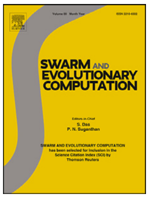
Swarm and Evolutionary Computation 70 (2022) 101041

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo



Combining a hybrid prediction strategy and a mutation strategy for dynamic multiobjective optimization

Ying Chen[a,b], Juan Zou[a,b,*], Yuan Liu[a,b], Shengxiang Yang[a,d], Jinhua Zheng[a,b,c], Weixiong Huang[a,b]

[a] *Key Laboratory of Intelligent Computing and Information Processing, Ministry of Education, School of Computer Science and School of Cyberspace Science of Xiangtan University, Xiangtan, Hunan Province, China*
[b] *Faculty of School of Computer Science and School of Cyberspace Science of Xiangtan University, Xiangtan, 411105, China*
[c] *Hunan Provincial Key Laboratory of Intelligent Information Processing and Application, Hengyang, 421002, China*
[d] *School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, U.K*

ARTICLE INFO

Keywords:
Dynamic multiobjective optimization problems
Evolutionary algorithms
Change response mechanism

ABSTRACT

The environments of the dynamic multiobjective optimization problems (DMOPs), such as Pareto optimal front (POF) or Pareto optimal set (POS), usually frequently change with the evolution process. This kind of problem poses a higher challenge for evolutionary algorithms because it requires the population to quickly track (i.e., converge) to the position of a new environment and be widely distributed in the search space. The prediction-based response mechanism is a commonly used method to deal with environmental changes, but it's only suitable for predictable changes. Moreover, the imbalance of population diversity and convergence in the process of tracking the dynamically changing POF has aggravated. In this paper, we proposed a new change response mechanism that combines a hybrid prediction strategy and a precision controllable mutation strategy (HPPCM) to solve the DMOPs. Specifically, the hybrid prediction strategy coordinates the center point-based prediction and the guiding individual-based prediction to make accurate predictions. Thus, the population can quickly adapt to the predictable environmental changes. Additionally, the precision controllable mutation strategy handles unpredictable environmental changes. It improves the diversity exploration of the population by controlling the variation degree of solutions. In this way, our change response mechanism can adapt to various environmental changes of DMOPs, such as predictable and unpredictable changes. This paper integrates the HPPCM mechanism into a prevalent regularity model-based multiobjective estimation of distribution algorithm (RM-MEDA) to optimize DMOPs. The results of comparative experiments with some state-of-the-art algorithms on various test instances have demonstrated the effectiveness and competitiveness of the change response mechanism proposed in this paper.

1. Introduction

Dynamic multiobjective optimization problems (DMOPs) [1] exist in numerous real-world applications, such as greenhouse control [2], scheduling jobs [3], intelligent robot navigation [4] and industrial design [5]. DMOPs are characterized by some changes that may occur in the Pareto-optimal set in the decision space (POS) or Pareto-optimal front in the objective space (POF). Researchers are committed to finding a set of feasible decision variables to satisfy multiple conflicting objective functions changing over time.

Without loss of generality, to minimize the multiobjective problem as the research object, the mathematical model of a DMOP with n decision variables and M objective functions can be described as:

$$\begin{array}{ll} \boldsymbol{Minimize} & \mathbf{F}(\mathbf{x},t) = \left(f_1(\mathbf{x},t), f_2(\mathbf{x},t), \ldots, f_M(\mathbf{x},t)\right)^T \\ & \text{s.t. } \mathbf{x} \in \Omega \end{array}, \tag{1}$$

where $x = (x_1, x_2, \ldots, x_n)$ is an n-dimensional decision variable vector from the decision space Ω. $F(x,t)$ is the objective function vector that evaluates solution x at time period t. $f_i(x,t)(i = 1, 2, \ldots, M)$ refers to the ith objective function of x at time period t. $\Omega = [L_1, U_1] \times [L_2, U_2] \times \ldots \times [L_n, U_n]$, where L_i and U_i are the lower limit and upper limit of the ith decision variable respectively. The descrete time parameter t is defined as the following mathematical form:

$$t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_t} \right\rfloor, \tag{2}$$

[*] Corresponding author at: Key Laboratory of Intelligent Computing and Information Processing, Ministry of Education, School of Computer Science and School of Cyberspace Science of Xiangtan University, Xiangtan, Hunan Province, China.
E-mail address: zoujuan@xtu.edu.cn (J. Zou).

https://doi.org/10.1016/j.swevo.2022.101041
Received 30 July 2021; Received in revised form 5 January 2022; Accepted 11 January 2022
Available online 12 January 2022

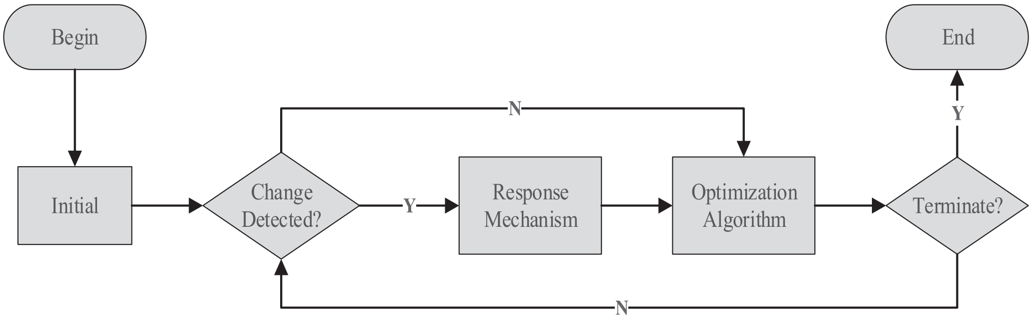


Fig. 1. General framework of DMOEA.

Table 1
Different types of a DMOP.

POF	POS	
2 - 3	No Change	Change
No Change	Type IV	Type I
Change	Type III	Type II

where τ is the iteration counter, n_t is the number of distinct time steps in t, and τ_t is the number of iterations where t remains the same. t controls the dynamics via n_t and τ_t where n_t governs the severity level and τ_t determines the frequency value.

According to whether POF or POS changes or not, DMOPs can be divided into four types, as shown in Table 1. Type I means that POS changes, but POF remains fixed. Type II illustrates that POF and POS both change. Type III means that POF changes, but POS remains unchanged. Type IV presents that both POF and POS remain unchanged, although the problem may change. Here, no attention is given to Type IV, although we recognize that the Type IV change may also occur in some exceptional cases. In addition, there are problems with a mixed type of change, such as JY9 [6]. According to the changes of POF or POS are predictable or not, DMOPs can be divided into problems with predictable changes and problems with unpredictable changes. Predictable change means that the POF or POS changes at least somewhat regularly according to some rules, rather than changing with no pattern between successive environments. Unpredictable changes mean changes of the POF or POS are dramatic or irregular.

Evolutionary algorithms (EAs) simulate natural selection and evolution of organisms and has been widely used to solve highly complex nonlinear problems [7]. Multiobjective evolutionary algorithms (MOEAs) [8–10] are capable of solving various static multiobjective optimization problems (MOPs). Because MOEAs use a multi-point simultaneous search mechanism, which can approximate the entire POS in one run. In recent years, dynamic multiobjective optimization evolutionary algorithms (DMOEAs) to address DMOPs have come into being. Some researchers assumed that a DMOP could be divided into a sequence of MOPs over time and directly migrated and adopted the techniques from static MOEAs [11]. However, environmental changes often take only a few iterations. Techniques directly from static MOEAs tend to fail in most DMOPs due to the lack of a quick change response mechanism.

An excellent DMOEA not only has a good MOEA optimizer but also designs a mechanism that can respond to changes quickly and well. Fig. 1 presents the flowchart of a typical DMOEA. The first key step of a classic DMOEA is change detection, assuming that the dynamics are detectable [12]. Re-evaluating the fitness of a solution is the most common method used to detect changes, provided that incorporated noise handling techniques [13]. Change response is the second key step. Because when a change is detected, the current evolutionary population is required to track and converge to a new POF within very few iterations. Moreover, the imbalance of population diversity and convergence in the process of tracking the dynamically changing POF has aggravated. In other words, the previously obtained optimal solution cannot remain optimal in the new environment. At this time, an excellent change response mechanism to obtain solutions with good convergence and diversity seems particularly important. Earlier methods for change response include memory-based [14] and diversity maintenance [15] [16]. The former directly reuses historical POS to accelerate convergence, while the latter focuses on addressing diversity loss when change occurs. In recent years, prediction-based methods have gained much more attention. They usually adopt the promising historical information as the input to different prediction models (e.g., AR [17], KF [18] and SVR [19]) to predict future optimalities. The third key step is optimization using MOEA regardless of whether environmental changes occur, here we employ RM-MEDA [20] as the underlying MOEA optimizer. As is well known to us, the change response mechanism is the most pivotal, although the three steps mentioned above contribute to efficient problem-solving. Without loss of generality, the contribution of this paper focuses on the second step, that is, change response.

There are limitations in most existing change response mechanisms. Prediction-based methods can speed up convergence, but they may not be effective anymore in responding to unpredictable changes. Sometimes the environmental change may lead to severe diversity loss, so diversity-based methods are also indispensable. However, blindly focusing on diversity in most diversity-based methods will impact the convergency of the evolutionary population. Almost majority of methods are designed only to increase the speed of convergence or only to improve diversity. Can we combine the advantages of prediction-based methods and diversity-based methods to develop a hybrid response mechanism that can converge quickly and maintain diversity? Based on the question in mind, this paper proposes a new change response mechanism that combines a hybrid prediction strategy and a precision controllable mutation strategy (HPPCM).

The hybrid prediction strategy in this paper is good at responding to predictable changes, which coordinates the center point-based prediction and the guiding individual-based prediction to cooperate and complement each other. The center point-based prediction can respond well and quickly to varying degrees of linear changes. In contrast, the guiding individual-based prediction is better at solving nonlinear changes (such as periodic changes). The two in the hybrid prediction strategy make full use of their strengths to make more accurate predictions and accelerate responding to different types of predictable changes. In addition, the precision controllable mutation strategy can respond to unpredictable changes and improve the diversity exploration of the population. The experimental results on various test instances have proved that the HPPCM mechanism proposed in this paper is very competitive in solving DMOPs with different dynamic characteristics. The technical contributions of this paper are summarized as follows.

(1) The center point-based prediction strategy proposed could not only quickly respond to the same two consecutive environmental changes but also flexibly respond to the situation where the degree of the new

change is different from the last one. It employs a revised first-order difference linear prediction model to predict the new locations of current solutions when a change is detected.

(2) The guiding individual-based prediction strategy utilizes the evolutionary direction obtained by the autonomous evolution of the population for several generations to guide future search process and ensure the evolutionary population to approximate the new optimal solutions. It generates some guiding individuals along the evolutionary direction so as to cope with nonlinear (such as periodic) changes and explore the most promising optimization area in the decision space.

