Evolutionary Dynamic Constrained Multiobjective Optimization: Test Suite and Algorithm

Guoyu Chen, Yinan Guo, Yong Wang, Jing Liang, Dunwei Gong, Shengxiang Yang

Abstract—Dynamic constrained multiobjective optimization problems (DCMOPs) abound in real-world applications and gain increasing attention in the evolutionary computation community. To evaluate the capability of an algorithm in solving DCMOPs, artificial test problems play a fundamental role. Nevertheless, some characteristics of real-world scenarios are not fully considered in the previous test suites, such as time-varying size, location and shape of feasible regions, the controllable change severity, as well as small feasible regions. Therefore, we develop the generators of objective functions and constraints to facilitate the systematic design of DCMOPs, and then a novel test suite consisting of nine benchmarks, termed as DCP, is put forward. To solve these problems, a dynamic constrained multiobjective evolutionary algorithm with a two-stage diversity compensation strategy (TDCEA) is proposed. Some initial individuals are randomly generated to replace historical ones in the first stage, improving the global diversity. In the second stage, the increment between center points of Pareto sets in the past two environments is calculated and employed to adaptively disturb solutions, forming an initial population with good diversity for the new environment. Intensive experiments show that the proposed test problems enable a good understanding of strengths and weaknesses of algorithms, and TDCEA outperforms other state-of-the-art comparative ones, achieving promising performance in tackling DCMOPs.

Index Terms—Dynamic constrained multiobjective optimization, evolutionary algorithm, test suite, diversity.

This work was supported in part of by National Natural Science Foundation of China under Grant 61973305, 61573361, in part of National Key R&D Program of China under Grant 2022YFB703701, in part of the Foundation of Key Laboratory of System Control and Information Processing, Ministry of Education, P.R. China under Grant Scip202203, and the in part of 111 Project under Grant B21014. (Corresponding author: Yinan Guo)
Guoyu Chen is with School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China. (e-mail: chenguoyumail@163.com)
Yinan Guo is with School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China, also with Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai, 200240; School of Mechanical Electronic and Information Engineering, China University of Mining and Technology (Beijing), Beijing 100083, China. (e-mail: nanfly@126.com)
Yong Wang is with School of Automation, Central South University, Changsha 410083, China (e-mail: ywang@csu.edu.cn)
Jing Liang is with School of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China. (e-mail: liangjing@zzu.edu.cn)
Dunwei Gong is with School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao, 266601, China. (email: dwgong@vip.163.com)
Shengxiang Yang is with the Institute of Artificial Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, U.K. (e-mail: syang@dmu.ac.uk)

I. INTRODUCTION

In many practical optimization applications, such as scheduling optimization [1], resource allocation [2], and hardware design [3], there are more than one objective conflicting with each other and constraints [4]-[6]. More especially, objectives and constraints may change over time, forming dynamic constrained multiobjective optimization problems (DCMOPs) [7]-[10]. For example, in cognitive radio networks, increasing spectrum utilization decreases the number of idle channels, leading to less energy efficiency. Apparently, it is a challenge to simultaneously maximize energy efficiency and spectrum utilization, while satisfying the time-varying signal-to-interference-plus-noise ratio [11]. In the fluid catalytic cracking-distillation process [12], the operating variables are optimized to minimize the energy consumption and maximize the economic benefits of products. To avoid large production fluctuations and production accidents, the yield constraints of products are changed once the production scenarios switch. Although rich studies [13]-[16] have been done on dynamic multiobjective optimization problems (DMOPs), there are still some gaps in DCMOPs, especially the related benchmarks and algorithms [17].

Regarding the benchmarks of DCMOPs, a few functions are proposed and only simple characteristics are concerned, which show insufficiency to test the capability of problem-solvers. Azzouz et al. [18] and Chen et al. [12] focused on one type of DCMOPs, in which both objective functions and constraints change over time. However, based on the analysis in [19], DCMOPs can be formulated to another two types, that is, static objective functions with dynamic constraints or only objective functions change over time. In addition, according to the characteristics of dynamics in real-world applications, more scenarios need to be considered. One is that complex changes of feasible regions, e.g., time-varying shape, size and location, widely exist in real-world, producing various shapes of Pareto front (PF) under dynamic environments. Secondly, in the existing test suites, only the change frequency has been considered to measure the performance in timely tracking. In order to further evaluate the robustness of algorithms in responding to the environmental changes, controllable change severity is necessary. Thirdly, in many practical applications, feasible solutions may cover small regions in decision space. Under dynamic environments, searching the true Pareto front is a challenging task.

To address DCMOPs, a few of dynamic constrained
multiobjective evolutionary algorithms (DCMOEAs) have been developed to track the feasible Pareto optima changing over time [18]. Chen et al. [12] and Azzouz et al. [18] introduced some random solutions to improve the diversity of an initial population under the new environment. However, these methods simply reuse historical information, and lack consideration in efficiently utilizing infeasible solutions, leading to slower convergence speed. Being different from them, the feasibility driven strategy [20] tried to repair infeasible solutions by feasible ones. However, finding feasible solutions may be difficult and time-consuming, especially for DCMOPs with small feasible regions. To sum up, the previous studies do not efficiently mine the implicit knowledge of past Pareto optima, leading to the weak robustness in tracking the changing Pareto optima over a sequence of environments. Thus, how to design a promising DCMOEA is still an open challenge.

To address these issues, we design a test suite of DCMOPs, and develop a DCMOEA to solve them. Twofold contributions are summarized as follows:

1) A set of benchmark functions on DCMOPs, called DCP, is proposed based on the developed generators of objective functions and constraints. The typical characteristics of real-world DCMOPs are covered in DCP, e.g., complex changes of feasible regions, controllable change severity, and small feasible region.

2) A two-stage diversity compensation strategy is developed to generate an initial population once a new environment occurs. In the first stage, a set of random individuals are generated. The number of random individuals introduced in feasible set is determined by the ratio of feasible ones in current population. The rest ones randomly generated replace the feasible solutions. Following that, an increment term determined by the feasibility and the dynamics of objectives is added to each individual at the second stage. Being different from the existing methods that replace the solutions by random ones, our method makes full use of the information about feasibility and the characteristics of dynamic environment to enhance the diversity of population.

The rest of this article is organized as follows. Section II discusses the preliminary and related work. In section III, a novel test suite of DCMOPs is designed. Also, section IV proposes a DCMOEA. Section V and Section VI present the experimental settings and discussions, respectively. Finally, section VII concludes this article and outlines future works.

