Indicator-Based Constrained Multiobjective Evolutionary Algorithms

Zhi-Zhong Liu, Yong Wang, *Senior Member, IEEE*, and Bing-Chuan Wang

***Abstract*—Solving constrained multiobjective optimization problems (CMOPs) is a challenging task since it is necessary to optimize several conflicting objective functions and handle various constraints simultaneously. A promising way to solve CMOPs is to integrate multiobjective evolutionary algorithms (MOEAs) with constraint-handling techniques, and the resultant algorithms are called constrained multiobjective evolutionary algorithms (CMOEAs). At present, many attempts have been made to combine dominance-based and decomposition-based MOEAs with diverse constraint-handling techniques together. However, for another main branch of MOEAs, i.e., indicator-based MOEAs, almost no effort has been devoted to extending them for solving CMOPs. In this paper, we make the first study on the possibility and rationality of combining indicator-based MOEAs with constraint-handling techniques together. Afterward, we develop an indicator-based CMOEA framework which can combine indicator-based MOEAs with constraint-handling techniques conveniently. Based on the proposed framework, nine indicator-based CMOEAs are developed. Systemic experiments have been conducted on 19 widely used constrained multiobjective optimization test functions to identify the characteristics of these nine indicator-based CMOEAs. The experimental results suggest that both indicator-based MOEAs and constraint-handing techniques play very important roles in the performance of indicator-based CMOEAs. Some practical suggestions are also given about how to select appropriate indicator-based CMOEAs. Besides, we select a superior approach from these nine indicator-based CMOEAs and compare its performance with five state-of-the-art CMOEAs. The comparison results suggest that the selected indicator-based CMOEA can obtain quite competitive performance. It is thus believed that our work would encourage researchers to pay more attention to indicator-based CMOEAs in the future.**

***Index Terms*—Constrained multiobjective optimization problems, constrained multiobjective evolutionary algorithms, indicator, constraint-handling technique**

I. Introduction

CONSTRAINED multiobjective optimization problems (CMOPs) refer to multiobjective optimization problems including constraints, which are frequently encountered in many real-world applications [1]–[5]. Without loss of generality, a CMOP can be described as:

$$\begin{aligned} \min \quad & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))^T \in \mathbb{F} \\ \text{s.t.} \quad & g_j(\mathbf{x}) \le 0,\ j = 1, 2, \ldots, l \\ & h_j(\mathbf{x}) = 0,\ j = l+1, l+2, \ldots, n \\ & \mathbf{x} = (x_1, x_2, \ldots, x_D)^T \in \mathbb{S} \end{aligned} \tag{1}$$

where $\mathbf{x}$ denotes a D-dimensional decision vector, $\mathbb{S}$ refers to the decision space, $\mathbf{F}(\mathbf{x})$ is the objective vector including m objective functions, $\mathbb{F}$ denotes the objective space, $f_i(\mathbf{x})$ is the ith objective function, $g_j(\mathbf{x})$ refers to the jth inequality constraint, $h_j(\mathbf{x})$ denotes the $(j-l)$th equality constraint, and l and $(n-l)$ are the number of inequality and equality constraints, respectively. Since several conflicting objective functions have to be optimized and various constraints should be satisfied at the same time [6], [7], obviously, solving CMOPs poses great difficulties and challenges to current methods [8]–[10].

Nowadays, a promising way to solve CMOPs is to integrate multiobjective evolutionary algorithms (MOEAs) [11] with constraint-handling techniques [12] [1], since MOEAs [13]–[17] and constraint-handling techniques [18]–[24] have gained great reputation in optimizing several conflicting objective functions and dealing with diverse constraints, respectively. Along this line, many researchers focused on combining different constraint-handling techniques with dominance-based [25] and decomposition-based [26] MOEAs, and thus many dominance-based and decomposition-based CMOEAs have been proposed during the last two decades [27]–[31]. Nevertheless, as another main branch of MOEAs, i.e., indicator-based MOEAs [32], to the best of our knowledge, almost no research has been conducted to apply them to solve CMOPs, even they have exhibited excellent performance in evolutionary multiobjective optimization [11].

To alleviate this issue, in this paper, we make the first attempt to investigate the rationality and superiority behind the combination of indicator-based MOEAs and constraint-handling techniques. Specifically, a generic indicator-based CMOEA framework is proposed, in which different indicator-based MOEAs can be readily integrated with different constraint-handling techniques. Following the proposed framework, nine indicator-based CMOEAs are developed in this study by considering three well-known indicator-based MOEAs (i.e., HypE [33], IBEA [34] and ISDE [35]) and three famous constraint-handling techniques (i.e., feasibility

[1]The resultant algorithms are called constrained multiobjective evolutionary algorithms (CMOEAs).

This work was supported in part by the Innovation-Driven Plan in Central South University under Grant 2018CX010, in part by the National Natural Science Foundation of China under Grants 61673397 and 61976225, and in part by the Beijing Advanced Innovation Center for Intelligent Robots and Systems under Grant 2018IRS06. (*Corresponding author: Yong Wang*)

Z.-Z. Liu is with the School of Automation, Central South University, Changsha 410083, China, and also with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China. (Email: zhizhongliu@csu.edu.cn)

Y. Wang and B.-C. Wang are with the School of Automation, Central South University, Changsha 410083, China. (Email: ywang@csu.edu.cn; wangbingchuancityu@gmail.com).

rule (FR) [36], stochastic ranking (SR) [37], and ε constrained method [38]). Thereafter, the performance of these nine indicator-based CMOEAs is tested on 19 widely used constrained multiobjective optimization test functions.

The main contributions of this paper are summarized as follows:

- By investigating the rationality and possibility of indicator-based MOEAs for CMOPs, this paper suggests that the combination of indicator-based MOEAs and constraint-handling techniques is a natural way to solve CMOPs. Thus, a new perspective for solving CMOPs is provided in this study.
- An open indicator-based CMOEA framework is proposed in this paper, which is able to accommodate different indicator-based MOEAs and different constraint-handling techniques. Based on this framework, nine new indicator-based CMOEAs are developed by taking three indicator-based MOEAs and three constraint-handling techniques into account.
- Systematic experiments have been conducted on 19 test functions to identify the properties of these nine indicator-based CMOEAs. The experimental results indicate that the performance of indicator-based CMOEAs relies on both indicator-based MOEAs and constraint-handling techniques. It is also observed that there is no such a combination of an indicator-based MOEA and a constraint-handling technique, which can obtain the best performance on each test function. Therefore, we give some practical suggestions on how to select desired indicator-based CMOEAs.
- The comparison experiments have been conducted between a superior indicator-based CMOEA (i.e., HypE-FR) and five other state-of-the-art CMOEAs. The experimental results suggest that, overall, HypE-FR can obtain quite competitive results. Therefore, we can conclude that indicator-based CMOEAs are worth extensively and intensively studying in the future.

The rest of this paper is organized as follows. Section II introduces the related work. The details of the proposed indicator-based CMOEA framework are given in Section III. Subsequently, the experimental setup is introduced in Section IV and the experiments and discussions are carried out in Section V. Finally, Section VI concludes this paper.

II. Related Work

In this section, we first introduce some basic definitions in CMOPs, and then give a brief introduction to some representative indicator-based MOEAs and constraint-handling techniques.

A. Basic Definitions in CMOPs

- *Pareto Dominance:* We only consider the m objective functions in (1). For two decision vectors $\mathbf{x}_u$ and $\mathbf{x}_v$, if $\forall i \in \{1,2,\ldots,m\}$, $f_i(\mathbf{x}_u) \leq f_i(\mathbf{x}_v)$ and $\exists j \in \{1,2,\ldots,m\}$, $f_j(\mathbf{x}_u) < f_j(\mathbf{x}_v)$, then $\mathbf{x}_u$ is said to Pareto dominate $\mathbf{x}_v$, denoted as $\mathbf{x}_u \prec \mathbf{x}_v$.
- *Feasible Region:* The feasible region of (1) is defined as $\mathbb{O} = \{\mathbf{x} \in \mathbb{S} | CV(\mathbf{x}) = 0\}$, where

$$CV(\mathbf{x}) = \sum_{i=1}^{n} CV_i(\mathbf{x}) \tag{2}$$

$$CV_i(\mathbf{x}) = \begin{cases} max(0, g_i(\mathbf{x})), \text{ if } i \leq l \\ max(0, |h_i(\mathbf{x})| - \delta), \text{ otherwise} \end{cases}, i = 1, \ldots, n \tag{3}$$

  It is cleat that $CV(\mathbf{x})$ is the degree of constraint violation of $\mathbf{x}$ on all constraints and $CV_i(\mathbf{x})$ is the degree of constraint violation of $\mathbf{x}$ on the ith constraint. In (3), δ is a very small positive value to relax equality constraints.
- *Pareto Optimal Solution:* For $\mathbf{x}_u \in \mathbb{O}$, if $\neg\exists \mathbf{x}_v \in \mathbb{O}$, $\mathbf{x}_v \prec \mathbf{x}_u$, then $\mathbf{x}_u$ is a Pareto optimal solution of (1).
- *Pareto Optimal Set:* The Pareto optimal set of (1) is the set of all the Pareto optimal solutions: $\mathcal{PS} = \{\mathbf{x}_u \in \mathbb{O} | \neg\exists \mathbf{x}_v \in \mathbb{O}, \mathbf{x}_v \prec \mathbf{x}_u\}$.
- *Pareto Front:* The Pareto front of (1) is the image of $\mathcal{PS}$ in the objective space: $\mathcal{PF} = \{\mathbf{F}(\mathbf{x}_u) | \mathbf{x}_u \in \mathcal{PS}\}$.
- *Approximation Set:* An approximation set $\mathcal{A} \subset \mathbb{F}$ is defined as the image of a feasible solution set in which no feasible solution Pareto dominates or is equal to any other feasible solution. The goal of solving (1) is to obtain a well-converged and well-distributed $\mathcal{A}$.

B. Indicator-Based MOEAs

Three representative indicator-based MOEAs, i.e., HypE [33], IBEA [34], and ISDE [35], are considered in this paper, which belong to hypervolume-based MOEAs, $I_{(\epsilon)+}$-based MOEAs, and I_{SDE}-based MOEAs, respectively. Next, we give a brief introduction to these three different kinds of indicator-based MOEAs.

1) Hypervolume-Based MOEAs: The hypervolume indicator is the most commonly used indicator in indicator-based MOEAs, since it has an attractive property, that is, it is strictly monotonic with regard to Pareto dominance [33]. For population $\mathcal{P}$, its hypervolume is calculated as:

$$HV(\mathcal{P}) = L\left(\bigcup_{\mathbf{a} \in \mathcal{NS}} \{\mathbf{b} \in \mathbb{F} \mid \mathbf{a} \prec \mathbf{b} \prec \mathbf{R}\}\right) \tag{4}$$

where L is the Lebesgue measure, $\mathcal{NS}$ denotes the set of nondominated solutions in $\mathcal{P}$, and $\mathbf{R}$ refers to the reference point in the objective space.

At present, many hypervolume-based MOEAs have been proposed [39] which adopt different schemes to assign fitness values to individuals. For most hypervolume-based MOEAs [40], [41], they usually first conduct a nondominated sorting procedure [25] to identify a particular nondominated level $\mathcal{F}_l$ [2]. Afterward, for individual $\mathbf{x}_u$ in $\mathcal{F}_l$, its indicator-based fitness value is defined as the hypervolume loss resulting from the removal of $\mathbf{x}_u$:

$$FV_{HV}(\mathbf{x}_u) = HV(\mathcal{F}_l) - HV(\mathcal{F}_l \backslash \mathbf{x}_u) \tag{5}$$

[2]Note that $\sum_{i=1}^{l-1} |\mathcal{F}_i| < N < \sum_{i=1}^{l} |\mathcal{F}_i|$, $\mathcal{F}_i$ denotes the ith nondominated level, and N is the population size.

Another approach to assign fitness value is presented in [34], in which a binary hypervolume indicator is designed. In this approach, for two individuals $\mathbf{x}_u$ and $\mathbf{x}_v$ in $\mathcal{P}$, an additive indicator I_{HD} is defined as:

$$I_{HD}(\mathbf{x}_u,\mathbf{x}_v)=\begin{cases} HV(\mathbf{x}_v)-HV(\mathbf{x}_u), & \text{if } \mathbf{x}_v \prec \mathbf{x}_u \\ HV(\{\mathbf{x}_v,\mathbf{x}_v\})-HV(\mathbf{x}_u), & \text{otherwise} \end{cases} \quad (6)$$

Then, the indicator-based fitness value of $\mathbf{x}_u$ is calculated as:

$$FV_{HV}(\mathbf{x}_u)=\sum_{\mathbf{x}_v\in\mathcal{P},\mathbf{x}_v\neq\mathbf{x}_u} -e^{-I_{HD}(\mathbf{x}_u,\mathbf{x}_v)/0.05} \quad (7)$$

Unlike the above approaches, HypE [33], which might be the current most famous hypervolume-based MOEA, proposes a brand-new strategy to assign fitness values to individuals in both the mating and environmental selection procedures. In this strategy, the space enclosed by the non-dominated solutions and the reference point $\mathbf{R}$ is split into a series of specific partitions. Subsequently, the hypervolume contribution of each partition is computed and the aggregation method is applied to assign the fitness value to each individual. For interested readers, the detailed information of this strategy can be obtained from [33] and the source code of HypE can be downloaded from: http://www.tik.ee.ethz.ch/sop/download/supplementary/hype/.

