Two-Stage Multiobjective Evolution Strategy for Constrained Multiobjective Optimization

Kai Zhang, Zhiwei Xu, *Graduate Student Member, IEEE*, Gary G. Yen, *Fellow, IEEE*, and Ling Zhang

***Abstract*—For the past many years, several constrained multiobjective evolutionary algorithms (CMOEAs) have been designed for solving constrained multiobjective optimization problems (CMOPs). In these CMOEAs, some constraint-handling techniques (CHTs) were proposed to balance the convergence and constrained satisfaction, however, they still face some serious challenges, such as premature convergence to the local optimal region and labor-intensive tuning of parameters for a specific CMOP. Furthermore, most of the existing CHTs are derived by solving constrained single-objective optimization. The information hidden from the feasible nondominated set (FNDS) has not been fully utilized. This study proposed a novel parameter-less constraint handling technique, which divides the entire population into three mutually exclusive subsets dynamically: 1) FNDS; 2) the subset dominated by FNDS; and 3) the subset not dominated by FNDS. According to the proposed division of labor, it is not necessary to balance the convergence and constrained satisfaction in each subset. To avoid being entrapped in local optima, the proposed algorithm adopts a two-stage strategy to solve CMOPs. In the first stage, the proposed algorithm focuses solely on converging toward the unconstrained Pareto front (PF) without considering the constrained satisfaction. In the second stage, the FNDS constraint handling technique is adopted to guide the population converging toward constrained PF effectively. The performance of the proposed algorithm was compared to that of nine state-of-the-art CMOEAs, and the comparison results show that the proposed algorithm performs significantly better on the CF, MW, and LIRCMOP test suites.**

***Index Terms*—Constrained multiobjective evolutionary algorithm (CMOEA), constrained multiobjective optimization problem (CMOP), evolution strategy.**

I. Introduction

IN THE past many years, several constrained multiobjective evolutionary algorithms (CMOEAs) have been designed to solve constrained multiobjective optimization problems (CMOPs), which can be formulated as follows:

$$\min f(x) = \min \left[f_1(x), f_2(x), \ldots, f_M(x)\right]^T$$
$$s.t. \begin{cases} g_j(x) \le 0, \ j = 1, \ldots, J \\ h_k(x) = 0, \ k = 1, \ldots, K \end{cases}$$

where $x \in \Omega$ is the decision vector with N decision variables $x_i, i = 1, \ldots, N$. $f(x) \in R^M$ is the objective vector with M objective functions, f_i. $i = 1, \ldots, M$. Moreover, Ω denotes the decision search space and R^M denotes the objective space. For a multiobjective optimization problem (MOP), a set of nondominated optimal solutions, called the Pareto set (PS), and the corresponding objective vectors, called the Pareto front (PF), exist. In addition, the optimal solutions for real-world CMOPs must satisfy J inequality constraints and K equality constraints. $g_j(x)$ is a function of the jth inequality constraint, whereas $h_k(x)$ is a function of the kth equality constraint [1].

In contrast to the unconstrained MOP, solving the CMOP is slightly different, because it needs to satisfy three criteria simultaneously: 1) constrained satisfaction; 2) good convergence; and 3) good distribution. In particular, it is difficult to balance constrained satisfaction and convergence. For example, some complex CMOPs have many infeasible regions and local optimal solutions, as shown in Fig. 1.

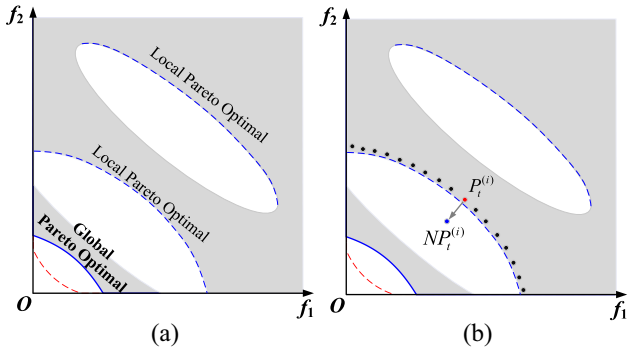


Fig. 1. (a) Global and local Pareto optimal solutions. (b) $NP_t^{(i)} \prec P_t^{(i)}$, however, CV values $CV(NP_t^{(i)}) > CV(P_t^{(i)})$.

Let $P_t^{(i)}$ be the ith feasible nondominated individual in population P_t, and $NP_t^{(i)}$ be the newly mutated individual. As shown in Fig. 1(b), the $NP_t^{(i)}$ dominates the original $P_t^{(i)}$, whereas the $NP_t^{(i)}$ is located in an infeasible region (displayed on a white background). Notably, the $NP_t^{(i)}$ has better convergence quality than $P_t^{(i)}$, whereas $P_t^{(i)}$ has better constrained satisfaction than $NP_t^{(i)}$. Regardless of the constraint handling technique

Manuscript received 5 January 2022; revised 20 April 2022 and 18 July 2022; accepted 23 August 2022. Date of publication 29 August 2022; date of current version 31 January 2024. This work was supported by the National Natural Science Foundation of China under Grant 62176191, Grant U1803262, and Grant 61602350. *(Corresponding author: Gary G. Yen.)*

Kai Zhang is with the School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430065, China (e-mail: zhangkai@wust.edu.cn).

Zhiwei Xu and Ling Zhang are with the Hubei Province Key Laboratory of Intelligent Information Processing and Real-Time Industrial System, Wuhan University of Science and Technology, Wuhan 430065, China (e-mail: kenxucn95@gmail.com; tobefuture315@163.com).

Gary G. Yen is with the School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078 USA (e-mail: gyen@okstate.edu).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TEVC.2022.3202723.

Digital Object Identifier 10.1109/TEVC.2022.3202723

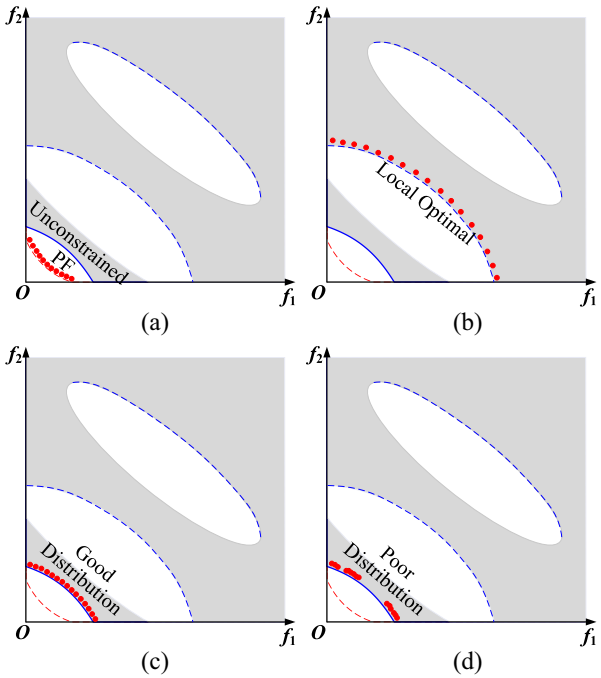


Fig. 2. (a) Convergence is preferable, the population is easy to converge to unstrained PF. (b) Constraint satisfaction is preferable, the population is easy to trap into local optimal. (c) PF with good distribution. (d) PF with poor distribution.

applied, when a CMOEA accepts $NP_t^{(i)}$, the population tends to converge toward an unconstrained PF, as shown in Fig. 2(a). If a CMOEA rejects $NP_t^{(i)}$, the population is easily trapped in a local optimal, as shown in Fig. 2(b). Ideally, a well-designed CMOEA should find a set of feasible, well-converged, and well-distributed nondominated solutions, as shown in Fig. 2(c). However, many CMOEAs can only obtain solutions with poor distributions, as shown in Fig. 2(d).

In terms of constraint-handling techniques (CHTs), existing CMOEAs can be broadly classified into four categories: 1) penalty function approaches [2], [3], [4], [5], [6]; 2) separation of objectives and constraints approaches [7], [8], [9], [10], [11], [12], [13], [14], [15], [16]; 3) special representation and operator approaches [17], [18], [19], [20]; and 4) hybrid approaches [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40].

First, the penalty function approach attempts to add a constraint violation (CV) to the fitness function of an infeasible individual to render it less desirable, which in turn can convert a CMOP into an unconstrained MOP. Second, some CMOEAs tend to handle objectives and constraints separately, such as the constraint-dominance principle [7], stochastic ranking [9], and epsilon constraint handling [11]. However, achieving an ideal balance between convergence and constrained satisfaction is difficult. Third, special representations and operators were designed to generate more promising solutions. Furthermore, the computational cost is always exceedingly high, and the parameter setting is often excessively sensitive. Finally, some CMOEAs hybridize multiple CHTs to achieve better performance than relying on a single CHT alone. However, existing CMOEAs still face challenges such as premature convergence to local optimal regions, poor diversity, and laborious tuning of parameters for a specific CMOP. In surveying the literature over the last two decades, nearly all existing CHTs have been derived by solving constrained single-objective optimization. Therefore, the information hidden from the feasible nondominated set (FNDS) has not been fully exploited for addressing constrained multiobjective optimization.

In this study, a novel two-stage evolution strategy, CMOES, is proposed to solve CMOPs. First, a novel parameter-less constraint handling technique is proposed, which divides the entire population into three mutually exclusive subsets dynamically: FNDS, the subset dominated by FNDS ($\text{FNDS}^{\prec}$), and the subset not dominated by FNDS ($\text{FNDS}^{\not\prec}$). Individuals in each subset are not required to balance constraints and objectives; therefore, different environmental selection strategies are designed for these subsets. The individuals in subset $\text{FNDS}^{\prec}$ are dominated by FNDS, which are targeted to pursue only better-convergence toward FNDS. By contrast, the individuals in the subset $\text{FNDS}^{\not\prec}$ are neither FNDS nor $\text{FNDS}^{\prec}$, which are always located in converged infeasible regions. These infeasible individuals in $\text{FNDS}^{\not\prec}$ need focus only on better-constrained satisfaction. The individuals in the subset FNDS are dedicated to searching for well-converged and well-distributed feasible solutions. Second, to avoid being trapped in the local optima, the proposed CMOES adopts a two-stage strategy to solve the CMOP. In the first stage, the proposed algorithm focuses on converging to an unconstrained PF without considering constraint satisfaction. In the second stage, the FNDS CHT is designed to guide the population converging toward the PF effectively. The performance of the proposed algorithm was compared with nine state-of-the-art CMOEAs, and the comparison results clearly show that the proposed CMOES performs significantly better on the CF, MW, and LIRCMOP test suites.

In recent years, some CMOEAs, such as PPS [21], C-TAEA [23], and MOEA/D-DAE [12], also adopted two-stage or two-population strategy to escape from local optimal regions. The main difference between CMOES and these state-of-the-art CMOEAs rests on the CHT. PPS and MOEA/D-DAE incorporated the epsilon constrained-handling method [5], while C-TAEA prefers the constraint-dominance principle (CDP) in the convergence archive. However, these CMOEAs still encounter difficulties in obtaining well-converged and well-distributed feasible solutions with existing CHTs. The experimental results show that the proposed CHT can provide more competitive performance on the three CMOP test suites.

The remainder of this article is organized as follows. Section II presents a comprehensive review of the CMOEAs. Section III provides details of the proposed FNDS CHT and the proposed CMOES algorithm. The experimental results of the LIRCMOP test suite using nine competing CMOEAs are presented in Section IV. Finally, the conclusion section concludes this article.