II. PRELIMINARY AND RELATED WORK

A. Dynamic Constrained Multiobjective Optimization

Without loss of generality, a standard DCMOP can be formulated as follows.

\[
\min F(x, t) = (f_1(x, t), \ldots, f_m(x, t))^T \]

\[\begin{align*}
\text{s.t.} & \quad h_i(x, t) = 0, k = 1, \ldots, a \\
& \quad g_i(x, t) \geq 0, k = a + 1, \ldots, a + b
\end{align*}\]  \hspace{1cm} (1)

where \(m\) is the number of objectives, \(x = (x_1, \ldots, x_n)^T\) denotes \(n\)-dimensional decision variables in decision space \(\Omega\), and \(h_i(x, t)\) and \(g_i(x, t)\) are \(k\)th equality or inequality constraint, respectively, that may vary with \(t\). Also, the definition of time index \(t\) is presented as follows:

\[
t = \frac{\tau}{\tau_i} n_i
\]  \hspace{1cm} (2)

where \(\tau\) is the generation counter, \(\tau_i\) and \(n_i\) denote the frequency and severity of change, respectively. Smaller \(\tau_i\) means that the environment changes quickly, while smaller \(n_i\) implies that environment severely changes over time.

**Definition 1 Dynamic feasible region**: The feasible region at time \(t\) is defined as follows:

\[
O(t) = \{x \in \Omega | CV(x, t) = 0\}
\]  \hspace{1cm} (3)

\[
CV(x, t) = \sum_{i=1}^{a+b} CV_i(x, t)
\]  \hspace{1cm} (4)

\[
CV_i(x, t) = \begin{cases} 
\max(0, h_i(x, t)), & k = 1, \ldots, a \\
\max(0, [g_i(x, t) - \eta]), & k = a + 1, \ldots, a + b
\end{cases}
\]  \hspace{1cm} (5)

where \(CV\) is the degree of constraint violation, and \(\eta\) is a very small positive value, e.g., \(\eta = 1e-4\).

**Definition 2 Dynamic Pareto dominance**: At time \(t\), a solution \(x\) is said to Pareto dominate another one \(y\) in decision space, where \(x \prec y\), if and only if:

\[
\forall i = [1, \ldots, m], \quad f_i(x, t) \leq f_i(y, t) \\
\exists j = [1, \ldots, m], \quad f_j(x, t) < f_j(y, t)
\]  \hspace{1cm} (6)

**Definition 3 Dynamic Pareto-optimal set**: Denoted \(PS(t)\) as a Pareto-optimal set at time \(t\), all solutions in it are not dominated by any other solutions in feasible region \(O(t)\), that is

\[
PS(t) = \{x | \sim y \in O(t) : y \prec x\}
\]  \hspace{1cm} (7)

**Definition 4 Dynamic Pareto-optimal front**: At time \(t\), a dynamic Pareto-optimal front, represented by \(PF(t)\), is the objective vector of \(PS(t)\) [19]. That is

\[
PF(t) = \{F(x, t) | x \in PS(t)\}
\]  \hspace{1cm} (8)

DCMOPs can be categorized into three types in terms of the dynamics of objective functions and constraints [19].

1. **Type I**: Objective functions change over time and constraints are fixed.
2. **Type II**: Objective functions are fixed while constraints change over time.
3. **Type III**: Both objective functions and constraints change over time.

B. Review of Existing Test DCMOPs

- **Suitable** or **inappropriate**
- **Suitable** or **inappropriate**
- **Suitable** or **inappropriate**
feasible regions is neglected. Additionally, Zeng et al. [22] extended constrained optimization problems to DCMOPs by considering their change over time in constraints. Azzouz et al. [18] designed eight test functions with dynamic objective functions and constraints by considering different characteristics of constraints [14]. However, the change of feasible region is rather simple under dynamic environments, and the change severity between two adjacent environments is small and uncontrollable. Recently, Chen et al. [12] constructed a benchmark suite by combining four types of dynamic constraints with two oscillatory modes of objective functions. Due to its flexible configuration, more dynamic scenarios are formed, e.g., the true PF varies from disconnected distribution to continuous one. But only both time-varying objective functions and constraints are considered, which corresponds to Type III DCMOPs.

To sum up, the previous test suites lack the comprehensive consideration of following characteristics: 1) The feature of feasible regions, such as shape, size and location, may be changed under dynamic environments in a complex mode. 2) A test suite shall fully cover three types of DCMOPs due to non-singleness of real-world applications. 3) The change severity and frequency shall be controllable so as to effectively evaluate the robustness of algorithms in responding to environmental changes. 4) Small feasible region is widely spread in many practical applications, producing the difficulty in searching the feasible Pareto optima that varies over time. The above-mentioned characteristics enrich the diversity of a test suite, forming the challenge for adapting dynamic constrained multiobjective optimization. Therefore, it is meaningful to design a new set of test functions.

C. Review of Existing DCMOEs and Related Methods

In [18], two versions of D-NSGA-II, called DC-NSGA-II-A, and DC-NSGA-II-B, were developed for solving DCMOPs. Especially, a self-adaptive penalty function was designed and integrated with NSGA-II, with the purpose of searching the Pareto optima under each environment. Following that, Azzouz et al. [20] proposed a feasibility-driven strategy that repairs infeasible individuals toward the new feasible regions once an environmental change appears. Chen et al. [12] reused a portion of historical Pareto-optimal solutions and combined them with ones randomly generated, forming a diverse initial population under a new environment. Recently, a multi-population evolution based dynamic constrained multiobjective optimization algorithm (nEDCMOA) [23] was proposed, in which a tribe classification operator was designed to divide the current solutions into different tribes in terms of their feasibility and objective values. Once the environment changed, the distances to approach the new Pareto front were estimated for tribes, and then an initial population was produced. The above-mentioned works have conducted preliminary exploration on solving DCMOPs, but neglected to further mine the implicit information of historical Pareto optima and considered fewer characteristics of constraints, providing limited ability for DCMOEs.

To the best of our knowledge, DCMOPs arise from the intersection of DMOPs and constrained multiobjective optimization problems (CMOPs) [20]. The change response strategies for solving DMOPs and the constraint-handling techniques for CMOPs, as their commonly-used problem-solvers, can provide the valuable and potential approaches for solving DCMOPs. Three mainstream change response strategies are summarized as follows.

1) Diversity instruction methods respond to environmental changes by introducing random individuals or mutated ones, with the purpose of enhancing the diversity of population [15]. Based on the above idea, Deb et al. [15] designed two kinds of diversity introduction strategies. Ma et al. [24] constructed a multi-regional diversity maintenance mechanism, in which the random individuals were generated within subregions, with the purpose of providing convergence pressure for the current population to explore new regions. However, the other valuable information, e.g., the sparse degree of distribution, may be lost after diversity introduction [16][25].

2) Memory mechanism generally preserves the Pareto optima of historical environments, and reuses them once a similar environment appears [26]. Sahmoud and Topcuoglu [27] constructed an external storage to store all non-dominated solutions, and reused them for the similar future environment. Azzouz et al. [28] adaptively adjusted the number of memorized solutions in terms of the change severity. But extracting useful knowledge from external storage ineluctably increases the computational cost [29].