2) $I_{(\epsilon)^+}$-Based MOEAs: The $I_{(\epsilon)^+}$ indicator is a binary quality indicator extending from the Pareto dominance relation [34]. In general, for two individuals $\mathbf{x}_u$ and $\mathbf{x}_v$ in $\mathcal{P}$, the $I_{(\epsilon)^+}$ indicator is defined as:

$$I_{(\epsilon)^+}(\mathbf{x}_u,\mathbf{x}_v)=\min_{\epsilon}(f_i(\mathbf{x}_u)-\epsilon\le f_i(\mathbf{x}_v), i\in\{1,\ldots,m\}) \quad (8)$$

Then, the indicator-based fitness value of $\mathbf{x}_u$ is assigned as:

$$FV_{(\epsilon)^+}(\mathbf{x}_u)=\sum_{\mathbf{x}_v\in\mathcal{P},\mathbf{x}_v\neq\mathbf{x}_u} -e^{-I_{(\epsilon)^+}(\mathbf{x}_u,\mathbf{x}_v)/0.05} \quad (9)$$

Based on the $I_{(\epsilon)^+}$ indicator, IBEA [34] is proposed which can obtain better performance compared with NSGA-II [25] and SPEA2 [42] on several continuous and discrete multiobjective benchmark test problems. Currently, the $I_{(\epsilon)^+}$ indicator has been widely employed in many-objective optimization. For example, in Two-Arch2 [43], two archives are adopted to take care of convergence and diversity, respectively, and for the convergence archive, it is updated according to the $I_{(\epsilon)^+}$ indicator. In SRA [44], the biases of different indicators are balanced by making use of stochastic ranking [37], and among these indictors, one is the $I_{(\epsilon)^+}$ indicator.

3) I_{SDE}-Based MOEAs: The I_{SDE} indicator is designed based on the shift-based density estimation [45]. For two individuals $\mathbf{x}_u$ and $\mathbf{x}_v$ in $\mathcal{P}$, the I_{SDE} indicator is defined as

$$I_{SDE}(\mathbf{x}_u,\mathbf{x}_v)=\sqrt{\sum_{1\le i\le m} sd(f_i(\mathbf{x}_u),f_i(\mathbf{x}_v))^2} \quad (10)$$

where

$$sd(f_i(\mathbf{x}_u),f_i(\mathbf{x}_v))=\begin{cases} f_i(\mathbf{x}_v)-f_i(\mathbf{x}_u), & \text{if } f_i(\mathbf{x}_u)<f_i(\mathbf{x}_v) \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

Then, the indicator-based fitness value of $\mathbf{x}_u$ is assigned as the kth minimum value [42] in the set of $\{I_{SDE}(\mathbf{x}_u,\mathbf{x}_v), \mathbf{x}_v\in\mathcal{P}\bigcap\mathbf{x}_v\neq\mathbf{x}_u\}$. Herein, k is set to $\sqrt{|\mathcal{P}|}$.

The I_{SDE} indicator is first proposed in SRA [44], in which the I_{SDE} indicator is in collaboration with the $I_{(\epsilon)^+}$ indicator for coping with many-objective optimization problems. Recently, the I_{SDE} indicator is extended to tackle both multi- and many-objective optimization problems by utilizing the information derived from the sum of objective function values [46]. Very recently, AnD is proposed by making use of both the angle-based selection and the shift-based density estimation for addressing many-objective optimization problems [35]. In this paper, a variant of AnD without angle-based selection is called ISDE.

C. Constraint-Handling Techniques

EAs are originally designed for unconstrained optimization, and thus additional constraint-handling techniques are required when EAs are applied to constrained optimization [12]. Next, we give a brief introduction to three well-known constraint-handling techniques for constrained single-objective optimization problems [3]: feasibility rule (FR) [25], stochastic ranking (SR) [37], and ε constrained method [38], [47].

1) FR: In FR, individual $\mathbf{x}_u$ is better than another individual $\mathbf{x}_v$ if one of the following three cases is satisfied:

- $\mathbf{x}_u$ is feasible and $\mathbf{x}_v$ is infeasible;
- Both $\mathbf{x}_u$ and $\mathbf{x}_v$ are infeasible, and $\mathbf{x}_u$ has smaller degree of constraint violation;
- Both $\mathbf{x}_u$ and $\mathbf{x}_v$ are feasible, and $\mathbf{x}_u$ has a smaller objective function value.

2) SR: SR employs a user-defined parameter P_f to control the criterion for comparison of two individuals $\mathbf{x}_u$ and $\mathbf{x}_v$:

- If both $\mathbf{x}_u$ and $\mathbf{x}_v$ are feasible, compare them according to objective function with probability 1.0;
- If one of $\mathbf{x}_u$ and $\mathbf{x}_v$ is infeasible, compare them according to objective function with probability P_f and according to the degree of constraint violation with probability $(1-P_f)$.

SR employs a bubble-sort-like process to rank the individuals. In general, by making use of P_f, SR aims to balance objective function and constraints in the whole optimization process [37].

3) ε Constrained Method: In this method, for two individuals $\mathbf{x}_u$ and $\mathbf{x}_v$, $\mathbf{x}_u$ is better than $\mathbf{x}_v$ if one of the following three rules is hold:

$$\begin{cases} f(\mathbf{x}_u)<f(\mathbf{x}_v), \text{ if } CV(\mathbf{x}_u)\le\varepsilon \text{ and } CV(\mathbf{x}_u)\le\varepsilon \\ f(\mathbf{x}_u)<f(\mathbf{x}_v), \text{ if } CV(\mathbf{x}_u)=CV(\mathbf{x}_u) \\ CV(\mathbf{x}_u)<CV(\mathbf{x}_v), \text{ otherwise} \end{cases} \quad (12)$$

For ε, it decreases as the generation increases:

$$\varepsilon=\begin{cases} \varepsilon_0(1-\frac{t}{T})^{cp}, \text{if } \frac{t}{T}<p \\ 0, \text{otherwise} \end{cases} \quad (13)$$

$$cp=-\frac{log\varepsilon_0+\lambda}{log(1-p)} \quad (14)$$

[3]In general, a constrained single-objective optimization problem is expressed as: min $\mathbf{f}(\mathbf{x})$, where $\mathbf{x}$ denotes a D-dimensional decision vector.

where ε_0 denotes the initial threshold and is set to the maximum degree of constraint violation in the initial population, t denotes the current generation number, and T refers to the maximum generation number. λ is set to 6 [48] and p controls the degree that the information of objective function is exploited. In this paper, p is set to 0.2.

Besides the above three methods, many other constraint-handling techniques have been proposed during the last two decades. One representative is constrained nondominated sorting (CNS) [28], which is a totally parameter-free method. Note that in CNS, each solution in the population is assigned a constrained nondominated rank (CNR) based on its constraint violation degree and Pareto rank. Then the solutions with lower values of CNR will survive into the next generation. For more information about other constraint-handling techniques, interested readers are referred to two survey papers, i.e., [49] and [50].

III. Indicator-Based CMOEAs

A. Principle Analysis

CMOPs contain both conflicting objective functions and various constraints. When solving CMOPs, two tradeoffs should be considered simultaneously: the tradeoff among objective functions, and the tradeoff between objective functions and constraints. Obviously, it is not an easy task to accomplish both tradeoffs during the evolution, thus posing great challenges and difficulties to current EAs.

However, if all objective function values can be outputted as a single value, the solution of CMOPs will become much easier since we only need to consider the balance between this single value and constraints. Fortunately, in indicator-based MOEAs, the indicator-based fitness value can serve as this single value, since it has the capability to evaluate an individual's performance on all objective functions in terms of both diversity and convergence [11], [39]. Therefore, if we intend to extend indicator-based MOEAs for solving CMOPs, we only need to consider the balance between the indicator-based fitness value and constraints. Specifically, for an individual $\mathbf{x}$ in the population, it is expected to obtain a big indicator-based fitness value $FV(\mathbf{x})$ while satisfying all constraints [4]:

$$\begin{aligned} \max \quad & FV(\mathbf{x}) \\ \text{s.t.} \quad & g_j(\mathbf{x}) \leq 0,\ j = 1, 2, \ldots, l \\ & h_j(\mathbf{x}) = 0,\ j = l+1, l+2, \ldots, n \\ & \mathbf{x} = (x_1, x_2, \ldots, x_D)^T \in \ \mathbb{S} \end{aligned} \tag{15}$$

Clearly, the above optimization problem is a constrained single-objective optimization problem. To solve this optimization problem, many constraint-handling techniques can be applied to strike a good balance between the indicator-based fineness value and constraints [12]. It is thus natural and reasonable to combine indicator-based MOEAs and constraint-handling techniques together for dealing with CMOPs.

[4]In each indicator-based MOEA considered in this paper, a big indicator-based fitness value is desired for an individual.

Algorithm 1 Framework of indicator-based CMOEAs

Input: population size N
Output: population $\mathcal{P}_t$
1: *Initialization*($\mathcal{P}_0$);
2: $t \leftarrow 0$
3: **while** *the stopping criterion is not met* **do**
4: $\quad \mathcal{P}_{t+1} = \emptyset$;
5: $\quad \mathcal{Q}_t \leftarrow$ *Mating*($\mathcal{P}_t$) ;
6: $\quad \mathcal{U}_t \leftarrow \mathcal{P}_t \bigcup \mathcal{Q}_t$;
7: $\quad \mathcal{P}_{t+1} \leftarrow$ *Enviromental-Selection*($\mathcal{U}_t$)
8: $\quad t \leftarrow t + 1$;
9: **end while**

Along this line, in this paper, a generic indicator-based CMOEA framework has been proposed and a series of indicator-based CMOEAs has been developed.

Remark 1: Constraint-handling techniques are not easy to be incorporated into dominance-based MOEAs. It is because in dominance-based MOEAs, there may exist many nondominated solutions in the population. However, in some constraint-handling techniques (e.g., SR), the individuals should be comparable in terms of both objective functions and constraints. Thus, some constraint-handling techniques cannot be incorporated into dominance-based MOEAs directly.

Remark 2: In principle, it is quite easy to combine decomposition-based MOEAs with constraint-handling techniques, since decomposition-based approaches can divide the original CMOP into a series of constrained single-objective optimization problems, and then constraint-handling techniques can be applied to solve these constrained single-objective optimization problems. However, due to the existence of constraints, the PF of a CMOP might be very complex; thus, it would be challenging to adapt the weight vectors to fit the shape of the PF. In addition, it is hard to set an appropriate ideal point since some solutions in the population might be infeasible yet with good objective function values.

B. Main Framework

Algorithm 1 presents the framework of indicator-based CMOEAs. Similar to most CMOEAs, first, an initial population $\mathcal{P}_0$ with N individuals is randomly generated in the decision space $\mathbb{S}$. Subsequently, at generation t, the mating is implemented to produce an offspring population $\mathcal{Q}_t$, and the environmental selection is performed on the union of $\mathcal{P}_t$ and $\mathcal{Q}_t$ (denoted as $\mathcal{U}_t$) to generate the next population $\mathcal{P}_{t+1}$. The above process continues until the stopping criterion is met.

It is clear that the mating and environmental selection are two main components in the proposed framework. In the mating, the parents are chosen from $\mathcal{P}_t$ through the binary tournament selection, and then the simulated binary crossover (SBX) and the polynomial mutation are executed to generate $\mathcal{Q}_t$ [25]. Note that the binary tournament selection is based on the individuals' performance in the environmental selection.

C. Environmental Selection

The environmental selection aims at selecting N individuals with good performance from $\mathcal{U}_t$ for the next generation, which can be achieved by employing constraint-handling techniques.

Algorithm 2 Combining an indicator-based MOEA with FR

Input: combined population $\mathcal{U}_t = \{\mathbf{u}_1, \ldots, \mathbf{u}_{2N}\}$
Output: $\mathcal{P}_{t+1}$

Divide $\mathcal{U}_t$ into the feasible solution set $\mathcal{S}_1 = \{\mathbf{u}_i \in \mathcal{U}_t | CV(\mathbf{u}_i) = 0\}$ and the infeasible solution set $\mathcal{S}_2 = \{\mathbf{u}_i \in \mathcal{U}_t | CV(\mathbf{u}_i) > 0\}$.
if $|\mathcal{S}_1| \geq N$ **then**
    Implement the environmental selection of the original indicator-based MOEA to select N individuals out of $\mathcal{S}_1$;
    Sort the selected N individuals according to their indicator-based fitness values and put the sorted N individuals into $\mathcal{P}_{t+1}$;
else
    Sort the individuals in $\mathcal{S}_1$ according to their indicator-based fitness values and put the sorted $\mathcal{S}_1$ into $\mathcal{P}_{t+1}$;
    Sort the individuals in $\mathcal{S}_2$ according to their degree of constraint violation, and put the best $(N - |\mathcal{S}_1|)$ individuals in the sorted $\mathcal{S}_2$ into the end of $\mathcal{P}_{t+1}$;
end if

Algorithm 3 Combining an indicator-based MOEA with SR

Input: combined population $\mathcal{U}_t = \{\mathbf{u}_1, \ldots, \mathbf{u}_{2N}\}$ and P_f
Output: $\mathcal{P}_{t+1}$

for $i = 1 : N$ **do**
    for $j = 1 : |\mathcal{U}_t| - 1$ **do**
        u = $rand$(0,1)
        if $u < P_f$ or $CV(\mathbf{u}_j) = CV(\mathbf{u}_{j+1}) = 0$ **then**
            if $FV(\mathbf{u}_j) < FV(\mathbf{u}_{j+1})$ **then**
                $swap(\mathbf{u}_j, \mathbf{u}_{j+1})$
            end if
        else
            if $CV(\mathbf{u}_j) > CV(\mathbf{u}_{j+1})$ **then**
                $swap(\mathbf{u}_j, \mathbf{u}_{j+1})$
            end if
        end if
    end for
    if $no\ swap\ is\ done$ **then**
        break;
    end if
end for
Put the top N individuals of $\mathcal{U}_t$ into $\mathcal{P}_{t+1}$

Algorithm 4 Combining an indicator-based MOEA with ε constrained method

Input: combined population $\mathcal{U}_t = \{\mathbf{u}_1, \ldots, \mathbf{u}_{2N}\}$ and ε
Output: $\mathcal{P}_{t+1}$

Divide $\mathcal{U}_t$ into $\mathcal{S}_1 = \{\mathbf{u}_i \in \mathcal{U}_t | CV(\mathbf{u}_i) \leq \varepsilon\}$ and $\mathcal{S}_2 = \{\mathbf{u}_i \in \mathcal{U}_t | CV(\mathbf{u}_i) > \varepsilon\}$.
if $|\mathcal{S}_1| \geq N$ **then**
    Implement the environmental selection of the original indicator-based MOEA to select N individuals from $\mathcal{S}_1$ ignoring all constraints;
    Sort the selected N individuals according to their indicator-based fitness values and put the sorted N individuals into $\mathcal{P}_{t+1}$;
else
    Sort the individuals in $\mathcal{S}_1$ according to their indicator-based fitness values and put the sorted $\mathcal{S}_1$ into $\mathcal{P}_{t+1}$;
    Sort the individuals in $\mathcal{S}_2$ according to their degree of constraint violation, select the top $(N - |\mathcal{S}_1|)$ individuals in the sorted $\mathcal{S}_2$, and put them into the end of $\mathcal{P}_{t+1}$;
end if

Next, we introduce how to integrate an indicator-based MOEA with three constraint-handling techniques (i.e., FR, SR, and ε constrained method), respectively.

