

# A Novel Evolutionary Algorithm for Dynamic Constrained Multiobjective Optimization Problems

Qingda Chen, *Student Member, IEEE*, Jinliang Ding, *Senior Member, IEEE*,

Shengxiang Yang, *Senior Member, IEEE*, and Tianyou Chai, *Fellow, IEEE*

**Abstract**—To promote research on dynamic constrained multiobjective optimization, we first propose a group of generic test problems with challenging characteristics, including different modes of the true Pareto front (e.g., convexity–concavity and connectedness–disconnectedness) and the changing feasible region. Subsequently, motivated by the challenges presented by dynamism and constraints, we design a dynamic constrained multiobjective optimization algorithm with a nondominated solution selection operator, a mating selection strategy, a population selection operator, a change detection method, and a change response strategy. The designed nondominated solution selection operator can obtain a nondominated population with diversity when the environment changes. The mating selection strategy and population selection operator can adaptively handle infeasible solutions. If a change is detected, the proposed change response strategy reuses some portion of the old solutions in combination with randomly generated solutions to reinitialize the population, and a steady-state update method is designed to improve the retained previous solutions. Experimental results show that the proposed test problems can be used to clearly distinguish the performance of algorithms, and that the proposed algorithm is very competitive for solving dynamic constrained multiobjective optimization problems in comparison with state-of-the-art algorithms.

**Index Terms**—Dynamic constrained multiobjective optimization, test problems, population selection, change response

## I. INTRODUCTION

ANY real-world optimization problems, particularly online optimization problems or optimal control problems, involve multiple objectives and constraints that may change over time throughout the optimization [1]; such problems can be called dynamic constrained multiobjective optimization problems (DCMOPs). Without loss of generality,

This work was supported in part by the National Natural Science Foundation of China under Grant 61988101, Grant 61525302, Grant 61590922, and Grant 61673331; in part by the National Key Research and Development Program of China under Grant 2018YFB1701104; and in part by the Xingliao Plan of Liaoning Province under Grant XLYC1808001. (*Corresponding author: Jinliang Ding.*)

Q. Chen, J. Ding, and T. Chai are with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang, 110819, China (e-mail: cq0309@126.com; jlding@mail.neu.edu.cn; tychai@mail.neu.edu.cn).

S. Yang is with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang, 110819, China, and also with the Centre for Computational Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester, LE1 9BH, U. K. (syang@dmu.ac.uk).

This paper has supplementary material.

a DCMOP can be mathematically formulated as follows:

$$\begin{aligned} \min \quad & F(\mathbf{x}, t) = (f_1(\mathbf{x}, t), \dots, f_m(\mathbf{x}, t), \dots, f_M(\mathbf{x}, t))^T \\ \text{s. t.} \quad & \begin{cases} h_k(\mathbf{x}, t) = 0, & k = 1, \dots, h \\ g_k(\mathbf{x}, t) \geq 0, & k = 1, \dots, g \\ \mathbf{x} = \{x_1, \dots, x_j, \dots, x_n\} \in \Omega \\ x_j^{\min} \leq x_j \leq x_j^{\max} \end{cases} \end{aligned} \quad (1)$$

where  $m$  is the objective function index, and  $M$  is the number of objective functions.  $j$  is the decision variable index, and  $n$  is the number of decision variables.  $\mathbf{x}$  is a solution that consists of  $n$  decision variables, and  $x_j$  is the  $j$ th decision variable, which is bounded by corresponding lower ( $x_j^{\min}$ ) and upper ( $x_j^{\max}$ ) bounds.  $\Omega$  is the decision space, and  $t$  represents a dynamic environment.  $h$  and  $g$  are the numbers of dynamic equality and inequality constraints, respectively.  $F(\mathbf{x}, t)$  is the objective function vector to be minimized at  $t$ , and  $f_m(\mathbf{x}, t)$  denotes the  $m$ th objective function.  $h_k(\mathbf{x}, t)$  and  $g_k(\mathbf{x}, t)$  represent the  $k$ th equality and inequality constraints, respectively, that vary with  $t$ .

If a solution satisfies all constraints, then it is called a feasible solution; otherwise, it is called an infeasible solution. For two feasible solutions  $\mathbf{x}_1$  and  $\mathbf{x}_2$  of a DCMOP at  $t$ ,  $\mathbf{x}_1$  is said to dominate  $\mathbf{x}_2$  if and only if  $f_m(\mathbf{x}_1, t) \leq f_m(\mathbf{x}_2, t)$  for every  $m \in \{1, \dots, M\}$  and  $f_b(\mathbf{x}_1, t) < f_b(\mathbf{x}_2, t)$  for at least one index  $b \in \{1, \dots, M\}$ . If there is no other solution in  $\Omega$  to dominate  $\mathbf{x}_1$ , then  $\mathbf{x}_1$  is a nondominated solution (i.e., a Pareto-optimal solution).  $F(\mathbf{x}_1, t)$  is then called a Pareto-optimal objective vector at  $t$ . The set of all Pareto-optimal solutions is called the Pareto-optimal set (POS). Similarly, the set of all Pareto-optimal objective vectors is called the Pareto-optimal front (POF).

The goal when solving a multiobjective optimization problem (MOP) is to obtain a set of nondominated solutions, all of which must be feasible. Compared with constrained MOPs without dynamism (CMOPs) and dynamic MOPs without constraints (DMOPs), DCMOPs are more challenging due to the simultaneous presence of constraints and dynamism. Specifically, a change is usually not observable and may even be unknown, and it may arise in either the objective functions or constraints. Detecting environmental changes is difficult if they are unforeseen in DCMOPs. The feasible region of objective functions (hereafter referred to simply as the feasible region) is affected by constraints and may increase or decrease with dynamism, changing the number of feasible solutions. Abandoning infeasible solutions may be beneficial for population convergence. However, some previous infeasible solutions close to the boundary of the feasible region can easily become feasible or even nondominated in new environments.

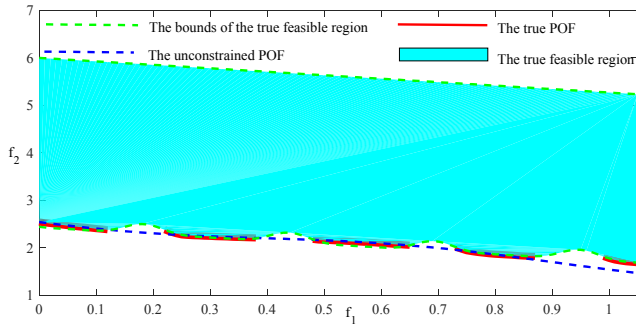


Fig. 1. The diagram of the true POF of a DCMOP.

Therefore, determining how to best handle infeasible solutions is a considerable challenge. For a DCMOP, the true POF of the objective functions (hereafter called the true POF) may be determined by the unconstrained POF (i.e., the POF of a DCMOP without constraints) and bounds of the feasible region. The true POF of a DCMOP may become disconnected when the lower bound of the feasible region is irregular. An example of the true POF of a DCMOP with two objectives (i.e.,  $f_1(\mathbf{x}, t)$  and  $f_2(\mathbf{x}, t)$ ) is given in Fig. 1, and its functions are given in Section S-I of the supplementary material. The disconnectedness of the true POF may cause the population to get stuck at local regions, decreasing the population diversity. Thus determining how to design the population update strategy that can quickly track the true POF and improve population diversity is a challenging task. Furthermore, dynamism and constraints can cause the true POF to move in an irregular manner. Hence, designing an effective change response strategy that can track the new POF is an arduous task.

In recent years, many contributions have been made regarding attempts to solve CMOPs and DMOPs in several important aspects, including benchmark problems [2]–[7], [63], performance metrics [2], [8]–[11], and algorithms [12]–[23]. It is worth noting that the moving peak benchmark problem designed in [63] transforms the single objective problem proposed in [56] into a two-objective problem by assigning a random objective value into each individual, which is a good attempt. However, many real-world problems are DCMOPs, and the current states of both benchmark problem and algorithm designs for DCMOPs are considerably less advanced compared to those for CMOPs and DMOPs [24]. Azzouz *et al.* [25] designed a set of test instances for DCMOPs and proposed a dynamic constrained algorithm to solve them. However, the true POSs in the designed test problems show only small variations between  $t$  and  $t+1$ , making it difficult to test the change response capability of an algorithm. Azzouz *et al.* [24] proposed a dynamic constrained multiobjective evolutionary algorithm (DC-MOEA) for solving the test problems designed in [25]. However, this algorithm does not update the retained previous feasible solutions when the environment changes, resulting in a slow tracking ability when the change in the true POS or POF between  $t$  and  $t+1$  is large.

To advance the state of research on DCMOPs, we design a set of test problems and propose a novel dynamic constrained multiobjective evolutionary algorithm (*d*CMOEA) to solve them. The main contributions of this work are as follows:

1) Following a few basic design principles proposed in [12] and [56], two important characteristics of DCMOPs are introduced into the test problems. Specifically, the true POF

in each proposed test problem is simultaneously determined by the unconstrained POF and bounds of the feasible region, causing the true POF to switch between different modes (e.g., connected and disconnected). This characteristic is useful for testing an algorithm’s convergence speed, reactivity, and rapid tracking capability with respect to the new true POF. The feasible region in each of the proposed test problems can increase or decrease with dynamism in an irregular manner, meaning that the number of infeasible solutions and even the true POF of a DCMOP can change. The second characteristic makes it possible to test an algorithm’s capability of handling infeasible solutions.

- 2) The nondominated solutions in DCMOPs usually vary with dynamism and constraints, and some previous nondominated solutions may become infeasible or dominated in a new environment. Based on this fact, we propose a nondominated solution selection operator that is suitable for DCMOPs. When a new environment arises, the proposed nondominated solution selection operator can reserve a part of nondominated solutions obtained in the previous environment, increasing the diversity of the nondominated set in a new environment.
- 3) For handling the infeasible solutions and improving population diversity, the constraint handling technique proposed in [15] is introduced into the designed mating selection strategy and population selection operator. Considering that the feasible region may vary with dynamism, the population selection operator integrates the information of the obtained feasible solutions. The proposed operator can adaptively update the population, balancing the trade-off between the population diversity and convergence capability of *d*CMOEA.
- 4) For a DCMOP in which the environmental changes are unforeseen, we propose a change detection strategy that can accurately detect dynamism arising in either objective functions or constraints. *d*CMOEA reinitializes a portion of the solutions and reuses some previous solutions with competitive performance (i.e., small objective function and constraint violation values) when the environment changes. Meanwhile, we exploit information collected from the reinitialized solutions and an approximate feasibility ratio to update the retained previous solutions, allowing the proposed algorithm to quickly track the new POF.

The remainder of this paper is organized as follows. Section II discusses some related work on the constraint handling techniques used for CMOPs and the change response strategies used for DMOPs. Section III presents the design of the test problems. Section IV describes the framework of the proposed *d*CMOEA, together with detailed descriptions of each component of the algorithm. Performance metrics and a comprehensive comparison of various multiobjective evolutionary algorithms (MOEAs) on the test problems are presented in Section V. Section VI offers a further discussion of the proposed algorithm. Section VII outlines some conclusions and suggests directions for future research.

## II. RELATED WORK

Considering that there are few works on solving DCMOPs, in this section, we first present some recent studies on constraint handling techniques for CMOPs and then discuss the

change response strategies adopted in DMOPs.

#### A. Constraint Handling Techniques

The constraint handling techniques for CMOPs can be divided into three categories [21]: techniques that place a higher priority on feasible solutions to survive to the next generation, techniques for balancing the trade-off between feasibility and convergence, and techniques for repairing infeasible solutions.

The first category is based on a preference for feasible solutions, which are considered better than infeasible ones. The constraint handling method proposed by Coello Coello and Christiansen [26] simply ignores infeasible solutions, which results in difficulty when solving problems with a narrow feasible region. Deb *et al.* [13] designed a constrained dominance relation in which a solution  $\mathbf{x}_1$  is said to dominate  $\mathbf{x}_2$  if 1) the former is feasible while the latter is not, 2) they are both infeasible, but  $\mathbf{x}_1$  has a smaller overall constraint violation, or 3) both of them are feasible and  $\mathbf{x}_1$  dominates  $\mathbf{x}_2$ . Following a similar idea, Cheng *et al.* [27] updated the population on the basis of the degree of constraint violation. Fonseca and Fleming [28] designed a unified framework for solving CMOPs in which a higher priority is assigned to the constraints than to the objective functions, thus the search process prioritizes finding feasible solutions over finding optimal ones. Takahama *et al.* [29] and Asafuddoula *et al.* [30] proposed an  $\varepsilon$ -constrained dominance relation in which the constraint violation values of two solutions are not considered if they are smaller than a certain threshold. In the method of [31], infeasible solutions do not survive to the next generation if the number of feasible solutions is sufficient. Fan *et al.* [32] designed an angle-based constrained dominance principle in which  $\mathbf{x}_1$  dominates  $\mathbf{x}_2$  if the former is feasible while the latter is not. For a DCMOP, the feasible region can increase or decrease with dynamism. Giving feasible solutions a higher priority is beneficial for speeding up the convergence of an algorithm when the feasible region is large. However, emphasizing the importance of feasibility may reduce the population diversity.

Techniques in the second category try to balance the trade-off between feasibility and convergence during the evolutionary process. Angantyr *et al.* [33] and Young [34] developed a constrained dominance relation by combining the ranks of a solution based on its objective functions and constraint violation values. Woldeesenbet *et al.* [15] proposed a new constraint handling technique in which each objective function of a solution is modified in accordance with its original objective function values and constraint violation values. The dominance relation is determined on the basis of the modified objective functions and the nondominated sorting procedure proposed in the nondominated sorting genetic algorithm II (NSGA-II [13]). To improve the population diversity, Li *et al.* [35] developed a method in which the worst solution is given a second chance for survival when it is associated with an isolated subregion. To utilize useful information included in infeasible solutions, Peng *et al.* [36] introduced infeasible weights, which change with smaller constraint violation values and better objective function values, to maintain many well-diversified infeasible individuals. Ning *et al.* [37] proposed a constrained nondominated sorting rank approach in which each solution is associated with a

constrained nondomination rank in accordance with its Pareto rank and constraint rank. Sorkhabi *et al.* [38] designed an efficient approach for constraint handling in which infeasible particles are evolved in the constraint region toward feasibility, and feasible particles are evolved toward Pareto optimality. However, for a DCMOP, a smaller feasible region may lead to a reduction in the number of feasible solutions. In this case, removing some feasible solutions may cause the algorithm to fail to converge because the solutions in the POS are feasible.

Techniques in the last category aim to repair infeasible solutions. In [39], Harada *et al.* proposed a constraint handling method called Pareto descent repair, which incorporates the gradient projection method. However, the gradient information for DCMOPs is usually unavailable. Singh *et al.* [40] used an approximate descent direction method to reduce the degree of constraint violation. Jiao *et al.* designed a feasible-guiding strategy that aims to find feasible solutions close to the feasible region in a feasible direction with the help of infeasible solutions [41]. For a DCMOP, however, the feasible region varies irregularly with dynamism, and it is difficult to find a good feasible direction in which to repair infeasible solutions.

Additionally, there are some effective constraint handling strategies to solve the constraints in single objective optimization problems. Runarsson and Yao [58] proposed a stochastic ranking in which a probability is introduced to balance the objective and overall constraints violation stochastically. This approach can significantly improve the optimization performance without any special constraint handling operator. In [59], they designed an improved stochastic ranking in which comparison between two solutions may be based on the overall constraint violation alone or objective value alone as randomly determined. Thus, some infeasible solutions with better value may be selected in evolution. Allmendinger and Knowles [60] investigated two interesting types of ephemeral resource constraints (ERCs): one encodes periodic resource availabilities, and the other models ‘commitment’ constraints that make the evaluable part of the space a function of earlier evaluations conducted. The studies on both types of constraints are of great significance to real-world applications, especially closed-loop optimization settings. Note that ERCs are not standard constraints but restrictions on a series of solutions that can be actually evaluable at a given time during the optimization, arising because of resourcing issues [61].

#### B. Change Response Strategies

The strategies for responding to changes can be divided into the following three categories.

1) *Diversity Enhancement*: These methods increase the population diversity by means of certain methodologies when a change is detected. Woldeesenbet and Yen [42] proposed a dynamic EA with variable relocation that relocates solutions based on the changes in the objective function values due to a change in the environment and the average sensitivities of their decision variables to the corresponding changes in the objective space. Yang and Tinos [43] and Mavrovouniotis and Yang [45] proposed a hybrid immigrant scheme based on memory-based immigrants [44] and elitism-based immigrants, and these methods are effective in dealing with changing DMOPs.

2) *Memory Mechanism*: These mechanisms reuse the past

information to improve the performance of EAs [56]. Goh and Tan [11] proposed an idea based on comparing the potential of new regions with past information to decide whether a subpopulation should be initialized when an environmental change occurs. Specifically, the particular subpopulation must be reinitialized in the space from which the winner is sampled. Wang and Li [46] designed a reinitialization strategy to respond to environmental changes. In their reinitialization strategy, the initialized solutions are chosen from the current population and archived solutions with a probability of 0.2. In [47], Azzouz *et al.* proposed generating some solutions via a memory-based strategy that makes use of previous optimal solutions.

3) *Prediction Strategy*: Such a strategy usually predicts the state of the new environment using existing information and other learning techniques. Considering the properties of continuous DMOPs, Zhou *et al.* proposed a population prediction strategy that is divided into two parts: a center point and a manifold [16]. A sequence of center points is used to predict the new center, and the previous manifolds are maintained to estimate the next manifold.

The three types of change response strategies discussed above perform well for solving different DMOPs. Nevertheless, for DCMOPs, the feasible region move with dynamism, and a previously feasible region may become infeasible in a new environment. In other words, previously feasible solutions may become infeasible. Therefore, a change response strategy must be designed with proper consideration of the constraints.

### III. PROPOSED TEST PROBLEMS

Test problems play a crucial role in judging whether an algorithm is a candidate for solving MOPs [3]–[4]. Currently, test instances for DMOPs [3]–[5], [63] and CMOPs [6], [31], [48] have been proposed. However, these test problems do not consider dynamism and constraints simultaneously. Analogous to the design principle of dynamic single-objective test problems proposed in [56], the test problems of a DCMOP should also be close to reality, easy to describe, simple, and easy to analyze. In real-world applications of DCMOPs, the objective functions and the feasible region may both change with dynamism, changing the true POF. Specifically, the bounds of the feasible region may be determined by the unconstrained POF and constraints, and its true POF is a part of the lower boundary of the feasible region. Dynamism and constraints can change the lower boundary, causing the true POF to alternate between connectedness and disconnectedness as well as between convexity and concavity, which pose a tremendous challenge for the convergence speed and response capability of an algorithm. The feasible region may move with dynamism, resulting in changes to the feasible solutions, infeasible solutions, and nondominated solutions. Some previously nondominated solutions may become dominated or even infeasible in the new environment. Meanwhile, some previously infeasible solutions may become feasible or even nondominated. This possibility poses a challenge regarding an algorithm's capability of handling infeasible solutions. Based on these characteristics arising in real-world applications, this paper considers the four types of test problems, and the real-world problems related to these four types are discussed in Section S-II of the supplementary material.

Type I: The feasible region first increases with  $t$  and then decreases, while the true POF changes from continuous to disconnected and finally back to continuous.

Type II: The feasible region first decreases with  $t$  and then increases, while the true POF changes from disconnected to continuous and finally back to disconnected.

Type III: The feasible region first increases with  $t$  and then decreases, while the true POF changes from disconnected to continuous and finally back to disconnected.

Type IV: The feasible region first decreases with  $t$  and then increases, while the true POF changes from continuous to disconnected and finally back to continuous.

Considering that the objective functions in the electric power supply problem studied in [49] oscillate among several optimization modes, this study considers two cases for the unconstrained POF of a DCMOP, called Case 1 and Case 2.

Case 1: The unconstrained POF has one mode that includes convexity and concavity. Algorithms with fast convergence can easily solve such problems

Case 2: The objective functions oscillate among several modes that may include convexity and concavity.

For the real-world optimization problems especially for the optimization of electric energy (e.g., the dynamic power supply problem in magnesia grain manufacturing [49], the peaking shaving and valley filling problem in plug-in electric vehicles [62]), their objective functions and constraints may include trigonometric functions. Additionally, constraints in real-world problems usually cause the irregularity of feasible region, leading to the irregularity of the true POF in DCMOPs. Designing test problems with trigonometric functions can easily reflect these characteristics arising in real-world problems by adjusting the corresponding parameters. Based on the above discussions and benchmarks designed in [4], [6], and [48], this study proposes the following instance generator.

$$\begin{aligned} \min F(\mathbf{x}, t) &= (f_1(\mathbf{x}, t), f_2(\mathbf{x}, t))^T \\ \text{obj} \left\{ \begin{aligned} f_1(\mathbf{x}, t) &= (1 + g(\mathbf{x}_{II}, t))(h(\mathbf{x}_I, t) + A_t \sin(W_t \pi h(\mathbf{x}_I, t))) \\ f_2(\mathbf{x}, t) &= (1 + g(\mathbf{x}_{II}, t))(s(t) - h(\mathbf{x}_I, t) + A_t \sin(W_t \pi h(\mathbf{x}_I, t))) \end{aligned} \right. \quad (2) \\ s. t. \left\{ \begin{aligned} \cos(\theta) \times (f_2(\mathbf{x}, t) - e) - \sin(\theta) \times f_1(\mathbf{x}, t) - m(t) &\geq \\ a \times \left| \sin(b\pi \times (\sin(\theta') \times (f_2(\mathbf{x}, t) - e) + \cos(\theta') \times f_1(\mathbf{x}, t))) \right|^d & \\ sI \times f_1(\mathbf{x}, t) + f_2(\mathbf{x}, t) - z(t) &< 0 \\ z(t) &> m(t) \end{aligned} \right. \end{aligned}$$

In the objective functions,  $s(t)$  is a nonnegative dynamic parameter that controls the movement of the unconstrained POF. A large movement distance of  $s(t)$  causes the POF to also exhibit a large movement.  $\mathbf{x}_I$  and  $\mathbf{x}_{II}$  are subvectors of  $\mathbf{x}$ .  $h(\mathbf{x}_I, t)$  is a nonnegative function such that  $0 \leq h(\mathbf{x}_I, t) \leq s(t)$ .  $A_t$  controls the curvature of the unconstrained POF, and  $W_t$  determines the number of mixed concave and convex segments of the unconstrained POF. Setting  $W_t$  properly can make the unconstrained POF oscillate among several modes.  $g(\mathbf{x}_{II}, t)$  is a nonnegative function that determines the starting position and change of solutions in POS. The minimum value of  $g(\mathbf{x}_{II}, t)$  is zero, so the objective functions can be formulated as follows:

$$f_1 + f_2 = s(t) + 2A_t \sin(W_t \pi \frac{f_1 - f_2 + s(t)}{2}) \quad (3)$$

Based on Equation (3), the unconstrained POFs with different settings for  $s(t)$ ,  $A_t$ , and  $W_t$  are shown in Fig. 2.

In the constraints of Equation (2), the effects of parameters  $\theta$ ,

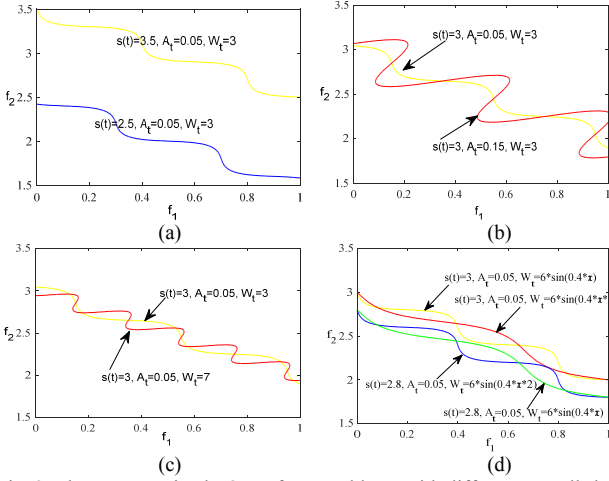


Fig. 2. The unconstrained POFs of test problems with different overall shapes.

$a$ ,  $b$ ,  $c$ ,  $d$ , and  $e$  on the feasible region caused by constraints are described in [6]. Additionally,  $sl$  controls the slope of the upper boundary of the feasible region caused by constraints.  $m(t)$  and  $z(t)$  are nonnegative dynamic parameters that control the lower and upper bounds, respectively, of the feasible region determined by constraints.

To embody the characteristic changes for problems of Types I-IV,  $s(t)$ ,  $m(t)$ , and  $z(t)$  are set as follows:

$$\begin{cases} s(t) = \alpha - \beta \times t \\ m(t) = \nu - \chi \times t \\ z(t) = \delta - \lambda \times t \end{cases} \quad (4)$$

where  $\alpha$  controls the starting position of the unconstrained POF;  $\nu$  and  $\delta$  control the starting positions of the lower and upper bounds, respectively, of the feasible region;  $\beta$  controls the step length of the unconstrained POF; and  $\chi$  and  $\lambda$  control the step lengths of the lower and upper bounds, respectively, of the feasible region.

For Cases 1 and 2, one can select between these two cases by setting  $W_t$  as desired.

Case 1: The unconstrained POF has one mode that includes convexity and concavity, with  $W_t$  set to a constant.

Case 2: The unconstrained POF oscillates among several optimization modes that may include convexity and concavity, with  $W_t = \varphi \sin(\eta \pi (t+1))$ . Note that  $\varphi$  controls the curvature of oscillation, while  $\eta$  determines the period of oscillation (i.e., the period is  $2\pi/\eta\pi$ ). Two diagrams are given in Fig. 3 to illustrate the impacts of  $\varphi$  and  $\eta$  on the unconstrained POFs. Their objective functions are given in Equations (5) and (6), respectively.

$$f_1 + f_2 = 3.64 - 0.07t + 0.1 \sin(3 \sin(0.4\pi t) \pi \frac{f_1 - f_2 + 3.64 - 0.07t}{2}) \quad (5)$$

$$f_1 + f_2 = 3.64 - 0.07t + 0.1 \sin(6 \sin(0.4\pi t) \pi \frac{f_1 - f_2 + 3.64 - 0.07t}{2}) \quad (6)$$

The ideas on scaling up the proposed test problems are discussed in Section S-III of the supplementary material.

#### IV. PROPOSED DCMOEA

In a DCMOP, the true POF and feasible region vary with dynamism. To handle infeasible solutions and respond to changes, this paper designs an algorithm called  $d$ CMOEA, whose basic framework is presented in Algorithm 1.

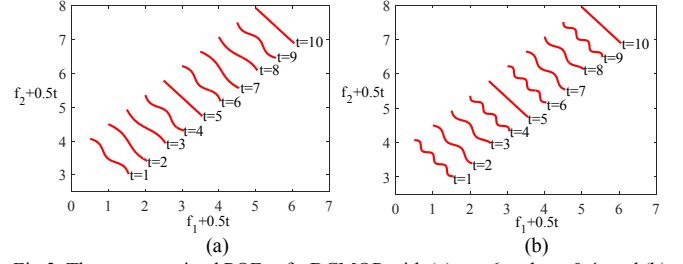


Fig. 3. The unconstrained POFs of a DCMOP with (a)  $\varphi=6$  and  $\eta=0.4$ ; and (b)  $\varphi=3$  and  $\eta=0.4$  at different times.

#### Algorithm 1 Framework of DCMOEA

---

```

1: Input:  $N$  (population size)
2: Output: a set of approximation POFs
3: Initialization: generate an initial parent population  $P$  randomly;
4:  $A = \text{Nonselection}(P, A)$ 
5: while termination condition not met do
6:   for  $i=1$  to  $N$  do
7:     if change is detected then
8:        $P = \text{ChangeResponse}(P)$ ;
9:        $A = \text{Nonselection}(P, A)$ ;
10:    end if
11:     $p_{i,1}$  and  $p_{i,2} = \text{MatingSelection}(P)$ 
12:     $q_i = \text{GenerateOffspring}(p_{i,1}, p_{i,2})$ ;
13:  end for
14:   $P = \text{PopulationSelection}(P, Q)$ ;
15:   $A = \text{Nonselection}(P, A)$ ;
16: end while
    
```

---

As shown in Algorithm 1,  $d$ CMOEA starts with an initial population  $P$ . Then, the nondominated solution set  $A$  is determined (line 4 of Algorithm 1). In the each generation, if a new environment is detected, then a change response strategy (line 8 of Algorithm 1) is used to update  $P$ , and a nondominated solution selection operator (line 9 of Algorithm 1) is applied to update  $A$ . Afterward, a mating selection operator (line 11 of Algorithm 1) is used to select two parents (i.e.,  $p_{i,1}$  and  $p_{i,2}$ ) for generating the  $i$ th child  $q_i$  (line 12 of Algorithm 1). Subsequently,  $q_i$  is stored in an offspring set  $Q$ . At the end of each generation, a population selection operator (line 14 of Algorithm 1) is applied to update  $P$ . Finally, a nondominated solution selection operator (line 15 of Algorithm 1) is used to update  $A$ . The above procedure is executed until the termination condition is met. In the following sections, each component of  $d$ CMOEA will be described in detail.

##### A. Initialization

In  $d$ CMOEA, the evolutionary process starts with an initial population  $P$  with  $N$  solutions. For a DCMOP, it is difficult to generate a feasible solution when the feasible region is small. However, the solutions in the POS and POF must be feasible, so this paper stipulates that there must be at least one feasible solution in the initial population to prevent the absence of nondominated solutions. During initialization, we first use Equation (7) to generate the components of each solution.

$$x_{i,j} = x_j^{\min} + \text{rand} \times (x_j^{\max} - x_j^{\min}) \quad (7)$$

where  $\text{rand}$  is a random number in the range  $[0, 1]$ .

Subsequently, the constraint deviation values of each solution are calculated using Equations (8) and (9).

$$\begin{cases} G_k(\mathbf{x}_i, t) = \max\{0, 0 - g_k(\mathbf{x}_i, t)\}, & 1 \leq k \leq n_g \\ H_k(\mathbf{x}_i, t) = \max\{0, |h_k(\mathbf{x}_i, t) - c_k|\}, & 1 \leq k \leq n_h \end{cases} \quad (8)$$

$$v(\mathbf{x}_i, t) = \sum_{k=1}^{n_g} G_k(\mathbf{x}_i, t) + \sum_{k=1}^{n_h} H_k(\mathbf{x}_i, t) \quad (9)$$

where  $G_k(\mathbf{x}_i, t)$  and  $H_k(\mathbf{x}_i, t)$  denote the  $k$ th inequality and equality constraint violations, respectively, at  $t$ . The parameter  $c$  is a tolerance value for relaxing the equality constraints to inequality constraints, and  $v(\mathbf{x}_i, t)$  is the total constraint violation of  $\mathbf{x}_i$  at  $t$ .  $\mathbf{x}_i$  is a feasible solution if  $v(\mathbf{x}_i, t)$  is zero.

