SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

December 2026, Vol. 69, Iss. 12, 222201:1–222201:22
https://doi.org/10.1007/s11432-025-5093-x

A multiregional lagrangian-dual-based algorithm for dynamic constrained multiobjective optimization

Miao RONG[1*], Na HU[1], Dunwei GONG[2*] & Yaochu JIN[3]

[1]*School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China*
[2]*School of Microelectronics, Qingdao University of Science and Technology, Qingdao 266061, China*
[3]*School of Engineering, Westlake University, Hangzhou 310030, China*

Received 5 July 2025/Revised 21 September 2025/Accepted 6 November 2025/Published online 16 November 2026

Abstract In dynamic constrained multiobjective optimization problems, the Pareto fronts (PFs) take on various appearances over time, influenced by changes in the unconstrained Pareto fronts and feasible regions. Aimed at accurately and rapidly tracking the changing PF, this paper proposes a multiregional Lagrangian-dual-based dynamic constrained multiobjective optimization algorithm (MLDCMOA). When a change occurs, a multiregional prediction strategy, involving multiregional division, population prediction, and population correction mechanisms, is triggered to estimate the new Pareto optimal set. To handle constraints, a Lagrangian-dual-factor-based technique is tailored to leverage feasibility, convergence, and diversity. It converts the original objectives of individuals and assigns their priorities, which are further embedded with the population selection operator to increase the selection pressure toward the PF in the improved optimizer. Moreover, a hybrid offspring generation approach, derived from the proposed M-objective search strategy, is designed to enhance the spread of the solutions and facilitate the capability of breaking the constrained limitation. To verify its effectiveness, MLDCMOA is compared with four state-of-the-art algorithms on two newly proposed test suites and one practical case. Experimental results demonstrate the superiority of MLDCMOA in providing a competitive forecast and accelerating convergence in a dynamic and constrained environment.

Keywords dynamic constrained multiobjective optimization, prediction, constraint handling, offspring generation, population selection

Citation Rong M, Hu N, Gong D W, et al. A multiregional lagrangian-dual-based algorithm for dynamic constrained multiobjective optimization. Sci China Inf Sci, 2026, 69(12): 222201, https://doi.org/10.1007/s11432-025-5093-x

1 Introduction

Many real-world optimization problems can be described as dynamic constrained multiobjective optimization problems (DCMOPs) and have gained increasing popularity in scientific research and industrial fields, such as fluid catalytic cracking-distillation processes [1], resource scheduling [2, 3], and antenna design [4]. Changes may exist in objectives, constraints, or parameters, giving rise to movements of global optimal solutions, continuous or discrete manifolds of Pareto fronts (PFs), shrinkage or expansion in feasible regions, and switching loss among various scenarios.

To address DCMOPs, algorithms need to simultaneously balance various conflicting objectives subject to constraints before a change occurs [5, 6], producing a set of optimal solutions instead of one optimum. Such solutions consist of the Pareto optimal set (PS) in the decision space, and the PF can be obtained by mapping the PS in the objective space [7].

Without any loss of generality, the PF of a DCMOP is determined using the feasible region (the search space that satisfies constraints) and the unconstrained Pareto front (UPF), which is merely evaluated by using objectives without constraints. Compared with dynamic multiobjective problems (DMOPs), the (dynamic) constraints in DCMOPs restrict the search space, making feasible regions irregularly narrow or disconnected [8]. Different from static constrained multiobjective problems (CMOPs), DCMOPs have dynamic characteristics, leading to a relatively high requirement for an algorithm's tracking ability.

In DCMOPs, due to changing factors, the feasible region and the UPF might exhibit dynamic behaviors [9]. Figure 1 provides an example, where the green area, the red line, and the blue line refer to the feasible region, the constrained PF (CPF, i.e., the PF of a DCMOP), and the UPF, respectively. In addition, the abscissa $f1$ and ordinate $f2$ in the figure stand for the objective values of the two objective functions, and t refers to the time

* Corresponding author (email: rongmiao307@163.com, dwgong@vip.163.com)

© Science China Press 2026

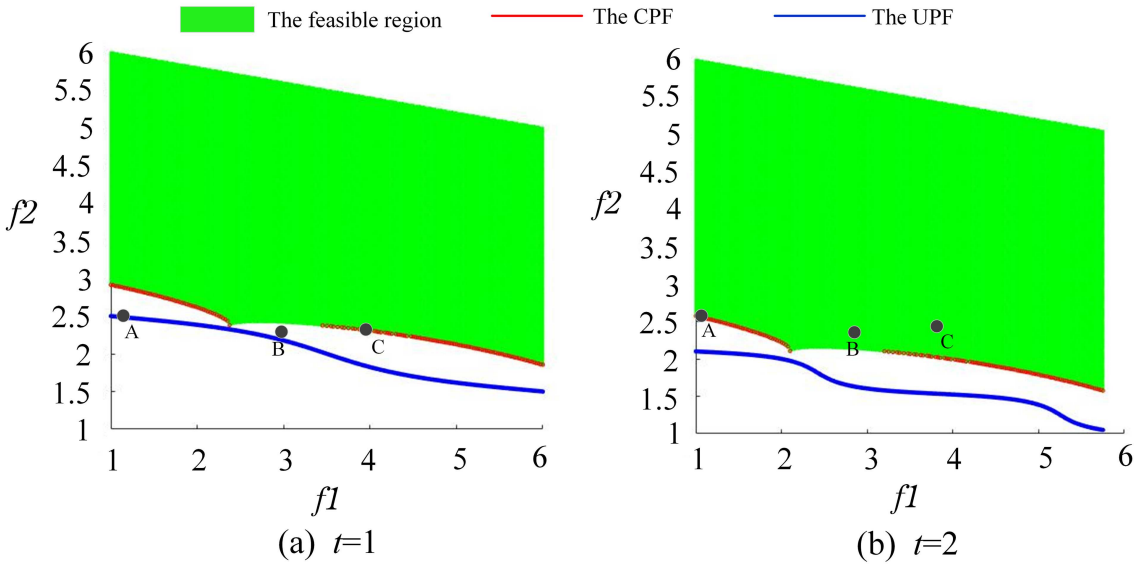


Figure 1 (Color online) Example of changes from $t = 1$ to $t = 2$.

instant of environmental change. From $t = 1$ (Figure 1(a)) to $t = 2$ (Figure 1(b)), the UPF moves down, and the feasible region extends. The UPF cannot be reached, although it is dominant, and the PF can only be located at the bottom of the feasible region. This kind of change also has an impact on individuals A–C. A transforms from an infeasible individual to a nondominated one, while B becomes feasible, and C is no longer nondominated.

There are two main challenges in solving DCMOPs.

(1) *Tradeoff among feasibility, convergence, and diversity.* When infeasible individuals are entirely discarded, the search space tends to gradually shrink, and the population is prone to falling into local optimal regions, making it hardly or even impossible to converge. However, if all infeasible individuals are retained, the population might be misguided and trapped in an unrelated search space, deteriorating feasibility and convergence [10]. Consequently, appropriate use of infeasible individuals, particularly the promising ones, such as those close to the feasible region or the PF and the most likely to become feasible or nondominated [11], is a vital issue in leveraging feasibility, convergence, and diversity.

(2) *Tracking of the dynamic constrained PF.* The shape change of the PF mainly relies on that of the UPF and the feasible region, for instance, regular or irregular, or continuous or discontinuous. However, the UPF might be partially or not overlapping with the feasible region, leading to a drastic PF shift and difficulty in tracking, particularly in scenarios with a very low ratio of feasible regions to the whole objective space [12]. An effective change response strategy is expected to provide accurate guidance, promote the tracking ability, and improve convergence in such cases [13].

Dynamic constrained multiobjective evolutionary algorithms (DCMOEAs) are promising methods to solve DCMOPs. Various approaches [14–16] have been proposed, such as adaptive penalty function [14]-based constraint-handling techniques (CHTs), memory-based change response strategies [15], and multipopulation-based diversity maintenance schemes [16]. However, most CHTs neglect the nonindependent and identically distributed nature of previous optimal solutions. Change response strategies seldom emphasize the significance of feasibility, and most diversity maintenance schemes fail when the change is drastic. Furthermore, considerable computational cost hinders their wider application.

Having realized these, for two-dimensional objective functions, we develop a multiregional Lagrangian-dual-based dynamic constrained multiobjective optimization algorithm (MLDCMOA) to track the changing CPF in acceptable evaluations. In terms of change response, a multiregional prediction strategy is formulated to respond to dynamic constrained changes. To handle constraints, an improved Lagrangian-dual-based CHT is proposed to assign the preference of feasibility and to identify promising infeasible individuals. In the improved optimizer, a hybrid offspring generation operator is designed to facilitate diversity maintenance. Feasibility, convergence, and diversity are achievable at every stage, enabling MLDCMOA to keep track of the changing CPF and making the algorithm computationally prohibitive. The main contributions of this study are as follows.

(1) The tailored multiregional prediction strategy guaranteed strong adaptability to dynamic and constrained scenarios. It included a multiregional division to realize a segmented study influenced by dynamism and constraints,

subregional prediction to construct the prediction model, and population correction to further promote the ratio of feasible regions to the whole objective space.

(2) A Lagrangian-dual-based CHT is adopted to adjust the emphasis on convergence and feasibility and build a bond with evolution. The step size of Lagrangian-dual factors increases with an increase in the penalty on constrained terms until the feasibility reaches a satisfactory value. Furthermore, promising infeasible individuals are reserved to balance feasibility, convergence, and diversity.

(3) The proposed M-objective search strategy, combined with two existing crossover and mutation mechanisms [16], is used for designing the hybrid offspring generation operator, to achieve a good spread of individuals over the PF and strengthen the ability of exploration and exploitation, subject to constraints. It can break the constrained limitation, particularly in the cases of shrinking, expanding, or even disjoint feasible regions.

The rest of this paper is organized as follows: Section 2 introduces the related work. Section 3 presents the overall framework of MLDCMOA and the details of each component. Section 4 describes the experimental setting. Experimental results are depicted and discussed in Section 5. In Section 6, the efficiency of MLDCMOA is further investigated using a real-world case. Section 7 concludes this paper.

2 Related work

When solving DCMOPs, an active DCMOEA tends to have three components: dynamic response, constraint handling, and optimizers. Problem formulation is presented in Appendix A. CHTs and dynamic response strategies referring to dynamic constrained problems with single and multiple objectives are as follows.

2.1 CHTs

CHTs play a vital role in solving constrained optimization problems. They can be divided into two main categories [17]: relatively high priority to feasible solutions for the next generation and tradeoff between diversity and convergence.

2.1.1 *Relatively high priority to feasible solutions for the next generation*

This category emphasizes more on feasible individuals than on infeasible ones [18]. The most widely used measurement scheme is the constrained dominance principle (CDP) based on constraint violation (CV) [19], with the consideration of convergence and feasibility [20–22]. Variants of CV-based CDP have a specified feasibility threshold [23] to determine feasibility, with an angle penalty distance [24] to select individuals, and an angle [25] for a small feasible region.

2.1.2 *Tradeoff between diversity and convergence*

This category attempts to balance the tradeoff between diversity and convergence by introducing promising infeasible individuals, together with feasible individuals, to constitute the whole population. Commonly used measurement schemes include alternative ranking [26], adaptive infeasible weights [27], adaptive penalty functions and distance measures [10], and Lagrangian-dual factors [28].

To summarize, the first category pays more attention to feasible solutions for the next generation, leading to the gradual shrinkage of the search space. Consequently, it may obtain individuals that are trapped in local optima or clustered near the outermost feasible regions. In contrast, the second category makes use of promising infeasible individuals for the next generation, balancing the tradeoff between diversity and convergence.

2.2 Dynamic response strategies

Dynamic response strategies considerably affect the purpose of accelerating convergence in a new environment by providing guidance corresponding to the changes in the environment based on a variety of indicators. They can be divided into three main categories [29]: diversity-based strategies, memory-based strategies, and prediction-based strategies.

Diversity-based strategies focus on diversity maintenance to prevent the population from falling into local optimal areas, where crossover [9] and mutation [30] operators, random generation [31], noise introduction [32], and probability-based gap filling [33] are the commonly used maintenance schemes.

Memory strategies are more advantageous among the three abovementioned types of strategies when the PF of a DMOP regularly changes with a change in the environment [16]. Previous optimal solutions can be reused directly [15], after the similarity comparison among historical environments, or with random selection [34].

Prediction strategies can achieve good results when there is a pattern of regular changes that follow certain rules by predicting variation trends and providing guiding directions [32]. The PS center point is widely incorporated into establishing the prediction model [35,36]. Variants include subpopulation center points [37], subregion center points [38], and orthogonal direction [39]-based prediction models. Transfer learning is also promising for constructing prediction models [40–42].

These strategies can be incorporated to receive better adaptability to changes. However, in DCMOPs, previous optimal solutions may no longer be feasible. Therefore, the dynamic response affected by constraints, even dynamic ones, should be investigated.

To summarize, traditional approaches to DMOPs and CMOPs have difficulties in dealing with DCMOPs. Convergence, diversity, and feasibility should be concurrently balanced during optimization. Two key concerns in constrained change response arouse our attention, i.e., the tracking of the changing constrained PF, and the use of essential infeasible individuals.