(3) The precision-controllable mutation strategy proposed could respond well to unpredictable changes by improving the diversity exploration of the population. It generates mutated solutions according to the required precision, which can not only increase the appropriate diversity but also reduce computational costs.

The remainder of this paper is organized as follows. Section 2 introduces the related work and motivation. The core strategies of HPPCM are presented in Section 3. Section 4 briefly introduces the comparison algorithm, test instances, performance indicators and experimental design. A series of experimental results are presented and analyzed in Section 5. Finally, Section 6 concludes this paper.

2. Preliminaries

In this section, some background knowledge related to our work and the motivation of this paper is introduced.

2.1. Related work

One common idea of prediction-based methods is that given predefined moving step size, the initial solutions to the new environment are obtained by moving the POS of the last time period along with the moving direction [21]. The moving direction is normally estimated by the first-order or second-order difference of some representative points from the previous POS. These representative points usually be the center points [22–25]; special points like knee points and extreme points [26,27]; and clustering centers of the POS [28]. After that, with the development and popularity of statistical machine learning, it gradually becomes the mainstream that utilizes historical information, such as the previous optimal solution sequence or the center point of the POS, to train the prediction model. Muruganantham et al. [18] proposed two variants of the Kalman filter (KF) predictor, which also performed the first-order and second-order difference prediction. In [19], the support vector regression (SVR) was employed to conduct nonlinear prediction. FPS proposed by Hatzakis and Wallace [29] and PPS proposed by Zhou et al. [17] both adopted autoregressive (AR) models. A mixture-of-experts-based ensemble framework [30] was introduced to conduct multiple predictors in a single run.

Generally, diversity-based methods use mutation of selected old solutions or random generation of some new solutions to improve diversity (such as [15], [31]), maintain diversity throughout the optimization process (such as [32]), or employ multipopulation mechanisms (such as [33]). For example, Vavak et al. [34] employed a mutation operator (VLS) to increase the diversity of their algorithm. In a triggered hypermutation method given by Woldesenbet and Yen [35], the mutation rate would increase immediately when an environmental change is detected. Goh and Tan [16] added randomly generated individuals into the competition pool to increase diversity. In [36], Camara et al. used nondominated individuals specifically for selection and mutation and applied crowding mechanisms to maintain diversity. Ruan et al. [37] proposed a stochastic diversity preservation strategy for diversity maintenance.

To summarize, prediction-based methods have been the dominant techniques for change response in solving DMOPs [21]. Diversity-based methods are widely used due to their simple implementation and effectiveness in addressing severe diversity loss. However, they are subject to the limitations that we have mentioned in the previous section. It is desirable to develop more efficient methods for change response in DMOPs.

2.2. Motivation

The high frequency (need to be quickly responded to) and drastic (may cause severe diversity loss) environmental changes sometimes happen during the evolution process, bringing great challenges to DMOEAs. As we all know, prediction-based methods predict the state of the changing environment, typically using various predictors to adapt to changes in advance. Although the prediction-based methods have significant advantages over other methods in accelerating the response (convergence) speed, they tend to perform poorly when the POF or POS of a DMOP changes sharply and/or irregularly. Proper diversity increased by diversity-based methods helps explore promising search regions. However, some existing diversity-based methods either increase excessive diversity, which may cause evolutionary stagnation (e.g., reinitialize randomly), or are only effective for local search (e.g., Gaussian perturbation). As both prediction-based approaches and diversity-based approaches have strengths and weaknesses, a natural question arises: How can we leverage these strengths together to design a better change response mechanism to solve various types of DMOPs? To the end, this paper proposes an enhanced change response mechanism named HPPCM by combining a hybrid prediction strategy with a precision controllable mutation strategy.

In the hybrid prediction strategy, we redefine the movement step size α instead of predefining it as 1.0 like that in the general first-order linear prediction model. In addition, we find out the most promising decision space as the exploration space. In this way, we can quickly respond to different types of predictable changes by considering the varying degrees of adjacent environmental changes and exploring the most promising decision space. In the precision controllable mutation strategy, a Gaussian distribution with controllable precision is added to $\Delta\alpha$ against the original formula of PCM in [38]. Under the premise of setting the same accuracy r, the PCM in this paper can search for more solutions within the required precision with a more uniform probability, thus improving the population's diversity exploration ability.

3. Proposed algorithm

In this section, we elaborate on the specific HPPCM proposed in this paper. It consists of three sub-strategies, namely the center point-based prediction strategy, the guiding individual-based prediction strategy and the precision controllable mutation strategy. Finally, the overall framework of the algorithm is introduced, which integrates the change response mechanism named HPPCM into a regularity model-based multiobjective estimation of distribution algorithm (RM-MEDA) [20].

3.1. Center point-based prediction strategy

Environmental changes present linear and regularity in many cases, and the current environment may have the same or similar degree of change with the previous last environment. In these cases, we assume that the center points moving trend represents the population's moving trend. In other words, the other members of the population are supposed to have a similar moving trend with the center point. Therefore, the prediction for the moving trend of the center point is vital. At the same time, we should take into consideration that similar degree of change but not totally the same between the current environment and the previous last one. So when an environmental change is detected, we should use the useful information (the center point's moving trend) from the previous environmental change with varying degrees.

In this subsection, we follow the principle that the moving direction between the two consecutive environments is the same, but the moving

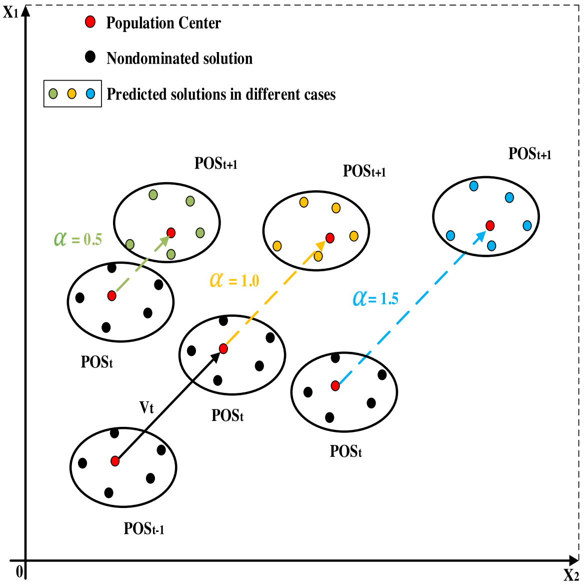


Fig. 2. Schematic diagram of the predicted population generation of population when $\alpha = 0.5$, 1.0 and 1.5.

step size is different due to the degree of change between two consecutive environments to develop the prediction strategy. The center point-based prediction strategy uses two populations, P_t^1 and P_t^2, respectively, to implement prediction with the same moving direction but different moving step sizes.

P_t^1 completely follows the step size of the center point's movement trend in the last environmental change, namely $\alpha = 1$. It means that our algorithm can accurately respond to very similar changes to the previous last environmental change. Because of its simplicity and clarity, this usage is widespread in many existing first-order difference linear prediction models in DMOEAs.

P_t^2 is specifically designed for situations where the current environmental change is similar to but different from the previous last one. Because the similarity between continuous environmental changes has not been accurately measured, the moving step size of P_t^2 along the moving direction is uncertain. The current change is normally smoother or more drastic than the previous last one. The moving step size (α) is therefore defined as a random number between 0.5 (refers to more slightly change) and 1.5 (refers to more drastic change). Fig. 2 represents the schematic diagram of the predicted population generation when α is 0.5, 1.0 and 1.5 respectively in the population search process after the occurrence of new environmental changes.

The process is given in Algorithm 1. First, the moving direction V_t is calculated by the moving direction of the center point from time period $t-1$ to t. Then the algorithm will randomly divide current population P_t into two, named P_t^1 and P_t^2. According to Formula (5), it will generate some solutions to predict the new locations at time period $t+1$ of old solutions (in P_t^1 and P_t^2). Finally, the algorithm will return the combination of the prediction solutions in P_{t+1}^1 and P_{t+1}^2 as P_1. The related definitions and formulas in the process will be introduced next.

Algorithm 1 Center Point-based Prediction.

```
Input: $P_t$ (population at time period $t$), $C_{t-1}$ (center point of $P$ at time
    period $t-1$), $C_t$ (center point of $P$ at time period $t$), $m$ (the number
    of initial individuals), $t$ (time period)
Output: $P_1$
$V_t = C_t - C_{t-1}$;    // refer to Formula 3
$P_t^1, P_t^2 \leftarrow$ divide $P_t$ into two randomly;
$P_{t+1}^1, P_{t+1}^2 \leftarrow$ generate new solutions at time period $t+1$;    // refer to
    Formula 5
$P_1 = P_{t+1}^1 \cup P_{t+1}^2$;
Return $P_1$;
```

The moving direction of center point at time period t (denoted by V_t) is calculated as follows:

$$V_t = C_t - C_{t-1} \tag{3}$$

The formula for generating the predicted solutions at time period $t+1$ is as follows:

$$x_{t+1} = x_t + \alpha V_t, \tag{4}$$

where α represents the moving step size, which means the step size of moving the old solutions of the P_t along the center point's movement direction V_t.

Based on the solution x_t at time period t, the prediction solution x_{t+1} at time period $t+1$ is generated as follows:

$$\begin{cases} x_{t+1} = x_t + V_t & x_t \in P_t^1 \\ x_{t+1} = x_t + \text{ random } (0.5, 1.5) \times V_t & x_t \in P_t^2 \end{cases} \tag{5}$$

3.2. Guiding individual-based prediction strategy

Some environmental changes of DMOPs are predictable but not linear (such as periodic). At this time, the prediction for the evolutionary direction of the population in the new environment is significant. If it is still assumed that the current population is moving in the same direction as the last time period like that in the first-order difference linear prediction model, it is likely to cause the prediction to fail. Therefore, how to accurately predict the evolutionary direction is the first key point. To this end, we let the current population spontaneously evolve some generations (Δt) when an environmental change is detected. Then the center point's moving direction of the nondominated solutions will serve as a rough reference direction for future predictions.