3) Prediction techniques produce an initial population by learning the changing rules from historical Pareto optima [30][31], with the purpose of speeding up the convergence under the new environment. Rich methods, e.g., autoregression model [30], Kalman filter [31] and transfer learning [32], have been introduced to develop prediction strategies. In order to make full use of various prediction models, Rambabu et al. [33] presented a mixture-of-experts-based ensemble framework to switch various predictors by a gating network. However, the historical optimum with worse convergence may mislead the search [33][34].

To solve CMOPs, four categories of constraint-handling techniques have been designed and incorporated into multiobjective evolutionary algorithms as follows.

1) Penalty function-based techniques construct a fitness function by adding a penalty term proportional to the constraint violation into the objective function [35]. Woldesenbet et al. [36] designed modified objective functions in terms of the original ones and constraint violation values. Ma and Wang [37] adaptively shifted the infeasible ones according to their neighboring feasible solutions, and then penalized them by constraint violations. Although this mechanism has been widely used in CMOPs, whereas the performance of algorithms is highly sensitive to the penalty factor [37].

2) Separation techniques evaluate individuals by their degree of constraint violation and objective values [38]. Feasibility rule [39], $e$-constraint [40], and stochastic ranking [41], as typical ones, all switch the focus between constraints and objective functions, with the purpose of improving the
exploration ability of algorithms [42]. Among them, feasibility rule focuses on the feasible solutions. The latter two pay more attention to the infeasible solutions, but have the difficulty in setting suitable parameters.

3) In multiobjective optimization-based techniques, constraints are regarded as optimization objectives, thus showing a significant advantage in tackling the problem with small feasible regions [43]. Ray et al. [44] treated a constraint violation as a new objective. Zhou et al. [45] constructed a tri-goal evolution framework, in which two indicators were designed for convergence and diversity, whereas the third one focused on feasibility. Apparently, the number of objectives becomes larger in these techniques, bringing higher selection pressure in the evolution [46].

4) Hybrid techniques generally integrate more than one constraint-handling strategy [47][48], with the purpose of fully utilizing their advantages to achieve better performance. Mallipeddi and Suganthan [49] developed an ensemble strategy consisting of four constraint handling methods, and each one optimized its own population. How to reasonably construct the combination is still an open challenge [50].

III. THE DESIGN OF TEST SUITE FOR DCMOPs

A. Characteristics of DCMOPs

In DCMOPs, the true PFs changing over time may contain various feasible parts restricted by constraints under dynamic environments, which emerge special characteristics that never exist in the unconstrained DMOPs and CMOPs [19]. In the following analysis, we define the original PF as the one produced by objective functions without constraints, and true PF as the one considering constraints, with the purpose of avoiding confusion.

First, constraints varying over time may form infeasible regions with dynamically-changed shape, size, and location. Fig. 1 depicts the dynamics of true PFs caused by the changes in different characteristics of dynamic infeasible regions. Intuitively, the shape of infeasible region varying over time may result in disconnection of PFs. Being different from it, the whole historical PF may become infeasible with the extension of infeasible region, and the true PF in the new environment may shift to the boundary of the current feasible region, as shown in Fig.1 (c). Compared with the above two cases, the relocation of infeasible region may have no influence on PF. To sum up, time-varying constraints produce diverse and complex dynamics on the historical PF.

Second, disconnected feasible regions, as the widely-spread type in real-world constrained optimization problems, may form the true PF with multiple segments. Under dynamic environments, the distribution, location and number of segments may change over time. That means the historical PSs may become infeasible in the current environment. Therefore, it is a significant issue to fully mine the correlation among historical segments, and utilize it to estimate the distribution of future PF.

Third, small feasible region, especially with time-varying size and location, is difficult detected under limited computational resource. With the increasing number and size of feasible regions, it is more possible to produce feasible initial solutions, benefiting to track the new Pareto optima, and vice versa. Therefore, the population is hungry for better diversity, with the purpose of quickly getting away from infeasible regions.

B. Basic Framework

1) Generator of objective functions: Recalling existing test suites of multiobjective optimization [14][51], the true PFs exhibit rich characteristics, such as linear, nonlinear, convex-concave, translation, and so on. Previous studies have pointed out that these characteristics can be produced by introducing trigonometric functions with rational parameters [12][52]. Based on this, a generator of objective functions is designed.

\[ f_1 = g(x_{II}, t)(h(x_I) + \alpha \sin(\beta \pi \phi)^\gamma), \]
\[ f_2 = g(x_{II}, t)(1 - h(x_I) + \alpha \sin(\beta \pi \phi)^\gamma) \]  \hspace{1cm} (9)

where \( h(x_I) \) is a non-negative function satisfying \( 0 \leq h(x_I) \leq 1 \). \( g(x_{II}, t) \) is also a non-negative function that determines the approximating degree of the population to the true PF. \( x_I \) and \( x_{II} \) are subvectors of the decision vector \( x \). \( \alpha \), \( \beta \), \( \phi \), and \( \gamma \) are the key parameters, control the local shape and location of the original PF. Without loss of generality, the minimum value of \( g(x_{II}, t) \) is one, thus the objective functions can be simplified as follows:

\[ f_1 + f_2 = 1 + 2\alpha \sin(\beta \pi \phi)^\gamma \] \hspace{1cm} (10)

Referring to Eq. (10), the PFs of unconstrained DMOPs with different settings of \( \alpha \), \( \beta \), \( \phi \), and \( \gamma \) are depicted in Fig. 2. Based on them, the final PFs restricted by constraints are generated. Apparently, various characteristics, e.g., disconnected, continuity-disconnect, concavity-convexity, are emerged in accordance with real-world applications.

2) Generator of constraints: In real-world constrained optimization problems, constraints are generally formulated by varied, even complex forms [53][54]. In order to produce the feasible region with diversified dynamic characteristics, such as translation, disconnection, small size and nonlinearity, an
improved generator of constraints is developed based on [55].

\[ l(F) - (a - b \sin(c))x^y \geq 0 \]  \hspace{1cm} (11)

where \( l(F) \) is a function related to the objective vector, determining the global control. \( (a - b \sin(c))x^y \) is the component of local adjustment. Based on Eq. (11), the global feasible regions in objective space are formed by \( l(F) \), and their local characteristics are adjusted by the latter term. Based on this, different geometries of true PFs can be generated.

In the component of global control, any commonly-used functions can be introduced to construct the global feasible region [14]. Four examples are presented as follows.

\[
\begin{align*}
1 \text{(a)}: & \quad f_1 + f_2 = 1 \\
2 \text{(b)}: & \quad f_1^2 + f_2^2 = 1 \\
3 \text{(c)}: & \quad \sqrt{f_1} + \sqrt{f_2} = 1 \\
4 \text{(d)}: & \quad 0.5f_1^2 + f_2 = 1
\end{align*}
\]  \hspace{1cm} (12)

Fig. 3 (a) depicts the boundaries of corresponding global feasible regions in objective space, and the infeasible region that is bounded by \( l(F) \), is shown in Fig. 3 (b) with gray.