3 Proposed method

To begin with, the overall framework of MLDCMOA is introduced. Subsequently, details of mechanisms in MLDCMOA are described, followed by an analysis of the computational complexity.

3.1 Framework

Algorithm 1 outlines the framework of MLDCMOA. For a given DCMOP, MLDCMOA starts with initializing a population, P, which includes N individuals (line 1). In each generation, the change detection mechanism introduced in [16] is used first. If a change occurs, the dynamic response strategy is triggered to reinitialize P (line 4). Thereafter, MLDCMOA adopts an improved Lagrangian-dual-factor-based optimizer to produce the offspring set, Q (line 6), following which, P can be updated by the population selection strategy (line 7). Finally, the nondominated solution set, AS, is obtained based on P by using the nondominated solution selection strategy. This process is repeated (lines 3–8) until the termination condition is met (line 2). MLDCMOA can respond to scenarios when changes exist in objectives, constraints, or both.

Algorithm 1 Framework of MLDCMOA.

Input: Population size, N;
Output: Approximated PS, AS;
1: Initialization: generate an initial population P;
2: **while** termination condition not met **do**
3: **if** a change is detected [16] **then**
4: P = Dynamic Response Strategy (P);
5: **end if**
6: Q = Improved Optimizer (P);
7: P = Population Selection Strategy (P, Q);
8: AS = Nondominated Solution Selection Strategy (P);
9: **end while**
10: **return** AS.

3.2 Initialization

In general, population initialization starts with randomly generating N individuals, where N denotes the population size. However, because of the constraints, there might be a large number of infeasible individuals or even no feasible solutions in the initial population, destroying the performance of a DCMOEA. Therefore, in this study, we assume that the initial population contains at least one feasible individual. The jth variable of the ith individual, $\mathbf{x}_{i,j}$, is generated using

$$\mathbf{x}_{i,j} = \mathbf{x}_j^{min} + rand \times (\mathbf{x}_j^{max} - \mathbf{x}_j^{min}), \tag{1}$$

where $rand$ denotes a random number belonging to $[0, 1]$.

Subsequently, the CV value, $V(\mathbf{x}_i, t)$, of the ith individual, $\mathbf{x}_i$, that contains the equality constraints and the inequality constraints at t is calculated using

$$V(\mathbf{x}_i, t) = \sum_{k=1}^{n_g} G_k(\mathbf{x}_i, t) + \sum_{j=1}^{n_h} H_j(\mathbf{x}_i, t). \tag{2}$$

Herein,

$$\begin{cases} G_k\left(\mathbf{x}_i,t\right)=\max\left\{0,g_k\left(\mathbf{x}_i,t\right)\right\}, & 1\leqslant k\leqslant n_g,\\ H_j\left(\mathbf{x}_i,t\right)=\max\left\{0,\left|h_j\left(\mathbf{x}_i,t\right)\right|-\varepsilon\right\}, & 1\leqslant j\leqslant n_h, \end{cases}$$

where ε denotes the tolerance value of the equality constraints. n_g and n_h represent the number of inequality and equality constraints, respectively.

If $V(\mathbf{x}_i,t)$ is equal to zero, then $\mathbf{x}_i$ is feasible; otherwise, $\mathbf{x}_i$ is infeasible. If there is at least one feasible individual in the initial population, the algorithm can begin to optimize; otherwise, the initial population has to be regenerated. According to the coupon collector problem theory, the probability of at least one feasible solution converges to 1 when the number of samples, m, and the proportion of the feasible region to the entire search space, p, satisfy $m\gg 1/p$.

3.3 Multiregional-based prediction strategy (MRP)

After a change occurs, the new PF and feasible regions are different from the previous ones. As their deviation can hardly be known a priori, it is unreasonable and inaccurate to simply reuse past solutions to track the new PF. Having realized this, an MRP, composed of three main schemes, namely multiregional division, subregional prediction, and population correction, was tailored to forecast the new PS.

The phase angle can visualize the distribution of historical PFs and map dynamism along with the geometry of changes from the time domain to an angle space. Concomitantly, the characterization of each partially changing PF can be extracted. When a change is detected, the multiregional division first allocates individuals into multiple subregions by using phase angles in the objective space. It is derived from the cosine similarity [43] to produce a number of multiple subregions. The cosine similarity is effective in measuring the similarity between two vectors by using the phase angle, where a result of 1 indicates that they share exactly the same orientation, while 0 refers to a perpendicular orientation. We take advantage of this parameter but differently: as our aim is to divide the objective space and to assign the appropriate subregion for each individual, the measurement between individuals is not required. Therefore, only the phase angle is regarded as the distribution indicator. The PF is evenly partitioned, and individuals are placed into subregions based on their phase angles. Thus, the influence of dynamism and constraints can be separately investigated, getting the utmost out of the characteristics in different subregions. Furthermore, the range of subregions changes dynamically because of the different PF distributions at different moments. The number of partitions can be adjusted according to the problem context.

As a classical prediction method for dynamic optimization problems, time series models can uncover patterns of change when PFs across different environments exhibit specific correlations across multiple time windows. Therefore, in subregional prediction, each subregion has an independent guiding direction to produce its estimated center point, which is generated based on the center points at the previous two time instances. This method employs various independent directions and produces one forecast center point for each region, regarded as the predictor. As predictors carry features related to changes and constraints in their subregion, producing predicted individuals around them guarantees the consideration of impacts caused by dynamism and constraints. This provides potential search areas with the consideration of feasibility and convergence. Moreover, subregional prediction generates partial individuals related to historical neighborhood information in the case of diversity loss and misleading.

Finally, in population correction, a part of the infeasible individuals is corrected under the guidance of leaders and immigrates toward the leaders to become feasible or approach feasible regions. Thus, the proportion of feasible individuals in the initialized population is increased. In general, the higher the feasibility ratio is, the more conducive it is to accelerate convergence, thereby further promoting the feasibility and convergence of the reinitialized population.

By using MRP, we finally produce a reinitialized population considering feasibility, diversity, and convergence. The abovementioned diversity refers to the distribution of individuals in the objective space.

3.3.1 *Multiregional division*

First, the phase angles of individuals in the PF are calculated. For an individual, $\mathbf{x}_i^t$, in PF^t, at time t, its phase angle, $\theta(\mathbf{x}_i,t)$, is obtained as

$$\theta_{(\mathbf{x}_i,t)}=arccos\left(\frac{\sum_{j=1}^{M}\left(f_j\left(\mathbf{x}_i,t\right)-z_j^*\right)}{\sqrt{\sum_{j=1}^{M}\left(f_j\left(\mathbf{x}_i,t\right)-z_j^*\right)^2}}\right), \tag{3}$$

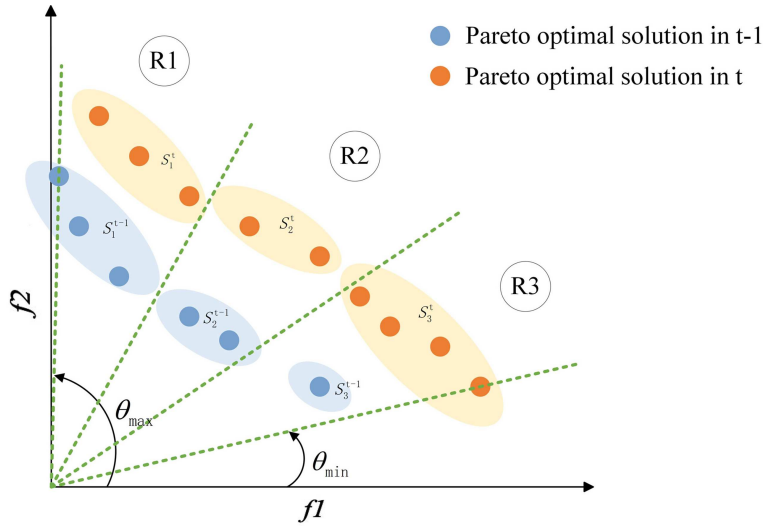


Figure 2 (Color online) Example of multiregional division with three subregions.

where $z_j^* = \min f_m(\mathbf{x}_i, t)$ and M represents the number of objectives. It is worth noting that the formulation applies for $M \leqslant 3$, and as the object dimension ($M > 3$) increases, the possibility of a geometric failure of the angle calculation increases. If one still wants to use the phase angle for multiregion segmentation, the corresponding auxiliary techniques are needed, e.g., target subspace decomposition to realize the hierarchical phase angle or dimensionality reduction.

Next, the maximal and minimal phase angles among all individuals in PF^t and PF^{t-1} can be found, denoted as θ_{max} and θ_{min}. Subsequently, the objective space is evenly divided into n subregions with θ_{max} and θ_{min} as the upper and lower bounds, denoted as ${S_i}^t$ $(i = 1, 2, ..., n)$, and individuals are categorized according to their phase angles.

Figure 2 depicts an example of multiregional division, where individuals in PF^t and PF^{t-1} are partitioned into three subregions. θ_{max} and θ_{min} can be obtained by comparing all phase angles of individuals in PF^t and PF^{t-1}, determining the upper and lower bounds of the area to be divided. Thereafter, R_1, R_2, and R_3 are evenly partitioned within the area, and individuals are categorized according to their phase angles.

Multiregional division splits the entire PF into segmentations, each of which indicates characteristics of the partial PF change influenced by dynamism and constraints. Local features of the PF change can be acquired by separately studying each subregion. Rather than treating the PF as a whole, making use of local features ensures a better description of the internal shape changes in different parts of the PF, particularly with wavy and disjoint changes. Note that the range of subregions varies with upper and lower bounds at different times, further strengthening the adaptability to a certain problem.

3.3.2 *Subregional prediction*

Subregional prediction mainly included three parts: (1) calculating the center points of S_i^t $(i = 1, 2, ..., n)$, (2) predicting the center point of each subregion in the new scenario, and (3) generating a series of predicted individuals. An example of its process is illustrated in Figure 3.

For S_i^t $(i = 1, 2, ..., n)$, its center point, C_i^t, is calculated as follows:

$$C_i^t = \frac{1}{|S_i^t|} \sum_{j=1}^{|S_i^t|} x_{i,j}, \tag{4}$$

where $|S_i^t|$ and $x_{i,j}$ denote the number of S_i^t and the jth individual in S_i^t, respectively.

Furthermore, note that when a subregion is empty, we employ the linear interpolation method to generate a virtual center to substitute the real center. In this case, the center points of two nonempty subregions at t, namely C_i^t and C_k^t, are randomly selected. The relationship among C_i^t, C_k^t, and the virtual center, C_j^t, can be expressed as

$$\frac{C_i^t - C_k^t}{i - k} = \frac{C_i^t - C_j^t}{i - j}. \tag{5}$$

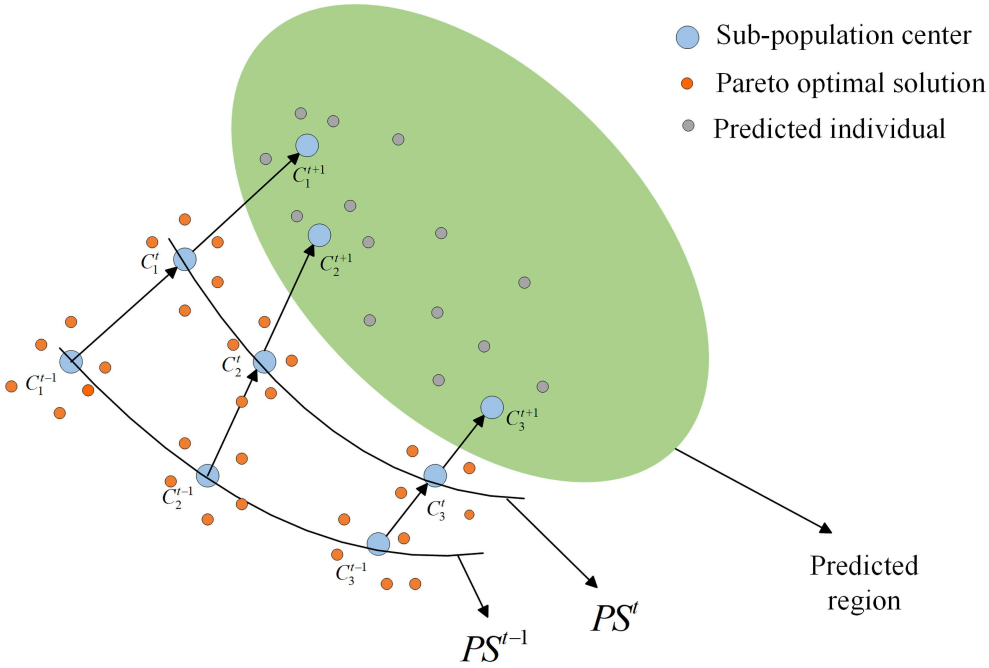


Figure 3 (Color online) Example of subregional prediction.

