A Survey on Evolutionary Constrained Multiobjective Optimization

Jing Liang, *Senior Member, IEEE*, Xuanxuan Ban, *Graduate Student Member, IEEE*, Kunjie Yu, *Member, IEEE*, Boyang Qu, *Member, IEEE*, Kangjia Qiao, *Member, IEEE*, Caitong Yue, *Member, IEEE*, Ke Chen, *Member, IEEE*, and Kay Chen Tan, *Fellow, IEEE*

***Abstract*—Handling constrained multiobjective optimization problems (CMOPs) is extremely challenging, since multiple conflicting objectives subject to various constraints require to be simultaneously optimized. To deal with CMOPs, numerous constrained multiobjective evolutionary algorithms (CMOEAs) have been proposed in recent years, and they have achieved promising performance. However, there has been few literature on the systematic review of the related studies currently. This article provides a comprehensive survey for evolutionary constrained multiobjective optimization. We first review a large number of CMOEAs through categorization and analyze their advantages and drawbacks in each category. Then, we summarize the benchmark test problems and investigate the performance of different constraint handling techniques (CHTs) and different algorithms, followed by some emerging and representative applications of CMOEAs. Finally, we discuss some new challenges and point out some directions of the future research in the field of evolutionary constrained multiobjective optimization.**

***Index Terms*—Benchmark test problems, constrained multiobjective optimization, constraint handling, evolutionary algorithms.**

I. Introduction

A VARIETY of optimization problems in real-world applications contains several conflicting objectives and multiple complicate constraints, such as the robot gripper optimization problem [1], the combined economic emission dispatch (CEED) problem [2], the urban bus scheduling problem [3], and energy saving optimization problem [4]. This kind of problems is denoted as constrained multiobjective optimization problems (CMOPs). CMOPs bring more challenges than their unconstrained counterparts, due to the coexistence of multiple objectives and constraints. For CMOPs, because of the conflict among multiple objectives, there exist no single solution but a set of solutions that indicates the best possible tradeoffs among the objectives. The set of solutions forms the so-called Pareto-optimal set (PS), and its image in objective space is called the Pareto front (PF). Meanwhile, because the presence of the constraints can change the landscape of the search region and cause infeasible regions, the feasibility of the solutions cannot be neglected. Consequently, the final goal of solving CMOPs is to obtain a set of feasible solutions to approach the PF with good convergence and diversity.

In the past two decades, multiobjective evolutionary algorithms (MOEAs) have become a very popular alternative for addressing the multiobjective optimization problems [5], [6]. The most important reason is that MOEAs belong to population-based stochastic metaheuristics that utilize the principles of natural selection to guide a set of solutions toward the PF. As a result, MOEAs are capable of providing a set of solutions for a multiobjective optimization problem in a single run. Up to now, a large number of MOEAs have been proposed, and they can be roughly divided into the methods based on dominance [7]–[9], the methods based on decomposition [10]–[12], and the methods based on indicator [13]–[15]. These MOEAs have shown competitive performance in solving unconstrained multiobjective optimization problems. However, they can not be directly applied to CMOPs due to the absence of a constraint handling technique (CHT). In fact, CHT can be seen as a selection mechanism to cope with constraints. In various practical application scenarios, CMOPs are more common than the unconstrained multiobjective optimization problems. Hence, a large amount of effort has been devoted to designing CHTs and specific mechanisms, promoting the emergence and development of various constrained MOEAs (CMOEAs) [16]. Meanwhile, to investigate the performance of different CMOEAs, a number of benchmark test suites have also been designed in recent years [17], [18].

Despite numerous studies that have been conducted on constrained multiobjective optimization in the community of evolutionary computation, to the best of our knowledge, there

Manuscript received 1 August 2021; revised 26 November 2021; accepted 22 February 2022. Date of publication 1 March 2022; date of current version 31 March 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62176238, Grant 61806179, Grant 61922072, Grant 61876169, Grant 61976237, and Grant 62106230; in part by the China Postdoctoral Science Foundation under Grant 2020M682347, Grant 2021T140616, and Grant 2021M692920; in part by the Training Program of Young Backbone Teachers in Colleges and Universities in Henan Province, China, under Grant 2020GGJS006; and in part by the Henan Provincial Young Talents Lifting Project, China, under Grant 2021HYTP007. *(Corresponding author: Kunjie Yu.)*

Jing Liang, Xuanxuan Ban, Kunjie Yu, Kangjia Qiao, Caitong Yue, and Ke Chen are with the School of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China (e-mail: liangjing@zzu.edu.cn; 18438622381@163.com; yukunjie@zzu.edu.cn; qiaokangjia@yeah.net; zzuyuecaitong@163.com; chenkezixf@163.com).

Boyang Qu is with the School of Electrical and Information Engineering, Zhongyuan University of Technology, Zhengzhou 450007, China (e-mail: qby1984@hotmail.com).

Kay Chen Tan is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, SAR (e-mail: kctan@polyu.edu.hk).

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.2022.3155533.

Digital Object Identifier 10.1109/TEVC.2022.3155533

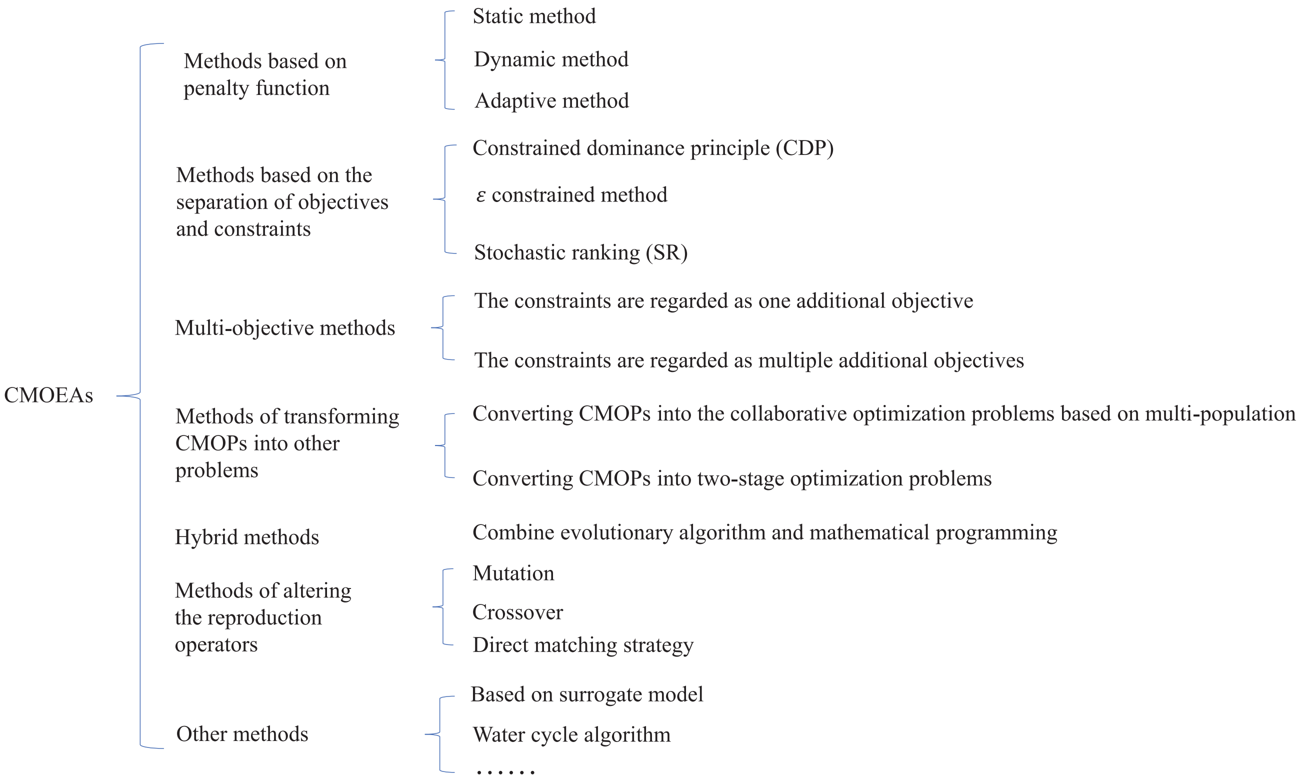

CMOEAs
Methods based on penalty function
Static method
Dynamic method
Adaptive method
Methods based on the separation of objectives and constraints
Constrained dominance principle (CDP)
ε constrained method
Stochastic ranking (SR)
Multi-objective methods
The constraints are regarded as one additional objective
The constraints are regarded as multiple additional objectives
Methods of transforming CMOPs into other problems
Converting CMOPs into the collaborative optimization problems based on multi-population
Converting CMOPs into two-stage optimization problems
Hybrid methods
Combine evolutionary algorithm and mathematical programming
Methods of altering the reproduction operators
Mutation
Crossover
Direct matching strategy
Other methods
Based on surrogate model
Water cycle algorithm
……

Fig. 1. Classification of existing CMOEAs.

has been no comprehensive survey of them presented so far. Hence, this article aims at providing a systematical survey on evolutionary constrained multiobjective optimization regarding the following aspects.

1) On the basis of the categorization shown in Fig. 1, existing CMOEAs are divided into seven classes, and each of them is introduced. Moreover, the advantages and limitations of each class are discussed. It is expected that this introduction of CMOEAs might help researchers to deeply and thoroughly understand the advanced techniques in solving CMOPs. Furthermore, the performances of different CHTs and algorithms are both investigated.
2) The benchmark test problems for evaluating the performance of CMOEAs are introduced and analyzed, and these problems are classified according to their respective characteristics. Hopefully, this might facilitate the performance comparison among different CMOEAs and the development of new CMOEAs.
3) The applications of evolutionary constrained multiobjective optimization in some typical fields are introduced. This not only provides a practical perspective for researchers but can also inspire practitioners to tackle CMOPs in other similar areas by means of CMOEAs.
4) Although the existing CMOEAs have exhibited competitive performance in solving many general CMOPs, as more complicated features are continually emerging in CMOPs, a number of challenges still need to be considered, such as the CMOPs with dynamic, multimodal, computationally expensive, or large-scale characteristics. Therefore, this article gives some topics of promising future research to promote the development of CMOEAs on more types of challenging CMOPs.

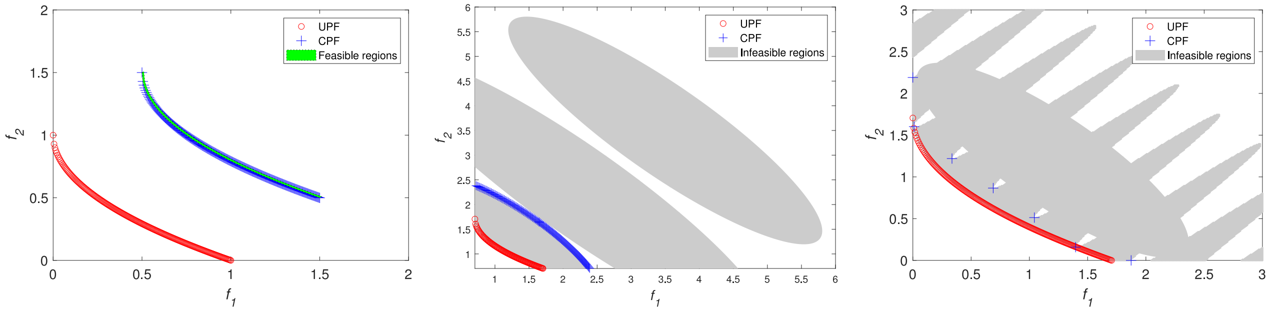
The remainder of this article is structured as follows. The basic concepts on CMOPs are introduced in Section II. The classification and introduction of the existing CMOEAs are described in Section III. Section IV introduces the benchmark test problems in detail, followed by the performance comparison of different CHTs and algorithms in Section V. The specific practical application scenarios are introduced in Section VI. Some directions of potential future research are outlined in Section VII. Finally, the conclusions are drawn in Section VIII.

II. Background

Generally, a CMOP can be described as

$$\begin{aligned} \min\ & \vec{F}(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), \ldots, f_m(\vec{x}))^T \\ \text{s.t.}\ & \begin{cases} g_j(\vec{x}) \leq 0, j = 1, \ldots, l \\ h_j(\vec{x}) = 0, j = l+1, \ldots, k \\ \vec{x} = (x_1, x_2, \ldots, x_D)^T \in \mathbb{S} \end{cases} \end{aligned} \tag{1}$$

where $\mathbb{S}$ is decision space and $\vec{x}$ is the decision vector with D-dimensions in $\mathbb{S}$. $\vec{F}(\vec{x})$ is the objective vector, which contains m objectives that need to be solved. $g_j(\vec{x}) \leq 0$ represents the jth inequality constraint. $h_j(\vec{x}) = 0$ is the $(j-l)$th equality constraint. l and $(k-l)$ denote the number of inequality and equality constraints, respectively. Note that the decision vectors can be continuous or/and discrete and, thus, this article focuses on both continuous and discrete problems.



(a) (b) (c)

Fig. 2. Illustrations for three different types of difficulties [22]: (a) LIR-CMOP2, (b) LIR-CMOP7, and (c) LIR-CMOP11.

For a decision vector $\vec{x}$, its constraint violation degree on the jth constraint is calculated as

$$cv_j(\vec{x}) = \begin{cases} \max\big(0, g_j(\vec{x})\big), & j = 1, \ldots, l \\ \max\big(0, \left|h_j(\vec{x})\right| - \delta\big), & j = l+1, \ldots, k \end{cases} \tag{2}$$

where δ is a positive tolerance value to relax the equality constraints. By adding the constraint violation degree of $\vec{x}$ on each constraint, its total constraint violation degree can be obtained

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

According to the total constraint violation CV, the feasibility of one solution can be identified. If the CV of one solution is 0, it is called feasible solution; otherwise, it is an infeasible solution.

For two feasible solutions $\vec{x}_1$ and $\vec{x}_2$, $\vec{x}_1$ is said to dominate $\vec{x}_2$ if $f_i(\vec{x}_1) \leq f_i(\vec{x}_2)$ for every $i \in \{1, \ldots, m\}$ and $f_j(\vec{x}_1) < f_j(\vec{x}_2)$ for at least one $j \in \{1, \ldots, m\}$. If a solution is not dominated by any solution, it is called the Pareto-optimal solution. The set composed of all Pareto-optimal solutions is PS, and the mapping of PS in objective space is PF.

The presence of constraints brings more difficulties in obtaining the PF of a CMOP than its unconstrained counterpart [i.e., ignoring the constraints in (1)]. The constraints can divide the feasible space into narrow disconnected regions, make a large proportion of the search space infeasible, and/or make the PF of unconstrained MOP partially or completely infeasible. For the sake of distinction, constrained PF (CPF) and unconstrained PF (UPF) are commonly used to denote the PFs for constrained and unconstrained multiobjective optimization problems, respectively [18]–[21]. Obviously, obtaining the UPF is the goal of unconstrained multiobjective optimization, while covering the CPF with good convergence and distribution is desired in CMOPs. To be specific, a good convergence means that the solutions can approach the PF accurately, and a good distribution stands for the solutions featuring good spread and evenness in the objective space.

Specifically, the following main challenges will be encountered when tackling the CMOPs [23].

1) *Difficulty of Feasibility:* As shown in Fig. 2(a), constraints may cause a larger proportion of the search space to become an infeasible region. As a result, a very small feasible region is hard to be located in.
2) *Difficulty of Convergence:* As shown in Fig. 2(b), the infeasible region may become a barrier on the way to the CPF. An algorithm is easy to be trapped in the feasible outside region, thus finding the true CPF is not easy.
3) *Difficulty of Diversity:* As seen in Fig. 2(c), the disconnected feasible region will make the true CPF consists of multiple disconnected segments, thus covering all of them is a challenging task.

