	2.2057e-2 (1.51e-2) ≈	5.8588e-2 (3.96e-2) -	4.5000e-2 (2.61e-2) ≈	4.3198e-1 (4.18e-1) -	4.3113e-1 (2.67e-1) -	4.3385e-2 (3.11e-2)
MW14	3	2.1173e-1 (1.01e-3) -	1.2433e-1 (6.04e-3) -	1.1007e-1 (3.07e-3) +	1.2893e-1 (1.87e-2) -	1.4194e-1 (3.76e-2) -	2.3854e-1 (1.86e-1) -	1.6169e-1 (1.08e-2) -	1.1840e-1 (5.38e-2)
+/-/≈		1/10/3	0/13/1	2/8/4	1/7/6	2/9/3	0/14/0	0/14/0	

Table 4
Statistical results of HV obtained by C-MOEA/D, C-NSGAII-CDP, C-TAEA, CMOEA-MS, DCNSGAIII, MOEA/D-DAE, TOP, TiGE2 and the proposed NACMOEA on the MW benchmark suite. Best result in each row is highlighted.

Problem	M	C-MOEA/D	C-NSGAII-CDP	C-TAEA	CMOEA-MS	DCNSGAIII	ToP	TiGE-2	NACMOEA
MW1	2	4.8742e-1 (3.47e-3) -	4.4922e-1 (3.95e-2) -	4.8899e-1 (2.13e-4) -	4.8971e-1 (2.00e-3) -	4.8893e-1 (1.76e-4) -	2.2173e-1 (1.51e-1) -	4.6145e-1 (2.88e-2) -	4.9005e-1 (2.34e-5)
MW2	2	5.6709e-1 (1.69e-2) ≈	5.6059e-1 (1.57e-2) ≈	5.7319e-1 (1.38e-2) +	5.6170e-1 (1.35e-2) ≈	5.7346e-1 (1.25e-2) +	4.5845e-1 (8.90e-2) -	5.1129e-1 (1.22e-1) -	5.6294e-1 (1.51e-2)
MW3	2	5.4477e-1 (3.47e-4) +	5.4247e-1 (5.36e-3) -	5.4507e-1 (2.17e-4) +	5.4467e-1 (2.73e-4) +	5.4507e-1 (2.24e-4) +	3.1253e-1 (2.40e-1) -	5.3102e-1 (1.83e-3) -	5.4446e-1 (2.34e-4)
MW4	3	8.4161e-1 (1.56e-4) +	8.2397e-1 (3.81e-3) -	8.3824e-1 (2.31e-4) -	8.3938e-1 (4.42e-4) ≈	8.4126e-1 (6.14e-4) +	5.4336e-1 (1.85e-1) -	7.8394e-1 (3.62e-2) -	8.3955e-1 (5.97e-4)
MW5	2	3.1504e-1 (3.51e-2) -	3.0074e-1 (5.78e-2) -	3.1522e-1 (1.87e-3) -	3.1250e-1 (1.22e-2) -	3.1077e-1 (5.70e-3) -	8.2000e-2 (4.72e-2) -	2.8979e-1 (8.46e-3) -	3.2424e-1 (2.32e-4)
MW6	2	3.1344e-1 (1.31e-2) ≈	3.0763e-1 (1.42e-2) -	3.2095e-1 (7.16e-3) ≈	3.1472e-1 (1.16e-2) ≈	3.1658e-1 (1.34e-2) ≈	1.7015e-1 (9.05e-2) -	2.8412e-1 (2.05e-2) -	3.1324e-1 (1.56e-2)
MW7	2	4.1137e-1 (9.93e-5) -	4.0655e-1 (3.05e-2) -	4.0941e-1 (6.60e-4) -	4.0953e-1 (3.98e-3) -	4.1199e-1 (3.14e-4) -	3.6098e-1 (7.83e-2) -	3.9279e-1 (4.56e-3) -	4.1288e-1 (2.34e-4)
MW8	3	5.3763e-1 (1.32e-2) ≈	5.2031e-1 (1.52e-2) -	5.3173e-1 (8.37e-3) -	5.4168e-1 (8.34e-3) ≈	5.3743e-1 (9.49e-3) -	2.9876e-1 (1.56e-1) -	1.5211e-1 (5.35e-2) -	5.4210e-1 (8.38e-3)
MW9	2	3.8980e-1 (2.85e-3) -	3.8880e-1 (3.52e-3) -	3.9244e-1 (1.78e-3) -	3.4469e-1 (9.85e-2) -	3.8545e-1 (4.64e-3) -	1.7460e-1 (1.57e-1) -	3.6109e-1 (6.97e-3) -	3.9857e-1 (2.20e-3)
MW10	2	4.2067e-1 (3.41e-2) -	4.0382e-1 (3.00e-2) -	4.4831e-1 (9.52e-3) ≈	4.3085e-1 (2.01e-2) ≈	4.3715e-1 (1.59e-2) ≈	3.4289e-1 (4.29e-2) -	4.1520e-1 (2.70e-2) -	4.2971e-1 (2.85e-2)
MW11	2	4.3483e-1 (4.42e-2) -	3.9943e-1 (7.80e-2) -	4.4242e-1 (1.10e-3) -	4.4738e-1 (5.15e-4) -	4.4381e-1 (3.08e-3) -	3.0570e-1 (6.67e-2) -	4.3502e-1 (2.29e-3) -	4.4808e-1 (1.13e-4)
MW12	2	6.0517e-1 (2.32e-4) +	6.0400e-1 (2.35e-4) -	6.0096e-1 (6.83e-4) -	6.0411e-1 (2.84e-4) -	6.0246e-1 (6.43e-4) -	1.1174e-1 (1.62e-1) -	5.7832e-1 (6.21e-3) -	6.0488e-1 (2.40e-4)
MW13	2	4.5835e-1 (1.67e-2) ≈	4.4014e-1 (4.42e-2) -	4.7095e-1 (8.55e-3) ≈	4.5381e-1 (2.07e-2) ≈	4.6310e-1 (1.02e-2) ≈	2.9427e-1 (1.22e-1) -	3.5085e-1 (5.91e-2) -	4.5928e-1 (1.48e-2)
MW14	3	4.4554e-1 (1.34e-3) -	4.5468e-1 (3.51e-3) -	4.6841e-1 (2.20e-3) ≈	4.6787e-1 (3.69e-3) +	4.6353e-1 (1.11e-2) -	4.0499e-1 (8.51e-2) -	4.5033e-1 (6.08e-3) -	4.6712e-1 (9.71e-3)
+/-/≈		3/7/4	0/13/1	2/8/4	2/6/6	3/8/3	0/14/0	0/14/0	