II. Related Works and Motivation

The design of the CMOEA must maintain a proper balance among three criteria: 1) constrained satisfaction;

2) convergence; and 3) distribution for a given CMOP. Over the past several years, many CMOEAs have been designed for solving CMOPs with different constrained handling strategies, which can be classified into four categories, as follows.

A. Penalty Functions Approaches

The penalty function approach attempts to add a CV value to the fitness function of infeasible individuals to render them less desirable, which in turn can convert a CMOP into an unconstrained MOP. The penalty factor λ is often used to maintain a balance between the objectives and constraints, as shown in (2)

$$\begin{aligned} \text{fitness}(x) &= f(x) + \lambda \times CV(x) \\ CV(x) &= \sum_{j=1}^{J} g_j(x) + \sum_{k=1}^{K} |h_k(x)|. \end{aligned} \tag{1}$$

To appropriately set the penalty factor, many penalty function methods have been proposed, such as static, dynamic, and adaptive penalties. Representative designs include the ShiP [5] and dCMOEA [6].

However, the setting of the penalty factor often depends on a specific problem, and it still needs to be tuned to find an optimal penalty degree for infeasible solutions. Although some self-adaptive penalty CMOEAs have been designed, feedback information for setting the self-adaptive penalty factor is often biased.

B. Separation of Objectives and Constraints Approaches

Some CMOEAs prefer to handle objectives and constraints separately for solving CMOPs, including the CDP, epsilon constraint handling, and stochastic ranking (SR).

C. Constraint Dominance Principle

Similar to the traditional dominant principle, CDP has two additional comparison principles for an infeasible solution. The feasible individual is the preferred candidate solution compared to the infeasible individuals, and the infeasible individual with a smaller CV value is superior to that with a greater CV value. Popular designs in this category are A-NSGA-III [7] and MOEA/D-ACDP [8].

Because the feasible solution often has a higher precedence than the infeasible solution, CMOEAs in this category are easily trapped in the local optimal feasible region.

D. Stochastic Ranking

SR methods compare the objectives or constraints randomly, and the comparison order is determined by a given probability parameter, *Pf*. When *Pf* is greater than a random number, the objectives are prioritized over the constraints. By contrast, when *Pf* is smaller than a random number, the constraints precede the objectives. The best-known examples are SRA [9] and CMOEA/D-DE-SR [10].

However, it is very difficult to obtain the optimal value of the probability parameter *Pf* for CMOPs with different problem characteristics. Additionally, because the objectives or constraints are compared randomly, a few well-converged infeasible solutions are often maintained.

E. Epsilon Constrained Handling

The epsilon constraint-handling adopts a CV tolerance mechanism to consider the infeasible solutions as feasible solutions within epsilon level ϵ. During the searching process, the parameter ϵ plays a critical role in controlling the relaxation degree of the constraints. The most representative designs include MOEA/D-IEpsilon [11] and MOEA/D-DAE [12].

Similarly, epsilon-constrained approaches must turn the parameters carefully to control the decreasing epsilon level. In addition, epsilon-constrained approaches show difficulties and limitations in finding Pareto optimal solutions that are located on the boundaries of the feasible regions.

F. Special Representation and Operator Approaches

This class of CHTs designs some special reproductions or operators to generate more promising solutions and repair infeasible solutions into feasible ones. In addition, some approaches prune the search space by removing infeasible regions. Typical CMOEAs include POCEA [18], C-PSA [19], and feasible-guiding NSGA-II [20].

However, most CMOEAs in this category require high computational costs to generate promising solutions and are extremely sensitive to parameter tuning.

G. Hybrid Approaches

In recent years, several CMOEAs have been proposed to integrate different CHTs to achieve better performance. PPS [21], ToP [22], and DD-CMOEA [37] separate the entire evolutionary process into multiple stages, and different constraint-handling strategies are adopted in each stage. Moreover, C-TAEA [23] and CCMO [24] divide the entire population into multiple subpopulations or archives, and different constraint-handling strategies are adopted in each subpopulation or archive.

The novelty and shortcomings of existing CHTs and representative CMOEAs are summarized in Table III of the supplementary document.

H. Motivation

Although many CMOEAs have been proposed for solving CMOPs, it remains difficult to balance constrained satisfaction and convergence. Fig. 3 shows the experimental results of eight state-of-the-art CMOEAs on the LIRCMOP8 test problem. The population size was 100, and the maximal evaluation was 100 000. The experiment was performed using PlatEMO v3.4 [27].

As shown in Fig. 3, LIRCMOP8 has a very large infeasible region (displayed on a white background), which partitions the feasible region (displayed on a shaded background) into several isolated parts. In addition, the constrained PF is different from the unconstrained PF, which is challenging for CHTs. Some CMOEAs, including A-NSGA-III [7], CMOEAD [7], ToP [22], C-TAEA [23], and SRA [9], obtain

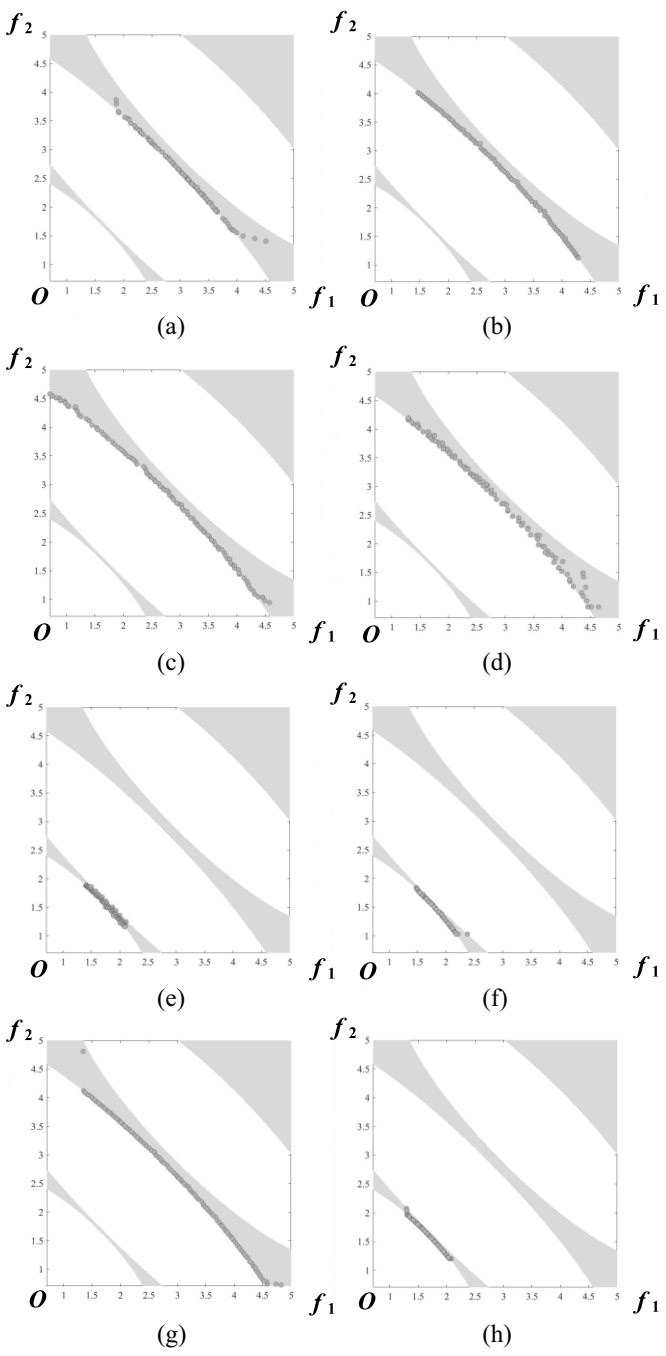


Fig. 3. Experimental results on LIRCMOP8 by eight state-of-the-art CMOEAs with 100 individuals under 100 000 evaluations. (a) A-NSGA-III. (b) CMOEAD. (c) ToP. (d) C-TAEA. (e) POCEA. (f) MOEA/D-DAE. (g) SRA. (h) PPS.

only feasible local optimal solutions. Clearly, these CMOEAs cannot manage a precise balance between constrained satisfaction and convergence for LIRCMOP8. Some CMOEAs, such as POCEA [18], PPS [21], and MOEA/D-DAE [12] can find global feasible regions; however, they cannot identify the entire constrained PF. These CMOEAs can converge to the global feasible region; however, the diversity allows for improvement.

However, most of the existing CHTs are derived by solving a constrained single-objective optimization, the information

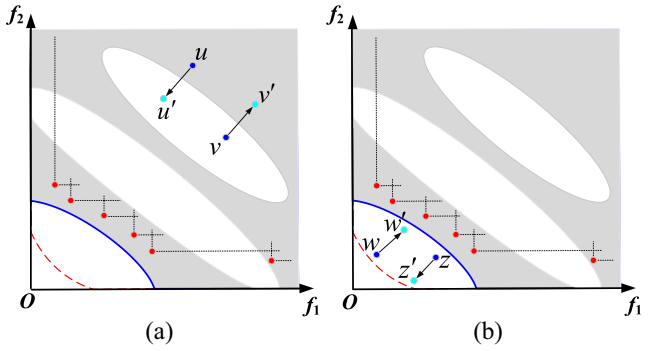


Fig. 4. (a) Individuals that are dominated by FNDS need to consider only convergence without reference to CV. (b) Individuals that are not dominated by FNDS need to consider only CV without reference to convergence quality.

hidden from the FNDS has not been fully exploited. In addition, many individuals need not to maintain a balance between constrained satisfaction and convergence during the evolutionary process.

Therefore, all individuals that are dominated by FNDS need only consider convergence without reference to CV before they converge to the FNDS. As shown in Fig. 4(a), u is an individual dominated by the FNDS, u' is the newly generated candidate solution. In this case, u' can replace u for better convergence without minimizing poor constrained satisfaction. It is unnecessary to waste the computation resources to balance the constrained satisfaction for u and v. For example, we assume that v' is accepted and replaced v because of better constraint satisfaction, as shown in Fig. 4(a). The evolutionary direction may be in the opposite direction to the convergence toward FNDS.

By contrast, all individuals that are not dominated by FNDS must be infeasible individuals, which need to consider only CV without reference to convergence. As shown in Fig. 4(b), w' can replace w for better CV without considering poor convergence quality. It is unnecessary to use the computation resources to balance convergence for w and z. For example, we assume that z' is accepted and replaced z owing to better convergence, as shown in Fig. 4(b). The evolutionary direction may be opposite to the convergence toward the feasible region.

III. Proposed Algorithm

In this study, a novel parameter-less CHT is proposed for solving CMOPs. In addition, our algorithm adopts a two-stage strategy to search for a globally constrained PF, as shown in Fig. 5.

In the first stage, the proposed algorithm focuses on searching for an unconstrained PF without considering constrained satisfaction, as shown in Fig. 5(a). In the second stage, an FNDS-based CHT is proposed to guide the entire population to converge toward the PF, as shown in Fig. 5(b). The two-stage strategy is beneficial for guiding the population to converge toward the global feasible optimal region and avoid being trapped in local optima.

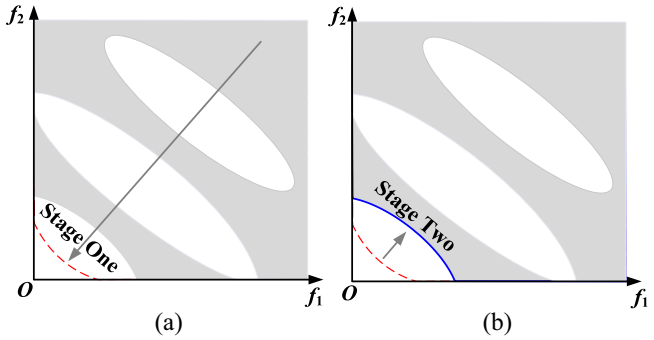


Fig. 5. (a) First stage without considering constraint satisfaction. (b) Second stage with proposed constraint handling technique.