1) Combining An Indicator-Based MOEA with FR: The combination of an indicator-based MOEA and FR is quite simple. The details are given in **Algorithm 2**. Firstly, we divide $\mathcal{U}_t$ into the feasible solution set $\mathcal{S}_1$ and the infeasible solution set $\mathcal{S}_2$. Note that in FR, feasible individuals are always better than infeasible individuals; thus, we give a priority to select feasible solutions. To be specific, if the number of individuals in $\mathcal{S}_1$, denoted as $|\mathcal{S}_1|$, is larger than or equal to N, then the environmental selection of the original indicator-based MOEA is implemented to select N individuals out of $\mathcal{S}_1$. Thereafter, the selected N individuals are sorted according to their indicator-based fitness values and added to $\mathcal{P}_{t+1}$. Otherwise, we first sort the individuals in $\mathcal{S}_1$ based on their indicator-based fitness values and put the sorted $\mathcal{S}_1$ into $\mathcal{P}_{t+1}$. Since $|\mathcal{S}_1|$ is less than N, the other $(N - |\mathcal{S}_1|)$ individuals should be selected from $\mathcal{S}_2$. Herein, we sort the individuals in $\mathcal{S}_2$ according to their degree of constraint violation, and copy the best $(N - |\mathcal{S}_1|)$ individuals in the sorted $\mathcal{S}_2$ to the end of $\mathcal{P}_{t+1}$.

Remark 3 : To facilitate the binary tournament selection in the mating, in this paper, the index of an individual in $\mathcal{P}_{t+1}$ is based on its performance. That is, if the individuals are sorted according to their indicator-based fitness values, the individual with better indicator-based fitness values will be ranked in front of the sorted population. Similarity, if the individuals are sorted based on their degree of constraint violation, the individuals with smaller constraint violations will be ranked in front of the sorted population.

2) Combining An Indicator-Based MOEA with SR: Similarly, it is easy to combine an indicator-based MOEA with SR. As presented in **Algorithm 3**, for each swap, a pair of two adjacent individuals is compared based on the indicator-based fitness value or the degree of constraint violation with the probability P_f or the probability $(1-P_f)$, respectively. For the parameter P_f, it controls the tradeoff between the indicator-based fitness value and the degree of constraint violation. In this paper, P_f is set to 0.3. Once the ranking procedure stops, the top N individuals are put into $\mathcal{P}_{t+1}$.

3) Combining An Indicator-Based MOEA with ε Constrained Method: The combination of an indicator-based MOEA and ε constrained method is also simple since ε constrained method can be regarded as an extension of FR [38]. As shown in **Algorithm 4**, first, ε is employed to divide $\mathcal{U}_t$ into two sets: $\mathcal{S}_1 = \{\mathbf{u}_i \in \mathcal{U}_t | CV(\mathbf{u}_i) \leq \varepsilon\}$ and $\mathcal{S}_2 = \{\mathbf{u}_i \in \mathcal{U}_t | CV(\mathbf{u}_i) > \varepsilon\}$. From Section II-C, we can find that in ε constrained method, the individuals in $\mathcal{S}_1$ are better than the individuals in $\mathcal{S}_2$; thus, we prefer to select the individuals in $\mathcal{S}_1$. To be specific, two cases are under our consideration:

- If $|\mathcal{S}_1| \geq N$, then the environmental selection of the original indicator-based MOEA is conducted to select N individuals out of $\mathcal{S}_1$ ignoring all constraints. For these selected individuals, they are sorted according to their indicator-based fitness values and subsequently added to $\mathcal{P}_{t+1}$.
- If $|\mathcal{S}_1| < N$, then all the individuals in $\mathcal{S}_1$ are sorted according to their indicator-based fitness values and added to $\mathcal{P}_{t+1}$. For the rest $(N - |\mathcal{S}_1|)$ individuals, they are selected from $\mathcal{S}_2$. Herein, we sort the individuals in $\mathcal{S}_2$ according to their degree of constraint violation, and then put the top $(N - |\mathcal{S}_1|)$ individuals in the sorted $\mathcal{S}_2$ into the end of $\mathcal{P}_{t+1}$.

D. Discussions

- By combining FR, SR, and ε constrained method with HypE following the above approaches, we can obtain HypE-FR, HypE-SR, and HypE-ε, respectively. Similarly, we can obtain IBEA-FR, IBEA-SR, and IBEA-ε, and ISDE-FR, ISDE-SR, and ISDE-ε by integrating these three constraint-handling techniques with IBEA and ISDE, respectively. As a whole, nine indicator-based CMOEAs are developed in this study.
- It is evident that the above three kinds of extensions do not cost significant computational resources. In **Algorithm 2**, **Algorithm 3**, and **Algorithm 4**, the additional computational time complexity is $O(NlogN)$, $O(N^2)$ and $O(NlogN)$, respectively. Considering that the computational time complexity of the original HypE, IBEA, and ISDE is equal to or higher than $O(N^2)$, the additional computational time complexity caused by these three kinds of extensions is acceptable.

IV. Experimental Setup

A. Test Instances and Performance Metrics

Our experiments were conducted on 19 widely used constrained multiobjective optimization test functions. For test functions 1-8, they came from the famous CTPs [51]. Test functions 9-16 were collected from real-world applications: BNH [52], CONSTR [53], disc brake design (DBD) [54], OSY [55], speed reducer design (SRD) [56], SRN [57], TNK [58], and welded beam design (WBD) [54]. With respect to test functions 17-19, they were selected from the well-known CDTLZ test suite [59] with three objective functions. Overall, test functions 1-16 are CMOPs with two objective functions, and test functions 17-19 are CMOPs with three objective functions.

To compare the performance of different algorithms, two metrics were employed in our experiments:

- *Inverted Generational Distance (IGD) [60]:* In this paper, only the feasible solutions in the final population are used to compute IGD. Suppose that $\mathcal{IP}$ is the set of images of the feasible solutions in the final population, and $\mathcal{IP}^*$ is a set of uniformly distributed points on the constrained Pareto front. Then, IGD is calculated as:

$$IGD(\mathcal{IP}) = \frac{1}{|\mathcal{IP}^*|} \sum_{z^* \in \mathcal{IP}^*} distance(z^*, \mathcal{IP}) \quad (16)$$

  where $distance(z^*, \mathcal{IP})$ denotes the minimum Euclidean distance between z^* and all members in $\mathcal{IP}$, and $|\mathcal{IP}^*|$ is the cardinality of $\mathcal{IP}^*$. In general, a small IGD value is desired for a CMOEA.
- *Hypervolume (HV) [61]:* Similarly, when computing HV, we only consider the feasible solutions in the final population. In principle, HV is able to evaluate both convergence and diversity of $\mathcal{IP}$. The details of the calculation of HV have been introduced in Section II-B1. Note that the larger the HV value, the better the performance of a CMOEA. In our experiments, HV is computed by employing the reference point which is set to 1.1 times of the upper bounds of the constrained Pareto front.

B. Algorithms for Comparison

The following five state-of-the-art CMOEAs were chosen for performance comparison.

- *NSGA-II-CDP [25]:* NSGA-II-CDP is a combination of NSGA-II [25] and constraint-domination principle (CDP) for solving CMOPs. CDP is extended from FR, in which feasible solutions are better than infeasible solutions; for two infeasible solutions, the one with smaller degree of constraint violation is preferred; and the comparison of two feasible solutions is based on Pareto dominance.
- *A-NSGA-III [59]:* A-NSGA-III is modified from the constrained NSGA-III by adaptively updating the new reference points. Compared with the original constrained NSGA-III, A-NSGA-III can obtain a denser representation of the constrained Pareto front with an identical computational effort.
- *MOEA/D-ACDP [31]:* MOEA/D-ACDP is a recently proposed CMOEA, in which an angle-based constrained dominance principle (ACDP) is incorporated into the framework of MOEA/D for handling CMOPs.
- *C-MOEA/DD [62]:* C-MOEA/DD is an extension of MOEA/DD [62] for addressing constrained multi- and many-objective optimization problems. C-MOEA/DD puts more emphasis on feasible solutions than infeasible solutions by modifying the updating and reproduction procedures of MOEA/DD.
- *C-AnD [35]:* C-AnD is a combination of AnD and FR. In C-AnD, the union population is divided into a feasible solution set and an infeasible solution set. If the number of individuals in the feasible solution set is larger than N, the original AnD is employed to select N individuals. Otherwise, the union population is sorted according to the degree of constraint violation, and the top N individuals survive into the next generation.

C. Parameter Settings

- *Population Size:* The population size was set to 100 for each algorithm on each test function [25].
- *Parameter Settings for Evolutionary Operators:* For all algorithms, according to the suggestion in [25], the crossover probability and the mutation probability were set to 1.0 and $1/n$, respectively, and the distribution indexes of both SBX and the polynomial mutation were set to 20.
- *Number of Independent Runs and Termination Condition:* All algorithms were independently run 30 times on each test function, and terminated when 50,000 function evaluations (FEs) were reached [25], [28].
- *Parameter Settings for Algorithms:* The number of sample points in HypE-FR, HypE-SR, and HypE-ε was set to 10,000 following the suggestion in [63]. For C-MOEA/DD [62], the neighborhood size was set to 20, penalty parameter θ was set to 5, and probability δ was set to 0.9.

In this paper, all the experiments were implemented on the platform developed by Tian *et al.* [63].

TABLE I
PERFORMANCE COMPARISON AMONG THE NINE INDICATOR-BASED CMOEAS IN TERMS OF THE AVERAGE IGD VALUE ON THE 19 TEST FUNCTIONS. THE BEST AND SECOND BEST AVERAGE IGD VALUES AMONG ALL THE ALGORITHMS ON EACH TEST FUNCTION ARE HIGHLIGHTED IN GRAY AND LIGHT GRAY, RESPECTIVELY.