Therefore, C_j^t can be generated by using

$$C_j^t = C_i^t - \frac{C_i^t - C_k^t}{i-k}(i-j). \tag{6}$$

To estimate the predicted center point for the ith subregion at $t+1$, its moving direction from $t-1$ to t, D_i^t $(i = 1, 2, ..., n)$, is first calculated by using

$$D_i^t = C_i^t - C_i^{t-1}. \tag{7}$$

Thereafter, the predicted center point of the ith subregion, C'^{t+1}_i, can be obtained by using C_i^t and D_i^t based on the linear regression model, shown as

$$C'^{t+1}_i = C_i^t + D_i^t. \tag{8}$$

Finally, two strategies [37] are adopted based on the uniform distribution and the Gaussian distribution to complete subregional prediction. The first one simultaneously considers the neighborhood relationship and the predicted direction of historical centers to increase the diversity of the predicted population. In this strategy, a predicted candidate of ith subregion, x_i^{t+1}, is generated as follows:

$$x_i^{t+1} = C'^{t+1}_i + R(-1,1) \times d_{1,i} + R(-1,1) \times d_{2,i}, \tag{9}$$

where $R(-1,1)$ refers to a random number obtained from a uniform distribution within $[-1,1]$ and $d_{1,i} = D_i^t$. This provides potential guidance to converge to promising and constrained optimal regions. $d_{2,i}$ represents the direction vector between two adjacent center points at t, which is calculated as follows:

$$d_{2,i} = \begin{cases} C_1^t - C_2^t\,, & i = 1, \\ \frac{1}{2}\left(\left(C'^t_{i+1} - C_i^t\right) + \left(C_i^t - C_{i-1}^t\right)\right)\,, & i = 2, \ldots, n-1, \\ C_n^t - C_{n-1}^t\,, & i = n. \end{cases} \tag{10}$$

It establishes the neighborhood relationship among subregions. As center points carry features of the segmented PF changes, controlling the weights on $d_{1,i}$ and $d_{2,i}$ can switch the emphasis on convergence and diversity under the restriction of feasibility.

The second strategy focuses on facilitating convergence in tracking the PF by generating a portion of individuals around the predicted centers. A candidate of the ith subregion, x_i^{t+1}, can be obtained as follows:

$$x_i^{t+1} = {C'}_i^{t+1} + N(0, d_i) \times D_i^t, \tag{11}$$

where d_i represents the Euclidean distance between C_i^t and C_i^{t-1}. $N(0, d_i)$ denotes a random number that follows the Gaussian distribution with the mean of zero and standard deviation (SD) of d_i. By introducing the probability model, we can further improve the diversity.

If the jth element of $\mathbf{x}$, x_j exceeds the boundary, it will be revised according to

$$x_j = \begin{cases} x_j, \ x_j^{min} \leqslant x_j \leqslant x_j^{max}, \\ x_j^{min}, \ x_j \leqslant x_j^{min}, \\ x_j^{max}, \ x_j \geqslant x_j^{max}. \end{cases} \tag{12}$$

The pseudo-code is presented in Algorithm 2.

Algorithm 2 Subregional prediction.

Input: PS^t, PF^t, PS^{t-1}, PF^{t-1};
Output: Predicted population, P';
Calculate the phase angle, θ, of each individual in PF^t and PF^{t-1};
Select θ_{max} and θ_{min};
Evenly divide the objective space into n subregions with θ_{max} and θ_{min} as upper and lower bounds;
According to the θ angle, assign each individual in PF^t and PF^{t-1} to the corresponding subregions;
PS^t and PS^{t-1} of the same subregion, respectively, form subpopulations S_1^t, S_2^t,..., S_n^t, S_1^{t-1}, S_2^{t-1},..., S_n^{t-1};
Calculate the subpopulation centers at time t and $t-1$, C_1^t, C_2^t,..., C_n^t, C_1^{t-1}, C_2^{t-1},..., C_n^{t-1} by using (4) or (6);
for $i = 1 : n$ **do**
    Calculate the predicted center, ${C'}_i^{t+1}$, based on C_i^t and C_i^{t-1} by using (8);
end for
for $i = 1 : n$ **do**
    On the basis of ${C'}_i^{t+1}$, the new individuals are generated using (9) and (11) in the ith subregion and are stored in P';
end for
return P'.

3.3.3 *Population correction*

Three viable correction alternatives are implemented based on the number of feasible individuals, N_{fea}, in P'.

If N_{fea} is less than n_{cor}, where n_{cor} denotes the size of satisfactory feasible individuals in the population, the number of feasible individuals is unsatisfactory. Because of the considerably large number of infeasible individuals, the population may evolve into infeasible regions, deteriorating convergence. Herein, we first discard the infeasible individuals in P' and supplement with randomly generated individuals around feasible ones to fill P'. For example, the jth variable of the filled individual, $fill_x_j$, can be acquired as follows:

$$fill_x_j = fea_x_j + rand \times (x_j^{\max} - x_j^{\min}), \tag{13}$$

where fea_x_j denotes a randomly selected feasible individual and $rand$ represents a random number belonging to $[0.01, 0.1]$. Therefore, the algorithm has a greater likelihood of acquiring a number of feasible individuals. In order to increase the population diversity, $N - |P'|$ individuals are randomly generated and stored in P'.

Subsequently, feasible and infeasible individuals in P' are deposited in archives P and A, respectively. Individuals in P act as leaders for the infeasible individuals in A. For the ith infeasible individual, inf_x_i, its new location, new_x_i, is corrected as follows:

$$new_x_i = inf_x_i + rand(\bar{\mathbf{x}} - inf_x_i), \tag{14}$$

where $\bar{\mathbf{x}}$ denotes a randomly selected leader and $rand$ represents a random number within $[0, 1]$.

If N_{fea} is more than n_{cor}, this situation is more common. Simultaneously, feasible individuals in P' are stored in P and assigned a rank by using nondominated sorting [19]. Individuals in the first and second frontiers are treated as leaders as they are closer to the PF. $N - |P'|$ individuals are randomly generated to fill P' to maintain diversity. Thereafter, all individuals in P' are corrected by using (14).

There is an extreme case where N_{fea} may be zero. In order to improve the feasible ratio as much as possible, $N - |P'|$ individuals are first randomly generated, and leaders are the feasible ones preserved to P, with infeasible ones being placed in P'. Similarly, individuals in P' evolve to leaders by using (14).

Finally, leaders, along with corrected ones, compose the whole P, which is the ultimate initial population. Thus far, population correction has been finished. As those leaders carry genes about feasibility and care about diversity maintenance, with their guidance, infeasible individuals can be pulled or be closer to promising feasible regions. Therefore, population correction can realize the goal of improving diversity, along with the feasibility of the reinitialized population. Its process is presented in Algorithm B1 in Appendix B.

3.4 Lagrangian-dual-factor-based CHT

In the evolutionary process, if we simply retain feasible individuals, the search space will gradually shrink, which may cause the population to fall into the local optimal regions. On the contrary, if too many infeasible individuals are reserved, it may mislead the search direction, which will prevent the population from converging to the PF or even hardly survive to converge.

Therefore, for maintaining the tradeoff between exploration and exploitation under a dynamic and constrained environment, inspired by [28], we develop a Lagrangian-dual-factor-based CHT. With the introduction of Lagrangian dual factors, the original DCMOP is converted into an unconstrained one. The update of these Lagrangian dual factors is associated with a proposed feasible ratio to build a bond with the population. The feasibility ratio adjusts the weights of the objectives and constraints, indicating the emphasis on convergence and feasibility. Moreover, a number of promising infeasible individuals are retained, guaranteeing population diversity.

First, the corrected mth objective of individual x at t, denoted as $F_m(\mathbf{x}, t)$, is formulated as follows:

$$F_m(\mathbf{x}, t) = r \times f_m(\mathbf{x}, t) + \sum_{i=1}^{n_c} \lambda_i \times v_i(\mathbf{x}, t), \tag{15}$$

where r indicates the feasible ratio of the current population, $r = \frac{N_{fea}}{N}$. N_{fea} denotes the number of feasible individuals in the current population. $f_m(\mathbf{x}, t)$ represents the original objective value. $v_i(\mathbf{x}, t)$ and λ_i denote the CV of individual x and a Lagrangian dual factor corresponding to the ith constraint, respectively. n_c represents the quantity of all constraints; herein, $n_c = n_g + n_h$. The ith Lagrangian dual factor, λ_i $(i = 1, 2, ..., n_c)$, is calculated as follows:

$$\lambda_i^{k+1} = \lambda_i^k + s_k \times \sum_{j=1}^{N} v_i(\mathbf{x}_j, t), \tag{16}$$

where s_k and k refer to the update step size and the current iteration, respectively. $\sum_{j=1}^{N} v_i(\mathbf{x}_j, t)$ denotes the CV of the population on the ith constraint at t. Herein, as the feasibility ratio can control the weight on the objectives and CV, we set s_k as $1 - r$ in (16).

By associating step sizes with the feasibility ratio, they are relative to population changes. If the number of feasible individuals is less than the expected value, the step sizes are increased, and a comparatively large variation will be placed on the Lagrangian dual factors. Accordingly, a relatively high penalty will be imposed on the modified objectives.

When r is not zero, there are both feasible and infeasible individuals in the current population. The corrected objectives of feasible individuals are their original ones, among which individuals with lower objective values are selected with preference. For infeasible individuals, their modified objective values include the original objective values and CV values. Similarly, the infeasible individuals with lower objective values and CV values are superior to the others.

On this occasion, if r is relatively considerable, the original objective values are assigned a greater weight, thereby weakening the impact of CV in the CHT. Infeasible individuals with lower objective values and large CV values still have the priority to be selected, as these individuals explore a wider search space and facilitate diversity. On the contrary, if r is smaller, CV will be punished with more weight. Infeasible individuals with lower CV values remain closer to the feasible region than those with larger CV values. Therefore, they are expected to generate feasible offspring, which can help accelerate convergence.

In particular, when the constraint conditions are extremely strict or the feasible regions are particularly small, there are no feasible individuals in the population. Under these circumstances, the corrected objective values are completely determined by CV. Therefore, the penalty pushes individuals with worse CV values toward individuals with better ones, guaranteeing an improvement in feasibility.

Algorithm 3 describes the pseudo-code of the proposed CHT.

3.5 Improved optimizer

This section proposes an improved optimizer based on mating selection and offspring generation for promoting convergence. On the one hand, the Lagrangian-dual-factor-based CHT is introduced into mating selection to balance the number of feasible and infeasible parents. On the other hand, at the stage of offspring generation, a hybrid method including an M-dimensional search strategy is developed to retain the promising offspring.

Algorithm 3 Pseudo-code of Lagrangian-dual-factor-based CHT.

Input: Population, P;
Output: Objective correction values;
Calculate the original objective values for each individual;
Calculate the feasible ratio of the current population, r;
for $i = 1 : n_c$ **do**
    Calculate the Lagrangian factor λ_i by using (16);
end for
for $i = 1 : N$ **do**
    Calculate the objective correction values of individual $\mathbf{x}_i$ by using (15);
end for
return objective correction values.

3.5.1 *Mating selection*

Mating selection is a significant segment in an algorithm to choose parents that can provide the guidance of feasibility, convergence, and diversity [44]. In MLDCMOA, mating selection is embedded with the Lagrangian-dual-factor-based CHT to assign the priority for an individual to be selected as a parent. Next, we adopt the binary tournament selection [21, 45] to determine parents. The pseudo-code of mating selection is depicted in Algorithm C1 in Appendix C.1.

3.5.2 *Offspring generation*

To speed up convergence without causing premature failure, we make a hybrid of the proposed M-objective search strategy and two existing schemes to generate offspring. Local search inevitably destroys diversity, which poses a significant challenge for DCMOEA to jump out of local search, particularly when feasible regions are blocked by multiple disjoint infeasible regions, making it tough for offspring to reach the PF.

Herein, we propose an M-objective search strategy to enhance the exploration and exploitation abilities, which can jump out of the local search if encountered with shrinking, expanding, or even disjoint feasible regions.

First, a parent, X, is randomly selected with its offspring, X', being generated as

$$X' = X + (-1 + 2 \times rand(1, N_{\text{dim}})), \tag{17}$$

where N_{dim} denotes the dimension of the decision variable and $rand(1, N_{\text{dim}})$ means to generate N_{dim} random numbers belonging to $[0, 1]$. If X' is beyond the boundary, it will be corrected by using

$$X'_j = \begin{cases} X'_j, & x_j^{\min} \leqslant X'_j \leqslant x_j^{\max}, \\ x_j^{min} + k(X'_j - x_j^{min}), & X'_j \leqslant x_j^{min}, \\ x_j^{max} + k(X'_j - x_j^{max}), & X'_j \geqslant x_j^{max}, \end{cases} \tag{18}$$

where k denotes the tolerance range. In this study, $k \in [0, 0.5]$.

Subsequently, we calculate the Mth objective function value $f_{X'}^M$ for X' and f_X^M for X. If $f_{X'}^M$ is better than f_X^M, then X' is retained as an offspring. The above process is repeated until n_{p1} offspring are obtained. To leverage the computational cost and convergence accuracy, the M-objective search strategy was executed every five iterations during the optimization.