If there is no feasible solutions, then the initial population is regenerated until there is at least one feasible solution.

### B. Nondominated Solution Selection

The nondominated solution selection procedure aims to preserve nondominated solutions from the Population  $P$  and  $A$ . For a DCMOP, some solutions in  $A$  may become dominated solutions or even infeasible solutions when a new environment arises. To obtain nondominated solutions with diversity, we propose that, when a change is detected, the infeasible and dominated solutions in  $A$  are first removed, followed by the infeasible and dominated solutions in  $P$ . Afterward,  $A$  is updated according to the dominance relationship between solutions in  $A$  and the nondominated solutions in  $P$ . Note that if the environment changes, then the infeasible and dominated solutions in  $P$  are removed, and  $A$  is updated according to the dominance relationship between solutions in  $A$  and the nondominated solutions in  $P$ . Specifically, when a new environment is detected, the constraint deviation values of all solutions in  $A$  are calculated using Equations (8) and (9), and solutions with constraint deviation values larger than zero are removed from  $A$ . Afterward, each solution in  $A$  is associated with a fitness value, which can be calculated via Equation (10).

$$F(i) = |\{i' \in R \mid i' \prec i\}| \quad (10)$$

where  $R$  is the set of solutions, and  $i$  and  $i'$  are the indices of two different solutions in  $R$ .  $|\cdot|$  represents the cardinality of a set, and  $i' \prec i$  indicates that  $i'$  dominates  $i$ .  $F(i)$  is the fitness value of the  $i$ th solution in  $R$ .

Subsequently, the dominated solutions in  $A$  are removed. After it, there are two cases for  $A$ : one is that  $A$  becomes an empty set, and another is that there is at least one solution in  $A$ .

To preserve nondominated solutions from the population  $P$ , the constraint deviation values of each solution in  $P$  are calculated using Equations (8) and (9). Afterward, all feasible solutions in  $P$  are copied into an empty set  $S$ . Each solution in  $S$  is associated with a fitness value according to Equation (10). All nondominated solutions in  $S$  are copied into an empty set  $S'$ . If  $A$  is empty, then all solutions in  $S'$  are copied into  $A$ . If there is at least one solution in  $A$ , then the dominance relations between the solutions in  $A$  and  $S'$  are assessed, and the following three cases are considered:

- 1) If a solution in  $S'$  is dominated by any solution in  $A$ , then this solution in  $S'$  is not considered.
- 2) If a solution in  $S'$  is not dominated by any solution in  $A$  and does not dominate any solution in  $A$ , then this solution in  $S'$  is added to  $A$ .
- 3) If a solution in  $S'$  is not dominated by any solution in  $A$  and dominates some solutions in  $A$ , then all solutions dominated by this solution in  $S'$  are removed from  $A$ .

Last, some solutions in  $A$  must be removed if  $|A|$  is greater than  $N$ . In this paper, the crowding distance operator applied in NSGA-II is used to perform a truncation operation to ensure

---

### Algorithm 2 Non-dominated solution selection

---

```

1: Input:  $P, A, S=\emptyset, S'=\emptyset$ 
2: Output:  $A$ 
3: if a new environment is detected then
4:   for  $i=1:|A|$ 
5:     Calculate constraint deviation value  $v(\mathbf{x}_i, t)$  of the  $i$ th solution in  $A$ 
       using Equations (8) and (9);
6:     if  $v(\mathbf{x}_i, t)$  is not zero then
7:       Remove the  $i$ th solution from  $A$ ;
8:     end if
9:   end for
10:  for  $i=1:|A|$ 
11:    Calculate the fitness value of the  $i$ th solution in  $A$  using Equation
       (10);
12:  end for
13:  Remove solutions having a fitness value that is more than one from  $A$ ;
14: end if
15: for  $i=1:N$ 
16:  Calculate constraint deviation value  $v(\mathbf{x}_i, t)$  of the  $i$ th solution in  $P$ ;
17:  if  $v(\mathbf{x}_i, t)$  is zero then
18:    Add the  $i$ th solution into  $S$ ;
19:  end if
20: end for
21: Calculate fitness values of solutions in  $S$  using Equation (10);
22: Copy all solutions having a fitness value of zero in  $S$  to an empty set  $S'$ ;
23: if  $A=\emptyset$  then
24:   All solutions in  $S'$  are copied into  $A$ ;
25: else
26:   for  $i=1:|S'|$ 
27:     if the  $i$ th solution in  $S'$  is dominated by anyone in  $A$  then
28:        $A$  remains unchanged;
29:     else
30:       if the  $i$ th solution in  $S'$  is not dominated by any one in  $A$ , and it does
         not dominate any one in  $A$  then
31:         The  $i$ th solution in  $S'$  is added into  $A$ ;
32:       else
33:         All solutions in  $A$  dominated by  $i$ th solution are removed;
34:       end if
35:     end if
36:   end for
37: end if

```

---

that  $|A|$  is equal to  $N$ . The pseudocode for nondominated solution selection is presented in Algorithm 2.

### C. Mating Selection and Genetic Operators

Mating selection plays an important role in producing new offspring. For MOPs, binary tournament selection based on the dominance relationships between solutions is a common selection method [18], [21], and [31]. Unlike CMOPs and DMOPs, DCMOPs involve constraints and dynamism simultaneously, implying that the feasible region of a DCMOP varies with dynamism. Thus, the method used to select parents from  $P$  influences the population diversity and convergence speed of the algorithm. Simply selecting from among the feasible solutions may reduce the diversity of the offspring, while selecting an excess number of infeasible solutions may slow the convergence speed of the algorithm. This study proposes to modify the objective function values based on the feasibility ratio of the solutions to balance the numbers of feasible and infeasible solutions in the parent population. The objective function value modification method proposed by Woldesenbet *et al.* [15] is based on the feasibility ratio and does not involve parameter tuning, which makes it easy to implement in  $d$ CMOEA. Therefore, this method is adopted in  $d$ CMOEA to modify the objective function values so that a promising parent population can be obtained. Note that this

modification method includes a distance measure and an adaptive penalty. The detailed modification procedure in calculating the distance measure is as follows:

First, *dCMOEA* normalizes the objective values by Equations (11)-(13).

$$f_{m,t}^{\max} = \max_{\mathbf{x} \in P} f_m(\mathbf{x}, t) \quad (11)$$

$$f_{m,t}^{\min} = \min_{\mathbf{x} \in P} f_m(\mathbf{x}, t) \quad (12)$$

$$\overline{f_m(\mathbf{x}, t)} = \frac{f_m(\mathbf{x}, t) - f_{m,t}^{\min}}{f_{m,t}^{\max} - f_{m,t}^{\min}} \quad (13)$$

where  $f_{m,t}^{\max}$  and  $f_{m,t}^{\min}$  denote the maximum and minimum of the *m*th objective function value in  $F(\mathbf{x}, t)$  at *t*, respectively.

$\overline{f_m(\mathbf{x}, t)}$  is the *m*th normalized objective function value of  $\mathbf{x}$  at *t*.

The total normalized constraint violation of a candidate solution at *t* is then calculated by Equations (8), (9), and (14):

$$\overline{v(\mathbf{x}_i, t)} = \frac{1}{n_g} \sum_{k=1}^{n_g} \frac{G_k(\mathbf{x}_i, t)}{G_{k,t}^{\max}} + \frac{1}{n_h} \sum_{k=1}^{n_h} \frac{H_k(\mathbf{x}_i, t)}{H_{k,t}^{\max}} \quad (14)$$

where  $\overline{v(\mathbf{x}_i, t)}$  is the total constraint violation of  $\mathbf{x}_i$  at *t*. It is worth noting that the smaller the value of  $\overline{v(\mathbf{x}_i, t)}$  is, the better the candidate solution.

The “distance” value of  $\mathbf{x}_i$  in the *m*th objective function (denoted as  $d_m(\mathbf{x}_i, t)$ ) is obtained by Equation (15).

$$d_m(\mathbf{x}_i, t) = \sqrt{\overline{f_m(\mathbf{x}_i, t)}^2 + \overline{v(\mathbf{x}_i, t)}^2} \quad (15)$$

The “penalty” value of  $\mathbf{x}_i$  in the *m*th objective function (denoted as  $p_m(\mathbf{x}_i, t)$ ) is obtained by Equation (16).

$$p_m(\mathbf{x}_i, t) = (1 - r_f) \times \overline{v(\mathbf{x}_i, t)} + r_f \times Y_m(\mathbf{x}_i, t) \quad (16)$$

where  $r_f$  denotes the feasible ratio of the current population,  $r_f = \frac{\text{number of feasible solutions in current population}}{\text{population size}}$ , and

$$Y_m(\mathbf{x}_i, t) = \begin{cases} 0, & \text{if } \overline{v(\mathbf{x}_i, t)} = 0 \\ \overline{f_m(\mathbf{x}_i, t)}, & \text{otherwise} \end{cases}$$

The *m*th modified objective value of  $\mathbf{x}_i$  is formulated as follows:

$$f'_m(\mathbf{x}_i, t) = d_m(\mathbf{x}_i, t) + p_m(\mathbf{x}_i, t) \quad (17)$$

After the objective function values of  $\mathbf{x}_i$  are modified, each solution is assigned a fitness value in accordance with Equation (10), and crowding distances are calculated using the method proposed in NSGA-II. Then choosing two parents (i.e.,  $p_{i,1}$  and  $p_{i,2}$ ) for the *i*th child employs the mating selection operator according to the fitness values and crowding distances of the solutions in *P*. The pseudocode for mating selection is presented in Algorithm 3.

Once a parent population has been constructed, the popular simulated binary crossover (SBX) and polynomial mutation (PM) operators proposed in [54] are used to generate offspring.

#### D. Population Selection Operator

For a DCMOP, if the feasible region decreases with dynamism, then more infeasible solutions may arise. Conversely, the number of infeasible solutions may decrease. Therefore, handling infeasible solutions while considering dynamism is key to obtaining a suitable set of approximations.

---

#### Algorithm 3 Mating selection

---

```

1: Input: P
2: Output:  $p_{i,1}, p_{i,2}$ 
3: for  $j=1:2$ 
4:   Random select two different individuals  $p_1, p_2$  from P;
5:   if  $F(p_1) < F(p_2)$  then
6:      $p_{i,j} = p_1$ ;
7:   else
8:     if  $F(p_1) > F(p_2)$  then
9:        $p_{i,j} = p_2$ ;
10:    else
11:      if  $d(p_1) > d(p_2)$ 
12:         $p_{i,j} = p_1$ ;
13:      else
14:        if  $d(p_1) < d(p_2)$  then
15:           $p_{i,j} = p_2$ ;
16:        else
17:          Random select an individual from  $p_1$  and  $p_2$ ;
18:        end if
19:      end if
20:    end if
21:  end if
22: end for

```

---

Simply discarding infeasible solutions might cause the algorithm to become trapped in local optima [15] whereas retaining more infeasible solutions may reduce the convergence speed of the algorithm. Note that *dCMOEA* must have a high convergence speed to allow it to track the POF quickly in the case of fast and frequent changes. To balance the convergence speed of the algorithm and the diversity of the population, *dCMOEA* considers two cases for handling infeasible solutions: the case where the number of feasible solutions of in *P* and *Q* is less than or equal to the threshold *NF*, and the case where the number of feasible solutions is more than *NF*.

a) *The number of feasible solutions is fewer than or equal to NF.* For convenience of description, the solutions in *P* and *Q* are copied to an empty set *C*. The feasible solutions in *C* are allocated to a feasible set *FC*, and the infeasible ones are allocated to an infeasible set *IC*. Considering that the solutions in the POS must be feasible, we propose that all solutions in *FC* should be retained for the next generation, and the remaining  $N - |FC|$  solutions in the population should be chosen from the infeasible solutions in *IC*. To select  $N - |FC|$  infeasible solutions from *IC*, we first modify the objective function values of these infeasible solutions. Because there are no feasible solutions in *IC*, Equation (16) is modified as shown in Equation (18). In other words, the objective function values of the infeasible solutions in *IC* are revised in accordance with Equations (8), (11)-(15), (17), and (18).

$$p_m(\mathbf{x}_i, t) = \overline{v(\mathbf{x}_i, t)} \quad (18)$$

Subsequently, each individual in *IC* is assigned a fitness value that is calculated using Equation (10). Finally, we select  $N - |FC|$  infeasible solutions by means of the fast nondominated sorting and crowded-comparison operators proposed in [13].

b) *The number of feasible solutions is more than NF.* When the number of feasible solutions is large, the number of feasible solutions that are propagated to the next generation may also be large. In this case, we propose to modify the objective function values of all solutions in *C* in accordance with Equations (11)-(17). Once the modified objective function values of all solutions have been obtained, *N* solutions are chosen by means of the fast nondominated sorting and crowded-comparison

**Algorithm 4** Population selection procedure

---

```

1: Input:  $P, Q, C=\Phi, FC=\Phi, IC=\Phi$ 
2: Output:  $P$ 
3: Cope all solutions in  $P$  and  $Q$  to  $C$ .
4: Calculate the number of feasible solutions in  $P$  and  $Q$ ;
5: if The number of feasible solutions  $\leq NF$  then
6:   All feasible solutions in  $C$  are assigned to  $FC$ , and infeasible ones are
   allocated to  $IC$ .
7:   Modify the objective function values of solutions in  $IC$  using
   Equations (8), (11)-(15), (17), and (18).
8:   Calculate the fitness value of each solution using Equation (10).
9:   Choose  $N-|FC|$  infeasible solutions by fast nondominated sorting and
   crow crowded-comparison operator.
10:  Let  $P=\Phi$ , and cope all feasible solutions in  $FC$  and the selected
    $N-|FC|$  infeasible solutions to  $P$ .
11: else
12:  Modify the objective function values of all solutions in  $C$  according
   to Equations (11)-(17)
13:  Calculate the fitness value of each solution, and choose  $N$  solutions
   from  $C$ .
14:  Let  $P=\Phi$ , and cope the selected  $N$  solutions to  $P$ .
15: end if

```

---

operators proposed in [13].

The pseudocode for the population selection procedure is presented in Algorithm 4.

### E. Dynamism Handling

If the optimizer is aware of the changes in a DCMOP, then it does not need to detect change; otherwise, change detection is usually needed. When a change arises, change response has an important role in tracking the new POF in a changing environment. In this section, we first discuss the change detection method that detects the changes arising in DCMOPs with unforeseen environmental changes. Subsequently, the proposed change response strategy is described.

#### 1) Change Detection

The authors in [11], [16], [17], and [18] proposed that changes can be detected by comparing previous objective function values with re-evaluated ones. However, this change detection method may also fail when only the constraints are dynamic. For a DCMOP, an environmental change may arise in either the objective functions or constraints. In this paper, we propose that both the objective functions and constraints are detected to accurately detect environmental changes. However, detection will be computationally expensive if each individual in the population is chosen as a detector. To improve detection efficiency, the number of individuals chosen as detectors is restricted to  $10\%N$ , which is suggested by [11] and [18]. The steps to detect environmental changes are as follows:

Step 1: Randomly choose  $10\%N$  individuals from the population.

Step 2: Let  $i=1$ ;

Step 3: If  $i \leq 10\%N$ , then recalculate each objective function value for the  $i$ th solution; otherwise, the detection procedure ends.

Step 4: If any of the recalculated objective function values of the  $i$ th solution is different from their corresponding previous value, then a change is successfully detected, and the detection procedure ends; otherwise, go to Step 5.

Step 5: Recalculate the difference between the left- and right-hand sides of each constraint. If this value differs from

the one in the previous generation, then a change is assumed to be successfully detected, and the detection procedure ends; otherwise, let  $i=i+1$  and go back to Step 3.

#### 2) Change Response

For a DCMOP, the true POF or feasible region may change with  $t$ . The nondominated solutions in the POF obtained by an algorithm at  $t-1$  may have become dominated or infeasible. Similarly, the infeasible solutions at  $t-1$  may have become feasible or nondominated. Additionally, a dynamic environment may cause the feasible region to become larger or smaller. Simply discarding all previous solutions and randomly reinitializing the population might be beneficial for population diversity, but more optimization time will be needed for the algorithm to converge. In contrast, reusing all previous solutions to search for new nondominated solutions may reduce the population diversity. For these reasons, this paper adopts random immigrants and memory schemes to reinitialize the population when a new environment arises.

Based on the above discussion, the population in a new environment will be composed of 50% reinitialized solutions and 50% previous ones. Note that the reinitialized solutions are generated using the method discussed in the *Initialization* section and are copied into an empty set  $R$ . To choose 50% of the previous solutions, all previous solutions are re-evaluated, and their modified objective function values are calculated using Equations (8), (9) and (11)-(15). Then, we select the desired number of feasible previous solutions from the population  $P$  by means of the fast nondominated sorting and crowded-comparison operators proposed in [13].

To enable the algorithm track the new POF quickly, once a new environment is detected, we exploit information collected from the reinitialized solutions and new approximation of the POF to update the retained previous solutions. Specifically, the reinitialized solutions and retained solutions are first combined, and the feasible solutions are then copied into a new set  $FS$ . Each solution in  $FS$  is assigned a fitness value. Considering that solutions with fitness values of less than two may be close to the new POF, we propose that each modified solution should include the information of these appealing solutions. For the  $i$ th retained previous solution, Equation (19) is used to locate a new position for each of its variables.

$$x_{i,j} = x_{i,j} + rand \times (x_{rbest,j} - x_{i,j}) \quad (19)$$

where  $x_{rbest,j}$  is the  $j$ th variable of a solution with a fitness value less than the fitness values of two randomly solutions from  $FS$ .

However, the variable value obtained via Equation (19) may be smaller or greater than the corresponding lower or upper bound on that variable, making this variable infeasible. The bound constraints on each decision variable are known, so such infeasibilities can be repaired in a timely manner. Considering that the feasible region may become smaller or larger with dynamism, the feasibility ratio is considered in the repair method. To quickly calculate the feasibility ratio, we use a rule of thumb (i.e., calculating the percentage of feasible solutions during the reinitialization stage) to estimate the feasibility ratio at  $t$ . The calculation method is given in Equation (20).

$$\rho = \frac{NR}{N/2} \quad (20)$$

where  $NR$  denotes the number of feasible solutions in the



**Algorithm 5** Change response

---

```

1: Input:  $P, CSC=\emptyset, R=\emptyset$ 
2: Output:  $P$ 
3: Generate  $N/2$  solutions by the method in Section IV-A, and copy them
   into  $R$ ;
4: Calculate the feasibility ratio of solutions in  $R$  using Equation (20);
5: Calculate the modified objective values of all solutions in  $P$ ;
6: Calculate the fitness values and crowding distances of all solutions in  $P$ ;
7: Select 50% feasible old solutions from population  $P$  according to the
   fitness values and crowding distances, and put them into a new set  $CS$ ;
8:  $CSC=CS\cup R$ 
9: Calculate the constraint violation values of solutions in  $CSC$ ;
10: for  $i=1:|FO|$ 
11:   Random select an individual with fitness value less than two from  $P$ ;
12:   for  $j=1:n$ 
13:     Generate  $j$ th variable  $x_{i,j}$  by Equation (19);
14:     if  $x_{i,j} < x_j^{\min}$  then
15:       Repair it by Equations (21) and (23);
16:     else
17:       if  $x_{i,j} > x_j^{\max}$  then
18:         Repair it by Equations (22) and (23);
19:       end if
20:     end if
21:   end for
22: end for

```

---

reinitialization stage.

For an infeasible variable that exceeds its lower bound, its optimal value may be near to the corresponding lower bound. Therefore, we find a feasible variable that is near its lower bound. In contrast, a feasible variable close to its upper bound is determined. When the feasible ratio is small, a minor revision is made for the corresponding infeasible variable. Conversely, a major revision is made. Based on the above discussions, this study uses Equations (20), (21) and (22) to repair the  $j$ th variable below  $x_j^{\min}$  and Equations (20), (22) and (23) to repair the  $j$ th variable above  $x_j^{\max}$ .

$$x_j = x_j^{\min} + r \times (x_{i,j} - x_j^{\min}) \quad (21)$$

$$x_j = x_j^{\max} + r \times (x_{i,j} - x_j^{\max}) \quad (22)$$

$$r = \text{rand} \times \rho \quad (23)$$

The pseudocode for the change response procedure is presented in Algorithm 5.

#### F. Computational Complexity of One Generation of $dCMOEA$

In the loop (lines 6–15 of Algorithm 1) of each generation, computational resources are mainly consumed by the mating selection, offspring reproduction, population selection, and nondominated solution selection procedures, while other procedures require less computational cost. Selecting parents (i.e., the mating selection operator in line 11 of Algorithm 1) takes  $O(MN)$  computations, where  $M$  is the number of objectives. Generating an offspring solution (line 12 of Algorithm 1) requires  $O(M)$  computations, so the offspring reproduction takes  $O(MN)$  computations. The population selection procedure (line 13 of Algorithm 1) requires  $O(MN^2)$  computations on the fitness assignment and  $O(N^2 \log N)$  computations on elitist preservation. The nondominated solution selection procedure (line 15 of Algorithm 1) spends  $O(MN^2)$  computations. Thus, the overall computational complexity of  $dCMOEA$  for one generation is  $O(MN^2)$  or  $O(N^2 \log N)$ , whichever is larger.

## V. EXPERIMENTAL STUDY

In this section, we will examine the effectiveness of the proposed test instances and  $dCMOEA$ . For the test instances, if the performances of several representative algorithms are clearly distinct, then this will show that the proposed test instances are effective. For  $dCMOEA$ , if it can outperform all algorithms considered for comparison on most or even some of the test instances, then this will suggest that  $dCMOEA$  is competitive for solving DCMOPs.

### A. Compared Algorithms

Only a few MOEAs have already been proposed for solving DCMOPs. Therefore, in addition to two algorithms for solving DCMOPs (i.e., DC-MOEA, proposed in [24], and DNSGA-II-A, proposed in [25]), this paper also considers three representative constrained multiobjective optimization algorithms for solving CMOPs (i.e., constrained NSGA-II (C-NSGA-II) [15], constrained NSGA-III (C-NSGA-III) [31], and constrained two-archive EA (C-TAEA) [21]) for comparison. For handling dynamic environments, the nondominated solution selection operator and the change detection method proposed in this study and the restart scheme for change response were incorporated into C-NSGA-II, C-NSGA-III, and C-TAEA. The resulting algorithms are called DC-NSGA-II, DC-NSGA-III, and DC-TAEA, respectively.

### B. Parameter Settings

- 1) To study the impact of the change frequency ( $\tau_t$ ),  $\tau_t$  was set to values of 10, 15, and 20. The number of changes was  $n_t=21$ . To minimize the effect of static optimization, each algorithm is allowed to run for 40 generations before the first change, and the total number of generations was  $n_t \times \tau_t + 40$ .
- 2) *Settings of the Test Problems:* According to [6], some of the parameters used in Equation (2) were set as follows:  $\theta=0.25\pi$ ,  $a=0.2$ ,  $b=1$ ,  $c=1$ ,  $d=0.5$ , and  $e=1$ . To cause the true POFs of the test problems to change in accordance with the characteristics of problems of Types I-IV,  $\theta'$ ,  $A_t$ , and  $z(t)$  were set to  $-\pi/16$ , 0.05, and 6, respectively, and  $s(t)$  and  $m(t)$  were set as follows:

$$\text{Type I: } \begin{cases} s(t) = \max(3.5 - 0.14 \times t, 0.7 + 0.14 \times t) \\ m(t) = \max(1.43 - 0.05 \times t, 0.43 + 0.05 \times t) \end{cases}$$

$$\text{Type II: } \begin{cases} s(t) = \max(2.5 - 0.05 \times t, 1.5 + 0.05 \times t) \\ m(t) = \max(1.16 - 0.075 \times t, -0.34 + 0.075 \times t) \end{cases}$$

$$\text{Type III: } \begin{cases} s(t) = \min(2.1 + 0.14 \times t, 4.9 - 0.14 \times t) \\ m(t) = \min(0.93 + 0.05 \times t, 1.93 - 0.05 \times t) \end{cases}$$

$$\text{Type IV: } \begin{cases} s(t) = \min(2 + 0.05 \times t, 3 - 0.05 \times t) \\ m(t) = \min(0.41 + 0.075 \times t, 1.91 - 0.075 \times t) \end{cases}$$

In the designed test problems,  $h(\mathbf{x}_t, t)$  is set to  $x_1$ , and  $g(\mathbf{x}_{II}, t)$  is denoted as follows:

$$g(\mathbf{x}_{II}, t) = \sum_{j=2}^n (x_j - (1 - 0.9 \times \sin(0.2\pi \times t)))^2 \quad (24)$$

To cause the objective functions to change in accordance with the modes corresponding to Cases 1 and 2,  $W_t$  was set as follows:

Case 1:  $W_t=2$ ;

TABLE I  
DIFFERENT BENCHMARKS THAT CONSIST OF TYPE I-IV AND CASE 1-2.

Instance No.	1	2	3	4
Combination	Type I+Case 1	Type II+Case 1	Type III+Case 1	Type IV+Case 1
Instance No.	5	6	7	8
Combination	Type I+Case 2	Type II+Case 2	Type III+Case 2	Type IV+Case 2

Case 2:  $W_r = 6 \times \sin(0.2 \times \pi(t+1))$ .

Based on the above settings, eight test problems can be obtained, which are shown in Table I. The true POF and POS of each problem are given in Section S-IV of the supplementary material.

- 3) *Decision Variables*. Each test problem has ten decision variables;  $x_1 \in [0, 1]$ , and  $x_j \in [0, 2]$  for  $j \geq 2$ .
- 4) *Algorithm Parameters*: Both the parent population size  $N$  and the offspring population size in all algorithms were set to 200.  $NF$  was set to 100. In SBX, the crossover probability ( $Cr$ ) and the distribution index ( $dc$ ) were set to 0.8 and 5, respectively. In PM, the mutation probability ( $mp$ ) and the distribution index ( $dm$ ) were set to 0.05 and 40, respectively. The detailed experiments that select these parameters are given in Section S-V of the supplementary material.

The other parameters for the five compared algorithms were the same as those used in the referenced papers.

### C. Performance Indicators

In our experimental studies, three performance metrics, i.e., the hypervolume (HV) [8], inverted generational distance (IGD) [9], and Schott's spacing metric (SP) [52] were adopted to compare the algorithms' performance. The HV and IGD metrics simultaneously measure the diversity and convergence of the results, and the SP metric measures how the solutions in the discovered POF are distributed. Let  $POF^*$  be the obtained approximation set of POF.

1) HV: HV can assess the size of the area covered by the obtained approximation set. The reference point for the calculation of HV is set to  $(z_1+1, z_2+1)$ , where  $z_1$  and  $z_2$  are the maximum values of two objective values of the true POF. A higher HV value means a better approximation set.

2) IGD: IGD is calculated by Equation (25).

$$IGD = \frac{1}{n_{POF}} \sum_{i=1}^{n_{POF}} d_i \quad (25)$$

where  $n_{POF} = |POF|$ , and  $d_i$  is the minimum Euclidean distance between the  $i$ th member in POF and one in  $POF^*$ . The lower the IGD value, the better the obtained approximation set.

3) SP: Schott's SP metric tests the distribution of the obtained POF, and SP is calculated by Equation (26). Note that a smaller SP value indicates a better quality of the results.

$$SP = \sqrt{\frac{1}{n_{POF^*} - 1} \sum_{i=1}^{n_{POF^*}} (D_i - \bar{D})^2} \quad (26)$$

where  $D_i$  is the Euclidean distance between the  $i$ th solution and its nearest member in  $POF^*$ .  $\bar{D}$  is the average value of  $D_i$ .

### D. Empirical Results

For each combination  $(n_r, \tau_r)$  for a test problem, each algorithm was run 30 times on each test instance, and the mean and standard deviation of the results were recorded. Note that the Wilcoxon rank-sum test [53] at the 0.05 significance level was used to determine whether the performance of one algorithm statistically differs from that of another algorithm with respect to each performance metric.

To judge whether the proposed test instances are effective, the ranking method described in [4] was used. Specifically, an algorithm ranks the highest if it outperforms the largest number of competitors, and the algorithm that outperforms the fewest competitors will be assigned the worst rank. Multiple algorithms may have the same rank if they outperform the same number of other algorithms. Note that for each algorithm, the average rank was calculated based on three combinations for a given test problem. An algorithm was considered the best if it had the smallest average rank. Conversely, an algorithm was considered the worst if it had the largest average rank. Based on this ranking method, the obtained rank of each algorithm is listed in Section S-VI of the supplementary material.

It can be observed that  $dCMOEA$  achieves the best performance in terms of all three metrics, followed by DC-NSGA-II. DC-NSGA-II-A performs the worst among the compared algorithms. DC-NSGA-III, DC-TAEA, and DC-MOEA show different performances on different test problems. The experimental results indicate that the designed test problems can clearly distinguish the performance of each algorithm. Therefore, the proposed test problems are effective. To analyze the algorithms' performance in terms of each of the three metrics in detail, the obtained average results and standard deviations for the HV, IGD, and SP metrics are shown in Tables II-IV, respectively. The computational time of each algorithm on each test problem is given in Section S-VII of the supplementary material.

It can be observed from Table II that on all test problems,  $dCMOEA$  achieves the best performance in terms of the HV metric. Clearly,  $dCMOEA$  is more promising than the other algorithms for solving these test instances. DC-NSGA-II achieves the second-best performance. Note that  $dCMOEA$  borrows the idea of modifying the objective function values that is used in DC-NSGA-II. DC-NSGA-II-A also modifies the objective function values, but it performs worse on all test problems, which may imply that an effective method of modifying the objective function values may be helpful for handling the infeasible solutions arising in DCMOPs.

As seen in Table III, on the proposed test instances, all five compared algorithms are outperformed by  $dCMOEA$  in terms of the IGD metric. The results obtained by DC-NSGA-II are second only to those obtained by  $dCMOEA$ , whereas DC-NSGA-II-A performs the worst among the compared algorithms. Overall, the experimental results demonstrate the capability of  $dCMOEA$  for solving DCMOPs. As a supplement to the tabular presentation, Fig. 4 shows the evolutionary curves of the average IGD values on the first two test problems with  $\tau_r=15$  and  $n_r=21$ . The evolutionary curves on the other test problems are given in Section S-VIII of the supplementary material. Note that the evolutionary curves of the results obtained by DC-NSGA-II-A are not included because of its poor performance. Compared with the other algorithms,  $dCMOEA$  responds to environmental changes more steadily and recovers faster for the most of the test problems, implying its higher convergence performance. Nevertheless,  $dCMOEA$  performs similarly to DC-NSGA-II but better than DC-MOEA, DC-NSGA-III, and DC-TAEA on all test problems.