The main idea of the guiding individual-based prediction strategy is as shown in Algorithm 2. First it will retrieve the nondominated solutions at time period t (POS_t) and the nondominated solutions after evolving for Δt generations ($POS_{t+\Delta t}$), thus calculating the center point of POS_t and $POS_{t+\Delta t}$ separately. Then a center point's moving direction of the nondominated solutions will be obtained by the vector difference from C_t to $C_{t+\Delta t}$. The evolutionary direction (dir) for further prediction can be obtained according to Formula (6), also can be find in lines from 2 to 11 of Algorithm 2. Then m initial individuals with the largest

Algorithm 2 Guiding Individual-based Prediction.

```
Input: $P_t$ (population at time period $t$), $m$ (the number of initial indi-
    viduals), $t$ (time period)
Output: $P_2$
1:  $Iter$ (iteration counter) $\leftarrow$ 0;
2:  $POS_t \leftarrow$ retrive the nodominated solutions at time period $t$;
3:  $C_t \leftarrow$ calulate the center point of $POS_t$;
4:  while $Iter \leq \Delta t$ do
5:      $P_{off} \leftarrow$ GetOffspring ($P_{t+Iter}$);
6:      $P_{t+Iter} \leftarrow$ EnvironmentalSelection ($P_{t+Iter} \cup P_{off}$);
        // use the environmental selection operator of RM-MEDA
7:      $Iter = Iter + 1$;
8:  end while
9:  $POS_{t+\Delta t} \leftarrow$ retrive the nodominated solutions in $P_{t+\Delta t}$;
10: $C_{t+\Delta t} \leftarrow$ calulate center point of $POS_{t+\Delta t}$;
11: $dir$ (evolutionary direction) $\leftarrow C_{t+\Delta t} - C_t$;
12: $Pop_{init}$ (initial individuals) $\leftarrow$ select $m$ solutions from $POS_{t+\Delta t}$ with
    the largest crowding distance;
13: $Pop_{guide}$ (guiding individuals) $\leftarrow$ generate $N - m$ new solutions;    //
    refer to Formula 7
14: $P_2 = Pop_{init} \cup Pop_{guide}$;
15: Return $P_2$;
```

crowding distance will be selected after sorting POS_t by crowding distance in ascending order. A bunch of guiding individuals are generated uniformly along the most promising decision space composed of current extreme points and boundary points according to the judgment of dir's direction. These guiding individuals are stored to P_2, and the algorithm returns P_2.

Assume that $POS_t = \{x_t\} = \{x_t^1, x_t^2, \ldots, x_t^n\}$ represents all the nondominated individuals at time t and C_t is the center point of the nondominated solution set at time period t. $|POS_t|$ is the cardinality of all nondominated individuals at time t. $i = 1, 2, \ldots, n$ represents the number of dimension of the decision variable. The definition of relevant variables in Algorithm 2 are as follows:

$$dir = (dir_1, dir_2, \ldots, dir_i, \ldots, dir_n), \quad i = 1, 2, 3, \ldots, n, \tag{6}$$

where

$$dir_i = C^i_{t+\Delta t} - C^i_t,$$
$$C^i_t = \frac{\sum_{x_t \in POS_t} x^i_t}{|POS_t|},$$

where dir represents evolutionary direction vector of population in decision space. $dir_i \quad i = 1, 2, \ldots, n)$ represents the ith dimension decision variable's direction vector. C_t is the population center pointat time period t, $C_{t+\Delta t}$ is the population center point after population evolving spontaneously for Δt generations.

The selection of the initial individual and the determination of the evolutionary direction dir are all to generate some influential guiding individuals. According to dir, every time the environment changes, our algorithm will predict a set of guiding individuals close to the new POS as possible based on m initial individuals. The way we generate guiding individuals is not to randomly sprinkle points without purpose. Instead, we obtain the current population's extreme points (i.e., maximum point and minimum point). Two vectors are connected from current extreme points to the corresponding boundary points (i.e., the maximum point of the boundary and the minimum point of the boundary). Specifically, the current minimum point is connected to the maximum point of the boundary, and the current maximum point is connected to the minimum point of the boundary. We select the boundary points and the current extreme point to search the most potential (i.e., promising) decision space as thoroughly as possible.

When a certain dimensional direction vector dir_i is positive, the vector D_i^+ formed by the ith dimension of minimum point and the ith dimension of maximum boundary point of the current population is equally divided into $\frac{N}{m}$ parts. Each initial individual generates $\frac{N}{m} - 1$ level individuals uniformly along the vector D_i^+. Otherwise, the vector D_i^- composed of the ith dimension of maximum point and the ith dimension of minimum boundary point of the current population is divided into $\frac{N}{m}$ parts. The m initial individuals uniformly generate $\frac{N}{m} - 1$ layers of individuals along the vector D_i^-. Finally, the $N - m$ guiding individuals combined with m initial individuals are nondominated sorted. Fig. 3 shows the process of the operator evenly dividing the decision space and generating guiding individuals when dir is positive.

The specific methods of generating $N - m$ guiding individuals are as follows:

$$y^i_{j\times k} = \begin{cases} x^i_j + \dfrac{UB_i - low^i_t}{\frac{N}{m}} \times k & \text{if } dir_i > 0 \\ x^i_j - \dfrac{high^i_t - LB_i}{\frac{N}{m}} \times k & \text{if } dir_i < 0 \end{cases}, \tag{7}$$

where

$$i = 1, 2, \ldots, n,$$
$$j = 1, 2, \ldots, m,$$
$$k = 1, 2, \ldots, \frac{N}{m} - 1,$$

where $y^i_{j\times k}$ means the ith dimension of the $j \times k$th solution. x^i_j means the ith dimension of the jth solution. UB_i is the largest and LB_i is the smallest boundary of the ith dimension. $high^i_t$ and low^i_t are the ith dimension values of the current maximum point and minimum point in the decision space at t time period. n is the dimension of the decision variable, m is the number of initial individuals, and N is the population size. dir_i represents the ith dimension decision variable's direction vector.

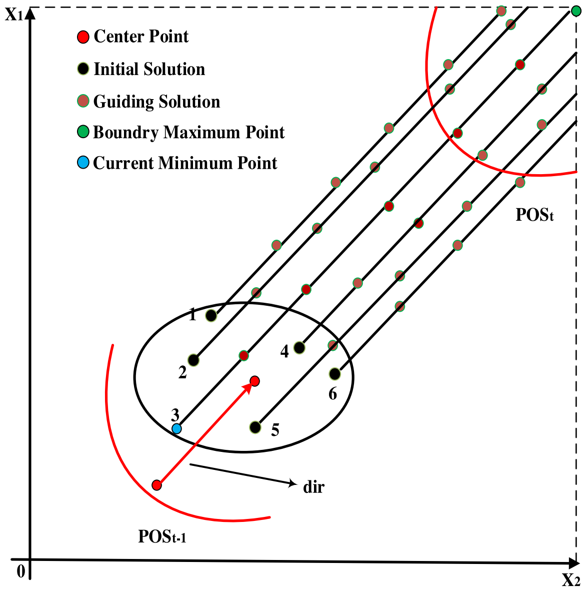


Fig. 3. When evolutionary direction $dir > 0$, the schematic diagram of the process of generating guiding individuals.

3.3. Precision controllable mutation strategy

In the process of tracking new POF or POS, the imbalance between convergence and diversity has increased. Especially when the environmental changes are unpredictable, the diversity-based strategy is widely used in DMOEAs. However, some existing methods like reinitialization randomly will increase excessive diversity, which may cause evolutionary stagnation. Other methods like Gaussian perturbation where Gaussian distribution is similar to the simulated binary crossover (SBX) [39] in that it generates offspring near the parent value with a high probability, so it is only effective for local search. Hence, it's desirable to design a more efficient diversity-based strategy.

This paper refers to the precision controllable mutation operator by Zhang et al. [38] and proposes another version of precision controllable mutation (PCM). The PCM operator formula in [38] is as follows:

$$x^i_{t+1} = x^i_t + \Delta\alpha, \tag{8}$$

$$x^i_{t+1} = x^i_t - \Delta\alpha, \tag{9}$$

where

$$\Delta\alpha = \frac{1}{10^{\text{Random }(r)}} \times \text{Random }(9)$$

Take $r = 2$ and $x = 3$ as an example, the mutation precision $\frac{1}{10^{\text{Random }(r)}} = \{0.1, 0.01\}$. Random (9) can be regarded as a random integer coefficient from 1 to 9. $\Delta\alpha$ could be $\{0.1, 0.2, \ldots, 0.9, 0.01, 0.02, \ldots, 0.09\}$, the value of x maybe $\{2.1, 2.2, \ldots, 2.9, 2.91, 2.92 \ldots, 2.99, 3.01, 3.02, \ldots 3.09, 3.1, 3.2, \ldots, 3.9\}$. Obviously, Formula (8) and (9) can generate all the neighboring solutions of 3.0 within the precision of 0.01 and 0.1.

The mutation operator formula of another version of PCM proposed in this paper is as follows:

$$x^i_{t+1} = x^i_t + \Delta\beta, \tag{10}$$

$$x^i_{t+1} = x^i_t - \Delta\beta, \tag{11}$$

where

$$\Delta\beta = \Delta\alpha + \frac{\text{Gaussian}(0, std)}{10^{\text{Random }(r)}},$$
$$std = \sqrt{\frac{\sum_{i=1}^{y}(x_i - \bar{x})^2}{y-1}},$$
$$\bar{x} = \frac{\sum_{i=1}^{y} x_i}{y},$$

where x_t^i is the ith dimension decision variable at time period t, and x_{t+1}^i is the new solution of the ith dimension decision variable mutated by x_t^i. $i = 1, 2, \dots, n$ represents the number of dimension of the decision variable. The Random(r) function randomly generates an integer from 1 to r. std is the variance of each dimension decision variable x_i, $\bar{x}$ is the mean value of the decision variable x, and y is the number of decision variables.

$\Delta\alpha$ and $\Delta\beta$ are used to control the precision of the variation to ensure that the range of the mutation is diverse. If the required precision needs to reach 0.0001, r can be set to 4. Random (r) might be 1, 2, 3 and 4. $\frac{1}{10^{\text{Random }(r)}} = \{0.1, 0.01, 0.001, 0.0001\}$. When the accuracy is 0.001 and smaller, the population can search for the optimal solution locally in the decision space with small steps. When the required precision is 0.1, the population can jump out of the local optimum and search in the global decision space with large strides.

From Algorithm 3 we can see that PCM in this paper is simple and easy to implement. First, it will randomly initialize an integer p between 1 to 2. If p is 1, the mutation operator will follow Formula (10), otherwise it will be implemented according to Formula (11). Then those mutated solutions at time period $t+1$ (P_{t+1}) from the old solutions at time period t (P_t) are stored to P_3. Finally, the algorithm returns P_3.

Take $r = 2$ and $x = 10$ as an example, let the original solution x mutate 10,000 times according to the Formulas (8) and (9) and Formulas (10) and (11), respectively. Fig. 4(a) and (b) show the frequency diagram of the mutation solution distribution with PCM in [38] and PCM in this paper, respectively. By comparing the two figures, we find that taking the variation solution of $x = 10.4$ as an example, the frequency in Fig. 4(a) has reached almost 600, and there is no value larger or smaller than 10.4 within the precision 0.01. However, in Fig. 4(b), the frequency of 10.4 is no more than 8, while the frequency of values around it within the precision 0.01 (e.g., 10.38 and 10.46) presents a Gaussian distribution. It means the mutation solutions (obtained by PCM in this paper) in Fig. 4(b) have better diversity and more uniform distribution. Under the premise of setting the same precision r, the mutation solution generated by PCM in this paper can search for the finer solution with a more uniform probability. It's because PCM in this paper adds precision controllable Gaussian distribution based on the original PCM in [38]. In the process of algorithm execution, the decision space under each required precision can be fully searched. In this way, the diversity exploration of the evolutionary population can be enhanced.