In this strategy, individuals with a better Mth objective are viewed as leaders, under whose guidance, other offspring immigrate close to them. This gives the offspring a chance to explore a wider area closer to the ideal position. When being trapped in a local optimal region, faced with the shrinkage or expansion of feasible regions, or blocked by disconnected infeasible regions, the population still has the opportunity to jump out of the bridge. One benefit of this is that it is able to balance the need for convergence toward the PF and the requirements of diversity and feasibility.

Let Figure 4 be an example. In Figure 4(a), the population consists of blue individuals trapped in the local search. After the M-objective search strategy, as shown in Figure 4(b), infeasible individuals are pulled down and jump the bridge.

To further eliminate the probability of falling into the local search, we propose a hybrid method by taking advantage of various schemes, i.e., popular simulated binary crossover (SBX) with polynomial mutation (PM) [46], one-dimensional search [16], and Mth objective search. Results of the running time in Subsection 5.2 reveal that the computational cost of these three approaches is acceptable.

These three procedures are simultaneously exploited to generate offspring, following with the population selection strategy discussed in the next section. The pseudo-code of offspring generation is provided in Algorithm C2 in Appendix C.2.

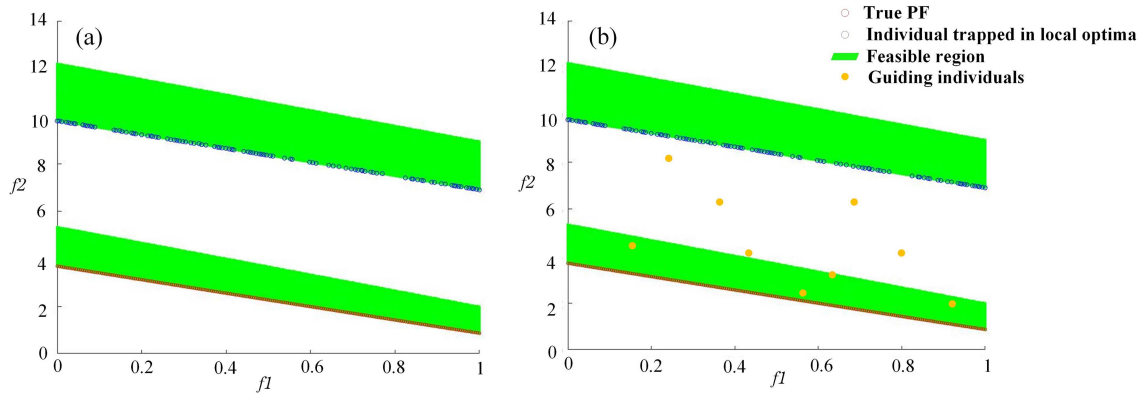


Figure 4 (Color online) Example of M-objective search strategy. (a) The population trapped in the local search; (b) M-objective search strategy takes effect.

3.6 Population selection

Choosing an appropriate population has an important impact on the distribution and search direction of the subsequent populations. If only feasible individuals are allowed to enter the next generation, it may be impossible to effectively explore the entire search space. If too many infeasible individuals are randomly chosen to survive to the next generation, the requirement that all solutions are feasible in PS cannot be satisfied. Consequently, we introduce the Lagrangian-dual-factor-based CHT in the population selection strategy to balance the diversity and convergence. The final population, P, consists of N individuals, which are selected based on the following nondominated solution selection strategy [19] from all candidates produced by the proposed offspring generation method.

3.7 Computational complexity of MLDCMOA

As shown in Algorithm 1, the computational complexity of MLDCMOA is mainly attributed to constraint handling, mating selection, offspring generation, and population selection. In the proposed CHT, the computational complexity of calculating the original objective values is $O(MN)$, and the calculation of CV requires $O(n_cN)$, where n_c is the number of constraints. Mating selection incurs a cost of $O(MN)$. Offspring generation is achieved at the expense of two aspects: (1) the SBX- and PM-based procedure with complexity $O(MN)$, and (2) the M-objective search strategy with complexity $O(Mn_{p1})$, where n_{p1} denotes the number of parents. It takes $O(MN^2)$ for the population selection. The total computational complexity is $O(MN+n_cN+MN+Mn_{p1}+MN^2)$. Since $n_{p1} \ll N^2$, and MLDCMOA finally consumes $O(MN^2)$.

4 Experimental setting

4.1 Test cases and compared algorithms

In terms of test cases, two types of newly proposed and commonly used DCMOPs are considered, i.e., Ins1-8 [15], and DCTP test suites [47], to evaluate the performance of all algorithms. The assessment on Ins1-8 is carried out to test the adaptability of switches between continuous and discontinuous PFs with various fluctuations. The latter is used to examine the ability to track scattering and shifting variations impacted by disjointed feasible regions. The descriptions of these test cases are provided in Appendix D.

Because of the limited number of dynamic constrained multiobjective optimization algorithms proposed and the generalization capability of MLDCMOA, the following conditions are considered when selecting the comparison algorithms: solving dynamic CMOPs, generalization capability, and recently proposed optimization methods. In summary, four state-of-the-art DCMOEAs are compared with MLDCMOA, namely, mEDCMOA with the computational complexity of $O(MN^2)$ or $O(N^2 \log N)$ [16], dCMOEA with the cost of $O(MN^2)$ or $O(N^2 \log N)$ [15], DCNSGA-II-A with the computational cost of $O(MN^2)$ or $O(N^2 \log N)$ [47], and DCNSGA-II-B with the consumption of $O(MN^2)$ or $O(N^2 \log N)$ [47]. Among them, mEDCMOA shows superiority on Ins1-8, while the other three perform noticeably well on the DCTP test suites. It is worth emphasizing that only the pairwise comparisons between each chosen DCMOEA and the proposed MLDCMOA are considered. The computational complexity description reveals that the computational cost of MLDCMOA is close to that of the compared algorithms.

In particular, to verify the performance of MLDCMOA in solving real-world problems, we deliberately apply it to a dynamic economic environment dispatch model (DEED) and compare it with the other algorithms. For details of DEED with the result analysis, please refer to Appendix F.

4.2 Parameter settings

The related parameters are set as follows. In the test cases, τ_t represents the frequency of environmental change, and a smaller value of τ_t indicates faster environmental changes. n_t represents the number of dynamic environments. Herein, τ_t is set to 5, 10, and 20, indicating the high, medium, and low frequencies of changes, respectively. n_t is set to 21 and 7 for the test cases in [15,47], respectively. n_{cor} is set as $N/4$ [16]. To eliminate the impact of stochastic initialization, algorithms will run 30 generations before the first change occurs. The total evaluation is $30+\tau_t \times n_t$. Settings with respect to the decision variables and the other parameters of the test cases are the same as those in [15,47]. MLDCMOA and its counterparts are independently run 20 times, with their statistical results recorded.

In MLDCMOA, the number of subregions is set to 6 after comparison; refer to Appendix E.1 for details. The population size is set to 200, with the PS size set to 100. Referring to [16], the crossover probability is set to 0.8 with the crossover distribution index set to 5 in SBX. In PM [48], the mutation probability is set to 0.05, and the mutation distribution index is set to 40. According to [28], the initial values of Lagrangian dual factors are randomly generated within $(0, 2]$. Parameters in the compared algorithms are the same as the original settings [15,16,47].

4.3 Performance indicators

Metrics of the mean inverted generational distance (MIGD) [49] and the mean hypervolume (MHV) [50] are used as measurements for evaluating the performance of the considered algorithms. A detailed description of the reason why the performance metrics were chosen, and the corresponding formulas can be found in Appendix D.2.

5 Experimental results and analysis

5.1 Comparison with DCMOEAs

In this section, the mean and SD results of MIGD, MHV, and IGD+ obtained by using MLDCMOA and its counterparts are investigated with different pairs of (τ_t, n_t) as listed in Tables 1–4, described as mean (SD), among which the best results are marked in bold. Results of the DCTP suite are presented in Tables E1 and E2 in Appendix E. Without any loss of generality, the Wilcoxon rank-sum test at the 0.05 confidence level [51] is used for determining whether MLDCMOA significantly differs from its comparisons. Herein, "+," "−," and "=" in the data table indicate that MLDCMOA can obtain significantly better or worse results than or equivalent results to those of the comparison algorithm, respectively.

As shown in Table 1, MLDCMOA can achieve the best results of MIGD on most test instances. For example, on Ins2 with $\tau_t = 20$, the mean MIGD value of MLDCMOA is around 10% less than that of dCMOEA. Although the mean MIGD value of mEDCMOA is slightly better than that of MLDCMOA on Ins6 with $\tau_t = 14$, there is no significant difference between them, as indicated by the Wilcoxon rank-sum test. Furthermore, dCMOEA has the worst results among all five algorithms, perhaps because the feasibility and misleading prediction in the dynamic response are overlooked. It is interesting that DCNSGA-II-A and DCNSGA-II-B can obtain similar mean and SD values of MIGD on all test instances with $\tau_t = 5$ and 10. This can be attributed to the same CHT and the partially same dynamic response strategy.

The MHV results corresponding to Table 1 are shown in Table 2. As shown in Table 2, MLDCMOA is significantly superior to its counterparts with different frequencies. The MHV results of the compared algorithms are relatively worse than those of MLDCMOA, probably because of the same CHT and less attention to feasibility in the dynamic response in the former. In contrast, MLDCMOA takes feasibility, convergence, and diversity into account at every stage.

Tables E1 and E2 in Appendix E present the MIGD and MHV results on the DCTP suite, where MLDCMOA obtains competitive mean and SD values. As shown in Table E1, mEDCMOA seems inferior to DCNSGA-II-B, possibly because of the diversity loss when encountered with problems with discrete PFs. Furthermore, dCMOEA performs the worst among all algorithms. It is worth noting that on DTCP6 and DTCP8, MLDCMOA can obtain competitive mean and SD results as compared to mEDCMOA, dCMOEA, and DCNSGA-II-A. This might be attributed to the fact that the population produced by the latter easily falls into the local optimal regions.

As shown in Table E2, MLDCMOA outperforms its counterparts on a majority of the test cases with different frequency settings, particularly, on DCTP6 and DCTP8, which are difficult to converge because of the splitting of

Table 1 Mean and standard deviation (SD) of MIGD obtained by five algorithms on Ins1-8 [15].

Prob.	(τ_t, n_t)	mEDCMOA	dCMOEA	DCNSGA-II-A	DCNSGA-II-B	MLDCMOA
Ins1	(5, 21)	0.0050 (1.4503e−04)=	0.0076 (2.7514e−04)+	0.0064 (1.7581e−04)+	0.0065 (1.4178e−04)+	**0.0047 (1.1131e−04)**
	(10, 21)	0.0051 (1.7594e−04)=	0.0171 (1.5146e−03)+	0.0088 (3.9624e−04)+	0.0097 (4.7991e−04)+	**0.0047 (1.6128e−04)**
	(20, 21)	0.0051 (2.7554e−04)=	0.0280 (3.3382e−03)+	0.0103 (6.1470e−04)+	0.0199 (3.2754e−03)+	**0.0045 (1.7750e−04)**
Ins2	(5, 21)	0.0029 (7.5095e−05)=	0.0054 (1.8158e−04)+	0.0045 (7.9839e−05)+	0.0046 (7.8277e−05)+	**0.0028 (7.7772e−05)**
	(10, 21)	0.0030 (8.2255e−05)=	0.0150 (2.1014e−03)+	0.0075 (3.4940e−04)+	0.0077 (4.3900e−04)+	**0.0029 (1.1339e−04)**
	(20, 21)	0.0037 (1.4639e−04)=	0.0328 (4.8885e−03)+	0.0119 (8.2380e−04)+	0.0149 (2.2645e−03)+	**0.0034 (2.1954e−04)**
Ins3	(5, 21)	0.0051 (1.4592e−04)=	0.0078 (2.9223e−04)+	0.0065 (1.5650e−04)+	0.0066 (1.4175e−04)+	**0.0048 (8.4925e−05)**
	(10, 21)	0.0054 (1.5195e−04)=	0.0178 (1.6315e−03)+	0.0091 (3.4230e−04)+	0.0098 (3.7351e−04)+	**0.0049 (1.6445e−04)**
	(20, 21)	0.0063 (2.5805e−04)=	0.0299 (4.6628e−03)+	0.0111 (7.3433e−04)+	0.0190 (3.7728e−03)+	**0.0057 (2.4390e−04)**
Ins4	(5, 21)	0.0038 (909520e−05)=	0.0061 (1.5254e−04)+	0.0051 (1.0522e−04)+	0.0051 (1.2436e−04)+	**0.0036 (6.8358e−05)**
	(10, 21)	0.0039 (1.2874e−04)=	0.0138 (1.7868e−03)+	0.0078 (2.5821e−04)+	0.0076 (2.0850e−04)+	**0.0037 (1.0672e−04)**
	(20, 21)	0.0039 (1.7733e−04)=	0.0350 (5.4235e−03)+	0.0114 (9.3381e−04)+	0.0131 (1.6519e−03)+	**0.0036 (1.4215e−04)**
Ins5	(5, 21)	0.0062 (1.6324e−04)+	0.0084 (3.2932e−04)+	0.0074 (1.3352e−04)+	0.0073 (1.5819e−04)+	**0.0057 (1.4263e−04)**
	(10, 21)	0.0063 (1.8920e−04)=	0.0171 (1.7088e−03)+	0.0095 (3.8560e−04)+	0.0102 (5.3231e−04)+	**0.0059 (1.6784e−04)**
	(20, 21)	0.0066 (4.6096e−04)=	0.0255 (2.8404e−03)+	0.0114 (7.4169e−04)+	0.0191 (2.4266e−03)+	**0.0061 (2.5559e−04)**
Ins6	(5, 21)	0.0040 (1.2250e−04)=	0.0063 (2.1433e−04)+	0.0055 (1.1043e−04)+	0.0054 (1.3475e−04)+	**0.0039 (1.2621e−04)**
	(10, 21)	**0.0043 (1.5081e−04)=**	0.0151 (2.7115e−03)+	0.0086 (4.0135e−04)+	0.0086 (2.8403e−04)+	0.0044 (2.4268e−04)
	(20, 21)	0.0060 (2.5263e−04)=	0.0313 (4.4276e−03)+	0.0135 (9.5991e−04)+	0.0148 (1.4807e−03)+	**0.0059 (3.1575e−04)**
Ins7	(5, 21)	0.0065 (1.3012e−04)=	0.0087 (2.6124e−04)+	0.0077 (1.5690e−04)+	0.0076 (1.4556e−04)+	**0.0060 (1.4101e−04)**
	(10, 21)	0.0068 (1.9111e−04)=	0.0175 (1.9431e−03)+	0.0102 (3.5264e−04)+	0.0106 (3.6129e−04)+	**0.0065 (1.8209e−04)**
	(20, 21)	0.0087 (4.6624e−04)=	0.0264 (3.4369e−03)+	0.0132 (9.8700e−04)+	0.0178 (1.8456e−03)+	**0.0082 (3.1147e−04)**
Ins8	(5, 21)	0.0049 (1.0661e−04)=	0.0070 (2.0620e−04)+	0.0059 (1.0599e−04)+	0.0059 (1.3271e−04)+	**0.0046 (1.1078e−04)**
	(10, 21)	0.0050 (2.5239e−04)=	0.0142 (1.9630e−03)+	0.0084 (2.1328e−04)+	0.0085 (3.2668e−04)+	**0.0048 (1.2091e−04)**
	(20, 21)	0.0053 (2.0679e−04)=	0.0283 (4.6752e−03)+	0.0119 (6.6857e−04)+	0.0134 (1.5667e−03)+	**0.0051 (2.2075e−04)**
+/=/−		0/24/0	24/0/0	24/0/0	24/0/0	