Table IV presents the results achieved by the six algorithms in terms of the SP metric. This table shows that  $dCMOEA$  achieves the best results on most of the test problems. For the

TABLE II  
MEAN AND STANDARD DEVIATION VALUES OF HV METRIC OBTAINED BY SIX ALGORITHMS

Ins	$(\tau_i, n_i)$	DC-MOEA	DC-NSGA-II-A	DC-NSGA-II	DC-NSGA-III	DC-TAEA	$d$ CMOEA
1	(10, 21)	2.60374(0.06625) <sup>‡</sup>	1.95383(0.05459) <sup>‡</sup>	3.28327(0.01632) <sup>‡</sup>	3.15395(0.02404) <sup>‡</sup>	3.10990(0.03625) <sup>‡</sup>	<b>3.34912(0.01415)</b>
	(15, 21)	3.13594(0.03853) <sup>‡</sup>	1.93553(0.06234) <sup>‡</sup>	3.42097(0.01150) <sup>‡</sup>	3.34736(0.01916) <sup>‡</sup>	3.34060(0.02179) <sup>‡</sup>	<b>3.45703(0.00782)</b>
	(20, 21)	3.34229(0.02054) <sup>‡</sup>	1.94408(0.08024) <sup>‡</sup>	3.46887(0.01025) <sup>‡</sup>	3.42745(0.01028) <sup>‡</sup>	3.42957(0.01280) <sup>‡</sup>	<b>3.49285(0.00384)</b>
2	(10, 21)	2.84153(0.04604) <sup>‡</sup>	1.99146(0.05402) <sup>‡</sup>	3.30633(0.01902) <sup>‡</sup>	3.19755(0.02965) <sup>‡</sup>	3.19393(0.01496) <sup>‡</sup>	<b>3.37086(0.00965)</b>
	(15, 21)	3.23280(0.01984) <sup>‡</sup>	2.01934(0.05110) <sup>‡</sup>	3.41326(0.01049) <sup>‡</sup>	3.34945(0.01293) <sup>‡</sup>	3.35799(0.01189) <sup>‡</sup>	<b>3.44178(0.00560)</b>
	(20, 21)	3.37416(0.01230) <sup>‡</sup>	2.02867(0.05891) <sup>‡</sup>	3.44747(0.00947) <sup>‡</sup>	3.40669(0.00745) <sup>‡</sup>	3.42321(0.00689) <sup>‡</sup>	<b>3.46634(0.00387)</b>
3	(10, 21)	2.92464(0.06816) <sup>‡</sup>	1.94140(0.07697) <sup>‡</sup>	3.28102(0.02151) <sup>‡</sup>	3.15196(0.02788) <sup>‡</sup>	3.10785(0.03041) <sup>‡</sup>	<b>3.35154(0.01319)</b>
	(15, 21)	3.23733(0.03320) <sup>‡</sup>	1.97205(0.06935) <sup>‡</sup>	3.41084(0.01378) <sup>‡</sup>	3.34460(0.01673) <sup>‡</sup>	3.33664(0.01845) <sup>‡</sup>	<b>3.45268(0.00701)</b>
	(20, 21)	3.38523(0.01847) <sup>‡</sup>	1.96287(0.07138) <sup>‡</sup>	3.46071(0.00931) <sup>‡</sup>	3.42168(0.00999) <sup>‡</sup>	3.42387(0.01216) <sup>‡</sup>	<b>3.48799(0.00401)</b>
4	(10, 21)	3.01242(0.04379) <sup>‡</sup>	1.98642(0.06159) <sup>‡</sup>	3.29439(0.01592) <sup>‡</sup>	3.18965(0.02119) <sup>‡</sup>	3.19466(0.01835) <sup>‡</sup>	<b>3.35571(0.01001)</b>
	(15, 21)	3.30370(0.01666) <sup>‡</sup>	1.97185(0.05262) <sup>‡</sup>	3.40346(0.01027) <sup>‡</sup>	3.34250(0.01108) <sup>‡</sup>	3.35474(0.01184) <sup>‡</sup>	<b>3.43481(0.00503)</b>
	(20, 21)	3.44553(0.01128) <sup>‡</sup>	2.00037(0.05305) <sup>‡</sup>	3.44100(0.00655) <sup>‡</sup>	3.40424(0.00653) <sup>‡</sup>	3.41575(0.00583) <sup>‡</sup>	<b>3.45823(0.00345)</b>
5	(10, 21)	2.63167(0.05798) <sup>‡</sup>	1.89519(0.06576) <sup>‡</sup>	3.27981(0.02171) <sup>‡</sup>	3.17105(0.02787) <sup>‡</sup>	3.10670(0.04173) <sup>‡</sup>	<b>3.37017(0.01575)</b>
	(15, 21)	3.15213(0.04511) <sup>‡</sup>	1.91493(0.07719) <sup>‡</sup>	3.43042(0.01743) <sup>‡</sup>	3.35195(0.02616) <sup>‡</sup>	3.34925(0.02209) <sup>‡</sup>	<b>3.47585(0.00845)</b>
	(20, 21)	3.36440(0.03241) <sup>‡</sup>	1.90705(0.05774) <sup>‡</sup>	3.47698(0.01349) <sup>‡</sup>	3.43951(0.01333) <sup>‡</sup>	3.44036(0.01377) <sup>‡</sup>	<b>3.51164(0.00337)</b>
6	(10, 21)	2.87412(0.04340) <sup>‡</sup>	1.95560(0.06411) <sup>‡</sup>	3.29883(0.01667) <sup>‡</sup>	3.20263(0.02323) <sup>‡</sup>	3.18370(0.02756) <sup>‡</sup>	<b>3.36944(0.01193)</b>
	(15, 21)	3.23933(0.02207) <sup>‡</sup>	1.94743(0.04891) <sup>‡</sup>	3.40850(0.01270) <sup>‡</sup>	3.34589(0.01407) <sup>‡</sup>	3.35631(0.01644) <sup>‡</sup>	<b>3.44045(0.00479)</b>
	(20, 21)	3.37102(0.00899) <sup>‡</sup>	1.97885(0.05960) <sup>‡</sup>	3.44211(0.00927) <sup>‡</sup>	3.41022(0.00866) <sup>‡</sup>	3.41913(0.00970) <sup>‡</sup>	<b>3.46181(0.00749)</b>
7	(10, 21)	2.97546(0.06900) <sup>‡</sup>	1.91102(0.06525) <sup>‡</sup>	3.27378(0.03021) <sup>‡</sup>	3.17135(0.02920) <sup>‡</sup>	3.09705(0.03963) <sup>‡</sup>	<b>3.37201(0.02073)</b>
	(15, 21)	3.25330(0.03195) <sup>‡</sup>	1.91702(0.06613) <sup>‡</sup>	3.41681(0.01849) <sup>‡</sup>	3.35258(0.02546) <sup>‡</sup>	3.33658(0.02393) <sup>‡</sup>	<b>3.46858(0.00956)</b>
	(20, 21)	3.40318(0.02562) <sup>‡</sup>	1.93060(0.06085) <sup>‡</sup>	3.46750(0.01662) <sup>‡</sup>	3.42836(0.01533) <sup>‡</sup>	3.43049(0.01510) <sup>‡</sup>	<b>3.50450(0.00315)</b>
8	(10, 21)	3.02515(0.04753) <sup>‡</sup>	1.92511(0.06544) <sup>‡</sup>	3.29749(0.01880) <sup>‡</sup>	3.19295(0.02305) <sup>‡</sup>	3.18289(0.02647) <sup>‡</sup>	<b>3.35664(0.01033)</b>
	(15, 21)	3.31749(0.02167) <sup>‡</sup>	1.93232(0.06606) <sup>‡</sup>	3.39846(0.01032) <sup>‡</sup>	3.34062(0.01181) <sup>‡</sup>	3.35233(0.01577) <sup>‡</sup>	<b>3.43260(0.00521)</b>
	(20, 21)	3.44713(0.01150) <sup>‡</sup>	1.94073(0.06526) <sup>‡</sup>	3.44152(0.00887) <sup>‡</sup>	3.40309(0.01013) <sup>‡</sup>	3.41177(0.01033) <sup>‡</sup>	<b>3.45812(0.00278)</b>

<sup>‡</sup> and <sup>†</sup> indicate  $d$ CMOEA performs significantly better than and equivalently to the corresponding algorithm, respectively.

TABLE III  
MEAN AND STANDARD DEVIATION VALUES OF IGD METRIC OBTAINED BY SIX ALGORITHMS

Ins	$(\tau_i, n_i)$	DC-MOEA	DC-NSGA-II-A	DC-NSGA-II	DC-NSGA-III	DC-TAEA	$d$ CMOEA
1	(10, 21)	0.36367(0.02941) <sup>‡</sup>	0.55258(0.02763) <sup>‡</sup>	0.10930(0.00523) <sup>‡</sup>	0.14410(0.00597) <sup>‡</sup>	0.16457(0.01049) <sup>‡</sup>	<b>0.09596(0.00535)</b>
	(15, 21)	0.16887(0.01251) <sup>‡</sup>	0.56751(0.02677) <sup>‡</sup>	0.06083(0.00205) <sup>‡</sup>	0.08358(0.00422) <sup>‡</sup>	0.08888(0.00544) <sup>‡</sup>	<b>0.05435(0.00281)</b>
	(20, 21)	0.09441(0.00773) <sup>‡</sup>	0.55722(0.03576) <sup>‡</sup>	0.04445(0.00222) <sup>‡</sup>	0.05735(0.00223) <sup>‡</sup>	0.05993(0.00304) <sup>‡</sup>	<b>0.04024(0.00137)</b>
2	(10, 21)	0.26860(0.01767) <sup>‡</sup>	0.57932(0.03038) <sup>‡</sup>	0.09618(0.00501) <sup>‡</sup>	0.12791(0.00738) <sup>‡</sup>	0.13335(0.00494) <sup>‡</sup>	<b>0.08331(0.00337)</b>
	(15, 21)	0.13074(0.00684) <sup>‡</sup>	0.58435(0.03085) <sup>‡</sup>	0.05715(0.00192) <sup>‡</sup>	0.07684(0.00375) <sup>‡</sup>	0.07781(0.00333) <sup>‡</sup>	<b>0.05322(0.00248)</b>
	(20, 21)	0.07730(0.00411) <sup>‡</sup>	0.57726(0.03067) <sup>‡</sup>	0.04439(0.00248) <sup>‡</sup>	0.05539(0.00195) <sup>‡</sup>	0.05565(0.00146) <sup>‡</sup>	<b>0.04149(0.00156)</b>
3	(10, 21)	0.30675(0.02202) <sup>‡</sup>	0.57476(0.03933) <sup>‡</sup>	0.11023(0.00652) <sup>‡</sup>	0.14496(0.00823) <sup>‡</sup>	0.16487(0.00842) <sup>‡</sup>	<b>0.09489(0.00486)</b>
	(15, 21)	0.16845(0.01148) <sup>‡</sup>	0.56217(0.03725) <sup>‡</sup>	0.06393(0.00335) <sup>‡</sup>	0.08444(0.00313) <sup>‡</sup>	0.09009(0.00473) <sup>‡</sup>	<b>0.05579(0.00291)</b>
	(20, 21)	0.09344(0.00757) <sup>‡</sup>	0.55948(0.04161) <sup>‡</sup>	0.04667(0.00179) <sup>‡</sup>	0.05834(0.00234) <sup>‡</sup>	0.06222(0.00272) <sup>‡</sup>	<b>0.04207(0.00151)</b>
4	(10, 21)	0.22815(0.01437) <sup>‡</sup>	0.57868(0.03066) <sup>‡</sup>	0.09464(0.00421) <sup>‡</sup>	0.12588(0.00506) <sup>‡</sup>	0.12843(0.00493) <sup>‡</sup>	<b>0.08380(0.00380)</b>
	(15, 21)	0.12929(0.00574) <sup>‡</sup>	0.59578(0.02847) <sup>‡</sup>	0.05494(0.00263) <sup>‡</sup>	0.07368(0.00276) <sup>‡</sup>	0.07368(0.00284) <sup>‡</sup>	<b>0.05071(0.00211)</b>
	(20, 21)	0.07888(0.00367) <sup>‡</sup>	0.57842(0.03002) <sup>‡</sup>	0.04259(0.00156) <sup>‡</sup>	0.05302(0.00195) <sup>‡</sup>	0.05332(0.00175) <sup>‡</sup>	<b>0.03976(0.00154)</b>
5	(10, 21)	0.36912(0.03031) <sup>‡</sup>	0.57969(0.03029) <sup>‡</sup>	0.11566(0.00626) <sup>‡</sup>	0.14714(0.00811) <sup>‡</sup>	0.16843(0.01479) <sup>‡</sup>	<b>0.09758(0.00501)</b>
	(15, 21)	0.17614(0.01750) <sup>‡</sup>	0.57996(0.03099) <sup>‡</sup>	0.07126(0.00391) <sup>‡</sup>	0.08955(0.00650) <sup>‡</sup>	0.09520(0.00541) <sup>‡</sup>	<b>0.06384(0.00282)</b>
	(20, 21)	0.10185(0.01038) <sup>‡</sup>	0.57734(0.03272) <sup>‡</sup>	0.05745(0.00268) <sup>‡</sup>	0.06588(0.00327) <sup>‡</sup>	0.06973(0.00287) <sup>‡</sup>	<b>0.05310(0.00119)</b>
6	(10, 21)	0.25978(0.01747) <sup>‡</sup>	0.59926(0.03289) <sup>‡</sup>	0.10008(0.00373) <sup>‡</sup>	0.12845(0.00566) <sup>‡</sup>	0.13359(0.00700) <sup>‡</sup>	<b>0.08608(0.00406)</b>
	(15, 21)	0.12993(0.00766) <sup>‡</sup>	0.60718(0.03230) <sup>‡</sup>	0.06609(0.00312) <sup>‡</sup>	0.08190(0.00311) <sup>‡</sup>	0.08209(0.00286) <sup>‡</sup>	<b>0.06087(0.00168)</b>
	(20, 21)	0.08208(0.00304) <sup>‡</sup>	0.59382(0.03105) <sup>‡</sup>	0.05554(0.00177) <sup>‡</sup>	0.06221(0.00166) <sup>‡</sup>	0.06465(0.00227) <sup>‡</sup>	<b>0.05372(0.00315)</b>
7	(10, 21)	0.29880(0.02390) <sup>‡</sup>	0.58474(0.03000) <sup>‡</sup>	0.11773(0.00752) <sup>‡</sup>	0.14594(0.00744) <sup>‡</sup>	0.17153(0.01266) <sup>‡</sup>	<b>0.09728(0.00678)</b>
	(15, 21)	0.17010(0.01270) <sup>‡</sup>	0.58376(0.03171) <sup>‡</sup>	0.07361(0.00407) <sup>‡</sup>	0.09092(0.00598) <sup>‡</sup>	0.09753(0.00605) <sup>‡</sup>	<b>0.06482(0.00238)</b>
	(20, 21)	0.09553(0.00590) <sup>‡</sup>	0.57417(0.03518) <sup>‡</sup>	0.05955(0.00345) <sup>‡</sup>	0.06847(0.00317) <sup>‡</sup>	0.07274(0.00344) <sup>‡</sup>	<b>0.05456(0.00105)</b>
8	(10, 21)	0.23048(0.01577) <sup>‡</sup>	0.61377(0.03536) <sup>‡</sup>	0.09832(0.00359) <sup>‡</sup>	0.12798(0.00568) <sup>‡</sup>	0.13090(0.00707) <sup>‡</sup>	<b>0.08715(0.00368)</b>
	(15, 21)	0.13034(0.00783) <sup>‡</sup>	0.61393(0.03757) <sup>‡</sup>	0.06507(0.00260) <sup>‡</sup>	0.07936(0.00281) <sup>‡</sup>	0.07983(0.00349) <sup>‡</sup>	<b>0.06048(0.00254)</b>
	(20, 21)	0.08536(0.00337) <sup>‡</sup>	0.60355(0.03762) <sup>‡</sup>	0.05362(0.00216) <sup>‡</sup>	0.06088(0.00219) <sup>‡</sup>	0.06320(0.00213) <sup>‡</sup>	<b>0.05185(0.00136)</b>

second and sixth test problems with  $n_i=21$  and  $\tau_i=15$  and the fifth test problem with  $n_i=21$  and  $\tau_i=20$ , however, DC-NSGA-II shows a performance similar to that of  $d$ CMOEA. This suggests that DC-NSGA-II is promising for problems with slow changes, which may be because DC-NSGA-II has a slow convergence capability. For all test problems, DC-MOEA, DC-NSGA-II-A, DC-NSGA-III, and DC-TAEA fail to show appealing performance in terms of the SP metric.

To judge the algorithms' tracking capability, we also plotted the final POFs of all algorithms over 21 time windows, as presented in Section S-IX of the supplementary material. The figures evidently show that  $d$ CMOEA is very capable of tracking the true POF in each environment.

## VI. DISCUSSION

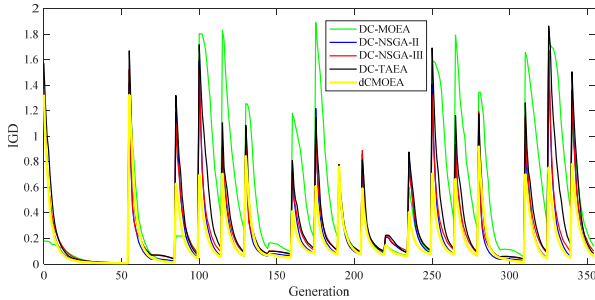
### A. Effectiveness of Each Component of $d$ CMOEA

This subsection investigates the effects of different components of  $d$ CMOEA, including three key components, i.e.,

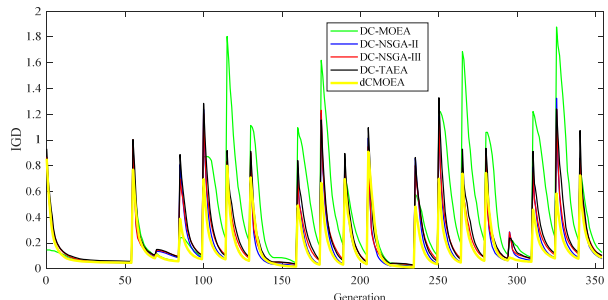
the mating selection operator, the population selection operator, and the change response strategy. To examine the effectiveness of each component of  $d$ CMOEA, we adapted the original  $d$ CMOEA into six variants. The first variant ( $d$ CMOEA-S1) uses the mating selection operator designed in [31] to replace the one proposed in this paper. In the second variant ( $d$ CMOEA-S2), the population selection operator proposed in this paper is replaced with that proposed in [31].  $d$ CMOEA-S3 is the third variant, in which the population selection operator designed in [24] is used to update the population.  $d$ CMOEA-S4 is the fourth variant, in which Equations (19)-(23), for updating the retained previous solutions, are discarded. In the fifth variant ( $d$ CMOEA-S5), all solutions are regenerated randomly when a change is detected. The last variant of  $d$ CMOEA ( $d$ CMOEA-S6) uses the change response strategy proposed in [24] in place of the one designed in this paper. All variants were compared with  $d$ CMOEA on the test problems with settings of  $(\tau_i, n_i)=(10, 21)$ . The average values and standard deviations of the three performance metrics (i.e., HV, IGD, and SP) for

TABLE IV  
MEAN AND STANDARD DEVIATION VALUES OF SP METRIC OBTAINED BY SIX ALGORITHMS

Ins	$(\pi, n_i)$	DC-MOEA	DC-NSGA-II-A	DC-NSGA-II	DC-NSGA-III	DC-TAEA	dCMOEA
1	(10, 21)	0.14115(0.02533) <sup>‡</sup>	0.29109(0.11887) <sup>‡</sup>	0.04131(0.00840) <sup>‡</sup>	0.12395(0.01604) <sup>‡</sup>	0.07438(0.01458) <sup>‡</sup>	<b>0.02845(0.00260)</b>
	(15, 21)	0.05362(0.00779) <sup>‡</sup>	0.24625(0.05206) <sup>‡</sup>	0.02096(0.00185) <sup>‡</sup>	0.07930(0.01323) <sup>‡</sup>	0.03170(0.00553) <sup>‡</sup>	<b>0.01655(0.00181)</b>
	(20, 21)	0.02990(0.00363) <sup>‡</sup>	0.25188(0.06279) <sup>‡</sup>	0.01661(0.00134) <sup>‡</sup>	0.06525(0.01930) <sup>‡</sup>	0.02186(0.00183) <sup>‡</sup>	<b>0.01381(0.00137)</b>
2	(10, 21)	0.11311(0.02056) <sup>‡</sup>	0.31757(0.10076) <sup>‡</sup>	0.03514(0.00633) <sup>‡</sup>	0.14055(0.02545) <sup>‡</sup>	0.05230(0.00653) <sup>‡</sup>	<b>0.02529(0.00202)</b>
	(15, 21)	0.04027(0.00336) <sup>‡</sup>	0.29466(0.09136) <sup>‡</sup>	0.01900(0.00174) <sup>‡</sup>	0.09175(0.02244) <sup>‡</sup>	0.02751(0.00448) <sup>‡</sup>	<b>0.01636(0.00154)</b>
	(20, 21)	0.02353(0.00190) <sup>‡</sup>	0.27475(0.06815) <sup>‡</sup>	0.01506(0.00129) <sup>‡</sup>	0.06547(0.01725) <sup>‡</sup>	0.02040(0.00154) <sup>‡</sup>	<b>0.01412(0.00125)</b>
3	(10, 21)	0.14194(0.02043) <sup>‡</sup>	0.26972(0.07586) <sup>‡</sup>	0.04599(0.00890) <sup>‡</sup>	0.12445(0.02877) <sup>‡</sup>	0.08431(0.01878) <sup>‡</sup>	<b>0.02899(0.00188)</b>
	(15, 21)	0.06409(0.00839) <sup>‡</sup>	0.25171(0.06333) <sup>‡</sup>	0.02279(0.00179) <sup>‡</sup>	0.08522(0.02013) <sup>‡</sup>	0.03442(0.00950) <sup>‡</sup>	<b>0.01803(0.00170)</b>
	(20, 21)	0.03755(0.00704) <sup>‡</sup>	0.26705(0.08285) <sup>‡</sup>	0.01810(0.00157) <sup>‡</sup>	0.06917(0.01913) <sup>‡</sup>	0.02397(0.00194) <sup>‡</sup>	<b>0.01582(0.00141)</b>
4	(10, 21)	0.10575(0.01617) <sup>‡</sup>	0.29303(0.06676) <sup>‡</sup>	0.03423(0.00594) <sup>‡</sup>	0.14095(0.02076) <sup>‡</sup>	0.05036(0.00875) <sup>‡</sup>	<b>0.02505(0.00213)</b>
	(15, 21)	0.03817(0.00336) <sup>‡</sup>	0.30270(0.06437) <sup>‡</sup>	0.01876(0.00350) <sup>‡</sup>	0.09487(0.01947) <sup>‡</sup>	0.02497(0.00284) <sup>‡</sup>	<b>0.01494(0.00130)</b>
	(20, 21)	0.02256(0.00192) <sup>‡</sup>	0.29209(0.07005) <sup>‡</sup>	0.01414(0.00116) <sup>‡</sup>	0.06928(0.01617) <sup>‡</sup>	0.01855(0.00140) <sup>‡</sup>	<b>0.01297(0.00160)</b>
5	(10, 21)	0.14139(0.02591) <sup>‡</sup>	0.27849(0.07592) <sup>‡</sup>	0.04184(0.00569) <sup>‡</sup>	0.12926(0.01884) <sup>‡</sup>	0.08197(0.01788) <sup>‡</sup>	<b>0.02881(0.00259)</b>
	(15, 21)	0.05640(0.00986) <sup>‡</sup>	0.29118(0.11025) <sup>‡</sup>	0.02238(0.00182) <sup>‡</sup>	0.08582(0.01797) <sup>‡</sup>	0.03727(0.01080) <sup>‡</sup>	<b>0.01746(0.00142)</b>
	(20, 21)	0.03066(0.00231) <sup>‡</sup>	0.30525(0.10851) <sup>‡</sup>	0.01784(0.00130) <sup>‡</sup>	0.05947(0.01589) <sup>‡</sup>	0.02348(0.00328) <sup>‡</sup>	<b>0.01432(0.00106)</b>
6	(10, 21)	0.12201(0.02190) <sup>‡</sup>	0.26981(0.06150) <sup>‡</sup>	0.03523(0.00380) <sup>‡</sup>	0.12998(0.02777) <sup>‡</sup>	0.05811(0.01227) <sup>‡</sup>	<b>0.02565(0.00297)</b>
	(15, 21)	0.04200(0.00352) <sup>‡</sup>	0.32059(0.08405) <sup>‡</sup>	0.01973(0.00145) <sup>‡</sup>	0.08668(0.02374) <sup>‡</sup>	0.02844(0.00256) <sup>‡</sup>	<b>0.01454(0.00139)</b>
	(20, 21)	0.02490(0.00172) <sup>‡</sup>	0.29533(0.04826) <sup>‡</sup>	0.01571(0.00123) <sup>‡</sup>	0.07367(0.01697) <sup>‡</sup>	0.02175(0.00233) <sup>‡</sup>	<b>0.01416(0.00097)</b>
7	(10, 21)	0.15437(0.02601) <sup>‡</sup>	0.28825(0.08546) <sup>‡</sup>	0.04275(0.00577) <sup>‡</sup>	0.13591(0.02666) <sup>‡</sup>	0.08314(0.01816) <sup>‡</sup>	<b>0.02962(0.00218)</b>
	(15, 21)	0.06414(0.00963) <sup>‡</sup>	0.26432(0.06374) <sup>‡</sup>	0.02480(0.00231) <sup>‡</sup>	0.09053(0.01809) <sup>‡</sup>	0.04084(0.00871) <sup>‡</sup>	<b>0.01932(0.00160)</b>
	(20, 21)	0.03762(0.00478) <sup>‡</sup>	0.27550(0.08123) <sup>‡</sup>	0.01957(0.00135) <sup>‡</sup>	0.06869(0.01996) <sup>‡</sup>	0.02567(0.00279) <sup>‡</sup>	<b>0.01663(0.00112)</b>
8	(10, 21)	0.11245(0.02100) <sup>‡</sup>	0.31584(0.08201) <sup>‡</sup>	0.03492(0.00513) <sup>‡</sup>	0.14087(0.02686) <sup>‡</sup>	0.05386(0.00959) <sup>‡</sup>	<b>0.02557(0.00186)</b>
	(15, 21)	0.03912(0.00413) <sup>‡</sup>	0.27246(0.04382) <sup>‡</sup>	0.01870(0.00167) <sup>‡</sup>	0.10270(0.02206) <sup>‡</sup>	0.02785(0.00579) <sup>‡</sup>	<b>0.01559(0.00142)</b>
	(20, 21)	0.02379(0.00246) <sup>‡</sup>	0.29940(0.08554) <sup>‡</sup>	0.01496(0.00122) <sup>‡</sup>	0.07737(0.01465) <sup>‡</sup>	0.01945(0.00177) <sup>‡</sup>	<b>0.01294(0.00158)</b>



(a)



(b)

Fig. 4. Evolution curves of average IGD values for the first two test problems with  $\tau_r=15$  and  $n_r=21$ .

dCMOEA and its variants are given in Table V.

As seen from Table V, dCMOEA performs significantly better than its variants on most of the test problems in terms of all three metrics, implying that these three key components are crucial for improving the performance of dCMOEA on DCMOPs. The IGD values of dCMOEA on the last four test problems are small, illustrating that dCMOEA has a better convergence capability for DCMOPs with oscillating objective function modes. Notably, dCMOEA-S3 shows similar performance in terms of the IGD metric on the first four test and the eighth problems, mainly because dCMOEA-S3 and dCMOEA adopt the similar constraint handling techniques that are from [15]. In addition, dCMOEA achieves quite small SP values on all test problems, implying that dCMOEA can search for nondominated solutions with a good distribution for DCMOPs. Notably, dCMOEA-S5 shows similar performance in terms of the SP metric on the sixth test problem, mainly due to the diversity of the randomly generated solutions.

In dCMOEA-S1, feasible solutions can be more easily chosen for generating offspring. However, many feasible solutions are far from the true POF. If there are many such solutions in the parent population, then more poor children are generated. Therefore, the poor performance of dCMOEA-S1 may be due to a large number of poor feasible solutions in the parent population.

dCMOEA-S2 achieves the worst performance among the compared variants on all test problems, which may be because

infeasible solutions are not considered if there are sufficient feasible solutions. Indeed, some infeasible solutions close to the true POF may carry promising population information and generate competitive offspring. Simply abandoning them may reduce the population diversity. The poor performance of dCMOEA-S3 on some test problems may be attributable to the fact that fewer feasible solutions are retained when the feasible region is small. Note that the solutions in the POF must be feasible. Fewer feasible solutions reduce the number of nondominated solutions in the POF, decreasing the HV values and increasing the IGD and SP values.

dCMOEA-S4 does not consider any information on the new POF when an environmental change is detected, resulting in a poor initial population in the search for nondominated solutions. By contrast, when a change arises, dCMOEA-S5 completely reinitializes the population. For a DCMOP, some previously dominated solutions may become nondominated solutions in the new environment. The initial population obtained by dCMOEA-S5 has good diversity, but it miss many promising previous solutions. In dCMOEA-S6, the retained solutions are not updated. However, the previous feasible region may become infeasible in the new environment, thus increasing the number of infeasible solutions. Therefore, dCMOEA-S6 fails to track the new POF.