Algorithm 3 Precision Controllable Mutation.

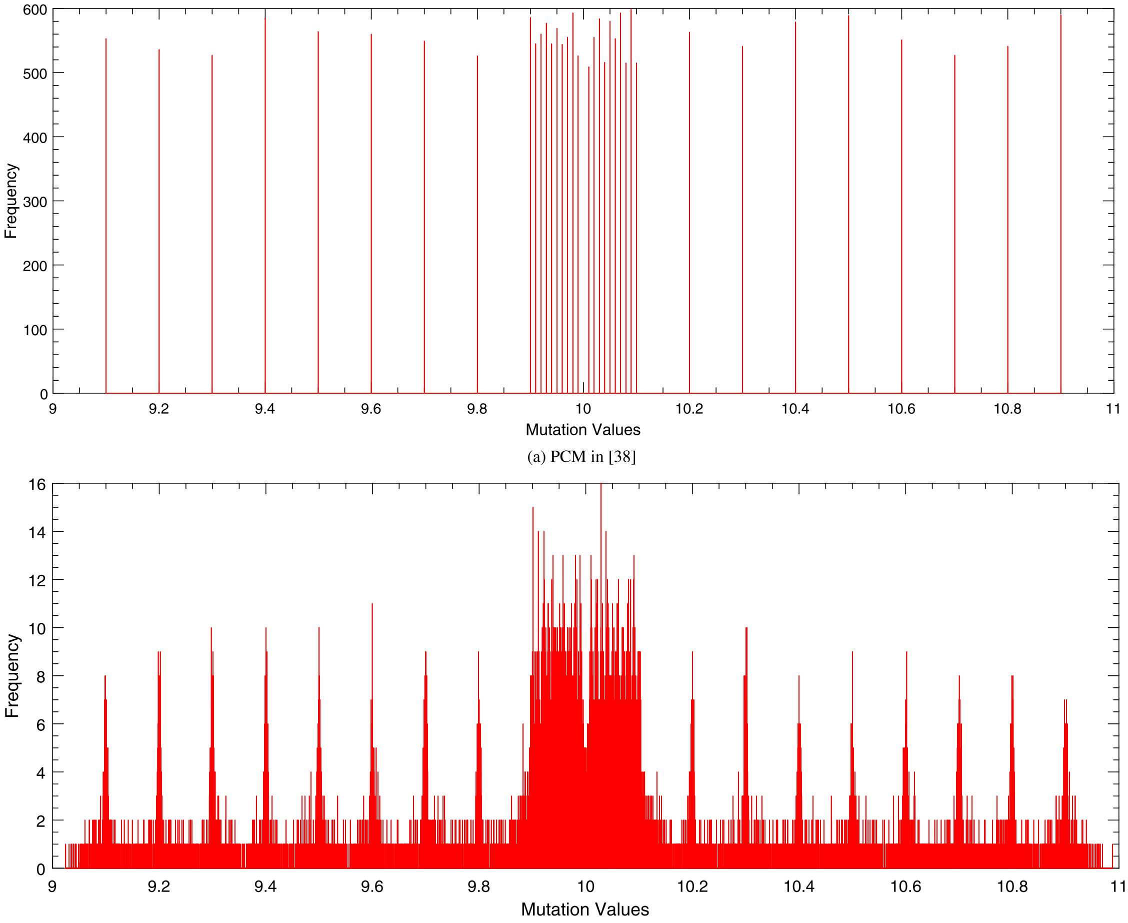
```
Input: $P_t$ (population at time period $t$), $t$ (time period)
Output: $P_3$
$p = Random(2)$;
$\Delta\alpha = \frac{1}{10^{\text{Random }(r)}} \times$ Random (9);
$\Delta\beta = \Delta\alpha + \frac{\text{Gaussian }(0, std)}{10^{\text{Random }(r)}}$;
if $p == 1$ then
    $P_{t+1} = P_t + \Delta\beta$;
else
    $P_{t+1} = P_t - \Delta\beta$;
end if
$P_3 \leftarrow P_{t+1}$;
Return $P_3$;
```

3.4. Algorithm framework

The algorithm combines HPPCM proposed in this paper (detailed in Sections 3.1, 3.2 and 3.3) with RM-MEDA to optimize DMOPs. RM-MEDA is adopted as the evolutionary optimizer in Algorithm 4, while other evolutionary solvers can also be employed. It models a promising area in the search space by a probability model whose center point is a piecewise continuous manifold [20]. The Local PCA [40] algorithm was employed for building such a model. New candidate solutions are sampled from the model thus built. According to Algorithm 4, the algorithm starts with randomly generating N initial solutions with a uniform distribution (line 1 of Algorithm 4). In each evolution generation $Iter$, $Iter$= 1, 2,...,τ_t of each time period t, t = 1, 2,..., the executive process are as follows. On the one hand, if an environmental change is detected, the center point C_t of the population at period t is calculated, and the change response mechanism HPPCM is triggered. The originality of this paper exactly lies in the change response mechanism HPPCM which consists three sub-strategies (lines 6–8 of Algorithm 4), where three subpopulations obtained by them P_1, P_2, P_3 are generated respectively by Algorithms 1–3. The key components of HPPCM are described in detail in the previous subsections. A merged population P_{achive} consists of the three subpopulations P_1, P_2 and P_3 after boundary correction operation. Then we select N solutions from the combination of P_{achive} and P by the environmental selection operators in RM-MEDA (line 10 of Algorithm 4), which is based on the nondominated sorting of NSGA-II [41] called NDS-Selection. On the other hand, if no environmental change is detected, it uses reproduction operators of RM-MEDA (i.e., modeling and sampling) to generate the offspring population P_{off} (line 12 of Algorithm 4). Then the algorithm will use the environment selection operator named NDS-Selection mentioned above to select N solutions between P and its offspring population P_{off} (line 13 of the

Algorithm 4 The overall framework of the algorithm.

```
Input: $N$ (population size), $\tau_t$ (the number of iterations where $t$ remainsthe same), $n_t$ (the severity of change), the stopping criterion
Output: $P$
1:  Initialize a population $P$, the discrete time parameter $t \leftarrow 0$, iteration counter $Iter \leftarrow 0$;
2:  while the stopping criterion is not met do
3:      if change is detected then
4:          $t = t + 1$;
5:          $C_t \leftarrow$ calulate the center point of $P$ at time period $t$;
            // the sub-strategies in HPPCM (change response mechanism) are the center point-based prediction strategy (line 6), the guiding individual-based prediction strategy (line 7) and the precision controllable mutation strategy (line 8)
6:          $P_1 \leftarrow$ Center Point-based Prediction $(P';, dir, C_t, t)$;   // refer to Algorithm~1
7:          $P_2 \leftarrow$ Guiding Individual-based Prediction $(P, C_t, V_t, t)$;   // refer to Algorithm~A2
8:          $P_3 \leftarrow$ Precision Controllable Mutation $(P, t)$;   // refer to Algorithm~A3
9:          $P_{achive} \leftarrow$ BoundaryCorrection $(P_1 \cup P_2 \cup P_3)$;
10:         $P \leftarrow$ EnvironmentalSelection $(P_{achive} \cup P)$;   // use the environmental selection operator of RM-MEDA
11:     else
12:         $P_{off} \leftarrow$ GetOffspring $(P)$;   // use the reproduction operator of RM-MEDA
13:         $P \leftarrow$ EnvironmentalSelection $(P \cup P_{off})$;   // perform environment selection operation of RM-MEDA
14:     end if
15:     $Iter = Iter + 1$;
16: end while
17: Return $P$;
```



(a) PCM in [38]

(b) PCM in this paper

Fig. 4. Take $r = 2$ and $x = 10$ as an example, let the original solution x mutate 10,000 times, the frequency diagram of the mutation solutions obtained by two versions of PCM.

Algorithm 4). Finally, the N solutions are stored to P. The algorithm will terminate when the stopping criterion is met and return P.

As for change detection(line 3 of Algorithm 4), the algorithms used in this paper uniformly randomly select 5% of the individuals in the current population for re-evaluation. If the objective value changes, it proves that the current generation environment has changed.

It is worth noting that the boundary correction operator (line 10 of Algorithm 4) is applied to check whether the generated solution is within the given boundary of the decision space and if not, it will be corrected. The procedure of the boundary correction is described in detail as follows:

$$x^i_{t+1} = \begin{cases} y & \text{if } LB_i \le y \le UB_i \\ 0.5 \times \left(x^i_t + LB_i\right) & \text{if } y < LB_i \\ 0.5 \times \left(x^i_t + UB_i\right) & \text{if } y > UB_i \end{cases}, \tag{12}$$

where x^i_{t+1} is the corrected solution, and y is the new solution generated by the change response mechanism in the algorithm proposed. x^i_t is the value of the decision variable of the ith dimension at time t, where UB_i is the maximum boundary of the ith dimension, and LB_i is the minimum boundary of the ith dimension.

4. Experimental setup

4.1. Test instances

Eighteen test instances are used in our proposed algorithm to compare with other algorithms. It contains three dMOP issues (dMOP1-dMOP3) [16], four FDA issues (FDA1-FDA4) [1], five ZJZ issues (F5-F9) [31] and six JY issues (JY1-JY6) [6]. The modified hypervolume difference (MHVD) and the modified inverted generational distance (MIGD) are used to test the performance of HPPCM and other popular algorithms. FDA test suit origins from ZDT [42] and DTLZ [43], while dMOP [1] and ZJZ [31] test suit origins from FDA [1]. The problem type of FDA and dMOP include Type I, II and III, while the ZJZ test suite only belongs to Type II, where there's no linkage between their variables. There exists nonlinear, time-varying and monotonic linkage between variables of the ZJZ test suite. JY test suite includes mixed type in addition to Type I, II and III, where there's nonlinear, time-varying and nonmonotonic linkage between variables. Among them, JY6 is a time-varying multimodal problem, where not only does the number of local optima change over time but also the POS is dynamically shifted.

4.2. Comparison algorithms

Five popular algorithms are used for comparison in this paper. They are DNSGA-II [15], PPS [17], PMS [44], PBDMO [11], and DVA [45] respectively, with different dynamic change response mechanisms. Among them, the DVA method is our previous work about solving DMOPs. It is worth reminding everyone that the work in this article is not an improved version based on the research of the DVA method. They are two quite different response mechanisms for solving DMOPs. The overview of DVA and the specific difference between DVA and HPPCM will be mentioned in the last two paragraphs of this section.

NSGA-II [41] is a classical algorithm for solving multi-objective problems. There are two versions of DNSGA-II: NSGA-II-A and NSGAII-B. In the DNSGA-II-A, the population is replaced by some individuals with new randomly created solutions, while in the DNSGA-II-B, diversity is maintained by replacing a percentage of the population with mutated solutions.