the feasible region. Under these circumstances, MLDCMOA is still prone to jumping out of the local search and maintaining good diversity. There is no significant difference between the results of MLDCMOA and dCMOEA on DCTP2, DCTP3, and DCTP5 with $\tau_t = 5$ and $\tau_t = 10$. However, the SD of the former is remarkably better than that of the latter, suggesting that MLDCMOA can remain stable to various changes. The mean MHV values of mEDCMOA, DCNSGA-II-A, and DCNSGA-II-B on DCTP2, DCTP3, and DCTP5 are relatively the same, while mEDCMOA, dCMOEA, and DCNSGA-II-A perform the worst on DCTP6 and DCTP8.

To slightly better reflect the quality of the solution set and reduce pseudo-advantages, IGD+ [52] has relatively high robustness to PF shapes, and it is better adapted to a variety of complex multiobjective optimization problems. Therefore, to further demonstrate the performance of MLDCMOA, the results of IGD+ are additionally presented in this paper, with Tables 3 and 4 indicating the two test sets. It can be seen that the overall performance of MLDCMOA is superior, and although it is close to the performance of mEDCMOA in Ins, the former has an absolute advantage in DCTP. In summary, both IGD and IGD+ prove that MLDCMOA is competitive in addressing DCMOPs.

5.2 Comparison of the runtime

Runtime plays an equivalent part in evaluating the performance of an algorithm [53]. Therefore, Tables 5 and 6 list the runtime consumed by MLDCMOA and its counterparts, where the black bolded fonts indicate the shortest run time. As depicted in Table 5, MLDCMOA always converges first and takes only one-third of the time consumed by mEDCMOA. As shown in Table 6, MLDCMOA and mEDCMOA are inferior to the others, and MLDCMOA operates at a smaller expense than mEDCMOA. This is because MLDCMOA requires significant time for the M-objective and one-dimensional search strategies, while mEDCMOA demands substantial time for population selection and nondominated solution selection operators. However, combined with the improvement with respect to the IGD and HV results, the increased runtime can still be acceptable.

Table 2 Mean and SD of MHV obtained by five algorithms on Ins1-8 [15].

Prob.	(τ_t,n_t)	mEDCMOA	dCMOEA	DCNSGA-II-A	DCNSGA-II-B	MLDCMOA
Ins1	(5, 21)	0.2716 (3.2329e−05)=	0.2700 (1.6012e−03)=	0.2712 (1.2910e−04)=	0.2712 (9.7782e−05)=	**0.2740 (1.2229e−03)**
	(10, 21)	0.2710 (5.3393e−05)=	0.2686 (1.0551e−03)=	0.2704 (3.7709e−04)=	0.2702 (4.1709e−04)=	**0.2727 (1.0994e−03)**
	(20, 21)	0.2677 (1.5918e−04)=	0.2658 (1.1651e−03)=	0.2671 (5.4205e−04)=	0.2677 (5.3402e−03)=	**0.2707 (1.6148e−03)**
Ins2	(5, 21)	0.3079 (2.8374e−05)=	0.3070 (9.5205e−04)=	0.3080 (1.1412e−04)=	0.3080 (1.0755e−04)=	**0.3100 (6.0000e−04)**
	(10, 21)	0.3077 (1.3869e−04)=	0.3075 (8.6716e−04)=	0.3080 (3.8377e−04)=	0.3079 (3.8176e−04)=	**0.3095 (1.5418e−03)**
	(20, 21)	0.3066 (7.7784e−04)=	0.3070 (1.6558e−03)=	0.3073 (1.0073e−03)=	0.3067 (5.3457e−03)=	**0.3093 (2.2475e−03)**
Ins3	(5, 21)	0.2727 (2.8688e−05)=	0.2712 (1.6966e−03)=	0.2723 (8.8560e−05)=	0.2723 (9.6829e−05)=	**0.2752 (9.4868e−04)**
	(10, 21)	0.2732 (6.8696e−05)=	0.2702 (1.3365e−03)=	0.2724 (3.2126e−04)=	0.2724 (3.7830e−04)=	**0.2749 (1.1924e−03)**
	(20, 21)	0.2772 (1.0084e−03)=	0.2757 (1.4292e−03)=	0.2768 (1.2295e−03)=	0.2782 (4.2499e−03)=	**0.2813 (3.0350e−03)**
Ins4	(5, 21)	0.3088 (1.4161e−04)=	0.3082 (5.6562e−04)=	0.3089 (7.6891e−05)=	0.3089 (8.1285e−05)=	**0.3117 (7.9729e−04)**
	(10, 21)	0.3090 (2.0150e−04)+	0.3076 (1.2223e−03)+	0.3089 (4.2061e−04)+	0.3090 (2.4171e−04)+	**0.3114 (1.0219e−03)**
	(20, 21)	0.3107 (4.3038e−04)=	0.3106 (2.0523e−03)=	0.3111 (9.0060e−04)=	0.3104 (2.6337e−03)=	**0.3144 (1.7427e−03)**
Ins5	(5, 21)	0.2730 (5.6830e−05)=	0.2723 (1.3404e−04)=	0.2728 (1.1174e−04)=	0.2728 (1.0573e−04)=	**0.2773 (1.4164e−03)**
	(10, 21)	0.2724 (5.5140e−05)=	0.2708 (7.3477e−04)=	0.2720 (3.0710e−04)=	0.2720 (3.3238e−04)=	**0.2753 (1.7454e−03)**
	(20, 21)	0.2684 (1.3360e−03)=	0.2669 (1.4361e−03)=	0.2682 (7.3222e−04)=	0.2694 (5.0190e−03)=	**0.2713 (3.7337e−03)**
Ins6	(5, 21)	0.3104 (3.3025e−05)=	0.3099 (1.3076e−04)=	0.3106 (1.4110e−04)=	0.3106 (1.3456e−04)=	**0.3138 (1.3628e−03)**
	(10, 21)	0.3103 (5.1403e−05)=	0.3100 (6.3730e−04)=	0.3106 (3.8604e−04)=	0.3105 (3.9659e−04)=	**0.3137 (1.1538e−03)**
	(20, 21)	0.3093 (6.7335e−05)=	0.3101 (2.3163e−03)=	0.3097 (1.9754e−03)=	0.3088 (2.7277e−03)=	**0.3128 (1.4974e−03)**
Ins7	(5, 21)	0.2742 (7.7991e−05)=	0.2734 (1.7961e−04)=	0.2740 (1.3494e−04)=	0.2740 (1.0091e−04)=	**0.2786 (1.5701e−03)**
	(10, 21)	0.2747 (1.0301e−04)=	0.2730 (8.5737e−04)=	0.2743 (3.5362e−04)=	0.2741 (3.8798e−04)=	**0.2782 (1.7230e−03)**
	(20, 21)	0.2779 (1.3199e−03)=	0.2777 (1.5657e−03)=	0.2784 (8.8435e−04)=	0.2746 (7.0762e−03)=	**0.2821 (2.7908e−03)**
Ins8	(5, 21)	0.3111 (4.3657e−05)=	0.3103 (1.1494e−04)=	0.3110 (1.0566e−04)=	0.3110 (1.5093e−04)=	**0.3150 (9.5899e−04)**
	(10, 21)	0.3113 (5.2790e−05)=	0.3103 (6.0677e−04)=	0.3113 (3.0183e−04)=	0.3111 (4.6914e−04)=	**0.3143 (1.5766e−03)**
	(20, 21)	0.3126 (7.9255e−05)=	0.3122 (1.6034e−03)=	0.3131 (1.2226e−03)=	0.3125 (2.3332e−03)=	**0.3167 (1.4252e−03)**
+/=/−		1/23/0	1/23/0	1/23/0	1/23/0	

Table 3 Mean and SD of MIGD+ obtained by five algorithms on Ins1-8.

Prob.	(τ_t,n_t)	mEDCMOA	dCMOEA	DCNSGA-II-A	DCNSGA-II-B	MLDCMOA
Ins1	(5, 21)	0.0043 (5.0209e−05)=	0.0095 (2.2633e−03)+	0.0076 (2.1475e−04)+	0.0077 (2.1981e−04)+	**0.0043 (1.2267e−04)**
	(10, 21)	**0.0045 (1.3441e−04)=**	0.0259 (2.6600e−03)+	0.0117 (6.3585e−04)+	0.0134 (7.8000e−04)+	0.0046 (3.4161e−04)
	(20, 21)	**0.0050 (3.4575e−04)=**	0.0467 (6.8287e−03)+	0.0147 (9.9642e−04)+	0.0566 (1.9642e−02)+	0.0054 (1.6828e−03)
Ins2	(5, 21)	**0.0041 (3.7989e−05)=**	0.0091 (1.3936e−03)+	0.0074 (1.5401e−04)+	0.0075 (1.0027e−04)+	0.0045 (7.3946e−04)
	(10, 21)	**0.0042 (2.2859e−04)**−	0.0261 (3.7576e−03)+	0.0125 (6.2808e−04)+	0.0128 (6.5098e−04)+	0.0051 (1.6294e−03)
	(20, 21)	**0.0047 (1.0031e−03)=**	0.0588 (1.0269e−03)+	0.0196 (1.5464e−03)+	0.0306 (7.1486e−03)+	0.0053 (2.3956e−03)
Ins3	(5, 21)	**0.0042 (3.7358e−05)=**	0.0094 (2.1978e−03)+	0.0075 (1.6832e−04)+	0.0075 (1.6144e−04)+	0.0043 (1.0628e−04)
	(10, 21)	**0.0043 (6.8334e−05)=**	0.0272 (3.8014e−03)+	0.0116 (5.3254e−04)+	0.0129 (6.7019e−04)+	0.0046 (1.4824e−04)
	(20, 21)	**0.0051 (1.0796e−03)=**	0.0467 (8.4653e−03)+	0.0141 (1.7217e−03)+	0.0431 (1.6288e−02)+	0.0055 (2.3049e−03)
Ins4	(5, 21)	0.0044 (1.7851e−04)=	0.0081 (6.4477e−04)+	0.0068 (1.2385e−04)+	0.0068 (1.3099e−04)+	**0.0042 (5.2858e−05)**
	(10, 21)	0.0045 (2.1320e−04)=	0.0219 (3.3568e−03)+	0.0114 (4.5060e−04)+	0.0111 (3.4580e−04)+	**0.0043 (6.5075e−05)**
	(20, 21)	0.0053 (4.9995e−04)+	0.0635 (1.0809e−03)+	0.0183 (1.6056e−03)+	0.0250 (5.6767e−03)+	**0.0044 (4.7961e−04)**
Ins5	(5, 21)	0.0040 (9.2417e−05)=	0.0085 (4.0186e−04)+	0.0076 (1.9618e−04)+	0.0075 (1.8539e−04)+	**0.0039 (2.0917e−04)**
	(10, 21)	0.0041 (1.2398e−04)=	0.0263 (3.5175e−03)+	0.0118 (5.8114e−04)+	0.0133 (8.9219e−04)+	**0.0044 (3.1133e−04)**
	(20, 21)	**0.0049 (1.8185e−03)**−	0.0481 (7.2114e−03)+	0.0157 (1.3608e−03)+	0.0563 (1.4045e−02)+	0.0064 (4.6241e−03)
Ins6	(5, 21)	**0.0037 (4.3412e−05)=**	0.0082 (3.2364e−04)+	0.0072 (1.8081e−04)+	0.0071 (1.9357e−04)+	0.0039 (5.4779e−04)
	(10, 21)	**0.0038 (5.3024e−05)=**	0.0254 (5.2524e−03)+	0.0126 (7.3343e−04)+	0.0125 (5.6266e−04)+	0.0041 (6.5416e−04)
	(20, 21)	**0.0039 (8.7681e−05)=**	0.0606 (1.1291e−03)+	0.0202 (2.3390e−03)+	0.0271 (5.2538e−03)+	0.0044 (2.1399e−03)
Ins7	(5, 21)	0.0040 (7.0517e−05)=	0.0084 (4.5874e−04)+	0.0075 (1.9823e−04)+	0.0075 (1.3784e−04)+	**0.0039 (1.6751e−04)**
	(10, 21)	**0.0040 (9.9585e−05)=**	0.0258 (3.7383e−03)+	0.0119 (5.9648e−04)+	0.0128 (6.1766e−04)+	0.0043 (3.6038e−04)
	(20, 21)	**0.0046 (1.1597e−03)**−	0.0431 (7.8774e−03)+	0.0142 (1.4708e−03)+	0.0431 (1.5458e−02)+	0.0060 (4.0502e−03)
Ins8	(5, 21)	0.0039 (5.7062e−05)=	0.0076 (2.7658e−04)+	0.0066 (1.7967e−04)+	0.0066 (1.6129e−04)+	**0.0038 (7.2259e−05)**
	(10, 21)	**0.0040 (3.4632e−05)=**	0.0218 (3.6046e−03)+	0.0111 (4.2668e−04)+	0.0114 (5.6713e−04)+	0.0041 (2.4052e−04)
	(20, 21)	0.0042 (1.4184e−04)=	0.0565 (1.2613e−02)+	0.0184 (1.8450e−03)+	0.0268 (9.9018e−03)+	**0.0041 (8.7899e−04)**
+/=/−		1/20/3	24/0/0	24/0/0	24/0/0	