In summary, by combining the three key components proposed in dCMOEA, dCMOEA outperforms all compared variants, showing that each component of dCMOEA plays an

TABLE V  
PERFORMANCE COMPARISON OF dCMOEA AND ITS VARIANTS WITH  $(\tau, n_s)=(10, 21)$

Ins	Indicator	dCMOEA-S1	dCMOEA-S2	dCMOEA-S3	dCMOEA-S4	dCMOEA-S5	dCMOEA-S6	dCMOEA
1	HV	3.23600(0.02333) <sup>‡</sup>	3.09479(0.06197) <sup>‡</sup>	<b>3.34999(0.01763)<sup>†</sup></b>	3.24582(0.02456) <sup>‡</sup>	3.34134(0.01844) <sup>†</sup>	2.94287(0.06076) <sup>‡</sup>	3.34912(0.01415)
	IGD	0.13381(0.00723) <sup>‡</sup>	0.16633(0.02138) <sup>‡</sup>	<b>0.09321(0.00566)</b>	0.13354(0.00885) <sup>‡</sup>	0.09738(0.00539) <sup>†</sup>	0.24439(0.02179) <sup>‡</sup>	0.09596(0.00535)
	SP	0.04339(0.00659) <sup>‡</sup>	0.09258(0.02439) <sup>‡</sup>	<b>0.03764(0.00385)<sup>‡</sup></b>	0.03535(0.00361) <sup>‡</sup>	0.03042(0.00291) <sup>‡</sup>	0.04714(0.00536) <sup>‡</sup>	<b>0.02845(0.00260)</b>
2	HV	3.28356(0.01647) <sup>‡</sup>	3.08864(0.04281) <sup>‡</sup>	3.36164(0.01060) <sup>‡</sup>	3.29059(0.01483) <sup>‡</sup>	3.35859(0.01104) <sup>‡</sup>	3.05064(0.05367) <sup>‡</sup>	<b>3.37086(0.00965)</b>
	IGD	0.11552(0.00483) <sup>‡</sup>	0.16810(0.01697) <sup>‡</sup>	0.08449(0.00376) <sup>‡</sup>	0.11485(0.00560) <sup>‡</sup>	0.08666(0.00293) <sup>‡</sup>	0.20389(0.01896) <sup>‡</sup>	<b>0.08331(0.00337)</b>
	SP	0.03791(0.00598) <sup>‡</sup>	0.09838(0.03125) <sup>‡</sup>	0.03393(0.00294) <sup>‡</sup>	0.03169(0.00274) <sup>‡</sup>	0.02725(0.00196) <sup>‡</sup>	0.04325(0.00450) <sup>‡</sup>	<b>0.02529(0.00202)</b>
3	HV	3.24221(0.03143) <sup>‡</sup>	3.03914(0.05939) <sup>‡</sup>	3.34662(0.01817) <sup>‡</sup>	3.24791(0.02973) <sup>‡</sup>	3.33345(0.01673) <sup>‡</sup>	2.92370(0.06056) <sup>‡</sup>	<b>3.35154(0.01319)</b>
	IGD	0.13353(0.01137) <sup>‡</sup>	0.18738(0.02449) <sup>‡</sup>	0.09453(0.00621) <sup>‡</sup>	0.13301(0.01042) <sup>‡</sup>	0.09968(0.00496) <sup>‡</sup>	0.24971(0.02220) <sup>‡</sup>	<b>0.09419(0.00486)</b>
	SP	0.04412(0.00665) <sup>‡</sup>	0.11218(0.03120) <sup>‡</sup>	0.03975(0.00302) <sup>‡</sup>	0.03826(0.00257) <sup>‡</sup>	0.03201(0.00267) <sup>‡</sup>	0.04808(0.00574) <sup>‡</sup>	<b>0.02899(0.00188)</b>
4	HV	3.26071(0.01831) <sup>‡</sup>	3.10608(0.04318) <sup>‡</sup>	3.34965(0.01406) <sup>‡</sup>	3.28397(0.01422) <sup>‡</sup>	3.35259(0.00967) <sup>‡</sup>	3.03569(0.07549) <sup>‡</sup>	<b>3.35571(0.01001)</b>
	IGD	0.12000(0.00548) <sup>‡</sup>	0.15818(0.01378) <sup>‡</sup>	<b>0.08258(0.00481)<sup>†</sup></b>	0.11280(0.00493) <sup>‡</sup>	0.08440(0.00292) <sup>‡</sup>	0.20570(0.02710) <sup>‡</sup>	0.08380(0.00380)
	SP	0.03653(0.00462) <sup>‡</sup>	0.11037(0.02822) <sup>‡</sup>	0.03310(0.00297) <sup>‡</sup>	0.03100(0.00247) <sup>‡</sup>	0.02616(0.00223) <sup>‡</sup>	0.04201(0.00520) <sup>‡</sup>	<b>0.02505(0.00213)</b>
5	HV	3.26599(0.02574) <sup>‡</sup>	3.11162(0.05083) <sup>‡</sup>	3.36187(0.01620) <sup>‡</sup>	3.27393(0.02094) <sup>‡</sup>	3.35415(0.01645) <sup>‡</sup>	2.95660(0.06020) <sup>‡</sup>	<b>3.37017(0.01575)</b>
	IGD	0.13124(0.00737) <sup>‡</sup>	0.17254(0.01723) <sup>‡</sup>	0.09908(0.00471) <sup>‡</sup>	0.13126(0.00754) <sup>‡</sup>	0.10186(0.00478) <sup>‡</sup>	0.24972(0.02242) <sup>‡</sup>	<b>0.09758(0.00501)</b>
	SP	0.04380(0.00604) <sup>‡</sup>	0.08651(0.02872) <sup>‡</sup>	0.03967(0.00702) <sup>‡</sup>	0.03822(0.00406) <sup>‡</sup>	0.03227(0.00253) <sup>‡</sup>	0.04892(0.00585) <sup>‡</sup>	<b>0.02881(0.00259)</b>
6	HV	3.29351(0.01474) <sup>‡</sup>	3.11023(0.05243) <sup>‡</sup>	3.36331(0.01023) <sup>‡</sup>	3.29265(0.02000) <sup>‡</sup>	3.35368(0.01153) <sup>‡</sup>	3.08168(0.06303) <sup>‡</sup>	<b>3.36944(0.01193)</b>
	IGD	0.11368(0.00484) <sup>‡</sup>	0.16535(0.01919) <sup>‡</sup>	0.08872(0.00374) <sup>‡</sup>	0.11474(0.00671) <sup>‡</sup>	0.09106(0.00353) <sup>‡</sup>	0.19333(0.02293) <sup>‡</sup>	<b>0.08608(0.00406)</b>
	SP	0.03697(0.00388) <sup>‡</sup>	0.10095(0.02731) <sup>‡</sup>	0.03479(0.00270) <sup>‡</sup>	0.03282(0.00311) <sup>‡</sup>	0.03001(0.00410) <sup>‡</sup>	0.04270(0.00513) <sup>‡</sup>	<b>0.02565(0.00207)</b>
7	HV	3.24416(0.04267) <sup>‡</sup>	3.06303(0.07185) <sup>‡</sup>	3.34695(0.02612) <sup>‡</sup>	3.26656(0.03718) <sup>‡</sup>	3.34290(0.01990) <sup>‡</sup>	2.99122(0.07194) <sup>‡</sup>	<b>3.37201(0.02073)</b>
	IGD	0.13876(0.01520) <sup>‡</sup>	0.19527(0.02842) <sup>‡</sup>	0.10506(0.00857) <sup>‡</sup>	0.13253(0.01313) <sup>‡</sup>	0.10523(0.00514) <sup>‡</sup>	0.23428(0.02559) <sup>‡</sup>	<b>0.09728(0.00678)</b>
	SP	0.04229(0.00459) <sup>‡</sup>	0.08423(0.02570) <sup>‡</sup>	0.04103(0.00437) <sup>‡</sup>	0.03980(0.00407) <sup>‡</sup>	0.03539(0.00387) <sup>‡</sup>	0.04763(0.00529) <sup>‡</sup>	<b>0.02962(0.00218)</b>
8	HV	3.25934(0.02773) <sup>‡</sup>	3.09790(0.06117) <sup>‡</sup>	3.35078(0.01355) <sup>‡</sup>	3.28343(0.01573) <sup>‡</sup>	3.34494(0.01371) <sup>‡</sup>	3.02581(0.06788) <sup>‡</sup>	<b>3.35664(0.01033)</b>
	IGD	0.12095(0.00883) <sup>‡</sup>	0.16716(0.02083) <sup>‡</sup>	0.08733(0.00393) <sup>‡</sup>	0.11382(0.00580) <sup>‡</sup>	0.09054(0.00382) <sup>‡</sup>	0.20989(0.02456) <sup>‡</sup>	<b>0.08715(0.00368)</b>
	SP	0.03863(0.00520) <sup>‡</sup>	0.09564(0.03359) <sup>‡</sup>	0.03351(0.00336) <sup>‡</sup>	0.03215(0.00249) <sup>‡</sup>	0.02816(0.00240) <sup>‡</sup>	0.04590(0.00583) <sup>‡</sup>	<b>0.02557(0.00186)</b>

important role in dealing with DCMOPs. Here, we would like to explain the role of each key component in more detail. The mating selection operator chooses the parents in accordance with the modified objective functions, and infeasible solutions with small constraint violations and objective function values are given a high probability of selection, thereby improving the diversity of the parents and allowing dCMOEA to utilize infeasible solutions efficiently and effectively to generate offspring with appealing performance. The population selection strategy can balance the convergence rate of the population with the population diversity. This is because when the number of feasible solutions becomes small, dCMOEA can retain all feasible solutions, thus helping to rapidly track the new POF and POS, whereas when the number of feasible solutions is large, infeasible solutions with small constraint violations and objective function values are propagated to the next generation, thereby diversifying the population. Introducing random solutions in the change response stage can allow the algorithm to explore more information on the new environment to search for members of the population close to the new POF, and the update strategy for the retained previous solutions is beneficial for convergence when the new true POF moves. In summary, these three key components of dCMOEA play important roles in solving DCMOPs.

### B. Influence of the Number of Decision Variables

Our preliminary experimental results show that the computation time needed to obtain a feasible solution increases as the number of decision variables increases. Indeed, for the test problems with the current parameters, the computation time is unacceptable when the number of decision variables is more than 15. To examine the performance of dCMOEA on instances with 15 and 20 decision variables, we revised  $z(t)$ , which controls the upper bound of the feasible region for the designed instances. For test problems with 15 decision variables,  $z(t)$  was revised to 8. The value of  $z(t)$  in test instances with 20 decision variables was revised to 12. Considering that DNSGA-II-A, DC-MOEA, and DC-TAEA performed poorly in the above experiments, for the experiments reported in this subsection, the test problems with 15 and 20 decision variables were solved

using only DC-NSGA-II, DC-NSGA-III, and dCMOEA. The values and standard deviations of the three performance metrics (i.e., HV, IGD, and MS) for these algorithms on the test problems with 15 and 20 decision variables are listed in Section S-X of the supplementary material.

It can be observed from the corresponding tables that dCMOEA achieves the best performance on these instances among the three tested algorithms, thus demonstrating that dCMOEA can successfully solve DCMOPs with up to 20 decision variables.

## VII. CONCLUSION

In a DCMOP, the true POF is determined by the unconstrained POF and feasible region. This paper proposes a set of test instances for DCMOPs that consider simultaneous changes in the feasible region and unconstrained POF. The unconstrained POFs in the proposed test problems exhibit two modes. In one mode, their shapes remain unchanged with dynamism; in the other mode, the shapes of POFs oscillate among several optimization modes. During optimization, the feasible region may increase or decrease with dynamism, causing the true POF to become disconnected. The proposed test problems can be used to judge an algorithm's capability of tracking the new POF and handling infeasible solutions. Furthermore, six dynamic constrained MOEAs for optimization were tested on eight test problems, and the results were evaluated in terms of three performance metrics. The comparison shows that the designed instances are effective and useful for distinguishing the performance of each algorithm.

We also propose an algorithm called dCMOEA for handling MOPs with time-varying constraints and objective functions. In dCMOEA, the mating selection and population selection operators apply the constraint handling mechanism proposed in [15], allowing infeasible solutions with small constraint violations and objective function values to play a useful role in searching for nondominated solutions. The proposed selection operator can adaptively select both feasible and infeasible solutions for inclusion in the population, thus balancing the trade-off between the population diversity and convergence capability of dCMOEA. When a change is detected, dCMOEA

reuses some previous solutions with small fitness values and re-evaluates them. Afterward, *dCMOEA* updates these previous solutions on the basis of useful information obtained from the new environment, i.e., the feasibility ratio and the newly obtained nondominated solutions, to provide a reinitialized population for the algorithm.

Despite that *dCMOEA* has shown appealing performance on the proposed test problems, this paper has several aspects that limit its applicability in certain situations. The parameter settings for  $s(t)$ ,  $m(t)$ , and  $z(t)$  used in the proposed benchmark problems depend heavily on trial and error. Moreover, this study does not use *dCMOEA* to solve the DCMOPs with three or many objectives. Regarding the change response, *dCMOEA* is designed for tracking the new true POF with no consideration about the transition cost of a new solution.

Considering that scalability plays a crucial role in designing an efficient algorithm [55], in the future we will study the scalable dynamic constrained many-objective optimization algorithms and design the benchmark problems that are close to real-world applications and can easily be extended to three or more objective functions. In many real-world applications, changing the production solution introduces additional cost [64]. Thus, we will borrow the ideas proposed in [57] to design a general framework that can find robust solutions. In addition, there are other types of changes in the true POF and the feasible region of objective functions, which we will study in the future.

#### REFERENCES

- [1] K. Deb, N. U. B. Rao, and S. Karthik, "Dynamic multiobjective optimization and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling," in *Proc. 4th Int. Conf. Evol. Multi Criterion Optim.*, vol. 3, 2007, pp. 803–817.
- [2] M. Farina, K. Deb, and P. Amato, "Dynamic multiobjective optimization problems: Test cases, approximations, and applications," *IEEE Trans. Evol. Comput.*, vol. 8, no. 5, pp. 425–442, Oct. 2004.
- [3] S. B. Gee, K. C. Tan, and H. A. Abbass, "A benchmark test suite for dynamic evolutionary multiobjective optimization," *IEEE Trans. Cyber.*, vol. 47, no. 2, pp. 461–472, Feb. 2017.
- [4] S. Y. Jiang and S. X. Yang, "Evolutionary dynamic multiobjective optimization benchmarks and algorithm comparisons," *IEEE Trans. Cyber.*, vol. 47, no. 1, pp. 198–211, Jan. 2017.
- [5] Y. Jin and B. Sendhoff, "Constructing dynamic optimization test problems using the multiobjective optimization concept," in *Proc. EvoWorkshops Appl. Evol. Comput.*, 2004, pp. 525–536.
- [6] K. Deb, A. Pratap, and T. Meyarivan, "Constrained test problems for multiobjective evolutionary optimization," in *Proc. 1st Int. Conf. Evolutionary Multi-Criterion Optimization*, 2001, pp. 284–298.
- [7] S. Biswas, S. Das, P. N. Suganthan, and C. A. Coello Coello, "Evolutionary multiobjective optimization in dynamic environments: A set of novel benchmark functions," in *Proc. IEEE Congr. Evol. Comput.*, Beijing, China, 2014, pp. 3192–3199.
- [8] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.
- [9] C. A. C. Coello and M. R. Sierra, "A study of the parallelization of a coevolutionary multiobjective evolutionary algorithm," in *Proc. Mexican Int. Conf. Artif. Intell.*, 2004, pp. 688–697.
- [10] C. K. Goh and K. C. Tan, "An investigation on noisy environment in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 11, no. 3, pp. 354–381, Jun. 2007.
- [11] C. K. Goh and K. C. Tan, "A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 1, pp. 103–127, Feb. 2009.
- [12] K. Deb, "Multiobjective genetic algorithms: Problem difficulties and construction of test problems," *Evol. Comput.*, vol. 7, no. 3, pp. 205–230, 1999.
- [13] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Aug. 2002.
- [14] S. Venkatraman, and G. G. Yen, "Genetic framework for constrained optimization using genetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 9, no. 4, pp. 424–435, Aug. 2005.
- [15] 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.
- [16] A. Zhou, Y. Jin, and Q. Zhang, "A population prediction strategy for evolutionary dynamic multiobjective optimization," *IEEE Trans. Cyber.*, vol. 44, no. 1, pp. 40–53, Jan. 2014.
- [17] Y. Wu, Y. Jin, and X. Liu, "A directed search strategy for evolutionary dynamic multiobjective optimization," *Soft Comput.*, vol. 19, no. 11, pp. 3221–3235, Nov. 2015.
- [18] S. Y. Jiang and S. X. Yang, "A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 65–82, Feb. 2017.
- [19] S. B. Gee, K. C. Tan, and C. Alippi, "Solving multiobjective optimization problems in unknown dynamic environments: an inverse modeling approach," *IEEE Trans. Cyber.*, vol. 47, no. 2, pp. 4223–4234, Dec. 2017.
- [20] M. Jiang, Z. Q. Huang, L. M. Qiu, W. Z. Huang, and G. G. Yen, "Transfer learning-based dynamic multiobjective optimization algorithms," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 501–514, Aug. 2018.
- [21] K. Li, R. Z. Chen, G. T. 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.
- [22] D. Gong, B. Xu, Y. Zhang, Y. Guo, and S. Yang, "A Similarity-based cooperative co-evolutionary algorithm for dynamic interval multi-objective optimization problems," *IEEE Trans. Evol. Comput.*, 2019, to be published. DOI: 10.1109/TEVC.2019.2912204.
- [23] M. Rong, D. W. Gong, Y. Zhang, Y. C. Jin, and W. Pedrycz, "Multidirectional prediction approach for dynamic multiobjective optimization problems," *IEEE Trans. Cyber.*, vol. 49, no. 9, pp. 3362–3374, Sept. 2019.
- [24] 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.
- [25] R. Azzouz, S. Bechikh, and L. Ben Said, "Multiobjective optimization with dynamic constraints and objective: new challenges for evolutionary algorithm," in *Proc. Annu. Conf. Genet. Evol. Comput. (GECCO)*, 2016, pp. 615–622.
- [26] C. A. C. Coello and A. D. Christiansen, "MOSES: A multiobjective optimization tool for engineering design," *Eng. Opt.*, vol. 31, no. 3, pp. 337–368, 1999.
- [27] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, Oct. 2016.
- [28] M. Cámara, J. Ortega, and F. de Toro, "A single front genetic algorithm for parallel multiobjective optimization in dynamic environments," *Neurocomputing*, vol. 72, no. 16–18, pp. 3570–3579, Oct. 2009.
- [29] T. Takahama and S. Sakai, "Efficient constrained optimization by the  $\epsilon$ -constrained rank-based differential evolution," in *Proc. IEEE World Congr. Comput. Intell.*, Jun. 2012, pp. 1–8.
- [30] M. Asafuddoula, T. Ray, and R. A. Sarker, "A decomposition-based evolutionary algorithm for many objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 3, pp. 445–460, Jun. 2015.
- [31] 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.
- [32] Z. Fan, W. Li, X. Cai, K. Hu, H. Lin, and H. Li, "Angle-based constrained dominance principle in MOEA/D for constrained multiobjective optimization problems," in *Proc. IEEE Congr. Evol. Comput.*, 2016, pp. 460–467.
- [33] A. Angantyr, J. Andersson, and J.-O. Aidanpaa, "Constrained optimization based on a multiobjective evolutionary algorithm," in *Proc. IEEE Congr. Evol. Comput.*, 2003, pp. 1560–1567.
- [34] N. Young, "Blended ranking to cross infeasible regions in constrained multiobjective problems," in *Proc. Int. Conf. Comput. Intell. Model. Control Autom.*, 2005, pp. 191–196.
- [35] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary manyobjective optimization algorithm based on dominance and decomposition," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 694–716, Dec. 2015.
- [36] C. Peng, H. Liu, and F. Gu, "An evolutionary algorithm with directed weights for constrained multiobjective optimization," *Appl. Soft Comput.*,

- vol. 60, pp. 613–622, Nov. 2017
- [37] W. Ning, B. Guo, Y. Yan, X. Wu, J. Wu, and D. Zhao, “Constrained multiobjective optimization using constrained nondominated sorting combined with an improved hybrid multiobjective evolutionary algorithm,” *Eng. Opt.*, vol. 49, no. 10, pp. 1645–1664, Jan. 2017.
- [38] A. E. Sorkhabi, M. D. Amiri, and A. R. Khanteymoori, “Duality evolution: an efficient approach to constraint handling in multiobjective particle swarm optimization,” *Soft Comput.*, vol. 21, no. 24, pp. 7251–7267, Dec. 2017.
- [39] K. Harada, J. Sakuma, I. Ono, and S. Kobayashi, “Constraint-handling method for multiobjective function optimization: Pareto descent repair operator,” in *Proc. Int. Conf. Evol. Multi-Criterion Opt.*, 2007, pp. 364–374.
- [40] H. K. Singh, T. Ray, and W. Smith, “C-PSA: constrained pareto simulated annealing for constrained multiobjective optimization,” *Inf. Sci.*, vol. 180, no. 13, pp. 2499–2513, Jul. 2010.
- [41] L. Jiao, J. Luo, R. Shang, and F. Liu, “A modified objective function method with feasible-guiding strategy to solve constrained multiobjective optimization problems,” *Appl. Soft Comput.*, vol. 14, pp. 363–380, Jan. 2014.
- [42] Y. G. Woldesenbet and G. G. Yen, “Dynamic evolutionary algorithm with variable relocation,” *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 500–513, Jun. 2009.
- [43] S. Yang and R. Tinós, “A hybrid immigrants scheme for genetic algorithms in dynamic environments,” *Int. J. Autom. Comput.*, vol. 4, no. 3, pp. 243–254, Jul. 2007.
- [44] S. Yang, “Genetic algorithms with memory-and elitism-based immigrants in dynamic environments,” *Evol. Comput.*, vol. 16, no. 3, pp. 385–416, Sept. 2008.
- [45] M. Mavrouniotis and S. Yang, “Genetic algorithms with adaptive immigrants for dynamic environments,” in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 2130–2137.
- [46] Y. Wang and B. Li, “Multi-strategy ensemble evolutionary algorithm for dynamic multiobjective optimization,” *Memetic Comput.*, vol. 2, no. 1, pp. 3–24, Mar. 2010.
- [47] R. Azzouz, S. Bechikh, and L. B. Said, “A dynamic multiobjective evolutionary algorithm using a change severity-based adaptive population management strategy,” *Soft Comput.*, vol. 21, no. 4, pp. 885–906, Feb. 2017.
- [48] J. P. Li, Y. Wang, S. X. Yang, and Z. X. Cai, “A comparative study of constraint-handling techniques in evolutionary constrained multiobjective optimization,” in *Proc. IEEE Congr. Evol. Comput.*, Vancouver, BC, Canada, 2016, pp. 4175–4182.
- [49] W. Kong, T. Chai, S. Yang, and J. Ding, “A hybrid evolutionary multiobjective optimization strategy for the dynamic power supply problem in magnesium grain manufacturing,” *Appl. Soft Comput.*, vol. 13, no. 5, pp. 2960–2969, 2013.
- [50] R. Liu, J. Fan, and L. Jiao, “Integration of improved predictive model and adaptive differential evolution based dynamic multiobjective evolutionary optimization algorithm,” *Appl. Intell.*, vol. 43, no. 1, pp. 192–207, 2015.
- [51] H. Richter, “Detecting change in dynamic fitness landscapes,” in *Proc. IEEE Congr. Evol. Comput. (CEC)*. Trondheim, Norway, 2009, pp. 1613–1620.
- [52] J. R. Schott, “Fault tolerant design using single and multicriteria genetic algorithm optimization,” M.S. thesis, Dept. Aeronaut. Astronaut., Massachusetts Inst. Technol., Cambridge, MA, USA, 1995.
- [53] F. Wilcoxon, “Individual comparisons by ranking methods,” *Biometrics Bull.*, vol. 1, no. 6, pp. 80–83, 1945.
- [54] K. Deb and R. B. Agrawal, “Simulated binary crossover for continuous search space,” in *Complex Syst.*, Apr. 1995, vol. 9, pp. 115–148.
- [55] D. Yazdani, M. N. Omidvar, J. Branke, T. T. Nguyen, and X. Yao, “Scaling up dynamic optimization problems: a divide-and-conquer approach,” *IEEE Trans. Evol. Comput.*, DOI: 10.1109/TEVC.2019.2902626.
- [56] J. Branke, “Memory enhanced evolutionary algorithms for changing optimization problems,” in *Proc. IEEE Congr. Evol. Comput. (CEC)*, vol. 3. Washington, DC, USA, 1999, pp. 1875–1882.
- [57] D. Yazdani, T. T. Nguyen, and J. Branke, “Robust optimization over time by learning problem space characteristics,” *IEEE Trans. Evol. Comput.*, vol. 23, no. 1, pp. 143–155, Feb. 2019.
- [58] T. P. Runarsson and X. Yao, “Stochastic ranking for constrained evolutionary optimization,” *IEEE Trans. Evol. Comput.*, vol. 4, no. 3, pp. 284–294, Sept. 2000.
- [59] T. P. Runarsson and X. Yao, “Search biases in constrained evolutionary optimization,” *IEEE Trans. Syst., Man, Cybern.*, vol. 35, no. 2, pp. 233–243, May 2005.
- [60] R. Allmendinger and J. Knowles, “On handling ephemeral resource constraints in evolutionary search,” *Evol. Comput.*, vol. 21, no. 3, pp. 497–531, Sept. 2013.
- [61] R. Allmendinger, J. Handl, and J. Knowles, “Multiobjective optimization: When objectives exhibit non-uniform latencies,” *Eur. J. Oper. Res.*, vol. 243, no. 2, pp. 497–513, Jun. 2015.
- [62] H. Liang, Y. Liu, F. Li, and Y. Shen, “Dynamic economic/emission dispatch including PEVs for peak shaving and valley filling,” *IEEE Trans. Ind. Electron.* Vol. 66, no. 4, pp. 2880–2890, Apr. 2019.
- [63] L. T. Bui, J. Branke, and H. A. Abbass, “Multiobjective optimization for dynamic environments,” in *Proc. IEEE Congr. Evol. Comput.*, 2005, pp. 2349–2356.
- [64] D. Yazdani, J. Branke, M. N. Omidvar, T. T. Nguyen, and X. Yao, “Changing or keeping solutions in dynamic optimization problems with switching costs,” in *Proc. Annu. Conf. Genet. Evol. Comput. (GECCO)*, Madrid, Spain, 2018, pp. 1095–1102.



**Qingda Chen** (S'19) received the B.S. degree from Yantai University, Yantai, China, in 2013. Since 2014, he has been working toward the Ph.D. degree in control theory and control engineering at Northeastern University.

His current research interests include computational intelligence and its application in the industrial processes.



**Jinliang Ding** (SM'14) received the Ph.D. degree in control theory and control engineering from Northeastern University, Shenyang, China, in 2012.

His current research interests include modeling, plant-wide control and optimization for the complex industrial systems, stochastic distribution control, and multiobjective evolutionary algorithms and its application.

Prof. Ding was a recipient of the Young Scholars Science and Technology Award of China in 2016, the National Science Fund for Distinguished Young Scholars in 2015 and the National Technological Invention Award in 2013.



**Shengxiang Yang** (M'00–SM'14) received the B.Sc. and M.Sc. degrees in automatic control and the Ph.D. degree in systems engineering from Northeastern University, Shenyang, China in 1993, 1996, and 1999, respectively.

His current research interests include evolutionary and genetic algorithms, swarm intelligence, computational intelligence in dynamic and uncertain environments, artificial neural networks for scheduling, and relevant real-world applications.

Prof. Yang serves as an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON CYBERNETICS etc.



**Tianyou Chai** (M'90–SM'97–F'08) received the Ph.D. degree in control theory and engineering from Northeastern University, Shenyang, China, in 1985, where he became a Professor in 1988 and a Chair Professor in 2004.

His current research interests include adaptive control, intelligent decoupling control, integrated plant control and systems, and the development of control technologies with applications to various industrial processes.

Prof. Chai is a member of the Chinese Academy of Engineering, an academican of International Eurasian Academy of Sciences, and IFAC Fellow.

# A Novel Evolutionary Algorithm for Dynamic Constrained Multiobjective Optimization Problems

## —Supplementary Material

Qingda Chen, *Student Member, IEEE*, Jinliang Ding, *Senior Member, IEEE*,  
Shengxiang Yang, *Senior Member, IEEE*, and Tianyou Chai, *Fellow, IEEE*

This is the supplementary material to the paper entitled “A Novel Evolutionary Algorithm for Dynamic Constrained Multiobjective Optimization Problems”, submitted to **IEEE Transactions on Evolutionary Computation**.

### CONTENTS

- Section S-I** Mathematical functions described in Fig. 1.
- Section S-II** Real-world applications linking to the four types of the proposed problems.
- Section S-III** Ideas on scaling up the proposed test problems
- Section S-IV** True POE and POS of each test problem
- Section S-V** Algorithm parameter selection
- Section S-VI** Performance ranking of each algorithm.
- Section S-VII** Computational time of each algorithm on each test problem.
- Section S-VIII** Evolution curves of average IGD values for the third to eight test problems.
- Section S-IX** The POEs obtained by the compared algorithms on the eight test problems.
- Section S-X** Three performance metrics of algorithms on the test problems with 15 and 20 decision variables.

This work was supported in part by the National Natural Science Foundation of China under Grant 61988101, Grant 61525302, Grant 61590922, and Grant 61673331; in part by the National Key Research and Development Program of China under Grant 2018YFB1701104; and in part by the Xingliao Plan of Liaoning Province under Grant XLYC1808001. (*Corresponding author: Jinliang Ding.*)

Q. Chen, J. Ding, and T. Chai are with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang, 110819, China (e-mail: cq0309@126.com; jlding@mail.neu.edu.cn; tychai@mail.neu.edu.cn).

S. Yang is with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang, 110819, China, and also with the Centre for Computational Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester, LE1 9BH, U. K. (syang@dmu.ac.uk).

### S-I. MATHEMATICAL FUNCTIONS DESCRIBED IN FIG. 1

In this paper, the true POE of a dynamic constrained multiobjective optimization problem (DCMOP) is given in Fig. 1 of this paper, and the corresponding mathematical functions of this DCMOP are Equation (S-1) in which  $t$  is set to 7.

$$\begin{cases}
 \min F(\mathbf{x}, t) = (f_1(\mathbf{x}, t), f_2(\mathbf{x}, t))^T \\
 \text{obj} \left\{ \begin{aligned}
 f_1(\mathbf{x}, t) &= (1 + \sum_{j=2}^{10} (x_j - \sin(0.05\pi t))^2)(x_1 + 0.05 \sin(4\pi x_1)) \\
 f_2(\mathbf{x}, t) &= (1 + \sum_{j=2}^{10} (x_j - \sin(0.05\pi t))^2)(3 - 0.05t - x_1 + 0.05 \sin(4\pi x_1))
 \end{aligned} \right. \quad (\text{S-1}) \\
 \text{s. t. } \left\{ \begin{aligned}
 f_2(\mathbf{x}, t) + \sin\frac{\pi}{4} \times f_1(\mathbf{x}, t) - (2.36 + 0.01t) &\geq \\
 0.2 \times \left| \sin(4\pi \times (\sin(-\frac{\pi}{16}) \times (f_2(\mathbf{x}, t) - 1) + \cos(-\frac{\pi}{16}) \times f_1(\mathbf{x}, t))) \right|^6 & \\
 0.73 \times f_1(\mathbf{x}, t) + f_2(\mathbf{x}, t) - 6 &< 0
 \end{aligned} \right.
 \end{cases}$$

### S-II. REAL-WORLD APPLICATIONS LINKING TO THE FOUR TYPES OF THE PROPOSED PROBLEMS

The real-world multiobjective optimization problems (MOPs) involved in the existing literature are mainly MOPs with constraints or dynamism. To the best of our knowledge, the real-world optimization problems solved in the existing literature do not consider dynamism, constraints, and multiobjective simultaneously. Indeed, many real-world problems are DCMOPs. Considering this fact, we take the fluid catalytic cracking-distillation (FCC-D) process, a DCMOP we are working on, as an example to illustrate the four types of the true POE and the feasible region mentioned in Section III of this paper.