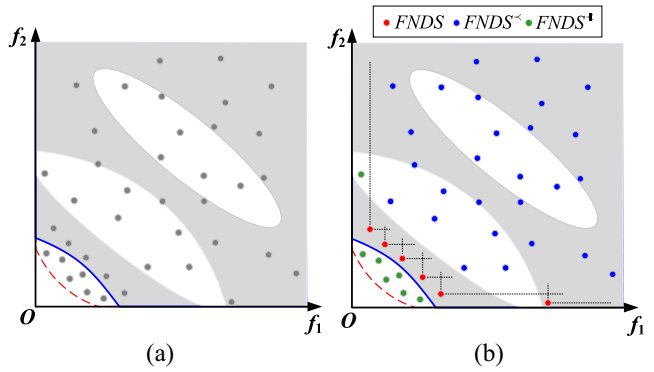


Fig. 6. (a) Randomly sampled individuals in the objective space. (b) Entire population divided into three mutually exclusive subsets dynamically: FNDS, $\text{FNDS}^{\prec}$, and $\text{FNDS}^{\not\prec}$.

A. Feasible Nondominated Set-Based Constraint Handle

The MOP is very different from the single-objective optimization problem, which has a set of tradeoff nondominated solutions. For a CMOP, an FNDS can provide important information for constraint handling. In this study, a novel parameter-less CHT is proposed for constrained multiobjective optimization.

Fig. 6 shows a schematic illustrating how the individuals are divided into three subsets dynamically. As shown in Fig. 6(a), individuals are sampled randomly in the objective space, and should be located in feasible or infeasible regions. In the proposed algorithm, the individuals are divided into three mutually exclusive subsets: 1) FNDS; 2) $\text{FNDS}^{\prec}$; and 3) $\text{FNDS}^{\not\prec}$ dynamically, which are denoted by red, blue, and green, respectively. In Fig. 6(b), the most important individual is FNDS, which is shown in red. The blue individuals are dominated by FNDS, denoted by $\text{FNDS}^{\prec}$. By contrast, the green individuals are not dominated by FNDS, as denoted by $\text{FNDS}^{\not\prec}$.

Notably, the nondominated solutions in the feasible regions are the best candidate solutions, which are shown in red in Fig. 6(b). Second, the blue individuals in $\text{FNDS}^{\prec}$ may be located in either the feasible or infeasible regions, both of which have inherently poorer convergence than FNDS. The most important task for blue individuals is to converge to the FNDS. During the evolutionary process, constrained satisfaction is not a necessary condition for the blue individuals in

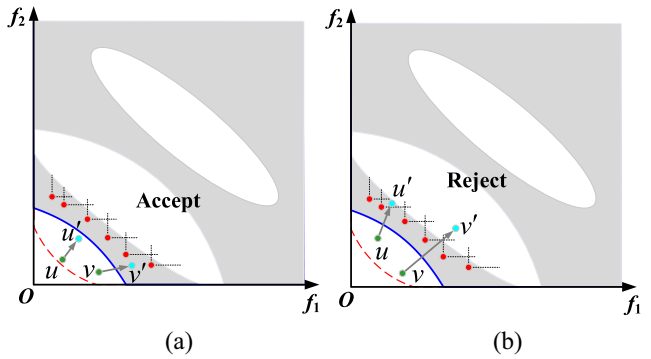


Fig. 7. (a) if $\text{FNDSDomCT}(u') = \text{FNDSDomCT}(u)$ and $CV(u') < CV(u)$, u' will replace u. (b) Although $CV(u') < CV(u)$, if $\text{FNDSDomCT}(u') > 0$, u' will be neglected, regardless of u' is feasible or infeasible solution.

$\text{FNDS}^{\prec}$. Third, the green individuals in $\text{FNDS}^{\not\prec}$ must be infeasible solutions. Therefore, the green individuals in $\text{FNDS}^{\not\prec}$ need to pursue only better constrained satisfaction without considering convergent performance.

In our algorithm, we tailor three different strategies for the three types of individuals in the second stage. Fig. 7 shows the environmental selection strategy for green individuals in $\text{FNDS}^{\not\prec}$. Let u and v be two individuals in $\text{FNDS}^{\not\prec}$, and u' and v' are the corresponding mutated candidate solutions. In Fig. 7(a), our algorithm replaces u and v with the new candidates u' and v', respectively, because u' and v' have better constraint values than u and v, respectively. Furthermore, it is necessary that the mutated u' and v' are not dominated by FNDS. By contrast, in Fig. 7(b), although u' and v' have better constraint values than u and v, respectively, they are dominated by FNDS. Because mutated u' and v' have poor convergence quality compared with the original u and v, respectively, our algorithm should reject the newly mutated candidate solutions.

Let $P_t^{(i)}$ be the ith individual in population P_t and $NP_t^{(i)}$ be the newly mutated candidate solution. If $P_t^{(i)}$ is an individual in $\text{FNDS}^{\not\prec}$, the environmental selection strategy is as follows:

$$P_t^{(i)} = \begin{cases} NP_t^{(i)}, \text{if } \text{FNDSDomCT}\left(\text{NP}_t^{(i)}\right) = \text{FNDSDomCT}\left(P_t^{(i)}\right) \\ \qquad \text{and } \text{CV}\left(\text{NP}_t^{(i)}\right) < \text{CV}\left(P_t^{(i)}\right) \\ P_t^{(i)}, \qquad \text{else} \end{cases} \tag{2}$$

where $\text{FNDSDomCT}(P_t^{(i)})$ is a function that counts the number of individuals of $P_t^{(k)}$ in FNDS that can dominate $P_t^{(i)}$. A smaller value of FNDSDomCT() implies better convergence, and fewer individuals in FNDS can dominate $P_t^{(i)}$

$$\begin{aligned} \text{FNDSDomCT}\left(P_t^{(i)}\right) &= \sum \text{BeDom}\left(P_t^{(i)}, P_t^{(k)}\right) \\ \text{BeDom}\left(P_t^{(i)}, P_t^{(k)}\right) &= \begin{cases} 1, \ P_t^{(k)} \prec P_t^{(i)} \\ 0, \ P_t^{(k)} \not\prec P_t^{(i)} \end{cases}, \quad \text{and } P_t^{(k)} \in \text{FNDS}. \end{aligned} \tag{3}$$

Fig. 8 shows the environmental selection strategy for blue individuals in $\text{FNDS}^{\prec}$. Let u and v be two individuals in $\text{FNDS}^{\prec}$, and u' and v' are the corresponding mutated candidate solutions. In Fig. 8(a), if u' can dominate u, our algorithm

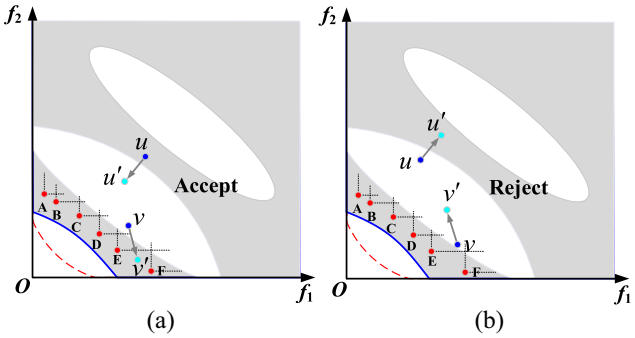


Fig. 8. (a) If FNDSDomCT(v') $\prec$ FNDSDomCT(v), v' will replace v; if FNDSDomCT(u') = FNDSDomCT(u), $u' \prec u$, u' will replace u. (b) If FNDSDomCT(u') = FNDSDomCT(u), $u \prec u'$, u' will be neglected; if FNDSDomCT(v') > FNDSDomCT(v), v' will be neglected.

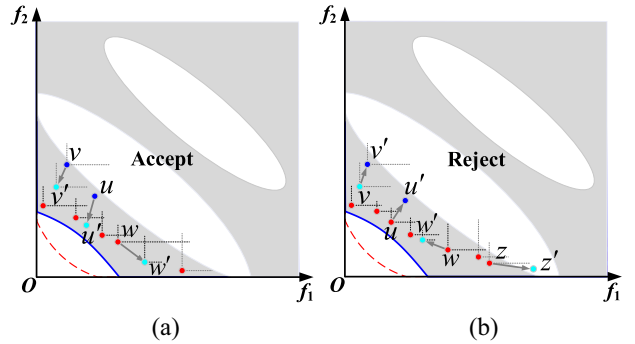


Fig. 9. (a) If FNDSDomCT(u') $\prec$ FNDSDomCT(u), u' will replace u; if FNDSDomCT(v') = FNDSDomCT(v), $CV(v') = CV(v) = 0$, $v' \prec v$, v' will replace v; if FNDSDomCT(w') = FNDSDomCT(w), $CV(w') = CV(w) = 0$, $MED(w') > MED(w)$, w' will replace w. (b) If FNDSDomCT(u') > FNDSDomCT(u), u' will be neglected. if FNDSDomCT(z') = FNDSDomCT(z), $CV(z') > CV(z)$, z' will be neglected; if FNDSDomCT(v') = FNDSDomCT(v), $CV(v') = CV(v) = 0$, $v \prec v'$, v' will be neglected; if FNDSDomCT(w') = FNDSDomCT(w), $CV(w') = CV(w) = 0$, $MED(w') < MED(w)$, w' will be neglected.

accepts the mutated candidate solution u' for a better convergence quality. As shown in Fig. 8(a), v is dominated by FNDS solutions D and E, whereas v' is not dominated by FNDS. Although v and v' are nondominated with respect to each other, the value of FNDSDomCT(v') is lower than that FNDSDomCT(v). Our algorithm replaces v with v' for better convergent quality.

By contrast, in Fig. 8(b), the mutated candidate solution u' is dominated by the original u, and the value of FNDSDomCT(v') is greater than FNDSDomCT(v). The proposed algorithm does not accept newly mutated u' and v'. If $P_t^{(i)}$ is an individual in FNDS$^{\prec}$, the environmental selection strategy is as follows:

$$P_t^{(i)} = \begin{cases} NP_t^{(i)}, \text{if FNDSDomCT}\left(NP_t^{(i)}\right) < \text{FNDSDomCT}\left(P_t^{(i)}\right) \\ NP_t^{(i)}, \text{if FNDSDomCT}\left(NP_t^{(i)}\right) = \text{FNDSDomCT}\left(P_t^{(i)}\right) \\ \qquad \text{and NP}_t^{(i)} \prec P_t^{(i)} \\ P_t^{(i)}, \qquad \text{else.} \end{cases} \tag{4}$$

Fig. 9 presents the environmental selection strategy for individuals in the FNDS. As shown in Fig. 9(a), u, v, and w are the three individuals in FNDS, and the u', v', and w' are the corresponding mutated candidate solutions that are still feasible. If the value of FNDSDomCT(u') is smaller than the value of FNDSDomCT(u), the original u is replaced by the mutated u' for better convergence. When the value of FNDSDomCT(v') is equal to the value of FNDSDomCT(v), and v' dominates v, the mutated solution v' can replace v for better convergence. When the value of FNDSDomCT(w') is equal to the value of FNDSDomCT(w), w', and w are nondominated with respect to each other, and the value of $MED(w')$ is greater than $MED(w)$, the mutated solution w' can replace w for better diversity [43], [44].