Problem	HypE-FR	HypE-SR	HypE-ε	IBEA-FR	IBEA-SR	IBEA-ε	ISDE-FR	ISDE-SR	ISDE-ε
CTP1	$1.60e-02$	$2.34e-01$	$4.07e-03$	$1.28e-01$	$2.24e-01$	$1.14e-01$	$1.83e-01$	$2.28e-01$	$1.74e-01$
CTP2	$1.92e-03$	$1.56e-01$	$1.88e-03$	$5.63e-03$	$1.15e-01$	$3.17e-03$	$2.54e-02$	$2.51e-02$	$2.25e-02$
CTP3	$1.76e-02$	$2.16e-01$	$2.15e-02$	$1.48e-01$	$4.06e-01$	$9.62e-02$	$8.03e-02$	$9.42e-02$	$6.39e-02$
CTP4	$1.25e-01$	$4.45e-01$	$1.13e-01$	$4.86e-01$	$4.76e-01$	$4.37e-01$	$1.71e-01$	$1.56e-01$	$1.62e-01$
CTP5	$7.48e-03$	$9.88e-02$	$5.96e-03$	$4.21e-02$	$1.42e-01$	$3.70e-02$	$3.24e-02$	$2.60e-02$	$3.11e-02$
CTP6	$8.35e-03$	$1.08e-01$	$8.30e-03$	$1.25e-02$	$9.25e-01$	$1.24e-02$	$4.87e-02$	$4.37e-02$	$4.75e-02$
CTP7	$1.17e-03$	$1.97e-03$	$1.17e-03$	$3.79e-03$	$2.12e-02$	$3.35e-03$	$1.83e-02$	$1.79e-02$	$1.76e-02$
CTP8	$8.36e-02$	$2.89e-01$	$3.38e-02$	$1.27e-01$	$8.60e-01$	$7.10e-02$	$4.10e-02$	$1.11e-01$	$4.19e-02$
BNH	$2.90e-01$	$2.90e-01$	$2.89e-01$	$2.96e-01$	$4.02e-01$	$2.98e-01$	$9.48e-01$	$9.81e-01$	$9.30e-01$
CONSTR	$1.52e-02$	$1.41e+00$	$1.52e-02$	$3.04e-02$	$6.88e-01$	$2.97e-02$	$1.03e-01$	$1.40e-01$	$1.21e-01$
DBD	$8.13e-02$	$5.47e-01$	$3.30e-01$	$4.38e-01$	$6.25e-01$	$4.28e-01$	$7.19e-01$	$7.57e-01$	$6.61e-01$
OSY	$1.56e+00$	$3.73e+01$	$5.27e+00$	$4.20e+00$	$1.19e+02$	$8.85e+00$	$8.86e+00$	$1.08e+01$	$1.19e+01$
SRD	$2.20e+01$	$4.51e+02$	$2.84e+02$	$1.54e+03$	$1.56e+02$	$1.71e+03$	$6.45e+02$	$6.46e+02$	$6.42e+02$
SRN	$7.96e-01$	$9.78e-01$	$7.94e-01$	$8.09e-01$	$1.43e+00$	$8.10e-01$	$2.90e+00$	$3.25e+00$	$3.00e+00$
TNK	$3.26e-03$	$8.75e-02$	$3.21e-03$	$2.79e-02$	$3.12e-01$	$2.84e-02$	$2.11e-02$	$8.10e-02$	$2.06e-02$
WBD	$1.62e-01$	$4.43e-01$	$1.60e-01$	$1.02e+01$	$1.39e+01$	$1.14e+01$	$9.22e+00$	$8.63e+00$	$9.22e+00$
C1-DTLZ1	$3.20e-02$	$4.86e-02$	$3.05e-02$	$2.59e-02$	$3.66e-02$	$2.64e-02$	$3.75e-02$	$3.87e-02$	$3.76e-02$
C2-DTLZ2	$9.27e-02$	$1.05e-01$	$9.11e-02$	$7.79e-02$	$1.27e-01$	$7.83e-02$	$1.93e-01$	$1.89e-01$	$1.86e-01$
C3-DTLZ4	$2.76e-01$	$5.73e-01$	$2.86e-01$	$1.58e-01$	$4.60e-01$	$1.11e-01$	$1.98e-01$	$2.13e-01$	$2.04e-01$

TABLE II
PERFORMANCE COMPARISON AMONG THE NINE INDICATOR-BASED CMOEAS IN TERMS OF THE AVERAGE HV VALUE ON THE 19 TEST FUNCTIONS. THE BEST AND SECOND BEST AVERAGE HV VALUES AMONG ALL THE ALGORITHMS ON EACH TEST FUNCTION ARE HIGHLIGHTED IN GRAY AND LIGHT GRAY, RESPECTIVELY.

Problem	HypE-FR	HypE-SR	HypE-ε	IBEA-FR	IBEA-SR	IBEA-ε	ISDE-FR	ISDE-SR	ISDE-ε
CTP1	$4.57e-01$	$3.76e-01$	$4.61e-01$	$4.19e-01$	$3.77e-01$	$4.24e-01$	$3.92e-01$	$3.74e-01$	$3.95e-01$
CTP2	$1.14e-01$	$7.26e-02$	$1.14e-01$	$1.12e-01$	$8.43e-02$	$1.13e-01$	$9.90e-02$	$1.00e-01$	$1.01e-01$
CTP3	$4.67e-01$	$3.29e-01$	$4.64e-01$	$3.83e-01$	$2.19e-01$	$4.15e-01$	$4.04e-01$	$4.01e-01$	$4.16e-01$
CTP4	$3.42e-01$	$1.45e-01$	$3.50e-01$	$1.37e-01$	$1.41e-01$	$1.63e-01$	$3.01e-01$	$3.16e-01$	$3.12e-01$
CTP5	$4.78e-01$	$2.39e-01$	$4.80e-01$	$4.09e-01$	$1.74e-01$	$4.31e-01$	$4.22e-01$	$4.19e-01$	$4.28e-01$
CTP6	$2.08e+00$	$1.83e+00$	$2.08e+00$	$2.08e+00$	$1.05e+00$	$2.08e+00$	$1.98e+00$	$1.99e+00$	$1.98e+00$
CTP7	$7.17e-01$	$7.04e-01$	$7.17e-01$	$7.16e-01$	$6.81e-01$	$7.17e-01$	$6.80e-01$	$6.80e-01$	$6.80e-01$
CTP8	$1.32e+00$	$1.06e+00$	$1.34e+00$	$1.29e+00$	$7.16e-01$	$1.32e+00$	$1.30e+00$	$1.26e+00$	$1.30e+00$
BNH	$2.85e+03$	$2.85e+03$	$2.85e+03$	$2.85e+03$	$2.84e+03$	$2.85e+03$	$2.80e+03$	$2.80e+03$	$2.80e+03$
CONSTR	$5.24e+00$	$3.78e+00$	$5.24e+00$	$5.21e+00$	$4.42e+00$	$5.22e+00$	$5.09e+00$	$5.02e+00$	$5.07e+00$
DBD	$4.31e+01$	$4.25e+01$	$4.30e+01$	$4.28e+01$	$4.24e+01$	$4.28e+01$	$4.22e+01$	$4.22e+01$	$4.23e+01$
OSY	$1.40e+04$	$9.10e+03$	$1.35e+04$	$1.37e+04$	$1.52e+03$	$1.32e+04$	$1.33e+04$	$1.32e+04$	$1.30e+04$
SRD	$2.58e+06$	$2.53e+06$	$2.57e+06$	$6.74e+05$	$2.50e+06$	$3.61e+05$	$2.53e+06$	$2.51e+06$	$2.53e+06$
SRN	$3.06e+04$	$3.04e+04$	$3.06e+04$	$3.06e+04$	$3.02e+04$	$3.06e+04$	$2.98e+04$	$2.98e+04$	$2.98e+04$
TNK	$5.25e-01$	$4.45e-01$	$5.25e-01$	$5.19e-01$	$2.44e-01$	$5.19e-01$	$4.98e-01$	$4.52e-01$	$5.01e-01$
WBD	$9.76e-01$	$9.69e-01$	$9.76e-01$	$8.95e-01$	$5.43e-01$	$8.71e-01$	$9.12e-01$	$9.24e-01$	$9.17e-01$
C1-DTLZ1	$1.33e-01$	$1.22e-01$	$1.34e-01$	$1.38e-01$	$1.32e-01$	$1.38e-01$	$1.29e-01$	$1.29e-01$	$1.29e-01$
C2-DTLZ2	$6.61e-01$	$6.53e-01$	$6.63e-01$	$6.95e-01$	$5.79e-01$	$6.95e-01$	$4.81e-01$	$4.72e-01$	$4.75e-01$
C3-DTLZ4	$7.65e+00$	$6.68e+00$	$7.73e+00$	$8.32e+00$	$7.39e+00$	$8.50e+00$	$7.90e+00$	$7.89e+00$	$7.88e+00$

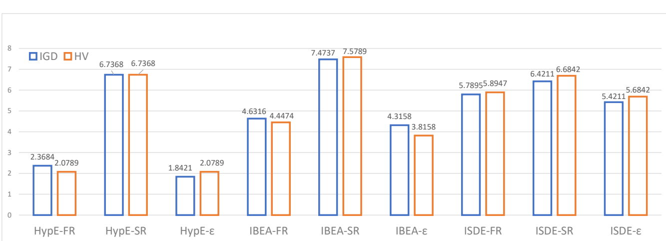


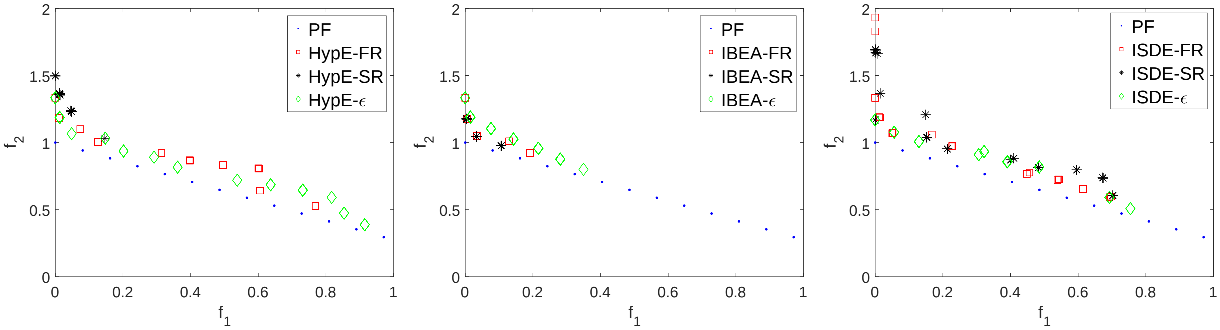
Fig. 1. Friedman's test among the nine indicator-based CMOEAs (i.e., HypE-FR, HypE-SR, HypE-ε, IBEA-FR, IBEA-SR, IBEA-ε, ISDE-FR, ISDE-SR, and ISDE-ε) on the 19 test functions in terms of IGD and HV. The smaller the ranking, the better the performance of an algorithm.

V. EXPERIMENTAL RESULTS AND DISCUSSIONS

A. Comparison among Indicator-Based CMOEAs

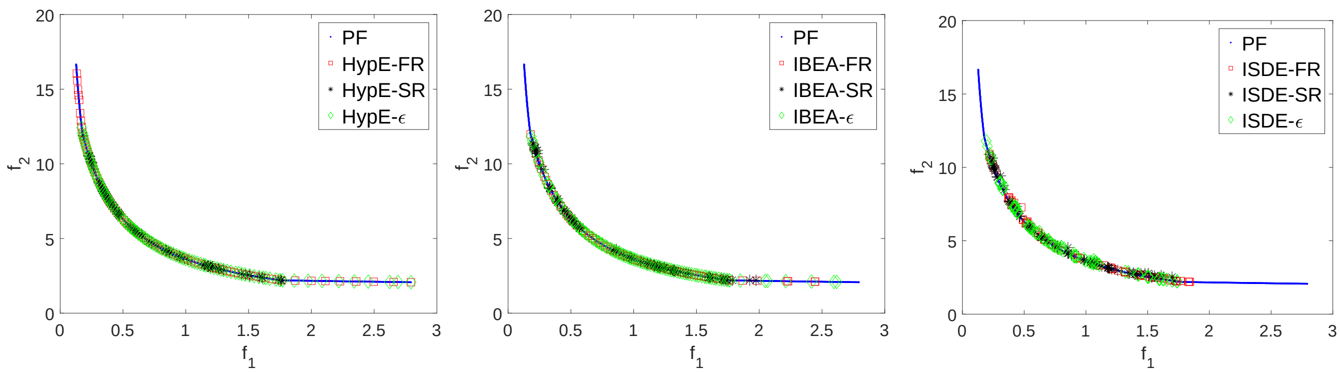
Firstly, we are interested in identifying the properties of the proposed nine indicator-based CMOEAs. To this end, we tested them on the 19 constrained multiobjective optimization test functions introduced in Section IV-A. The comparison results in terms of IGD and HV are presented in Table I and Table II, respectively. At our first glance, HypE-FR and HypE-ε can achieve the superior performance on CTPs and real-world CMOPs, while IBEA-FR and IBEA-ε can obtain better results on the CDTLZ problems. Next, we will give the detailed discussions.

1) CTPs: From Tables I and II, we can observe that HypE-ε achieves the best overall performance in terms of both IGD



(a) HypE-based CMOEAs (b) IBEA-based CMOEAs (c) ISDE-based CMOEAs

Fig. 2. Images of the feasible solutions provided by HypE, IBEA, and ISDE with different constraint-handing techniques in a run on CTP4.



(a) HypE-based CMOEAs (b) IBEA-based CMOEAs (c) ISDE-based CMOEAs

Fig. 3. Images of the feasible solutions provided by HypE, IBEA, and ISDE with different constraint-handing techniques in a run on DBD.

and HV, followed by HypE-FR. For HypE-ε, it obtains the best and second best results on seven instances (i.e., CTP1, CTP2, CTP4, CTP5, CTP6, CTP7, and CTP8) and one instance (i.e., CTP3), respectively. As for HypE-FR, it performs the best on one instance (i.e., CTP3) with respect to both IGD and HV, and performs the second best on six instances (i.e., CTP1, CTP2, CTP4, CTP5, CTP6, and CTP7) and seven instances (i.e., CTP1, CTP2, CTP4, CTP5, CTP6, CTP7, and CTP8) in terms of IGD and HV, respectively. The competitive performance of these two algorithms can be largely attributed to the utilization of the hypervolume indicator, which is the only indicator strictly monotonic with regard to Pareto dominance [33]. For another hypervolume-based CMOEA, i.e., HypE-SR, it cannot obtain promising results. The reason might be that the usage of SR will keep some infeasible solutions with good indicator-based fitness values in the final population. However, these infeasible solutions will be deleted before the calculation of IGD or HV. Due to the less feasible solutions, HypE-SR provides worse IGD and HV values, compared with HypE-FR and HypE-ε. For other algorithms, ISDE-FR reaches the second best performance on CTP8 in terms of IGD, while IBEA-FR, IBEA-SR, IBEA-ε, ISDE-SR, and ISDE-ε fail to achieve any of the best or second best results on these CTPs.