Table 5
Statistical results of IGD obtained by NACMOEA-ARC, NACMOEA-IF and the original NACMOEA on the MW benchmark suite. Best result in each row is highlighted.

Problem	M	NACMOEA-ARC	NACMOEA-IF	NACMOEA
MW1	2	9.8597e-3 (2.78e-3) -	1.6650e-3 (1.98e-5) +	1.6778e-3 (1.84e-5)
MW2	2	2.4404e-2 (7.17e-3) -	1.4694e-2 (8.38e-3) ≈	1.3798e-2 (9.88e-3)
MW3	2	1.6669e-2 (2.82e-3) -	5.2215e-3 (2.72e-4) -	5.0323e-3 (2.02e-4)
MW4	3	8.4703e-2 (8.18e-3) -	4.2451e-2 (5.13e-4) ≈	4.2284e-2 (5.81e-4)
MW5	2	2.8989e-3 (6.43e-3) ≈	3.5553e-3 (2.95e-3) -	6.8643e-4 (5.47e-4)
MW6	2	2.5836e-2 (1.28e-2) -	1.4665e-2 (1.56e-2) ≈	1.4953e-2 (1.42e-2)
MW7	2	1.2514e-2 (2.09e-3) -	5.9121e-3 (8.64e-3) ≈	4.2336e-3 (2.37e-4)
MW8	3	1.0818e-1 (1.83e-2) -	4.4219e-2 (2.35e-3) ≈	4.4389e-2 (2.22e-3)
MW9	2	1.5321e-2 (2.43e-3) -	4.5408e-3 (2.68e-4) -	4.4631e-3 (3.75e-4)
MW10	2	3.1349e-2 (2.74e-2) -	1.2871e-2 (1.42e-2) ≈	2.0970e-2 (2.26e-2)
MW11	2	1.6987e-2 (3.11e-3) -	5.9809e-3 (1.06e-4) ≈	5.9678e-3 (1.14e-4)
MW12	2	1.5053e-2 (3.16e-3) -	4.7991e-3 (1.10e-4) -	4.7379e-3 (1.05e-4)
MW13	2	8.0508e-2 (4.28e-2) -	3.1691e-2 (2.86e-2) ≈	4.2194e-2 (2.91e-2)
MW14	3	1.8823e-1 (3.07e-2) -	1.0449e-1 (2.16e-2) ≈	1.0535e-1 (2.10e-2)
+/-/≈		0/13/1	1/4/9	

of the narrow tunnel-like feasible region, which requires a high diversity of the population. The convergence of C-MOEA/D, C-TAEA, TiGE-2, and ToP is insufficient, with a significant number of solutions failing to converge near the CPF. The absence of diversity in C-NSGAII-CDP and DCNSGAIII leads to a partial loss of CPF. Our proposed NACMOEA demonstrates exceptional performance in both convergence and diversity, effectively handling complex feasible region cases.