In view of the aforementioned challenges, achieving the balance among convergence, diversity, and feasibility is important and critical in solving CMOPs. This research conception has been widely adopted by researchers and a variety of CMOEAs has been tailored in the past years. The details of these CMOEAs will be introduced in the following section.

III. Summary of Existing CMOEAs

In this section, the currently available CMOEAs are classified and introduced. It is worth noting that CHTs are indispensable when combining MOEAs to form CMOEAs. Furthermore, for most of the existing CMOEAs, their efforts usually focus on the CHTs. Thus, as shown in Fig. 1, we use a taxonomy referring to the CHTs instead of the type of MOEAs. The existing CMOEAs can be divided into seven categories: 1) methods based on the penalty function [24], [25]; 2) methods based on the separation of objectives and constraints [26], [27]; 3) multiobjective methods (MOs) [28], [29]; 4) methods of transforming CMOPs into other problems [19], [30]; 5) hybrid methods [31], [32]; 6) methods of altering the reproduction operators [33], [34]; and 7) other methods.

A. Methods Based on Penalty Function

These methods are mainly based on the constraint violation degree to construct the penalty term. By adding the penalty term into objective functions, the constrained optimization problem is transformed into an unconstrained one.

How to set the penalty coefficient is the most critical issue in the penalty function methods, playing a decisive role in terms of efficiency of the algorithm. According to the different setting methods, the penalty method can be divided into the static method, dynamic method, and adaptive method. Obviously, the static method means that the penalty coefficient will not change during the evolution process, but in this way, the preference between objectives and constraints will remain unchanged in the early and late stages of evolution, which is not conducive to the balance between the objectives and constraints. This kind of methods effectively solves simple problems, but it cannot achieve the ideal effect when solving complex problems. Dynamic method is a method where the penalty coefficient changes with the increase of evolutionary generations or other indicators. Through the change of penalty parameters, the weight of objectives and constraints is different in different stages of evolution, so as to achieve the balance between them. However, the difficulty of this method is to design the appropriate change rules, and different change rules need to be set on different problems. The adaptive method preserves the information of the population in the evolution process, and then feeds the obtained information back to the population to adjust the penalty coefficient. Thus, compared with the above two methods, the adaptive strategy can obtain better performance for more complicated problems, since the feedback information can provide guidance of the potential evolution.

Jadaan *et al.* [35] combined a new nondominated ranked genetic algorithm (NRGA) with a nonparameter penalty method to continuously search for the better PS. Zhang *et al.* [36] designed a multihive colony artificial bee colony method to deal with CMOPs, by employing a general coevolution model. In this method, the adaptive penalty function is employed to deal with the constraints, which penalizes infeasible individuals by the number of feasible individuals in the ant colony. By using the proportion of current feasible solutions, Woldesenbet *et al.* [37] employed the distance measure in the objective space to adaptively guide the population evolution. Jan and Zhang [38] employed a threshold to control the degree of punishment for infeasible solutions. Based on this work, Jan *et al.* [39] put forward an adaptive and dynamic version for the threshold, and then embedded it into the MOEA based on decomposition (MOEA/D) framework to deal with CMOPs. Jiao *et al.* [40] suggested a feasible-guiding strategy, which utilizes the degree of constraint violation to modify the objective functions to get the new fitness. Furthermore, in order to achieve the balance between the objectives and constraints, the penalty coefficient is adjusted by the feasibility proportion in the current population. Maldonado and Zapotecas-Martínez [41] proposed a dynamic penalty function method, in which the parameters gradually change with generation number and are embedded into the MOEA/D framework to solve CMOPs. Vaz *et al.* [42] designed a three-step penalty, in which three different penalty coefficients were used at different stages of evolution. Inspired by deep learning, Fan *et al.* [43] proposed a learning-guided parameter setting method, which can adaptively generate penalty parameters. This penalty function combines constraint violation degree, objective function values, and current generation to design an exponential decay model, which is embedded into the push-and-pull search (PPS) [19] framework to solve CMOPs.

The following algorithms have similar mechanisms to the penalty function methods, so they are also classified in this category. Ma *et al.* [44] designed a CHT based on two rankings, named ToR, in which each individual is sorted in two ways: one based on the objective function and the other based on the constrained dominance principle (CDP) [45]. Then, according to the ratio of feasible solutions in the current population, the two rankings are adaptively weighted to obtain a new fitness function. However, its parameters change from 0.5, which means that the preference on constraints is always equal to or greater than the objectives. In addition, because only the proportion of feasible solutions is considered. If the proportion is equal to 0, the weight of objectives and constraints is always equal, which will make the algorithm fail to find feasible solutions on some complex problems. To alleviate these issues, Yu *et al.* [46] proposed a dynamic selection preference-assisted mechanism to transform the preference between objectives and constraints. To be specific, the preference for the objectives is gradually reduced from 1 to 0 according to the designed cosine function curve [47], [48]. Ming *et al.* [20] also carried out two kinds of rankings for each individual, one based on convergence and the other based on diversity. Finally, according to the generation information, these two kinds of rankings are weighted by the rule of the sigmoid function. In order to guide the population in entering the feasible region from different directions in the early stage of evolution, and approach the Pareto-optimal solutions in the later stage, Ma and Wang [25] proposed a shift-based penalty method, named ShiP, in which the infeasible solutions are first shifted according to the distribution of its adjacent feasible solutions, and the degree of migration is adaptively controlled by the proportion of feasible solutions. Then, the shifted infeasible solutions are punished considering their constraint violations. García *et al.* [49] used the penalty function method to select a group of points that are close to the feasible space and have small objective functions. Moreover, the method is embedded in the indicator-based framework to solve the equality CMOP.

B. Methods Based on the Separation of Objectives and Constraints

These methods compare objectives and constraints separately, mainly including the CDP [45], ε constrained method [50], [51], and stochastic ranking (SR) [52].

1) CDP: CDP was proposed by Deb *et al.* [45], which is most commonly used due to its simplicity and ease of implementation. CDP compares paired individuals A and B using the following criteria.

1) When both individuals A and B are feasible solutions, if A Pareto dominates B, A is selected to enter the next generation.
2) When A is the feasible solution and B is the infeasible solution, A is selected.
3) When both A and B are infeasible solutions, the individual with the smallest constraint violation degree is selected.

The operation of CDP is relatively simple, but it prefers feasible solutions, which will cause the population to fall into some local feasible regions when CMOPs have discrete feasible regions or infeasible barriers. In order to remedy this shortcoming of CDP, Jimenez *et al.* [53] employed the niche technology to increase the diversity of the population.

Saha and Ray [54] proposed an equality constraint repair method based on probability points, which combines CDP and clustering methods to repair infeasible solutions. With the aim of using the effective information of infeasible solutions, Wei and Wang [55] applied CDP to the infeasible elitist-based particle swarm algorithm. In the early stage, individuals with good objective values will be retained, and individuals with small objective values and constraint violations will be saved in the later stage. Based on the NSGA-II and MOEA/D framework, Jain and Deb [56] proposed NSGA-III and C-MOEA/D to constrained multiobjective optimization. Fan *et al.* [57] designed a new angle-based CHT, named ACDP, which integrates angle information into the CDP, and further uses the information of infeasible solutions. Later, the ACDP was embedded into the MOEA/D framework to solve CMOPs [58]. In addition, Ning *et al.* [59] proposed an improved hybrid multiobjective optimization algorithm to solve CMOPs, in which each solution is given a constrained nondominated rank according to its constraint violation degree and Pareto rank. Wang and Xu [27] used the ACDP to solve the constrained many-objective optimization problem, in which individuals with good diversity will be first selected to cope with the challenge caused by the large infeasible obstacles.

2) ε Constrained Method: This method was proposed by Takahama and Sakai [50], which uses a parameter ε to relax constraints. ε is gradually reduced, and when the constraint violation degree of an individual is less than ε, it is considered as a feasible solution. Obviously, when ε is reduced to 0, the ε constrained method is the same as CDP. When comparing individuals A and B, the ε constrained method uses the following criteria.

1) If $CV(A) \leq \varepsilon$, $CV(B) \leq \varepsilon$, and A Pareto dominates B, then A is selected.
2) If $CV(A) \leq \varepsilon$, $CV(B) > \varepsilon$, then A is selected.
3) If $CV(A) > \varepsilon$, $CV(B) > \varepsilon$, and $CV(A) < CV(B)$, then A is selected.

Saxena *et al.* [60] integrated the ε constrained method into the NSGA-II framework to control the infeasibility of the population and enhance the convergence of the population. Zapotecas-Martínez *et al.* [61] proposed a multiobjective particle swarm optimization algorithm (MOPSO) based on decomposition, and the ε constrained method is integrated into it to solve CMOPs. Yang *et al.* [62] also integrated the ε constrained method and adaptive operation selection into the multiobjective framework based on decomposition. Becerra *et al.* [63] used the ε constrained method to obtain several points on (or very close to) the PF, and then the method based on the rough set was employed to extend these solutions to cover the whole PF. Martinez and Coello Coello [64] showed a method based on the ε constrained method, in which the relevant information of the neighborhood in MOEA/D is used to obtain the promising solutions in the allowable feasible region. The ε constrained method was improved and embedded into the MOEA/D framework by Fan *et al.* [65], in which the proportion of feasible solutions in the current population is employed to dynamically adjust the ε parameter level. Yang *et al.* [66] designed a multiobjective differential evolution algorithm (MODE-SaE) based on the improved ε constrained method, in which the ε level can be adaptively adjusted based on the maximum and minimum constraint violation values of infeasible individuals. Zapotecas-Martínez and Ponsich [67] established the CMOP as a biobjective problem by considering the constraint violation degree and scalar function as objectives, and integrated the ε constrained method into MOEA/D to solve the constructed problem. Yang *et al.* [68] proposed a dynamic constraint handling mechanism, which divides the search process into two modes: 1) unconstrained search and 2) constrained search. In the constrained search mode, an improved ε constrained method is used to improve population diversity. To ensure the diversity of PF solutions, Wang *et al.* [69] combined the ε constrained method with a niche strategy to solve CMOPs. To prevent the population from falling into the local feasible region or the infeasible region, Zhu *et al.* [70] proposed a technique to help the population escape from the stagnation status, and then an improved ε constrained method is used to search for CPF.

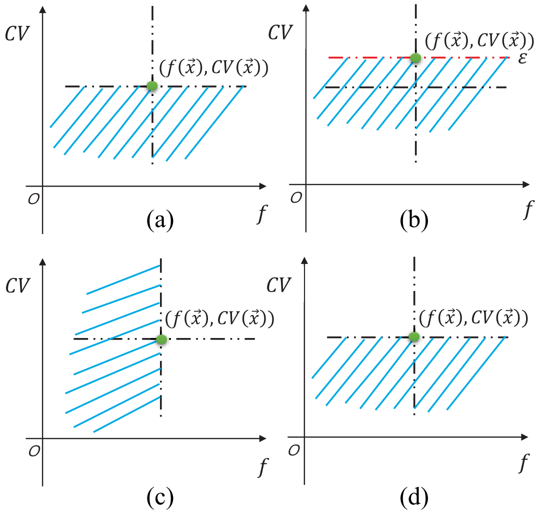

CV
O
f
$(f(\vec{x}), CV(\vec{x}))$
(a)
CV
O
f
$(f(\vec{x}), CV(\vec{x}))$
ε
(b)
CV
O
f
$(f(\vec{x}), CV(\vec{x}))$
(c)
CV
O
f
$(f(\vec{x}), CV(\vec{x}))$
(d)

Fig. 3. Methodologies of CDP, ε constrained method, and SR.

3) SR: In this method, a probability parameter *pf* is introduced. When two individuals are compared, the probability with *pf* only compares their objective function value, and the probability with (1-*pf*) compares their constraint violation degree. This method is able to use the information of the objective function to some extent. Geng *et al.* [71] used SR to balance objectives and constraints when solving CMOPs. Jan and Khanum [72] embedded the modified SR into the MOEA/D framework to solve CMOPs. Ying *et al.* [73] proposed an adaptive stochastic ranking mechanism, which dynamically adjusts the probability parameter according to the current evolution stage and the difference of individuals' violation degree. Liu *et al.* [74] studied CMOEAs based on indicators by combining the indicator-based MOEA with CDP, the ε constrained method, and SR, respectively. Gu *et al.* [75] proposed an evolutionary algorithm based on the surrogate, in which an improved SR strategy based on fitness mechanism and adaptive probability operator was proposed. This strategy considers the convergence and diversity to improve the quality of candidate solutions.

Fig. 3 shows the methodologies of CDP, the ε constrained method, and SR. $\vec{x}$ represents an individual and its objective value and constraint violation degree are $f(\vec{x})$ and $CV(\vec{x})$, respectively. When the other individuals are compared with $\vec{x}$, if they are located in the blue shadow, it means that they are better than $\vec{x}$. Obviously, when CDP is used to select individuals, it will give priority to feasible solutions. In this way, the convergence rate of the population is very fast, but it will also cause the population to easily fall into the local optimum. While both the ε constrained method and SR can make up for the defect of CDP to a certain extent, since they consider the information of infeasible solutions. However, the parameter settings in the ε constrained method and SR are also challenging because the parameter may be problem-dependent.

C. Multiobjective Methods

In the MOs, the constraints are regarded as one additional objective or multiple additional objectives, the CMOP is transformed into an unconstrained counterpart, and then the MOEAs can be employed to solve the converted problem. The objective functions after transformation are as follows:

$$\min \vec{F}(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), \ldots, f_m(\vec{x}), CV(\vec{x}))^T \quad (4)$$

or

$$\min \vec{F}(\vec{x}) = (f_1(\vec{x}), \ldots, f_m(\vec{x}), cv_1(\vec{x}), \ldots, cv_k(\vec{x}))^T. \quad (5)$$

In [76], the constraints were transformed into two new objectives: one is based on the penalty function, the other is equal to the number of violation constraints. Based on the constraint violation measure, Ray *et al.* [77] added a new objective, and designed an infeasibility-driven evolutionary algorithm to promote the population to approach the constraint boundary from the infeasible region. Isaacs *et al.* [78] regarded constraints as a new objective and redefined the constraint minimization problem of the original m objectives as an unconstrained minimization problem with $m + 1$ objectives. Long [79] constructed a novel CHT for solving CMOPs, in which the convergence, diversity, and feasibility of the obtained solutions are used as three new objectives for the multiobjective subproblem. Taking the constraint violation degree as a new objective, Peng *et al.* [29] designed a new CHT based on directed weights to solve CMOPs, in which two types of weights, respectively, distributed in feasible and infeasible regions are employed to guide the search toward the promising regions. Zhou *et al.* [80] proposed a tri-goal evolutionary framework for solving constrained many-objective problems. This framework transforms constraints into feasibility indicators, and then is combined with convergence indicator and diversity indicator to form three new objectives.

D. Methods of Transforming CMOPs Into Other Problems

In order to more efficiently solve CMOPs, many researchers convert CMOPs to other problems, such as converting a CMOP to collaborative optimization problems or two-stage optimization problems. After converting, some promising operators like coevolution can help the population to better explore the search space, discover some new and potential information, and obtain the complete CPF in the end.