The FCC-D process is one of the most energy-consuming steps in refineries, which involves complicated physical and chemical reactions. In the FCC-D process, the heavy oil is cracked into the light hydrocarbons in the presence of catalyst, and then the light hydrocarbons are separated into different products (i.e., gasoline, diesel, and natural gas) by the distillation process. To minimize the energy consumption and maximize the economic benefits of products, refineries usually optimize the operating variables that control the stable operation of production units according to production constraints (i.e., yield constraints of different products). During



the optimization of the operating variables, the operator usually changes the yield constraints of products to avoid large production fluctuations and production accidents when the production scenarios are switched. Therefore, optimizing the operating variables of the FCC-D process is a DCMOP, which aims at minimizing energy consumption and maximizing the economic benefits of the products, and takes product yields as dynamic constraints.

For the ease of readers understanding, we will describe the mathematical model of the FCC-D process in Equation (S-2). The definitions of the indices, variables and parameters employed in the model are given as follows:

#### Indices

- $p$  Product index,  $p=1$  (natural gas), 2 (gasoline), and 3 (diesel).  
 $t$  Dynamic environment index.  
 $w$  Pumparound (PA) index .

#### Variables

- $\mathbf{X}(t)$  Operating variable vector at  $t$ .  
 $x_j(t)$  The  $j$ th operating variable at  $t$ .  
 $Y_p(t)$  The  $p$ th product yield at  $t$ .  
 $YP(t)$  Yield of pollutant at  $t$ .  
 $HF(t)$  Mass flow rate of heavy oil at  $t$ .  
 $CT(t)$  Temperature of the heated catalyst at  $t$ .  
 $RT(t)$  Outlet temperature of the riser at  $t$ .  
 $CF(t)$  Mass flow rate of the catalyst circulation rate at  $t$ .  
 $SF(t)$  Flow rate of the stripping steam at  $t$ .  
 $LF(t)$  Flow rate of the lifting steam at  $t$ .  
 $PQ_w(t)$  Calorific value of heat recovery in the  $w$ th PA of the fractionator at  $t$ .  
 $HT(t)$  Temperature of the heavy oil at  $t$ .  
 $Yl_p(t)$  Lower bound of demand for the  $p$ th product at  $t$ .  
 $Yu_p(t)$  Upper bound of demand for the  $p$ th product at  $t$ .

#### Parameters

- $PY_p$  Unit price of the  $p$ th product.  
 $cYS$  Unit price of removing pollutant  
 $cSS$  Unit price of heavy oil  
 $cFF$  Unit price of steam  
 $\rho$  Heat transfer efficiency in the PAs  
 $FT$  Energy required to raise 1 mol of heavy oil by 1 °C  
 $lx_j$  Lower bound of  $x_j(t)$ .  
 $ux_j$  Upper bound of  $x_j(t)$ .

The mathematical model of the FCC-D process can be described as follows [i.e., Equation (S-2)]:

$$\begin{aligned}
 & \left\{ \begin{array}{l} \max f_{\text{product}}(\mathbf{X}(t), t) = \sum_{p=1}^3 (Y_p(t) \times PY_p) - cYS \times YP(t) - cSS \times HF(t) \\ \min f_{\text{energy}}(\mathbf{X}(t), t) = (CT(t) - RT(t)) \times CF(t) \times \rho + (SF(t) + LF(t)) \times cFF \\ \quad - \sum_{w=1}^3 PQ_w(t) + HF(t) \times HT(t) \times FT \end{array} \right. \\
 & \text{s.t.} \left\{ \begin{array}{l} h_k(t) = 0, \quad k=1, \dots, h \\ Yl_p(t) \leq Y_p(t) \leq Yu_p(t) \\ Y_p(t) = g_p(\mathbf{X}(t)) \\ YP(t) = g_{YP}(CF(t), CT(t), SF(t), LF(t), HF(t), RT(t), HT(t)) \\ \mathbf{X}(t) = [CF(t), CT(t), SF(t), LF(t), HF(t), RT(t), HT(t), PQ_1(t), PQ_2(t), PQ_3(t)] \\ lx_j \leq x_j(t) \leq ux_j \end{array} \right. \quad (\text{S-2})
 \end{aligned}$$

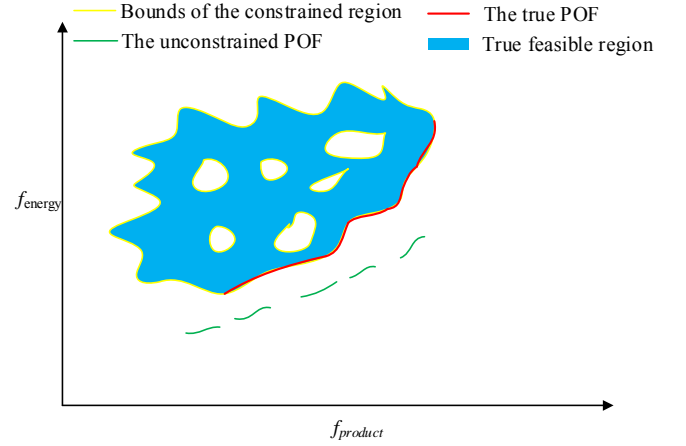


Fig. S-1-a). Diagram of true POF and feasible region of objectives that correspond to Type I at  $t=1$

where  $h$  is the number of equality constraints, and  $h_k(\mathbf{x}, t)$  is the  $k$ th equality constraint.  $g_p(\mathbf{X}(t))$  denotes the complicated nonlinear relationship between the  $p$ th product yield and  $\mathbf{X}(t)$   $g_{YP}(CF(t), CT(t), SF(t), LF(t), HF(t), RT(t), HT(t))$  denotes the nonlinear relationship between the pollutant yield and operating variables (i.e.,  $CF(t), CT(t), SF(t), LF(t), HF(t), RT(t), HT(t)$ ). Note that the above nonlinear relationships can be modeled either by using neural networks (e.g., reference [s1]) or by using [s2] as a rigorous model.

This paper uses  $t=1, 2$ , and 3 to denote different dynamic environments of the DCMOP in the FCC-D process (FCC-D-DCMOP). Subsequently, we provides the diagrams of true POF and feasible regions arising in the four types mentioned in Section III of this paper. Specifically,  $t=1$  denotes the current environment of the FCC-D-DCMOP,  $t=2$  represents the environment in which product yield constraints change due to different production scenarios, and  $t=3$  denotes the environment in which the operator adjusts the product yield constraints based on the optimized operating variables.

Note that the real POF and the feasible region of the FCC-D-DCMOP, in practice, show hundreds of types that vary with dynamic environments, and we only describe four typical types corresponding to the test problems proposed in this paper.

#### 1) Type I.

The feasible region of the FCC-D-DCMOP becomes small if the feasible ranges of product yields are small at  $t=1$  (e.g.,  $Yu_p(t)$  decreases, and  $Yl_p(t)$  increases at  $t=1$ ), which may lead to the infeasibility of the unconstrained POF of the FCC-D-DCMOP (i.e., the unconstrained POF is not in the feasible region) and cause that the true POF of the FCC-D-DCMOP is only determined by the lower bound of the feasible region. In practice, the lower bound of the feasible region may be continuous if the solutions (i.e.,  $\mathbf{X}(t)$ ) corresponding to the lower bound of the feasible region are far from the bound of  $\mathbf{X}(t)$ , causing that the true POF of the FCC-D-DCMOP is continuous at  $t=1$ , as shown in Fig. S-1-a).

The feasible region of the FCC-D-DCMOP becomes large when the feasible ranges of product yields become large at  $t=2$  (e.g.,  $Yu_p(t)$  increases, and  $Yl_p(t)$  decreases at  $t=2$ ), which may cause that the true POF of the FCC-D-DCMOP is determined by both the unconstrained POF and the lower bound of the feasible region. The true POF of the FCC-D-DCMOP may

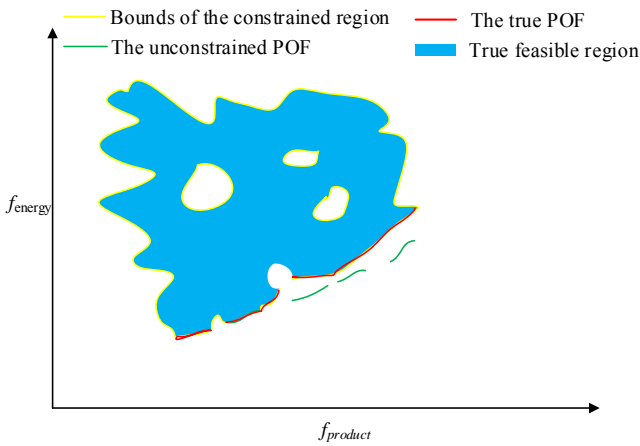


Fig. S-1-b). The diagram of true PO and feasible region of objectives that correspond to Type I at  $t=2$

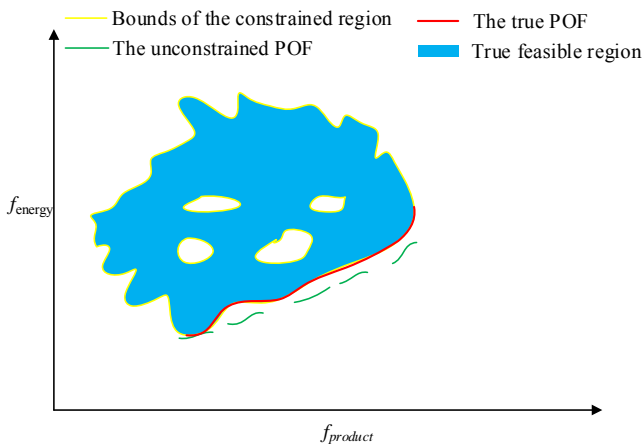


Fig. S-1-c). The diagram of true PO and feasible region of objectives that correspond to Type I at  $t=3$

change from continuous to disconnected if there are infeasible regions on the bound of the feasible region, as shown in Fig. S-1-b).

The operator needs to adjust the product yield constraints (i.e.,  $t=3$ ) if the optimized operating variables differ greatly from the current ones, which can avoid large production fluctuations and production accidents when the operating variables are switched. Narrowing the feasible range of product yields leads to the decrease of the feasible region and the movement of the true PO, which may cause that the true PO of the FCC-D-DCMOP is determined by the unconstrained PO and the lower bound of the feasible region. The true PO may return to continuous from disconnected when a part of the unconstrained PO coincides with the lower bound part of the feasible region, as shown in Fig. S-1-c).

It can be observed from Fig. S-1-a), Fig. S-1-b), and Fig. S-1-c) that the true PO and the feasible region accord with the characteristics discussed in Type I of this paper at  $t=1$ ,  $t=2$ , and  $t=3$ , respectively. (i.e., the feasible region first increases with time  $t$  and then decreases, while the true PO changes from continuous to disconnected and finally back to continuous).

## 2) Type II

The feasible region of the FCC-D-DCMOP becomes large if the feasible ranges of product yields are large at  $t=1$  (e.g.,  $Y_{u_p}(t)$  increases, and  $Y_{l_p}(t)$  decreases at  $t=1$ ), which may cause that the

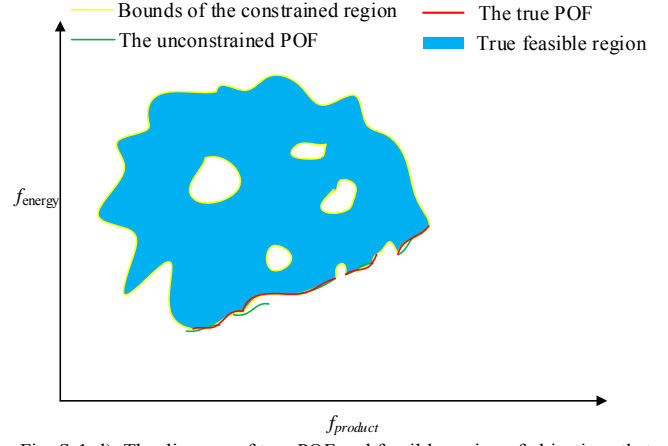


Fig. S-1-d). The diagram of true PO and feasible region of objectives that correspond to Type II at  $t=1$

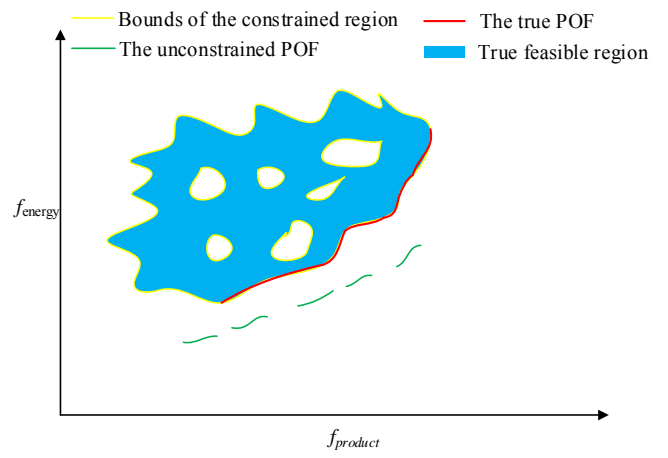


Fig. S-1-e). The diagram of true PO and feasible region of objectives that correspond to Type II at  $t=2$

true PO of the FCC-D-DCMOP is determined by both the unconstrained PO and the lower bound of the feasible region. The lower bound of the feasible region may be disconnected if there are infeasible regions in the lower bound of the feasible region, causing that the true PO of the FCC-D-DCMOP is disconnected at  $t=1$ , as shown in Fig. S-1-d).

The feasible region of the FCC-D-DCMOP becomes small if the feasible ranges of product yields are small at  $t=2$  (e.g.,  $Y_{u_p}(t)$  decreases, and  $Y_{l_p}(t)$  increases at  $t=2$ ), which may lead to the unconstrained PO of the FCC-D-DCMOP to be infeasible (i.e., the unconstrained PO is not in the feasible region) and cause that the true PO of the FCC-D-DCMOP is only determined by the lower bound of the feasible region. In practice, the lower bound of the feasible region may be continuous when the solutions (i.e.,  $X(t)$ ) corresponding to the lower bound of the feasible region are far from the bound of  $X(t)$ , causing the true PO of the FCC-D-DCMOP to change from continuous to disconnected at  $t=2$ , as shown in Fig. S-1-e).

The operator needs to adjust the product yield constraints (i.e.,  $t=3$ ) if the optimized operating variables differ greatly from the current ones, which can avoid large production fluctuations and production accidents when the operating variables are switched. Enlarging the feasible ranges of product yields can lead to the increase of the feasible region of objectives and the movement of the true PO, which may cause that the true PO of the FCC-D-DCMOP is determined by both

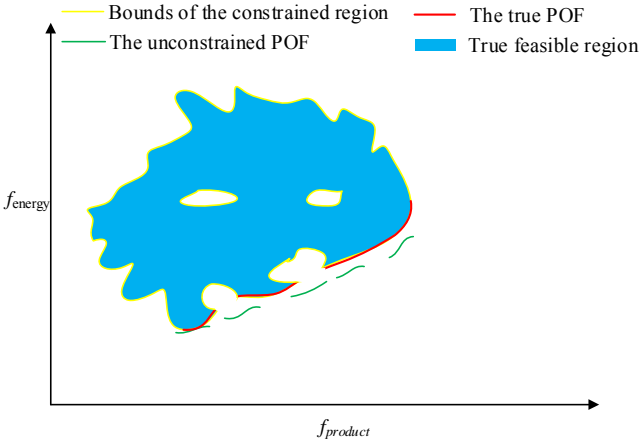


Fig. S-1-f). The diagram of true PO of objectives that correspond to Type II at  $t=3$

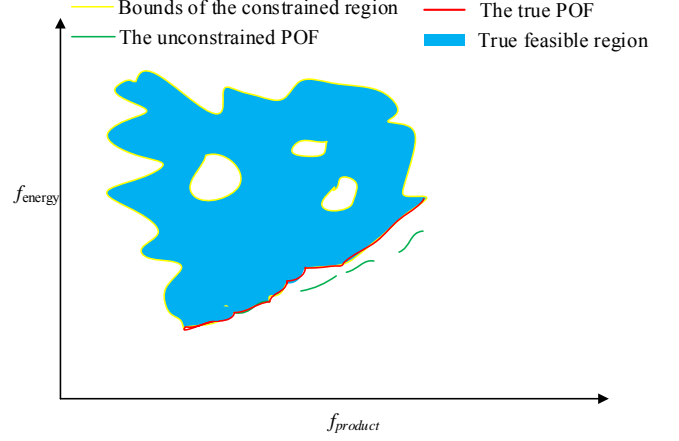


Fig. S-1-h). The diagram of true PO of objectives that correspond to Type III at  $t=2$

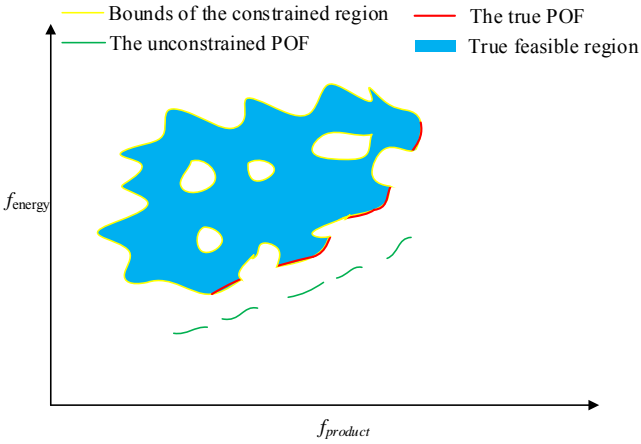


Fig. S-1-g). The diagram of true PO of objectives that correspond to Type III at  $t=1$

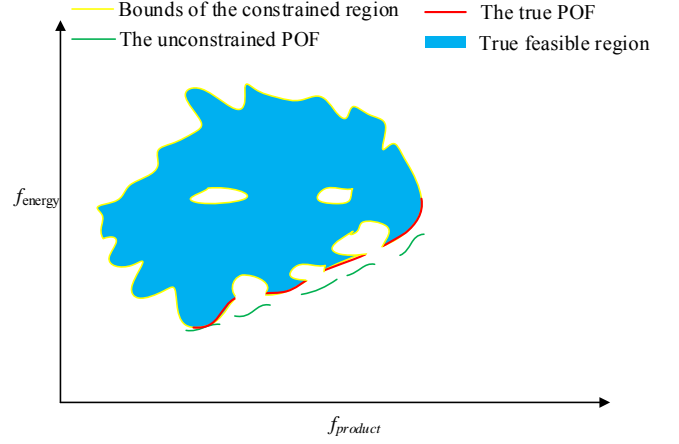


Fig. S-1-i). The diagram of true PO of objectives that correspond to Type III at  $t=3$

the unconstrained PO of and the lower bound of the feasible region. The true PO of may return to disconnected from continuous if there are infeasible regions on the bound of the feasible region, as shown in Fig. S-1-f).

It can be observed from Fig. S-1-d), Fig. S-1-e), and Fig. S-1-f) that the true PO of and the feasible region accord with the characteristics discussed in Type II of this paper at  $t=1$ ,  $t=2$ , and  $t=3$ , respectively. (i.e., the feasible region first decreases with time  $t$  and then increases, while the true PO of changes from disconnected to continuous and finally back to disconnected).

### 3) Type III

The feasible region of the FCC-D-DCMOP becomes small if the feasible ranges of product yields are small at  $t=1$  (e.g.,  $Y_{up}(t)$  decreases, and  $Y_{lp}(t)$  increases at  $t=1$ ), which may lead to the infeasibility of the unconstrained PO of the FCC-D-DCMOP (i.e., the unconstrained PO of is not in the feasible region) and cause that the true PO of the FCC-D-DCMOP is only determined by the lower bound of the feasible region. In practice, the lower bound of the feasible region may be disconnected if the solutions (i.e.,  $X(t)$ ) corresponding to the lower bound of the feasible region are close to the bound of  $X(t)$ , causing the true PO of the FCC-D-DCMOP to be disconnected at  $t=1$ , as shown in Fig. S-1-g).

The feasible region of the FCC-D-DCMOP becomes large when the feasible ranges of product yields are large at  $t=2$  (e.g.,

$Y_{up}(t)$  increases, and  $Y_{lp}(t)$  decreases at  $t=2$ ), which may cause that the true PO of the FCC-D-DCMOP is determined by both the unconstrained PO of and the lower bound of the feasible region. The true PO of may return to continuous from disconnected if a part of the unconstrained PO of coincides with the lower bound part of the feasible region, as shown in Fig. S-1-h).

The operator has to adjust the product yield constraints (i.e.,  $t=3$ ) if the optimized operating variables differ greatly from the current ones, which can avoid large production fluctuations and production accidents when the operating variables are switched. Narrowing the feasible range of product yields leads to the decrease of the feasible region and the movement of the true PO of, which may cause that the true PO of the FCC-D-DCMOP is determined by both the unconstrained PO of and the lower bound of the feasible region. The true PO of may return to disconnected from continuous if some infeasible regions move to the bound of the feasible region, as shown in Fig. S-1-i).

It can be observed from Fig. S-1-g), Fig. S-1-h), and Fig. S-1-i) that the true PO of and the feasible region accord with the characteristics discussed in Type III of this paper at  $t=1$ ,  $t=2$ , and  $t=3$ , respectively. (i.e., the feasible region first increases with time  $t$  and then decreases, while the true PO of changes from disconnected to continuous and finally back to disconnected).

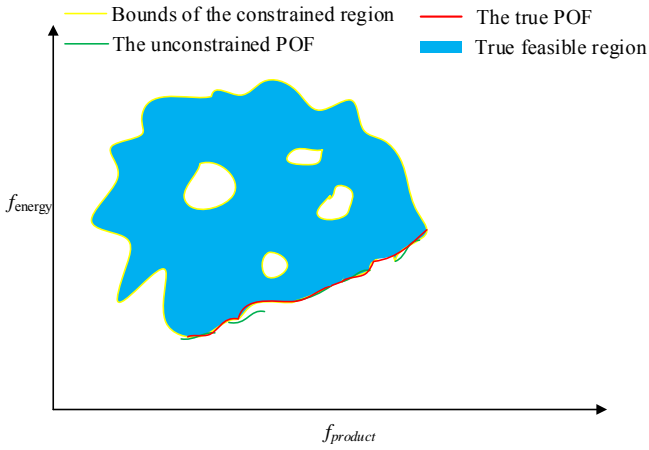


Fig. S-1-j). The diagram of true PO and feasible region of objectives that correspond to Type IV at  $t=1$

#### 4) Type IV.

The feasible region of the FCC-D-DCMOP becomes small if the feasible ranges of product yields are small at  $t=1$  (e.g.,  $Y_{up}(t)$  decreases, and  $Y_{lp}(t)$  increases at  $t=1$ ), which may lead to the unconstrained PO of the FCC-D-DCMOP to be infeasible (i.e., the unconstrained PO is not in the feasible region) and cause that the true PO of the FCC-D-DCMOP is only determined by the lower bound of the feasible region. In practice, the lower bound of the feasible region may be continuous if the solutions (i.e.,  $X(t)$ ) corresponding to the lower bound of the feasible region are far from the bound of  $X(t)$ . Therefore, the true PO of the FCC-D-DCMOP is continuous at  $t=1$  if a part of the unconstrained PO coincides with the lower bound part of the feasible region, as shown in Fig. S-1-j).

The feasible region of the FCC-D-DCMOP becomes small if the feasible ranges of product yields are small at  $t=2$  (e.g.,  $Y_{up}(t)$  decreases, and  $Y_{lp}(t)$  increases at  $t=2$ ), which may lead to the infeasibility of the unconstrained PO of the FCC-D-DCMOP (i.e., the unconstrained PO is not in the feasible region) and cause that the true PO of the FCC-D-DCMOP is only determined by the lower bound of the feasible region. In practice, the lower bound of the feasible region may be disconnected if the solutions (i.e.,  $X(t)$ ) corresponding to the lower bound of the feasible region are close to the bound of  $X(t)$ , causing the true PO of the FCC-D-DCMOP to be disconnected at  $t=1$ , as shown in Fig. S-1-k).

The operator has to adjust the product yield constraints (i.e.,  $t=3$ ) if the optimized operating variables differ greatly from the current ones, which can avoid large production fluctuations and production accidents when the operating variables are switched. Enlarging the feasible range of product yields leads to the increase of the feasible region of objectives and the movement of the true PO, which may cause that the true PO of the FCC-D-DCMOP is determined by the unconstrained PO and the lower bound of the feasible region. The true PO may return to continuous from disconnected when a part of the unconstrained PO coincides with the lower bound part of the feasible region, as shown in Fig. S-1-l). It can be observed from Fig. S-1-j), Fig. S-1-k), and Fig. S-1-l) that the true PO and the feasible region accord with the characteristics discussed in Type IV of this paper at  $t=1$ ,  $t=2$ , and  $t=3$ , respectively. (i.e., the

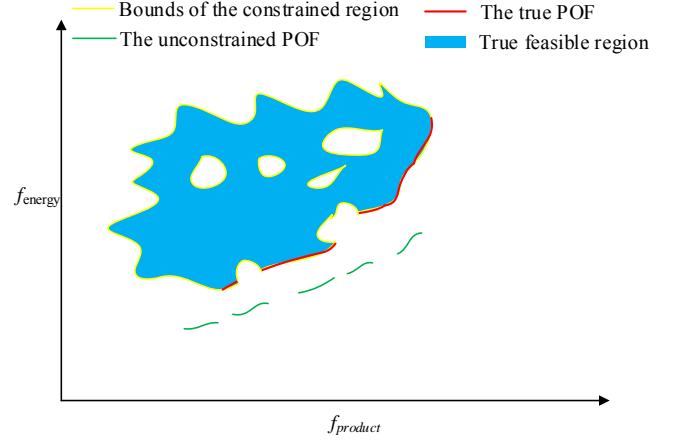


Fig. S-1-k). The diagram of true PO and feasible region of objectives that correspond to Type IV at  $t=2$

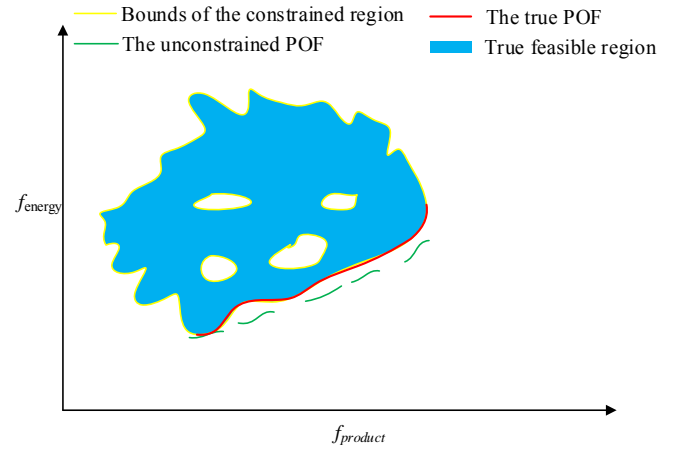


Fig. S-1-l). The diagram of true PO and feasible region of objectives that correspond to Type IV at  $t=3$

feasible region first decreases with time  $t$  and then increases, while the true PO changes from continuous to disconnected and finally back to continuous).

**Remark:** Indeed, the true PO and the feasible region of the FCC-D-DCMOP show hundreds of types that vary with dynamic environments, and we only describe four typical types corresponding to the test problems proposed in this paper. In the future, other types of the true PO and the feasible region arising in real-world applications will be gradually explored.

### S-III. IDEAS ON SCALING UP THE PROPOSED TEST PROBLEMS

The test problems proposed in Section III of this paper can be extended from the perspectives of the decision space and the objective space, as follows:

#### 1) Decision Space

$h(\mathbf{x}_I, t)$  and  $g(\mathbf{x}_{II}, t)$  in the proposed test problems determine the decision space, where  $\mathbf{x}_I \cup \mathbf{x}_{II} = \mathbf{x}$  and  $\mathbf{x}_I \cap \mathbf{x}_{II} = \emptyset$ . Indeed, the decision variables used in the proposed test problems can be extended in any dimension as long as the following conditions are satisfied:

- The minimum value of  $g(\mathbf{x}_{II}, t)$  is zero.
- $h(\mathbf{x}_I, t)$  is set to  $x_1$ , and it satisfies  $0 \leq h(\mathbf{x}_I, t) \leq s(t)$ .

Note that for the test problems designed in this paper, the real PO does not depend on the dimension of the decision

variables. Therefore, the real POF and the feasible region of the test problems can still change according to the four types mentioned in this paper as long as the settings of  $s(t)$ ,  $m(t)$ , and  $z(t)$  are reasonable. We recommend that their settings (i.e.,  $s(t)$ ,  $m(t)$ , and  $z(t)$ ) are consistent with those used in our paper when the dimensions of the test problem are extended.

## 2) Objective Function Space

In the proposed problems, the constraints cause the bound of the feasible region of the objectives to be irregular, leading to the true POF exhibiting different characteristics (i.e., disconnected and continuous) with dynamism. Therefore, we should first consider how to scale up the feasible regions of the objective functions.

In the proposed test problems, the irregularity of the bounds of the objective functions is caused by  $a \times |\sin(b\pi \times (\sin(\theta) \times (f_2(\mathbf{x}, t) - e) + \cos(\theta) \times f_1(\mathbf{x}, t))^c)|^d$ . Therefore,  $a \times |\sin(b\pi \times (\sin(\theta) \times (f_2(\mathbf{x}, t) - e) + \cos(\theta) \times f_1(\mathbf{x}, t))^c)|^d$  needs to involve each objective function once the number of objective functions proposed in this paper is extended to  $M$ . Additionally,  $\cos(\theta) \times (f_2(\mathbf{x}, t) - e) - \sin(\theta) \times f_1(\mathbf{x}, t) - m(t)$  controls the feasible regions of each objective function, so  $\cos(\theta) \times (f_2(\mathbf{x}, t) - e) - \sin(\theta) \times f_1(\mathbf{x}, t) - m(t)$  also needs to involve each objective function. This paper provides an example of scaling up the constraints used in the proposed test problems, which is as follows:

$$s.t. \begin{cases} \cos(\theta) \times ((f_2(\mathbf{x}, t) - e) + \dots + (f_M(\mathbf{x}, t) - e)) - \sin(\theta) \times f_1(\mathbf{x}, t) - m(t) \geq \\ a \times |\sin(b\pi \times (\sin(\theta) \times ((f_2(\mathbf{x}, t) - e) + \dots + (f_M(\mathbf{x}, t) - e)) + \cos(\theta) \times f_1(\mathbf{x}, t))^c)|^d \\ sl \times f_1(\mathbf{x}, t) + (f_2(\mathbf{x}, t) + \dots + f_M(\mathbf{x}, t)) - z(t) < 0 \\ z(t) > m(t) \end{cases} \quad (S-3)$$

where  $M > 2$ .