To enhance the nonlinearity of infeasible regions, the component of local adjustment introduces a trigonometric function to adjust the area bounded by \( l(F) \). By tuning five key parameters, i.e., \( a, b, c, d, \) and \( e \), diverse geometries of feasible parts are formed. Taking three constraint functions listed in Eq. (13) as examples, Fig. 3 (c) depicts the boundaries of corresponding feasible regions, and Fig. 3 (d) shows the infeasible region bounded by \( Y_2 \) with gray.

\[
\begin{align*}
Y_1: & \quad f_1 + f_2 - 1.4 + 0.2 \sin(2\sqrt{2}\pi(f_2 - f_1))^2 = 0 \\
Y_2: & \quad f_1^2 + f_2^2 - (0.8 - 0.2 \sin(6\arctan(f_2/f_1)))^2 = 0 \\
Y_3: & \quad 1.2f_1 + f_2 - 0.6 - 0.08 \sin(2\pi(f_2 - f_1)) = 0 \\
Y_4: & \quad 1.2f_1 + f_2 - 1.3 \leq 0
\end{align*}
\]  \hspace{1cm} (13)

Additionally, it is worth mentioning that \( l(F) \) and all parameters can be set to time-related functions, which construct constraints changing over time.

C. Test Instances

Based on the generators of objective functions and constraints developed in the article, we design a novel test suite for DCMOPs, called DCP. It consists of nine distinct test functions that have various types and characteristics as mentioned in Section II-B, with the purpose of comprehensively measuring the algorithm performance in tracking the feasible Pareto optima changing over time.

DCP1:

\[
\begin{align*}
\min & \quad \begin{cases}
    f_1 = g_1x_1 + G \\
    f_2 = g(1 - x_1) + G
\end{cases} \\
\text{s.t.} & \quad c_1 = \cos(-0.15\pi)f_2 - \sin(-0.15\pi)f_1 \geq 0 \\
& \quad (2\sin(5\pi(\sin(-0.15\pi)f_2 + \cos(-0.15\pi)f_1)))^6
\end{align*}
\]  \hspace{1cm} (14)

with

\[ g = 1 + \sum_{i=2}^{n}(x_i - G)^2 \]

where \( G = |\sin(0.5\pi)| \), and the search space is \([0,1]^n\).

Intuitively, the objective functions of DCP1 change over time. Under the fixed constraint, the original PF is partitioned into multiple feasible segments. Fig. 4 depicts PFs and the corresponding feasible regions under dynamic environments. A similar case for DCP1 is nurse scheduling [56]. Under disconnected feasible regions, scheduling cost and the deviation of a shift between required and really allotted nurses should be minimized simultaneously.

DCP2:

\[
\begin{align*}
\min & \quad \begin{cases}
    f_1 = g(x_1 + 0.25G\sin(\pi x_1)) \\
    f_2 = g(1 - x_1 + 0.25G\sin(\pi x_1))
\end{cases} \\
\text{s.t.} & \quad c_1 = (4f_1 + f_2 - 1)(0.3f_1 + f_2 - 0.3) \geq 0 \\
& \quad c_2 = (1.85 - f_1 + f_2 - 0.3\sin(3\pi(f_2 - f_1)))^2 \\
& \quad (f_1 + f_2 - 1.3) \leq 0
\end{align*}
\]  \hspace{1cm} (15)
with
\[ g = 1 + \sum_{i=2}^{n} ((x_i - G)^2 + \sin(0.5\pi(x_i - G))^2) \]
where \( G = \sin(0.5\pi t) \), and the search space is \([0,1] \times [-1,1]^{n-1}\).

Similar to DCP1, DCP2 contains dynamic objective functions and fixed constraints, actually belonging to type I of DCMOOPs. Under constraint \( c_1 \), the true PFs in different environments may be the whole or partial original one, as shown in Fig. 5 (b). Being different from \( c_1 \), the infeasible region produced by \( c_2 \) may prevent a population from converging to the true PFs. Therefore, DCMOEAs need to well balance the feasibility and convergence. In practice, electrical power dispatch can be formulated to the similar form [57], electric generators are required to have a fast supply speed satisfying the power demand at different times.

**DCP4:**
\[
\begin{align*}
\min \quad & f_i = g(x_i + 0.25\sin(\pi x_i)) \\
\text{s.t.} \quad & c_1 = (f_i - 1)^2 + (f_i - 0.2)^2 - 0.3^2 \\
& (f_i - 0.2)^2 + (f_i - 1)^2 - 0.3^2 \geq 0 \\
& c_2 = (f_i^2 + f_j^2 - 4^2) \leq 0 \\
& c_3 = (f_i^2 + f_j^2 - (3.1 + 0.2\sin(4\arctan(f_i/f_j))^2)^2) \\
& (f_i^2 + f_j^2 - 2.3^2) \geq 0
\end{align*}
\]
with
\[ g = 1 + \sum_{i=2}^{n} |x_i - G| \]
where \( G = \sin(0.5\pi t) \), and the search space is \([0,1]^n\).

In DCP3, the objective functions change over time, but the constraints remain static, which is similar to the case with evolvable hardware design [3]. The dynamism of this problem lies in PFs and PSs both shifting over time, as shown in Fig. 6. Additionally, small feasible regions make the algorithms difficultly track the changing PSs.

**DCP5:**
\[
\begin{align*}
\min \quad & f_1 = g(x_1) \\
\text{s.t.} \quad & c_1 = ((0.2 + G)f_1^2 + f_2 - 2)(0.7f_1^2 + f_2 - 2.5) \geq 0 \\
& c_2 = f_1^2 + f_2^2 - (0.6 + G)^2 \geq 0
\end{align*}
\]
with
\[ g = 1 + \sum_{i=2}^{n} |x_i - 0.5 \sin(2\pi x_i)| \]
where \( G = 0.5 |\sin(0.5 \pi t)| \) and the search space is \([0, 1] \times [-1, 1]^{n-1}\).

DCP5 also has fixed objective functions and dynamic constraints. Being different from DCP4, the feasible regions depicted in Fig. 8 are more regular. Additionally, convergence pressure may switch from one objective to another due to the adverse effect of the constraint biasing from \( t = 0 \) to \( t = 0.6 \), as shown in Fig. 8 (a). In real-world applications, dynamic scheduling optimization problems, such as ship scheduling problems [58], may have similar characteristics.

**DCP6**:  
\[
\begin{align*}
\min & \quad f_1 = g x_i \\
\text{s.t.} & \quad f_2 = g (1 - x_i) \\
\end{align*}
\]
with
\[
g = 1 + 6 \sum_{i=2}^{n} x_i^2
\]
where \( G = |\sin(0.5 \pi t)| \) and the search space is \([0, 1] \times [-1, 1]^{n-1}\).