1) Converting CMOPs Into the Collaborative Optimization Problems Based on Multipopulation: Chafekar *et al.* [81] transformed a CMOP into multiple single-objective optimization problems, in which several genetic algorithms are implemented, and each of them optimizes an objective and then exchanges objective information. Wang *et al.* [82] proposed a cooperative differential evolution (DE) framework also in the use of m subpopulations, each of which optimizes an objective with constraints, i.e., single constrained optimization. In addition, an archive population is used to preserve the obtained constrained nondominated solutions to approach the CPF. Liu and Wang [83] proposed a CMOEA based on decomposition and temporary archive, which decomposes a CMOP into several subproblems, and each of the subproblems has its own subpopulation and temporary archive. Then, each subproblem is optimized by a coevolution strategy. Later, they proposed a constraint handing scheme based on boundary search and archive [84], in which a CMOP is decomposed into several subproblems, and each subproblem has its own archive. In addition, the boundary search strategy is constructed to improve the efficiency of the algorithm. In order to maintain the distribution of solutions, Yang *et al.* [85] decomposed CMOP into multiple subproblems by partitioning the objective space and used multiple CHTs to solve the optimization problems. Liu *et al.* [86] changed a CMOP into a two-population optimization problem, where one population only optimizes the constraints, and the other population focuses on optimizing the objectives. Meanwhile, the interaction of information and the transfer of knowledge exists between these two populations. Tian *et al.* [87] proposed a coevolutionary framework (CCMO) for solving CMOPs, in which one population is using to solve the original CMOP, that is, to search for CPF, and the other population ignores constraints to find UPF. In addition, these two populations assist each other to solve CMOPs. Li *et al.* [88] designed a dual-archive evolutionary algorithm (C-TAEA), one is a convergence-oriented archive (CA) that aims at pushing the population along the PF3 and the other is a diversity-oriented archive (DA) used to explore the undeveloped areas of CA and maintain population diversity. Wang *et al.* [89] designed a cooperative MOEA by employing two populations: 1) propulsive population and 2) normal population, Concentrating on convergence, the propulsive population considers no constraints in the early stage, while in the late stage, only constraints are considered. The normal population searches the whole CPF and prioritizes the feasibility and the diversity of the population. A bidirectional coevolution algorithm was designed by Liu *et al.*, in which the main population and an archive population are employed simultaneously [90]. Specifically, the main population keeps the feasibility and moves from the feasible side to the CPF, while the archive population uses angle information to maintain population diversity and approximate the CPF from the infeasible side.

2) Converting CMOPs Into Two-Stage Optimization Problems: Santana-Quintero *et al.* [91] combined MOEA with the local search method based on rough set theory to solve CMOPs. In the first stage, MOEA was used to approach the PF. In the second stage, the fuzzy set theory was used to improve the diversity and the convergence of PF solutions. Fan *et al.* [19] proposed a PPS framework. The goal of the push stage is to cross the infeasible region to the UPF. In the pull stage, the improved ε constrained method is employed to search the CPF. Based on the PPS framework, a CMOP was decomposed into a group of simple subproblems [92], with each subproblem corresponding to a subpopulation, and the PPS framework is applied to each subpopulation to solve the related subproblem. Based on the diversity distance measure, Wang *et al.* [93] proposed an objective space modified mechanism, which enables the promising infeasible solutions to be more effective in finding the optimal solutions. In addition, the PPS search framework is used to adjust the position of the PF to prevent the population from falling into the local optimum, and the time complexity can be reduced. Garcia-Garcia *et al.* [94] utilized the characteristics of cellular genetic algorithms (CGAs) and combined PPS technology to solve CMOPs. Tian *et al.* [95] designed a two-stage evolutionary algorithm, named CMOEA-MS, in which one stage can help the population reach the feasible region, and the other stage can make the population spread along the feasible boundary. In addition, based on the status of the population, the algorithm can adaptively switch between these two stages. For CMOPs with constraints in both decision and objective spaces, Liu and Wang [96] suggested a two-stage optimization idea. The first stage is a single objective problem, which aims to search for the promising feasible region; the second stage requires searching for the final PF and reaching the CPF. Xiang *et al.* [21] proposed a two-stage algorithm, named CIC-MOEA/D, the goal of the first stage is to find UPF, so only the objectives are considered; moreover, the constraints are gradually emphasized with the goal of approaching the CPF well in the second stage. In [97], the whole evolutionary process was divided into two stages. The first stage aims to keep the balance between convergence and diversity, and the second stage is devoted to maintaining feasibility and diversity and, thus, covering a well-distributed PF. Yu and Lu [98] proposed a corner point algorithm based on DE by including two stages. The first stage is to find the corner points, and the second stage is to search the real CPF. Zhang *et al.* [30] used the framework of artificial bee colony to divide the optimization process into two stages. In their first stage, fast nondominated sorting is employed in promoting the population to reach the PF. In the second stage, the Tchebycheff method is employed to improve population diversity. Ming *et al.* [99] proposed a two-stage evolutionary algorithm, in which the first stage is used to find UPF and store the feasible solutions obtained, and the second stage concentrates on exploring CPF.

E. Hybrid Methods

Although EA can find promising areas from global perspectives, its subtle search in local regions may be inferior to mathematical programming (MP) [100], [101], especially facing equality constraints that usually reduce the dimensions of search space. Many constrained optimization algorithms relax equality constraints into inequality ones, which fails to ensure the strict feasibility of the obtained solutions. Some researchers have combined EA with MP to deal with CMOPs. Kelner *et al.* [31] proposed a hybrid optimization technology by combining genetic algorithm with local search strategy based on interior-point method. Datta *et al.* [102] proposed a new multiobjective optimization technology, which combines an interior-point method with the nonconvex radial boundary intersection. The radial boundary intersection decomposes the multiobjective optimization problem into several subproblems, which find the solutions of the nearest reference point radially outward along equidistant lines. Morovati and Pourkarimi [103] used the Zoutendijk method to solve CMOPs, where the algorithm considers all objectives and constraints, and a convex quadratic subproblem is proposed to generate a convenient improved feasible direction. Schütze *et al.* [104] proposed a method to calculate the search direction by using neighborhood information. In this method, with regards to a given point in the population, the greedy search direction of given data is calculated in the use of the neighborhood solution of this point. Lara *et al.* [32] proposed the construction method of subspace-based movements in the search process, in which the multiobjective stochastic local search is used to guide individuals. Uribe *et al.* [105] hybridized MOEAs with a special local search mechanism, proposed a descent directions calculation method for the biobjective optimization problem, in which the constraints information is considered in local search. Cuate *et al.* [106] combined MOEA with continuation-like technology to obtain a fast and reliable numerical solver. Hernández *et al.* [107] proposed a method of mixing the hypervolume Newton method and evolutionary strategy to obtain a fast and reliable algorithm to deal with CMOPs. Schütze *et al.* [108] combined the gradient subspace approximation with EA to solve CMOPs, which allows the descent direction to be calculated in a best-fitting manner from the given neighborhood information.

F. Methods of Altering the Reproduction Operators

This type of methods focuses on the design of efficient and specific operators in the process of reproduction. Qian *et al.* [109] combined an adaptive DE with the ε constrained method to solve CMOPs, in which the generation strategies of trial vector and DE parameters are gradually and adaptively adjusted according to the knowledge learned in the previous search process. Yu *et al.* [110] designed a new mutation mechanism to deal with infeasible and feasible solutions, which can produce a well-distributed CPF. Qu and Suganthan [111] proposed a diversity promotion mechanism to prevent the population from falling into the local optimum. Xu *et al.* [33] designed a new DE variant with an infeasible-guiding mutation operator to solve CMOPs, in which the good infeasible solutions are employed to guide the population into the promising region. Wang *et al.* [112] proposed an adaptive DE with Pareto dominance, in which the parameters are adjusted adaptively. Ramesh *et al.* [34] proposed an improved generalized DE for CMOPs by replacing the commonly used binomial/exponential crossover with the simulated binary crossover. Liu *et al.* [113] designed a new strategy based on group-sorting to maintain population diversity, and the crossover operator was designed for each subpopulation to improve the exploration ability. In order to solve CMOPs with a large number of objectives or variable dimensions, He *et al.* [114] emphasized the role of offspring generation when reproducing promising feasible or valuable infeasible offspring solutions. Miyakawa *et al.* [115] proposed a direct matching strategy, which can assign a unique search direction to individuals according to their positions in the objective space. Afterwards, they further proposed a method to control the selection area of promising infeasible solutions by determining the dominant region of the solutions [116].

G. Other Methods

Herein, some methods that do not belong to any of the above categories are discussed. For some problems, it may be more beneficial to allow the population to perform nongreedy or uphill movement, such as the simulated annealing (SA) [117]. Singh *et al.* [118] integrated the surrogate assisted method into SA to solve CMOPs. The optimization method based on the surrogate model can reduce the number of computationally expensive simulations. Singh *et al.* [119] combined the Kriging model with improved multiobjective probability and feasibility probability to solve CMOPs. Datta and Regis [120] proposed a surrogate-assisted evolution strategy to solve CMOPs with expensive black-box inequality constraints. Zhang and Qian [121] employed an artificial immune system model for solving nonlinear constrained multiobjective problems in a time-varying environment. Based on the super rectangle search, Wei and Wang *et al.* [122] proposed a PSO variant, in which the optimal solution of the next time step can be predicted. Considering that the single CHT cannot perform better than other CHTs in every problem, Qu and Suganthan [123] hybridized multiple CHTs to solve CMOPs. Zapotecas-Martínez and Coello Coello [124] proposed an algorithm for continuous box-CMOPs, in which the nonlinear simplex search scheme is used to obtain the PS. Sadollah *et al.* [125] used the water cycle algorithm to solve CMOPs, in which one archive is employed to save the obtained nondominated solutions. Based on the biological immune system, Qian *et al.* [126] designed an immune optimization method for solving CMOPs. Yuan *et al.* [16] introduced an indicator-based CHT for CMOPs to guide the population to uniformly explore the promising areas. Li *et al.* [127] designed a pioneer selection strategy, in which the pioneer individuals give no consideration to constraints, and the proportion of the pioneer individuals is gradually reduced to obtain a well-distributed CPF.

H. Discussion

The relationship between CHTs and MOEAs is discussed here. In general, one CMOEA is composed of CHT and MOEA. To be specific, the CHT is used to tackle the constraints and ensure the feasibility of solutions, while MOEA is taken as the main evolution power to approximate the PF with a set of well-converged and well-distributed feasible solutions. Naturally, designing CMOEAs by integrating MOEAs (i.e., dominance-based [9], decomposition-based [10], and indicator-based algorithms [15]) with CHTs is a practicable and simple way to cope with CMOPs, since all of them have been widely studied individually. Based on the current studies, it is observed that CHTs are often combined with dominance-based and decomposition-based MOEAs [44], [58], since they have better flexibility and generalization than the indicator-based MOEAs. Especially, for decomposition-based MOEAs, they transform a multiobjective optimization problem into multiple single-objective optimization problems. Therefore, many CHTs applied to constrained single-objective optimization can be directly embedded. For the indicator-based MOEAs using quality indicators to define selection mechanisms, they have represented a promising way to deal with many-objective optimization problems due to the increase of selection pressure [15]. Thus, combining indicator-based MOEAs with CHTs is a viable alternative for solving constrained many-objective optimization problems.

In addition, based on the above reviews on existing CMOEAs, the advantages and limitations of each category are summarized in Table I. Although the existing algorithms have exhibited excellent performance in solving CMOPs, some research gaps need to be considered.

1) CMOPs can be solved on the principle of multitask optimization [128]–[130]. Based on the knowledge transfer, evolutionary multitask (EMT) has shown exceedingly promising performance in solving multiple different but related tasks simultaneously. Naturally, the optimization of objectives and the satisfaction of constraints can be seen as two related tasks, thus some knowledge transfer mechanisms in EMT can be readily adopted to solve the CMOPs.
2) In order to assist the solving of CMOPs, some auxiliary tasks can be created by dynamically considering some ones of all constraints. Moreover, for different auxiliary tasks, different CHTs can be employed to exert the complementary effects.
3) The fitness landscape of CMOPs can be analyzed. Based on the characteristics of fitness landscape, more suitable CHTs and evolution operators can be recommended to a specific CMOP.
4) The idea of hyperheuristic algorithm [131], [132] is also beneficial to the utilization of CHTs. The hyperheuristic algorithm has the self-learning ability, which uses the feedback information of heuristic algorithms to improve its performance. Thus, some hyperheuristic strategies are expected to be designed to efficiently use the advantages of different CHTs in the evolution process.

IV. Benchmark Test Problems

In order to comprehensively test the performance of CMOEAs, many benchmark test functions have been proposed, including the SRN test function [133], TNK test

TABLE I
ADVANTAGES AND LIMITATIONS OF VARIOUS METHODS

Categories	Advantages	Limitations
Methods based on penalty function	The methods are simple and easy to implement, flexible, and extensively applicable.	The penalty coefficient is difficult to adjust, and the performance of algorithms will be unsatisfied when dealing with more complicated problems.
Methods based on the separation of objectives and constraints	The methods are simple and easy to implement, and the convergence speed is fast.	It is difficult to balance constraints and objectives, and the related parameters are difficult to set.
Multi-objective methods	The methods can balance constraints and objectives, and make the population have good diversity.	It is difficult to design the additional objective function. In addition, this method may result in a many-objective optimization problem and make the algorithm inefficient.
The methods of transforming CMOPs into other problems	The methods help the population make better use of the information in the search space, and avoid the difficulties encountered in directly solving the original CMOPs.	It is difficult to design an effective conversion technique to ensure the equivalence to the original problems. If the converting mechanism is unreasonable, the algorithm performance will be deteriorated sharply, because of the poor efficiency of computational resource or the deviation to the original problems.
Hybrid methods	The methods promote the convergence of population and enhance the refine search in promising areas. Due to the existence of descent direction, each nonlocal optimal solution can be further improved at each step.	It requires the problem to be differentiable, and needs gradient and even Hessian matrix on each candidate solution.
The methods of altering reproduction operators	The methods are conducive to balancing the exploration and development ability of the population and improving the performance of the algorithm.	The universality of such methods may require further consideration.

function [134], OSY test function [135], CTP test set [136], CF test set [137], NCTP test set [138], C-DTLZ test set [139], CMOP test set[140], LIR-CMOP test set [22], MW test set [18], DOC test set [96], DAS-CMOPs test set [23], CF test set [17], some test problems derived from actual problems [141], and equality constrained test sets (CZDT test set [54] and Eq-DTLZ test set [142]). They make substantial contributions to improving the performance of evolutionary algorithms.

SRN, TNK, and OSY are the first three test problems proposed in the literature, and are often employed for the performance comparison of algorithms. The constraints used in SRN eliminate the partial unconstrained PS, that is, its CPF is part of UPF. The feasible objective space of TNK is the same as the feasible decision variable space. It has a disconnected CPF on the boundary of nonlinear constraints. OSY contains six constraints, and its Pareto-optimal region consists of five segments, each segment located at the intersection of some constraints. The main limitations of these three test problems are as follows: 1) low dimension; 2) most objective functions and constraints are not completely nonlinear, so it is easy to find good solutions; and 3) the complexity of different constraints introduced by them in constrained optimization is not adjustable.

Based on the main defects of the above three problems, the CTP test suite was designed. In the CTP test function set, the difficulty of constraint functions can be adjusted. The constraint functions used mainly provide two types of difficulties: 1) the difficulty of approaching CPF and 2) the difficulty of the whole search space. The CTP test suite contains seven test problems, CTP1–CTP7. The landscape of the region near CPF in test problem CTP1 is complicated, because each constraint is an implicit nonlinear function of decision variables. The presence of constraints makes the search area that is near to the PF infeasible. In addition, by using more constraints, multimodal, and deceptive functions, the complexity of the test problem can be further increased. The test problems CTP2–CTP7 make it difficult to optimize the whole search space. The constraints of CTP2–CTP7 contain six parameters, and the constraint test problems with different difficulties can be obtained by adjusting these six parameters. Although the CTP problem set has a certain improvement compared with the previous three, there are still some defects, i.e., the dimension is not high enough, the feasible region is large, and all problems are restricted to two objectives. In addition, although the number of constraints of CTP1 is adjustable, the shape of its CPF remains unchanged with the increase of constraints. The shapes of CPFs for CTP2–CTP7 are adjustable, but these problems have only one constraint and usually have large feasible regions.