It can be ascertained from Section III of this paper that the true POF of DCMOPs may be determined by the bound of constraints. Therefore, the constraints must be considered when scaling up the objective functions. The objective functions that satisfy the following conditions can be used as the objectives functions of the extended constraints (i.e., Equation (S-3)).

- During the dynamic environments change, all objective values in the unconstrained POF must be infeasible in at least one environment.
- The true POF of the extended DCMOP is determined by the unconstrained POF in at least one environment.

This paper provides an example of the extended objective functions (i.e., Equation (S-4)) for Equation (S-3), which is as follows:

$$obj \begin{cases} \min F(\mathbf{x}, t) = (f_1(\mathbf{x}, t), \dots, f_k(\mathbf{x}, t), \dots, f_M(\mathbf{x}, t))^T \\ f_1(\mathbf{x}, t) = (s(t) + g(\mathbf{x}_{II}, t)) \prod_i^{M-1} A_i \sin(W_i \pi x_i) \\ f_k(\mathbf{x}, t) = (s(t) + g(\mathbf{x}_{II}, t)) \prod_i^{M-k} A_i \sin(W_i \pi x_i) \cos(W_i \pi x_{M-k+1}) \\ f_M(\mathbf{x}, t) = (s(t) + g(\mathbf{x}_{II}, t)) \cos(W_i \pi x_i) \end{cases} \quad (S-4)$$

## S-IV. TRUE POF AND POS OF EACH TEST PROBLEM

The true POF and POS of each test problem are given in Table S-I.

## S-V. ALGORITHM PARAMETER SELECTION

In this paper, we adopt the popular simulated binary crossover (SBX) and polynomial mutation (PM) operators proposed in [54] to generate offspring. The main parameters in SBX and PM are the crossover probability ( $Cr$ ), the distribution index for SBX ( $dc$ ), the mutation probability ( $mp$ ), and the distribution index for mutation ( $dm$ ). Additionally, the other parameters in  $dCMOEA$  are the population size ( $N$ ) and the threshold of the number of feasible solutions ( $NF$ ).

According to our preliminary experiments, a larger  $N$  leads to better performance of  $dCMOEA$  and a longer computation time. To eliminate the disturbance of  $N$  on the algorithms' performance comparison,  $N$  of all algorithms was set to 200 (i.e.,  $N=200$ ).

In Section IV-D, we proposed a population selection operator that updates the population (i.e.,  $P$ ) according to population and offspring (i.e.,  $Q$ ). The number of feasible solutions in  $P$  and  $Q$  of each generation usually is not the same (i.e., it may be zero, 100, 200, or 400). To determine  $NF$ , we set  $NF$  to 0, 100, 200, 300, and 400 to analyze the sensitivity on  $NF$  of  $dCMOEA$ . The results of sensitivity analysis on  $NF$  are given in Table S-II based on the eight test problems.

It can be seen from Table S-II that  $dCMOEA$  with  $NF=100$  can perform better than its variants on most of the test problems in terms of three metrics, implying that  $NF=100$  is suitable for  $dCMOEA$ .

The parameters  $Cr$  and  $dc$  used in SBX are coupled to affect the performance of  $dCMOEA$ , and a similar situation for the other two parameters used in PM also arise. Therefore,  $Cr$  and  $dc$  are determined first, followed by  $mp$  and  $dm$ . In this paper, we use three performance metrics (i.e.,  $HV$ ,  $IGD$ , and  $SP$ ) to evaluate the performance of an algorithm. For  $HV$ , the larger the value, the better the performance of an algorithm. For  $IGD$  and  $SP$ , however, the lower the value, the better the performance of an algorithm. To facilitate the analysis of the four parameters on the performance of  $dCMOEA$ , we use the average regularization indicator (ARI) to represent these three metrics, which is calculated as Equations (S-5) and (S-6).

$$RI_{bf} = \frac{HV_{bf}}{HV_{\min}} + \left( \frac{-IGD_{bf}}{-IGD_{\max}} \right) + \left( \frac{-SP_{bf}}{-SP_{\max}} \right) \quad (S-5)$$

$$ARI_b = \frac{1}{8} \sum_{f=1}^8 RI_{bf} \quad (S-6)$$

where  $RI_{bf}$  is the regularization value of the performance metrics obtained by  $dCMOEA$  that uses the  $b$ th parameter combination to solve the  $f$ th test problem.  $HV_{bf}$ ,  $IGD_{bf}$ , and  $SP_{bf}$  are, respectively, the  $HV$ ,  $IGD$ , and  $SP$  metrics obtained by  $dCMOEA$  based on the  $b$ th parameter combination to solve the  $f$ th test problem.  $ARI_b$  denotes the sum of the average regularization indicators of  $dCMOEA$  that uses the  $b$ th parameter combination to solve eight test problems.

Table S-III provides the five levels of these four parameters based on our preliminary experiments. For example,  $Cr$ ,  $dc$ ,  $mp$ , and  $dm$  are, respectively, equal to 0.8, 7, 0.15, and 40 when their levels are 4. For brevity, an orthogonal array  $L_{25}$  [i.e., these two parameters have 25 (i.e.,  $5^2$ ) parameter combinations under five levels] was used to examine the first two parameters (i.e.,  $Cr$  and  $dc$ ). For each parameter combination (e.g.,  $Cr=0.6$  and  $dc=3$ ),  $dCMOEA$  was independently executed 30 times on

TABLE S-I  
THE TRUE POF AND POS OF EACH TEST PROBLEM

Instance No.	Description	Remarks
1	$s(t)=\max(3.5-0.14\times t, 0.7+0.14\times t); m(t)=\max(1.43-0.05\times t, 0.43+0.05\times t)$ $W_i=2; g(\mathbf{x}_{11}, t) = \sum_{j=2}^d(x_j - (1 - 0.9 \times \sin(0.2\pi\times t)))^2$ $f_1(\mathbf{x}, t)=(1+g(\mathbf{x}_{11}, t))(x_1+0.05\times\sin(W_i\pi x_1)); f_2(\mathbf{x}, t)=(1+g(\mathbf{x}_{11}, t))(s(t)-x_1+0.05\times\sin(W_i\pi x_1))$ POS( $t$ ): $0\leq x_1\leq s(t), x_j=1-0.9\times\sin(0.2\pi\times t), j=2, \dots, d;$ POF( $t$ ): $f_1 + f_2 = s(t) + 2A_t \sin(W_t \pi \frac{f_1 - f_2 + s(t)}{2})$ $s.j. \cos\frac{\pi}{4} \times f_2 - \sin\frac{\pi}{4} \times f_1 - m(t) \geq a \times \left  \sin(\pi \times (\sin\frac{\pi}{16} \times (f_2 - 1) + \cos\frac{\pi}{16} \times f_1)) \right ^{0.5};$ $f_1 + f_2 - 6 < 0; m(t) < 6$	The unconstrained POF has one mode. The feasible region first increases with time $t$ and then decreases, while the true POF changes from continuous to disconnected and finally back to continuous.
2	$s(t)=\max(2.5-0.05\times t, 1.5+0.05\times t); m(t)=\max(1.16-0.075\times t, -0.34+0.075\times t)$ $W_i=2; g(\mathbf{x}_{11}, t) = \sum_{j=2}^d(x_j - (1 - 0.9 \times \sin(0.2\pi\times t)))^2$ $f_1(\mathbf{x}, t)=(1+g(\mathbf{x}_{11}, t))(x_1+0.05\times\sin(W_i\pi x_1)); f_2(\mathbf{x}, t)=(1+g(\mathbf{x}_{11}, t))(s(t)-x_1+0.05\times\sin(W_i\pi x_1))$ POS( $t$ ): $0\leq x_1\leq s(t), x_j=1-0.9\times\sin(0.2\pi\times t), j=2, \dots, d;$ POF( $t$ ): $f_1 + f_2 = s(t) + 2A_t \sin(W_t \pi \frac{f_1 - f_2 + s(t)}{2})$ $s.j. \cos\frac{\pi}{4} \times f_2 - \sin\frac{\pi}{4} \times f_1 - m(t) \geq a \times \left  \sin(\pi \times (\sin\frac{\pi}{16} \times (f_2 - 1) + \cos\frac{\pi}{16} \times f_1)) \right ^{0.5};$ $f_1 + f_2 - 6 < 0; m(t) < 6$	The unconstrained POF has one mode. The feasible region first decreases with time $t$ and then increases, while the true POF changes from disconnected to continuous and finally back to disconnected.
3	$s(t)=\max(2.1-0.14\times t, 4.9+0.14\times t); m(t)=\max(0.93+0.05\times t, 1.93-0.05\times t)$ $W_i=2; g(\mathbf{x}_{11}, t) = \sum_{j=2}^d(x_j - (1 - 0.9 \times \sin(0.2\pi\times t)))^2$ $f_1(\mathbf{x}, t)=(1+g(\mathbf{x}_{11}, t))(x_1+0.05\times\sin(W_i\pi x_1)); f_2(\mathbf{x}, t)=(1+g(\mathbf{x}_{11}, t))(s(t)-x_1+0.05\times\sin(W_i\pi x_1))$ POS( $t$ ): $0\leq x_1\leq s(t), x_j=1-0.9\times\sin(0.2\pi\times t), j=2, \dots, d;$ POF( $t$ ): $f_1 + f_2 = s(t) + 2A_t \sin(W_t \pi \frac{f_1 - f_2 + s(t)}{2})$ $s.j. \cos\frac{\pi}{4} \times f_2 - \sin\frac{\pi}{4} \times f_1 - m(t) \geq a \times \left  \sin(\pi \times (\sin\frac{\pi}{16} \times (f_2 - 1) + \cos\frac{\pi}{16} \times f_1)) \right ^{0.5};$ $f_1 + f_2 - 6 < 0; m(t) < 6$	The unconstrained POF has one mode. The feasible region first increases with time $t$ and then decreases, while the true POF changes from disconnected to continuous and finally back to disconnected.
4	$s(t)=\max(2+0.05\times t, 3-0.05\times t); m(t)=\max(0.41+0.075\times t, 1.91-0.075\times t)$ $W_i=2; g(\mathbf{x}_{11}, t) = \sum_{j=2}^d(x_j - (1 - 0.9 \times \sin(0.2\pi\times t)))^2$ $f_1(\mathbf{x}, t)=(1+g(\mathbf{x}_{11}, t))(x_1+0.05\times\sin(W_i\pi x_1)); f_2(\mathbf{x}, t)=(1+g(\mathbf{x}_{11}, t))(s(t)-x_1+0.05\times\sin(W_i\pi x_1))$ POS( $t$ ): $0\leq x_1\leq s(t), x_j=1-0.9\times\sin(0.2\pi\times t), j=2, \dots, d;$ POF( $t$ ): $f_1 + f_2 = s(t) + 2A_t \sin(W_t \pi \frac{f_1 - f_2 + s(t)}{2})$ $s.j. \cos\frac{\pi}{4} \times f_2 - \sin\frac{\pi}{4} \times f_1 - m(t) \geq a \times \left  \sin(\pi \times (\sin\frac{\pi}{16} \times (f_2 - 1) + \cos\frac{\pi}{16} \times f_1)) \right ^{0.5};$ $f_1 + f_2 - 6 < 0; m(t) < 6$	The unconstrained POF has one mode. The feasible region first decreases with time $t$ and then increases, while the true POF changes from continuous to disconnected and finally back to continuous.
5	$s(t)=\max(3.5-0.14\times t, 0.7+0.14\times t); m(t)=\max(1.43-0.05\times t, 0.43+0.05\times t)$ $W_i=6\times\sin(0.2\times\pi(t+1)); g(\mathbf{x}_{11}, t) = \sum_{j=2}^d(x_j - (1 - 0.9 \times \sin(0.2\pi\times t)))^2$ $f_1(\mathbf{x}, t)=(1+g(\mathbf{x}_{11}, t))(x_1+0.05\times\sin(W_i\pi x_1)); f_2(\mathbf{x}, t)=(1+g(\mathbf{x}_{11}, t))(s(t)-x_1+0.05\times\sin(W_i\pi x_1))$ POS( $t$ ): $0\leq x_1\leq s(t), x_j=1-0.9\times\sin(0.2\pi\times t), j=2, \dots, d;$ POF( $t$ ): $f_1 + f_2 = s(t) + 2A_t \sin(W_t \pi \frac{f_1 - f_2 + s(t)}{2})$ $s.j. \cos\frac{\pi}{4} \times f_2 - \sin\frac{\pi}{4} \times f_1 - m(t) \geq a \times \left  \sin(\pi \times (\sin\frac{\pi}{16} \times (f_2 - 1) + \cos\frac{\pi}{16} \times f_1)) \right ^{0.5};$ $f_1 + f_2 - 6 < 0; m(t) < 6$	The unconstrained POF oscillates among ten optimization modes. The feasible region first increases with time $t$ and then decreases, while the true POF changes from continuous to disconnected and finally back to continuous.
6	$s(t)=\max(2.5-0.05\times t, 1.5+0.05\times t); m(t)=\max(1.16-0.075\times t, -0.34+0.075\times t)$ $W_i=6\times\sin(0.2\times\pi(t+1)); g(\mathbf{x}_{11}, t) = \sum_{j=2}^d(x_j - (1 - 0.9 \times \sin(0.2\pi\times t)))^2$ $f_1(\mathbf{x}, t)=(1+g(\mathbf{x}_{11}, t))(x_1+0.05\times\sin(W_i\pi x_1)); f_2(\mathbf{x}, t)=(1+g(\mathbf{x}_{11}, t))(s(t)-x_1+0.05\times\sin(W_i\pi x_1))$ POS( $t$ ): $0\leq x_1\leq s(t), x_j=1-0.9\times\sin(0.2\pi\times t), j=2, \dots, d;$ POF( $t$ ): $f_1 + f_2 = s(t) + 2A_t \sin(W_t \pi \frac{f_1 - f_2 + s(t)}{2})$ $s.j. \cos\frac{\pi}{4} \times f_2 - \sin\frac{\pi}{4} \times f_1 - m(t) \geq a \times \left  \sin(\pi \times (\sin\frac{\pi}{16} \times (f_2 - 1) + \cos\frac{\pi}{16} \times f_1)) \right ^{0.5};$ $f_1 + f_2 - 6 < 0; m(t) < 6$	The unconstrained POF oscillates among ten optimization modes. The feasible region first decreases with time $t$ and then increases, while the true POF changes from disconnected to continuous and finally back to disconnected.
7	$s(t)=\max(2.1-0.14\times t, 4.9+0.14\times t); m(t)=\max(0.93+0.05\times t, 1.93-0.05\times t)$ $W_i=6\times\sin(0.2\times\pi(t+1)); g(\mathbf{x}_{11}, t) = \sum_{j=2}^d(x_j - (1 - 0.9 \times \sin(0.2\pi\times t)))^2$ $f_1(\mathbf{x}, t)=(1+g(\mathbf{x}_{11}, t))(x_1+0.05\times\sin(W_i\pi x_1)); f_2(\mathbf{x}, t)=(1+g(\mathbf{x}_{11}, t))(s(t)-x_1+0.05\times\sin(W_i\pi x_1))$ POS( $t$ ): $0\leq x_1\leq s(t), x_j=1-0.9\times\sin(0.2\pi\times t), j=2, \dots, d;$ POF( $t$ ): $f_1 + f_2 = s(t) + 2A_t \sin(W_t \pi \frac{f_1 - f_2 + s(t)}{2})$ $s.j. \cos\frac{\pi}{4} \times f_2 - \sin\frac{\pi}{4} \times f_1 - m(t) \geq a \times \left  \sin(\pi \times (\sin\frac{\pi}{16} \times (f_2 - 1) + \cos\frac{\pi}{16} \times f_1)) \right ^{0.5};$ $f_1 + f_2 - 6 < 0; m(t) < 6$	The unconstrained POF oscillates among ten optimization modes. The feasible region first increases with time $t$ and then decreases, while the true POF changes from disconnected to continuous and finally back to disconnected.
8	$s(t)=\max(2+0.05\times t, 3-0.05\times t); m(t)=\max(0.41+0.075\times t, 1.91-0.075\times t)$ $W_i=6\times\sin(0.2\times\pi(t+1)); g(\mathbf{x}_{11}, t) = \sum_{j=2}^d(x_j - (1 - 0.9 \times \sin(0.2\pi\times t)))^2$ $f_1(\mathbf{x}, t)=(1+g(\mathbf{x}_{11}, t))(x_1+0.05\times\sin(W_i\pi x_1)); f_2(\mathbf{x}, t)=(1+g(\mathbf{x}_{11}, t))(s(t)-x_1+0.05\times\sin(W_i\pi x_1))$ POS( $t$ ): $0\leq x_1\leq s(t), x_j=1-0.9\times\sin(0.2\pi\times t), j=2, \dots, d;$ POF( $t$ ): $f_1 + f_2 = s(t) + 2A_t \sin(W_t \pi \frac{f_1 - f_2 + s(t)}{2})$ $s.j. \cos\frac{\pi}{4} \times f_2 - \sin\frac{\pi}{4} \times f_1 - m(t) \geq a \times \left  \sin(\pi \times (\sin\frac{\pi}{16} \times (f_2 - 1) + \cos\frac{\pi}{16} \times f_1)) \right ^{0.5};$ $f_1 + f_2 - 6 < 0; m(t) < 6$	The unconstrained POF oscillates among ten optimization modes. The feasible region first decreases with time $t$ and then increases, while the true POF changes from continuous to disconnected and finally back to continuous.

TABLE S-II  
SENSITIVITY ANALYSIS ON DIFFERENT THRESHOLDS OF NUMBER OF THE NUMBER OF FEASIBLE SOLUTIONS

Instance No.	metrics	0	100	200	300	400
1	HV	3.49065	<b>3.49285</b>	3.48939	3.48679	3.48833
	IGD	0.04087	<b>0.04024</b>	0.04127	0.04192	0.0415
	SP	0.01360	0.01381	0.01330	<b>0.01326</b>	0.01391
2	HV	3.46302	<b>3.46634</b>	3.46176	3.46187	3.46345
	IGD	0.04264	<b>0.04149</b>	0.04279	0.04311	0.04242
	SP	<b>0.01320</b>	0.01412	0.01351	0.01364	0.01353
3	HV	3.48398	<b>3.48799</b>	3.48526	3.48581	3.48432
	IGD	0.04307	<b>0.04207</b>	0.04264	0.04257	0.04296
	SP	0.01531	0.01582	0.01507	0.01523	<b>0.01500</b>
4	HV	3.45399	<b>3.45823</b>	3.45304	3.45481	3.45249
	IGD	0.04097	<b>0.03976</b>	0.0413	0.04054	0.04171
	SP	0.01232	0.01297	<b>0.01224</b>	0.01247	0.0125
5	HV	3.50560	<b>3.51164</b>	3.50515	3.50667	3.49893
	IGD	0.05456	<b>0.05310</b>	0.05415	0.05413	0.05595
	SP	0.01460	<b>0.01432</b>	0.01482	0.01446	0.01445
6	HV	3.46023	<b>3.46181</b>	3.45962	3.4588	3.46096
	IGD	0.05377	<b>0.05372</b>	0.05417	0.0539	0.05385
	SP	0.01391	0.01416	0.01360	0.0138	<b>0.01379</b>
7	HV	3.49832	<b>3.50450</b>	3.49677	3.49849	3.49712
	IGD	0.05590	<b>0.05456</b>	0.05619	0.05612	0.05602
	SP	<b>0.01608</b>	0.01663	0.01594	0.01625	0.01621
8	HV	3.45648	<b>3.45812</b>	3.45276	3.45662	3.45516
	IGD	0.05201	0.05185	0.05271	<b>0.05182</b>	0.05214
	SP	0.01306	0.01294	0.01306	<b>0.01247</b>	0.01305

TABLE S-III

THE COMPUTATIONAL TIME OF EACH ALGORITHM ON EACH TEST PROBLEM WITH  $\tau_i=10, 15, \text{ AND } 20$

Parameters	Level 1	Level 2	Level 3	Level 4	Level 5
$Cr$	0.5	0.6	0.7	0.8	0.9
$dc$	1	3	5	7	9
$mp$	0.01	0.05	0.1	0.15	0.2
$dm$	10	20	30	40	50

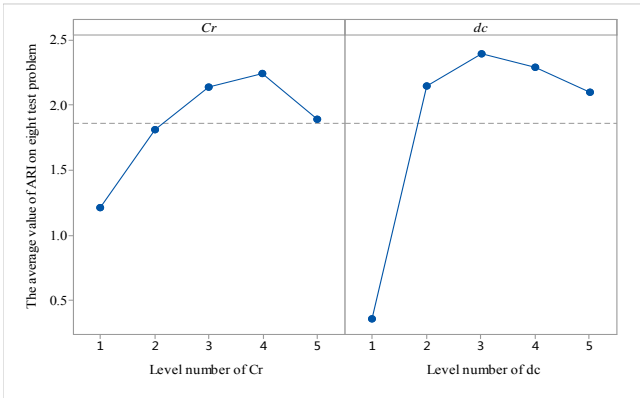


Fig. S-2. Factor-level trends of  $Cr$  and  $dc$

each test problem to obtain the corresponding  $HV$ ,  $IGD$ , and  $SP$ . Subsequently, we use Equations (S-3) and (S-4) to calculate the  $ARI_b$  of  $dCMOEA$  that uses the  $b$ th combination of  $Cr$  and  $dc$ . Then, the Taguchi method proposed in [s3] to examine the impacts of parameters on the performance of  $dCMOEA$  and the factor-level trends of  $ARI_b$  of  $Cr$  and  $dc$  can be obtained, which are given in Fig. S-2.

It can be seen from Table S-III and Fig. S-2 that  $Cr$  with a level of 4 and  $dc$  with a level of 3 yield the best result. A small value of  $Cr$  results in a good exploitation capability but poor exploration capability. Similarly, a large value of  $Cr$  results in a good exploration capability but poor exploitation capability. Thus,  $Cr$  was set to 0.8 to balance the exploitation and exploration capabilities of  $dCMOEA$ . For a similar reason,  $dc$  was set to 5.

The method using to determine  $mp$  and  $dm$  is used to

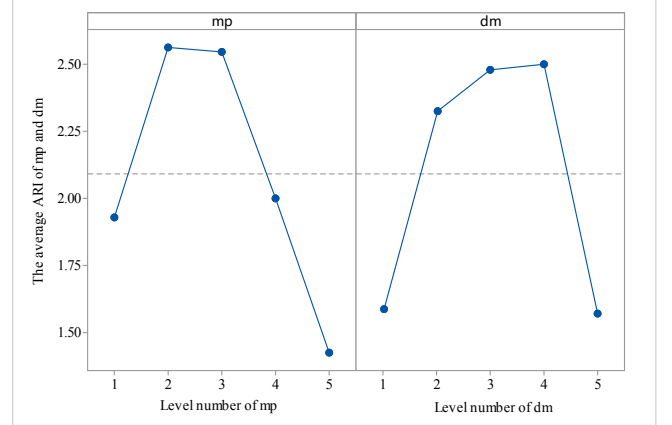


Fig. S-3. Factor-level trends of  $mp$  and  $dm$

determine  $Cr$  and  $dc$ . The factor-level trends of  $mp$  and  $dm$  are given in Fig. S-2. It can be seen from Table S-III and Fig. S-3 that  $mp$  with a level of 4 and  $dm$  with a level of 3 yield the best result. In other words,  $mp$  and  $dm$  were set to 0.05 and 40, respectively.

#### S-VI. PERFORMANCE RANKING OF EACH ALGORITHM

The final rank of each algorithm under performance metrics for each test problem is indicated in Table S-IV.

#### S-VII. COMPUTATIONAL TIME OF EACH ALGORITHM ON EACH TEST PROBLEM

In Section V of this paper, we set the change frequency ( $\tau_i$ ) as 10, 15, and 20 (i.e.,  $\tau_i=10, 15, \text{ and } 20$ ), and  $\tau_i=10$  means that an algorithm iterates 10 generations in a dynamic environment. To further examine the performance of each algorithm, we compare the computational time of each algorithm in an environment. For a test problem with  $\tau_i=10$  and  $n_i=21$ , for example, the computational time means the time required for 10 iterations of an algorithm. We use Equation (S-7) to calculate the average computational time required for the  $p$ th algorithm to run for  $\tau_i$  generations on the test problems.

$$CT_{p\tau_i} = \frac{1}{r} \frac{1}{n_i} \sum_{i=1}^r \sum_{j=1}^{n_i} T_{p\tau_i,ij} \quad (S-7)$$

where  $T_{p\tau_i,ij}$  denotes the time that is required to iterate  $\tau_i$  times under the  $j$ th dynamic environment in the  $i$ th run of the  $p$ th algorithm.  $CT_{p\tau_i}$  denotes the average computational time required for the  $p$ th algorithm to run for  $\tau_i$  generations.  $r$  is the number of runs of an algorithm on a test problem,

The average computational time of each algorithm on each test problem with  $\tau_i=10, 15, \text{ and } 20$  is given in Table S-V. It can be seen from Table S-V that the computational time of  $dCMOEA$  is longer than that of DC-MOEA, DC-NSGA-II-A, DC-NSGA-II, and DC-NSGA-III on the most of the test problems.

To eliminate the influence of running time on the algorithm's performance, we used an elapsed CPU time limit of CP=2000ms as the termination criterion of each algorithm in each dynamic environment. The average values and standard deviations of the three performance metrics (i.e.,  $HV$ ,  $IGD$ , and  $SP$ ) for DC-MOEA, DC-NSGA-II-A, DC-NSGA-II,

TABLE S-IV  
PERFORMANCE RANKING ON THREE METRICS FOR TEST PROBLEMS.

Instance No.	Rank	Ranking by HV	Ranking by IGD	Ranking by SP
1	1th	dCMOEA	dCMOEA	dCMOEA
	2nd	DC-NSGA-II	DC-NSGA-II	DC-NSGA-II
	3rd	DC-NSGA-III	DC-NSGA-III	DC-TAEA
	4th	DC-TAEA	DC-TAEA	DC-MOEA
	5th	DC-MOEA	DC-MOEA	DC-NSGA-III
	6th	DC-NSGA-II-A	DC-NSGA-II-A	DC-NSGA-II-A
2	1th	dCMOEA	dCMOEA	dCMOEA
	2nd	DC-NSGA-II	DC-NSGA-II	DC-NSGA-II
	3rd	DC-TAEA	DC-NSGA-III	DC-TAEA
	4th	DC-NSGA-III	DC-TAEA	DC-MOEA
	5th	DC-MOEA	DC-MOEA	DC-NSGA-III
	6th	DC-NSGA-II-A	DC-NSGA-II-A	DC-NSGA-II-A
3	1th	dCMOEA	dCMOEA	dCMOEA
	2nd	DC-NSGA-II	DC-NSGA-II	DC-NSGA-II
	3rd	DC-NSGA-III	DC-NSGA-III	DC-TAEA
	4th	DC-TAEA	DC-TAEA	DC-MOEA
	5th	DC-MOEA	DC-MOEA	DC-NSGA-III
	6th	DC-NSGA-II-A	DC-NSGA-II-A	DC-NSGA-II-A
4	1th	dCMOEA	dCMOEA	dCMOEA
	2nd	DC-NSGA-II	DC-NSGA-II	DC-NSGA-II
	3rd	DC-TAEA	DC-NSGA-III/DC-TAEA	DC-TAEA
	4th	DC-MOEA	DC-MOEA	DC-MOEA
	5th	DC-NSGA-III	DC-NSGA-II-A	DC-NSGA-III
	6th	DC-NSGA-II-A		DC-NSGA-II-A
5	1th	dCMOEA	dCMOEA	dCMOEA
	2nd	DC-NSGA-II	DC-NSGA-II	DC-NSGA-II
	3rd	DC-NSGA-III	DC-NSGA-III	DC-TAEA
	4th	DC-TAEA	DC-TAEA	DC-MOEA
	5th	DC-MOEA	DC-MOEA	DC-NSGA-III
	6th	DC-NSGA-II-A	DC-NSGA-II-A	DC-NSGA-II-A
6	1th	dCMOEA	dCMOEA	dCMOEA
	2nd	DC-NSGA-II	DC-NSGA-II	DC-NSGA-II
	3rd	DC-TAEA	DC-NSGA-III	DC-TAEA
	4th	DC-NSGA-III	DC-TAEA	DC-MOEA
	5th	DC-MOEA	DC-MOEA	DC-NSGA-III
	6th	DC-NSGA-II-A	DC-NSGA-II-A	DC-NSGA-II-A
7	1th	dCMOEA	dCMOEA	dCMOEA
	2nd	DC-NSGA-II	DC-NSGA-II	DC-NSGA-II
	3rd	DC-NSGA-III	DC-NSGA-III	DC-TAEA
	4th	DC-TAEA	DC-TAEA	DC-MOEA
	5th	DC-MOEA	DC-MOEA	DC-NSGA-III
	6th	DC-NSGA-II-A	DC-NSGA-II-A	DC-NSGA-II-A
8	1th	dCMOEA	dCMOEA	dCMOEA
	2nd	DC-NSGA-II	DC-NSGA-II	DC-NSGA-II
	3rd	DC-NSGA-III/DC-MOEA	DC-NSGA-III	DC-TAEA
	4th	DC-TAEA	DC-TAEA	DC-MOEA
	5th	DC-NSGA-II-A	DC-MOEA	DC-NSGA-III
	6th		DC-NSGA-II-A	DC-NSGA-II-A

TABLE S-V  
THE COMPUTATIONAL TIME OF EACH ALGORITHM ON EACH TEST PROBLEM WITH  $\tau_r=10, 15, \text{ AND } 20$

		DC-MOEA	DC-NSGA-II-A	DC-NSGA-II	DC-NSGA-III	DC-TAEA	dCMOEA
Ins 1	$\tau_r=10$	1079.3	1158.0	1309.2	919.9	16391.7	1565.7
	$\tau_r=15$	1813.3	1831.4	2230.8	1497.9	24944.5	2437.0
	$\tau_r=20$	2372.4	2373.2	2966.1	2030.3	30668.0	3236.8
Ins 2	$\tau_r=10$	991.9	1162.3	1262.9	896.5	13091.8	1498.4
	$\tau_r=15$	1668.7	1808.6	2005.4	1470.7	19857.0	2238.7
	$\tau_r=20$	2261.4	2276.9	2918.7	2013.1	29274.4	3244.2
Ins 3	$\tau_r=10$	958.9	1145.2	1230.3	829.0	15897.3	1357.7
	$\tau_r=15$	1622.6	1795.1	1954.6	1353.8	24041.3	2115.9
	$\tau_r=20$	2146.6	2155.2	2845.4	1894.7	30641.0	3179.5
Ins 4	$\tau_r=10$	1038.1	1032.6	1310.9	958.7	15046.7	1369.3
	$\tau_r=15$	1790.8	1826.5	2069.9	1441.9	18841.1	2110.8
	$\tau_r=20$	2369.8	2052.2	3036.2	2108.4	25892.7	3356.4
Ins 5	$\tau_r=10$	1060.9	1008.0	1288.3	890.5	15891.3	1346.9
	$\tau_r=15$	1726.1	1747.7	2031.5	1391.1	24171.7	2073.8
	$\tau_r=20$	2388.9	2047.9	2913.0	2037.9	31323.5	3246.0
Ins 6	$\tau_r=10$	986.2	1037.0	1267.4	893.7	14548.5	1415.6
	$\tau_r=15$	1569.7	1710.7	1836.7	1361.2	21218.1	2036.4
	$\tau_r=20$	2204.7	2080.4	2820.7	1976.7	30785.3	3007.4
Ins 7	$\tau_r=10$	961.2	997.7	1191.4	838.5	17493.0	1453.6
	$\tau_r=15$	1506.9	1683.8	1790.2	1299.2	24459.3	2196.6
	$\tau_r=20$	2124.6	2237.4	2463.1	1882.8	31751.0	3160.1
Ins 8	$\tau_r=10$	1041.9	1148.5	1231.8	934.0	12883.9	1589.9
	$\tau_r=15$	1661.4	1722.1	1902.8	1436.9	18468.9	2396.4
	$\tau_r=20$	2220.2	2357.7	3018.7	2073.7	24909.8	3112.5

ON THE EIGHT TEST PROBLEMS

Figures S-5–S-12 show the obtained POFs obtained by the compared algorithms on the eight test problems.