The maximum extension distance (MED) is designed to guide individuals to maintain a uniform distance and extend to approximate the entire PF automatically. MED is defined in (6)

$$\text{MED}\left(P_t^{(i)}\right) = \text{NearDist}\left(P_t^{(i)}\right) \times \text{TotalDist}\left(P_t^{(i)}\right) \tag{5}$$

where

$$\text{NearDist}\left(P_t^{(i)}\right) = \min_{j,j\neq i}\left(\sqrt{\sum_{m=1}^{M}\left(f_m^{(j)} - f_m^{(i)}\right)^2}\right) \tag{6}$$

$$\text{TotalDist}\left(P_t^{(i)}\right) = \sum_{j=1}^{P}\left(\sqrt{\sum_{m=1}^{M}\left(f_m^{(j)} - f_m^{(i)}\right)^2}\right). \tag{7}$$

In this equation, $P_t^{(i)}$ is the ith individual in population P_t at the t-th generation. TotalDist($P_t^{(i)}$) calculates the summation of Euclidean distances between $P_t^{(i)}$ and $P_t^{(j)}$, $j = 1, \ldots, P$. A greater value of TotalDist($P_t^{(i)}$) implies that solution $P_t^{(i)}$ has shifted away from other individuals. NearDist($P_t^{(i)}$) calculates the minimum Euclidean distance between $P_t^{(i)}$ and $P_t^{(j)}$, $j = 1, \ldots, P$ and $j \neq i$. A greater value of NearDist($P_t^{(i)}$) implies better individual distance.

By contrast, as shown in Fig. 9(b), u, v, w, and z be four individuals in FNDS, and the u', v', w', and z' are the corresponding mutated candidate solutions, respectively, which are still feasible solutions, except for z'. If the value of FNDSDomCT(u') is greater than the value of FNDSDomCT(u), the original u is neglected because of its poor convergence. When the value of FNDSDomCT(v') is equal to the value of FNDSDomCT(v), whereas v dominates v', the mutated solution v' is neglected for worse convergence. When w' and w are nondominated with respect to each other and the value of MED(w') is smaller than MED(w), the mutated solution w' should be neglected because of its poor diversity. Because the value of $CV(z')$ is greater than $CV(z)$, the mutated solution z' should be ignored. If $P_t^{(i)}$ is an individual in FNDS, the environmental selection strategy is as

follows:

$$P_t^{(i)} = \begin{cases} NP_t^{(i)}, & \text{if FNDSDomCT}\left(NP_t^{(i)}\right) < \text{FNDSDomCT}\left(P_t^{(i)}\right) \\ NP_t^{(i)}, & \text{if FNDSDomCT}\left(NP_t^{(i)}\right) = \text{FNDSDomCT}\left(P_t^{(i)}\right) \\ & \quad \text{and } CV\left(NP_t^{(i)}\right) = CV\left(P_t^{(i)}\right) \\ & \quad \text{and } NP_t^{(i)} \prec P_t^{(i)} \\ NP_t^{(i)}, & \text{if FNDSDomCT}\left(NP_t^{(i)}\right) = \text{FNDSDomCT}\left(P_t^{(i)}\right) \\ & \quad \text{and } CV\left(NP_t^{(i)}\right) = CV\left(P_t^{(i)}\right) \\ & \quad \text{and } NP_t^{(i)} \not\prec P_t^{(i)} P_t^{(i)} \not\prec NP_t^{(i)} \\ & \quad \text{and } MED\left(NP_t^{(i)}\right) > \text{MED}\left(P_t^{(i)}\right) \\ P_t^{(i)}, & \text{else.} \end{cases} \tag{8}$$

Please note that individuals in the proposed CHT need not to compromise the tradeoff between the three goals: 1) constrained satisfaction; 2) convergence; and 3) diversity.

1) The red individuals in FNDS are feasible nondominated solutions, which should prefer better convergence and diversity simultaneously.
2) Blue individuals are dominated by FNDS, which should pursue better convergence. Therefore, the individuals in FNDS$^{\prec}$ will not be trapped in the local feasible region and will quickly converge to the FNDS.
3) Green individuals are neither FNDS nor dominated by FNDS, thus, they should focus on better constraint satisfaction. The individuals in FNDS$^{\not\prec}$ will not be trapped in unconstrained PF and will shift rapidly toward the feasible region.

B. First Stage of CMOES

In the first stage, the proposed CMOES should focus on searching for an unconstrained PF without considering the constrained satisfaction. The computational complexity for each generation is $O(MP^2)$, where M is the number of objective functions and P is the population size. The pseudocode is shown in Algorithm 1.

The proposed algorithm adopts two mutation operators to generate new mutated solutions: 1) the Gaussian mutation operator [41] and 2) differential evolution (DE) mutation operator [42]. The Gaussian mutation operator can increase the exploitation ability, which creates a new mutated solution x_i' by adding the original x_i to a random number in the Gaussian distribution, as shown in (8)

$$x_i' = x_i + N(0, \sigma). \tag{9}$$

The DE operator can increase the exploration ability, in which three different solutions $P_t^{(r1)}$, $P_t^{(r2)}$, and $P_t^{(r3)}$ are chosen randomly (i.e., they are three different individuals), and the new offspring x' is generated by the three individuals with a random scaling factor, as shown in (9). The variable random represents a random real number between zero and one

$$x_i' = x_i^{(r1)} + \text{random} \times \left(x_i^{(r2)} - x_i^{(r3)}\right). \tag{10}$$

Algorithm 1: First Stage of CMOES

```
1  Initialization P_t, t =0
2  while ( t< 1/2 maximum generation) {
3      for I = 1 to P {
4          NP_t^(i) = GaussMutation(P_t^(i)) // generate new
             offspring
5          if (random < P_mutation) NP_t^(i) = DEMutation(P_t^(i))
6          Objective Functions Evaluation (NP_t^(i))
7          if (NP_t^(i) ≺ P_t^(i)) {
8              P_t^(i) = NP_t^(i) // better convergence
9          else if (NP_t^(i) ⊀ P_t^(i)) and (P_t^(i) ⊀ NP_t^(i))
10             if BeDomCT(NP_t^(i)) < BeDomCT(P_t^(i))
11                 P_t^(i) = NP_t^(i) // better convergence
12             else if BeDomCT(NP_t^(i)) = BeDomCT(P_t^(i))
13                 if MED(NP_t^(i)) > MED(P_t^(i))
14                     P_t^(i) = NP_t^(i) // better diversity
15                 end if
16             end if
17         end if
18     end for
19 end while
```

Let P_t be the population in the t-th generation and $P_t^{(i)}$ be the ith individual in P_t. The mutated candidate individual $NP_t^{(i)}$ is generated by the Gaussian mutation and DE mutation, as shown in lines 4 and 5 of Algorithm 1. The original $P_t^{(i)}$ is dominated by $NP_t^{(i)}$, and should be replaced by $NP_t^{(i)}$ for convergence improvement, as shown in lines 7 and 8 of Algorithm 1. If $P_t^{(i)}$ and $NP_t^{(i)}$ are nondominated with respect to each other, their BeDomCT() values should be further compared. The function BeDomCT($P_t^{(i)}$) counts the number of other solutions that dominate $P_t^{(i)}$ without considering the constraint satisfaction.

The proposed CMOES prefers to maintain the individual $NP_t^{(i)}$ with a smaller value of BeDomCT(), which implies that fewer solutions can dominate $NP_t^{(i)}$, as shown in lines 10 and 11 of Algorithm 1. If their values are equal, their distribution quality should be further compared. An individual with a greater MED value should be maintained for better individual distribution quality, as shown in lines 12 to 14 of Algorithm 1.

C. Second Stage of CMOES

In the second stage, the FNDS CHT was adopted to search for the global constrained PF. The computational complexity for each generation is $O(MP^2)$, where M is the number of objective functions and P is the population size. The pseudocode is shown in Algorithm 2.

After the feasible nondominated set *FNDS* is identified in each generation, individuals are divided into three subsets, as shown in lines 2 to 5 of Algorithm 2. The newly mutated solution is then generated with the Gaussian mutation and DE mutation operators, as shown in

Algorithm 2: Second Stage of CMOES

```
1  while ( t < maximum generation )
2      FNDS = FeasibleNonDominateSet()
3      for i = 1 to P
4          if FNDS ≺ P_t^(i) then FNDS^≺ = P_t^(i) ∪ FNDS^≺
5          if FNDS ⊀ P_t^(i) then FNDS^⊀ = P_t^(i) ∪ FNDS^⊀
             // divide the individual by FNDS dynamically
6          NP_t^(i) = GaussMutation(P_t^(i)) // generate new
             offspring
7          Objective Functions and CV Evaluation (NP_t^(i))
8          OldDomCT = FNDSDomCT(P_t^(i))
9          NewDomCT = FNDSDomCT(NP_t^(i))
10         if P_t^(i) ∈ FNDS^⊀
11             if (NewDomCT = OldDomCT) and
12                 (CV(NP_t^(i)) < CV(P_t^(i))) then
13                 P_t^(i) = NP_t^(i) // better constraint violation
14             end if
15         else if P_t^(i) ∈ FNDS^≺
16             if (NewDomCT < OldDomCT)
17                 P_t^(i) = NP_t^(i) // better convergence
18             else (NewDomCT = OldDomCT) and
19                 (NP_t^(i) ≺ P_t^(i)) then
20                 P_t^(i) = NP_t^(i) // better convergence
21             end if
22         else if P_t^(i) ∈ FNDS
23             if CV(NP_t^(i)) ≠ CV(P_t^(i)) then continue
24             if (NewDomCT < OldDomCT)
25                 P_t^(i) = NP_t^(i) // better convergence
26             else (NewDomCT = OldDomCT) and
27                 if (NP_t^(i) ≺ P_t^(i))
28                     P_t^(i) = NP_t^(i) // better convergence
29                 else (NP_t^(i) ⊀ P_t^(i)) and (P_t^(i) ⊀ NP_t^(i)) and
30                     (MED(NP_t^(i)) > MED(P_t^(i))) then
31                     P_t^(i) = NP_t^(i) // better diversity
32                 end if
33             end if
34         end if
35     end for
36 end while
```

line 6 of Algorithm 2. After the objective functions and CV are evaluated, the values of FNDSDomCT($P_t^{(i)}$) and FNDSDomCT($NP_t^{(i)}$) are calculated and assigned to the variables *OldDomCT* and *NewDomCT*, respectively, as shown in lines 8 and 9.

If $P_t^{(i)} \in FNDS^{\not\prec}$, no individual in *FNDS* can dominate $P_t^{(i)}$, and the value of *OldDomCT* is zero. The value of *NewDomCT* is equal to that of *OldDomCT*, which in turn can prevent the convergence from worsening. If $CV(NP_t^{(i)})$ is smaller than $CV(P_t^{(i)})$, the mutated $NP_t^{(i)}$ replaces the original $P_t^{(i)}$ for a better CV, as shown in lines 10 to 14.

If $P_t^{(i)} \in FNDS^{\prec}$, more than one individual in *FNDS* can dominate $P_t^{(i)}$, and the value of *OldDomCT* is greater than zero. When the value of FNDSDomCT($P_t^{(i)}$) decreases to zero, individual $P_t^{(i)}$ may become a feasible nondominated solution. Therefore, if the value of *NewDomCT* is less than the value of *OldDomCT*, the mutated $NP_t^{(i)}$ will replace the original $P_t^{(i)}$ for better convergence, as shown in lines 15 to 17. If the value of *NewDomCT* is equal to the value of *OldDomCT*, and $NP_t^{(i)}$ can dominate $P_t^{(i)}$, $P_t^{(i)}$ should be replaced by $NP_t^{(i)}$ for better convergence, as shown in lines 18 to 21.