Based on the framework of CTP test suite, the CF test suite is designed. The CF test set has a nonlinear PF in decision space, its CPF has unconnected geometry, and the CF problems also introduce complex variable linkages, so the algorithm finds it difficult to converge when solving the CF problems. However, the CF problems have some defects. For example, the difficulty is not adjustable, the feasible regions are large, and the number of objectives is not extensible.

The NCTP test set is developed to address the defects of CTP problems, and has the following characteristics: 1) the distance function adopts the Rosenbrock function, so the convergence difficulty is increased; 2) high-dimensional decision space is considered; and 3) in order to explicitly reduce the feasibility ratio, an additional constraint is added. In the NCTP test suite, the CPFs of NCTP1–NCTP6 are discontinuous, the CPFs of NCTP7–NCTP12 are composed of continuous and discontinuous parts, and the CPFs of NCTP13–NCTP18 are continuous.

The C-DTLZ test suite is designed on the basis of the unconstrained DTLZ test problems [143]. The C-DTLZ test set mainly contains three kinds of constraints. An infeasible barrier can be provided by introducing the first kind of constraints, while the UPF is still feasible. In addition, several isolated feasible regions along the UPF are defined by the second kind of constraints, and the CPF is part of UPF. The third type of constraints makes the whole UPF infeasible, and CPF is formed by the boundary of the feasible region. Since the C-DTLZ test suite is scalable with multiple difficulties, such as multimodality, irregularity, and bias, it has been widely used to evaluate the performance of CMOEAs. However, the position variables on the PSs of the C-DTLZ test suite lack internal independence, which is also its main limitation.

Fan *et al.* [140] suggested eight constrained optimization problems, CMOP1–CMOP8. In these problems, a total of three types of constraint functions are defined. The first type of constraints can determine the shape of the CPF, the second type can determine the feasible proportion of the search space, and the third type can simultaneously focus on the CPF shape and the proportion of feasible solutions in the entire search space.

The LIR-CMOP test suite contains 14 CMOPs with small feasible regions, and there exists a complex relationship between location and distance variables. The general characteristic of LIR-CMOP test problems is that their true CPFs are blocked by a large number of infeasible regions, making it difficult to be discovered during the evolution process. Its constraint function is composed of a controllable shape function and distance function. Specifically, the shape function is used to make the shapes of CPFs convex and concave, and the distance function can adjust the convergence difficulty.

The MW test suite contains 14 test problems with various characteristics. Some problems have narrower parts in their continuous/discontinuous feasible regions, and the CPFs of some problems contain only a few isolated Pareto-optimal solutions. Hence, the CPFs of MW test problems are relatively difficult to find. In the construction of the MW test set, a global control process and a local adjustment process are introduced, the former can control the size of the feasible region and the latter is able to adjust the complexity of the boundary in the feasible region. The purpose is to generate CPF with different geometric constraints. In addition, in the MW test set, because the distance function can be extended to any number of decision variables, all test problems are scalable in terms of the number of decision variables. Regarding the number of objectives, only test problems MW4, MW8, and MW14 are scalable.

The previous constraint test sets have no consideration to the constraints both in decision and objective spaces. In light of this case, the DOC test set is designed including nine problems. The characteristics of the DOC test set are as follows: 1) it contains objective constraints and decision constraints simultaneously; 2) it contains both inequality constraints and equality constraints; 3) its CPFs have a variety of properties, for example: continuous, disconnected, convex, concave, linear, mixed, degenerate, and multimodal; and 4) the feasible region in the decision space also exhibits various properties, such as nonlinearity, minima, and multimodality. The disadvantage of DOC is that the dimensions are fixed and not adjustable.

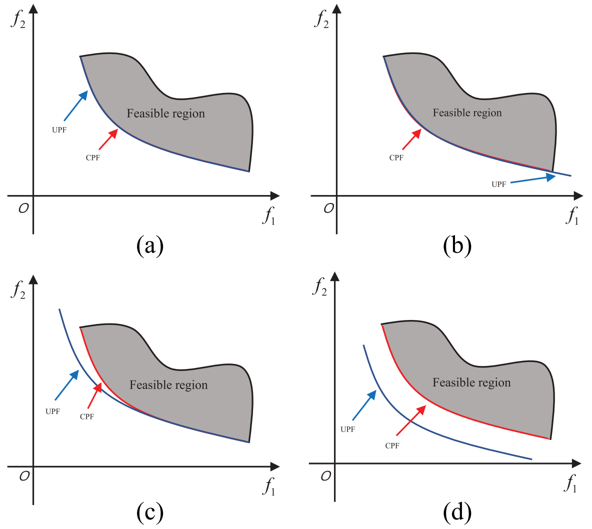


Fig. 4. Classification of CMOPs. (a) Type I. (b) Type II. (c) Type III. (d) Type IV.

DAS-CMOP test suite has adjustable difficulty and expandable objective by employing three primary types of difficulties to characterize the constraints. The level of difficulty in DAS-CMOP is defined by a triplet with each of its parameters specifying each primary difficulty-type level, respectively. In addition, by combining the three primary constraint types with different difficulty triplets, a variety of constraints can be provided for CMOPs.

Zhou *et al.* [17] designed 16 scalable constrained test problems (CF1–CF16) by taking the variable dependencies into account. The constraints introduced by these problems can be divided into convergence difficulty and diversity difficulty. The former refers to the introduction of infeasible obstacles when approaching the CPF, and the position and the distance variables are related to each other. The diversity difficulty constraints will limit the feasible optimal regions, with the result that different shapes of CPFs can be obtained. Among these problems, CF1 and CF2 are the simplest, because they both have regular CPFs, while the CPFs of the remaining problems are irregular, causing difficulties in solving them.

Picard and Schiffmann [141] used the design of electromechanical actuators to develop realistic benchmark CMOPs. Twenty problems are derived with four different constraint levels and up to five objectives. These problems are representative of a variety of mechanical design applications with discontinuities in the objective space.

On the basis of the unconstrained test set ZDT [144], the constrained test suite CZDT was proposed in [54]. It contains five equality constrained test problems, and each test problem has two objectives. In addition, the only feasible solution sets of these problems are the global Pareto-optimal fronts.

Based on the DTLZ test problems [143], eight equality constrained optimization problems were proposed in the Eq-DTLZ test suite [142]. These test problems are adjustable in terms of the number of decision variables, objectives, and equality constraints.

TABLE II
TEST PROBLEMS CLASSIFICATION

Categories	Test problems
Type I	NCTP13-18, C1-DTLZs, DC1-DTLZs, LIR-CMOP5, LIR-CMOP6, LIR-CMOP13, MW2, MW4, MW14
Type II	SRN, C2-DTLZs, DC2-DTLZs, CMOP1, CMOP2, CMOP7 CMOP8, LIR-CMOP10, MW1, MW5, MW6, CTP7, MW8
Type III	CTP1, CTP7, NCTP8-12, LIR-CMOP9, LIR-CMOP11, LIR-CMOP12, MW3, MW7, MW10, MW13
Type IV	TNK, OSY, CTP2, CTP3, CTP4, CTP5, CTP6,CTP8, NCTP1-6, C3-DTLZs, DC3-DTLZs, CMOP3, CMOP4, CMOP5, CMOP6, LIR-CMOP1-4, LIR-CMOP7, LIR-CMOP8, LIR-CMOP14, MW9, MW11, MW12

According to the position of the CPF and UPF of each test function in the objective space, the test problems can be divided into the following four types [18].

1) *Type I:* The UPF is also feasible, as shown in Fig. 4(a), and it is completely overlapped with CPF.
2) *Type II:* The UPF is partially feasible, as shown in Fig. 4(b), CPF is a part of UPF.
3) *Type III:* As shown in Fig. 4(c), the UPF is partly feasible and partially overlapped with the CPF.
4) *Type IV:* As seen in Fig. 4(d), the UPF is located in the infeasible region; thus, UPF and CPF are completely separated.

Obviously, for Type I and Type II problems, the objectives can be given more preference since the CPF is part of the UPF. That is, finding the UPF is extremely beneficial to obtaining CPF. For Type III problems, the balance between objectives and constraints should be considered reasonably. While for the problems that belong to Type IV, it will be better to emphasize the importance of constraints since there is no intersection between UPF and CPF. Hence, different types of problems require different configurations of preference in objectives and constraints.

According to the above classification method, various test sets are classified in Table II. In addition, the characteristics of these test suites are presented in Table III.

V. PERFORMANCE ANALYSIS

In this section, the performance of different CHTs and CMOEAs on different types of problems are investigated. MW and LIR-CMOP test suites are employed because they are widely used and can be classified into four types of problems, as shown in Table II. All experiments are carried out on the PlatEMO platform [145], and the parameter settings for experiments are given in the supplementary material to retain space.

A. Comparison of Different CHTs

Five popular CHTs are chosen for performance comparison. They are self-adaptive penalty method (SP) [37], CDP [45], ε constrained method [109], SR [71], and MO [77]. For a fair comparison, these CHTs are embedded into the widely used NSGA-II framework to form CMOEAs. The detailed experimental results are shown in Table S-1 of the supplementary material. To make the analysis easy, Table IV summarizes the percentages of the best and second-best IGD results of all CHTs on each type of problems.

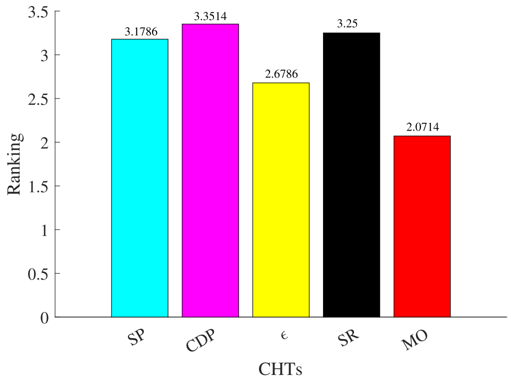


Fig. 5. Rankings of five compared CHTs by Friedman's test.

From Table IV, for Type I CMOPs, SP outperforms other CHTs since it performs either best or second-best on 83.3% of this type of problems. For this kind of problems, the UPF completely overlap with CPF, so the information of objectives and constraints on the UPF is very desirable. Since SP prefers individuals with better objectives and lower constraint violation, it encourages the population to overcome the infeasible obstacle to reach the CPF. During the evolution, the degree of punishment on infeasible solutions is increasing; thus, more efforts can be focused on approaching the CPF. Hence, the SP method has an advantage on solving Type I CMOPs.

For Type II CMOPs, the ε constrained method and SR achieve the better performance than other CHTs. ε constrained method takes some infeasible solutions that satisfy the ε-level as pseudo-feasible solutions. By doing this, these pseudo-feasible solutions can be evolved toward the UPF. The CPF of Type II CMOPs is part of the UPFs, thus searching for the UPF is very beneficial for finding the CPF. As the ε value gradually decreases, the population is capable of gradually approaching the CPF. In addition, SR has a certain probability to evolve the population only considering objectives, thus some infeasible solutions will be utilized to cover the UPF. This property is beneficial for searching for CPF in this type of CMOPs.

For Type III and Type IV problems, MO obviously outperforms other CHTs. This method regards the constraint violation as an additional objective when comparing infeasible solutions, and some infeasible solutions close to the boundary of the feasible region will be preserved. Consequently, the population can simultaneously approach the UPF and the boundary of the feasible region. Since the CPF is part of the boundary of feasible region in Type III and Type IV CMOPs, MO has the ability to approximate the CPF from both the feasible and infeasible sides.

Fig. S-1 of the supplementary material shows the convergence curves of IGD with different CHTs on MW2, MW6, MW7, and LIR-CMOP7. It is worth noting that these four problems belong to Types I–IV, respectively. We can see that different CHTs are suitable to different problems, which are consistent with the above observations. In addition, Fig. 5 shows the rankings of all CHTs on MW and LIR-CMOP test sets obtained by Friedman's test [146]. Overall, the

TABLE III
CHARACTERISTICS OF TEST PROBLEMS. "√" INDICATES THAT THE TEST SET HAS THIS FEATURE; "×" MEANS THE TEST SET DOES NOT HAVE THIS FEATURE; AND "⊗" INDICATES THAT SOME FUNCTIONS IN THE TEST SET HAVE THIS FEATURE

Test sets	Small feasible region	Decision constraints and objective constraint	Equality constraint	Objectives are scalable	The expansibility of the decision variables	Features of PF	Adjustable constraint difficulty	High-dimensional decision space
SRN [133]	√	×	×	×	×	Continuous	×	×
TNK [134]	√	×	×	×	×	Disconnected	×	×
OSY [135]	√	×	×	×	×	Continuous	×	×
CTP [136]	×	×	×	×	×	Disconnected, Discrete	√	×
CF [137]	×	×	×	×	×	Disconnected, Connected	×	×
NCTP [138]	⊗	×	×	×	√	Disconnected, Connected	×	√
C-DTLZ [139]	⊗	×	×	√	√	Disconnected, Connected, Convex	√	√
CMOP [140]	⊗	×	×	×	√	Disconnected, Discrete, Concave, Convex	√	√
LIR-CMOP [22]	⊗	×	⊗	×	×	Convex, Concave	×	√
MW [18]	√	×	×	⊗	√	Disconnected, Linear, Mixed, Discrete, Concave, Convex	×	√
DOC [96]	⊗	√	⊗	×	×	Continuous, Disconnected, Convex, Concave, Linear, Degenerate, and Multimodal	×	×
DAS-CMOP [23]	⊗	⊗	×	√	√	Convex, Concave and Discrete	√	√
CF [17]	√	√	×	√	√	Linear, Concare, Convex, Mixed	×	√
CZDT [54]	√	⊗	√	×	×	Nonlinear	×	√
Eq-DTLZ [142]	√	×	√	√	√	Nonlinear	×	×

TABLE IV
PERCENTAGE OF THE BEST AND SECOND-BEST IGD RESULTS OBTAINED BY DIFFERENT CHTS UNDER NSGA-II FRAMEWORK ON MW AND LIR-CMOP TEST SUITES. THE BEST AND SECOND-BEST PERCENTAGES ARE SHADED WITH A DARK GRAY AND A LIGHT GRAY BACKGROUND, RESPECTIVELY

Type	SP	CDP	ε	SR	MO
Type I	**83.30%**	16.70%	16.70%	33.30%	50.00%
Type II	40.00%	0.00%	**60.00%**	**60.00%**	20.00%
Type III	14.29%	28.57%	57.14%	0.00%	**100.00%**
Type IV	40.00%	10.00%	40.00%	10.00%	**90.00%**

TABLE V
PERCENTAGE OF THE BEST AND SECOND BEST IGD RESULTS OBTAINED BY DIFFERENT CMOEAS ON MW AND LIR-CMOP TEST SUITES. THE BEST AND SECOND-BEST PERCENTAGES ARE SHADED WITH A DARK GRAY AND A LIGHT GRAY BACKGROUNDS, RESPECTIVELY

Type	BiCo	C-TAEA	CCMO	MOEA/D -DAE	PPS -MOEA/D
Type I	**50.00%**	33.33%	**50.00%**	33.33%	33.33%
Type II	20.00%	60.00%	**80.00%**	20.00%	20.00%
Type III	14.29%	28.57%	**71.43%**	57.00%	29.00%
Type IV	60.00%	20.00%	50.00%	0.00%	**70.00%**

performance of MO achieves the first ranking, followed by the ε constrained method.