DCP6 is a type II problem, which only takes the dynamic constraints into account. As shown in Fig. 9 (a), small feasible regions may become narrow over time, producing the progressive difficulty in converging to the true PFs. A case similar to DCP6 can be the ship weather routing optimization [59], in which the suitable route is difficult to determine under the complex and time-varying weather.

**DCP7**:  
\[
\begin{align*}
\min & \quad f_1 = g (x_i + 0.02 \sin((10 - \|10G\|) \pi x_i)) \\
& \quad f_2 = g (1 - x_i + 0.02 \sin((10 - \|10G\|) \pi x_i)) \\
\text{s.t.} & \quad c_1 = f_1 + f_2 - G - \sin(5\pi (f_1 - f_2 + 1))^2 \geq 0
\end{align*}
\]
with
\[
g = 1 + \sum_{i=2}^{n} (x_i - G)^2
\]
where \( G = \sin(0.5 \pi t) \) and the search space is \([0, 1] \times [-1, 1]^{n-1}\).

Different from the above benchmarks, DCP7 is a Type III one. Time-varying objective functions produce the oscillating original PFs. Under the dynamic constraint, they are partitioned into multiple feasible points or segments, as shown in Fig. 10 (b). A similar real-world application is the power supply optimization in magnesia grain manufacturing [60], the objective functions oscillate among several optimization modes under dynamic constraints.

**DCP8**:  
\[
\begin{align*}
\min & \quad f_1 = g x_i \\
\text{s.t.} & \quad f_2 = g (1 - x_i) \\
& \quad c_1 = (\sqrt{f_1} + \sqrt{f_2} - 0.95 - 0.5 |G|) \\
& \quad c_2 = (0.8 f_1 + f_2 - (2.5 + 0.08 \sin(2\pi (f_2 - f_1)))) \\
& \quad c_3 = ((0.93 + |G|/3) f_1 + f_2 - (2.7 + 0.5 |G| + 0.08 \sin(2\pi (f_2 - f_1)))) \geq 0
\end{align*}
\]
with
\[
g = 1 + \sum_{i=2}^{n} |G(x_i - G)| - \cos(\pi(x_i - G) + 1)^2
\]
where \( G = \sin(0.5 \pi t) \) and the search space is \([0, 1] \times [-1, 1]^{n-1}\).

DCP8 also has both dynamic objective functions and constraints illustrated in Fig. 11. The feasible region changing from \( t = 0 \) to \( t = 0.6 \) may increase the convergence pressure near the boundary of objective space, which is close to real-world problems like the wastewater treatment process [61].
More especially, the infeasible regions cover the whole original PF at \( t = 0 \), thus the true one is the boundary of feasible region.

Fig. 11. The visualization of DCP8.

DCP9:

\[
\min \begin{cases}
    f_1 = g x_r, \\
    f_2 = g (1-x_r)
\end{cases}
\]

s.t. \( c_1 = (f_1^2 + f_2^2 - 4)^2 (f_1^2 + f_2^2 - (0.2 + G)^2) \leq 0 \)

\( c_2 = (2.1 - (f_1/(1 + 0.15 \cos(6\arctan(f_2/f_1)^3^{101}))^2)^2 \)

\( -(f_2/(1 + 0.75 \cos(6\arctan(f_2/f_1)^3^{101})))^2 \)

\( f_1^2 + f_2^2 - 1.6^2 \leq 0 \)

with

\[
g = 1 + 10 \sum_{i=1}^{n} (x_i - G)^2
\]

where \( G = \left| \sin(0.5 \pi t) \right|, \quad h = 0.75 + 1.25 G, \quad r = 1 + \left\lceil (n-1) G \right\rceil\) and the search space is \([0,1]^n\).

DCP9 is a more challenging task. Dynamic constraints produce more than one narrow infeasible regions, which partition objective space into multiple feasible subspaces, easily inducing the evolution into local optimum. However, the infeasible region has no influence on the original PF at \( t = 0 \) in Fig. 12, which is similar to fluid catalytic cracking-distillation process [12].

Fig. 12. The visualization of DCP9.

IV. THE PROPOSED TDCEA

A. The Framework of TDCEA

Tracking the feasible PS under the new environments as soon as possible is the main task for solving DCMOPs. To achieve this goal, a dynamic constrained multiobjective evolutionary algorithm with two-stage diversity compensation strategy is proposed, termed as TDCEA. To the best of our knowledge, historical Pareto-optima as the source information of change response strategies and various static optimizers both have a direct influence on algorithm performance under a new environment. As the framework given in Algorithm 1, CCMO [62] is conducted as a static optimizer to find the feasible Pareto-optimal solutions under each environment. It is worth noting that CCMO evolves the main population to solve original CMOP while a helper problem derived from the original one is solved by assisted population, and a weak cooperation paradigm that only shares the offspring generated by two populations is designed for co-evolution. Based on the population evolved by CCMO, several solutions are randomly selected to monitor the environmental change. Once \( flag = true \), an environmental change appears, and the two-stage diversity compensation strategy is triggered to generate an initial population under the new environment.

B. Change Detection

In DCMOPs, environmental changes may arise from either the objective functions or constraints [12]. To improve the computational efficiency of change detection, a two-stage change detection strategy is developed. As shown in Algorithm 2, 10% \( N \) solutions are randomly selected from the current population, constructing the detection set \( DS \). Following that, the difference of objective values, represented by \( DO \), is evaluated as follows:

\[
DO = \sum_{j=1}^{\lfloor DS \rfloor} \sum_{i=1}^{n} \left| f_j(x_i) - f_j'(x_i) \right|
\]
denoted as $\delta$, is further re-evaluated as follows:

$$DC = \sum_{j=1}^{[DS]} \sum_{j=1}^{k} [c_j(x_i) - c'_j(x_i)]$$ (24)

where $c_j(x_i)$ and $c'_j(x_i)$ denotes the $j$th current and re-evaluated constraint value of $i$th individual in $DS$, respectively. Once $DC > \delta$ (e.g., $\delta = 10^{-4}$), a new environmental change occurs. Changing the feasible regions, flag is set to true, and $\lambda = 0$. Apparently, an environmental change is assumed to be successfully detected once flag = true.

C. Two-stage Diversity Compensation

To the best of our knowledge, the rational number of random individuals provides the appropriate diversity of the population and convergence pressure for tracking the new PS timely [11],[52]. In DCMOPIs, the whole or partial Pareto-optimal solutions obtained in last environment may become infeasible at current time. In this case, the diversity of a population needs to be improved, with the purpose of guiding the evolution to escape from infeasible regions. Therefore, a two-stage diversity compensation strategy is developed. As its pseudocode shown in Algorithm 3, the first stage introduces random solutions to enhance the global diversity of a population, achieving better exploration ability. Following that, different increment terms are added to the solutions in terms of their feasibility, with the purpose of well exploiting the search space nearby them.