If $P_t^{(i)} \in FNDS$, $P_t^{(i)}$ is a feasible solution, the value of $CV(P_t^{(i)})$ is zero. If the mutated $NP_t^{(i)}$ becomes an infeasible solution, the proposed CMOES should neglect it, as shown in lines 22 and 23. Alternatively, convergence and diversity qualities are considered in the subset *FNDS*. If the value of *NewDomCT* is less than the value of *OldDomCT*, the mutated $NP_t^{(i)}$ will replace the original $P_t^{(i)}$ for better convergence, as shown in lines 24 and 25. When the value of *NewDomCT* is equal to that of *OldDomCT*, if $NP_t^{(i)}$ can dominate $P_t^{(i)}$, $P_t^{(i)}$ should be replaced by $NP_t^{(i)}$ to achieve better convergence. If $NP_t^{(i)}$ and $P_t^{(i)}$ are nondominated with respect to each other and $MED(NP_t^{(i)})$ is greater than $MED(P_t^{(i)})$, $P_t^{(i)}$ should be replaced by $NP_t^{(i)}$ for a better distribution, as shown in lines 26 to 34.

We note that the computational complexity of the function *FeasibleNonDominateSet*() is $O(MP^2)$, where M is the number of objective functions, and P is the population size. The second stage of CMOES includes two functions, *FeasibleDomCT* and *MED*, and its time complexity is $O(MP)$. Therefore, the time complexity of the proposed CHT was $O(MP^2)$. Because both stages have the same time complexity $O(MP^2)$, the time complexity for the overall algorithm is also $O(MP^2)$.

IV. Experiment

A. Competing CMOEAs and Experiment Settings

The performance of the proposed CMOES was compared with that of nine state-of-the-art CMOEAs to validate its effectiveness and efficiency. The nine competing CMOEAs were ShiP [5], A-NSGA-III [7], MOEA/D-IEpsilon [11], MOEA/D-SR [10], POCEA [18], PPS [21], C-TAEA [23], CCMO [24], and DD-CMOEA [37]. These competing approaches are commonly regarded as well-established representatives employing different CHTs, including penalty, CDP, epsilon, SR, special representation and operator, multiple stages, and multiple populations.

These competing CMOEAs were evaluated on the CF [48], MW [49], and LIRCMOP [10] test suites, which consist of 38 test instances with large infeasible regions. In addition, the CMOP test instances have 2- or 3-D PFs, which are characterized as convex, concave, mixed, and disconnected. The inverted generational distance (IGD) [45] was adopted to evaluate the performance of the competing algorithms, as shown in (9). For each test function, 10 000 true Pareto optimal

solutions were sampled evenly

$$\mathrm{IGD} = \frac{\left(\sum_{i=1}^{|PF|} d_i^2\right)^{1/2}}{|PF|} \tag{11}$$

where d_i represents the Euclidean distance between the ith reference objective vector and nearest objective values obtained.

The experimental parameters are explained as follows.

1) The population size, P, was set to 100, and the maximum number of iterations was set to 1000. Therefore, the maximum number of function evaluations was 100 000.
2) For the Gaussian mutation, the distribution means μ was set to zero, and the standard deviation σ was set to 0.2. The SBX distribution index [46] and polynomial mutation distribution index [47] were both set to 20. The other parameters were chosen to be exactly the same as those in the original articles.
3) To ensure a fair comparison, all competing CMOEAs were run 30 times independently for every test instance.

The proposed CMOES algorithm was implemented and a corresponding GUI program was developed to facilitate the experiments. For convenience, interested readers can verify the performance of CMOES and obtain the source code from the GitHub website: https://github.com/MaOEA/CMOES.

B. Comparison Result on Benchmark Test Suites

The IGD values obtained by ten competing CMOEAs are shown in Table I. From 39 test instances of the CF, MW, and LIRCMOP benchmark suites, the proposed CMOES excelled in 19 test instances. For the remaining instances, DD-CMOEA and CCMO excelled in six test instances each. MOEADSR and PPS excelled in two and one test instance, respectively. The empirical results show that CMOES outperforms the other CMOEAs.

As shown in Table I, CMOES obtained significantly better IGD performance than the other CMOEAs on the six CF instances, two MW instances, and 11 LIRCMOP instances. The proposed CHT was validated using four types of CMOPs [49]. For Type-I CMOPs, the constrained PF is the same as the unconstrained PF. The proposed CMOES can maintain good diversity to avoid losing the unconstrained PF, such as MW14 and LIRCMOP6. For Type-II CMOPs, the constrained PF is part of the unconstrained PF. The proposed CMOES adopts a two-stage strategy to switch the objective priority to a constraint priority, such as CF10 and LIRCMOP9. For Type-III CMOPs, the constrained PF consists of part of the unconstrained PF and part of the boundary of the feasible region. The proposed CHT divides the population into three subsets dynamically and guides individuals to simultaneously pursue better objective values and lower CVs, such as LIRCMOP11 and LIRCMOP12. For Type-IV CMOPs, the unconstrained PF is located entirely outside the feasible region. In the second stage, the proposed CHT focuses on guiding the infeasible individuals in $FNDS^{\not\prec}$ shifting toward feasible regions, such as LIRCMOP4 and LIRCMOP7.

Fig. 10 shows the results obtained on the LIRCMOP test suite using the proposed CMOES. As shown in Fig. 10(a)–(d), although LIRCMOP1-4 have extremely large infeasible regions in the entire search space, the proposed CMOES can converge to very small feasible regions and maintain a good distribution on convex, concave, and mixed disconnected PFs. For LIRCMOP5-6, the constrained PFs are the same as the unconstrained PFs; therefore, the PFs can be obtained by the first stage in the CMOES. As shown in Fig. 10(e) and (f), although there are few individuals in $FNDS^{\not\prec}$ for the second stage, the proposed CMOES still performs well. By contrast, the constrained PFs of LIRCMOP7-8 are different from the unconstrained PFs, and the proposed two-stage strategy and FNDS CHT are very effective for finding the global optimal solutions in a feasible region.

As shown in Fig. 10(i)–(l), LIRCMOP9-12 has not only large infeasible regions, but also disconnected PFs. Moreover, the PFs of LIRCMOP9-10 were part of the unconstrained PFs, whereas the PFs of LIRCMOP11-12 were located on the constraint boundaries. In Fig. 10(i)–(l), we can observe that CMOES can find all disconnected constrained PFs efficiently and effectively in narrow and small feasible regions. Fig. 10(m) and (n) show the obtained results for LIRCMOP13-14, which are three-objective CMOP instances. The constrained PF of LIRCMOP13 is the same as that of the unconstrained PF, whereas the constrained PF of LIRCMOP14 is located at the boundaries of the feasible region. The proposed CMOES obtained good performance on these three-objective instances, and the solutions were well-converged and well-distributed over the entire 3-D concave PFs. For the LIRCMOP13-14, the individuals in FNDS should form a 3-D domination space, the proposed FNDS constraint handling technique can still guide the $FNDS^{\prec}$ and FNDS to converge to the PF effectively. Moreover, the two-stage strategy is helpful in directing the population through the large-scale infeasible region and reaching the boundaries of the unconstrained PF.

Fig. 11 shows the results obtained for LIRCMOP8 using the competing ten CMOEAs. LIRCMOP8 not only has a very large infeasible region but also an entirely unconstrained PF located outside the feasible region. Four CMOEAs, ANSGAIII, POCEA, MOEADSR, and MOEAD-IEpsilon, only obtained local feasible optimal solutions. Four other CMOEAs, PPS, CTAEA, CCMO, and DDCOMEA, can obtain part of the global optimal solutions, although the diversity is extremely poor. With limited fitness evaluations, only a portion of the solutions obtained by ShiP are located in the global optimal region, whereas the others are still located in the local optimal region. The proposed CMOES achieved the best performance on LIRCMOP8; the solutions were well converged and well distributed on the constrained PF.

C. Discussion on the Friedman Statistical Test

In the comparison experiment, 38 CMOPs in three test suites were adopted to evaluate the performance of ten competing CMOEAs. Fig. 12 shows the average ranking of the Friedman statistical test for multiple comparisons. In Fig. 12, the

TABLE I
AVERAGE IGD VALUES OVER 30 RUNS ON BENCHMARK INSTANCES (POPULATION SIZE 100), WHERE THE BEST MEAN IS SHOWN IN A GRAY BACKGROUND