B. Comparison of Different CMOEAs

In order to analyze the performance of CMOEAs on different types of problems, five representative CMOEAs are selected, which are BiCo [90], C-TAEA [88], CCMO [87], MOEA/D-DAE [70], and PPS-MOEA/D [19]. These CMOEAs have achieved exceedingly competitive performance on CMOPs and are widely used in comparison experiments. The detailed results are shown in Table S-2 of the supplementary material. Table V summarizes percentages of the best and second-best IGD results obtained by different algorithms for each type of problems.

From the results in Table V and Table S-2, BiCo exhibits competitive performance both on Type I and Type IV CMOPs. The reasons are as follows. BiCo includes one main population and one archive population. The main population uses the CDP to approach the CPF from the feasible region side. While in the archive population, the nondominated infeasible solutions are found and an angle-based selection scheme is developed to guide the population toward the CPF also from the infeasible side. In addition, for Type IV problems, the solutions between UPF and CPF are very effective to find CPF. Therefore, BiCo is capable of utilizing these solutions stored in the archive population to obtain good performance.

C-TAEA performs the second-best performance on Type II problems, this superiority mainly derives from MW1, MW5, and MW6. The CPFs of these MW problems are discontinuous, and are part of the UPFs. These problems require more diversity of population and information of objective function to find more CPF fragments. The CA in C-TAEA is used to find CPF, and the DA ignores the constraints, and more objective function information is considered. In addition, DA is also used to find areas that have not been searched by CA, which increases the diversity of population and makes more CPF fragments be achieved.

CCMO performs better than other algorithms on the first three types of problems, especially Type II and Type III problems. This is mainly because the UPFs and CPFs of these problems have a relatively high degree of overlap. CCMO uses a dual-population mechanism, in which population 1 focuses on approaching the CPF by both considering the objectives and constraints, while population 2 aims at searching the UPF by completely ignoring the constraints. Meanwhile, these two populations exchange information in the environment selection. By doing this, population 2 can help population 1 maintain population diversity and pass through the infeasible barrier, which encourages the search for CPF on the first three types of problems. While the superiority of CCMO degrades on Type IV problems with the completely detached UPF and

TABLE VI
SUMMARY OF PRACTICAL APPLICATIONS OF CMOEAs

Scenarios	Problems	Algorithms	The number of objectives	The number of constraints	Continuous or Discrete
Engineering design problems	I-beam optimization problem [147]	ACDP [58]	2	1	Continuous
	welded beam design problem [91]	SADE-εDE [109], MOWCA [125], IMDE [33]	2	3	Continuous
	speed reducer design [148]	SADE-εDE [109], MOWCA [125], [29], IMDE [33]	2	11	Continuous
	disc brake design [148]	SADE-εDE [109], MOWCA [125], [29], IMDE [33]	2	5	Discrete
	steady-state bi-source compressed-air pipeline optimization [149]	MODE-SaE [66]	2	4	Continuous
	car side impact problem [139]	CMOEA-MS [95]	3	10	Continuous
	vibration platform design [150]	CMOEA-MS [95]	2	5	Continuous
	robot gripper optimization [1]	PPS [19]	2	8	Continuous
	software engineering [151]	CIC-MOEA/D [21]	2,3,4	20	Discrete
	four-bar truss design [152]	MOWCA [125]	2	4	Continuous
	spring design [153]	MOWCA [125], IMDE [33]	2	8	Discrete
	servo turntable system [154]	MOPSO [154]	4	3	Discrete
	antenna design optimization [155]	ECMO [156]	3	2	Continuous
	turning process parameters optimization [157]	[157]	3	3	Continuous
	reinforced concrete structures optimization [158]	[158]	2	4	Continuous
Scheduling optimization problems	urban bus scheduling problem [3]	ShiP [25]	2	5	Discrete
	combined economic emission dispatch (CEED) problem [159]	CMOPEO-EED [110, 160, 161]	2	9	Continuous
	the optimal reactive power dispatch (ORPD) problem [162]	[162]	2	5	Discrete
Path planning problems	vehicle routing problems with time windows [163]	CCMO [87]	2	7	Discrete
	robot trajectory planning [164]	MOGA [164]	2	5	Continuous
	constrained multi-objective solid travelling salesman problems (CMOSTSPs) [165]	R-MOGA [165]	2	1	Discrete
	unmanned aerial vehicle planning [166]	NSGA-II-CSP [166]	6	4	Continuous
Resource optimization problems	water resources optimization [167]	CMOEA-MS [95]	5	7	Continuous
	water distribution network optimization [168]	C-TAEA [88]	4	2	Continuous
	water resource management problem [169]	C-AnEA [27]	5	7	Continuous
	groundwater remediation design [170]	AMALGAM [170]	2	4	Continuous
	wind farm layout optimization [171]	CHCP [171]	2	3	Continuous
	thermocline filling layer thermal energy storage [172]	[172]	2	3	Continuous
	test resource allocation optimization [173]	ECHTs [173]	3	3	Continuous

CPF, because the complementary effect will deteriorate as CPF is far away from UPF.

MOEA/D-DAE mainly performs well on some LIR-CMOP problems, because its detect-and-escape strategy once detects the population falls into the local feasible or infeasible regions, the constraints will be relaxed by using the improved ε constrained method, to make the population escape from the local regions, and then gradually move closer to the CPF.

PPS-MOEA/D surpasses other competitors on Type IV problems. The evolution process of PPS-MOEA/D involves push and pull stages. The former prefers crossing the infeasible regions in front of the UPF, so as to prevent the population from falling into the local feasible regions. The latter focuses on driving the solutions obtained by the push stage toward the CPF. By doing this, the individuals will be gradually pulled back from the UPF to CPF.

Fig. S-2 in the supplementary material shows the convergence curves of IGD with different CMOEAs on MW2, MW6, MW7, and LIR-CMOP7. In general, the convergence speed of CCMO and BiCo has a narrow advantage since their second population can help the main population quickly converge to the CPF. In addition, Fig. 6 presents the rankings of CMOEAs on MW and LIR-CMOP test suites obtained by Friedman's test [146]. Obviously, CCMO performs the best, and the performance rankings of other algorithms are comparable.

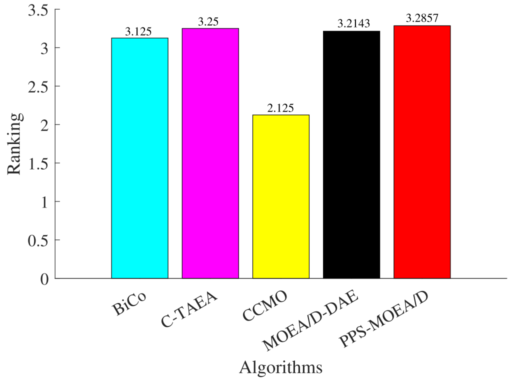


Fig. 6. Rankings of five compared algorithms by Friedman's test.

VI. APPLICATIONS

Many practical problems can be regarded as CMOPs, and the CMOEAs have achieved great success in a variety of fields. In this section, the applications of evolutionary constrained multiobjective optimization in some representative areas are introduced. The detailed information of the real-world problems solved by CMOEAs is summarized in Table VI.

A. Engineering Design Problems

Engineering design is one of the most common problems in real-world, which mainly includes the I-beam optimization problem [147], welded beam design problem [91], speed reducer design [148], disc brake design [148], steady-state bi-source compressed-air pipeline optimization [149], car side impact problem [139], vibration platform design [150], robot gripper optimization [1], software engineering [151], four-bar truss design [152], spring design [153], servo turntable system [154], antenna design optimization [155], turning process parameters optimization [157], and reinforced concrete structures optimization [158].

Fan *et al.* [58] combined ACDP with the MOEA/D framework to solve the I-beam optimization problem. Qian *et al.* [109] used SADE-εDE to solve the welded beam design problem, speed reducer design problem, and disc brake design problem. Peng *et al.* [29] also applied their algorithm to these problems. Xu *et al.* [33] used IMDE to address four common engineering design problems, i.e., the design of welding beam, disk brake design problem, speed reducer design problem, and the spring design problem. Sadollah *et al.* [125] solved six practical engineering problems by using the MOWCA algorithm. Yang *et al.* [66] used the MODE-SaE algorithm to deal with the steady-state bi-source compressed-air pipeline optimization problem. Tian *et al.* [95] used the CMOEA-MS algorithm to solve the car side impact problem, vibration platform design problem, and water resource problem. Fan *et al.* [19] proposed the PPS framework and achieved good results in solving the optimization problem of robot gripper. Maminov and Posypkin [174] constructed the Pareto boundary of the CMOP to solve the robot design optimization problem. Xiang *et al.* [21] applied the CIC-MOEA/D algorithm to the search-based software engineering problem. Singh *et al.* [156] used CMOEA based on the surrogate model (ECMO) to solve antenna design problems. Zhang *et al.* [154] proposed a MOPSO and then applied it to solve the problem of multimodel switching controller based on the switched system model. The turning of composite materials needs to consider the feed speed, spindle speed, and other parameters, which can be regarded as a CMOP. Gadagi and Adake [157] solved this problem based on the genetic algorithm and particle swarm optimization algorithm. For the optimal design problem of reinforced concrete structures, Afshari *et al.* [158] tested a variety of multiobjective optimization algorithms, compared their performance, and analyzed the effects of different algorithms on this problem.

B. Scheduling Optimization Problems

The problems of dispatching optimization have emerged unremittingly in recent years, including the urban bus scheduling problem [3], CEED problem [159], the optimal reactive power dispatch (ORPD) problem [162], and so on.

The urban bus scheduling problem is a common practical problem, Ma and Wang [25] employed ShiP to solve this problem and achieved good results. In the economic emission scheduling problem of the power system, it is necessary to minimize the cost and emission while satisfying various constraints. Chen *et al.* [160] designed a constrained multiobjective population extremal optimization algorithm (CMOPEO-EED) to solve this kind of problems. El-Shorbagy and Mousa [161] proposed an algorithm, which combined the dominance principle, a clustering method, and a constraint repair method to solve the economic emission scheduling problem. Yu *et al.* [110] designed a new DE mutation strategy for solving the CEED problem. In order to solve the problem of reactive power optimal dispatch in power system operation, Mohseni-Bonab *et al.* [162] used the ε constrained method and a fuzzy satisfaction method to select the optimal compromise solutions.

C. Path Planning Problems

The optimization of the path problem can be used in all aspects of scientific and technological life, such as vehicle routing problems with time windows [163], robot trajectory planning [164], constrained multiobjective solid traveling salesman problems [165], and unmanned aerial vehicle (UAV) planning [166].

Vehicle routing problem with time windows is a widely studied combinatorial optimization problem with a complex decision space and strict constraints. Tian *et al.* [87] used the CCMO framework to solve the vehicle routing problems with time windows. In order to solve the trajectory optimization problem of a motor-driven parallel robot, the constrained multiobjective genetic algorithm (MOGA) was used by Chen and Pham [164]. The constrained multiobjective entity traveling salesman problems in rough, fuzzy rough, and random rough environments is a particularly challenging problem, Maity *et al.* [165] proposed a rough MOGA (R-MOGA) to solve this problem. UAV has been developing rapidly in recent years. Ramirez-Atencia and Camacho [166] combined a MOEA with a constraint satisfaction problem model (NSGA-II-CSP) to solve the UAV task planning problem.

D. Resource Optimization Problems

In order to make better use of resources or avoid the waste of resources, the optimization of resources is important. At present, it mainly includes water resources optimization [167], water distribution network optimization [168], water resource management problem [169], groundwater remediation design [170], wind farm layout optimization [171], thermocline filling layer thermal energy storage [172], and test resource allocation optimization [173].

Li *et al.* [88] proposed the C-TAEA algorithm and applied it to optimize the water distribution network. Wang and Xu [27] proposed a method, named C-AnEA, to solve the five-objective water resource management problem. Ouyang *et al.* [170] proposed an adaptive CMOEA based on the multialgorithm (AMALGAM), then employed it to solve the multiobjective optimization design of groundwater remediation in nonaqueous liquid contaminated sites. Sorkhabi *et al.* [171] combined NSGA-II with a penalty function method and constrained programming (CHCP) to solve the constrained multiobjective wind farm layout optimization problems. Marti *et al.* [172] used CMOEA to optimize the efficiency and the material cost of the thermocline packed bed energy storage system with air as heat transfer fluid. Chen and Zhou [175] transformed a complex CMOP into several simple subtasks, and then applied this thought to the constrained multiobjective portfolio problem. Su *et al.* [173] first proposed a multiobjective testing resource allocation problem model with predetermined reliability (ECHTs), which allocated the limited test time to each module, and optimized the system reliability, test cost, and test time effectively. Yu and Lu [98] suggested a corner point-based algorithm to cope with the problem of resource arrangement in emergency management that is built as a CMOP.

VII. Future Directions

Despite the numerous CMOPs that have been tackled by evolutionary algorithms, this is a research area with several potential future research topics. Some of them are discussed as follows.

1) *Handling Dynamical CMOPs:* In various real-world scenarios, the dynamic environment is involved in many CMOPs. That is, the objectives and/or constraints of CMOPs may change over time, such as the optimal control problems [176], the optimization of fluid catalytic cracking operation [177], and the operational indices optimization of beneficiation process [178]. These problems are called dynamic CMOPs (DCMOPs), and they are more challenging due to the simultaneous presence of objectives, constraints, and dynamism. For a DCMOP, its CPF will be changed due to the dynamic environment. As a result, the CPFs in different environments may differ in terms of position, shape, and geometry. The goal of solving DCMOPs is to rapidly obtain all CPFs in different environments. Hence, the main challenge in solving such problems is that the optimization algorithm needs to track changes in the environment and adaptively modify the search strategy when changes are detected. However, only a few efforts have been focused on developing the evolutionary algorithms to solve DCMOPs [179], [180]. Therefore, more efficient algorithms are expected to be developed for DCMOPs.

2) *Handling Multimodal CMOPs:* Multimodal optimization is a kind of problems, in which multiple Pareto-optimal solutions correspond to the same objective, and all Pareto-optimal solutions are desired [181], [182]. Multimodal CMOPs exist in many practical applications, such as the speed reducer model [183] and the pressure vessel model [184]. Solving this type of problems requires a strong diversity to find multiple Pareto-optimal solutions. Despite many MOEAs being customized to solve multimodal multiobjective optimization problems [185], it is difficult to address multimodal CMOPs, since constraints complicate the decision space. To efficiently solve multimodal CMOPs, some attempts should be focused on the diversity preservation mechanisms in the decision space to obtain multiple feasible Pareto-optimal solutions simultaneously.

3) *Handling Computationally Expensive CMOPs:* Some practical optimization problems require large computing resources and an amount of time for function evaluation [186], [187], such as computational electromagnetics [188] and computational fluid dynamics [189]. They can consume several minutes to many hours for conducting one evaluation of a candidate solution. Such problems are computationally expensive. In order to solve these problems, surrogates, the computationally cheap models, are used to reduce the function evaluations. However, the existing surrogate model-based algorithms have not been well expanded to the computationally expensive CMOPs, which are often faced in real-world applications [120]. Therefore, more effective surrogate model-based methods should be developed and combined with the existing CMOEAs to solve computationally expensive CMOPs.

4) *Handling Large-Scale CMOPs:* Despite the existing CMOEAs achieving promising performance in solving general CMOPs with small scale decision variables and objectives, as the decision variables or objectives are scaled up, their effectiveness may dramatically deteriorate since the curse of dimensionality [190]. Large-scale CMOPs are also common, such as the multiobjective vehicle routing problem [191], time-varying ratio error estimation problem [192], and configuring software optimization [193]. To address these problems, the mechanisms [194] customized for large-scale problems can be embedded into the existing CMOEAs framework.