2) Real-World CMOPs: On these eight real-world CMOPs, again, HypE-FR and HypE-ε can obtain the superior performance. Regarding IGD, HypE-FR can obtain the best performance on DBD, OSY, and SRD, and achieve the second best performance on BNH, CONSTR, SRN, TNK and WBD; while HypE-ε can obtain the best performance on BND, CONSTR, SRN, TNK, and DBD, and the second best performance on DBD and SRN. With respect to HV, HypE-FR performs the best on BNH, CONSTR, DBD, OSY, SRD, SRN, and TNK, and performs the second best on DBD; while HypE-ε performs the best on WBD, and performs the second best on BHN, CONSTR, DBD, SRD, SRN, and TNK. The above phenomena again verify the effectiveness of the hypervolume indicator for coping with CMOPs with two objective functions. It is also observed that HypE-SR still cannot obtain promising results on these real-world CMOPs, which might be attributed to the maintenance of infeasible solutions in the final population as analyzed before. As for other algorithms, IBEA-FR can attain the second best performance on OSY regarding both IGD and HV, while IBEA-SR, IBEA-ε, ISDE-FR, ISDE-SR, and ISDE-ε are unable to obtain any of the best or second best results on these eight real-world CMOPs.

3) CDTLZ Problems: Interestingly, HypE-FR and HypE-ε cannot show their superiorities on the CDTLZ problems. It is probably because the Monte Carlo method is used for hypervolume estimation, when these two algorithms are used for handing CMOPs with three objective functions. Owing to

TABLE III
PERFORMANCE COMPARISON BETWEEN HYPE-FR AND FIVE STATE-OF-THE-ART CMOEAS IN TERMS OF THE AVERAGE AND STANDARD DEVIATION OF THE IGD VALUES ON THE 19 TEST FUNCTIONS. THE BEST AVERAGE IGD VALUE AMONG ALL THE ALGORITHMS ON EACH TEST FUNCTION WAS HIGHLIGHTED IN BOLDFACE.

Problem	NSGA-II-CDP	A-NSGA-III	MOEA/D-ACDP	C-MOEA/DD	C-AnD	HypE-FR
CTP1	8.1067e-2 (8.10e-2)−	1.0747e-1 (8.72e-2)−	1.6081e-2 (1.39e-2)−	**1.3388e-2 (2.18e-3)**+	1.3651e-1 (4.14e-2)−	1.5984e-2 (4.18e-2)
CTP2	2.2526e-3 (3.28e-4)−	2.8563e-3 (4.31e-4)−	5.2737e-3 (1.28e-3)−	9.3850e-3 (3.00e-3)−	4.1252e-2 (2.59e-2)−	**1.9208e-3 (2.32e-4)**
CTP3	6.2697e-2 (8.03e-2)−	3.2048e-2 (4.58e-2)−	2.9864e-2 (3.74e-3)−	4.8233e-2 (5.12e-3)−	1.4240e-1 (4.31e-2)−	**1.7568e-2 (8.46e-3)**
CTP4	2.1376e-1 (1.51e-1)−	2.0740e-1 (1.39e-1)−	1.6414e-1 (3.35e-2)−	**1.2083e-1 (1.48e-2)** ≈	2.7084e-1 (8.84e-2)−	1.2545e-1 (4.31e-2)
CTP5	9.8773e-3 (4.70e-3)≈	7.4848e-3 (2.40e-3)≈	2.0178e-2 (8.94e-3)−	1.1432e-2 (2.64e-3)−	2.5913e-2 (1.31e-2)−	**7.4821e-3 (2.84e-3)**
CTP6	1.1617e-2 (3.90e-4)−	1.7224e-2 (9.99e-3)−	1.8301e-2 (8.40e-4)−	2.1604e-2 (2.50e-3)−	1.3003e-1 (4.18e-2)−	**8.3547e-3 (1.58e-4)**
CTP7	1.7321e-3 (1.41e-3)−	1.9550e-3 (1.43e-3)−	3.3652e-3 (3.21e-5)−	4.0297e-3 (2.05e-3)−	1.2341e-1 (3.41e-2)−	**1.1661e-3 (7.20e-6)**
CTP8	1.4255e-1 (1.46e-1)−	1.7405e-1 (1.48e-1)−	**3.6822e-2 (6.14e-2)**+	1.8420e-1 (9.19e-1)−	1.0016e-1 (7.34e-2)−	8.3572e-2 (1.33e-1)
BNH	4.3484e-1 (2.99e-2)−	5.2000e-1 (1.21e-1)−	2.7511e-1 (1.26e-4)+	**2.6638e-1 (4.66e-4)**+	4.6801e-1 (5.40e-2)−	2.8967e-1 (4.69e-3)
CONSTR	2.0453e-2 (6.60e-4)−	2.3775e-2 (2.01e-3)−	4.3999e-2 (6.37e-4)−	6.1968e-2 (9.87e-3)−	3.4766e-2 (2.60e-3)−	**1.5192e-2 (9.93e-5)**
DBD	8.3875e-2 (4.88e-2)≈	4.0564e-1 (1.05e-1)−	1.9603e-1 (6.47e-2)−	4.6390e-1 (2.46e-2)−	3.2154e-1 (1.04e-1)−	**8.1280e-2 (4.08e-2)**
OSY	4.0696e+0 (9.21e+0)−	3.1830e+0 (1.66e+0)−	2.7214e+1 (1.37e+1)−	1.0626e+2 (9.15e+0)−	3.3951e+0 (1.00e+0)−	**1.5645e+0 (9.28e-1)**
SRD	**1.1373e+1 (1.48e+0)**+	3.5192e+2 (1.96e+2)−	2.6468e+2 (4.68e+1)−	5.6208e+1 (1.24e+1)−	5.9861e+2 (1.20e+2)−	2.1964e+1 (2.93e+0)
SRN	1.0904e+0 (6.34e-2)−	**7.6610e-1 (4.78e-3)**+	2.2303e+0 (1.23e-2)−	7.1229e+1 (1.19e+1)−	1.0036e+0 (5.29e-2)−	7.9605e-1 (5.64e-3)
TNK	4.3557e-3 (1.58e-4)−	4.1239e-3 (2.84e-4)−	6.1697e-3 (4.96e-4)−	2.9281e-2 (5.81e-3)−	9.6892e-3 (1.02e-3)−	**3.2624e-3 (1.36e-4)**
WBD	1.8295e-1 (1.19e-2)−	4.3149e-1 (3.54e-2)−	7.3427e+0 (2.15e-1)−	1.5704e+1 (8.19e-1)−	7.2724e-1 (1.67e-1)−	**1.6199e-1 (6.25e-3)**
C1-DTLZ1	2.6927e-2 (1.47e-3)+	2.2035e-2 (1.46e-3)+	2.1989e-2 (2.52e-4)+	**2.0371e-2 (1.57e-4)**+	2.1616e-2 (3.50e-4)+	3.2020e-2 (4.23e-3)
C2-DTLZ2	5.6755e-2 (3.58e-3)+	**4.4552e-2 (4.62e-4)**+	5.1819e-2 (3.09e-4)+	4.9736e-2 (4.42e-4)+	4.6135e-2 (1.47e-3)+	9.2727e-2 (5.15e-3)
C3-DTLZ4	1.7464e-1 (1.82e-1)+	1.5406e-1 (1.87e-1)+	2.0711e-1 (2.53e-1)+	1.8416e-1 (2.96e-1)+	**1.0716e-1 (2.81e-3)**+	2.7576e-1 (1.98e-1)
+	4	4	5	5	3	/
−	13	14	14	13	16	/
≈	2	1	0	1	0	/

the existence of various constraints, the Monte Carlo method might lose its effectiveness in the approximation process. Instead, IBEA-FR and IBEA-ε can obtain quite competitive performance on these three CDTLZ problems in terms of both IGD and HV. Specifically, IBEA-FR can obtain the best performance on C1-DTLZ1 and C2-DTLZ2, and obtain the second best performance on C3-DTLZ4; while IBEA-ε performs the best on C3-DTLZ4, and perform the second best on C1-DTLZ1 and C2-DTLZ2. The excellent performance of IBEA-FR and IBEA-ε suggests that the $I_{(\epsilon)+}$ indictor is an effective indicator for CMOPs with three objective functions.

4) Discussions: From the above experiments, we can conclude that both indicator-based MOEAs and constraint-handling techniques play very important roles in the performance of indicator-based CMOEAs. It is also observed that there does not exist an indicator-based CMOEA which can beat all other indicator-based CMOEAs on each test function. In general, HypE-FR and HypE-ε are highly suggested for two-objective CMOPs, while IBEA-FR and IBEA-ε are recommended for solving CMOPs with three objective functions.

To analyze the overall performance of these nine indicator-based CMOEAs, the Friedman's test was implemented on the 19 test functions as shown in Fig. 1. From the results of the Friedman's test, it is evident that HypE-FR and HypE-ε are the two best algorithms in terms of both IGD and HV, followed by IBEA-ε and IBEA-FR. However, for indicator-based CMOEAs with the usage of the I_{SDE} indicator (i.e., ISDE-FR, ISDE-SR, and ISDE-ε) or SR (i.e., HypE-SR, IBEA-SR, and ISDE-SR), they cannot obtain any promising results in terms of either IGD or HV. The above phenomena suggest that the I_{SDE} indicator is not suitable for solving CMOPs with two and three objective functions, even its potential has been demonstrated on solving unconstrained many-objective optimization problems [35], [44], [45]. The failure of SR, as discussed above, might be due to its balance mechanism, which would keep a certain proportion of infeasible solutions in the final population. Thus, we can also conclude that not all constraint-handling techniques in constrained single-objective optimization are effective in solving CMOPs. Actually, how to select proper constraint-handling techniques and indicator-based MOEAs, and how to combine them together are still open issues worthy of in-depth studies.

The images of the feasible solutions provided by these nine indicator-based CMOEAs in the end of a run are plotted in Fig. 2, Fig. 3, and Fig. S-1 in the supplementary file on CTP4, DBD, and C1-DTLZ1, respectively.

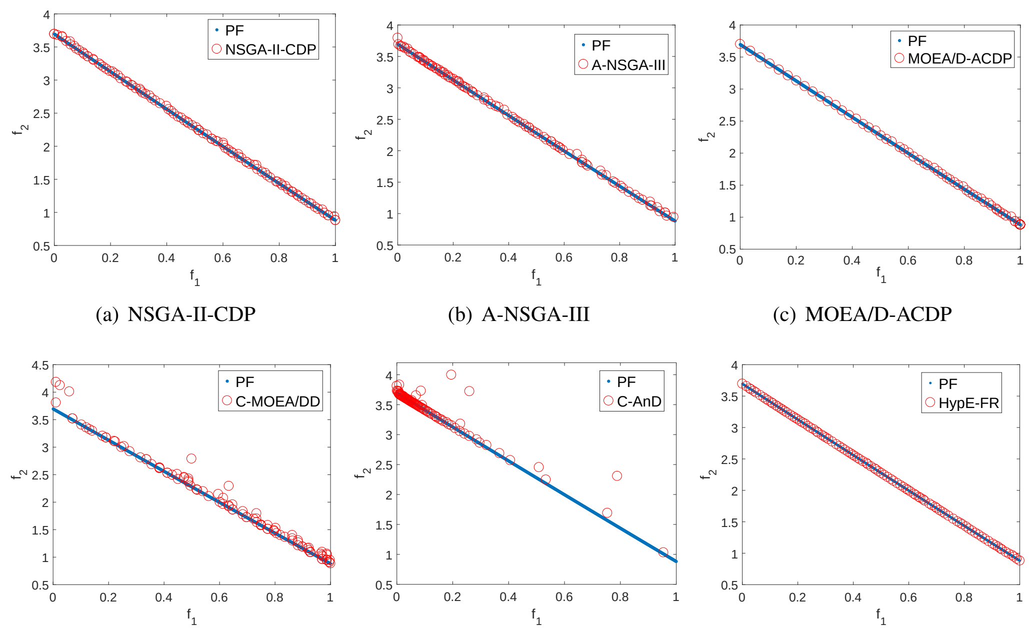
B. Comparison between HypE-FR and Five State-of-the-Art CMOEAs

Subsequently, we compared the performance of a superior indicator-based CMOEA (i.e., HypE-FR) with that of five state-of-the-art CMOEAs (i.e., NSGA-II-CDP, A-NSGA-III, MOEA/D-ACDP, C-MOEA/DD, and C-AnD) on the 19 test functions. The Wilcoxon's rank-sum test at a 0.05 significance level was implemented between HypE-FR and each of NSGA-II-CDP, A-NSGA-III, MOEA/D-ACDP, C-MOEA/DD, and C-AnD on each test function. The comparison results are summarized in Table III and Table IV in terms of IGD and HV, respectively. The detailed discussions are presented in the following.

It is clear that HypE-FR can achieve the best overall performance in terms of both IGD and HV. Regarding IGD, HypE-FR outperforms NSGA-II-CDP, A-NSGA-III, MOEA/D-ACDP, C-MOEA/DD, and C-AnD on 13, 14, 14, 13, and 16 instances, respectively, while loses on four, four, five, five, and three instances, respectively. As far as HV is concerned, HypE-FR obtains better performance than its competitors on more than 14 instances, while provides worse results on no more than four instances.

TABLE IV
PERFORMANCE COMPARISON BETWEEN HYPE-FR AND FIVE STATE-OF-THE-ART CMOEAS IN TERMS OF THE AVERAGE AND STANDARD DEVIATION OF THE HV VALUES ON THE 19 TEST FUNCTIONS. THE BEST AVERAGE HV VALUE AMONG ALL THE ALGORITHMS ON EACH TEST FUNCTION WAS HIGHLIGHTED IN BOLDFACE.