Stage I: Under the new environment, all Pareto optima obtained in the last environment are re-evaluated, and then divided into feasible set $FS$, and infeasible set $IS$, respectively. According to their sizes, we define the feasibility ratio as $\rho = |FS|/N$. Following that, $\rho N$ individuals are randomly generated. Among them, $\rho N$ ones randomly replace solutions in $FS$, and the others for $IS$. Denoted $FS'$ and $IS'$ as the replaced solutions in $FS$ and $IS$, respectively,

$$\begin{align*}
FS &= FS \setminus FS' \cup \sum_{j=1}^{\rho N} x'_j \\
IS &= IS \setminus IS' \cup \sum_{j=1}^{(1-\rho)N} x'_j
\end{align*}$$ (25)

where $x'$ is the random solution.

Stage II: Denoted $\mu_{t-1}$ and $\mu_{t-2}$ are the center points of $PS_{t-1}$ and $PS_{t-2}$, respectively, $\Delta = \mu_{t-1} - \mu_{t-2}$ is the corresponding increment. Based on this, a $n$-dimensional random vector obeying the uniform distribution $U(0,\Delta)$ in $[0,\Delta]$ is generated. According to the feasibility ratio, different perturbation terms are added to feasible or infeasible individuals.

$$x = \begin{cases} 
    x + \lambda \rho U(0,\Delta) & \forall x \in FS \\
    x - (1-\rho) U(0,\Delta) & \forall x \in IS
\end{cases}$$ (26)

By adding the perturbation term, feasible solutions can achieve better performance to track the new optima as soon as possible. However, the current feasible solutions may be potential Pareto optima at the new time for type II DCMOPIs. Inappropriate perturbation may cause the loss of valuable information. In this case, $\lambda = 0$, and no perturbation is added to any solutions in $FS$. In addition, adding a perturbation to infeasible solutions can better search their nearby space, helping them jumping into feasible regions quickly. No matter which kind of solutions, a larger perturbation is encouraged with the increasing size of $FS$ or $IS$. Finally, the updated $FS$ and $IS$ are merged to construct an initial population under the new environment.

V. EXPERIMENTAL SETTINGS

A. COMPARATIVE ALGORITHMS

Six state-of-the-art DCMOEAs are employed for comparisons, including DC-NSGA-II-A [18], DC-NSGA-II-B [18], DC-MOEA [20], dCMOEA [12], memory-driven manifold transfer learning strategy (MMLT) [32] with feasibility rule, mEDCMOA [23]. Specially, three widely-used change response strategies derived from D-NSGA-II [15], PPS [30], MNSGA-II [27] and random initialization are compared with the proposed two-stage diversity compensation by integrating them to CCMO. Moreover, the algorithm performance of the proposed change response strategy with three constraint handling mechanisms, i.e., feasibility rule [39], stochastic ranking [41] and shift-based penalty function [37], are further investigated. For DC-NSGA-II-A, DC-MOEA, and D-NSGA-II, 20% of non-dominated solutions are replaced by random individuals as a new environment appears. Different from them, 20% of non-dominated solutions after mutation substitute for the original ones in DC-NSGA-II-B. In PPS, the order of AR model is set to 3, and the length of history mean point series is set to 23. In MNSGA-II, the memory size is set to 200, and the update frequency is 10.

B. PARAMETER SETTINGS

1) Population size: For all test instances, $N = 100$.
2) Number of decision variables: The dimension of decision space is set to 10 in all test problems.
3) Dynamic parameters and the number of runs: Each algorithm performs 20 independent runs on each test instance, and each run consists of 60 environmental changes. It is worth noting that no change takes place in the first 80 generations, so
as to minimize the effect of static optimization [23][25]. Following that, the frequency of change is set to \( \tau_i = 10, 20, 30, \) and \( n_t = 5, 10, 20 \) for the severity of change.

### C. Performance Indicators

In order to evaluate the algorithm performance, mean inverted generational distance (MIGD) [66], mean Hypervolume (MHV) [67], robustness (R) [52], and relative error distance (RED) [68] are employed as metrics.

1) MIGD can comprehensively reflect the distribution and convergence of Pareto-optimal solutions obtained under the total environments. Assuming that a set of uniformly distributed samples of true PF\((t)\) are sampled and represented by \( S(t) \), the MIGD value is calculated as follows:

\[
\text{MIGD} = \frac{1}{T} \sum_{t=1}^{T} \text{IGD}(t) \tag{27}
\]

\[
\text{IGD}(t) = \frac{1}{|S(t)|} \sum_{i=1}^{|S(t)|} D(\text{PF}(t), \tau_i) \tag{28}
\]

where \( T \) is the number of environments in a run. \( D(\text{PF}(t), \tau_i) \) denotes the minimum Euclidean distance in objective space between \( i \)th sampling point in \( S(t) \) and the feasible \( \text{PF}(t) \) obtained by an algorithm. A smaller MIGD value indicates better algorithm performance.

2) MHV is a representative metric that measures the comprehensive performance of algorithm.

\[
\text{MHV} = \frac{1}{T} \sum_{t=1}^{T} \bigcup_{i=1}^{\text{|PF(t)|}} \{ \text{Vol}_i \} \tag{29}
\]

where \( \text{Vol}_i \) denotes the volume of a hypercube formed by reference point \( z_{\text{ref}} \) and \( i \)th feasible solution in \( \text{PF}(t) \).

3) \( R \) is a metric that expresses the stability of algorithm performance over a sequence of dynamic environments, and formulated as follows:

\[
R(\text{IGD}) = \frac{1}{T-1} \sum_{t=1}^{T} (\text{IGD}(t) - \text{MIGD})^2 \tag{30}
\]

Apparently, a smaller value of \( R(\text{IGD}) \) indicates a better robustness on \( \text{IGD} \) metric, that is, the algorithm can achieve robust performance in tracking the changing PS.

4) Relative error distance (RED) measures how close to the optima an algorithm performs. Denoted \( \text{HV}_i \) and \( \text{HV}_i^{\text{ref}} \) as HV value of the obtained PF and true one under \( i \)th environment, respectively, RED is defined as follows.

\[
\text{RED} = \frac{\sum_{t=1}^{T} \left( 1 - \frac{\text{HV}_i(t)}{\text{HV}_i^{\text{ref}}} \right)}{T} \tag{31}
\]

Intuitively, smaller RED means better algorithm performance.

For all test instances, the best ones among the comparative algorithms are highlighted. Also, Wilcoxon rank-sum test [69][70] is employed to point out the significance between different results at the 0.05 significance level, where ‘+’, ‘-‘, and ‘=’ indicate that the results obtained by another algorithm are significantly better, significantly worse, and statistically similar to that obtained by compared algorithm, respectively.