Table 4 Mean and SD of MIGD+ obtained by five algorithms on DCTP.

Prob.	(τ_t, n_t)	mEDCMOA	dCMOEA	DCNSGA-II-A	DCNSGA-II-B	MLDCMOA
DCTP2	(5, 21)	0.0957 (1.8711e−02)+	0.1363 (2.0610e−02)+	0.1769 (3.8101e−02)+	0.1700 (3.2685e−02)+	**0.0032 (1.7862e−04)**
	(10, 21)	0.1964 (4.0374e−02)+	0.2119 (5.3195e−02)+	0.3349 (8.5685e−02)+	0.3120 (8.7675e−02)+	**0.0033 (2.8026e−04)**
	(20, 21)	0.3074 (7.4467e−02)+	0.3297 (7.0849e−02)+	1.2070 (7.6717e−01)+	0.5947 (1.6346e−01)+	**0.0037 (4.2794e−04)**
DCTP3	(5, 21)	0.1268 (2.3479e−02)+	0.1686 (2.2937e−02)+	0.1878 (3.7197e−02)+	0.1924 (2.8192e−02)+	**0.0204 (1.1882e−03)**
	(10, 21)	0.2097 (4.6672e−02)+	2.2586 (4.7336e−02)+	0.3768 (1.7869e−01)+	0.3218 (1.0509e−01)+	**0.0205 (1.1682e−03)**
	(20, 21)	0.3471 (7.7921e−02)+	0.3503 (7.5624e−02)+	1.2700 (8.6899e−01)+	0.5759 (1.0625e−01)+	**0.0212 (2.5212e−03)**
DCTP5	(5, 21)	0.1018 (1.7545e−02)+	0.1427 (2.7476e−02)+	0.1735 (3.5417e−02)+	0.1683 (3.1033e−02)+	**0.0029 (3.5593e−04)**
	(10, 21)	0.1871 (3.8891e−02)+	0.2199 (5.2711e−02)+	0.3548 (1.2378e−01)+	0.3000 (6.1290e−02)+	**0.0033 (4.0552e−04)**
	(20, 21)	0.3198 (4.8819e−02)+	0.3484 (5.8019e−02)+	1.1984 (7.1428e−01)+	0.5480 (1.2046e−01)+	**0.0032 (4.6633e−04)**
DCTP6	(5, 21)	10.9458 (2.0196e−00)+	11.5732 (1.6823e−03)+	11.5721 (1.5635e−03)+	1.7907 (1.4666e−00)+	**0.0090 (7.1267e−04)**
	(10, 21)	11.5740 (3.0130e−03)+	11.5733 (1.7980e−03)+	11.5735 (2.3263e−03)+	2.1664 (1.6647e−00)+	**0.0098 (1.4836e−03)**
	(20, 21)	11.5750 (2.8446e−03)+	11.5747 (2.8327e−03)+	11.5747 (2.0995e−03)+	3.4497 (3.2622e−00)+	**0.0116 (1.1737e−03)**
DCTP8	(5, 21)	10.9127 (5.7551e−01)+	11.0408 (1.8626e−03)+	11.0694 (4.4105e−02)+	1.9774 (1.5774e−00)+	**0.0246 (1.1783e−02)**
	(10, 21)	11.0359 (3.1418e−03)+	11.0510 (6.0276e−02)+	11.2272 (3.2411e−01)+	2.8509 (1.9862e−00)+	**0.0491 (2.8341e−02)**
	(20, 21)	11.0615 (1.2155e−02)+	11.0605 (9.3768e−03)+	11.6105 (4.9486e−01)+	4.0196 (2.6855e−00)+	**0.0578 (2.7497e−02)**
+/=/−		15/0/0	15/0/0	15/0/0	15/0/0	

Table 5 Runtime (s) on Ins1-8.

Prob.	(τ_t, n_t)	mEDCMOA	dCMOEA	DCNSGA-II-A	DCNSGA-II-B	MLDCMOA
Ins1	(5, 21)	6845.469	3536.658	2287.947	2661.204	**2130.754**
	(10, 21)	3309.213	1454.232	1233.087	1260.508	**1122.704**
	(20, 21)	1718.785	778.352	684.389	680.102	**620.873**
Ins2	(5, 21)	6615.656	3453.296	2422.593	2432.199	**2127.707**
	(10, 21)	3325.548	1604.462	1234.213	1240.388	**1117.236**
	(20, 21)	1764.091	808.299	687.385	676.851	**607.410**
Ins3	(5, 21)	6848.338	3206.080	2529.163	2414.004	**2172.0450**
	(10, 21)	3277.049	1458.892	1237.752	1257.561	**1194.080**
	(20, 21)	1715.070	756.274	692.497	690.375	**603.665**
Ins4	(5, 21)	6735.360	3218.839	2525.457	2446.470	**2214.455**
	(10, 21)	3247.533	1487.603	1206.738	1242.547	**1118.392**
	(20, 21)	1777.568	829.051	658.574	671.793	**615.065**
Ins5	(5, 21)	6629.818	3333.522	2333.280	2451.090	**2196.723**
	(10, 21)	3298.777	1513.854	1229.218	1236.121	**1202.616**
	(20, 21)	1952.760	765.064	702.063	674.574	**651.604**
Ins6	(5, 21)	6605.887	3392.240	2308.217	2442.246	**2143.061**
	(10, 21)	3313.931	1536.258	1236.974	1260.953	**1119.042**
	(20, 21)	1727.502	773.506	700.779	674.335	**631.207**
Ins7	(5, 21)	6808.083	3389.986	2482.457	2497.734	**2172.031**
	(10, 21)	3239.384	1523.958	1244.556	1254.507	**1112.221**
	(20, 21)	1667.773	772.655	668.543	687.977	**618.075**
Ins8	(5, 21)	6669.748	3342.090	2519.327	2426.980	**2211.655**
	(10, 21)	3260.784	1452.840	1216.453	1244.158	**1124.379**
	(20, 21)	1713.569	733.465	703.270	673.889	**635.440**

5.3 Ranking comparison

Furthermore, according to [54, 55], Table 7 provides statistics on the wins and losses recorded when comparing algorithms two-by-two on IGD+, HV, and runtime. The better the performance of an algorithm, the greater the difference between the wins and the losses. If both algorithms obtain approximate values, then their overall performance is similar. The results show that MLDCMOA not only ranks first on IGD+ and HV, which testifies to superior convergence and distribution, but also has the shortest running time, which satisfies the need for a fast search.

Table 6 Runtime (s) on DCTP.

Prob.	(τ_t, n_t)	mEDCMOA	dCMOEA	DCNSGA-II-A	DCNSGA-II-B	MLDCMOA
DCTP2	(5, 21)	699.008	471.197	**450.927**	459.778	477.042
	(10, 21)	439.212	**263.045**	273.630	270.688	263.265
	(20, 21)	241.371	152.392	**143.899**	148.264	178.263
DCTP3	(5, 21)	620.578	452.319	467.033	**452.251**	494.153
	(10, 21)	328.850	262.860	**256.601**	273.302	270.199
	(20, 21)	212.123	138.383	**133.051**	141.761	187.013
DCTP5	(5, 21)	719.052	464.017	**443.864**	452.378	499.142
	(10, 21)	548.871	240.309	**238.701**	246.890	271.602
	(20, 21)	233.789	143.161	146.450	**141.722**	187.993
DCTP6	(5, 21)	823.616	543.911	424.336	**420.298**	520.872
	(10, 21)	657.432	371.188	249.346	**233.373**	293.726
	(20, 21)	297.966	184.469	142.727	**141.834**	191.383
DCTP8	(5, 21)	827.713	596.714	426.011	**406.649**	563.936
	(10, 21)	439.949	295.322	**233.526**	236.099	306.978
	(20, 21)	283.466	178.583	143.388	**138.138**	211.264

Table 7 Ranking comparison on IGD+, HV, and runtime (RT).

Metrics	Results	mEDCMOA	MLDCMOA	dCMOEA	DCNSGA-II-B	DCNSGA-II-A
IGD+	Wins	104	132	16	42	37
	Losses	22	3	125	94	92
	Diff	82	129	−109	−52	−55
	Rank	2	**1**	5	3	4
HV	Wins	10	58	25	23	3
	Losses	29	0	15	27	40
	Diff	−19	58	10	−4	−37
	Rank	4	**1**	2	3	5
RT	Wins	0	116	56	89	80
	Losses	156	36	96	27	30
	Diff	−156	80	−40	62	50
	Rank	5	**1**	4	3	2

5.4 Variation of inverted generational distance

To take a deeper look at the IGD changes, Figures 5 and 6 plot the variation curves of IGD over time obtained by all five algorithms on two test suites. The IGD curves of MLDCMOA show fewer shifts and appear considerably lower than their counterparts. Note that in Figure 5, on Ins5-8, extremely sharp peaks of the IGD variation curves are observed. On the one hand, the changes modify the shape of the unconstrained PF, and oscillations are observed along with switches between continuous and discrete states. On the other hand, dynamic characteristics in constraints make the feasible regions shrink, expand, or get disconnected. Consequently, tracking their PF is relatively challenging. Furthermore, dCMOEA exhibits the worst results as its IGD curves frequently move up and down. It can be inferred that dCMOEA fails to estimate the stable approximated PFs on Ins1-8.

As shown in Figure 6, on DCTP2 and DCTP5, IGD curves exhibit large initial values as they experience disjoint feasible regions. MLDCMOA is able to eliminate the impact brought about by the disconnected and changing feasible regions. However, the results of DCNSGA-II-A, dCMOEA, and mEDCMOA gradually move to a higher attitude on DCTP6 and DCTP8.