5) *Handling Robust CMOPs:* The objectives in many practical problems are easily affected by uncertain environmental factors [195], [196]. This type of problem is called the robust optimization problem [197], [198]. In robust CMOPs, both constraints and objectives may be affected by external factors. When solving such problems, it is necessary to obtain the solutions that can quickly return to the initial position after being disturbed by influencing factors. This kind of robust solutions is more acceptable than the Pareto-optimal solutions. However, the robust mechanism is more difficult to design due to the complicated uncertain environment. Therefore, it is expected to design reasonable robustness mechanisms and combine them with constraint processing technology to solve robust CMOPs.

6) *Problem-Type-Guided Evolution:* As mentioned in Section IV, there are different types in the relationship between UPF and CPF. If the relationship can be learned and then used to guide the subsequent evolution operators, the searchability will be enhanced efficiently. To be specific, for different types of problems, UPF has different guiding effects on CPF. For example, when the problem belongs to Type I or Type II, the information on the UPF can be directly used by the CPF. In this case, the UPF information can help the population find CPF through knowledge transfer [199]. Hence, developing the strategy to obtain the type of problems will be beneficial to the solving of CMOPs.

7) *Algorithm Recommendation System:* According to the principle of no free lunch [200], no algorithm can achieve good results on all types of problems. It is expected that each algorithm is allocated to its own suited problems. Hence, various CMOEAs can be combined with machine learning [201], deep learning [202],

and other techniques to extract the characteristics of different problems, and then recommend the matched algorithms according to the characteristics of the CMOPs [203].

8) *Landscape-Based Evolution Operators:* Different fitness landscapes and constraint landscapes [204], [205] exist in different problems. Landscapes have different characteristics and can reflect the internal properties of problems, so the required evolution operators are also different. Detecting the fitness landscape of a problem, the algorithm applied to it can be selected according to the landscape, so the landscape information can be considered and used to guide the design of evolution operators in CMOEAs [206].

VIII. Conclusion

This article reviewed the research of evolutionary constrained multiobjective optimization, covering the basic concepts, existing algorithms, benchmark test functions, applications, and future research directions. First, the relevant theoretical background and some concepts have been introduced. Second, this article has reviewed the existing CMOEAs and divided them into seven categories according to their CHTs and internal mechanisms. Third, some existing test function suites have been introduced and classified in detail, and the advantages and disadvantages of each test suite are summarized. Fourth, some popular applications of evolutionary constrained multiobjective optimization have been introduced and summarized. Fifth, the performance of different CHTs and CMOEAs on different types of problems was investigated. Finally, some future research directions have been discussed. It is hoped that the work presented in this article will help researchers to become familiar with evolutionary constrained multiobjective optimization, and promote the future development of this research direction.

References

[1] R. Saravanan, S. Ramabalan, N. G. R. Ebenezer, and C. Dharmaraja, "Evolutionary multi criteria design optimization of robot grippers," *Appl. Soft Comput.*, vol. 9, no. 1, pp. 159–172, 2009.

[2] M. Basu, "Economic environmental dispatch using multi-objective differential evolution," *Appl. Soft Comput.*, vol. 11, no. 2, pp. 2845–2853, 2011.

[3] X. Zuo, C. Chen, W. Tan, and M. Zhou, "Vehicle scheduling of an urban bus line via an improved multiobjective genetic algorithm," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 1030–1041, Apr. 2015.

[4] Y. Cui, Z. Geng, Q. Zhu, and Y. Han, "Multi-objective optimization methods and application in energy saving," *Energy*, vol. 125, pp. 681–704, Apr. 2017.

[5] K. Shang, H. Ishibuchi, L. He, and L. M. Pang, "A survey on the hypervolume indicator in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 25, no. 1, pp. 1–20, Feb. 2021.

[6] X. Ma, Y. Yu, X. Li, Y. Qi, and Z. Zhu, "A survey of weight vector adjustment methods for decomposition-based multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 24, no. 4, pp. 634–649, Aug. 2020.

[7] A. Asrari, S. Lotfifard, and M. S. Payam, "Pareto dominance-based multiobjective optimization method for distribution network reconfiguration," *IEEE Trans. Smart Grid*, vol. 7, no. 3, pp. 1401–1410, May 2016.

[8] H. Zhang, D. Yue, X. Xie, S. Hu, and S. Weng, "Pareto-dominance based adaptive multi-objective optimization for hydrothermal coordinated scheduling with environmental emission," *Appl. Soft Comput.*, vol. 69, pp. 270–287, Aug. 2018.

[9] S. Dutta and K. N. Das, "A survey on Pareto-based EAs to solve multi-objective optimization problems," in *Soft Computing for Problem Solving*. Singapore: Springer, 2019, pp. 807–820.

[10] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, "A survey of multiobjective evolutionary algorithms based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 440–462, Jun. 2017.

[11] M. Behtash and M. J. Alexander-Ramos, "A decomposition-based optimization algorithm for combined plant and control design of interconnected dynamic systems," *J. Mech. Des.*, vol. 142, no. 6, 2020, Art. no. 061703.

[12] S. S. Meghwani and M. Thakur, "Adaptively weighted decomposition based multi-objective evolutionary algorithm," *Appl. Intell.*, vol. 51, no. 6, pp. 3801–3823, 2021.

[13] Y. Tian, R. Cheng, X. Zhang, F. Cheng, and Y. Jin, "An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 609–622, Aug. 2018.

[14] Y. Sun, G. G. Yen, and Z. Yi, "IGD indicator-based evolutionary algorithm for many-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 173–187, Apr. 2019.

[15] J. G. Falcón-Cardona and C. A. Coello Coello, "Indicator-based multiobjective evolutionary algorithms: A comprehensive survey," *ACM Comput. Surv.*, vol. 53, no. 2, pp. 1–35, 2020.

[16] 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.*, early access, Jun. 14, 2021, doi: 10.1109/TEVC.2021.3089155.

[17] Y. Zhou, Y. Xiang, and X. He, "Constrained multiobjective optimization: Test problem construction and performance evaluations," *IEEE Trans. Evol. Comput.*, vol. 25, no. 1, pp. 172–186, Feb. 2021, doi: 10.1109/TEVC.2020.3011829.

[18] 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.

[19] 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.

[20] M. Ming, A. Trivedi, R. Wang, D. Srinivasan, and T. Zhang, "A dual-population-based evolutionary algorithm for constrained multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 25, no. 4, pp. 739–753, Aug. 2021, doi: 10.1109/TEVC.2021.3066301.

[21] Y. Xiang, X. Yang, H. Huang, and J. Wang, "Balancing constraints and objectives by considering problem types in constrained multiobjective optimization," *IEEE Trans. Cybern.*, early access, Jul. 8, 2021, doi: 10.1109/TCYB.2021.3089633.

[22] 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, 2019.

[23] Z. Fan *et al.*, "Difficulty adjustable and scalable constrained multiobjective test problem toolkit," *Evol. Comput.*, vol. 28, no. 3, pp. 339–378, Sep. 2020.

[24] Z. Xia, Y. Liu, J. Lu, J. Cao, and L. Rutkowski, "Penalty method for constrained distributed quaternion-variable optimization," *IEEE Trans. Cybern.*, vol. 51, no. 11, pp. 5631–5636, Nov. 2021, doi: 10.1109/TCYB.2020.3031687.

[25] 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.

[26] B.-C. Wang, H.-X. Li, J.-P. Li, and Y. Wang, "Composite differential evolution for constrained evolutionary optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 7, pp. 1482–1495, Jul. 2019.

[27] C. Wang and R. Xu, "An angle based evolutionary algorithm with infeasibility information for constrained many-objective optimization," *Appl. Soft Comput.*, vol. 86, Jan. 2020, Art. no. 105911.

[28] C. Segura, C. A. Coello Coello, G. Miranda, and C. León, "Using multi-objective evolutionary algorithms for single-objective constrained and unconstrained optimization," *Ann. Oper. Res.*, vol. 240, no. 1, pp. 217–250, 2016.

[29] C. Peng, H.-L. Liu, and F. Gu, "An evolutionary algorithm with directed weights for constrained multi-objective optimization," *Appl. Soft Comput.*, vol. 60, pp. 613–622, Nov. 2017.

[30] H. Zhang, K. Tao, L. Ma, and Y. Yong, "Handling constrained multiobjective optimization with objective space mapping to decision space based on extreme learning machine," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Glasgow, U.K., 2020, pp. 1–7.

[31] V. Kelner, F. Capitanescu, O. Léonard, and L. Wehenkel, "A hybrid optimization technique coupling an evolutionary and a local search algorithm," *J. Comput. Appl. Math.*, vol. 215, no. 2, pp. 448–456, 2008.

[32] A. Lara, L. Uribe, S. Alvarado, V. A. Sosa, H. Wang, and O. Schütze, "On the choice of neighborhood sampling to build effective search operators for constrained mops," *Memetic Comput.*, vol. 11, no. 2, pp. 155–173, 2019.

[33] B. Xu, W. Duan, H. Zhang, and Z. Li, "Differential evolution with infeasible-guiding mutation operators for constrained multi-objective optimization," *Appl. Intell.*, vol. 50, no. 12, pp. 4459–4481, 2020.

[34] S. Ramesh, S. Kannan, and S. Baskar, "An improved generalized differential evolution algorithm for multi-objective reactive power dispatch," *Eng. Optim.*, vol. 44, no. 4, pp. 391–405, 2012.

[35] O. A. Jadaan, C. R. Rao, and L. Rajamani, "Solving constrained multi-objective optimization problems using non-dominated ranked genetic algorithm," in *Proc. 3rd Asia Int. Conf. Model. Simul.*, Bundang, Indonesia, 2009, pp. 113–118.

[36] H. Zhang, Y. Zhu, and X. Yan, "Multi-hive artificial bee colony algorithm for constrained multi-objective optimization," in *Proc. IEEE Congr. Evol. Comput.*, Brisbane, QLD, Australia, 2012, pp. 1–8.

[37] 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.

[38] M. A. Jan and Q. Zhang, "MOEA/D for constrained multiobjective optimization: Some preliminary experimental results," in *Proc. U.K. Workshop Comput. Intell. (UKCI)*, Colchester, U.K., 2010, pp. 1–6.

[39] M. A. Jan, N. Tairan, and R. A. Khanum, "Threshold based dynamic and adaptive penalty functions for constrained multiobjective optimization," in *Proc. 1st Int. Conf. Artif. Intell. Model. Simul.*, Kota Kinabalu, Malaysia, 2013, pp. 49–54.

[40] 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.

[41] H. M. Maldonado and S. Zapotecas-Martínez, "A dynamic penalty function within MOEA/D for constrained multi-objective optimization problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Kraków, Poland, 2021, pp. 1470–1477.

[42] F. Vaz, Y. Lavinas, C. Aranha, and M. Ladeira, "Exploring constraint handling techniques in real-world problems on MOEA/D with limited budget of evaluations," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.*, 2021, pp. 555–566.

[43] Z. Fan *et al.*, "A learning guided parameter setting for constrained multi-objective optimization," in *Proc. 1st Int. Conf. Ind. Artif. Intell. (IAI)*, Shenyang, China, 2019, pp. 1–6.

[44] 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.

[45] 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.

[46] 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.*, early access, Mar. 9, 2021, doi: 10.1109/TSMC.2021.3061698.

[47] W. Gong and Z. Cai, "Differential evolution with ranking-based mutation operators," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 2066–2081, Dec. 2013.

[48] H. Ma, "An analysis of the equilibrium of migration models for biogeography-based optimization," *Inf. Sci.*, vol. 180, no. 18, pp. 3444–3464, 2010.

[49] J. L. L. García, R. Monroy, V. A. S. Hernández, and C. A. Coello Coello, "COARSE-EMOA: An indicator-based evolutionary algorithm for solving equality constrained multi-objective optimization problems," *Swarm Evol. Comput.*, vol. 67, Dec. 2021, Art. no. 100983.

[50] T. Takahama and S. Sakai, "Constrained optimization by the ϵ constrained differential evolution with gradient-based mutation and feasible elites," in *Proc. IEEE Int. Conf. Evol. Comput.*, Vancouver, BC, Canada, 2006, pp. 1–8.

[51] K. Qiao, J. Liang, K. Yu, M. Yuan, B. Qu, and C. Yue, "Self-adaptive resources allocation-based differential evolution for constrained evolutionary optimization," *Knowl. Based Syst.*, vol. 235, Jan. 2022, Art. no. 107653.

[52] 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.

[53] F. Jimenez, A. F. Gomez-Skarmeta, G. Sanchez, and K. Deb, "An evolutionary algorithm for constrained multi-objective optimization," in *Proc. Congr. Evol. Comput. (CEC)*, vol. 2. Honolulu, HI, USA, 2002, pp. 1133–1138.

[54] A. Saha and T. Ray, "Equality constrained multi-objective optimization," in *Proc. IEEE Congr. Evol. Comput.*, Brisbane, QLD, Australia, 2012, pp. 1–7.

[55] J. Wei and Y. Wang, "An infeasible elitist based particle swarm optimization for constrained multiobjective optimization and its convergence," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 24, no. 3, pp. 381–400, 2010.

[56] 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.

[57] Z. Fan, W. Li, X. Cai, K. Hu, H. Lin, and H. Li, "Angle-based constrained dominance principle in MOEA/D for constrained multi-objective optimization problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Vancouver, BC, Canada, 2016, pp. 460–467.

[58] 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.

[59] W. Ning, B. Guo, Y. Yan, X. Wu, J. Wu, and D. Zhao, "Constrained multi-objective optimization using constrained non-dominated sorting combined with an improved hybrid multi-objective evolutionary algorithm," *Eng. Optim.*, vol. 49, no. 10, pp. 1645–1664, 2017.

[60] D. K. Saxena, T. Ray, K. Deb, and A. Tiwari, "Constrained many-objective optimization: A way forward," in *Proc. IEEE Congr. Evol. Comput.*, Trondheim, Norway, 2009, pp. 545–552.

[61] S. Zapotecas-Martínez, A. A. Montano, and C. A. Coello Coello, "Constrained multi-objective aerodynamic shape optimization via swarm intelligence," in *Proc. Annu. Conf. Genet. Evol. Comput.*, 2014, pp. 81–88.

[62] Z. Yang, X. Cai, and Z. Fan, "Epsilon constrained method for constrained multiobjective optimization problems: some preliminary results," in *Proc. Companion Publ. Annu. Conf. Genet. Evol. Comput.*, 2014, pp. 1181–1186.

[63] R. L. Becerra, C. A. Coello Coello, A. G. Hernández-Díaz, R. Caballero, and J. Molina, "Alternative techniques to solve hard multi-objective optimization problems," in *Proc. 9th Annu. Conf. Genet. Evol. Comput.*, 2007, pp. 754–757.

[64] S. Zapotecas-Martínez and C. A. Coello Coello, "A multi-objective evolutionary algorithm based on decomposition for constrained multi-objective optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Beijing, China, 2014, pp. 429–436.

[65] Z. Fan *et al.*, "An improved epsilon constraint handling method embedded in MOEA/D for constrained multi-objective optimization problems," in *Proc. IEEE Symp. Series Comput. Intell. (SSCI)*, Athens, Greece, 2016, pp. 1–8.

[66] 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.

[67] S. Zapotecas-Martínez and A. Ponsich, "Constraint handling within MOEA/D through an additional scalarizing function," in *Proc. Genet. Evol. Comput. Conf.*, 2020, pp. 595–602.

[68] Y. Yang, J. Liu, and S. Tan, "A constrained multi-objective evolutionary algorithm based on decomposition and dynamic constraint-handling mechanism," *Appl. Soft Comput.*, vol. 89, Apr. 2020, Art. no. 106104.

[69] Z. Wang, J. Wei, and Y. Zhang, "A multi-constraint handling technique-based niching evolutionary algorithm for constrained multi-objective optimization problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Glasgow, U.K., 2020, pp. 1–6.