3) Effectiveness of each component in NACMOEA: To evaluate the efficacy of components within the NACMOEA, ablation studies are conducted in this section. The original NACMOEA and its two variant versions are compared on the MW benchmark suite. The first variant, NACMOEA-ARC, excludes the dual archive cooperation mechanism present in the original NACMOEA and employs only the main population for evolution. It is important to note that as this variant does not utilize the archive for infeasible solutions, the associated module is also omitted. The second variant, NACMOEA-IF, retains the dual archive cooperation mechanism but eliminates the infeasible solutions utilization mechanism.

The experimental outcomes of NACMOEA and its variants based on IGD are presented in Table 5. A comparison reveals that NACMOEA outperforms the first variant, NACMOEA-ARC, on the majority of test problems. This is attributed to the fact that NACMOEA-ARC has eliminated the dual archive cooperation mechanism and infeasible solution utilization strategy, resulting in a considerable performance degradation. The underlying architecture of NACMOEA-ARC is a straightforward two-stage algorithm. The second variant, NACMOEA-IF, demonstrates a comparable performance to NACMOEA on six test problems. This is attributed to the partial or complete overlap between the CPF and UPF of these test problems. For instance, there is a partial overlap between CPF and UPF for MW1, MW6, and MW8. The CPF and UPF for MW10 and MW13 are highly similar, and the CPF of MW14 completely overlaps with the UPF. Given the negligible or non-existent gap between UPF and CPF in the objective space, the proposed infeasible solution utilization strategy is ineffective, resulting in NACMOEA-IF exhibiting approximately the same performance as NACMOEA on these types of test problems. However, the performance of NACMOEA-IF is inferior to NACMOEA for test problems with a distinct separation between CPF and UPF, such as MW3, MW9, and MW12, thereby demonstrating the efficacy of the proposed infeasible solution utilization strategy.

5. Conclusions

In this paper, we propose a novel two-stage CMOEA called NACMOEA to address the challenging CMOPs with intricate feasible regions. The algorithm's core concept involves dividing the search process into two stages: convergence-oriented search and feasibility-oriented search. At different stages of the search, dual archives perform distinct roles and collaborate effectively. The transition between these stages is determined by evaluating the non-dominant coverage rate metric, which accurately reflects the dominance relationship of the current generation population compared to its history counterparts. In stage one, our proposed approach can rapidly converge to UPF since the influence of constraints is relatively weak. At this stage, the obtained UPF is saved by CA, while FA collects the elite individuals that are clustered around the optimal feasible region. In stage two, to approach the CPF from the feasible side, FA employs a niche-based feasibility priority select strategy, whereas the population utilizes an ϵ-constrained technique to approximate the CPF from the infeasible side. In this process, crucial niche information is provided by CA. Furthermore, a niche-based strategy for utilizing infeasible solutions is established to enhance convergence speed towards CPF from both feasible and infeasible sides.

In the experimental section, we compare the proposed algorithm with seven other advanced CMOEAs on a total of 28 test problems from the LIR-CMOP and MW test sets. These test problems encompass various forms of feasible regions, effectively validating the performance of CMOEA across different types of CMOPs. The statistical results demonstrate that NACMOEA outperforms the compared algorithms in terms of overall performance. To further validate the effectiveness of components in NACMOEA, we compare two variants with the original NACMOEA using MW suite. The experimental findings reveal that incorporating a dual archive cooperation mechanism and an infeasible solution utilization strategy significantly enhance the performance of NACMOEA.

Through an analysis of the limitations of existing CMOEAs, this paper proposes a multi-stage framework based on the niche technique. This study substantiates the significance of a balanced strategy encompassing convergence, feasibility, and diversity. Furthermore, it unveils the potential of infeasible solutions in exploring unknown feasible regions and enhancing algorithm performance. While the proposed algorithm exhibits promising results in experiments, there is scope for extending its application to tackle more challenging CMOPs. For instance, improvements can be made to environment selection and archive update strategies while augmenting selection pressure to address many-objective constrained optimization problems. Additionally, embedding multi-modal optimization techniques could prove beneficial for solving constrained multi-modal multi-objective optimization problems.