Problem	NSGA-II-CDP	A-NSGA-III	MOEA/D-ACDP	C-MOEA/DD	C-AnD	HypE-FR
CTP1	4.3399e-1 (2.89e-2)−	4.2309e-1 (3.18e-2)−	4.5695e-1 (3.61e-3)−	4.5246e-1 (2.10e-3)−	3.8056e-1 (1.96e-2)−	**4.5736e-1 (1.51e-2)**
CTP2	**1.1380e-1 (1.86e-4)**≈	1.1321e-1 (3.42e-4)−	1.1202e-1 (1.14e-3)−	1.0824e-1 (2.45e-3)−	9.9489e-2 (7.76e-3)−	1.1376e-1 (2.15e-4)
CTP3	4.3692e-1 (5.20e-2)−	4.5644e-1 (3.12e-2)−	4.5282e-1 (3.82e-3)−	4.2969e-1 (5.84e-3)−	3.2098e-1 (3.44e-2)−	**4.6729e-1 (6.52e-3)**
CTP4	2.8591e-1 (9.50e-2)−	2.9121e-1 (8.13e-2)−	2.8046e-1 (2.38e-2)−	3.3712e-1 (1.95e-2)≈	1.9875e-1 (4.86e-2)−	**3.4214e-1 (2.92e-2)**
CTP5	4.3634e-1 (5.82e-2)−	4.7074e-1 (1.45e-2)−	4.7065e-1 (3.70e-3)−	4.4256e-1 (7.44e-3)−	3.2207e-1 (4.09e-2)−	**4.7760e-1 (1.88e-2)**
CTP6	2.0714e+0 (1.26e-3)−	2.0575e+0 (2.04e-2)−	2.0579e+0 (2.66e-3)−	2.0365e+0 (8.45e-3)−	1.8196e+0 (6.12e-2)−	**2.0826e+0 (1.10e-3)**
CTP7	7.1672e-1 (2.82e-3)−	7.1532e-1 (2.74e-3)−	7.1601e-1 (5.89e-4)−	7.0463e-1 (5.17e-3)−	5.6434e-1 (3.57e-2)−	**7.1739e-1 (5.96e-5)**
CTP8	1.2765e+0 (9.06e-2)−	1.2536e+0 (8.94e-2)−	**1.3186e+0 (7.64e-2)**+	1.2975e+0 (2.45e-1)−	1.2039e+0 (4.50e-2)−	1.3156e+0 (8.18e-2)
BNH	2.8408e+3 (2.09e+0)−	2.8318e+3 (1.10e+1)−	2.8536e+3 (2.85e-2)−	2.8531e+3 (1.04e-1)−	2.8375e+3 (3.96e+0)−	**2.8538e+3 (5.02e-1)**
CONSTR	5.2259e+0 (1.87e-3)−	5.2220e+0 (5.05e-3)−	5.2020e+0 (1.63e-3)−	5.1243e+0 (2.03e-2)−	5.2088e+0 (4.16e-3)−	**5.2423e+0 (4.39e-4)**
DBD	4.3019e+1 (2.24e-2)−	4.2794e+1 (8.81e-2)−	4.2955e+1 (4.47e-2)−	4.2694e+1 (3.81e-2)−	4.2854e+1 (7.03e-2)−	**4.3099e+1 (2.43e-2)**
OSY	1.3592e+4 (1.22e+3)−	1.3809e+4 (2.04e+2)−	1.0343e+4 (2.00e+3)−	2.0892e+3 (9.76e+2)−	1.3875e+4 (1.43e+2)−	**1.3953e+4 (1.35e+2)**
SRD	2.5766e+6 (3.52e+2)−	2.5707e+6 (3.72e+3)−	2.5720e+6 (1.55e+3)−	2.5722e+6 (1.08e+3)−	2.5698e+6 (1.67e+3)−	**2.5780e+6 (2.14e+1)**
SRN	3.0481e+4 (1.67e+1)−	3.0607e+4 (4.55e+0)−	3.0158e+4 (1.03e+1)−	8.8632e+3 (3.42e+3)−	3.0477e+4 (2.99e+1)−	**3.0616e+4 (1.81e+0)**
TNK	5.2377e-1 (2.28e-4)−	5.2374e-1 (5.02e-4)−	5.2279e-1 (2.67e-4)−	4.9262e-1 (7.57e-3)−	5.1812e-1 (1.14e-3)−	**5.2535e-1 (2.46e-4)**
WBD	9.7470e-1 (1.96e-4)−	9.7354e-1 (4.40e-4)−	2.7555e-1 (2.54e-2)−	2.9751e-1 (1.91e-1)−	9.7033e-1 (1.10e-3)−	**9.7578e-1 (3.36e-4)**
C1-DTLZ1	1.3606e-1 (1.26e-3)+	1.3738e-1 (2.27e-3)+	1.3742e-1 (1.16e-3)+	**1.3904e-1 (6.84e-4)**+	1.3843e-1 (7.95e-4)+	1.3287e-1 (1.92e-3)
C2-DTLZ2	6.5074e-1 (7.05e-3)−	6.8091e-1 (5.08e-3)+	6.7060e-1 (5.88e-4)+	6.7260e-1 (2.65e-3)+	**6.8577e-1 (1.55e-3)**+	6.6149e-1 (4.45e-3)
C3-DTLZ4	7.9926e+0 (6.21e-1)≈	8.1796e+0 (6.86e-1)+	8.0787e+0 (9.62e-1)+	7.9206e+0 (1.31e+0)≈	**8.3528e+0 (1.74e-2)**+	7.6524e+0 (8.96e-1)
+	1	3	4	2	3	/
−	16	16	15	15	16	/
≈	2	0	0	2	0	/



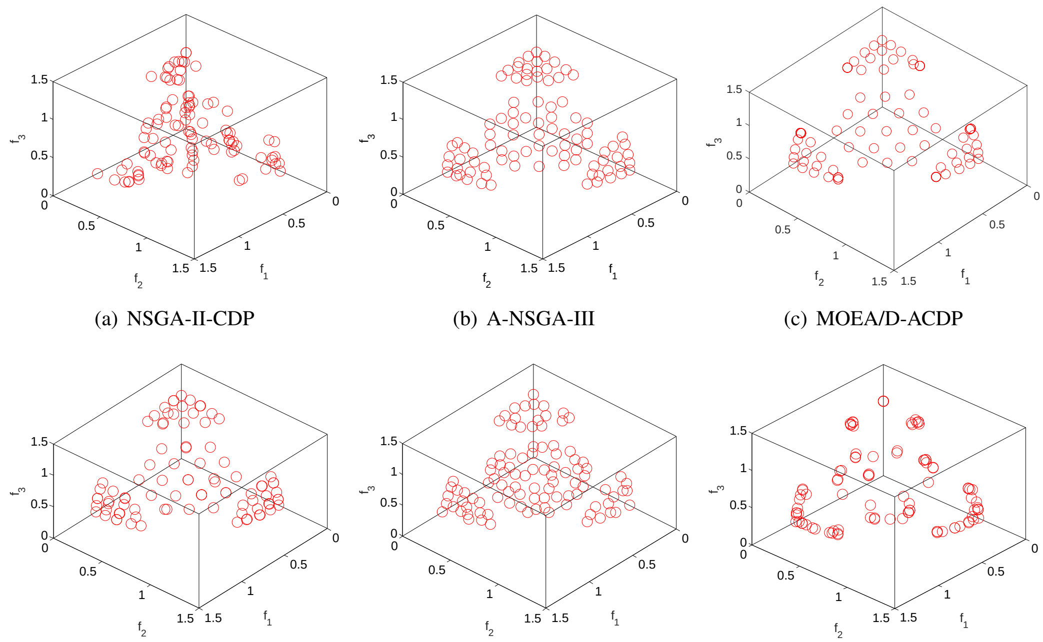
(a) NSGA-II-CDP (b) A-NSGA-III (c) MOEA/D-ACDP

(d) C-MOEA/DD (e) C-AnD (f) HypE-FR

Fig. 4. Images of the feasible solutions provided by NSGA-II-CDP, A-NSGA-III, MOEA/D-ACDP, C-MOEA/DD, C-AnD, and HypE-FR in a run on CTP6.

It can be seen that the advantage of HypE-FR focuses mainly on solving two-objective CMOPs (i.e., CTPs and real-world CMOPs). Specifically, for CTPs, HypE-FR achieves the similar or better performance on all instances in terms of both IGD and HV compared with its competitors, except for C-MOEA/DD on CTP1 (with respect to IGD) and MOEA/D-ACDP on CTP8 (regarding both IGD and HV). For real-world CMOPs, HypE-FR outperforms its competitors on most instances in terms of both IGD and HV, and loses on no more than one instance with respect to IGD. However, for the CDTLZ problems, HypE-FR does not show its advantage again. Instead, C-AnD can obtain the best overall performance on these three CDTLZ problems, followed by C-MOEA/DD and A-NSGA-III. The superiority of C-AnD may be attributed to its angle-based selection and shift-based density estimation, which can maintain the diversity of search directions and delete poor individuals in a reasonable way. For C-MOEA/DD and A-NSGA-III, their competitive performance demonstrates the potential of using reference vectors or reference points to guide the search in constrained multiobjective optimization.

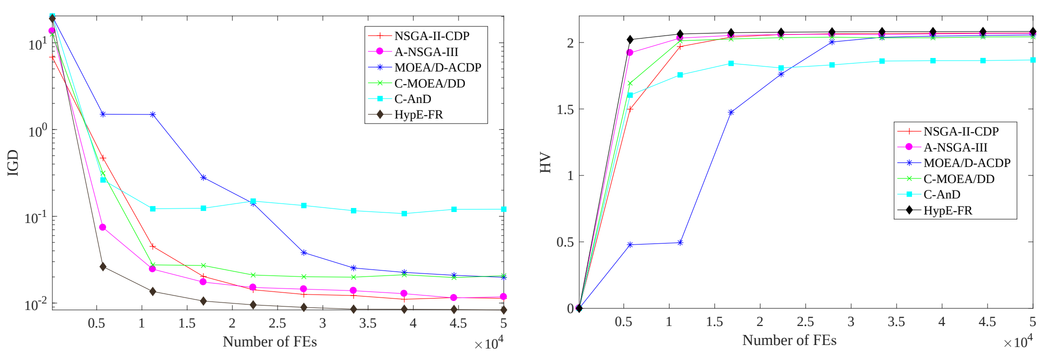
In summary, the excellent performance of HypE-FR verifies that indicator-based CMOEAs are worth extensively and inten-



(a) NSGA-II-CDP (b) A-NSGA-III (c) MOEA/D-ACDP

(d) C-MOEA/DD (e) C-AnD (f) HypE-FR

Fig. 5. Images of the feasible solutions provided by NSGA-II-CDP, A-NSGA-III, MOEA/D-ACDP, C-MOEA/DD, C-AnD, and HypE-FR in a run on C2-DTLZ2.



(a) IGD (b) HV

Fig. 6. Convergence graphs of NSGA-II-CDP, A-NSGA-III, MOEA/D-ACDP, C-MOEA/DD, C-AnD, and HypE on CTP6 in terms of IGD and HV.

sively studying in the future. For an engineering application, HypE-FR is highly recommended for solving two-objective CMOPs, while C-AnD is suggested for dealing with CMOPs with three objective functions. The images of the feasible solutions provided by these six CMOEAs in the end of a run are plotted in Fig. 4 and Fig. 5 on CTP6 and C2-DTLZ2, respectively. Besides, the convergence graphs of these six compared algorithms on CTP6 are depicted in Fig. 6.

Remark 4: In the supplementary file, we also investigated the effect of constraint-handling techniques in indicator-based CMOEAs in Section S-I-A, and conducted the parameter sensitivity analysis in Section S-I-B.

VI. Conclusion

In this paper, we made an attempt to extend indicator-based MOEAs for solving CMOPs. Firstly, we discussed the rationality and superiority behind the combination of indicator-based MOEAs and constraint-handling techniques. Subsequently, we proposed an indicator-based CMOEA framework which can combine indicator-based MOEAs with constraint-handling techniques in an easy way. Based on this proposed framework, nine indicator-based CMOEAs were developed in this study. The performance of these nine indicator-based CMOEAs was tested on 19 widely used constrained multiobjective optimization test functions. The comparison results revealed that the indicator-based CMOEAs' performance relies both on indicator-based MOEAs and constraint-handling techniques, and there does not exist an indicator-based CMOEA which can beat all other indicator-based CMOEAs on each test function. Some practical suggestions were given on how to select appropriate indicator-based CMOEAs for addressing different CMOPs. We also compared the performance of a superior indicator-based CMOEA (i.e., HypE-FR) with that of five state-of-the-art CMOEAs on the above 19 test functions. The experimental results demonstrated that HypE can achieve the best overall performance with respect to both the IGD and HV metrics. It was thus concluded that the research on indicator-based CMOEAs deserves much attention from the researchers in the community of evolutionary computation for further promoting the development of evolutionary constrained multiobjective optimization.

REFERENCES

[1] X. L. Zheng and L. Wang, "A collaborative multiobjective fruit fly optimization algorithm for the resource constrained unrelated parallel machine green scheduling problem," *IEEE Transactions on Systems Man and Cybernetics: Systems*, vol. 48, no. 5, pp. 790–800, 2018.

[2] W. Lakhdhar, R. Mzid, M. Khalgui, Z. Li, G. Frey, and A. Al-Ahmari, "Multiobjective optimization approach for a portable development of reconfigurable real-time systems: From specification to implementation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 3, pp. 623–637, 2019.

[3] C. H. Chen and J. H. Chou, "Multiobjective optimization of airline crew roster recovery problems under disruption conditions," *IEEE Transactions on Systems Man and Cybernetics: Systems*, vol. 47, no. 1, pp. 133–144, 2017.