[70] 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.

[71] H. Geng, M. Zhang, L. Huang, and X. Wang, "Infeasible elitists and stochastic ranking selection in constrained evolutionary multi-objective optimization," in *Proc. Asia–Pacific Conf. Simul. Evol. Learn.*, 2006, pp. 336–344.

[72] 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, 2013.

[73] W.-Q. Ying, W.-P. He, Y.-X. Huang, D.-T. Li, and Y. Wu, "An adaptive stochastic ranking mechanism in MOEA/D for constrained multi-objective optimization," in *Proc. Int. Conf. Inf. Syst. Artif. Intell. (ISAI)*, Hong Kong, 2016, pp. 514–518.

[74] 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, doi: 10.1109/TSMC.2019.2954491.

[75] Q. Gu, Q. Wang, N. N. Xiong, S. Jiang, and L. Chen, "Surrogate-assisted evolutionary algorithm for expensive constrained multi-objective discrete optimization problems," *Complex Intell. Syst.*, vol. 302, pp. 1–20, Jan. 2021.

[76] D. A. G. Vieira, R. L. S. Adriano, J. A. Vasconcelos, and L. Krahenbuhl, "Treating constraints as objectives in multiobjective optimization problems using niched Pareto genetic algorithm," *IEEE Trans. Magn.*, vol. 40, no. 2, pp. 1188–1191, Mar. 2004.

[77] T. Ray, H. K. Singh, A. Isaacs, and W. Smith, "Infeasibility driven evolutionary algorithm for constrained optimization," in *Constraint-Handling in Evolutionary Optimization*. Heidelberg, Germany: Springer, 2009, pp. 145–165.

[78] A. Isaacs, T. Ray, and W. Smith, "Blessings of maintaining infeasible solutions for constrained multi-objective optimization problems," in *Proc. IEEE Congr. Evol. Comput. World Congr. Comput. Intell.*, Hong Kong, 2008, pp. 2780–2787.

[79] Q. Long, "A constraint handling technique for constrained multi-objective genetic algorithm," *Swarm Evol. Comput.*, vol. 15, pp. 66–79, Apr. 2014.

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

[81] D. Chafekar, J. Xuan, and K. Rasheed, "Constrained multi-objective optimization using steady state genetic algorithms," in *Proc. Genet. Evol. Comput. Conf.*, 2003, pp. 813–824.

[82] J. Wang, G. Liang, and J. Zhang, "Cooperative differential evolution framework for constrained multiobjective optimization," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2060–2072, Jun. 2019.

[83] H.-L. Liu and D. Wang, "A constrained multiobjective evolutionary algorithm based decomposition and temporary register," in *Proc. IEEE Congr. Evol. Comput.*, Cancun, Mexico, 2013, pp. 3058–3063.

[84] H.-L. Liu, C. Peng, F. Gu, and J. Wen, "A constrained multi-objective evolutionary algorithm based on boundary search and archive," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 30, no. 1, 2016, Art. no. 1659002.

[85] Y. Yang, J. Liu, and S. Tan, "A partition-based constrained multi-objective evolutionary algorithm," *Swarm Evol. Comput.*, vol. 66, Oct. 2021, Art. no. 100940. doi: 10.1016/j.swevo.2021.100940.

[86] B. Liu, H. Ma, X. Zhang, and Y. Zhou, "A memetic co-evolutionary differential evolution algorithm for constrained optimization," in *Proc. IEEE Congr. Evol. Comput.*, Singapore, 2007, pp. 2996–3002.

[87] 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, doi: 10.1109/TEVC.2020.3004012.

[88] 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.

[89] 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.*, early access, Apr. 9, 2021, doi: 10.1109/TSMC.2021.3069986.

[90] 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.

[91] L. V. Santana-Quintero, A. G. Hernández-Díaz, J. Molina, C. A. Coello Coello, and R. Caballero, "DEMORS: A hybrid multi-objective optimization algorithm using differential evolution and rough set theory for constrained problems," *Comput. Oper. Res.*, vol. 37, no. 3, pp. 470–480, 2010.

[92] 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.

[93] H. Wang, T. Cai, K. Li, and W. Pedrycz, "Constraint handling technique based on lebesgue measure for constrained multiobjective particle swarm optimization algorithm," *Knowl. Based Syst.*, vol. 221, Sep. 2021, Art. no. 107131, doi: 10.1016/j.knosys.2021.107131.

[94] C. Garcia-Garcia, M.-G. Martínez-Peñaloza, and A. Morales-Reyes, "CMOGA/D: A novel cellular ga based on decomposition to tackle constrained multiobjetive problems," in *Proc. Genet. Evol. Comput. Conf. Compan.*, 2020, pp. 1721–1729.

[95] 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.*, early access, Mar. 17, 2021, doi: 10.1109/TCYB.2020.3021138.

[96] Z.-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.

[97] K. Yu, J. Liang, B. Qu, and C. Yue, "Purpose-directed two-phase multiobjective differential evolution for constrained multiobjective optimization," *Swarm Evol. Comput.*, vol. 60, Feb. 2021, Art. no. 100799.

[98] X. Yu and Y. Lu, "A corner point-based algorithm to solve constrained multi-objective optimization problems," *Appl. Intell.*, vol. 48, no. 9, pp. 3019–3037, 2018.

[99] F. Ming, W. Gong, H. Zhen, S. Li, L. Wang, and Z. Liao, "A simple two-stage evolutionary algorithm for constrained multi-objective optimization," *Knowl. Based Syst.*, vol. 228, Sep. 2021, Art. no. 107263.

[100] F. Caraffini, F. Neri, G. Iacca, and A. Mol, "Parallel memetic structures," *Inf. Sci.*, vol. 227, pp. 60–82, Apr. 2013.

[101] F. Caraffini, F. Neri, and L. Picinali, "An analysis on separability for memetic computing automatic design," *Inf. Sci.*, vol. 265, pp. 1–22, May 2014.

[102] S. Datta, A. Ghosh, K. Sanyal, and S. Das, "A radial boundary intersection aided interior point method for multi-objective optimization," *Inf. Sci.*, vol. 377, pp. 1–16, Jan. 2017.

[103] V. Morovati and L. Pourkarimi, "Extension of zoutendijk method for solving constrained multiobjective optimization problems," *Eur. J. Oper. Res.*, vol. 273, no. 1, pp. 44–57, 2019.

[104] O. Schütze, S. Alvarado, C. Segura, and R. Landa, "Gradient subspace approximation: A direct search method for memetic computing," *Soft Comput.*, vol. 21, no. 21, pp. 6331–6350, 2017.

[105] L. Uribe, A. Lara, and O. Schütze, "On the efficient computation and use of multi-objective descent directions within constrained MOEAs," *Swarm Evol. Comput.*, vol. 52, Feb. 2020, Art. no. 100617.

[106] O. Cuate, A. Ponsich, L. Uribe, S. Zapotecas-Martínez, A. Lara, and O. Schütze, "A new hybrid evolutionary algorithm for the treatment of equality constrained mops," *Mathematics*, vol. 8, no. 1, p. 7, 2020.

[107] V. A. S. Hernández, O. Schütze, H. Wang, A. Deutz, and M. Emmerich, "The set-based hypervolume newton method for bi-objective optimization," *IEEE Trans. Cybern.*, vol. 50, no. 5, pp. 2186–2196, May 2020.

[108] O. Schütze, L. Uribe, and A. Lara, "The gradient subspace approximation and its application to bi-objective optimization problems," in *Proc. Workshop Dyn. Optim. Comput. Honor 60th Birthday Michael Dellnitz*, 2020, pp. 355–390.

[109] F. Qian, B. Xu, R. Qi, and H. Tianfield, "Self-adaptive differential evolution algorithm with α-constrained-domination principle for constrained multi-objective optimization," *Soft Comput.*, vol. 16, no. 8, pp. 1353–1372, 2012.

[110] 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, pp. 452–466, Jun. 2018.

[111] B. Qu and P. N. Suganthan, "Constrained multi-objective optimization algorithm with diversity enhanced differential evolution," in *Proc. IEEE Congr. Evol. Comput.*, 2010, pp. 1–5.

[112] Y.-N. Wang, L.-H. Wu, and X.-F. Yuan, "Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure," *Soft Comput.*, vol. 14, no. 3, pp. 193–209, 2010.

[113] Y. Liu, X. Li, and Q. Hao, "A new constrained multi-objective optimization problems algorithm based on group-sorting," in *Proc. Genet. Evol. Comput. Conf. Compan.*, 2019, pp. 221–222.

[114] 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, doi: 10.1109/TEVC.2020.3047835.

[115] M. Miyakawa, K. Takadama, and H. Sato, "Directed mating using inverted PBI function for constrained multi-objective optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2015, pp. 2929–2936.

[116] M. Miyakawa, K. Takadama, and H. Sato, "Controlling selection areas of useful infeasible solutions for directed mating in evolutionary constrained multi-objective optimization," *Ann. Math. Artif. Intell.*, vol. 76, nos. 1–2, pp. 25–46, 2016.

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

[118] H. K. Singh, T. Ray, and W. Smith, "Surrogate assisted simulated annealing (SASA) for constrained multi-objective optimization," in *Proc. IEEE Congr. Evol. Comput.*, 2010, pp. 1–8.

[119] P. Singh, I. Couckuyt, F. Ferranti, and T. Dhaene, "A constrained multi-objective surrogate-based optimization algorithm," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2014, pp. 3080–3087.

[120] R. Datta and R. G. Regis, "A surrogate-assisted evolution strategy for constrained multi-objective optimization," *Expert Syst. Appl.*, vol. 57, pp. 270–284, Sep. 2016.

[121] Z. Zhang and S. Qian, "Artificial immune system in dynamic environments solving time-varying non-linear constrained multi-objective problems," *Soft Comput.*, vol. 15, no. 7, pp. 1333–1349, 2011.

[122] J. Wei and Y. Wang, "Hyper rectangle search based particle swarm algorithm for dynamic constrained multi-objective optimization problems," in *Proc. Evol. Comput.*, 2012, pp. 1–8.

[123] B. Qu and P. N. Suganthan, "Constrained multi-objective optimization algorithm with an ensemble of constraint handling methods," *Eng. Optim.*, vol. 43, no. 4, pp. 403–416, 2011.

[124] S. Zapotecas-Martínez and C. A. Coello Coello, "MONSS: A multi-objective nonlinear simplex search approach," *Eng. Optim.*, vol. 48, no. 1, pp. 16–38, 2016.

[125] A. Sadollah, H. Eskandar, and J. H. Kim, "Water cycle algorithm for solving constrained multi-objective optimization problems," *Appl. Soft Comput.*, vol. 27, pp. 279–298, Feb. 2015.

[126] S. Qian, Y. Ye, B. Jiang, and J. Wang, "Constrained multiobjective optimization algorithm based on immune system model," *IEEE Trans. Cybern.*, vol. 46, no. 9, pp. 2056–2069, Sep. 2016.

[127] L. Li, C. He, W. Xu, and L. Pan, "Pioneer selection for evolutionary multiobjective optimization with discontinuous feasible region," *Swarm Evol. Comput.*, vol. 65, Aug. 2021, Art. no. 100932, doi: 10.1016/j.swevo.2021.100932.

[128] A. Gupta, Y.-S. Ong, and L. Feng, "Multifactorial evolution: Toward evolutionary multitasking," *IEEE Trans. Evol. Comput.*, vol. 20, no. 3, pp. 343–357, Jun. 2016.

[129] J. Liang *et al.*, "Evolutionary multi-task optimization for parameters extraction of photovoltaic models," *Energy Convers. Manag.*, vol. 207, Mar. 2020, Art. no. 112509.

[130] Y.-S. Ong and A. Gupta, "Evolutionary multitasking: A computer science view of cognitive multitasking," *Cogn. Comput.*, vol. 8, no. 2, pp. 125–142, 2016.

[131] E. K. Burke, M. R. Hyde, G. Kendall, G. Ochoa, E. Özcan, and J. R. Woodward, "A classification of hyper-heuristic approaches: Revisited," in *Handbook of Metaheuristics*. Cham, Switzerland: Springer, 2019, pp. 453–477.

[132] M. Abd Elaziz and S. Mirjalili, "A hyper-heuristic for improving the initial population of whale optimization algorithm," *Knowl. Based Syst.*, vol. 172, pp. 42–63, May 2019.

[133] N. Srinivas and K. Deb, "Muiltiobjective optimization using nondominated sorting in genetic algorithms," *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, 1994.

[134] M. Tanaka, H. Watanabe, Y. Furukawa, and T. Tanino, "GA-based decision support system for multicriteria optimization," in *Proc. IEEE Int. Conf. Syst. Man Cybern. Intell. Syst. 21st Century*, vol. 2, 1995, pp. 1556–1561.

[135] A. Osyczka and S. Kundu, "A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm," *Struct. Optim.*, vol. 10, no. 2, pp. 94–99, 1995.

[136] K. Deb, A. Pratap, and T. Meyarivan, "Constrained test problems for multi-objective evolutionary optimization," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.*, 2001, pp. 284–298.

[137] 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.; School Elect. Electron. Eng., Nanyang Tech. Univ., Nanyang Avenue, Singapore; Dept. Mech. Eng., Clemson Univ., Clemson, SC, USA, Rep. CES-887, 2008.

[138] J.-P. Li, Y. Wang, S. Yang, and Z. Cai, "A comparative study of constraint-handling techniques in evolutionary constrained multiobjective optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2016, pp. 4175–4182.

[139] 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.

[140] Z. Fan, W. Li, X. Cai, H. Li, K. Hu, and H. Yin, "Difficulty controllable and scalable constrained multi-objective test problems," in *Proc. Int. Conf. Ind. Informat. Comput. Technol. Intell. Technol. Ind. Inf. Integr.*, 2015, pp. 76–83.

[141] C. Picard and J. Schiffmann, "Realistic constrained multiobjective optimization benchmark problems from design," *IEEE Trans. Evol. Comput.*, vol. 25, no. 2, pp. 234–246, Apr. 2021.

[142] O. Cuate, L. Uribe, A. Lara, and O. Schütze, "A benchmark for equality constrained multi-objective optimization," *Swarm Evol. Comput.*, vol. 52, Feb. 2020, Art. no. 100619.

[143] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization*. London, U.K.: Springer, 2005, pp. 105–145.

[144] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.

[145] 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.

[146] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, 2011.

[147] A. Osyczka, "Multicriteria optimization for engineering design," in *Design Optimization*. Oxford, U.K.: Elsevier, 1985, pp. 193–227.

[148] M. A. Tawhid and V. Savsani, "A novel multi-objective optimization algorithm based on artificial algae for multi-objective engineering design problems," *Appl. Intell.*, vol. 48, no. 10, pp. 3762–3781, 2018.

[149] Y. Yang, J. Liu, S. Tan, and H. Wang, "Application of constrained multi-objective evolutionary algorithm in multi-source compressed-air pipeline optimization problems," *IFAC-PapersOnLine*, vol. 51, no. 18, pp. 168–173, 2018.

[150] T. Ray, K. Tai, and K. C. Seow, "Multiobjective design optimization by an evolutionary algorithm," *Eng. Optim.*, vol. 33, no. 4, pp. 399–424, 2001.

[151] Y. Xiang, X. Yang, Y. Zhou, and H. Huang, "Enhancing decomposition-based algorithms by estimation of distribution for constrained optimal software product selection," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 245–259, Apr. 2020.

[152] C. A. Coello Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 256–279, Jun. 2004.

[153] M. Kotinis, "A particle swarm optimizer for constrained multi-objective engineering design problems," *Eng. Optim.*, vol. 42, no. 10, pp. 907–926, 2010.