Problem	CMOES	ShiP	ANSGAIII	MOEADIEpsilon	MOEADSR	POCEA	PPS	CTAEA	CCMO	DDCMOEA
CF1	0.0028	0.0261(+)	0.0189(+)	0.0032(+)	0.0016(-)	0.0541(+)	0.0046(+)	0.0435(+)	0.0055(+)	0.0055(+)
CF2	0.0098	0.0410(+)	0.1068(+)	0.0041(-)	0.0039(-)	0.0766(+)	0.0043(-)	0.0779(+)	0.0526(+)	0.0397(+)
CF3	0.2414	0.2312(-)	0.3315(+)	0.2736(+)	0.2212(-)	0.4707(+)	0.4811(+)	0.1652(-)	0.1865(-)	0.1756(-)
CF4	0.0124	0.0884(+)	0.0706(+)	0.2139(+)	0.0480(+)	0.0996(+)	0.0349(+)	0.0686(+)	0.1009(+)	0.0916(+)
CF5	0.2603	0.2791(+)	0.2574(-)	0.3057(+)	0.2931(+)	0.3591(+)	0.2984(+)	0.1751(-)	0.3462(+)	0.3106(+)
CF6	0.0170	0.1094(+)	0.0706(+)	0.0568(+)	0.0915(+)	0.2364(+)	0.0563(+)	0.0564(+)	0.0574(+)	0.1226(+)
CF7	0.0819	0.2949(+)	0.4426(+)	0.6716(+)	0.2776(+)	0.3827(+)	0.2081(+)	0.3476(+)	0.3033(+)	0.3337(+)
CF8	0.1071	0.3723(+)	0.3836(+)	0.1887(+)	0.1937(+)	0.5342(+)	0.2043(+)	0.4768(+)	0.2337(+)	0.4220(+)
CF9	0.0505	0.1476(+)	0.1105(+)	0.1278(+)	0.1504(+)	0.2868(+)	0.1235(+)	0.1398(+)	0.0906(+)	0.1132(+)
CF10	0.1954	NaN	NaN	NaN	NaN	0.3667(+)	0.3027(+)	0.2147(+)	0.5593(+)	0.5578(+)
MW1	0.0057	0.2762(+)	0.0021(-)	0.0038(-)	0.0030(-)	0.0143(+)	0.0031(-)	0.0019(-)	0.0016(-)	0.0017(-)
MW2	0.0217	0.0239(+)	0.0982(+)	0.2043(+)	0.0886(+)	0.0392(+)	0.1127(+)	0.0115(-)	0.0208(-)	0.0038(-)
MW3	0.0058	0.0055(-)	0.0073(+)	0.0059(=)	0.9194(+)	0.0134(+)	0.0059(=)	0.0053(-)	0.0052(-)	0.0049(-)
MW4	0.1571	0.0541(-)	0.0428(-)	0.0576(-)	0.0636(-)	0.0564(-)	0.0593(-)	0.0463(-)	0.0426(-)	0.0425(-)
MW5	0.0169	0.7431(+)	0.2097(+)	0.7347(+)	0.0124(-)	0.065(+)	0.7385(+)	0.0097(-)	0.0005(-)	0.0007(-)
MW6	0.0102	0.0304(+)	0.0383(+)	0.5765(+)	1.0608(+)	1.0681(+)	0.5452(+)	0.0166(+)	0.0151(+)	0.0233(+)
MW7	0.0064	0.0052(-)	0.0048(-)	0.0055(-)	0.0052(-)	0.0135(+)	0.0049(-)	0.0069(-)	0.0045(-)	0.0045(-)
MW8	0.0468	0.0791(+)	0.4463(+)	0.1240(+)	0.1801(+)	0.0653(+)	0.2114(+)	0.0531(+)	0.0557(+)	0.0473(+)
MW9	0.0087	0.0202(+)	0.0226(+)	0.0097(+)	0.0088(=)	0.0334(+)	0.0123(+)	0.0069(-)	0.0049(-)	0.0045(-)
MW10	0.0254	0.1242(+)	0.0782(+)	0.6253(+)	0.2714(+)	0.2928(+)	0.2391(+)	0.0374(+)	0.0848(+)	0.0359(+)
MW11	0.0113	0.0138(+)	0.6993(+)	0.0070(-)	0.0076(-)	0.0544(+)	0.0072(-)	0.0134(+)	0.0059(-)	0.0053(-)
MW12	0.0074	0.0055(-)	0.0064(-)	0.0068(-)	0.0074(=)	0.0151(+)	0.0104(+)	0.0082(+)	0.0046(-)	0.0048(-)
MW13	0.0442	0.1224(+)	0.1512(+)	1.1874(+)	0.1422(+)	0.3207(+)	0.4112(+)	0.0516(+)	0.0756(+)	0.0878(+)
MW14	0.0922	0.1353(+)	0.1148(+)	0.1371(+)	0.1426(+)	0.1465(+)	0.1603(+)	0.1053(+)	0.0993(+)	0.1045(+)
LIRCMOP1	0.0072	NaN	0.2768(+)	0.0094(+)	0.2801(+)	0.0421(+)	0.0715(+)	0.1097(+)	0.2844(+)	0.3647(+)
LIRCMOP2	0.0088	NaN	0.2930(+)	0.1550(+)	0.1952(+)	0.1084(+)	0.0158(+)	0.1604(+)	0.2356(+)	0.3187(+)
LIRCMOP3	0.0038	NaN	0.3396(+)	0.1601(+)	0.3268(+)	0.1496(+)	0.0193(+)	0.3784(+)	0.3701(+)	0.3282(+)
LIRCMOP4	0.0028	NaN	0.3212(+)	0.1529(+)	0.2434(+)	0.3702(+)	0.0151(+)	0.3186(+)	0.268(+)	0.1744(+)
LIRCMOP5	0.0209	1.2271(+)	1.2349(+)	1.2176(+)	1.2208(+)	0.2240(+)	0.0066(-)	1.2163(+)	0.3044(+)	0.2188(+)
LIRCMOP6	0.0143	1.3456(+)	1.3456(+)	1.3465(+)	1.3458(+)	1.3513(+)	0.0186(+)	15.517(+)	0.3961(+)	0.2891(+)
LIRCMOP7	0.0144	0.1265(+)	0.2044(+)	1.6813(+)	1.6813(+)	0.1594(+)	0.2306(+)	0.132(+)	0.1205(+)	0.1339(+)
LIRCMOP8	0.0419	1.6817(+)	1.6826(+)	1.6827(+)	1.6828(+)	1.6926(+)	0.2509(+)	1.6856(+)	0.1821(+)	0.1088(+)
LIRCMOP9	0.0328	1.0344(+)	1.0389(+)	0.7516(+)	0.6193(+)	0.4238(+)	0.3856(+)	0.3926(+)	0.3652(+)	0.6369(+)
LIRCMOP10	0.0096	0.9918(+)	0.9516(+)	0.4115(+)	0.4109(+)	0.3235(+)	0.4109(+)	0.3555(+)	0.0803(+)	0.1298(+)
LIRCMOP11	0.0064	0.7962(+)	0.9049(+)	0.2824(+)	0.2823(+)	0.6879(+)	0.0724(+)	0.1771(+)	0.1601(+)	0.0741(+)
LIRCMOP12	0.0055	0.7059(+)	0.7286(+)	0.3672(+)	0.2502(+)	0.8052(+)	0.2497(+)	0.1731(+)	0.1337(+)	0.1455(+)
LIRCMOP13	0.1392	1.3255(+)	1.322(+)	1.3167(+)	1.3303(+)	1.3484(+)	1.3222(+)	0.1071(-)	0.0969(-)	0.0977(-)
LIRCMOP14	0.1378	1.2781(+)	1.2722(+)	1.2742(+)	1.2812(+)	0.1078(-)	0.1126(-)	0.1109(-)	0.1022(-)	0.1002(-)

Wilcoxon's rank sum test at a 0.05 significance level is performed between CMOES and compared CMOEAs. "+" means CMOES better than compared algorithm, "–" means compared algorithm better than CMOES, "=" means not comparable)

proposed CMOES exhibits the lowest ranking and significantly better performance than the other nine CMOEAs. CCMO and DD-CMOEA also obtained a good average ranking, second and third, respectively.

To further analyze the significant differences for pairwise comparisons, the Holm test was adopted as a *post-hoc* test, as shown in Table II. Notably, a Bonferroni correction for multiple testing was applied (the *p*-values in the "Sig" column

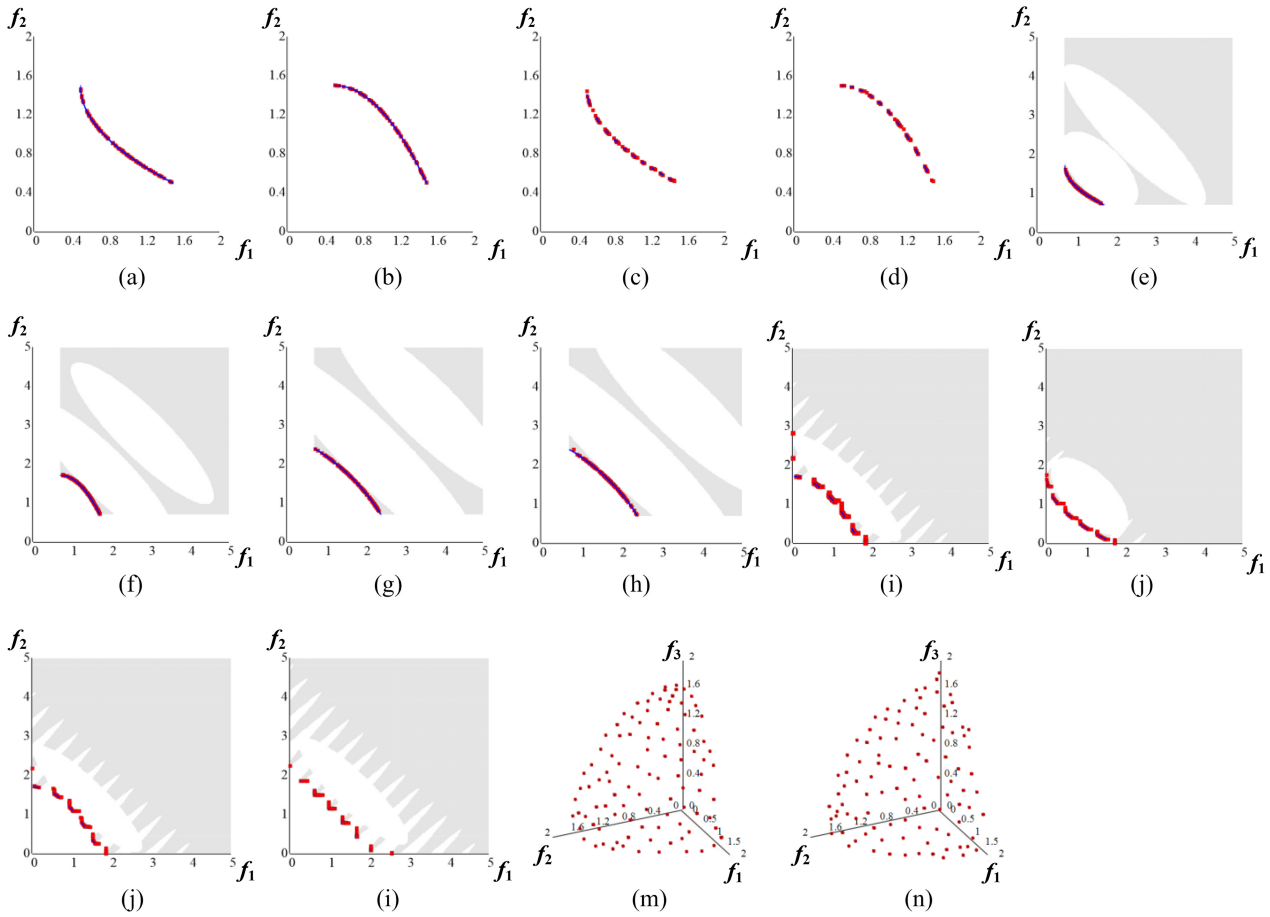


Fig. 10. Experimental results on LIRCMOP test suite by CMOES with population size 100. (a) LIRCMOP1. (b) LIRCMOP2. (c) LIRCMOP3. (d) LIRCMOP4. (e) LIRCMOP5. (f) LIRCMOP6. (g) LIRCMOP7. (h) LIRCMOP8. (i) LIRCMOP9. (j) LIRCMOP10. (k) LIRCMOP11. (l) LIRCMOP12. (m) LIRCMOP13. and (n) LIRCMOP14.

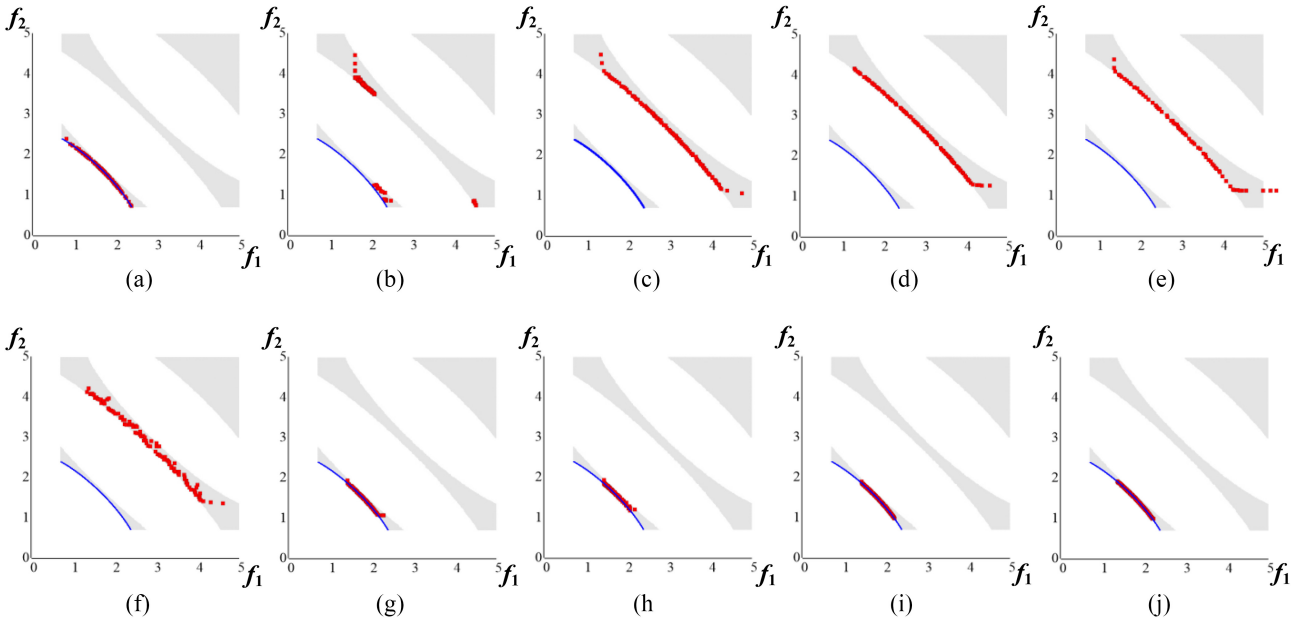


Fig. 11. Experimental results on LIRCMOP8 by ten competing CMOEAs with population size 100. (a) CMOES. (b) ShiP. (c) A-NSGA-III. (d) MOEA/D-IEpsilon. (e) MOEA/D-SR. (f) POCEA. (g) PPS. (h) C-TAEA. (i) CCMO. and (j) DD-CMOEA.

are multiplied by the number of tests being performed to provide the "Adj. Sig" p-values). As the adjusted p-values were smaller than 0.05, there remained a significant difference between CMOES and six competing CMOEAs, including ShiP, ANSGA-III, MOEAD-IEpsilon, MEAD-SR, POCEA, and PPS.