5.5 Comparison of approximated PFs

The approximated PFs obtained by MLDCMOA and its counterparts are depicted in Figure 7. Among the executed test cases, the PF of Ins1 switches from continuous to discrete, and its unconstrained PF has one optimization mode that contains convex and concave parts. More challengingly, Ins7 not only exhibited continuous and discrete shapes, but its unconstrained PF also experiences oscillations among several optimization modes, including convexity and concavity. DCTP6 and DCTP8 have disconnected, changing feasible regions, making the population tend to get trapped in the local optimal areas [47]. These factors place higher requirements on algorithms for the leverage

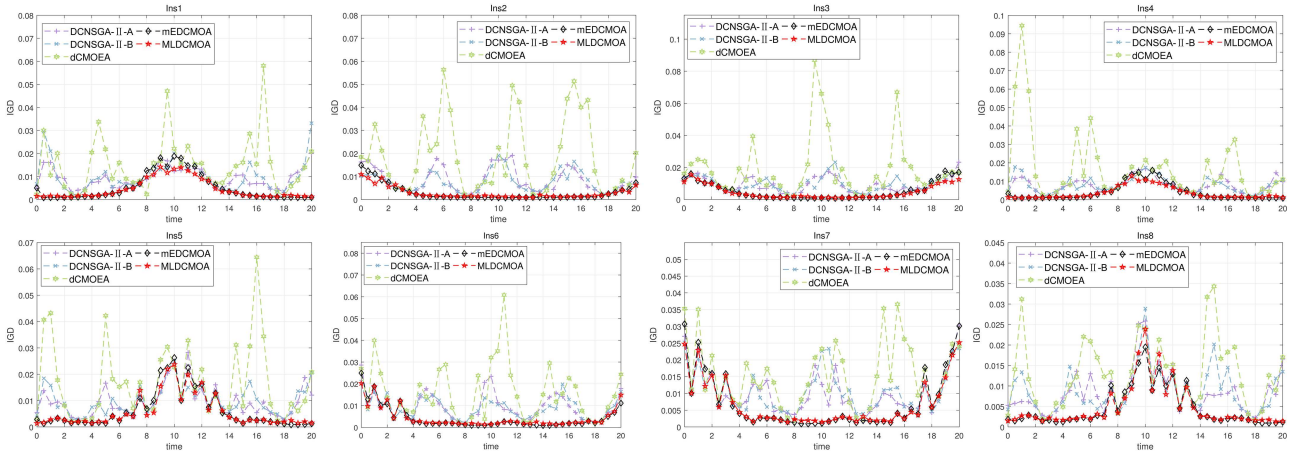


Figure 5 (Color online) Variation of IGD over time obtained by the five algorithms on the first set of test instances [15] with $\tau_t = 10$ and $n_t = 21$, where the purple, blue, green, black, and red dotted lines depict the results of the DCNSGA-II-A, DCNSGA-II-B, dCMOEA, mEDCMOA, and MLDCMOA, respectively.

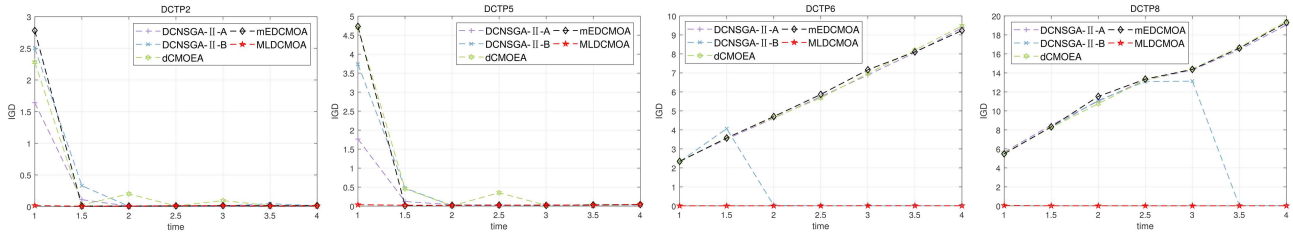


Figure 6 (Color online) Variation of IGD over time obtained by the five algorithms on a part of DCTP with $\tau_t = 10$ and $n_t = 21$, where the purple, blue, green, black, and red dotted lines depict the results of the DCNSGA-II-A, DCNSGA-II-B, dCMOEA, mEDCMOA, and MLDCMOA, respectively.

among feasibility, convergence, and diversity under dynamic scenarios.

In Figure 7, the red and the blue curves correspond to the true and estimated PFs, respectively. As illustrated in Figure 7, MLDCMOA can rapidly lead individuals to immigrate to the true optimal regions when encountering a change and maintain a good distribution. For instance, on DCTP6 and DCTP8, MLDCMOA can jump out of the local optimal areas, probably because of a certain offspring generation operator. In contrast, the compared algorithms show unsatisfactory performance. Note that on Ins7, the performance of these algorithms is inferior to that on Ins1, mainly because of the oscillations within various modes.

5.6 Study on different components of MLDCMOA

To validate the effectiveness of the three components proposed in MLDCMOA, i.e., the MRP, the M-objective search strategy, and the Lagrangian-dual-factor-based CHT, we remove one component from each of them to form three variants, referred to as MLDCMOA-V1, MLDCMOA-V2, and MLDCMOA-V3, respectively. Here, MLDCMOA-V1 uses a randomly generated initial population in each environment instead of MRP. In MLDCMOA-V2, the offspring generation process deletes the M-objective search strategy, and MLDCMOA-V3 searches for the optimal solutions without considering the constraints. The above variants are compared with MLDCMOA on the test problems of Ins3, Ins5 with $(\tau_t, n_t) = (20, 21)$, DCTP2, and DCTP6 with $(\tau_t, n_t) = (10, 7)$. Figure 8 presents the obtained approximate PFs.

The final search results for each variant displayed in Figure 8 visualize the effectiveness of the three components in MLDCMOA. MLDCMOA-V1 blindly generates the initial population that cannot effectively guide the evolutionary direction and ultimately reduces the convergence speed and diversity of the population, e.g., V1-DCTP2 and V1-DCTP6. The removal of the M-objective search strategy causes the MLDCMOA-V2 to fail to converge across the infeasible regions to the true PF when solving the feasible regions' discrete problem, as shown in V2-DCTP6. Without the introduction of CHTs, MLDCMOA-V3 may lead to fewer or even no feasible solutions being searched when the feasible region is narrow or discrete, e.g., V3-Ins3 and V3-Ins5. In particular, in V3-DCTP2 and V3-DCTP6, MLDCMOA-V3 fails to search for a feasible solution in each environment. It is clear that the results obtained by MLDCMOA are far superior to those obtained by the variant algorithms, both in terms of convergence

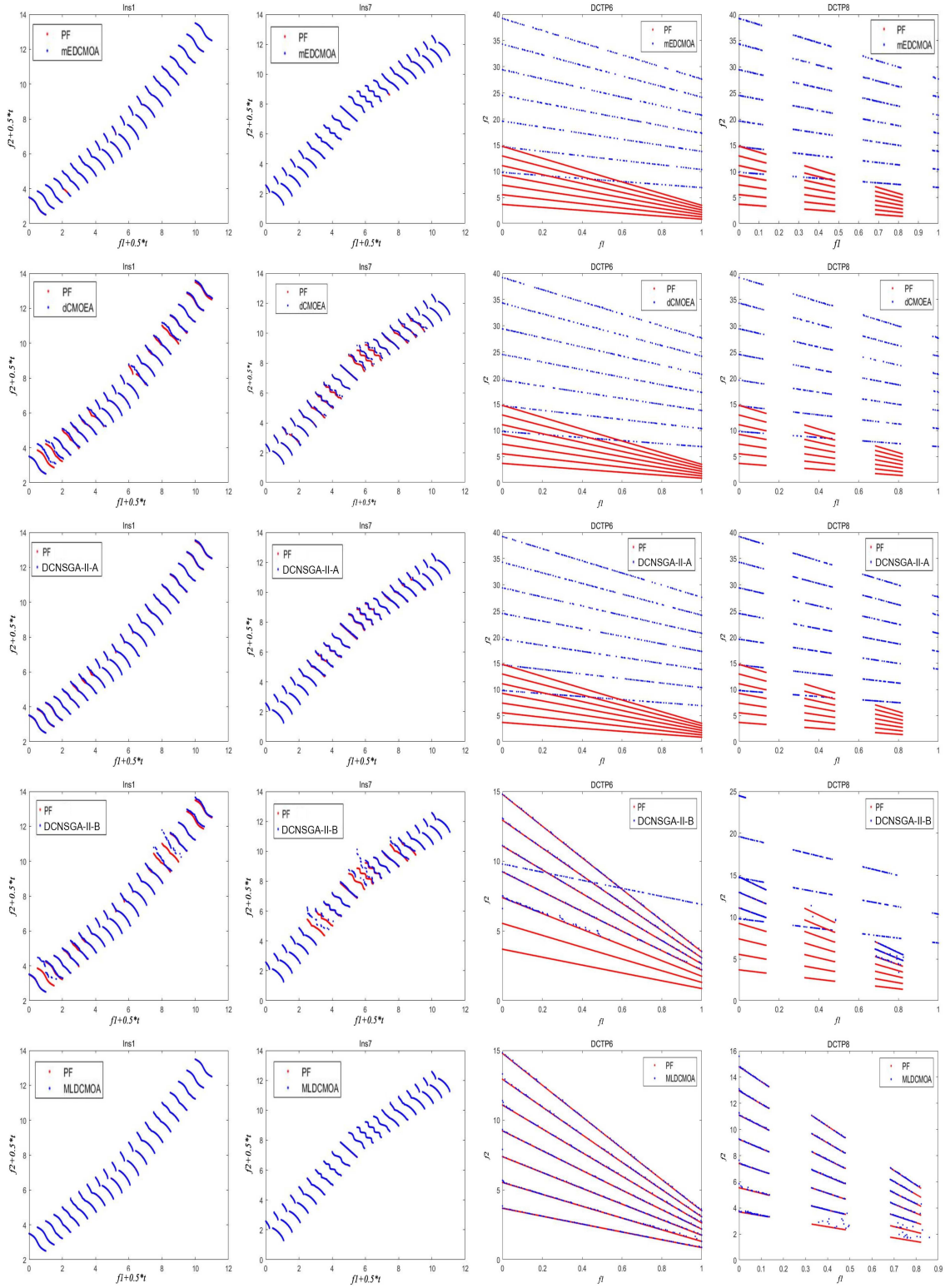


Figure 7 (Color online) Approximated PFs obtained by five algorithms on Ins1 and Ins7 proposed in [15] with $n_t = 21$ and $\tau_t = 20$; DCTP6 and DCTP8 on DCTP with $n_t = 21$ and $\tau_t = 10$, where the red curves indicate the true PF and the blue color denotes the PF obtained by the algorithm.

and distributivity.

In summary, the components designed in MLDCMOA can not only guide the evolutionary direction of the population but also adapt to a variety of optimization environments, and play an important role in improving the

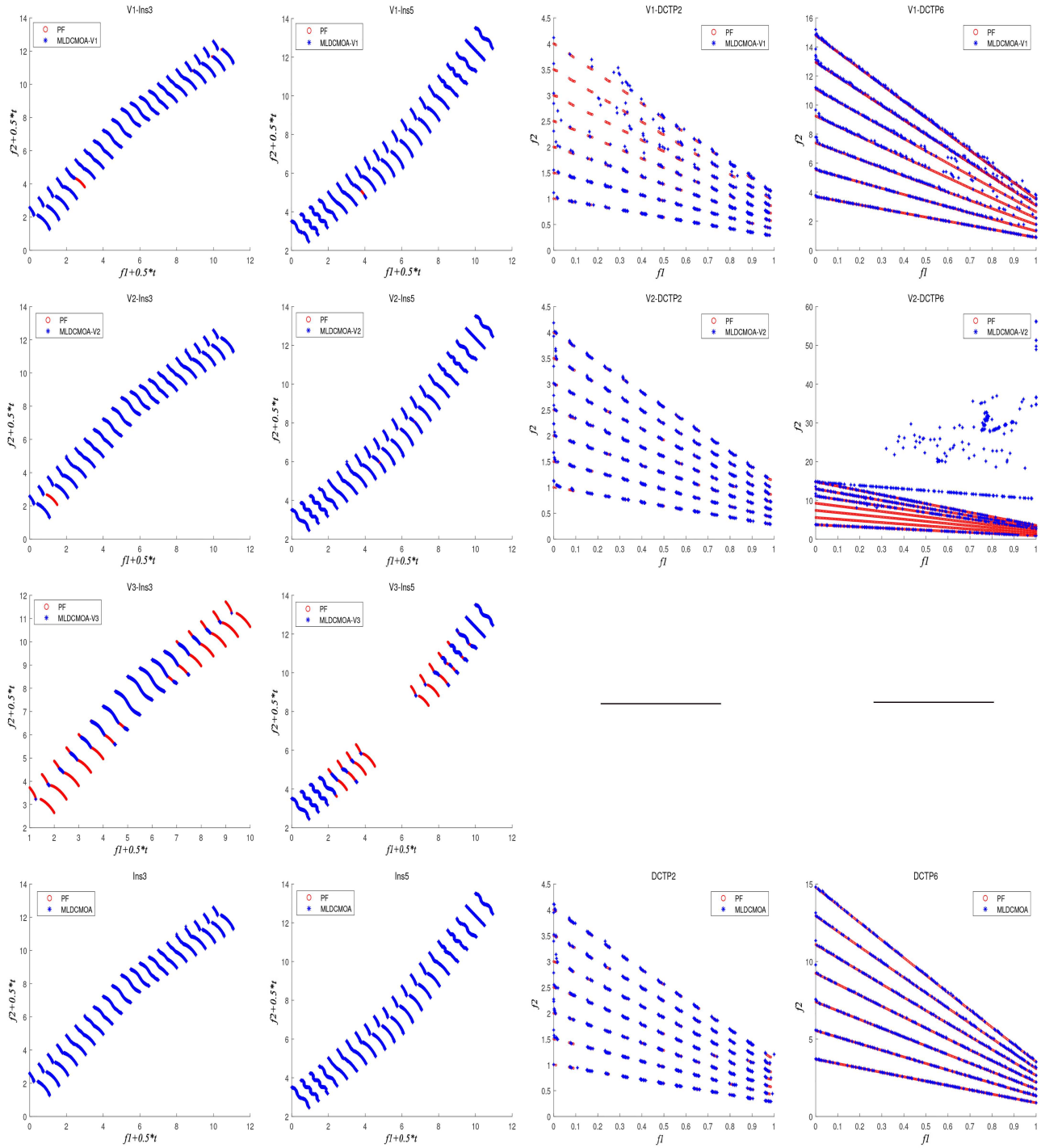


Figure 8 (Color online) Approximated PFs obtained by MLDCMOA with three variants on Ins3, Ins5 with $(\tau_t, n_t) = (20, 21)$, DCTP2, and DCTP6 with $(\tau_t, n_t) = (10, 7)$. In V3-DCTP2 and V3 DCTP6, MLDCMOA-V3 fails to search for a feasible solution in each environment.

algorithm's performance.