CRediT authorship contribution statement

Fengyu Guo: Conceptualization, Methodology, Software, Writing – original draft. **Hecheng Li:** Supervision, 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.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (No. 61966030).

References

[1] Agrawal RB, Deb K, Agrawal RB. Simulated binary crossover for continuous search space. Complex Syst 1994;9:115–48.
[2] Asafuddoula M, Ray T, Sarker R. A decomposition-based evolutionary algorithm for many objective optimization. IEEE Trans Evol Comput 2015;19:445–60.
[3] Bader J, Zitzler E. HypE: an algorithm for fast hypervolume-based many-objective optimization. Evol Comput 2011;19:45–76. https://doi.org/10.1162/EVCO_a_00009.
[4] Bosman PAN, Thierens D. The balance between proximity and diversity in multiobjective evolutionary algorithms. IEEE Trans Evol Comput 2003;7(2):174–88.
[5] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 2002;6:182–97.
[6] Fan L, Tatsuo Y, Tao X, Lin Y, Liu H. A novel hybrid algorithm for solving multiobjective optimization problems with engineering applications. Math Probl Eng 2018;2018:1–15.
[7] Fan Z, Li W, Cai X, Huang H, Fang Y, You Y, et al. An improved epsilon constraint-handling method in MOEA/D for CMOPs with large infeasible regions. Soft Comput 2019.
[8] Fan Z, Li W, Cai X, Li H, Wei C, Zhang Q, et al. Push and pull search for solving constrained multi-objective optimization problems. Swarm Evol Comput 2019;44:665–79. https://doi.org/10.1016/j.swevo.2018.08.017. https://www.sciencedirect.com/science/article/pii/S2210650218300233.
[9] Jain H, Deb K. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: handling constraints and extending to an adaptive approach. IEEE Trans Evol Comput 2014;18:602–22.
[10] Jiao R, Xue B, Zhang M. A multiform optimization framework for constrained multiobjective optimization. IEEE Trans Cybern 2022:1–13. https://doi.org/10.1109/TCYB.2022.3178132.
[11] Jiao R, Zeng S, Li C, Yang S, Ong YS. Handling constrained many-objective optimization problems via problem transformation. IEEE Trans Cybern 2021;51:4834–47. https://doi.org/10.1109/TCYB.2020.3031642.
[12] Kannan BK, Kramer SN. An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. Trans ASME J Mech Des 1994;116:405–11.
[13] Li Ke, Deb Kalyanmoy. An evolutionary many-objective optimization algorithm based on dominance and decomposition. IEEE Trans Evol Comput 2014.
[14] Li K, Chen R, Fu G, Yao X. Two-archive evolutionary algorithm for constrained multiobjective optimization. IEEE Trans Evol Comput 2019;23:303–15. https://doi.org/10.1109/TEVC.2018.2855411.
[15] Liu Z, Wang B, Tang K. Handling constrained multiobjective optimization problems via bidirectional coevolution. IEEE Trans Cybern 2021.
[16] Liu ZZ, Qin Y, Song W, Zhang J, Li K. Multiobjective-based constraint-handling technique for evolutionary constrained multiobjective optimization: a new perspective. IEEE Trans Evol Comput 2022:1. https://doi.org/10.1109/TEVC.2022.3194729.
[17] Liu ZZ, Wang Y. Handling constrained multiobjective optimization problems with constraints in both the decision and objective spaces. IEEE Trans Evol Comput 2019:1.
[18] Ma Z, Wang Y. Evolutionary constrained multiobjective optimization: test suite construction and performance comparisons. IEEE Trans Evol Comput 2019:1.
[19] Ma Z, Wang Y. Shift-based penalty for evolutionary constrained multiobjective optimization and its application. IEEE Trans Cybern 2023;53:18–30. https://doi.org/10.1109/TCYB.2021.3069814.
[20] Ma Z, Wang Y, Song W. A new fitness function with two rankings for evolutionary constrained multiobjective optimization. IEEE Trans Syst Man Cybern Syst 2021;51:5005–16. https://doi.org/10.1109/TSMC.2019.2943973.
[21] Maldonado HM, Zapotecas-Martínez S. A dynamic penalty function within MOEA/D for constrained multi-objective optimization problems. In: IEEE congress on evolutionary computation; 2021.
[22] Ming F, Gong W, Li D, Wang L, Gao L. A competitive and cooperative swarm optimizer for constrained multi-objective optimization problems. IEEE Trans Evol Comput 2022:1. https://doi.org/10.1109/TEVC.2022.3199775.