TABLE II
RESULTS OF EFFECT SIZE FOR THE FRIEDMAN TEST AND HOLM TEST WITH BONFERRONI CORRECTION

Sample 1-Sample 2	Test Statistic	Std. Error	Std. Test Statistic	Sig.	Adj. Sig.	Kendall's W (effect size)	Degree of difference
CMOES-ShiP	4.157895	0.694591	5.986108	0.000000	0.000000	0.542936	strong
CMOES-A-NSGA-III	4.105263	0.694591	5.910335	0.000000	0.000000	0.542936	strong
CMOES-MOEA/D-IEpsilon	3.763158	0.694591	5.417807	0.000000	0.000003	0.468144	moderate
CMOES-MOEA/D-SR	3.552632	0.694591	5.114713	0.000000	0.000014	0.277008	small
CMOES-POCEA	4.842105	0.694591	6.971164	0.000000	0.000000	0.800554	strong
CMOES-PPS	2.394737	0.694591	3.447695	0.000565	0.025443	0.398892	moderate
CMOES-C-TAEA	2.184211	0.694591	3.144601	0.001663	0.074841	0.224377	small
CMOES-CCMO	1.000000	0.694591	1.439697	0.149953	1.000000	0.135734	small
CMOES-DD-CMOEA	1.105263	0.694591	1.591244	0.111555	1.000000	0.135734	small

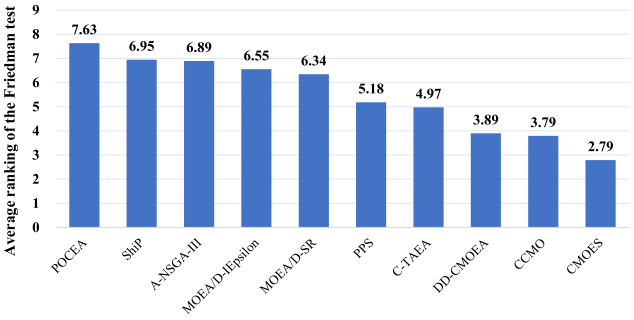


Fig. 12. Average ranking of the Friedman test by ten competing CMOEAs on 38 CMOPs.

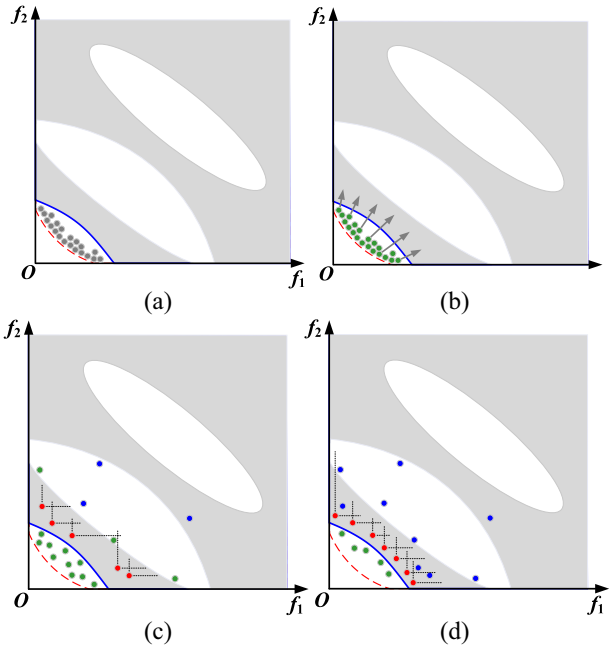


Fig. 13. (a) In the first scenario, all individuals have crossed the CPF and reached the UPF, FNDS does not exist and all the individuals belong to FNDS$^{\not\prec}$. (b) During the second stage, all these individuals shift toward the feasible region with better constraint satisfaction. (c) Some individuals could reach the global feasible region, becoming the individuals in FNDS. (d) With the FNDS constraint handling technique, more individuals should become FNDS.

D. Discussion on the Effectiveness of FNDS CHT

The proposed CMOES divides the population into three mutually exclusive subsets: 1) FNDS; 2) FNDS$^{\prec}$; and 3) FNDS$^{\not\prec}$ dynamically, and different strategies were applied to these subsets. However, during the first stage, the population may cross the CPF and reach the UPF, or still not have crossed the infeasible region and be located outside the feasible region. In these cases, the proposed FNDS CHT can still function properly, although no feasible nondominated solution exists at the beginning of the second stage.

In the first scenario, the entire population may converge to an infeasible region near the unconstrained PF in the first stage, as shown in Fig. 13(a). When there is no feasible solution, the subset FNDS must be an empty set, i.e., FNDS $= \emptyset$. Therefore, no individual is dominated by FNDS, the subset FNDS$^{\prec}$ is also an empty set, i.e., FNDS$^{\prec} = \emptyset$. In this situation, all the individuals are infeasible solutions and belongs to FNDS$^{\not\prec}$. Although all individuals are in FNDS$^{\not\prec}$, these infeasible individuals should shift toward the feasible region with better constraint satisfaction during the second stage, as shown in Fig. 13(b).

Some of the mutated individuals may shift to the global feasible region, whereas a few individuals may also shift toward local feasible or infeasible regions with smaller CV values. Therefore, an increasing number of infeasible individuals become FNDS, as shown in Fig. 13(c) and (d).

Although the proposed two-stage algorithm attempts to reach the unconstrained PF in the first stage, the feasible nondominated solutions may only be located in the local feasible region and not the global feasible region. If feasible individuals exist near the global feasible region, the proposed FNDS CHT performs satisfactorily.

Under the second scenario, the entire population may be located outside the global feasible region that does not cross the infeasible region, as shown in Fig. 14(a) and (b). During the second stage, the infeasible individuals in FNDS$^{\not\prec}$ should shift toward the feasible region with better constraint satisfaction, as shown in Fig. 14(c). Mutated individuals may shift to global or local feasible regions randomly, with smaller

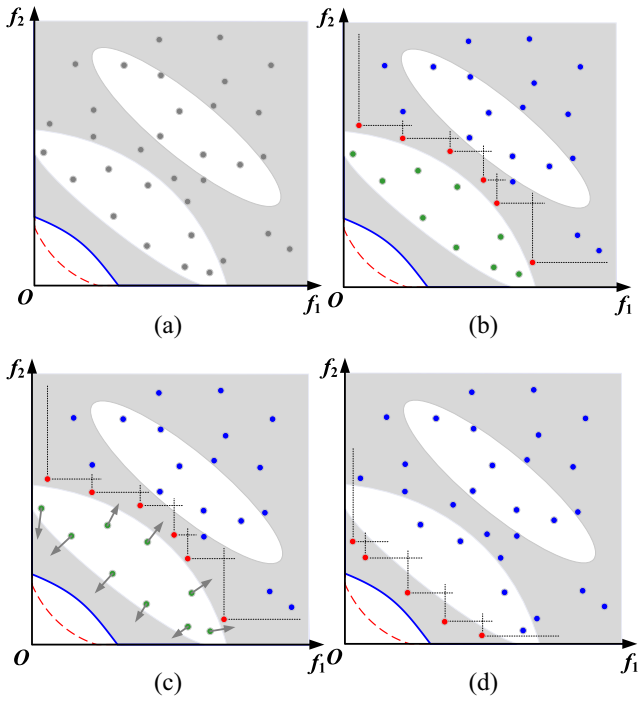


Fig. 14. (a) Under the second scenario, the entire population may locate outside the global feasible region which have not crossed the infeasible region. (b) FNDS is only located in the local feasible region. (c) In the second stage, these individuals in FNDS$^{\not\prec}$ should shift toward the feasible region with better constraint satisfaction. (d) When some individuals reach the global feasible region, the FNDS should change from the local feasible region to the global feasible region.

CV values. When some individuals reach the global feasible region, the FNDS changes from a local feasible region to a global feasible region, as shown in Fig. 14(d).

V. Conclusion

In this study, a two-stage CMOEA was proposed for solving CMOPs based on an evolutionary strategy. First, a parameter-less CHT is designed for constrained multiobjective optimization, which is based on an FNDS. The FNDS CHT divides the entire population into three mutually exclusive subsets: 1) FNDS; 2) FNDS$^{\prec}$; and 3) FNDS$^{\not\prec}$, dynamically, then different environmental selection strategies are applied to these subsets. The individuals in subset FNDS$^{\prec}$ are dominated by FNDS and are instructed to pursue only better convergence toward FNDS. The individuals in subset FNDS$^{\not\prec}$ are neither FNDS nor FNDS$^{\prec}$, which are always located in converged infeasible regions. These infeasible individuals in FNDS$^{\not\prec}$ need to focus only on better constraint satisfaction. The individuals in the subset FNDS are dedicated to searching for well-distributed and well-converged feasible nondominated solutions. Second, the proposed algorithm solves CMOPs in two stages. In the first stage, the proposed algorithm focuses solely on converging to an unconstrained PF without considering constrained satisfaction. In the second stage, the FNDS CHT is adopted to guide the population converging toward the global constrained PF effectively. The performance of the CMOES was compared with eight state-of-the-art CMOEAs. The experimental results confirm that the proposed CMOES outperforms the chosen competing CMOEAs on 14 LIRCMOP test instances with IGD performance indicators.

Although CMOES can achieve competitive performance on CF, MW, and LIRCMOP test suites, it is still challenging to satisfy and balance the three criteria simultaneously: 1) constrained satisfaction; 2) good convergence; and 3) good distribution. Clearly, CMOES prioritizes constrained satisfaction and convergence, although diversity performance still has scope for improvement. In addition, we plan to exploit the proposed CHT built into other dynamic, many-objective, or large-scale EMO frameworks for solving more real-world CMOPs. It is also planned to develop a knee-based CMOEA to alleviate the burden of decision makers.