### VI. DISCUSSIONS

To verify the rationality of DCP and TDCEA proposed in the article, experimental studies consist of five components, including effect on the number of decision variables, sensitivity analysis of parameter, performance comparisons with existing DCMOEAs, traditional change response strategies, and constraint handling mechanism, respectively.

#### A. Effect on the Number of Decision Variables

In a multiobjective optimization problem, the number of decision variables has a significant effect on the difficulty of solving it. In order to quantitatively investigate it, TDCEA is tested under different number of decision variables as \( n \) varies from 5 to 20 every 5. Here, \( \tau_i = 10, 20 \) and \( n_t = 5, 10, 20 \) are used for experiment. Fig. 13 depicts the MIGD values obtained by TDCEA under different number of decision variables. Intuitively, with the increase of \( n \), MIGD becomes larger, indicating that the problem is more difficult to be solved. Compared with \( n = 5 \), MIGD values show more significant difference as \( n = 10, 15, 20 \), thus having better evaluation ability. In order to keep a tradeoff between identifiability of algorithm performance and difficulty of problems, \( n = 10 \) is a good choice in following experiments.

#### B. Sensitivity Analysis of Parameter

In TDCEA, \( \omega \) determines how many initial individuals are randomly generated, which plays a direct influence on global diversity. As \( \omega \) varies from 0 to 0.5 every 0.1, TABLE 1 summarizes significance test of MIGD on DCP, while the corresponding statistical results are recorded in TABLE S1 of Supplementary. We observed from statistical results that appropriate global diversity can assist population to obtain good results under dynamic objectives, especially on the large severity of change. It is worth noting that only Stage II of two-stage diversity compensation strategy is conducted as \( \omega = 0 \). Although no random solutions are introduced, perturbation adding to the population can guarantee good performance in solving DCMOPs with type II, i.e., DCP4-DCP6. Apparently, it is beneficial to fully utilize historical Pareto optima to maintain the diversity of an initial population. In order to guarantee the robustness of algorithm performance on different types of DCMOPs, it is a good choice for \( \omega = 0.1 \).

In addition, the accuracy rate of change detection is also investigated under various \( \delta \), and \( n_t = 1, 5, 10, 20, 40, 80, 100 \) every 100 environmental changes. Fig. S1 of Supplementary depicts their experimental results, and indicates that the detection accuracy can be guaranteed as \( \delta \) is less than or equal to 1e-4. Thus, we set the threshold to 1e-4.

#### C. Comparisons with the State-of-the-art DCMOEAs

In this subsection, TDCEA is compared with four state-of-the-art DCMOEAs on DCP. TABLE 2 listed the significance test of MIGD on DCP, while the detailed statistical results of MIGD, MHV, R and RED are recorded in TABLE S2-S5 of
Unlike the other comparative algorithms, DC outperforms the above three methods in terms of four indicators. We observe from the experimental results that TDCEA proposed in this article outperforms the other competitors on MIGD, MHV, R and RED for about 56, 51, 63 and 51 out of 81, respectively.

DC-NSGA-II-A and DC-NSGA-II-B, as two variants of NSGA-II, improve the diversity of population by introducing random and mutated solutions, respectively. The replacement strategies show promising performance in solving DMOPs, but lack consideration in the feasibility of non-dominated solutions, resulting in low-efficiency initialization as a new environment appears. Similar to them, $\text{DCMOEA}$ combines a set of historical feasible solutions with random ones to form the initial population. The individuals randomly produced provide the limited information for tracking the new PSs, thus the above three methods achieve significantly worse performance than TDCEA on DCP in terms of four indicators.

Unlike the other comparative algorithms, DC-MOEA repairs infeasible individuals in an initial population by feasible ones. However, finding feasible solutions as the guide may be time-consuming, especially for DCMOPs with small feasible regions, e.g., DCP3, DCP6, and DCP9. Also, not all feasible solutions are superior enough to provide the promising guide for convergence on problems with complex constraints. Thus, how to obtain and select rational feasible individuals is still an open challenge.

$\text{MMTL}$ and $\text{mEDCMOA}$ are more competitive in solving DCP. The former introduces transfer learning to initialize a high-quality population, whereas $\text{mEDCMOA}$ tries to find the optimal distance of decision variables so as to generate initial ones close to the optimal PFs. Fig. S2 of Supplementary depicts IGD of the initial population generated by all comparative methods under each environment, respectively. Intuitively, $\text{MMTL}$ achieves better MIGD, R, and RED on the test problems with no significant change in constraints, e.g., DCP1 and DCP3. However, the boundary of feasible region may be weakly explored, causing the worse MHV values especially on DCP5 and DCP6 that have fixed objective functions. By comparison, $\text{mEDCMOA}$ is suitable for solving DCMOPs with more regular infeasible regions, e.g., DCP2, DCP3, and DCP7. This is also verified by the optimal PFs of the first 30 environments obtained by all comparative algorithms shown in Fig. S3-S11 of Supplementary.

To further investigate algorithm performances, the test suite proposed in [18], called DCTP, is introduced for comparisons. The significance test of MIGD obtained by all comparative algorithms is summarized in TABLE 1, and the detailed statistical results of MIGD, MHV, R, and RED are recorded in TABLE S6-S9 of Supplementary, respectively. We observe

### TABLE 1

<table>
<thead>
<tr>
<th>Problem</th>
<th>$\omega = 0$ vs. $\omega = 0.1$</th>
<th>$\omega = 0$ vs. $\omega = 0.2$</th>
<th>$\omega = 0$ vs. $\omega = 0.3$</th>
<th>$\omega = 0$ vs. $\omega = 0.4$</th>
<th>$\omega = 0$ vs. $\omega = 0.5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCP1</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DCP2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DCP3</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>DCP4</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>DCP5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DCP6</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>DCP7</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DCP8</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>DCP9</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>ALL</td>
<td>24</td>
<td>27</td>
<td>30</td>
<td>25</td>
<td>36</td>
</tr>
</tbody>
</table>

Supplementary, respectively. We observe from the experimental results that TDCEA proposed in this article outperforms the other competitors on MIGD, MHV, R and RED for about 56, 51, 63 and 51 out of 81, respectively.

**Fig. 13.** Effect on different number of decision variables.
from the experimental results that TDCEA achieves the most competitive and robust performance than other competitors in solving DCMOPs. More especially, the initial population achieves good diversity with the help of CCMO, which ensures the stable tracking of PSs changing over time.

D. Comparisons of Change Response Strategies in DCMOPs

To further analyze the effectiveness of two-stage diversity compensation strategy proposed in this article, diversity introduction, prediction approach, memory mechanism, and random initialization are employed for comparisons. TDCEA is renamed as C1, while D-NSGA-II-A, PPS, MNSGA-II and random initialization are termed as C2-C5, respectively. For fair comparison, CCMO is adopted as the static optimizer in all above algorithms.