6 Practical case

In this section, a real-world case, DEED, is used for evaluating and comparing the efficiency of MLDCMOA with its counterparts. A detailed description of the problem and an analysis of the experimental results are provided in Appendix F.

7 Conclusion

This paper proposes MLDCMOA to solve DCMOPs with diverse changes in objectives and constraints. When detecting a change, the dynamic response strategy is triggered, including multiregional division, population prediction, and population correction. The multiregional division mechanism is proposed to divide the objective space into n subregions and correspondingly generate n predicted directions. Based on these directions, the population prediction strategy estimates the PS, which is hopefully close to or even covers the true PS in a dynamic and constrained environment. Furthermore, the population correction strategy is proposed to increase the feasibility ratio for accelerating convergence with feasibility into consideration. Thereafter, an improved Lagrangian-dual-factor-based CHT is proposed to facilitate exploration and exploitation. Certain criteria are defined to identify promising infeasible individuals and produce guidance for subsequent evolution. Furthermore, in terms of offspring generation, a hybrid procedure of the one proposed and two existing schemes is utilized to facilitate convergence. MLDCMOA considers feasibility, convergence, and diversity at every stage from dynamic response to optimization, increasing its adaptability to DCMOPs with various shape changes, particularly subject to shrinking, expanding, or disjointed feasible regions. MLDCMOA shows competitive performance.

Although MLDCMOA is universal and novel, it has some limitations. More topics need to be studied. First, in the current dynamic response, the linear regression model is used to predict center points for subregions. If encountered with step or sharp changes, it will produce a wrong forecast direction and mislead the population. Second, although subregional prediction and population correction have mechanisms to promote feasibility, they cannot guarantee that all sampling individuals are feasible. Furthermore, while the current number of subregions is fixed, practical applications may require adjusting this count according to the problem complexity or environmental changes. This would create a dynamically adjustable variable, enabling better adaptation to the evolving conditions of the problem. Third, this paper only considers optimization problems with no more than three objective functions. When solving problems with more than three objectives, MLDCMOA still has limitations as it directly maps the original objective space to a two-dimensional axis: differences among other dimensions are eliminated, and information related to convergence, diversity, and feasibility on these dimensions is overlooked. These aforementioned limitations will motivate us to conduct in-depth research in our future studies.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant No. 62103255).

Supporting information Appendixes A–F. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Chen Q, Ding J, Chai T, et al. Evolutionary optimization under uncertainty: the strategies to handle varied constraints for fluid catalytic cracking operation. IEEE Trans Cybern, 2022, 52: 2249–2262
2 Ding J L, Yang C, Xiao Q, et al. Dynamic evolutionary multiobjective optimization for raw ore allocation in mineral processing. IEEE Trans Emerg Top Comput Intell, 2019, 3: 36–48
3 Sundaram A. Multiobjective multi verse optimization algorithm to solve dynamic economic emission dispatch problem with transmission loss prediction by an artificial neural network. Appl Soft Computing, 2022, 124: 109021
4 Jiao R, Sun Y, Sun J, et al. Antenna design using dynamic multi-objective evolutionary algorithm. IET Microwaves Antenna Prop, 2018, 12: 2065–2072
5 Branke J. Evolutionary Optimization in Dynamic Environments. New York: Springer, 2012
6 Zhou A, Qu B Y, Li H, et al. Multiobjective evolutionary algorithms: a survey of the state of the art. Swarm Evolary Computation, 2011, 1: 32–49
7 Coello C A C. Evolutionary Algorithms for Solving Multi-Objective Problems. New York: Springer, 2007
8 Jiang S, Yang S. Evolutionary dynamic multiobjective optimization: benchmarks and algorithm comparisons. IEEE Trans Cybern, 2016, 47: 198–211
9 Deb K, Rao N U B, Karthik S. Dynamic multi-objective optimization and decision-making using modified NSGA-II: a case study on hydro-thermal power scheduling. In: Proceedings of International Conference on Evolutionary Multi-Criterion Optimization, 2007. 803–817
10 Woldesenbet Y G, Yen G G, Tessema B G. Constraint handling in multiobjective evolutionary optimization. IEEE Trans Evol Computat, 2009, 13: 514–525
11 Jiao L, Luo J, Shang R, et al. A modified objective function method with feasible-guiding strategy to solve constrained multi-objective optimization problems. Appl Soft Computing, 2014, 14: 363–380
12 Helbig M, Engelbrecht A P. Population-based metaheuristics for continuous boundary-constrained dynamic multi-objective optimisation problems. Swarm Evolary Computation, 2014, 14: 31–47

13 Wang Z Z, Mei Z R, Jiang M, et al. Knee points based transfer dynamic multi-objective optimization evolutionary algorithm. In: Proceedings of IEEE Congress on Evolutionary Computation (CEC), 2020. 1–7

14 Wang F, Huang M, Yang S, et al. Penalty and prediction methods for dynamic constrained multi-objective optimization. Swarm Evolary Computation, 2023, 80: 101317

15 Chen Q, Ding J, Yang S, et al. A novel evolutionary algorithm for dynamic constrained multiobjective optimization problems. IEEE Trans Evol Computat, 2019, 24: 792–806

16 Chen Q, Ding J, Yen G G, et al. Multipopulation evolution-based dynamic constrained multiobjective optimization under diverse changing environments. IEEE Trans Evol Computat, 2024, 28: 763–777

17 Azzouz R, Bechikh S, Said L B, et al. Handling time-varying constraints and objectives in dynamic evolutionary multi-objective optimization. Swarm Evolary Computation, 2018, 39: 222–248

18 Coello Coello C A, Christiansen A D. MOSES: a multiobjective optimization tool for engineering design. Eng Optimization, 1999, 31: 337–368

19 Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Computat, 2002, 6: 182–197

20 Tian Y, Zhang T, Xiao J, et al. A coevolutionary framework for constrained multiobjective optimization problems. IEEE Trans Evol Computat, 2021, 25: 102–116

21 Li K, Chen R, Fu G, et al. Two-archive evolutionary algorithm for constrained multiobjective optimization. IEEE Trans Evol Computat, 2019, 23: 303–315

22 Takahama T, Sakai S. Efficient constrained optimization by the ε constrained differential evolution with rough approximation using kernel regression. In: Proceedings of IEEE Congress on Evolutionary Computation, Cancun, 2013. 1334–1341

23 Asafuddoula M, Ray T, Sarker R. A decomposition-based evolutionary algorithm for many objective optimization. IEEE Trans Evol Computat, 2014, 19: 445–460

24 Cheng R, Jin Y, Olhofer M, et al. A reference vector guided evolutionary algorithm for many-objective optimization. IEEE Trans Evol Computat, 2016, 20: 773–791

25 Fan Z, Li W J, Cai X Y, et al. Angle-based constrained dominance principle in MOEA/D for constrained multi-objective optimization problems. In: Proceedings of IEEE Congress on Evolutionary Computation (CEC), 2016. 460–467

26 Angantyr A, Andersson J, Aidanpaa J O. Constrained optimization based on a multiobjective evolutionary algorithm. In: Proceedings of Congress on Evolutionary Computation (CEC), 2003. 1560–1567

27 Peng C, Liu H L, Gu F. An evolutionary algorithm with directed weights for constrained multi-objective optimization. Appl Soft Computing, 2017, 60: 613–622

28 Rong M, Zhang D, Wang N. A Lagrangian dual-based theory-guided deep neural network. Complex Intell Syst, 2022, 8: 4849–4862

29 Ma Y J, Chen M, Gong Y, et al. Research progress of dynamic multi-objective optimization evolutionary algorithm (in Chinese). Acta Autom Sin, 2020, 46: 2302–2318

30 Zhang K, Shen C, Liu X, et al. Multiobjective evolution strategy for dynamic multiobjective optimization. IEEE Trans Evol Computat, 2020, 24: 974–988

31 Ruan G, Yu G, Zheng J, et al. The effect of diversity maintenance on prediction in dynamic multi-objective optimization. Appl Soft Computing, 2017, 58: 631–647

32 Zhou A M, Jin Y C, Zhang Q F. A population prediction strategy for evolutionary dynamic multiobjective optimization. IEEE Trans Cybern, 2013, 44: 40–53

33 Hu Y, Zheng J, Jiang S, et al. Handling dynamic multiobjective optimization environments via layered prediction and subspace-based diversity maintenance. IEEE Trans Cybern, 2023, 53: 2572–2585

34 Zhang Y, Yang K, Hao G S, et al. Evolutionary optimization framework based on transfer learning of similar historical information (in Chinese). Acta Autom Sin, 2021, 47: 652–665

35 Wang C, Yen G G, Jiang M. A grey prediction-based evolutionary algorithm for dynamic multiobjective optimization. Swarm Evolary Computation, 2020, 56: 100695

36 Cao L, Xu L, Goodman E D, et al. Evolutionary dynamic multiobjective optimization assisted by a support vector regression predictor. IEEE Trans Evol Computat, 2020, 24: 305–319

37 Jin Y C, Yang C, Ding J L, et al. Reference point based prediction for evolutionary dynamic multiobjective optimization. In: Proceedings of IEEE Congress on Evolutionary Computation (CEC), 2016. 3769–3776

38 Rong M, Gong D, Zhang Y, et al. Multidirectional prediction approach for dynamic multiobjective optimization problems. IEEE Trans Cybern, 2018, 49: 3362–3374

39 Wu Y, Jin Y, Liu X. A directed search strategy for evolutionary dynamic multiobjective optimization. Soft Comput, 2014, 19: 3221–3235

40 Jiang M, Huang Z, Qiu L, et al. Transfer learning-based dynamic multiobjective optimization algorithms. IEEE Trans Evol Computat, 2018, 22: 501–514

41 Guo Y, Chen G, Jiang M, et al. A knowledge guided transfer strategy for evolutionary dynamic multiobjective optimization. IEEE Trans Evol Computat, 2023, 27: 1750–1764

42 Zhang X, Jin Y, Qian F. A self-adaptive dynamic multi-objective optimization algorithm based on transfer learning and elitism-based mutation. Neurocomputing, 2023, 559: 126761

43 Liu Y, Gong D, Sun J, et al. A many-objective evolutionary algorithm using a one-by-one selection strategy. IEEE Trans Cybern, 2017, 47: 2689–2702

44 Tessema B, Yen G G. An adaptive penalty formulation for constrained evolutionary optimization. IEEE Trans Syst Man Cybern A, 2009, 39: 565–578

45 Jiang S, Yang S. A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization. IEEE Trans Evol Computat, 2016, 21: 65–82

46 Deb K, Agrawal R B. Simulated binary crossover for continuous search space. Complex Syst, 1995, 9: 115–148

47 Azzouz R, Bechikh S, Lamjed B S. Multi-objective optimization with dynamic constraints and objectives: new challenges for evolutionary algorithms. In: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, 2015. 615–622

48 Deb K, Goyal M. A combined genetic adaptive search (GeneAS) for engineering design. Comput Sci Inform, 1996, 26: 30–45

49 Muruganantham A, Tan K C, Vadakkepat P. Evolutionary dynamic multiobjective optimization via Kalman filter prediction. IEEE Trans Cybern, 2015, 46: 2862–2873

50 Schutze O, Esquivel X, Lara A, et al. Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective optimization. IEEE Trans Evol Computat, 2012, 16: 504–522

51 Wilcoxon F. Individual comparisons by ranking methods. In: Breakthroughs in Statistics. New York: Springer, 1992. 196–202

52 Lopez E M, Coello C A C. IGD+-EMOA: a multi-objective evolutionary algorithm based on IGD+. In: Proceedings of IEEE Congress on Evolutionary Computation (CEC), 2016. 999–1006

53 Tian Y, Zhang X, Cheng R, et al. Guiding evolutionary multiobjective optimization with generic front modeling. IEEE Trans Cybern, 2018, 50: 1106–1119

54 Helbig M, Engelbrecht A P. Analysing the performance of dynamic multi-objective optimisation algorithms. In: Proceedings of IEEE Congress on Evolutionary Computation, 2013. 1531–1539

55 van der Stockt S A G, Pamparà G, Engelbrecht A P, et al. Performance analysis of dynamic optimization algorithms using relative error distance. Swarm Evolary Computation, 2021, 66: 100930