DC-NSGA-III, and dCMOEA are given in Table S-VI.

As seen from Table S-VI, dCMOEA performs significantly better than DC-MOEA, DC-NSGA-II-A, DC-NSGA-II, and DC-NSGA-III on most of the test problems in terms of the three metrics. Therefore, even with the same running time, the proposed algorithm still outperforms the others (i.e., DC-MOEA, DC-NSGA-II-A, DC-NSGA-II, and DC-NSGA-III).

S-VIII. EVOLUTION CURVES OF AVERAGE IGD VALUES FOR THE THIRD TO EIGHT TEST PROBLEMS

Fig. S-4 depicts the evolution curves of average IGD values for the third to eighth test problems with  $\tau_r=10$  and  $n_r=21$ .

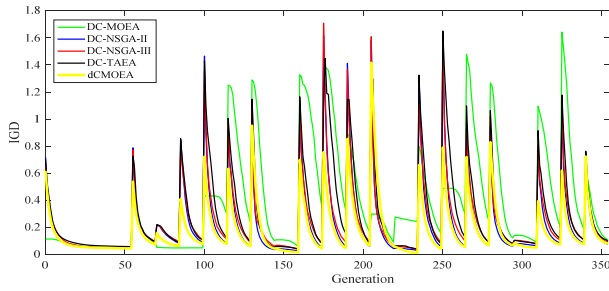
S-IX. THE POFs OBTAINED BY THE COMPARED ALGORITHMS



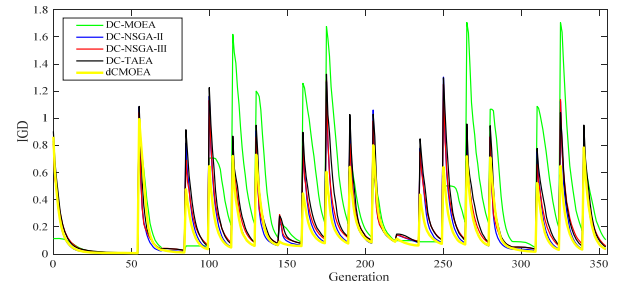
TABLE S-VI  
PERFORMANCE METRICS OF DC-MOEA, DC-NSGA-II-A, DC-NSGA-II, DC-NSGA-III, AND dCMOEA WHEN THE TERMINATION CRITERION OF EACH DYNAMIC ENVIRONMENT IS THE ELAPSED CPU TIME LIMIT OF CP=2000MS

Ins	Indicator	DC-MOEA	DC-NSGA-II-A	DC-NSGA-II	DC-NSGA-III	dCMOEA
1	HV	3.42028(0.01367) <sup>‡</sup>	1.95596(0.06050) <sup>‡</sup>	3.46001(0.00800) <sup>‡</sup>	3.46607(0.00762) <sup>‡</sup>	<b>3.48292(0.00540)</b>
	IGD	0.06530(0.00479) <sup>‡</sup>	0.55941(0.02783) <sup>‡</sup>	0.04748(0.00217) <sup>‡</sup>	<b>0.04372(0.00206)<sup>†</sup></b>	0.04407(0.00208)
	SP	0.02100(0.00190) <sup>‡</sup>	0.28754(0.11983) <sup>‡</sup>	0.01725(0.00127) <sup>‡</sup>	0.04929(0.01621) <sup>‡</sup>	<b>0.01476(0.00127)</b>
2	HV	3.41652(0.00656) <sup>‡</sup>	2.03286(0.05694) <sup>‡</sup>	3.43709(0.01024) <sup>‡</sup>	3.43683(0.00680) <sup>‡</sup>	<b>3.45826(0.00372)</b>
	IGD	0.05959(0.00242) <sup>‡</sup>	0.57744(0.03328) <sup>‡</sup>	0.04786(0.00267) <sup>‡</sup>	<b>0.04489(0.00169)<sup>†</sup></b>	0.04518(0.00150)
	SP	0.01805(0.00147) <sup>‡</sup>	0.27087(0.05832) <sup>‡</sup>	0.01666(0.00356) <sup>‡</sup>	0.05629(0.01778) <sup>‡</sup>	<b>0.01472(0.00131)</b>
3	HV	3.45002(0.01501) <sup>‡</sup>	1.96126(0.07233) <sup>‡</sup>	3.45606(0.00884) <sup>‡</sup>	3.45979(0.00977) <sup>‡</sup>	<b>3.47854(0.00422)</b>
	IGD	0.06611(0.00462) <sup>‡</sup>	0.56339(0.03208) <sup>‡</sup>	0.04957(0.00243) <sup>‡</sup>	<b>0.04593(0.00183)<sup>†</sup></b>	0.04562(0.00150)
	SP	0.02681(0.00352) <sup>‡</sup>	0.25218(0.06797) <sup>‡</sup>	0.01938(0.00157) <sup>‡</sup>	0.05469(0.01309) <sup>‡</sup>	<b>0.01607(0.00128)</b>
4	HV	<b>3.48568(0.00659)</b>	1.99611(0.06786) <sup>‡</sup>	3.43254(0.00786) <sup>‡</sup>	3.42957(0.00684) <sup>‡</sup>	3.45491(0.00521)
	IGD	0.06292(0.00208) <sup>‡</sup>	0.57849(0.03928) <sup>‡</sup>	0.04553(0.00156) <sup>‡</sup>	0.04308(0.00158) <sup>‡</sup>	<b>0.04118(0.00246)</b>
	SP	0.01764(0.00102) <sup>‡</sup>	0.31397(0.08128) <sup>‡</sup>	0.01532(0.00112) <sup>‡</sup>	0.05786(0.01552) <sup>‡</sup>	<b>0.01345(0.00122)</b>
5	HV	3.42831(0.01292) <sup>‡</sup>	1.90029(0.06814) <sup>‡</sup>	3.46388(0.01690) <sup>‡</sup>	<sup>‡</sup> 3.47716(0.01092) <sup>‡</sup>	<b>3.51900(0.00241)</b>
	IGD	0.07515(0.00350) <sup>‡</sup>	0.57517(0.03168) <sup>‡</sup>	0.06155(0.00427) <sup>‡</sup>	0.05541(0.00177) <sup>‡</sup>	<b>0.05032(0.00113)</b>
	SP	0.02327(0.00171) <sup>‡</sup>	0.30633(0.10234) <sup>‡</sup>	0.01846(0.00154) <sup>‡</sup>	0.04784(0.01133) <sup>‡</sup>	<b>0.01362(0.00102)</b>
6	HV	3.41270(0.00803) <sup>‡</sup>	1.96625(0.06804) <sup>‡</sup>	3.43540(0.01153) <sup>‡</sup>	3.43478(0.00637) <sup>‡</sup>	<b>3.46965(0.00248)</b>
	IGD	0.06720(0.00220) <sup>‡</sup>	0.59598(0.03641) <sup>‡</sup>	0.05788(0.00261) <sup>‡</sup>	0.05435(0.00108) <sup>‡</sup>	<b>0.05048(0.00095)</b>
	SP	0.01990(0.00123) <sup>‡</sup>	0.30314(0.07770) <sup>‡</sup>	0.01696(0.00171) <sup>‡</sup>	0.06117(0.01958) <sup>‡</sup>	<b>0.01327(0.00131)</b>
7	HV	3.46214(0.03329) <sup>‡</sup>	1.91381(0.05672) <sup>‡</sup>	3.45509(0.01810) <sup>‡</sup>	3.46492(0.01639) <sup>‡</sup>	<b>3.51152(0.00457)</b>
	IGD	0.07860(0.00794) <sup>‡</sup>	0.58786(0.03195) <sup>‡</sup>	0.06312(0.00434) <sup>‡</sup>	0.05815(0.00349) <sup>‡</sup>	<b>0.05195(0.00106)</b>
	SP	0.03057(0.00472) <sup>‡</sup>	0.27717(0.06678) <sup>‡</sup>	0.02062(0.00239) <sup>‡</sup>	0.05180(0.01481) <sup>‡</sup>	<b>0.01569(0.00124)</b>
8	HV	<b>3.49254(0.00730)</b>	1.93954(0.06004) <sup>‡</sup>	3.43038(0.01384) <sup>‡</sup>	3.43042(0.00865) <sup>‡</sup>	3.46412(0.00324)
	IGD	0.07078(0.00279) <sup>‡</sup>	0.60735(0.03522) <sup>‡</sup>	0.05584(0.00343) <sup>‡</sup>	0.05320(0.00155) <sup>‡</sup>	<b>0.04938(0.00145)</b>
	SP	0.01881(0.00186) <sup>‡</sup>	0.29879(0.07863) <sup>‡</sup>	0.01519(0.00180) <sup>‡</sup>	0.06691(0.01500) <sup>‡</sup>	<b>0.01231(0.00100)</b>

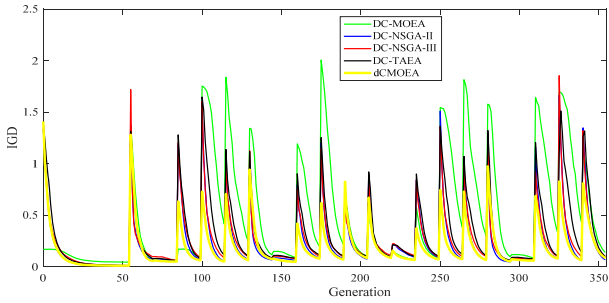
‡ and † indicate dCMOEA performs significantly better than and equivalently to the corresponding algorithm, respectively.



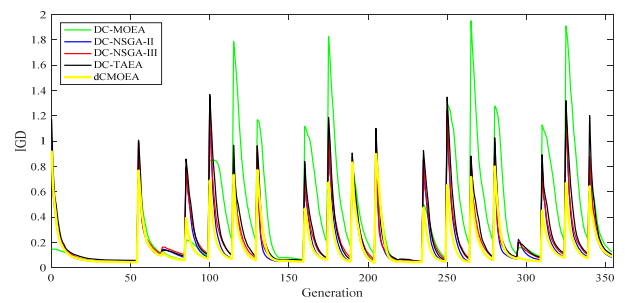
Instance 3



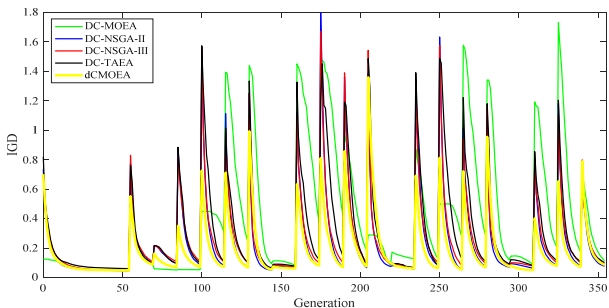
Instance 4



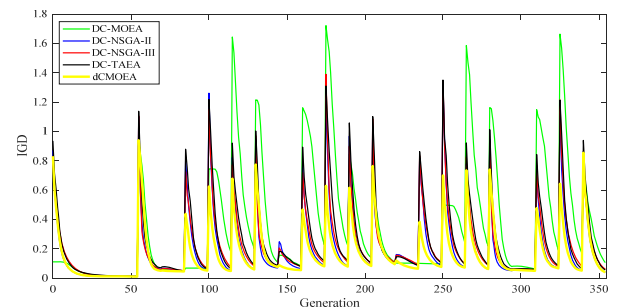
Instance 5



Instance 6



Instance 7



Instance 8

Fig. S-4. Evolution curves of average IGD values for the third to eighth test problems with  $n=15$  and  $n=21$ .

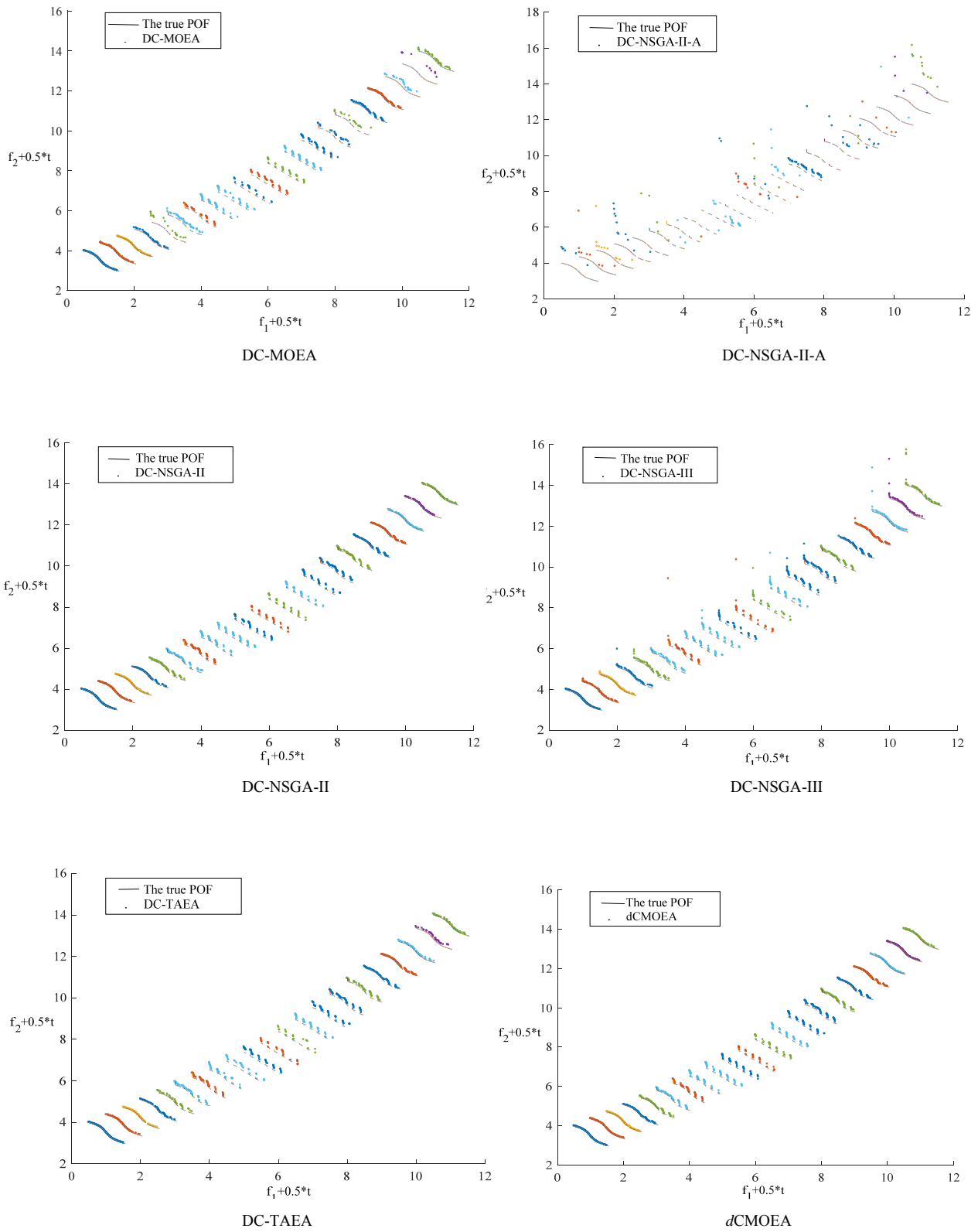


Fig. S-5. Obtained POFs obtained by the compared algorithms on the first test problem.

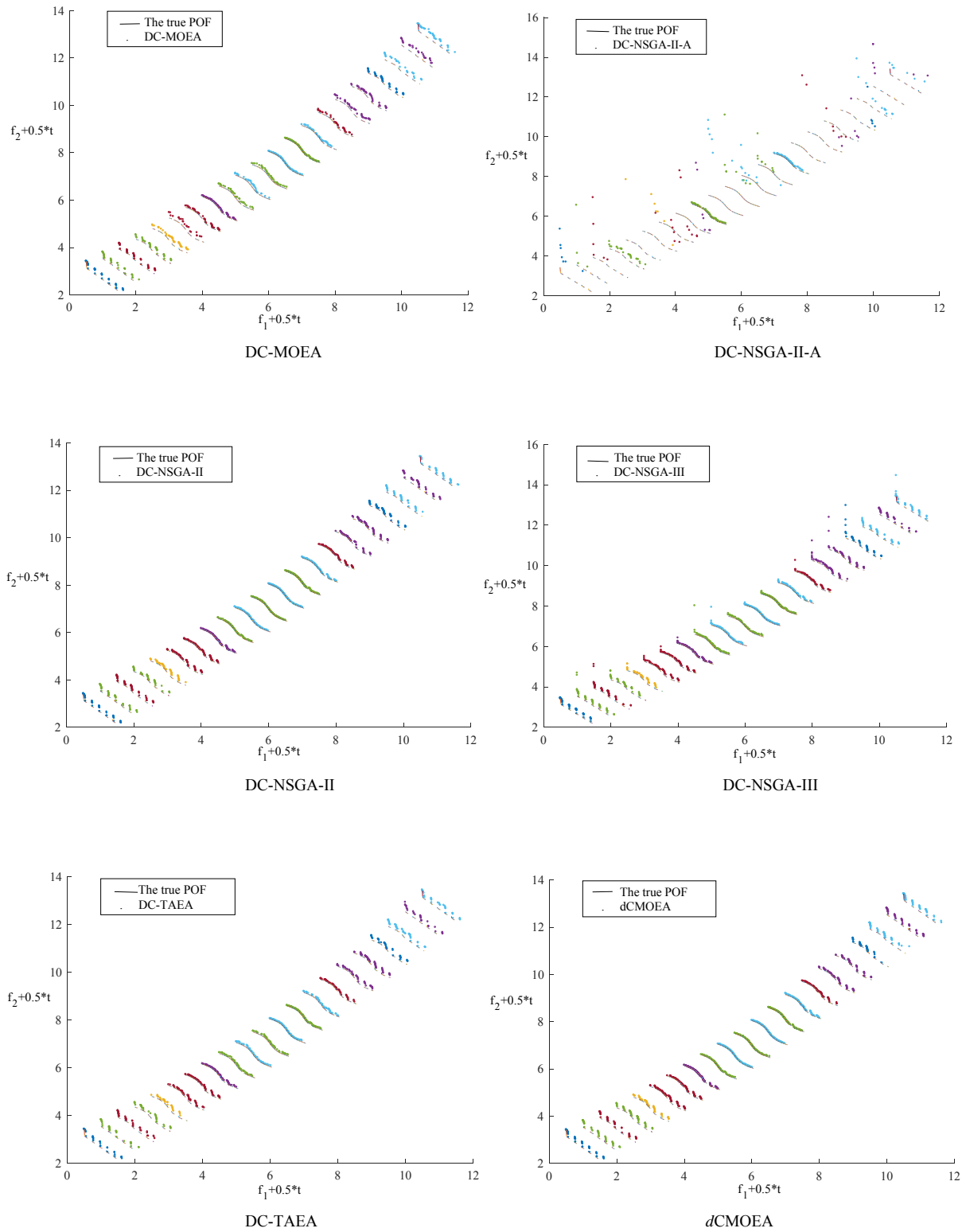


Fig. S-6. Obtained POFs obtained by the compared algorithms on the second test problem.

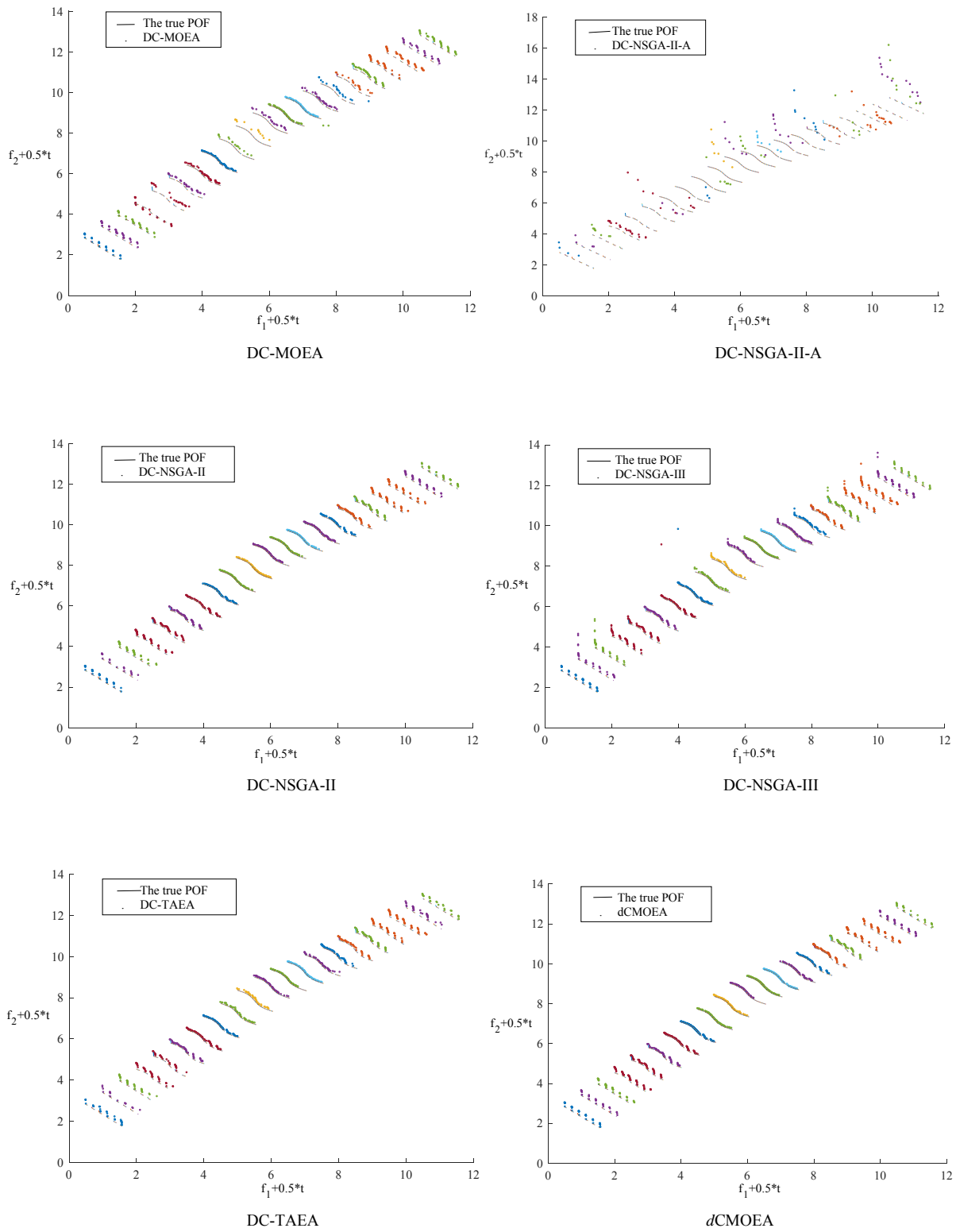


Fig. S-7. Obtained POFs obtained by the compared algorithms on the third test problems.

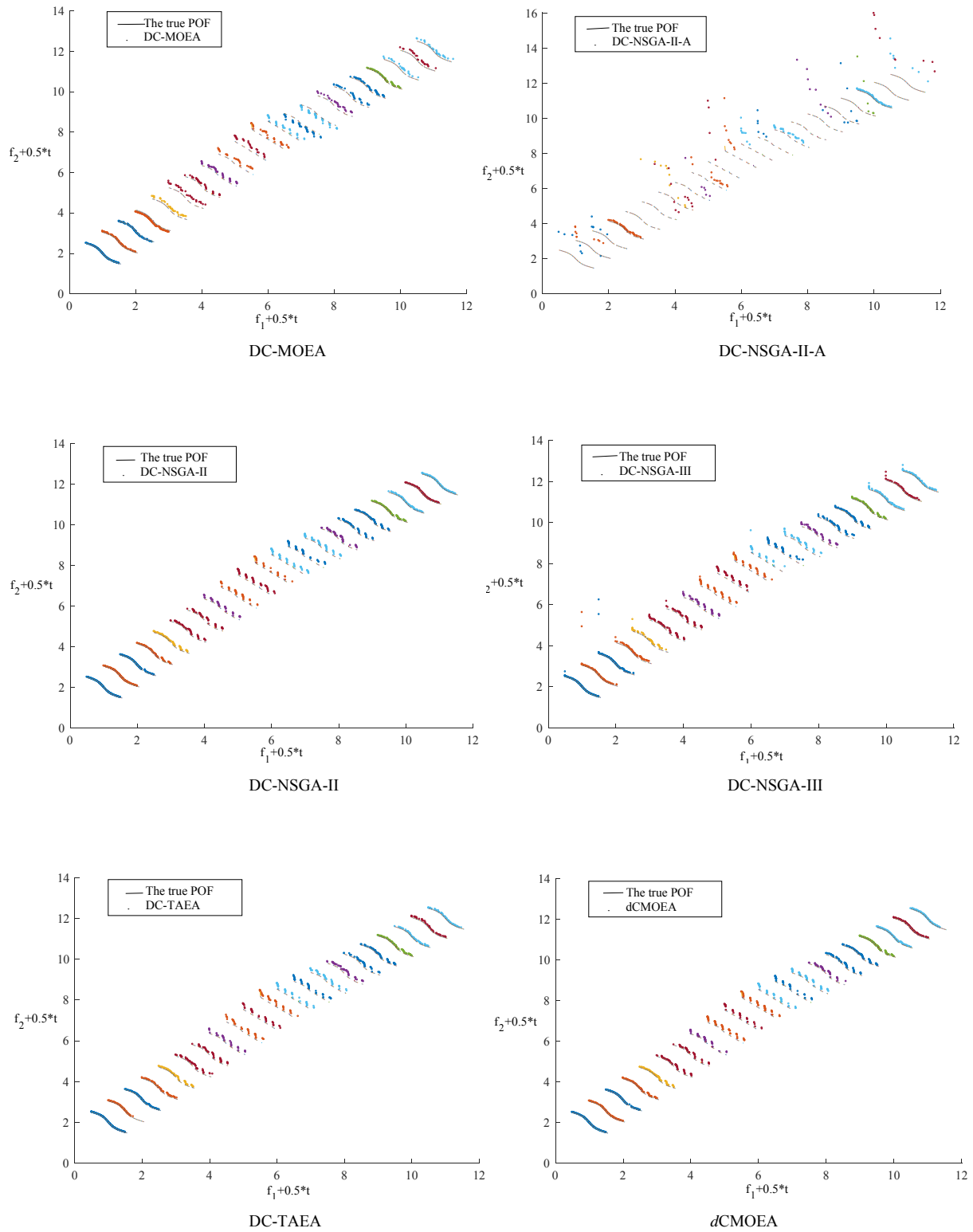


Fig. S-8. Obtained POFs obtained by the compared algorithms on the fourth test problem.

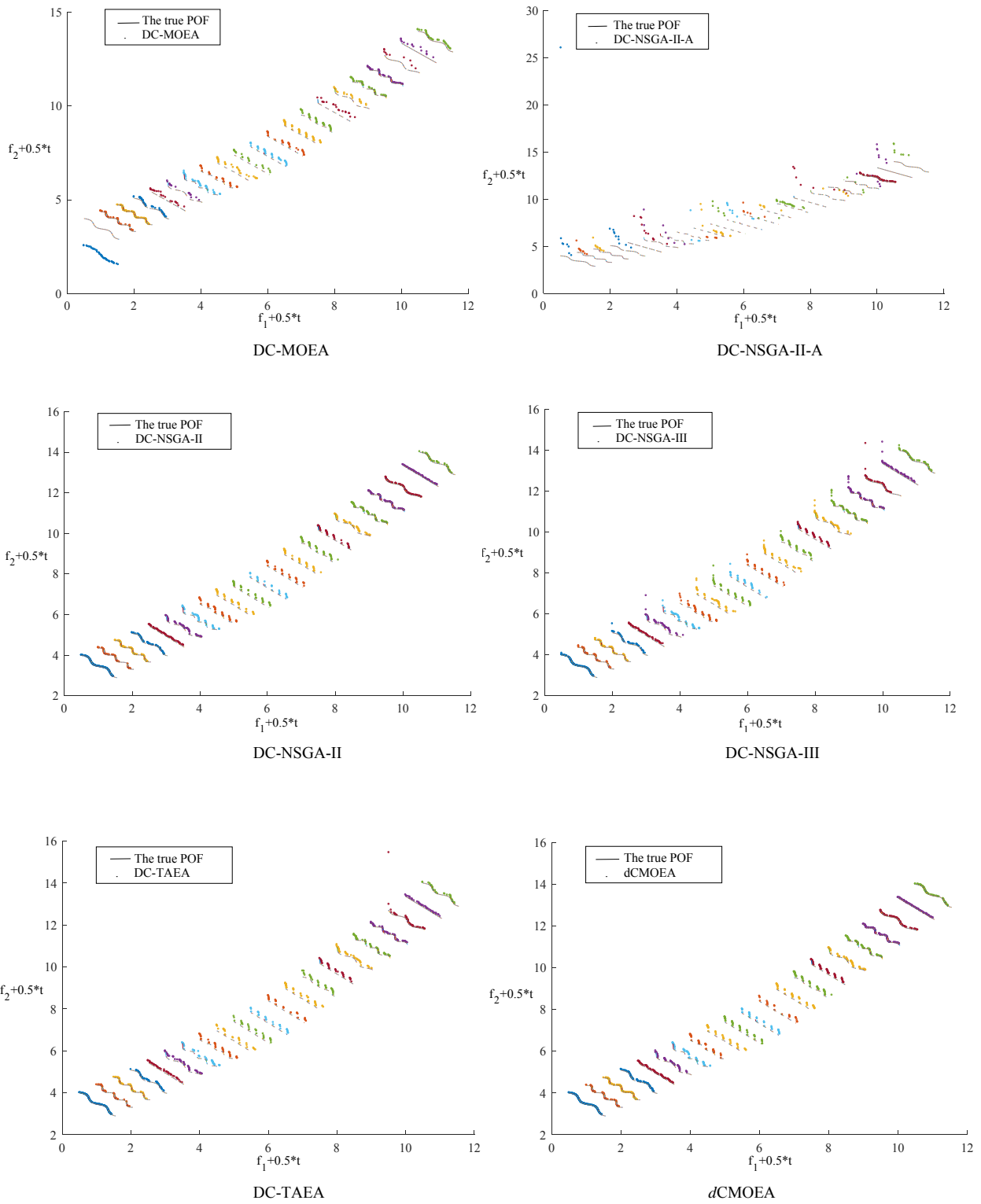


Fig. S-9. Obtained POFs obtained by the compared algorithms on the fifth test problem.