References

[1] A. Kumar *et al.*, "A benchmark-suite of real-world constrained multi-objective optimization problems and some baseline results," *Swarm Evol. Comput.*, vol. 67, Dec. 2021, Art. no. 100961.
[2] A. Homaifar, C. X. Qi, and S. H. Lai, "Constrained optimization via genetic algorithms," *Simulation*, vol. 62, no. 4, pp. 242–253, 1994.
[3] B. Tessema and G. G. Yen, "An adaptive penalty formulation for constrained evolutionary optimization," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 39, no. 3, pp. 565–578, May 2009.
[4] Y. G. Woldesenbet, G. G. Yen, and B. G. Tessema, "Constraint handling in multiobjective evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 514–525, Jun. 2009.
[5] Z. Ma and Y. Wang, "Shift-based penalty for evolutionary constrained multiobjective optimization and its application," *IEEE Trans. Cybern.*, early access, May 25, 2021, doi: 10.1109/TCYB.2021.3069814.
[6] Q. Chen, J. Ding, S. Yang, and T. Chai, "A novel evolutionary algorithm for dynamic constrained multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 4, pp. 792–806, Aug. 2020.
[7] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 602–622, Aug. 2014.
[8] Z. Fan, Y. Fang, W. Li, X. Cai, C. Wei, and E. Goodman, "MOEA/D with angle-based constrained dominance principle for constrained multi-objective optimization problems," *Appl. Soft Comput.*, vol. 74, pp. 621–633, Jan. 2019.
[9] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 4, no. 3, pp. 284–294, Sep. 2000.
[10] M. A. Jan and R. A. Khanum, "A study of two penalty-parameterless constraint handling techniques in the framework of MOEA/D," *Appl. Soft Comput.*, vol. 13, no. 1, pp. 128–148, Jan. 2013.
[11] Z. Fan *et al.*, "An improved epsilon constraint-handling method in MOEA/D for CMOPs with large infeasible regions," *Soft Comput.*, vol. 23, no. 23, pp. 12491–12510, Dec. 2019.
[12] Q. Zhu, Q. Zhang, and Q. Lin, "A constrained multiobjective evolutionary algorithm with detect-and-escape strategy," *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 938–947, Oct. 2020.
[13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[14] S. Y. Zeng, L. S. Kang, and L. X. Ding, "An orthogonal multi-objective evolutionary algorithm for multi-objective optimization problems with constraints," *Evol. Comput.*, vol. 12, no. 1, pp. 77–98, 2004.
[15] Y. Lin, W. Du, and W. Du, "Multi-objective differential evolution with dynamic hybrid constraint handling mechanism," *Soft Comput.*, vol. 23, no. 12, pp. 4341–4355, Jun. 2019.
[16] Y. Yang, J. Liu, S. Tan, and H. Wang, "A multi-objective differential evolutionary algorithm for constrained multi-objective optimization problems with low feasible ratio," *Appl. Soft Comput.*, vol. 80, pp. 42–56, Jul. 2019.
[17] X. Yu, X. Yu, Y. Lu, G. G. Yen, and M. Cai, "Differential evolution mutation operators for constrained multi-objective optimization," *Appl. Soft Comput.*, vol. 67, Jun. 2018, pp. 452–466.

[18] C. He, R. Cheng, Y. Tian, X. Zhang, K. C. Tan, and Y. Jin, "Paired offspring generation for constrained large-scale multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 25, no. 3, pp. 448–462, Jun. 2021.

[19] H. K. Singh, T. Ray, and W. Smith, "C-PSA: Constrained Pareto simulated annealing for constrained multi-objective optimization," *Inform. Sci.*, vol. 180, no. 13, pp. 2499–2513, Jul. 2010.

[20] L. Jiao, J. Luo, R. Shang, and F. Liu, "A modified objective function method with feasible-guiding strategy to solve constrained multi-objective optimization problems," *Appl. Soft Comput.*, vol. 14, pp. 363–380, Jan. 2014.

[21] Z. Fan *et al.*, "Push and pull search for solving constrained multi-objective optimization problems," *Swarm Evol. Comput.*, vol. 44, pp. 665–679, Feb. 2019.

[22] Z. Liu and Y. Wang, "Handling constrained multiobjective optimization problems with constraints in both the decision and objective spaces," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 870–884, Oct. 2019.

[23] K. Li, R. Chen, G. Fu, and X. Yao, "Two-archive evolutionary algorithm for constrained multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 303–315, Apr. 2019.

[24] Y. Tian, T. Zhang, J. Xiao, X. Zhang, and Y. Jin, "A coevolutionary framework for constrained multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 25, no. 1, pp. 102–116, Feb. 2021.

[25] J. Yuan, H.-L. Liu, Y.-S. Ong, and Z. He, "Indicator-based evolutionary algorithm for solving constrained multi-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 26, no. 2, pp. 379–391, Apr. 2022.

[26] Z.-Z. Liu, Y. Wang, and B.-C. Wang, "Indicator-based constrained multiobjective evolutionary algorithms," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 9, pp. 5414–5426, Sep. 2021.

[27] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]," *IEEE Comput. Intell. Mag.*, vol. 12, no. 4, pp. 73–87, Nov. 2017.

[28] Z. Ma, Y. Wang, and W. Song, "A new fitness function with two rankings for evolutionary constrained multiobjective optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 8, pp. 5005–5016, Aug. 2021.

[29] Z. Fan *et al.*, "Push and pull search embedded in an M2M framework for solving constrained multi-objective optimization problems," *Swarm Evol. Comput.*, vol. 54, May 2020, Art. no. 100651.

[30] Y. Tian, Y. Zhang, Y. Su, X. Zhang, K. C. Tan, and Y. Jin, "Balancing objective optimization and constraint satisfaction in constrained evolutionary multiobjective optimization," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 9559–9572, Sep. 2022.

[31] M. Daneshyari and G. G. Yen, "Constrained multiple-swarm particle swarm optimization within a cultural frmaework," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 42, no. 2, pp. 475–490, Mar. 2012.

[32] W. Zhang, G. G. Yen, and Z. He, "Constrained optimization via artificial immune system," *IEEE Trans. Cybern.*, vol. 44, no. 2, pp. 185–198, Feb. 2014.

[33] J. Wang, Y. Li, Q. Zhang, Z. Zhang, and S. Gao, "Cooperative multiobjective evolutionary algorithm with propulsive population for constrained multiobjective optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 6, pp. 3476–3491, Jun. 2022.

[34] W. Gao, G. G. Yen, and S.-Y. Liu, "A dual-population differential evolution with coevolution for constrained optimization," *IEEE Trans. Cybern.*, vol. 45, no. 5, pp. 1094–1107, May 2015.

[35] H. Ma, H. Wei, Y. Tian, R. Cheng, and X. Zhang, "A multi-stage evolutionary algorithm for multi-objective optimization with complex constraints," *Inf. Sci.*, vol. 560, pp. 68–91, Jun. 2021.

[36] Y. Wang, B. C. Wang, H. X. Li, and G. G. Yen, "Incorporating objective function information into the feasibility rule for constrained evolutionary optimization," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2938–2952, Dec. 2016.

[37] M. Ming, R. Wang, H. Ishibuchi, and T. Zhang, "A novel dual-stage dual-population evolutionary algorithm for constrained multi-objective optimization," *IEEE Trans. Evol. Comput.*, early access, Nov. 29, 2021, doi: 10.1109/TEVC.2021.3131124.

[38] Y. Zhou, M. Zhu, J. Wang, Z. Zhang, Y. Xiang, and J. Zhang, "Trigoal evolution framework for constrained many-objective optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 8, pp. 3086–3099, Aug. 2020.

[39] Z.-Z. Liu, B.-C. Wang, and K. Tang, "Handling constrained multiobjective optimization problems via bidirectional coevolution," *IEEE Trans. Cybern.*, early access, Apr. 6, 2021, doi: 10.1109/TCYB.2021.3056176.

[40] K. Yu, J. Liang, B. Qu, Y. Luo, and C. Yue, "Dynamic selection preference-assisted constrained multiobjective differential evolution," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 5, pp. 2954–2965, May 2022.

[41] K. Deb and D. Deb, "Analysing mutation schemes for real-parameter genetic algorithms," *J. Artif. Intell. Soft.*, vol. 4, no. 1, pp. 1–28, Jan. 2014.

[42] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[43] K. Zhang, Z. Xu, S. Xie, and G. G. Yen, "Evolution strategy-based many-objective evolutionary algorithm through vector equilibrium," *IEEE Trans. Cybern.*, vol. 51, no. 11, pp. 5455–5467, Nov. 2021.

[44] K. Zhang, C. Shen, X. Liu, and G. G. Yen, "Multiobjective evolution strategy for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 974–988, Oct. 2020.

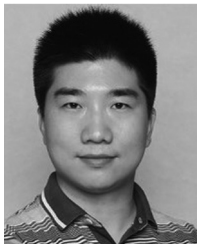
[45] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.

[46] K. Deb and R. Agrawal, "Simulated binary crossover for continuous search space," *Complex Syst.*, vol. 9, no. 2, pp. 115–148, 1995.

[47] K. Deb and S. Tiwari, "Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization," *Eur. J. Oper. Res.*, vol. 185, no. 3, pp. 1062–1087, Mar. 2008.

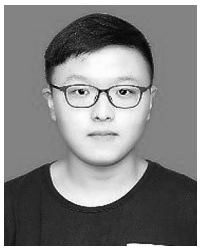
[48] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multiobjective optimization test instances for the CEC 2009 special session and competition," School Comput. Sci. Electron. Eng., Univ. Essex, Colchester, U.K., Rep. CES-487, 2009.

[49] Z. Ma and Y. Wang, "Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons," *IEEE Trans. Evol. Comput.*, vol. 23, no. 6, pp. 972–986, Dec. 2019.



Kai Zhang received the Ph.D. degree in system analyses and integrate from the Huazhong University of Science and Technology, Wuhan, China, in 2008.

He was a Postdoctoral Research Fellow with the School of Electronics Engineering and Computer Science, Peking University, Beijing, China, from 2008 to 2010. He is currently a Professor with the School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan. His research focuses on evolutionary computation and multicriteria decision making.



Zhiwei Xu (Graduate Student Member, IEEE) received the B.S. degree in information security from the Wuhan University of Science and Technology, Wuhan, China, in 2017, where he is currently pursuing the Ph.D. degree with the Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System.

His research interests include evolutionary computation, and many-objective optimization.

Gary G. Yen (Fellow, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Notre Dame, Notre Dame, IN, USA, in 1992.

He is currently a Regents Professor with the School of Electrical and Computer Engineering, Oklahoma State University (OSU), Stillwater, OK, USA. Before joined OSU in 1997, he was with the Structure Control Division, U.S. Air Force Research Laboratory in Albuquerque, NM, USA. His research interest includes intelligent control, computational intelligence, conditional health monitoring, and signal processing and their industrial/defense applications.

Dr. Yen received the Andrew P. Sage Best Transactions Paper Award from IEEE Systems, Man, and Cybernetics Society in 2011. In 2014, he received the Meritorious Service Award from IEEE Computational Intelligence Society. He was an Associate Editor of the *IEEE Control Systems Magazine*, IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, *Automatica*, *Mechantronics*, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, PART B: CYBERNETICS, and IEEE TRANSACTIONS ON NEURAL NETWORKS. He is currently serving as an Associate Editor for the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and IEEE TRANSACTIONS ON CYBERNETICS. He served as the General Chair for the 2003 IEEE International Symposium on Intelligent Control held in Houston, TX, USA, and 2006 IEEE World Congress on Computational Intelligence held in Vancouver, BC, Canada. He served as a Vice President for the Technical Activities from 2005 to 2006 and the President of the IEEE Computational Intelligence Society from 2010 to 2011 and is the Founding Editor-in-Chief of the *IEEE Computational Intelligence Magazine* from 2006 to 2009. He is a Fellow of IET.

Ling Zhang received the B.S. degree in computer engineering technology with minor in electrical engineering from Indiana State University, Terre Haute, IN, USA, in 2014, and the M.S. degree in data science from Michigan Technological University, Houghton, MI, USA, in 2016. He is currently pursuing the Ph.D. degree with the Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan University of Science and Technology, Wuhan, China.

His interests include deep learning, image processing, and evolutionary computation.