PPS is representative of a prediction-based method. It models the trajectory of POF or POS in a dynamic environment and then uses the model to predict the new position of POS in PPS. The POS history information is divided into two parts, which are population center and manifold. When historical information is sufficient in the later stages of environmental changes, PPS has apparent advantages.

The overall framework of PMS is similar to our algorithm. It also uses RM-MEDA to optimize the population. But the change response mechanisms between it and our algorithm are quite different, especially in exploitation operators and diversity-based strategies.

Whenever a change is detected, the PBDMO proposed by Zhang et al. generated three subpopulations to reinitialize the population effectively. The three subpopulations were generated according to prediction, sampling and shrinking strategies.

The DVA method first analyzes the influence of each decision variable on individuals to distinguish between distribution-related decision variables and convergence-related decision variables in the new environment. Then for distribution-related decision variables, a hybrid prediction strategy combining Latin hypercube sampling [46] and random sampling is adopted to enhance the exploration ability of the population in the new environment. Otherwise, for convergence-related decision variables, the central point prediction method is used to accelerate the convergence speed of the population in the new environment. After the analysis of decision variables and the different prediction strategies based on the analysis, several solutions are generated and merged into a new population. Finally, an angle-based adaptive selection method is applied to the new merged population. Therefore, the DVA method can obtain the solutions with good convergence and diversity to make the initial population more adaptable to the new environment.

There is some obvious difference between DVA and HPPCM as follows. Firstly, DVA relies on the decision variable analysis method and executes different prediction strategies for different decision variables according to whether the decision variables are related to convergence or diversity. However, HPPCM does not distinguish the differences of decision variables and uses the same response strategy for each dimension of an individual's decision variables. Secondly, DVA generates the merged population based on a hybrid strategy combining Latin hypercube sampling and random sampling and the central point prediction method. However, HPPCM combines a hybrid prediction strategy and a precision controllable mutation strategy to generate a merged population. Finally, DVA proposes an angle-based adaptive selection method while HPPCM uses the classic NDS-Selection method to select better convergence and distribution solutions from the merged population.

4.3. Performance metrics

There are two typical comprehensive indicators for judging MOEAs in the field of multi-objective optimization, hypervolume difference (HVD) [22] and Inverted Generational Distance (IGD) [47,48]. These two indicators are so typical that they can evaluate both the convergence and the distribution of an algorithm. Unfortunately, the particularity of the DMO makes the original IGD and HVD inappropriate as an evaluation indicator. So we adopt the modified hypervolume difference (MHVD) and the modified inverted generational distance (MIGD).

HVD measures the gap between the hypervolume of the approximate POF obtained by the algorithm and the hypervolume of the true POF. The definition of this indicator is described as:

$$HVD\left(POF_t, P_t\right) = HV\left(POF_t\right) - HV\left(P_t\right), \tag{13}$$

where HV(s) is the hypervolume of the set s. MHVD is defined as the average value of HVD for some time steps in an environmental change under one operation. The formula to determine this indicator is calculated as:

$$MHVD = \frac{\sum_{t\in T} HVD\left(POF_t, P_t\right)}{|T|}, \tag{14}$$

where T is some time periods under one run. $|T|$ is the cardinality of T. The reference point is $\left(z_1+0.5, z_2+0.5, \ldots, z_M+0.5\right)$. Among them, z_j is the maximum value of the jth dimension objective value of the true POF, and M is the objective dimension. Like HVD, MHVD evaluates the convergence and distribution of the algorithm at the same time. The larger the MHVD value, the better the algorithm performance.

Zhou et al. [17] modified the IGD metric and adopted the average of the IGD values in some time steps over a run as a performance indicator for DMOEA, which is computed as:

$$IGD\left(POF_t, P_t\right) = \frac{\sum_{v\in POF_t}^{|POF_t|} d\left(v, P_t\right)}{\left|POF_t\right|}, \tag{15}$$

where

$$d\left(v, P_t\right) = \min_{u\in P_t} \sqrt{\sum_{j=1}^{n} \left(f_j^v - f_j^u\right)^2},$$

where n is the dimensionality of the decision variable. d(v, P_t) is the minimum Euclidean distance between v and P_t. POF_t is a set of Pareto optimal solutions obtained by a uniform sampling at time t. $\left|POF_t\right|$ is the cardinality of POF_t. P_t is a set of solutions approximate to POF_t obtained by the algorithm.

MIGD is defined as the average value of IGD at some time steps in an environmental change under one operation. This index can comprehensively evaluate algorithm performance like IGD, including evaluation of convergence and distribution. The specific calculation formula is as follows:

$$MIGD = \frac{\sum_{t\in T} IGD\left(POF_t, P_t\right)}{|T|}, \tag{16}$$

where T is some time periods under one run. $|T|$ is the cardinality of T. Like IGD, MIGD evaluates the convergence and distribution of the algorithm at the same time. The smaller the MIGD value, the better the algorithm performance.

4.4. General parameter settings

The parameter settings for the test instances and comparison algorithms are as follows. RM-MEDA was adopted as the evolutionary optimizer of all comparison algorithms. In addition, the change detection method in all comparison algorithms was the same as that used in our work. That is, 5% of the population was randomly selected and re-evaluated to detect environmental changes. We fixed the change detection method and the static evolutionary optimizer to compare the performance of the response mechanism of the used algorithm more fairly and objectively.

General parameter settings in all test instances: Except for FDA4 and F8, which are tri-objective problems, all other test cases are bi-objective problems. We set the population capacity $N = 100$ for all test problems. Decision variable dimension D was set to 20. We fixed the environmental change frequency τ_t to 30 and set n_t to 5, 10 and 20, representing severe, gentle and slight environmental changes, respectively. Each algorithm ran 20 times independently for each test instance. Every simulation experiment was carried out for 3600 generations, and 120 environmental changes were tracked.

Specific parameter settings in each algorithm: All the parameters in the compared algorithms used the same settings as in their original studies.

(1) Parameters in PPS: The order in AR(p) model was $p = 3$, the number of cluster was set 5 and the maximum length of history center point sequence was $M = 23$.

Table 2
MIGD indicators of six algorithms on dMOP and FDA test suites.

Problem	n_t	DNSGA-II	PPS	PMS	PBDMO	DVA	HPPCM
dMOP1	5	1.4481e-4 (1.23e-4)	7.6059e-3 (4.75e-3)	1.2708e-4 (3.95e-6)	1.2577e-4 (1.51e-5)	**1.2570e-4 (6.35e-6)**	**4.5702e-5 (1.32e-6)**
	10	1.4017e-4 (1.74e-4)	9.2806e-3 (4.10e-3)	1.2102e-4 (3.39e-6)	1.2534e-4 (1.24e-5)	**1.2043e-4 (5.66e-6)**	**4.6014e-5 (6.60e-7)**
	20	1.1196e-4 (1.06e-4)	7.2721e-3 (3.91e-3)	**1.1760e-4 (2.55e-6)**	1.3784e-4 (3.43e-5)	1.1870e-4 (8.05e-6)	**4.6434e-5 (9.91e-7)**
dMOP2	5	3.4842e-1 (9.39e-3)	**6.5130e-3 (9.64e-4)**	2.9998e-2 (1.90e-2)	2.2609e-1 (2.03e-2)	1.0615e-2 (4.34e-3)	1.1559e-2 (6.18e-3)
	10	1.1914e-2 (1.71e-4)	**1.0595e-3 (7.28e-5)**	2.9201e-3 (1.37e-3) -	7.8626e-2 (2.78e-2)	3.2910e-3 (2.48e-4)	**4.7595e-4 (6.81e-5)**
	20	1.4316e-3 (2.36e-5)	**2.1654e-4 (7.37e-6)**	6.2255e-4 (1.34e-5)	1.0809e-3 (2.85e-4)	5.7252e-4 (2.53e-5)	**8.8123e-5 (4.57e-6)**
dMOP3	5	1.6267e-1 (4.76e-3)	1.7455e-1 (1.53e-2)	9.7512e-2 (1.93e-2)	**8.4759e-2 (2.11e-3)**	1.5699e-1 (3.36e-2)	**3.6396e-2 (2.59e-3)**
	10	1.0246e-1 (4.77e-3)	9.2050e-2 (5.82e-3)	4.1971e-2 (3.39e-3)	7.5929e-2 (5.82e-3)	**6.7068e-2 (6.89e-3)**	**1.4950e-2 (1.04e-3)**
	20	9.4839e-2 (3.58e-3)	9.9646e-2 (8.57e-3)	4.5317e-2 (3.70e-3)	**6.0027e-2 (7.01e-3)**	6.0392e-2 (5.82e-3)	**1.3868e-2 (1.59e-3)**
FDA1	5	1.8949e+0 (1.27e-1)	**1.1900e-2 (1.32e-3)**	9.7872e-2 (7.48e-2)	3.1189e-1 (7.26e-2)	2.2179e-2 (1.37e-3)	**2.1367e-2 (4.16e-3)**
	10	4.5404e-2 (1.10e-3)	**4.5348e-3 (8.12e-3)**	5.4057e-3 (3.13e-4)	1.4078e-1 (1.59e-2)	5.2743e-3 (8.57e-4)	**1.3117e-3 (1.00e-4)**
	20	4.6191e-3 (4.73e-5)	4.9111e-4 (1.62e-4)	**1.3046e-3 (6.52e-5)**	2.1500e-3 (4.92e-4)	1.3428e-3 (2.24e-4)	**1.6656e-4 (1.95e-5)**
FDA2	5	**7.7448e-5 (2.88e-6)**	1.1946e-3 (8.28e-4)	2.2012e-4 (1.01e-4)	2.7756e-4 (9.12e-5)	2.9454e-4 (1.18e-4)	**9.9827e-5 (3.57e-5)**
	10	**8.2572e-5 (7.82e-6)**	4.8446e-4 (2.77e-4)	1.9875e-4 (8.99e-5)	2.8255e-4 (8.47e-5)	2.8371e-4 (9.08e-5)	**7.4949e-5 (1.78e-5)**
	20	**8.0628e-5 (2.42e-6)**	4.6180e-4 (1.79e-4)	1.7148e-4 (4.56e-5)	2.5810e-4 (7.09e-5)	1.8475e-4 (1.62e-5)	**7.8743e-5 (1.81e-5)**
FDA3	5	6.6636e-1 (2.07e-2)	4.0000e+0 (6.75e-1)	8.8286e-1 (2.17e-1)	**8.0350e-2 (4.84e-3)**	2.6494e-1 (1.00e-1)	**3.0841e-2 (4.35e-2)**
	10	4.9520e-2 (2.14e-3)	6.6532e-2 (1.37e-2)	1.0524e-2 (1.33e-3)	1.6213e-2 (5.68e-3)	**8.7369e-3 (4.13e-3)**	**3.2073e-3 (7.67e-4)**
	20	4.1130e-3 (5.02e-4)	2.4677e-3 (4.72e-4)	1.3149e-3 (2.84e-4)	1.4546e-3 (7.66e-4)	**7.6634e-4 (1.61e-4)**	**2.8730e-4 (2.53e-5)**
FDA4	5	1.8687e+1 (6.11e-2)	2.6033e+1 (5.02e-1)	1.8526e+1 (4.87e-2)	**1.4222e+1 (2.36e-1)**	**1.3973e+1 (9.46e-2)**	1.8237e+1 (2.00e-2)
	10	**1.9080e+1 (4.73e-3)**	2.2298e+1 (1.82e-1)	1.9530e+1 (3.34e-2)	2.1217e+1 (9.73e-2)	2.1287e+1 (1.19e-1)	**1.9376e+1 (2.68e-2)**
	20	1.2690e+1 (5.38e-3)	1.4173e+1 (4.19e-2)	1.2977e+1 (2.87e-2)	**1.3710e-2 (1.34e-3)**	**1.3729e-2 (1.00e-3)**	1.2907e+1 (2.56e-2)

(2) Parameters in PMS: The number of guide individual in exploration operator was $Obsize = 10$, the number of initial individual was $OP = 10$, the number of aliquots was $OB = 9$, and the time of evolving independently was $\delta t = 2$. The number of selected optimal individuals from memory pool was $Msize = 5$ (three objectives: 10).