[4] S. Wang and H. Chiang, "Constrained multiobjective nonlinear optimization: A user preference enabling method," *IEEE Transactions on Cybernetics*, 2019, in press. DOI: 10.1109/TCYB.2018.2833281.

[5] J. Wang, W. Ren, Z. Zhang, H. Huang, and Y. Zhou, "A hybrid multiobjective memetic algorithm for multiobjective periodic vehicle routing problem with time windows," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019, in press, DOI: 10.1109/TSMC.2018.2861879.

[6] Z. Liu and Y. Wang, "Handling constrained multiobjective optimization problems with constraints in both the decision and objective spaces," *IEEE Transactions on Evolutionary Computation*, 2019, in press. DOI: 10.1109/TEVC.2019.2894743.

[7] Z. Ma and Y. Wang, "Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons," *IEEE Transactions on Evolutionary Computation*, 2019, in press. DOI: 10.1109/TEVC.2019.2896967.

[8] Z. Fan, W. Li, X. Cai, H. Han, F. Yi, Y. You, J. Mo, C. Wei, and E. Goodman, "An improved epsilon constraint-handling method in MOEA/D for CMOPs with large infeasible regions," *Soft Computing*, 2019, in press. DOI: https://doi.org/10.1007/s00500-019-03794-x.

[9] Z. Fan, W. Li, X. Cai, H. Li, K. Hu, Q. Zhang, K. Deb, and E. D. Goodman, "Difficulty adjustable and scalable constrained multi-objective test problem toolkit," *Evolutionary Computation*, 2019, in press. DOI: https://doi.org/10.1162/evco_a_00259.

[10] Y. Zhou, M. Zhu, J. Wang, Z. Zhang, Y. Xiang, and J. Zhang, "Tri-goal evolution framework for constrained many-objective optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019, in press, DOI: 10.1109/TSMC.2018.2858843.

[11] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.

[12] C. A. Coello Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 11-12, pp. 1245–1287, 2002.

[13] Q. Kang, X. Song, M. Zhou, and L. Li, "A collaborative resource allocation strategy for decomposition-based multiobjective evolutionary algorithms," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019, in press, DOI: 10.1109/TSMC.2018.2818175.

[14] Y. R. Naidu and A. K. Ojha, "Solving multiobjective optimization problems using hybrid cooperative invasive weed optimization with multiple populations," *IEEE Transactions on Systems Man and Cybernetics: Systems*, vol. 48, no. 6, pp. 821–832, 2018.

[15] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired co-evolutionary algorithms for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 4, pp. 474–494, 2012.

[16] R. Wang, Z. Zhou, H. Ishibuchi, T. Liao, and T. Zhang, "Localized weighted sum method for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, 2017, in press, DOI: 10.1109/TEVC.2016.2611642.

[17] K. Li, R. Wang, T. Zhang, and H. Ishibuchi, "Evolutionary many-objective optimization: A comparative study of the state-of-the-art," *IEEE Access*, vol. 6, pp. 26 194–26 214, 2018.

[18] B. Wang, H. Li, Q. Zhang, and Y. Wang, "Decomposition-based multi-objective optimization for constrained evolutionary optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019, in press, DOI: 10.1109/TSMC.2018.2876335.

[19] Y. Wang, B. Wang, H. Li, and G. G. Yen, "Incorporating objective function information into the feasibility rule for constrained evolutionary optimization," *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 2938–2952, 2016.

[20] Y. Wang and Z. Cai, "Constrained evolutionary optimization by means of (μ+λ)-differential evolution and improved adaptive trade-off model," *Evolutionary Computation*, vol. 19, no. 2, pp. 249–285, 2011.

[21] ——, "Combining multiobjective optimization with differential evolution to solve constrained optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, pp. 117–134, 2012.

[22] ——, "A dynamic hybrid framework for constrained evolutionary optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 1, pp. 203–217, 2012.

[23] S. Zeng, R. Jiao, C. Li, and R. Wang, "Constrained optimisation by solving equivalent dynamic loosely-constrained multiobjective optimisation problem," *International Journal of Bio-Inspired Computation*, vol. 13, no. 2, pp. 86–101, 2019.

[24] S. Zeng, R. Jiao, C. Li, X. Li, and J. S. Alkasassbeh, "A general framework of dynamic constrained multiobjective evolutionary algorithms for constrained optimization," *IEEE transactions on cybernetics*, vol. 47, no. 9, pp. 2678–2688, 2017.

[25] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[26] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[27] Z. Fan, Y. Fang, W. Li, J. Lu, X. Cai, and C. Wei, "A comparative study of constrained multi-objective evolutionary algorithms on constrained multi-objective optimization problems," in *2017 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2017, pp. 209–216.

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

[29] Z. Fan, H. Li, C. Wei, W. Li, H. Huang, X. Cai, and Z. Cai, "An improved epsilon constraint handling method embedded in moea/d for constrained multi-objective optimization problems," *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–8, Dec 2016.

[30] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, and E. D. Goodman, "Push and pull search for solving constrained multi-objective optimization problems," *Swarm and Evolutionary Computation*, vol. 44, no. 2, pp. 665–679, 2019.

[31] 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," *Applied Soft Computing*, vol. 74, no. 1, pp. 621–633, 2019.

[32] S. Jiang, J. Zhang, Y. S. Ong, A. N. Zhang, and P. S. Tan, "A simple and fast hypervolume indicator-based multiobjective evolutionary algorithm," *IEEE Transactions on Cybernetics*, vol. 45, no. 10, pp. 2202–2213, Oct 2015.

[33] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.

[34] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2004, pp. 832–842.

[35] Z.-Z. Liu, Y. Wang, and P.-Q. Huang, "A many-objective evolutionary algorithm with angle-based selection and shift-based density estimation," *Information Sciences*, 2019, in press, DOI: https://doi.org/10.1016/j.ins.2018.06.063.

[36] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2, pp. 311–338, 2000.

[37] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 284–294, Sep 2000.
[38] T. Takahama and S. Sakai, "Constrained optimization by the ϵ constrained differential evolution with an archive and gradient-based mutation," in *IEEE Congress on Evolutionary Computation*, July 2010, pp. 1–9.
[39] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Computing Surveys (CSUR)*, vol. 48, no. 1, p. 13, 2015.
[40] M. Emmerich, N. Beume, and B. Naujoks, "An EMO algorithm using the hypervolume measure as selection criterion." in *EMO*, vol. 3410. Springer, 2005, pp. 62–76.
[41] I. Christian, H. Nikolaus, and R. Stefan, "Covariance matrix adaptation for multi-objective optimization," *Evolutionary Computation*, vol. 15, no. 1, pp. 1–28, 2007.
[42] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization," *Proceedings of Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems, EUROGEN'2001*, pp. 95–100, 2001.
[43] H. Wang, L. Jiao, and X. Yao, "Two_Arch2: An improved two-archive algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 524–541, 2015.
[44] B. Li, K. Tang, J. Li, and X. Yao, "Stochastic ranking algorithm for many-objective optimization based on multiple indicators," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 924–938, 2016.
[45] M. Li, S. Yang, and X. Liu, "Shift-based density estimation for pareto-based algorithms in many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 348–365, 2014.
[46] T. Pamulapati, R. Mallipeddi, and P. N. Suganthan, "ISDE+ -— an indicator for multi and many-objective optimization," *IEEE Transactions on Evolutionary Computation*, 2019, in press, DOI: 10.1109/TEVC.2018.2848921.
[47] T. Takahama and S. Sakai, "Constrained optimization by ε constrained particle swarm optimizer with ε-level control," in *Soft Computing as Transdisciplinary Science and Technology*. Springer, 2005, pp. 1019–1029.
[48] B.-C. Wang, H.-X. Li, J.-P. Li, and Y. Wang, "Composite differential evolution for constrained evolutionary optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 7, pp. 1482–1495, 2019.
[49] E. Mezura-Montes and C. A. Coello Coello, "Constraint-handling in nature-inspired numerical optimization: Past, present and future," *Swarm and Evolutionary Computation*, vol. 1, no. 4, pp. 173–194, 2011.
[50] C. Segura, C. A. Coello Coello, G. Miranda, and C. Len, "Using multi-objective evolutionary algorithms for single-objective constrained and unconstrained optimization," *Annals of Operations Research*, vol. 240, no. 1, pp. 217–250, 2016.
[51] K. Deb, A. Pratap, and T. Meyarivan, "Constrained test problems for multi-objective evolutionary optimization," in *First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001)*, 2000, pp. 284–298.
[52] T. T. Binh and U. Korn, "MOBES: A multiobjective evolution strategy for constrained optimization problems," in *The Third International Conference on Genetic Algorithms (Mendel 97)*, vol. 25, 1997, p. 27.
[53] P. D. Justesen, "Multi-objective optimization using evolutionary algorithms," *University of Aarhus, Department of Computer Science, Denmark*, 2009.
[54] T. Ray and K. M. Liew, "A swarm metaphor for multiobjective design optimization," *Engineering Optimization*, vol. 34, no. 2, pp. 141–153, 2002.
[55] A. Osyczka and S. Kundu, "A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm," *Structural Optimization*, vol. 10, no. 2, pp. 94–99, Oct 1995.
[56] C. A. Coello Coello and G. T. Pulido, "Multiobjective structural optimization using a microgenetic algorithm," *Structural and Multidisciplinary Optimization*, vol. 30, no. 5, pp. 388–403, 2005.
[57] N. Srinivas and K. Deb, "Multiobjective function optimization using nondominated sorting genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 2, no. 3, pp. 1301–1308, 1994.
[58] M. Tanaka, H. Watanabe, Y. Furukawa, and T. Tanino, "GA-based decision support system for multicriteria optimization," in *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*, vol. 2, Oct 1995, pp. 1556–1561 vol.2.
[59] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, Part II: Handling constraints and extending to an adaptive approach," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2014.
[60] C. A. Coello Coello, G. B. Lamont, D. A. Van Veldhuizen *et al.*, *Evolutionary Algorithms for Solving Multi-objective Problems*. Springer, 2007, vol. 5.
[61] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms – A comparative case study," in *International Conference on Parallel Problem Solving from Nature*. Springer, 1998, pp. 292–301.
[62] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 694–716, 2015.
[63] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A matlab platform for evolutionary multi-objective optimization [educational forum]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.

Zhi-Zhong Liu received the B.S. degree in automation and the Ph.D. degree in control science and engineering from the Central South University, Changsha, China, in 2013 and 2019, respectively. His current research interests include evolutionary computation, bioinformatics, swarm intelligence, constrained optimization, and multimodal optimization.

Yong Wang (M'08–SM'17) received the Ph.D. degree in control science and engineering from the Central South University, Changsha, China, in 2011.

He is a Professor with the School of Automation, Central South University, Changsha, China. His current research interests include the theory, algorithm design, and interdisciplinary applications of computational intelligence.

Dr. Wang is an Associate Editor for the *Swarm and Evolutionary Computation*. He was a Web of Science highly cited researcher in Computer Science in 2017 and 2018.

Bing-Chuan Wang Bing-Chuan Wang received the B.S. degree in automation and the M.S. degree in control science and engineering from Central South University, Changsha, China, in 2013 and 2016, respectively, and the Ph.D. degree in system engineering and engineering management from City University of Hong Kong, Hong Kong, in 2019.

He is an Associate Professor with the School of Automation, Central South University, Changsha, China. His current research interests include evolutionary computation, constrained optimization, and intelligent modeling.

Supplementary File for "Indicator-Based Constrained Multiobjective Evolutionary Algorithms"

S-I. Additional Results and Discussions

A. Effect of Constraint-handling Techniques in Indicator-based CMOEAs

In this subsection, we investigated the influence of different constraint-handling techniques in indicator-based CMOEAs. To this end, we considered three HypE-based CMOEAs (i.e., HypE-FR, HypE-SR, and HypE-ε). We tested these three algorithms on CTP1 and recorded the feasible solutions provided by them during the evolution in Fig. S-2, Fig. S-3, and Fig. S-4, respectively.

From Fig. S-2 and Fig. S-4, it is observed that the feasible solutions obtained by HypE-ε have a better distribution than the feasible solutions obtained by HypE-FR at the early stage of evolution (i.e. the 50th generation). The reason might be that compared with FR which prefers feasible solutions, in ε constrained method, the constrained boundary can be relaxed to some extent (i.e., up to ε), which helps HypE-ε maintain some infeasible solutions with good objective function values and low constraint violations. It should be noted that the maintenance of this kind of infeasible solutions is very necessary, since they can help HypE-ε approach the PF from diverse search directions. As a result, HypE-ε can obtain a more widely-distributed approximation set than HypE-FR. As for HypE-SR, it is interesting to observe from Fig. S-3 that as the evolution continues, the obtained feasible solutions are biased toward a relatively small area. The reason might be that for CTP1, only the top one-third portion of the original unconstrained Pareto-optimal region is feasible, while the other two-third region of the constrained Pareto-optimal region comes from the constraints [1]. Note that in HypE-SR, some infeasible solutions with good objective function values might be retained which affect the assignment of the fitness values for feasible solutions. An example is presented in Fig. S-5. Due to the existence of C, A (lying near the top-one third portion of PF) will be preferred to B (lying near the bottom two-third of PF) during the environmental selection of HypE-SR. As a result, feasible solutions are biased toward the top one-third portion of PF as shown in Fig. S-3.

From the above discussion, we can conclude that not all constraint-handling techniques in constrained single-objective optimization are effective in solving CMOPs. There is still a lot of work to be done in the future on how to select appropriate constraint-handling techniques and indicator-based MOEAs, and how to combine them together.