[23] Ming F, Gong W, Wang L, Gao L. A constrained many-objective optimization evolutionary algorithm with enhanced mating and environmental selections. IEEE Trans Cybern 2022:1–13. https://doi.org/10.1109/TCYB.2022.3151793.
[24] Ming M, Trivedi A, Wang R, Srinivasan D, Zhang T. A dual-population-based evolutionary algorithm for constrained multiobjective optimization. IEEE Trans Evol Comput 2021;25:739–53. https://doi.org/10.1109/TEVC.2021.3066301.
[25] Ming M, Wang R, Ishibuchi H, Zhang T. A novel dual-stage dual-population evolutionary algorithm for constrained multiobjective optimization. IEEE Trans Evol Comput 2022;26:1129–43. https://doi.org/10.1109/TEVC.2021.3131124.
[26] Qiao K, Yu K, Qu B, Liang J, Song H, Yue C, et al. Dynamic auxiliary task-based evolutionary multitasking for constrained multi-objective optimization. IEEE Trans Evol Comput 2022:1. https://doi.org/10.1109/TEVC.2022.3175065.
[27] Qin C, Ming F, Gong W, Gu Q. Constrained multi-objective optimization via two archives assisted push–pull evolutionary algorithm. Swarm Evol Comput 2022;75:101178. https://doi.org/10.1016/j.swevo.2022.101178. https://www.sciencedirect.com/science/article/pii/S2210650222001456.
[28] Sun R, Zou J, Liu Y, Yang S, Zheng J. A multi-stage algorithm for solving multi-objective optimization problems with multi-constraints. IEEE Trans Evol Comput 2022:1. https://doi.org/10.1109/TEVC.2022.3224600.
[29] Tian Y, Cheng R, Zhang X, Su Y, Jin Y. A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization. IEEE Trans Evol Comput 2018.
[30] Tian Y, Zhang T, Xiao J, Zhang X, Jin Y. A coevolutionary framework for constrained multiobjective optimization problems. IEEE Trans Evol Comput 2021;25:102–16. https://doi.org/10.1109/TEVC.2020.3004012.
[31] Tian Y, Zhang Y, Su Y, Zhang X, Tan KC, Jin Y. Balancing objective optimization and constraint satisfaction in constrained evolutionary multiobjective optimization. IEEE Trans Cybern 2022;52:9559–72. https://doi.org/10.1109/TCYB.2020.3021138.
[32] Wang BC, Shui ZY, Feng Y, Ma Z. Evolutionary algorithm with dynamic population size for constrained multiobjective optimization. Swarm Evol Comput 2022;73:101104. https://doi.org/10.1016/j.swevo.2022.101104. https://www.sciencedirect.com/science/article/pii/S2210650222000748.
[33] Wang C, Xu R. An angle based evolutionary algorithm with infeasibility information for constrained many-objective optimization. Appl Soft Comput 2020;86:105911. https://doi.org/10.1016/j.asoc.2019.105911. https://www.sciencedirect.com/science/article/pii/S1568494619306921.
[34] Wang Y, Liu Y, Zou J, Zheng J, Yang S. A novel two-phase evolutionary algorithm for solving constrained multi-objective optimization problems. Swarm Evol Comput 2022;75:101166. https://doi.org/10.1016/j.swevo.2022.101166. https://www.sciencedirect.com/science/article/pii/S2210650222001341.
[35] Ye T, Ran C, Zhang X, Jin Y. PlatEMO: a Matlab platform for evolutionary multi-objective optimization. IEEE Comput Intell Mag 2017;12:73–87.
[36] Zhang K, Xu Z, Yen GG, Zhang L. Two-stage multi-objective evolution strategy for constrained multi-objective optimization. IEEE Trans Evol Comput 2022:1. https://doi.org/10.1109/TEVC.2022.3202723.
[37] Zhang Q, Hui L. MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput 2008;11:712–31.
[38] Zhou Y, Min Z, Wang J, Zhang Z, Zhang J. Tri-goal evolution framework for constrained many-objective optimization. IEEE Trans Syst Man Cybern Syst 2018.
[39] Zhu Q, Zhang Q, Lin Q. A constrained multiobjective evolutionary algorithm with detect-and-escape strategy. IEEE Trans Evol Comput 2020;24:938–47. https://doi.org/10.1109/TEVC.2020.2981949.
[40] Zitzler E, Künzli S. Indicator-based selection in multiobjective search. In: 8th international conference on parallel problem solving from nature; 2004.
[41] Zitzler E, Laumanns M, Thiele L. SPEA2: improving the strength Pareto evolutionary algorithm. Technical Report Gloriastrasse. 2001.
[42] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Trans Evol Comput 1999;3:257–71.