TABLE 3 presents significance test of MIGD on DCP, while the corresponding statistical results of MIGD, MVH, R, and RED are recorded in TABLE S10-S13 of Supplementary, respectively. We observed from experimental results that TDCEA proposed in this article outperforms other comparative algorithms on the most test instances. More especially, the initial population generated by proposed method is significantly superior than the other competitors. Diversity introduction strategy in D-NSGA-II-A obtains the worst performance on three indicators. Although random individuals provide population with good diversity, the information of historical Pareto optima may not fully utilized. Intuitively, PPS is more suitable to solve DCMOPs with dynamic objective functions, because prediction model can produce an initial population with better convergence in this case. For type II of DCMOPs, prediction model may provide less valuable information for static original PFs. In addition, MNSGA-II also shows potential performance in solving different types of DCMOPs. The quality of the initial population can be improved by reusing the solutions of multiple historical environments, especially for periodic problems.

E. Comparisons of Constraint Handling Strategies

To analyze the influence of constraint handling mechanism on algorithm performance, two-stage diversity compensation strategy is combined with feasibility rule, stochastic ranking, shift-based penalty function, C-TAEA, CAEAD or CMOEA-MS for comparisons. TDCEA is renamed as K1, its variants with six constraint handling mechanisms are termed as K2-K7, respectively.

TABLE 4 presents significance test of MIGD achieved by seven comparative algorithms on DCP, while the
corresponding statistical results of MIGD, MHV, R, and RED are recorded in TABLE S14–S17 of Supplementary, respectively. We observed from the experimental results that constraint handling mechanisms have the obvious effect on the performance in searching feasible Pareto optima changing over time. Intuitively, CCMO achieves the most stable and competitive performance than the other six methods because the assisted population can guide the main one quickly across the infeasible regions, providing tradeoff between distribution and convergence. Feasibility rule prefers to the feasible solutions during the evolution, causing the weak distribution. Stochastic ranking method relaxes the rule of the former one, which helps the evolution across infeasible regions, especially on DCP3 with small and complex feasible regions. Although the performance of shift-based penalty function is significantly worse than CCMO, its convergence speed is potentially quicker than feasibility rule, C-TAEA and CMOEASMS. Additionally, CAEAD can also make good use of its assisted population to provide better information for the main one, achieving good performance on DCP4, DCP6 and DCP8.

VII. CONCLUSIONS

In this article, we develop the generators of objective functions and constraints to facilitate the design of DCMOPs. In order to cover various characteristics of real-world scenarios, a new test suite consisting of nine benchmarks, termed as DCP, is then put forward. Following that, a dynamic constrained multiobjective evolutionary algorithm with two-stage diversity compensation strategy (TDCEA) is proposed. Some initial individuals are randomly generated to replace historical ones in Stage I, improving the global diversity. At the following stage, the increment between center points of PSs in the past two environments is calculated, and employed to adaptively disturb solutions, forming an initial population with good diversity under the new environment. Experimental results on DCP demonstrate that TDCEA is superior to other state-of-the-art DCMOEAs, achieving good performance in solving DCMOPs. Furthermore, two-stage diversity compensation strategy fully utilizes historical knowledge to generate an initial population, improving the robustness of algorithms in tracking the changing Pareto optima.

Though the proposed generators can produce a series of benchmarks with various characteristics for evaluating the performance of DCMOEAs, the scalable test problems are still a challenging and meaningful work. Besides, the irregular change is also an interesting study. Therefore, how to detect environmental changes, especially caused by constraints, stably and accurately is also a meaningful research direction. Also, more diversified constraints, especially equality ones, shall be studied in depth. These issues will be left for further discussions in our future work.

REFERENCES


Guoyu Chen received the B.E. degree from Shandong Technology and Business University, China, in 2017, and the M.E. degree from Nanchang Hangkong University, China, in 2020. He is currently pursing the Ph.D. degree with the School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, China. His main research interests include dynamic multiobjective optimization and dynamic constrained multiobjective optimization.

Yinan Guo (Member, IEEE) received the B.E. degree in automation and the Ph.D. degree in control theory and control engineering from the China University of Mining and Technology, Xuzhou, China, in 1997 and 2003, respectively. She currently is a Professor of Computational Intelligence and Machine Learning with the School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, China, and also with the School of Mechanical Electronic and Information Engineering, China University of Mining and Technology (Beijing), Beijing, China. She has more than 90 publications. Her current research interests include computation intelligence in dynamic and uncertain optimization and its applications in scheduling, path planning, big data processing, and class imbalance learning and its applications in fault diagnosis.

Yong Wang (Senior Member, IEEE) received the Ph.D. degree in control science and engineering from the Central South University, Changsha, China, in 2011. He is a Professor with the School of Automation, Central South University, Changsha, China. His current research interests include intelligent learning and optimization and their interdisciplinary applications.

Dr. Wang is an Associate Editor of the IEEE Transactions on Evolutionary Computation and the Swarm and Evolutionary Computation. He was a recipient of Cheung Kong Young Scholar by the Ministry of Education, China, in 2018, and a Web of Science highly cited researcher in Computer Science in 2017 and 2018.

Jing Liang (Senior Member, IEEE) received the B.E. degree in automation from Harbin Institute of Technology, Harbin, China, in 2003, and the Ph.D. degree in Electrical and Electronic Engineering from Nanyang Technological University, Singapore, in 2009. She is currently a Professor with the School of Electrical Engineering, Zhengzhou University, Zhengzhou, China. Her main research interests are evolutionary computation, swarm intelligence, multiobjective optimization, and neural network. She currently serves as an Associate Editor for the IEEE Transactions on Evolutionary Computation, and the Swarm and Evolutionary Computation.

Dunwei Gong (Member, IEEE) received the B.Sc. degree in mathematics and Ph.D. degree in control theory and control engineering from the China University of Mining and Technology, Xuzhou, China, in 1992 and 1999, respectively, and the M.E. degree in automation from the Beijing University, Beijing, China, in 1995.

He is currently a professor with School of Information Science and Technology at Qingdao University of Science and Technology. He has more than 180 publications. His current research interests include computation intelligence in multiobjective optimization, dynamic and uncertain optimization, and applications in software engineering, scheduling, path planning, big data processing, and analysis.

Shengxiang Yang (Senior Member, IEEE) received the Ph.D. degree from Northeastern University, Shenyang, China, in 1999.

He is currently a Professor of Computational Intelligence and the Deputy Director of the Institute of Artificial Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester, U.K. He has over 400 publications with the H-index of 71 according to Google Scholar. His current research interests include evolutionary computation, swarm intelligence, artificial neural networks, data mining and data stream mining, and relevant real-world applications. Dr. Yang serves as an Associate Editor/Editorial Board Member of a number of international journals, such as the IEEE Transactions on Evolutionary Computation, IEEE Transactions on Cybernetics, Information Sciences, and CAAI Transactions on Intelligence Technology.