(3) Parameters in PBDMO: The maximum memory size of the two archive sets D and C was set to 50. The threshold ϵ introduced in Algorithm 2 of PBDMO was empirically set to 10^{-4}.

(4) Parameters in DVA: $N2 = 0.8N$, $N3 = 0.2N$ in the prediction strategy and $\delta = 0.9998$ in the adaptive selection strategy.

(5) Parameters in this paper(HPPCM): The algebra Δt of autonomous evolution in Section 3.3 is set to 2; the parameter r in Section 3.4 is set to 2.

5. Experimental results and analysis

Under parameter configurations described in the previous subsection, HPPCM is evaluated on four test suites summed up to eighteen test instances, including dMOP1-dMOP3, FDA1-FDA4, F5-F9 and JY1-JY6. The algorithm is independently run 20 times and terminated on each test instance when the number of function evaluations reaches 360,000 for each run. The statistics (the mean and standard deviation, where standard deviation displays in brackets) of each metric (MIGD and MHVD) over 20 runs compared with other state-of-the-art dynamic multiobjective algorithms are computed and listed in Tables 2 - 7. Those compete algorithms include DNSGA-II [15], PPS [17], PMS [44], PBDMO [11], and DVA [45] respectively. To intuitively compare the performance of HPPCM and its competitors, the evolution curve of the MIGD value and approximate POF under the first 30 time periods are plotted in Figs. 5–10, with $(n_t, \tau_t) = (10,30)$. Table 8 represents the results of the competition between HPPCM and two outstanding multi-modal MOEAs. Discussion on the three components of the proposed algorithm HPPCM are shown in Table 9 and analyzed in Section 5.6. To make the comparison results in Tables 2–9 much clearer and obvious, the best results for the performance of each test instance are bolded, and the background color is filled with gray, while the second-best ones are only bolded. To scientifically assess each algorithm, the multiple comparisons test(Friedman test in this paper) combined with a set of post hoc analysis tests(Holm, Hochberg, and Bonferroni adjustment used in this paper) is applied to the obtained results in Tables 2–9. In the end, the average rankings of each method on all test instances and the unadjusted/adjusted p-values obtained by the application of post hoc procedures are listed in Tables 10–12.

5.1. Results on dMOPs and FDAs

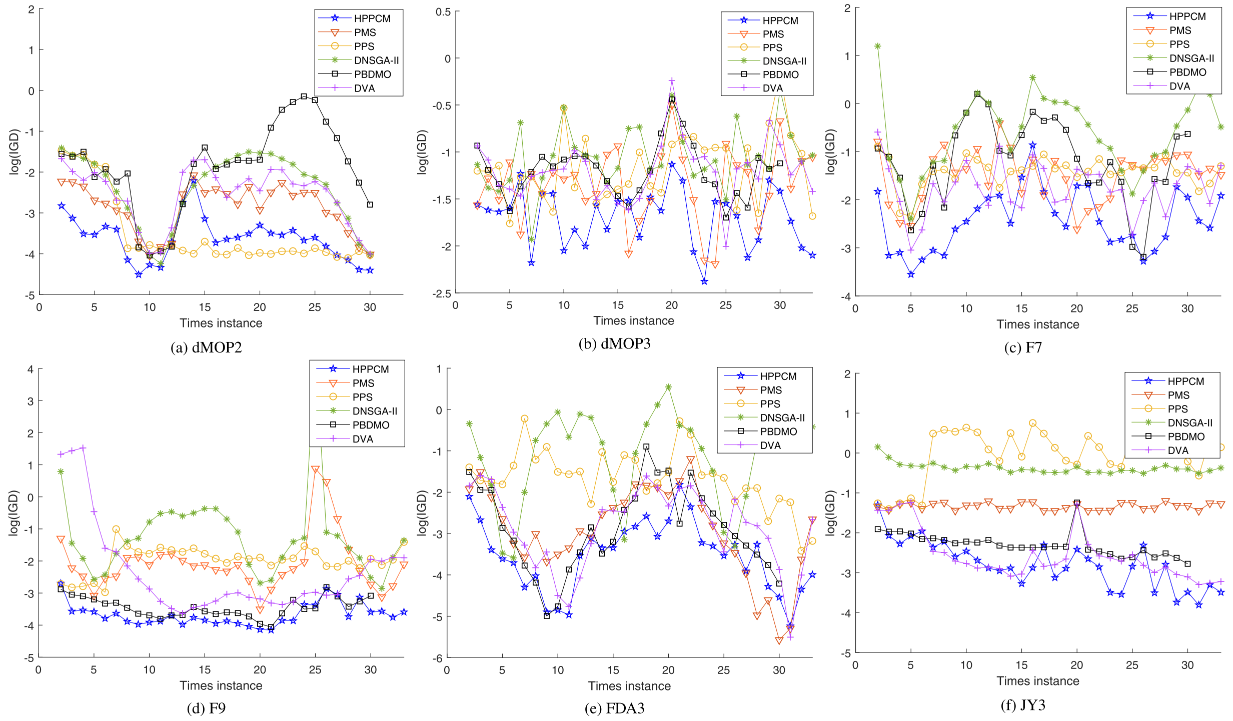
It is shown by Tables 2 and 3 that HPPCM is very competitive compared to DNSGA-II, PPS, PMS, PBDMO and DVA, concerning the number of the cases where it obtains the best and the second-best results. dMOP1-dMOP3, FDA1- FDA4 sums up to 7 test instances. Each problem corresponds to 3 cases where the value of n_t differs, so there are 21 cases altogether for each algorithm. Among 21 cases for each algorithm presented in Table 2, HPPCM obtains the best MIGD results in 13 cases and the second-best ones in 5 cases. Table 3 shows that HPPCM obtains the best MHVD results in 16 cases and the second-best in 3 cases. We know that MIGD and MHVD distinguish algorithms more reliably in terms of overall performance. Despite the changes in n_t, other algorithms lost to HPPCM on these two metrics. It demonstrates that HPPCM can better adapt to the drastic changes in different environments to obtain better distribution and convergence solutions.

There is no connection between the decision variables of dMOP1-dMOP3. The statistics of 20 runs on these three simple test issues show that HPPCM is the best in most cases, and if not, the gap with the best method is extremely small. When $n_t = 5$, PPS has the best performance on dMOP2 and FDA1. PPS accumulates and makes full use of rich historical information in the later stage, which significantly improves prediction accuracy. At this time, PPS reacts very quickly to environmental changes, and the population is evenly distributed after converging to the new POS.

DNSGA-II has the smallest MIGD and biggest MHVD on FDA2. HPPCM proposed in this paper is second only to DNSGA-II, which introduces random individuals in response to change. From the performance of the six algorithms used in this paper on these two indicators, except for the fixed POS problem FDA2, they can accurately approximate the true POF. In addition, HPPCM is better at tracking the POS (or POF) with major changes. On the FDA4 issue of the three objectives, the statistics of each algorithm running 20 times are relatively poor.

5.2. Results on ZJZs

It is shown by Tables 4 and 5 that HPPCM is the most competitive one on the ZJZ test suite (F5-F9) compared to other well-known algorithms because it almost always obtains the best MIGD and MHVD no matter what the value of n_t is large or small. It indicates that the POF obtained



(a) dMOP2

(b) dMOP3

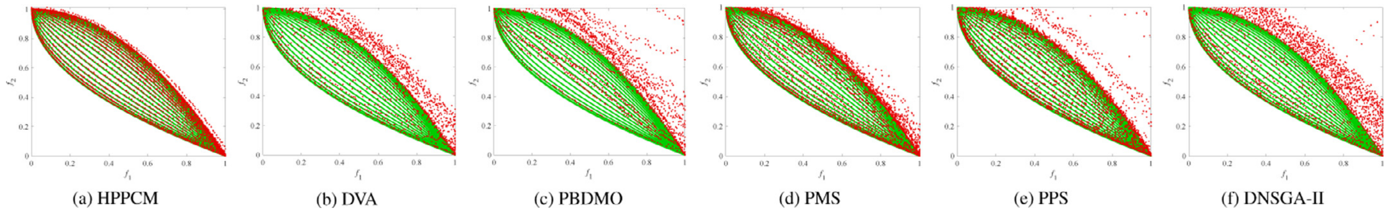
(c) F7

(d) F9

(e) FDA3

(f) JY3

Fig. 5. Evolution curve of MIGD value of six algorithms on representative test cases for time period t from 0 to 30, with (n_t, τ_t)=(10,30).



(a) HPPCM

(b) DVA

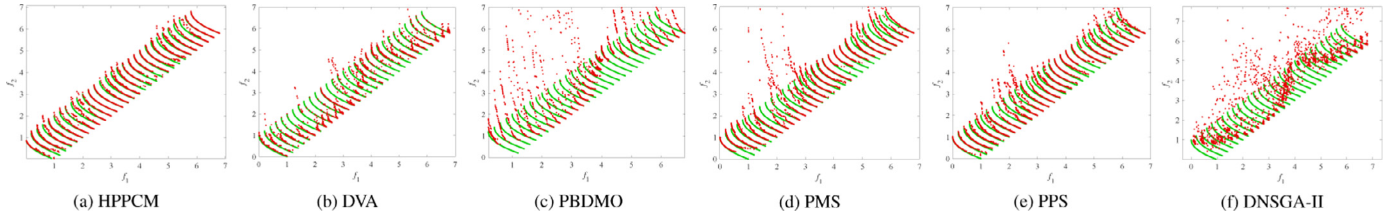
(c) PBDMO

(d) PMS

(e) PPS

(f) DNSGA-II

Fig. 6. Scatter plots of the population obtained by HPPCM and the other peer algorithms (mean MIGD value) on dMOP2 for time period t from 0 to 30. The green line is POF in different environments. Red dots represent solutions obtained by five algorithms.