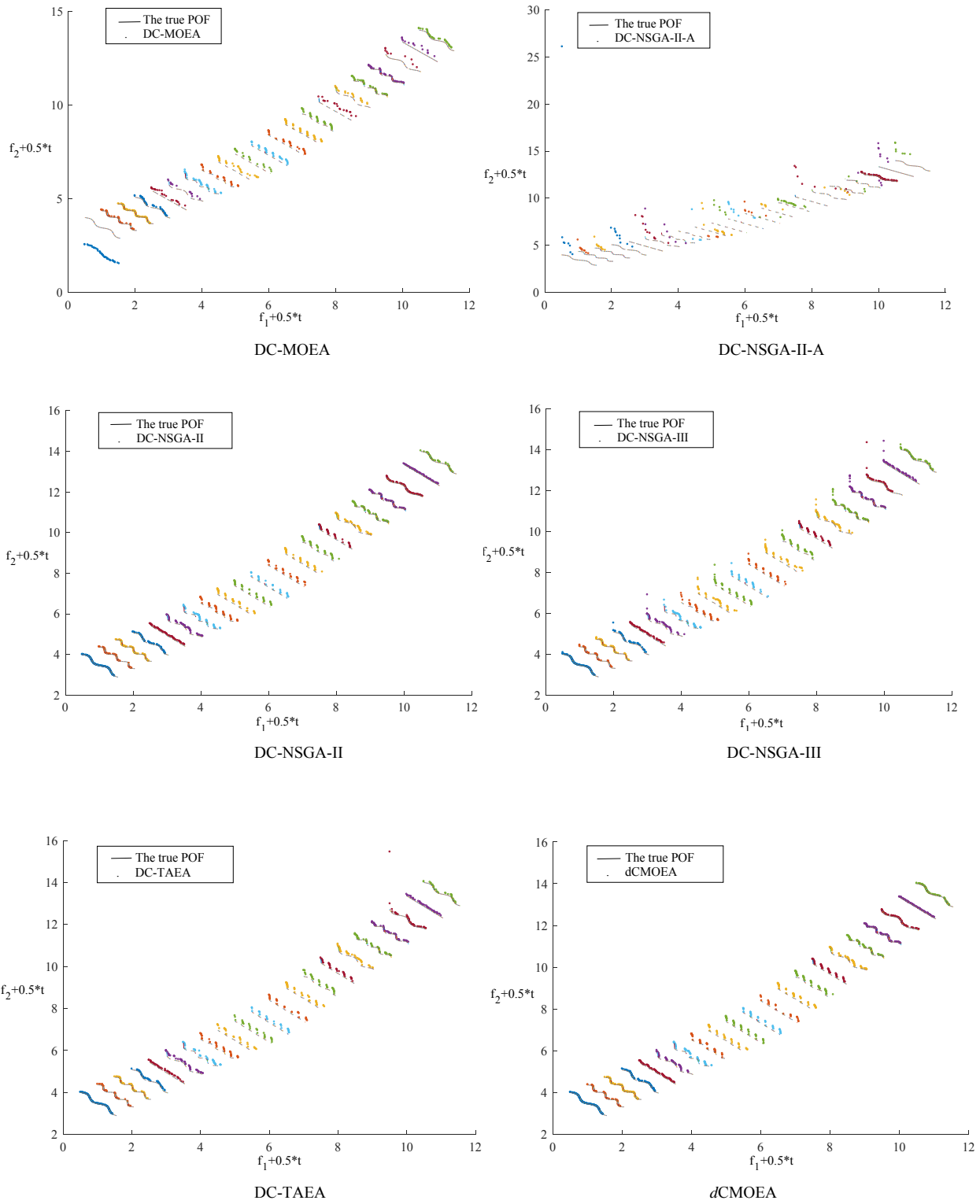


Fig. S-10. Obtained POFs obtained by the compared algorithms on the fifth test problem.

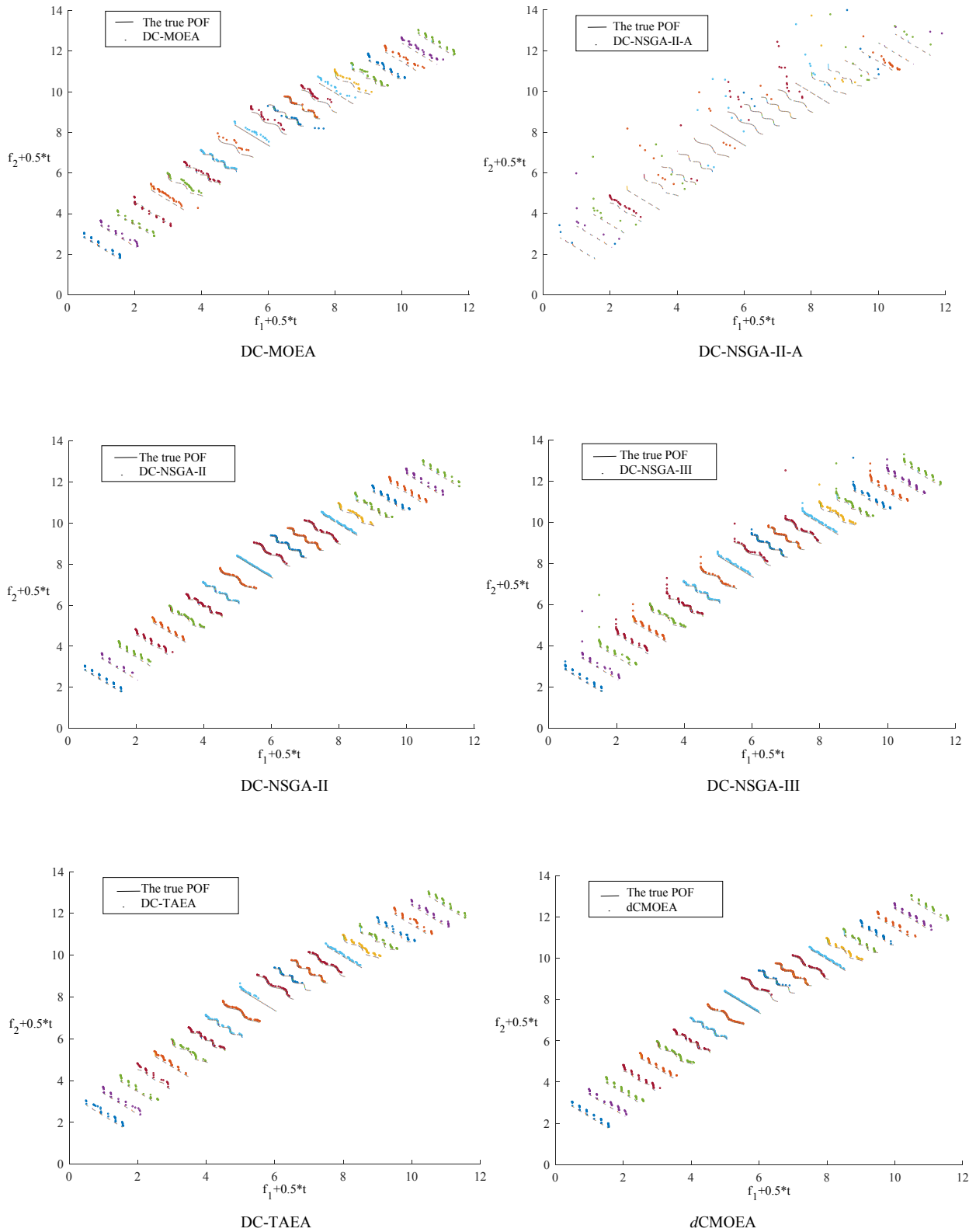


Fig. S-11. Obtained POFs obtained by the compared algorithms on the seventh test problem.



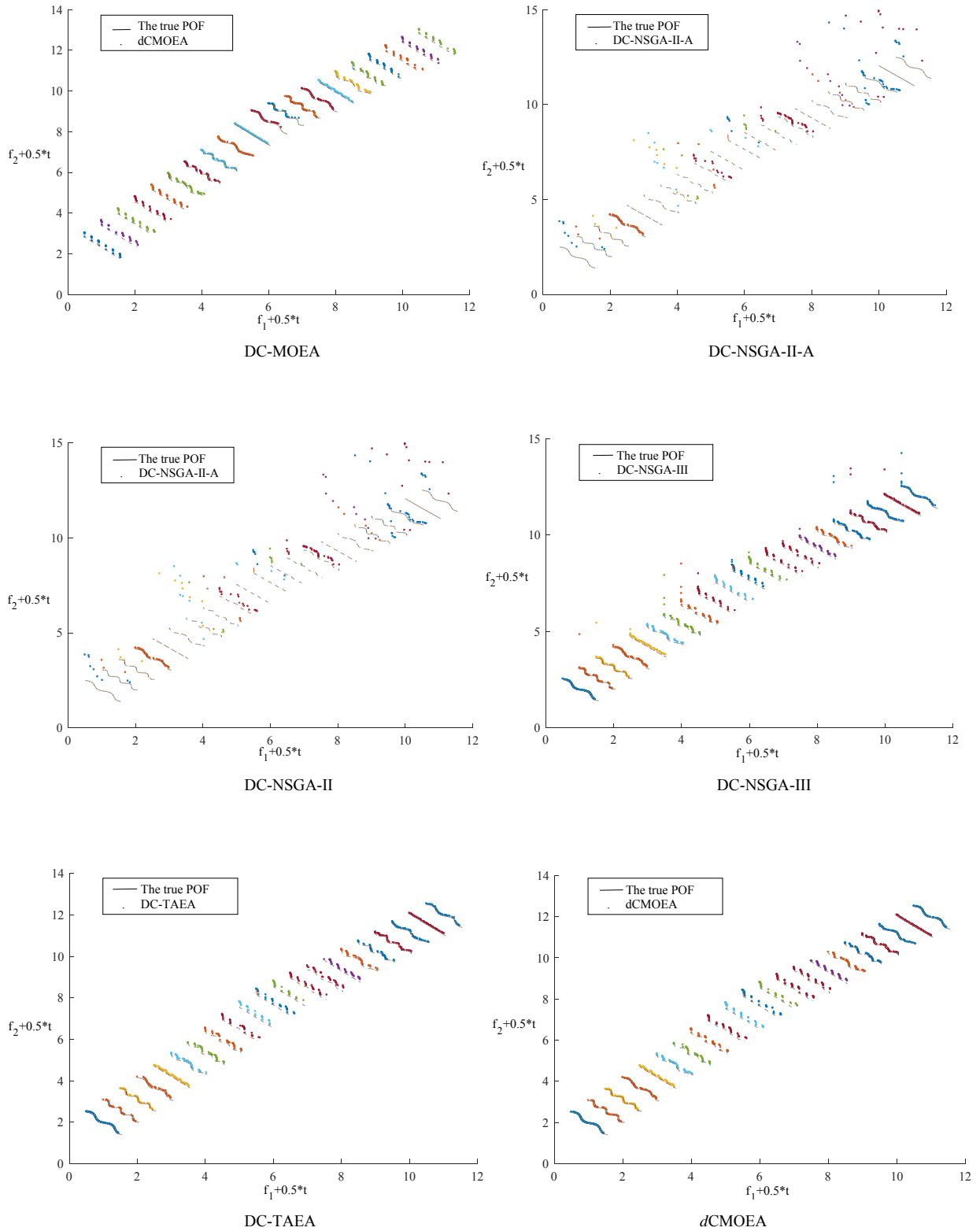


Fig. S-12. Obtained POFs obtained by the compared algorithms on the eighth test problem.

TABLE S-VII

STATISTIC RESULTS OF THE THREE COMPARED ALGORITHMS ON TEST PROBLEMS WITH 15 DECISION VARIABLES AND  $\tau=10$ 

Ins	Indicator	DC-NSGA-II	DC-NSGA-III	dCMOEA
1	HV	2.95561(0.03845)	2.71664(0.04526)	<b>3.07657(0.04432)</b>
	IGD	0.22164(0.01072)	0.28576(0.01551)	<b>0.19730(0.01493)</b>
	SP	0.08249(0.02708)	0.25777(0.05624)	<b>0.04468(0.00519)</b>
2	HV	3.04555(0.03332)	2.82611(0.02948)	<b>3.17697(0.03777)</b>
	IGD	0.18786(0.01023)	0.24986(0.00798)	<b>0.15900(0.01451)</b>
	SP	0.06232(0.01547)	0.24800(0.04405)	<b>0.03812(0.00370)</b>
3	HV	2.93815(0.03630)	2.70643(0.03838)	<b>3.08019(0.05695)</b>
	IGD	0.22849(0.01343)	0.28738(0.01364)	<b>0.19555(0.02247)</b>
	SP	0.08378(0.02467)	0.24008(0.04629)	<b>0.04671(0.00825)</b>
4	HV	3.04671(0.02292)	2.82254(0.03190)	<b>3.14364(0.02389)</b>
	IGD	0.18369(0.00673)	0.24763(0.00949)	<b>0.16633(0.00858)</b>
	SP	0.06516(0.01861)	0.23242(0.04734)	<b>0.03724(0.00282)</b>
5	HV	2.92605(0.04016)	2.72323(0.03660)	<b>3.08923(0.05100)</b>
	IGD	0.23497(0.01344)	0.29468(0.01245)	<b>0.19696(0.01791)</b>
	SP	0.08598(0.01794)	0.26784(0.05613)	<b>0.04626(0.00428)</b>
6	HV	3.02309(0.03391)	2.82055(0.03017)	<b>3.17003(0.02427)</b>
	IGD	0.19389(0.01060)	0.25246(0.00982)	<b>0.15876(0.00768)</b>
	SP	0.06429(0.01623)	0.25048(0.04104)	<b>0.04156(0.00548)</b>
7	HV	2.94013(0.03380)	2.71784(0.03460)	<b>3.07288(0.05690)</b>
	IGD	0.23214(0.01205)	0.29627(0.01335)	<b>0.20485(0.02385)</b>
	SP	0.08255(0.01929)	0.23726(0.04453)	<b>0.04583(0.00621)</b>
8	HV	3.02142(0.03148)	2.81701(0.03274)	<b>3.13580(0.03975)</b>
	IGD	0.19183(0.00998)	0.25197(0.00866)	<b>0.16869(0.01484)</b>
	SP	0.06223(0.01409)	0.23989(0.04734)	<b>0.04035(0.00372)</b>

TABLE S-VIII

STATISTIC RESULTS OF THE THREE COMPARED ALGORITHMS ON TEST PROBLEMS WITH 15 DECISION VARIABLES AND  $\tau=15$ 

Ins	Indicator	DC-NSGA-II	DC-NSGA-III	dCMOEA
1	HV	3.26166(0.02473)	3.10064(0.02337)	<b>3.33491(0.01589)</b>
	IGD	0.11810(0.00769)	0.16809(0.00737)	<b>0.10484(0.00628)</b>
	SP	0.03541(0.01141)	0.18341(0.04286)	<b>0.02200(0.00147)</b>
2	HV	3.29301(0.01727)	3.16679(0.02114)	<b>3.35993(0.01110)</b>
	IGD	0.10191(0.00455)	0.14221(0.00543)	<b>0.09081(0.00400)</b>
	SP	0.02771(0.00602)	0.15573(0.03687)	<b>0.01961(0.00152)</b>
3	HV	3.25308(0.03219)	3.08977(0.03268)	<b>3.32878(0.03004)</b>
	IGD	0.12021(0.01027)	0.17080(0.00940)	<b>0.10736(0.01189)</b>
	SP	0.03609(0.01153)	0.17350(0.03904)	<b>0.02333(0.00159)</b>
4	HV	3.29219(0.01704)	3.15399(0.02251)	<b>3.34662(0.01393)</b>
	IGD	0.09763(0.00456)	0.14145(0.00732)	<b>0.09101(0.00536)</b>
	SP	0.02634(0.00345)	0.18179(0.03329)	<b>0.01791(0.00120)</b>
5	HV	3.24134(0.02396)	3.09250(0.03797)	<b>3.34417(0.02430)</b>
	IGD	0.12794(0.00672)	0.17548(0.01275)	<b>0.10806(0.00797)</b>
	SP	0.03258(0.00305)	0.19911(0.05104)	<b>0.02368(0.00175)</b>
6	HV	3.28364(0.02645)	3.15064(0.03038)	<b>3.35260(0.01125)</b>
	IGD	0.10561(0.00655)	0.14745(0.00760)	<b>0.09415(0.00366)</b>
	SP	0.02935(0.00445)	0.16061(0.03214)	<b>0.02030(0.00150)</b>
7	HV	3.23989(0.03297)	3.11045(0.03148)	<b>3.34284(0.03295)</b>
	IGD	0.12949(0.00921)	0.16992(0.00901)	<b>0.10903(0.01232)</b>
	SP	0.03391(0.00427)	0.16614(0.03040)	<b>0.02535(0.00201)</b>
8	HV	3.27579(0.03015)	3.14596(0.02475)	<b>3.34292(0.02463)</b>
	IGD	0.10535(0.00700)	0.14512(0.00644)	<b>0.09372(0.00799)</b>
	SP	0.02774(0.00395)	0.18234(0.04376)	<b>0.01928(0.00151)</b>

## S-X. THREE PERFORMANCE METRICS OF ALGORITHMS ON THE TEST PROBLEMS WITH 15 AND 20 DECISION VARIABLES

Tables S-VII–S-IX give the values and standard deviations of the three performance metrics (i.e., HV, IGD, and MS) for these algorithms (i.e., DC-NSGA-II, DC-NSGA-III, and dCMOEA) on the test problems with 15 decision variables. The results of these algorithms on the test problems with 20 decision variables are given in Tables S-X–S-XII.

[s1] M. Alhamdoosh and D. H. Wang, "Fast decorrelated neural network ensembles with random weights," *Inf. Sci.*, vol. 264, no. 6, pp. 104–117, Apr. 2014.

[s2] Y. M. John, R. Patel, and I. M. Mujtaba, "Maximization of gasoline in an industrial fluidized catalytic cracking unit," *Energ. Fuel*, vol. 31, no. 5, pp. 5645–5661, 2017.

TABLE S-IX

STATISTIC RESULTS OF THE THREE COMPARED ALGORITHMS ON TEST PROBLEMS WITH 15 DECISION VARIABLES AND  $\tau=20$ 

Ins	Indicator	DC-NSGA-II	DC-NSGA-III	dCMOEA
1	HV	3.37414(0.01367)	3.27083(0.01930)	<b>3.42926(0.00895)</b>
	IGD	0.07672(0.00324)	0.11075(0.00496)	<b>0.06703(0.00412)</b>
	SP	0.02140(0.00425)	0.14309(0.04702)	<b>0.01508(0.00112)</b>
2	HV	3.38071(0.01683)	3.29468(0.01822)	<b>3.42353(0.01279)</b>
	IGD	0.07026(0.00542)	0.09715(0.00497)	<b>0.06352(0.00595)</b>
	SP	0.01931(0.00429)	0.11906(0.02749)	<b>0.01346(0.00095)</b>
3	HV	3.36813(0.01459)	3.27253(0.02240)	<b>3.42217(0.01544)</b>
	IGD	0.07973(0.00350)	0.11039(0.00496)	<b>0.06980(0.00591)</b>
	SP	0.02187(0.00139)	0.13585(0.02686)	<b>0.01631(0.00129)</b>
4	HV	3.37570(0.01632)	3.28654(0.01673)	<b>3.41747(0.00465)</b>
	IGD	0.06799(0.00457)	0.09478(0.00384)	<b>0.06069(0.00234)</b>
	SP	0.01167(0.00113)	0.13267(0.03044)	<b>0.01255(0.00094)</b>
5	HV	3.37069(0.02909)	3.27013(0.02577)	<b>3.44924(0.01267)</b>
	IGD	0.08642(0.00629)	0.11615(0.00740)	<b>0.07260(0.00364)</b>
	SP	0.02265(0.00157)	0.13963(0.04095)	<b>0.01597(0.00138)</b>
6	HV	3.37158(0.01845)	3.29438(0.01977)	<b>3.41892(0.00837)</b>
	IGD	0.07657(0.00428)	0.09920(0.00410)	<b>0.06948(0.00322)</b>
	SP	0.01909(0.00161)	0.12696(0.03420)	<b>0.01441(0.00103)</b>
7	HV	3.36137(0.03189)	3.26438(0.02859)	<b>3.44012(0.01927)</b>
	IGD	0.08931(0.00761)	0.11806(0.00687)	<b>0.07370(0.00277)</b>
	SP	0.02462(0.00316)	0.12776(0.03288)	<b>0.01776(0.00102)</b>
8	HV	3.36956(0.01475)	3.28433(0.01690)	<b>3.41302(0.00614)</b>
	IGD	0.07456(0.00337)	0.09937(0.00416)	<b>0.06758(0.00242)</b>
	SP	0.01836(0.00174)	0.13312(0.03946)	<b>0.01381(0.00080)</b>

TABLE S-X

STATISTIC RESULTS OF THE THREE COMPARED ALGORITHMS ON TEST PROBLEMS WITH 20 DECISION VARIABLES AND  $\tau=10$ 

Ins	Indicator	DC-NSGA-II	DC-NSGA-III	dCMOEA
1	HV	2.46101(0.06867)	2.15308(0.05064)	<b>2.60879(0.06029)</b>
	IGD	0.39943(0.02527)	0.49303(0.02918)	<b>0.36733(0.02349)</b>
	SP	0.11399(0.03403)	0.47251(0.08141)	<b>0.07339(0.01500)</b>
2	HV	2.65836(0.03044)	2.32834(0.04449)	<b>2.84469(0.03805)</b>
	IGD	0.32515(0.01001)	0.42900(0.01758)	<b>0.27805(0.01420)</b>
	SP	0.08591(0.01680)	0.39985(0.10046)	<b>0.05793(0.00707)</b>
3	HV	2.44064(0.05276)	2.12862(0.04705)	<b>2.64504(0.06816)</b>
	IGD	0.40801(0.02062)	0.51164(0.01780)	<b>0.35603(0.03056)</b>
	SP	0.12594(0.03605)	0.40724(0.08440)	<b>0.06637(0.01018)</b>
4	HV	2.63752(0.04306)	2.33186(0.04964)	<b>2.81244(0.06428)</b>
	IGD	0.32536(0.01532)	0.42376(0.01982)	<b>0.28422(0.02353)</b>
	SP	0.09582(0.02913)	0.42987(0.08518)	<b>0.05863(0.00719)</b>
5	HV	2.42237(0.06522)	2.12443(0.06244)	<b>2.62456(0.06490)</b>
	IGD	0.42872(0.02622)	0.52313(0.02623)	<b>0.36884(0.02601)</b>
	SP	0.12531(0.03076)	0.42483(0.08529)	<b>0.07403(0.01646)</b>
6	HV	2.60910(0.04052)	2.31038(0.05384)	<b>2.84768(0.03996)</b>
	IGD	0.34569(0.01647)	0.44344(0.02366)	<b>0.27439(0.01585)</b>
	SP	0.11013(0.02887)	0.42365(0.08678)	<b>0.06443(0.00825)</b>
7	HV	2.40782(0.05399)	2.10162(0.05985)	<b>2.62790(0.07623)</b>
	IGD	0.43175(0.02467)	0.53468(0.02545)	<b>0.36861(0.03287)</b>
	SP	0.12788(0.03366)	0.40795(0.08804)	<b>0.06936(0.00939)</b>
8	HV	2.62186(0.04274)	2.28635(0.05430)	<b>2.80565(0.05456)</b>
	IGD	0.33446(0.01578)	0.45240(0.02144)	<b>0.28490(0.01931)</b>
	SP	0.09567(0.02576)	0.41556(0.07941)	<b>0.06207(0.00705)</b>

[s3] D. C. Montgomery, *Design and Analysis of Experiments*. Hoboken, NJ, USA: Wiley, 2005.

TABLE S-XI

STATISTIC RESULTS OF THE THREE COMPARED ALGORITHMS ON TEST PROBLEMS WITH 20 DECISION VARIABLES AND  $\tau=15$ 

Ins	Indicator	DC-NSGA-II	DC-NSGA-III	dCMOEA
1	<i>HV</i>	2.99175(0.03964)	2.73604(0.05486)	<b>3.09502(0.04121)</b>
	<i>IGD</i>	0.21273(0.01448)	0.29390(0.01775)	<b>0.19442(0.01614)</b>
	<i>SP</i>	0.04755(0.00557)	0.33048(0.08027)	<b>0.03316(0.00361)</b>
2	<i>HV</i>	3.10505(0.02876)	2.86168(0.02896)	<b>3.21185(0.01903)</b>
	<i>IGD</i>	0.16986(0.00899)	0.24754(0.00969)	<b>0.14988(0.00718)</b>
	<i>SP</i>	0.03864(0.00844)	0.30651(0.07453)	<b>0.02627(0.00244)</b>
3	<i>HV</i>	2.98844(0.03836)	2.72258(0.05321)	<b>3.08711(0.06343)</b>
	<i>IGD</i>	0.21324(0.01382)	0.29803(0.01852)	<b>0.19860(0.02728)</b>
	<i>SP</i>	0.05173(0.01671)	0.32764(0.08317)	<b>0.03419(0.00262)</b>
4	<i>HV</i>	3.10153(0.02342)	2.85421(0.03439)	<b>3.17731(0.05114)</b>
	<i>IGD</i>	0.16652(0.00685)	0.24521(0.01194)	<b>0.15792(0.01958)</b>
	<i>SP</i>	0.04025(0.00955)	0.31903(0.07458)	<b>0.02594(0.00226)</b>
5	<i>HV</i>	2.96950(0.04691)	2.72226(0.05009)	<b>3.11278(0.03584)</b>
	<i>IGD</i>	0.22522(0.01547)	0.30830(0.01672)	<b>0.19155(0.01335)</b>
	<i>SP</i>	0.05318(0.01196)	0.34777(0.06148)	<b>0.03527(0.00359)</b>
6	<i>HV</i>	3.08356(0.04182)	2.83883(0.03859)	<b>3.21087(0.02122)</b>
	<i>IGD</i>	0.17503(0.01153)	0.25629(0.01275)	<b>0.14742(0.00719)</b>
	<i>SP</i>	0.04158(0.00542)	0.30272(0.08688)	<b>0.02858(0.00218)</b>
7	<i>HV</i>	2.97017(0.04580)	2.70561(0.04851)	<b>3.09921(0.05977)</b>
	<i>IGD</i>	0.22356(0.01541)	0.31223(0.01765)	<b>0.19792(0.02236)</b>
	<i>SP</i>	0.05207(0.00824)	0.34496(0.08266)	<b>0.03688(0.00430)</b>
8	<i>HV</i>	3.07434(0.03415)	2.84302(0.03142)	<b>3.18531(0.04201)</b>
	<i>IGD</i>	0.17324(0.01034)	0.25339(0.01085)	<b>0.15401(0.01758)</b>
	<i>SP</i>	0.04032(0.00354)	0.32526(0.07144)	<b>0.02787(0.00235)</b>

TABLE S-XII

STATISTIC RESULTS OF THE THREE COMPARED ALGORITHMS ON TEST PROBLEMS WITH 20 DECISION VARIABLES AND  $\tau=20$ 

Ins	Indicator	DC-NSGA-II	DC-NSGA-III	dCMOEA
1	<i>HV</i>	3.22532(0.02751)	3.03690(0.03052)	<b>3.30644(0.02241)</b>
	<i>IGD</i>	0.13188(0.00965)	0.19382(0.00858)	<b>0.11694(0.00862)</b>
	<i>SP</i>	0.02800(0.00198)	0.25148(0.06941)	<b>0.01900(0.00178)</b>
2	<i>HV</i>	3.28536(0.01835)	3.11828(0.02190)	<b>3.35219(0.01192)</b>
	<i>IGD</i>	0.10783(0.00487)	0.16145(0.00626)	<b>0.09639(0.00482)</b>
	<i>SP</i>	0.02382(0.00136)	0.22809(0.05975)	<b>0.01605(0.00095)</b>
3	<i>HV</i>	3.21682(0.03162)	3.03655(0.03823)	<b>3.28854(0.04665)</b>
	<i>IGD</i>	0.13517(0.00963)	0.19409(0.01195)	<b>0.12412(0.01932)</b>
	<i>SP</i>	0.02913(0.00392)	0.25096(0.07053)	<b>0.02032(0.00177)</b>
4	<i>HV</i>	3.26925(0.01741)	3.10962(0.02253)	<b>3.33032(0.02938)</b>
	<i>IGD</i>	0.10763(0.00532)	0.15952(0.00619)	<b>0.09913(0.01137)</b>
	<i>SP</i>	0.02284(0.00152)	0.23149(0.06193)	<b>0.01542(0.00128)</b>
5	<i>HV</i>	3.19646(0.04413)	3.00460(0.04191)	<b>3.31384(0.02694)</b>
	<i>IGD</i>	0.14596(0.01450)	0.20924(0.01355)	<b>0.12001(0.00973)</b>
	<i>SP</i>	0.03231(0.00418)	0.29157(0.08143)	<b>0.02151(0.00185)</b>
6	<i>HV</i>	3.25965(0.02539)	3.08380(0.03825)	<b>3.33587(0.02909)</b>
	<i>IGD</i>	0.11513(0.00689)	0.17088(0.01125)	<b>0.10210(0.01133)</b>
	<i>SP</i>	0.02714(0.00281)	0.23181(0.05326)	<b>0.01735(0.00112)</b>
7	<i>HV</i>	3.19837(0.04319)	3.00263(0.03698)	<b>3.31185(0.02598)</b>
	<i>IGD</i>	0.14354(0.01303)	0.20818(0.01281)	<b>0.12038(0.01002)</b>
	<i>SP</i>	0.03175(0.00236)	0.26979(0.06914)	<b>0.02202(0.00155)</b>
8	<i>HV</i>	3.24394(0.03493)	3.08310(0.03243)	<b>3.33412(0.01652)</b>
	<i>IGD</i>	0.11673(0.01006)	0.16823(0.01038)	<b>0.09845(0.00535)</b>
	<i>SP</i>	0.02530(0.00192)	0.25538(0.06303)	<b>0.01665(0.00111)</b>