B. Parameter Sensitivity Analysis

For ε constrained method and SR, they are not parameter-free methods. Parameter p in ε constrained method and parameter P_f in SR can control the extent that the information of objective function is utilized, which is of essential importance in balancing constraints and objective function.

First of all, we investigated the sensitivity of p in ε constrained method. We selected HypE-ε as the instance algorithm and ran it with six different values of p: $p = 0$, $p = 0.2$, $p = 0.4$, $p = 0.6$, $p = 0.8$, and $p = 0.99$ [1] over 30 independent runs on CTPs. The average IGD values are shown in Fig. S-6. From Fig. S-6, it is observed that HypE-ε with $p = 0.2$ can achieve the best performance in most instances (i.e. CTP1, CTP2, CTP4, CPT5, CTP6, CTP7, and CTP8). Hence, in this paper, p was set to 0.2.

Similarly, we studied the sensitivity of P_f in SR. Herein, we chose IBEA-SR as the instance algorithm and ran it with six different values of P_f: $P_f = 0$, $P_f = 0.1$, $P_f = 0.2$, $P_f = 0.3$, $P_f = 0.4$, and $P_f = 0.5$ over 30 independent runs on CTPs [2]. The average IGD values are shown in Fig. S-7. From Fig. S-7, we can find that for CTP1, CTP3, CTP4, CTP5, and CTP8, better results are obtained when P_f is close to 0.3. As for CTP2, CTP6, and CTP7, the best results are achieved when P_f is close to 0.4, 0.1, and 0.5, respectively. In general, $P_f = 0.3$ is a good choice and is recommended in this paper.

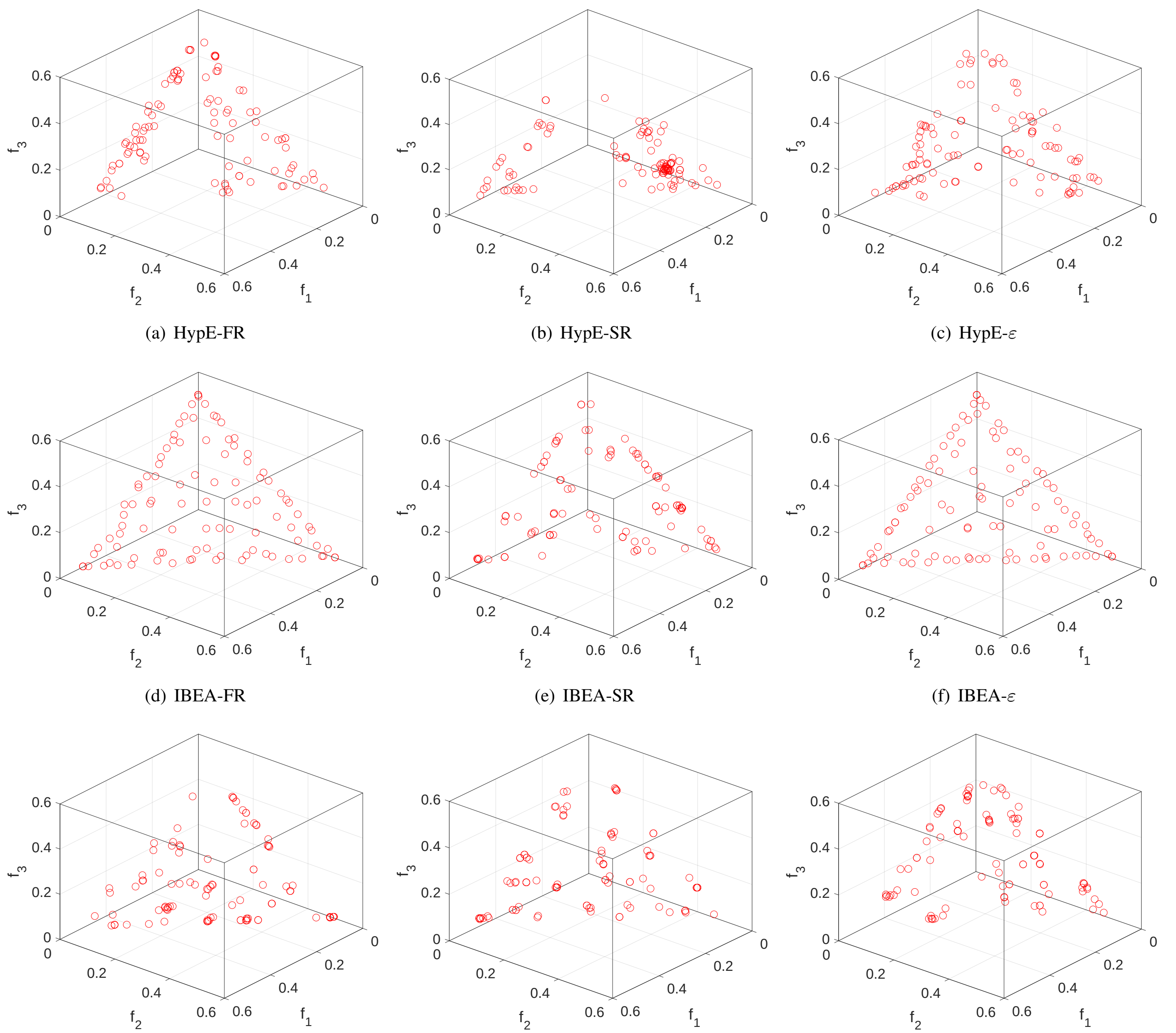
References

[1] K. Deb, A. Pratap, and T. Meyarivan, "Constrained test problems for multi-objective evolutionary optimization," in *First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001)*, 2000, pp. 284–298.

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

[1] From Eq. (14), we can find that the value of p cannot be equal to 1.

[2] According to [2], the value of P_f should not be larger than 0.5. It is claimed in [2] that a minor bias toward constraints encourages the evolution of feasible solutions while still maintaining infeasible regions as potential bridges to move among feasible regions in the whole search space.

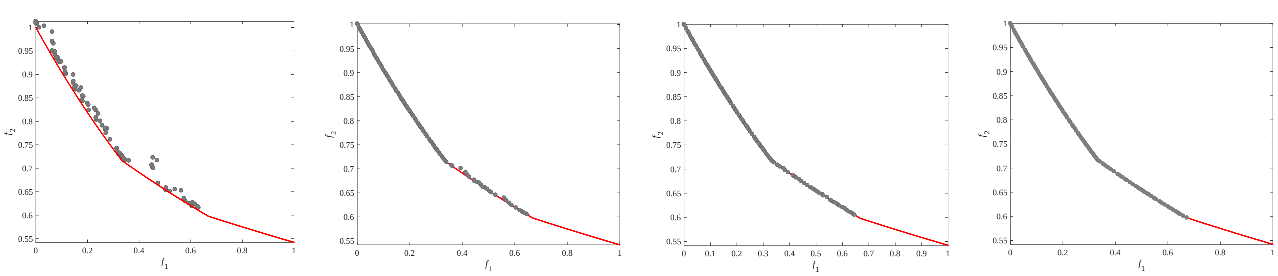


(a) HypE-FR (b) HypE-SR (c) HypE-ε

(d) IBEA-FR (e) IBEA-SR (f) IBEA-ε

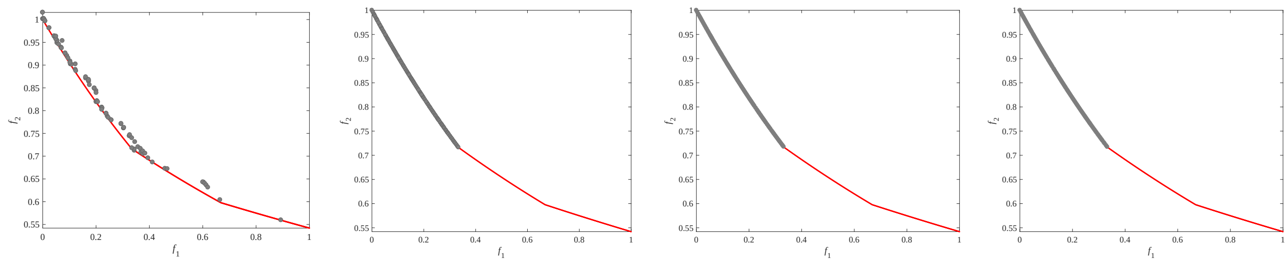
(g) ISDE-FR (h) ISDE-SR (i) ISDE-ε

Fig. S-1. Images of the feasible solutions provided by HypE-FR, HypE-SR, HypE-ε, IBEA-FR, IBEA-SR, IBEA-ε, ISDE-FR, ISDE-SR, and ISDE-ε in a run on C1-DTLZ1.



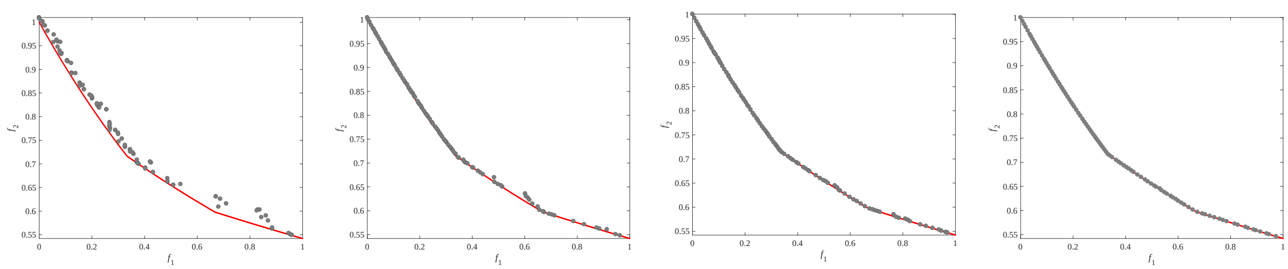
(a) 50 generations (b) 100 generations (c) 200 generations (d) 500 generations

Fig. S-2. Images of the feasible solutions provided by HypE-FR during the evolution on CTP1.



(a) 50 generations (b) 100 generations (c) 200 generations (d) 500 generations

Fig. S-3. Images of the feasible solutions provided by HypE-SR during the evolution on CTP1.



(a) 50 generations (b) 100 generations (c) 200 generations (d) 500 generations

Fig. S-4. Images of the feasible solutions provided by HypE-ε during the evolution on CTP1.

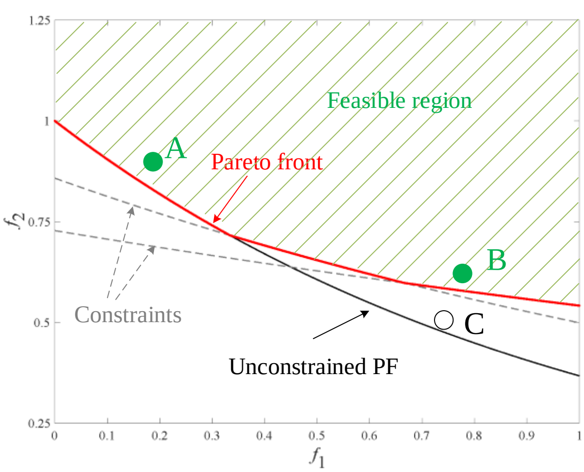


Fig. S-5. Illustration of the effect of infeasible solutions in HypE-SR. A and B are feasible solutions while C is an infeasible solution.

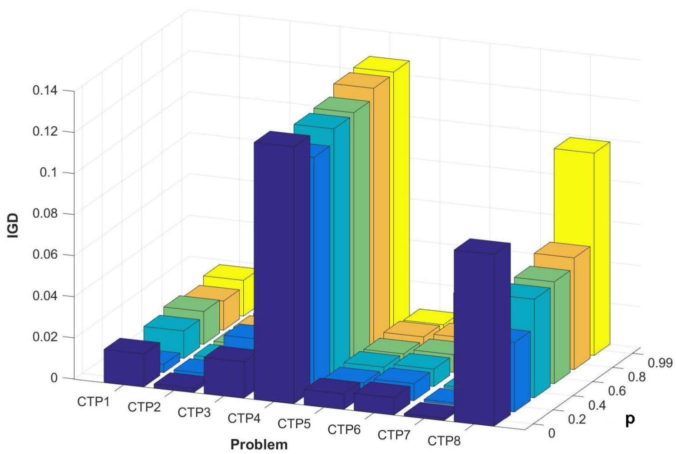


Fig. S-6. Average IGD values obtained by HypE-ε with different δ values over 30 runs in CTPs

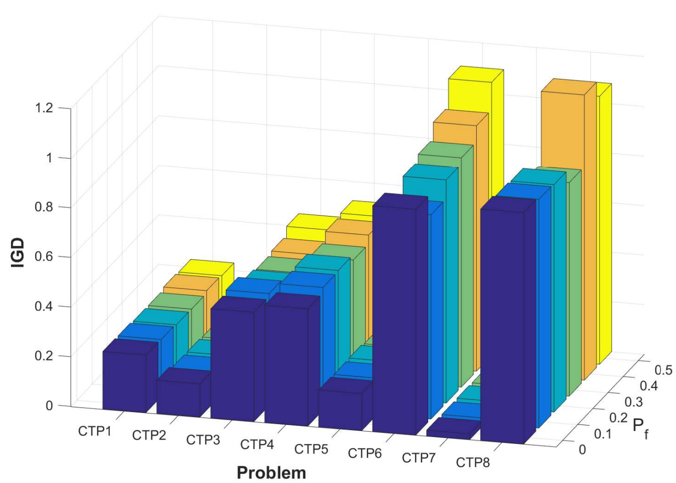


Fig. S-7. Average IGD values obtained by IBEA-SR with different P_f values over 30 runs in CTPs