(a) HPPCM

(b) DVA

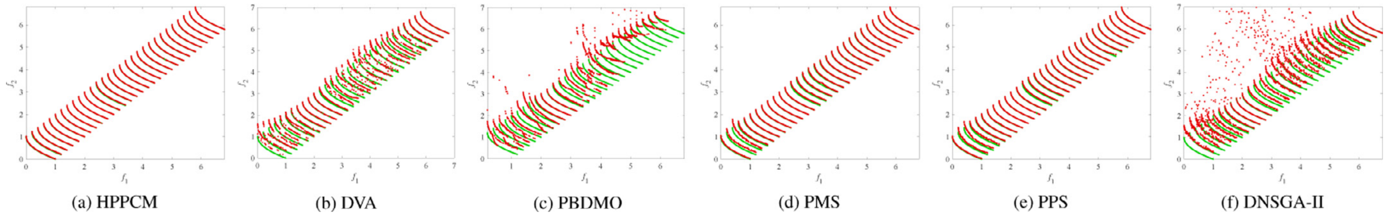
(c) PBDMO

(d) PMS

(e) PPS

(f) DNSGA-II

Fig. 7. Scatter plots of the population obtained by HPPCM and the other peer algorithms (mean MIGD value) on dMOP3 for time period t from 0 to 30. The green line is POF in different environments. Red dots represent solutions obtained by five algorithms.



(a) HPPCM

(b) DVA

(c) PBDMO

(d) PMS

(e) PPS

(f) DNSGA-II

Fig. 8. Scatter plots of the population obtained by HPPCM and the other peer algorithms (mean MIGD value) on FDA1 for time period t from 0 to 30. The green line is POF in different environments. Red dots represent solutions obtained by five algorithms.

Table 3
MHVD indicators of six algorithms on dMOP and FDA test suites.

Problem	n_t	DNSGA-II	PPS	PMS	PBDMO	DVA	HPPCM
dMOP1	5	**6.4974e-1 (6.07e-3)**	6.0034e-1 (5.84e-3)	6.4909e-1 (4.74e-3)	6.4855e-1 (4.40e-4)	6.4720e-1 (1.82e-4)	**6.5324e-1 (4.84e-3)**
	10	6.4918e-1 (3.43e-2)	6.1378e-1 (2.52e-3)	**6.5903e-1 (1.67e-4)**	5.5714e-1 (2.60e-4)	5.5601e-1 (1.32e-4)	**6.6298e-1 (1.38e-4)**
	20	**5.5791e-1 (3.71e-4)**	5.0651e-1 (2.39e-3)	5.5617e-1 (6.63e-5)	5.1964e-1 (6.04e-3)	5.5601e-1 (1.32e-4)	**5.5968e-1 (1.37e-4)**
dMOP2	5	3.1767e-1 (1.07e-1)	**5.8987e-1 (1.34e-2)**	5.5106e-1 (2.85e-2)	2.9603e-1 (1.79e-2)	5.3664e-1 (1.14e-2)	**5.9216e-1 (2.49e-2)**
	10	5.3468e-1 (1.40e-3)	**6.1624e-1 (8.60e-4)**	6.0783e-1 (3.23e-3)	5.2065e-1 (3.21e-3)	5.3238e-1 (1.25e-3)	**6.4074e-1 (1.63e-3)**
	20	5.1285e-1 (5.98e-4)	**5.3494e-1 (4.35e-4)**	5.3344e-1 (9.90e-4)	5.1964e-1 (6.04e-3)	5.3288e-1 (9.05e-4)	**5.5165e-1 (5.76e-4)**
dMOP3	5	4.0815e-1 (2.75e-2)	4.7919e-1 (1.76e-2)	**5.7644e-1 (3.25e-2)**	5.4377e-1 (3.53e-3)	4.1977e-1 (3.44e-2)	**6.5427e-1 (2.51e-2)**
	10	4.7470e-1 (3.19e-2)	5.2690e-1 (1.33e-2)	**6.1963e-1 (2.19e-2)**	5.6871e-1 (1.15e-2)	5.4960e-1 (1.73e-2)	**6.9971e-1 (1.41e-2)**
	20	4.9145e-1 (1.86e-2)	5.2450e-1 (2.28e-2)	**6.0577e-1 (1.31e-2)**	5.6714e-1 (1.57e-2)	5.4000e-1 (9.29e-3)	**7.0601e-1 (1.76e-2)**
FDA1	5	3.3255e-1 (1.48e-1)	**7.6260e-1 (1.82e-2)**	7.0222e-1 (3.74e-2)	4.2724e-1 (1.91e-2)	7.0154e-1 (1.21e-2)	**7.6226e-1 (3.23e-2)**
	10	6.3132e-1 (4.23e-3)	7.9794e-1 (4.12e-3)	7.8923e-1 (5.10e-3)	8.1571e-1 (2.87e-3)	**8.2836e-1 (2.80e-3)**	**8.2845e-1 (4.47e-3)**
	20	7.9348e-1 (1.43e-3)	**8.3149e-1 (1.75e-3)**	8.2971e-1 (2.12e-3)	8.1624e-1 (3.28e-3)	8.3027e-1 (2.48e-3)	**8.5571e-1 (1.21e-3)**
FDA2	5	**1.1714e+0 (1.50e-3)**	1.1167e+0 (2.42e-2)	1.1610e+0 (4.50e-3)	1.1607e+0 (2.46e-3)	1.1606e+0 (4.25e-3)	**1.1660e+0 (3.63e-3)**
	10	**1.1686e+0 (2.16e-4)**	1.1191e+0 (1.18e-2)	1.1591e+0 (3.67e-3)	1.1813e+0 (2.36e-3)	**1.1822e+0 (2.66e-3)**	1.1623e+0 (1.82e-3)
	20	**1.1887e+0 (1.54e-4)**	1.1625e+0 (1.22e-3)	1.1831e+0 (2.07e-3)	1.1815e+0 (3.34e-3)	1.1800e+0 (2.33e-3)	**1.1832e+0 (6.97e-4)**
FDA3	5	5.2649e-1 (1.58e-1)	2.3192e-1 (6.05e-2)	**5.9222e-1 (2.07e-1)**	5.7872e-1 (8.80e-3)	4.4121e-1 (6.60e-2)	**7.2262e-1 (1.17e-1)**
	10	**8.4422e-1 (6.45e-3)**	5.0572e-1 (1.20e-1)	8.2425e-1 (3.58e-2)	1.1287e+0 (5.89e-3)	1.1319e+0 (3.47e-3)	**8.7351e+0 (1.17e-2)**
	20	1.1063e+0 (9.90e-4)	1.1084e+0 (1.58e-2)	1.1358e+0 (4.67e-3)	1.1296e+0 (6.48e-3)	**1.1373e+0 (5.50e-3)**	**1.1690e+0 (1.95e-3)**
FDA4	5	1.4061e-1 (3.54e-2)	2.0856e-1 (4.44e-2)	**2.5423e-1 (3.10e-2)**	2.1152e-1 (6.22e-3)	2.4414e-1 (8.63e-3)	**3.4571e-1 (1.70e-2)**
	10	1.8522e-1 (2.52e-3)	2.6494e-1 (6.47e-3)	2.9261e-1 (4.44e-3)	**5.8472e-1 (8.97e-3)**	**5.8167e-1 (7.09e-3)**	3.6837e-1 (4.07e-3)
	20	5.6199e-1 (1.63e-3)	5.3839e-1 (2.53e-3)	5.8652e-1 (1.59e-2)	5.8168e-1 (1.32e-2)	**5.8717e-1 (1.07e-2)**	**6.4364e-1 (5.24e-3)**

Table 4
MIGD indicators of six algorithms on ZJZ test suites.

Problem	n_t	DNSGA-II	PPS	PMS	PBDMO	DVA	HPPCM
F5	5	1.6633e+1 (9.76e-1)	**3.8839e-1 (1.44e-1)**	1.4125e+0 (3.28e-1)	7.2257e-1 (8.29e-3)	1.3003e+0 (7.18e-1)	**1.6701e-1 (2.26e-2)**
	10	1.8170e+0 (1.01e-1)	1.4482e-1 (2.17e-2)	**3.1907e-2 (4.90e-3)**	4.3093e-1 (1.58e-2)	3.4219e-2 (8.21e-3)	**1.1356e-2 (1.23e-3)**
	20	1.7467e-1 (4.17e-2)	2.3893e-2 (1.76e-3)	**6.3331e-3 (5.04e-4)**	4.6497e-2 (3.15e-3)	7.5041e-3 (8.63e-4)	**1.3241e-3 (7.99e-5)**
F6	5	5.3326e+0 (1.92e-1)	**3.6509e-1 (1.16e-1)**	1.1246e+0 (3.15e-1)	3.8274e-1 (1.73e-2)	8.1421e-1 (4.52e-1)	**2.2993e-1 (1.17e-1)**
	10	8.5315e-1 (3.26e-2)	1.4700e-1 (7.70e-2)	**3.2520e-2 (7.09e-3)**	1.6387e-1 (9.39e-3)	4.7432e-2 (1.75e-2)	**1.1371e-2 (1.30e-3)**
	20	2.1045e-1 (3.55e-2)	8.7108e-2 (6.38e-2)	**9.5311e-3 (4.33e-3)**	3.8997e-2 (1.61e-3)	1.8302e-2 (1.11e-2)	**1.3848e-3 (5.50e-5)**
F7	5	5.2921e+0 (1.70e-1)	**2.7746e-1 (1.99e-2)**	3.7883e-1 (4.41e-2)	1.0802e+0 (8.49e-1)	4.1850e-1 (1.10e-1)	**1.3541e-1 (1.74e-2)**
	10	8.1724e-1 (5.08e-2)	5.0375e-2 (2.99e-2)	**3.3622e-2 (7.49e-3)**	1.6387e-1 (9.39e-3)	3.3933e-2 (9.71e-3)	**9.4294e-3 (1.10e-3)**
	20	2.0472e-1 (2.28e-2)	9.0249e-2 (3.93e-2)	**8.6027e-3 (2.52e-3)**	5.5929e-2 (3.98e-3)	1.3205e-2 (1.42e-2)	**1.3802e-3 (9.97e-5)**
F8	5	2.4524e-1 (7.21e-3)	**4.5332e-2 (2.70e-3)**	6.8489e-2 (2.14e-3)	4.2453e-1 (1.67e-2)	8.8367e-2 (6.33e-3)	**2.0990e-2 (3.95e-4)**
	10	5.5261e-2 (1.53e-3)	3.1018e-2 (1.96e-2)	**2.7007e-2 (6.73e-4)**	1.1235e-1 (5.39e-3)	2.8718e-2 (1.07e-3)	**9.6343e-3 (1.27e-4)**
	20	2.5369e-2 (1.80e-3)	2.1402e-2 (2.72e-4)	**1.2968e-2 (3.89e-4)**	2.3542e-2 (1.74e-3)	1.3463e-2 (3.88e-4)	**8.0578e-3 (1.04e-4)**
F9	5	9.9330e+2 (1.04e+2)	8.4395e+1 (3.54e+1)	**4.9092e-1 (1.06e-1)**	1.7952e+0 (1.32e-1)	2.1613e+0 (8.02e-1)	**2.5725e-2 (1.47e-2)**
	10	2.3389e+2 (1.90e+1)	1.5398e+1 (1.45e+1)	**1.2711e-1 (7.16e-2)**	3.1089e-1 (1.36e-1)	2.0945e-1 (2.62e-1)	**8.8778e-3 (6.99e-3)**
	20	1.2002e-2 (1.90e-2)	1.9008e-2 (5.93e-3)	**1.7285e-4 (9.22e-5)**	1.3900e+0 (3.79e-1)	3.7794e-2 (6.50e-2)	**7.3260e-5 (8.05e-6)**

Table 5
MHVD indicators of six algorithms on ZJZ test suites.