[154] Q. Zhang, Q. Wang, and G. Li, "Switched system identification based on the constrained multi-objective optimization problem with application to the servo turntable," *Int. J. Control Autom. Syst.*, vol. 14, no. 5, pp. 1153–1159, 2016.

[155] L. Vallozzi, W. Vandendriessche, H. Rogier, C. Hertleer, and M. Scarpello, "Design of a protective garment GPS antenna," *Microw. Opt. Technol. Lett.*, vol. 51, no. 6, pp. 1504–1508, 2009.

[156] P. Singh, M. Rossi, I. Couckuyt, D. Deschrijver, H. Rogier, and T. Dhaene, "Constrained multi-objective antenna design optimization using surrogates," *Int. J. Numer. Model. Electron. Netw. Devices Fields*, vol. 30, no. 6, pp. 1–5, 2017.

[157] A. Gadagi and C. Adake, "A constrained multi-objective optimization of turning process parameters by genetic algorithm and particle swarm optimization techniques," *Mater. Today Proc.*, vol. 42, pp. 1207–1212, Feb. 2021.

[158] H. Afshari, W. Hare, and S. Tesfamariam, "Constrained multi-objective optimization algorithms: Review and comparison with application in reinforced concrete structures," *Appl. Soft Comput.*, vol. 83, Oct. 2019, Art. no. 105631.

[159] U. Güvenc, Y. Sönmez, S. Duman, and N. Yörükeren, "Combined economic and emission dispatch solution using gravitational search algorithm," *Scientia Iranica*, vol. 19, no. 6, pp. 1754–1762, 2012.

[160] M.-R. Chen, G.-Q. Zeng, and K.-D. Lu, "Constrained multi-objective population extremal optimization based economic-emission dispatch incorporating renewable energy resources," *Renew. Energy*, vol. 143, pp. 277–294, Dec. 2019.

[161] M. A. El-Shorbagy and A. Mousa, "Constrained multiobjective equilibrium optimizer algorithm for solving combined economic emission dispatch problem," *Complexity*, vol. 2021, Jan. 2021, Art. no. 6672131.

[162] S. M. Mohseni-Bonab, A. Rabiee, and B. Mohammadi-Ivatloo, "Voltage stability constrained multi-objective optimal reactive power dispatch under load and wind power uncertainties: A stochastic approach," *Renew. Energy*, vol. 85, pp. 598–609, Jan. 2016.

[163] L. H. Lee, K. C. Tan, K. Ou, and Y. H. Chew, "Vehicle capacity planning system: A case study on vehicle routing problem with time windows," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 33, no. 2, pp. 169–178, Mar. 2003.

[164] C.-T. Chen and H.-V. Pham, "Trajectory planning in parallel kinematic manipulators using a constrained multi-objective evolutionary algorithm," *Nonlinear Dyn.*, vol. 67, no. 2, pp. 1669–1681, 2012.

[165] S. Maity, A. Roy, and M. Maiti, "A rough multi-objective genetic algorithm for uncertain constrained multi-objective solid travelling salesman problem," *Granular Comput.*, vol. 4, no. 1, pp. 125–142, 2019.

[166] C. Ramirez-Atencia and D. Camacho, "Constrained multi-objective optimization for multi-UAV planning," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 6, pp. 2467–2484, 2019.

[167] K. Musselman and J. Talavage, "A tradeoff cut approach to multiple objective optimization," *Oper. Res.*, vol. 28, no. 6, pp. 1424–1435, 1980.

[168] H. Mala-Jetmarova, N. Sultanova, and D. Savic, "Lost in optimisation of water distribution systems? A literature review of system operation," *Environ. Model. Softw.*, vol. 93, pp. 209–254, Jul. 2017.

[169] T. Ray, K. Tai, and C. Seow, "An evolutionary algorithm for multiobjective optimization," *Eng. Optim*, vol. 33, no. 3, pp. 399–424, 2001.

[170] Q. Ouyang, W. Lu, Z. Hou, Y. Zhang, S. Li, and J. Luo, "Chance-constrained multi-objective optimization of groundwater remediation design at dnapls-contaminated sites using a multi-algorithm genetically adaptive method," *J. Contaminant Hydrol.*, vol. 200, pp. 15–23, May 2017.

[171] S. Y. D. Sorkhabi, D. A. Romero, J. C. Beck, and C. H. Amon, "Constrained multi-objective wind farm layout optimization: Novel constraint handling approach based on constraint programming," *Renew. Energy*, vol. 126, pp. 341–353, Oct. 2018.

[172] J. Marti, L. Geissbühler, V. Becattini, A. Haselbacher, and A. Steinfeld, "Constrained multi-objective optimization of thermocline packed-bed thermal-energy storage," *Appl. Energy*, vol. 216, pp. 694–708, Apr. 2018.

[173] Z. Su *et al.*, "Enhanced constraint handling for reliability-constrained multiobjective testing resource allocation," *IEEE Trans. Evol. Comput.*, vol. 25, no. 3, pp. 537–551, Jun. 2021.

[174] A. Maminov and M. Posypkin, "Constrained multi-objective robot's design optimization," in *Proc. IEEE Conf. Russian Young Res. Electr. Electron. Eng. (EIConRus)*, 2020, pp. 1992–1995.

[175] Y. Chen and A. Zhou, "Variable division and optimization for constrained multiobjective portfolio problems," 2021, *arXiv:2101.08552*.

[176] M. Schlegel and W. Marquardt, "Adaptive switching structure detection for the solution of dynamic optimization problems," *Ind. Eng. Chem. Res.*, vol. 45, no. 24, pp. 8083–8094, 2006.

[177] Q. Chen, J. Ding, T. Chai, and Q. Pan, "Evolutionary optimization under uncertainty: The strategies to handle varied constraints for fluid catalytic cracking operation," *IEEE Trans. Cybern.*, early access, Jul. 28, 2020, doi: 10.1109/TCYB.2020.3005893.

[178] C. Yang and J. Ding, "Constrained dynamic multi-objective evolutionary optimization for operational indices of beneficiation process," *J. Intell. Manuf.*, vol. 30, no. 7, pp. 2701–2713, 2019.

[179] R. Azzouz, S. Bechikh, L. B. Said, and W. Trabelsi, "Handling time-varying constraints and objectives in dynamic evolutionary multiobjective optimization," *Swarm Evol. Comput.*, vol. 39, pp. 222–248, Apr. 2018.

[180] 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.

[181] C. Yue *et al.*, "Differential evolution using improved crowding distance for multimodal multiobjective optimization," *Swarm Evol. Comput.*, vol. 62, Apr. 2021, Art. no. 100849.

[182] J. Liang *et al.*, "A clustering-based differential evolution algorithm for solving multimodal multi-objective optimization problems," *Swarm Evol. Comput.*, vol. 60, Feb. 2021, Art. no. 100788.

[183] A. Kurpati, S. Azarm, and J. Wu, "Constraint handling improvements for multiobjective genetic algorithms," *Struct. Multidiscipl. Optim.*, vol. 23, no. 3, pp. 204–213, 2002.

[184] K. Deb and A. Srinivasan, "Monotonicity analysis, evolutionary multiobjective optimization, and discovery of design principles," Indian Inst. Technol., Kanpur Genet. Algorithm Lab. (KanGAL), Kanpur, India, Rep. 2006004, 2006.

[185] R. Tanabe and H. Ishibuchi, "A review of evolutionary multimodal multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 193–200, Feb. 2020.

[186] D. Zhan and H. Xing, "Expected improvement for expensive optimization: A review," *J. Global Optim.*, vol. 78, no. 3, pp. 507–544, 2020.

[187] J.-S. Pan, N. Liu, S.-C. Chu, and T. Lai, "An efficient surrogate-assisted hybrid optimization algorithm for expensive optimization problems," *Inf. Sci.*, vol. 561, pp. 304–325, Jun. 2021.

[188] B. Liu, Q. Zhang, and G. G. E. Gielen, "A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 180–192, Apr. 2014.

[189] A. I. Forrester, N. W. Bressloff, and A. J. Keane, "Optimization using surrogate models and partially converged computational fluid dynamics simulations," *Proc. Royal Soc. A, Math. Phys. Eng. Sci.*, vol. 462, no. 2071, pp. 2177–2204, 2006.

[190] Y. Tian, C. Lu, X. Zhang, K. C. Tan, and Y. Jin, "Solving large-scale multiobjective optimization problems with sparse optimal solutions via unsupervised neural networks," *IEEE Trans. Cybern.*, vol. 51, no. 6, pp. 3115–3128, Jun. 2021.

[191] T.-C. Chiang and W.-H. Hsu, "A knowledge-based evolutionary algorithm for the multiobjective vehicle routing problem with time windows," *Comput. Oper. Res.*, vol. 45, pp. 25–37, May 2014.

[192] C. He, R. Cheng, C. Zhang, Y. Tian, Q. Chen, and X. Yao, "Evolutionary large-scale multiobjective optimization for ratio error estimation of voltage transformers," *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 868–881, Oct. 2020.

[193] Y. Xiang, Y. Zhou, Z. Zheng, and M. Li, "Configuring software product lines by combining many-objective optimization and sat solvers," *ACM Trans. Softw. Eng. Methodol.*, vol. 26, no. 4, pp. 1–46, 2018.

[194] Y. Tian *et al.*, "Evolutionary large-scale multi-objective optimization: A survey," *J. ACM*, vol. 54, no. 8, p. 174, 2022.

[195] W. Du, W. Song, Y. Tang, Y. Jin, and F. Qian, "Searching for robustness intervals in evolutionary robust optimization," *IEEE Trans. Evol. Comput.*, vol. 26, no. 1, pp. 58–72, Feb. 2022, doi: 10.1109/TEVC.2021.3092343.

[196] Y. He, M. Shahidehpour, Z. Li, C. Guo, and B. Zhu, "Robust constrained operation of integrated electricity-natural gas system considering distributed natural gas storage," *IEEE Trans. Sustain. Energy*, vol. 9, no. 3, pp. 1061–1071, Jul. 2018.

[197] G. Nicosia, S. Rinaudo, and E. Sciacca, "An evolutionary algorithm-based approach to robust analog circuit design using constrained multi-objective optimization," in *Proc. Int. Conf. Innov. Techn. Appl. Artif. Intell.*, 2007, pp. 7–20.

[198] S. Kwag and S.-Y. Ok, "Robust design of seismic isolation system using constrained multi-objective optimization technique," *KSCE J. Civil Eng.*, vol. 17, no. 5, pp. 1051–1063, 2013.

[199] L. Argote, P. Ingram, J. M. Levine, and R. L. Moreland, "Knowledge transfer in organizations: Learning from the experience of others," *Org. Behav. Human Decis. Processes*, vol. 82, no. 1, pp. 1–8, 2000.

[200] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[201] B. Mahesh, "Machine learning algorithms—A review," *Int. J. Sci. Res.*, vol. 9, no. 1, pp. 381–386, 2020.

[202] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 1, pp. 249–270, Jan. 2022. doi: 10.1109/TKDE.2020.2981333.

[203] H. Jiang, Y. Wang, Y. Tian, X. Zhang, and J. Xiao, "Feature construction for meta-heuristic algorithm recommendation of capacitated vehicle routing problems," *ACM Trans. Evol. Learn. Optim.*, vol. 1, no. 1, p. 3, Apr. 2021. [Online]. Available: https://doi.org/10.1145/3447540

[204] K. M. Sallam, S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Landscape-assisted multi-operator differential evolution for solving constrained optimization problems," *Expert Syst. Appl.*, vol. 162, Dec. 2020, Art. no. 113033.

[205] K. Sallam, S. Elsayed, R. Sarker, and D. Essam, "Landscape-based differential evolution for constrained optimization problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2018, pp. 1–8.

[206] P. A. Consoli, Y. Mei, L. L. Minku, and X. Yao, "Dynamic selection of evolutionary operators based on online learning and fitness landscape analysis," *Soft Comput.*, vol. 20, no. 10, pp. 3889–3914, 2016.

Jing Liang (Senior Member, IEEE) received the B.E. degree in automation from the Harbin Institute of Technology, Harbin, China, in 2003, and the Ph.D. degree in electrical and electronic engineering from Nanyang Technological University, Singapore, in 2009.

She is currently a Professor with the School of Electrical Engineering, Zhengzhou University, Zhengzhou, China. Her main research interests are evolutionary computation, swarm intelligence, multiobjective optimization, and neural network.

Prof. Liang currently serves as an Associate Editor for the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and the *Swarm and Evolutionary Computation*.

Xuanxuan Ban (Graduate Student Member, IEEE) received the B.E. degree from Luoyang Institute of Science and Technology, Luoyang, China, in 2019. He is currently pursuing the M.S. degree with Zhengzhou University, Zhengzhou, China.

His current research interests include evolutionary computing, differential evolution, and constrained optimization.

Kunjie Yu (Member, IEEE) received the B.E. degree in automation from the Zhengzhou University of Light Industry, Zhengzhou, China, in 2012, and the Ph.D. degree in control science and engineering from the East China University of Science and Technology, Shanghai, China, in 2017.

He is currently an Associate Professor with the School of Electrical Engineering, Zhengzhou University, Zhengzhou. His current research interests include evolutionary computation, constrained optimization, multiobjective optimization, and their applications in chemical process, photovoltaic system, and energy system.

Boyang Qu (Member, IEEE) received the B.E. and Ph.D. degrees in electrical and electronic engineering from Nanyang Technological University, Singapore, in 2007 and 2012, respectively.

He is a Professor with the School of Electric and Information Engineering, Zhongyuan University of Technology, Zhengzhou, China. His research interests include machine learning, neural network, genetic and evolutionary algorithms, swarm intelligence, and multiobjective optimization.

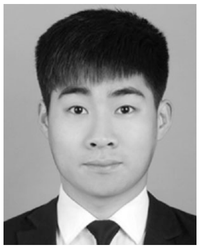
Prof. Qu is an Associate Editor of the *Swarm and Evolutionary Computation*.

Kangjia Qiao (Member, IEEE) received the B.E. degree from Jianghan University, Wuhan, China, in 2018, and the M.S. degree from Zhengzhou University, Zhengzhou, China, in 2021, where he is currently pursuing the Ph.D. degree.

His current research interests include differential evolution, multitasking optimization, constrained optimization, and multimodal multiobjective optimization.

Caitong Yue (Member, IEEE) received the B.E. degree in bioengineering and the Ph.D. degree in control science and engineering from Zhengzhou University, Zhengzhou, China, in 2014 and 2020, respectively.

He is currently a Lecturer with the School of Electrical Engineering, Zhengzhou University. His research interests include multimodal multiobjective optimization, pattern recognition, neural network, and particle swarm optimization.



Ke Chen (Member, IEEE) received the B.E. degree from Weifang University, Weifang, China, in 2014, the M.S. degree from Yanshan University, Qinhuangdao, China, in 2017, and the Ph.D. degree from Shandong University, Jinan, China, in 2021.

He is currently a Research Associate Professor with the School of Electrical Engineering, Zhengzhou University, Zhengzhou, China. His current research interests include evolutionary computation, feature selection, multitasking optimization, multiobjective optimization, and machine learning.

Kay Chen Tan (Fellow, IEEE) received the B.Eng. degree (First Class Hons.) in electronics and electrical engineering and the Ph.D. degree from the University of Glasgow, Glasgow, U.K., in 1994 and 1997, respectively.

He is currently a Chair Professor with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, SAR. He is an Honorary Professor with the University of Nottingham, Nottingham, U.K., and the Chief Co-Editor of Springer Book Series on *Machine Learning: Foundations, Methodologies, and Applications*. He has published over 300 refereed articles and seven books.

Dr. Tan was the Editor-in-Chief of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION from 2015 to 2020, and currently serves as an editorial board member of over ten journals.