Problem	n_t	DNSGA-II	PPS	PMS	PBDMO	DVA	HPPCM
F5	5	6.3106e-2 (9.74e-3)	**4.2695e-1 (2.67e-2)**	3.6699e-1 (1.99e-2)	2.5030e-1 (6.15e-3)	3.3231e-1 (2.21e-2)	**4.5286e-1 (2.70e-2)**
	10	4.3050e-1 (4.34e-3)	5.9389e-1 (5.90e-3)	**6.9750e-1 (4.68e-3)**	5.4759e-1 (1.12e-2)	6.9473e-1 (8.02e-3)	**7.4704e-1 (2.57e-3)**
	20	1.4428e-1 (1.44e-2)	5.2581e-1 (1.13e-2)	6.0554e-1 (2.89e-2)	5.4886e-1 (3.43e-3)	**6.7041e-1 (1.17e-2)**	**6.8198e-1 (9.34e-3)**
F6	5	8.8188e-2 (5.24e-3)	2.4063e-1 (1.85e-2)	2.8787e-1 (3.19e-2)	**2.9741e-1 (5.10e-3)**	2.8257e-1 (2.55e-2)	**4.0176e-1 (1.25e-2)**
	10	3.8198e-1 (5.86e-3)	3.6644e-1 (3.31e-2)	**6.7972e-1 (6.58e-3)**	5.6595e-1 (1.17e-2)	6.6396e-1 (1.03e-2)	**7.4125e-1 (9.28e-4)**
	20	1.5431e-1 (5.64e-3)	3.1162e-1 (1.21e-2)	5.8676e-1 (1.57e-2)	5.6852e-1 (1.06e-2)	**6.7317e-1 (1.48e-2)**	**6.7407e-1 (6.28e-3)**
F7	5	1.8540e-1 (1.42e-2)	3.0984e-1 (2.95e-2)	**4.5310e-1 (1.00e-2)**	3.3748e-1 (6.96e-3)	4.5003e-1 (1.93e-2)	**4.8398e-1 (3.03e-2)**
	10	5.1094e-1 (5.17e-3)	4.5518e-1 (8.51e-3)	6.9369e-1 (1.09e-2)	5.5272e-1 (1.55e-1)	**6.9449e-1 (9.93e-3)**	**7.4730e-1 (2.33e-3)**
	20	3.2740e-1 (1.97e-2)	4.5443e-1 (2.10e-2)	6.2154e-1 (9.44e-3)	4.9087e-1 (2.21e-1)	**6.9495e-1 (3.47e-3)**	**7.0832e-1 (4.68e-3)**
F8	5	5.1133e-2 (2.44e-3)	5.7529e-2 (8.69e-3)	1.9871e-1 (1.84e-2)	1.9305e-1 (7.44e-3)	**3.2776e-1 (1.21e-2)**	**3.4080e-1 (7.63e-3)**
	10	4.0431e-1 (6.95e-3)	2.0480e-1 (8.32e-3)	5.0924e-1 (9.22e-3)	5.2452e-1 (7.72e-3)	**5.6914e-1 (8.33e-3)**	**5.7613e-1 (1.94e-3)**
	20	1.9078e-1 (4.86e-3)	1.3960e-1 (1.49e-2)	3.5125e-1 (7.56e-3)	**6.1125e-1 (3.88e-3)**	**5.8919e-1 (2.47e-3)**	5.0549e-1 (4.04e-3)
F9	5	1.3562e-3 (3.06e-3)	2.3397e-2 (5.22e-3)	2.2877e-1 (1.01e-1)	2.8643e-1 (4.96e-2)	**4.1602e-1 (2.15e-2)**	**4.5454e-1 (2.46e-2)**
	10	6.4471e-3 (1.29e-2)	1.3406e-2 (2.31e-2)	**6.5870e-1 (2.86e-2)**	6.5729e-1 (2.79e-2)	6.4763e-1 (1.45e-2)	**6.7845e-1 (1.87e-2)**
	20	5.1327e-3 (2.53e-3)	3.0270e-2 (2.19e-2)	2.7423e-1 (1.42e-2)	6.5696e-1 (1.25e-2)	**6.5964e-1 (1.30e-2)**	**6.9266e-1 (2.50e-2)**

by the algorithm proposed in this paper is close to the true POF, showing its excellent tracking ability. From the tables, with the decrease of n_t (that is, the dramatic increase in environmental changes), the performance of other algorithms except for HPPCM significantly deteriorates. It proves that HPPCM has significant advantages in the ZJZ test suite, and the ability to solve problems does not change considerably with the drastic changes in the environment.

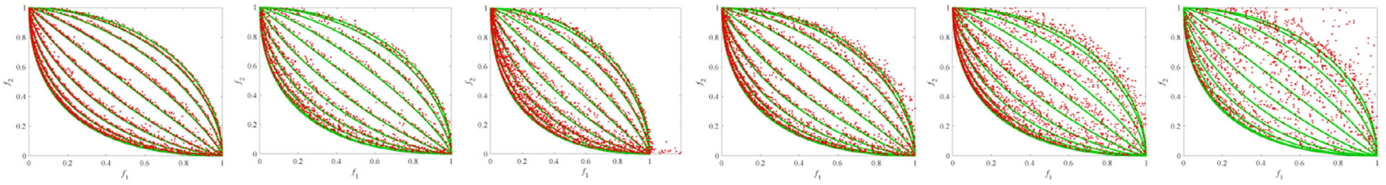
We know that there are nonlinear, time-varying and monotonic links between the decision variables of F5-F9, which are more complicated than the FDA and dMOP issues. Therefore, even though our algorithm performs significantly better than other algorithms on these problems, the statistics of MIGD and MHVD are worse than those on the FDA and dMOP issues.

Table 6
MIGD indicators of six algorithms on JY test suites.

Problem	n_t	DNSGA-II	PPS	PMS	PBDMO	DVA	HPPCM
JY1	5	9.2780e-1 (1.54e-1)	4.3532e-1 (2.70e-2)	**9.6021e-2 (1.17e-2)**	2.4464e-1 (3.62e-2)	9.0845e-1 (3.61e-1)	**1.2319e-1 (9.17e-3)**
	10	1.5286e-1 (5.98e-2)	2.6418e-1 (8.50e-3)	**2.3358e-2 (2.60e-3)**	6.8557e-2 (6.59e-3)	2.3287e-1 (9.50e-2)	**3.7127e-2 (4.20e-3)**
	20	**2.4849e-3 (1.69e-4)**	2.1365e-1 (1.48e-2)	5.9802e-3 (5.24e-4)	1.5175e-2 (1.68e-3)	7.3411e-2 (2.65e-2)	**1.9575e-3 (3.71e-5)**
JY2	5	2.4975e+0 (1.87e-1)	5.9372e+0 (9.58e-1)	**7.8376e-2 (6.71e-2)**	1.8624e-1 (1.04e-2)	2.3829e-1 (1.04e-1)	**2.5051e-2 (3.51e-3)**
	10	1.0446e-1 (4.60e-3)	1.0288e+1 (1.56e+0)	**1.4674e-2 (2.21e-3)**	4.0321e-2 (7.42e-3)	4.5996e-2 (1.04e-2)	**1.6070e-2 (1.31e-3)**
	20	1.0769e-3 (1.86e-5)	4.0338e-1 (1.18e-1)	**8.2064e-4 (6.95e-5)**	3.5461e-3 (6.86e-4)	4.5996e-2 (1.04e-2)	**7.1653e-4 (5.84e-5)**
JY3	5	1.9242e-2 (2.24e-2)	2.6497e-1 (3.33e-2)	**3.6756e-4 (4.57e-5)**	2.7016e-3 (3.29e-3)	3.6741e-2 (1.75e-2)	**3.0423e-4 (2.85e-5)**
	10	3.8699e-2 (1.62e-2)	2.0538e+0 (1.37e-1)	2.6716e-2 (2.12e-2)	**5.4587e-3 (6.58e-3)**	**7.3198e-3 (1.41e-2)**	1.6858e-2 (1.92e-2)
	20	4.3211e-2 (3.90e-4)	1.4439e+0 (9.58e-2)	3.8097e-2 (1.10e-2)	**2.8829e-3 (1.32e-3)**	**2.5613e-2 (2.24e-2)**	2.6206e-2 (2.06e-2)
JY4	5	2.3183e+1 (2.15e+0)	6.8423e+0 (7.44e-1)	2.5234e-1 (4.04e-2)	**1.9360e-1 (3.83e-2)**	3.2932e-1 (1.54e-1)	**2.0372e-1 (3.34e-2)**
	10	4.9113e+0 (6.53e-1)	3.1691e+0 (4.19e-1)	1.1713e-1 (2.69e-2)	**3.3258e-2 (1.22e-2)**	1.3428e-1 (5.14e-2)	**8.9616e-2 (7.77e-3)**
	20	1.7119e+0 (5.21e-1)	5.2227e+0 (8.08e-1)	4.5724e-2 (9.48e-3)	**2.2386e-2 (1.58e-2)**	5.4695e-2 (1.23e-2)	**5.0983e-2 (2.74e-3)**
JY5	5	**3.2053e-4 (3.85e-5)**	7.1897e-1 (5.39e-2)	1.1179e-3 (2.17e-4)	1.3733e-3 (4.82e-4)	1.7811e-3 (6.37e-4)	**1.3597e-4 (2.73e-5)**
	10	**3.8760e-4 (5.60e-5)**	6.6942e-1 (3.89e-2)	9.0069e-4 (1.51e-4)	8.1458e-4 (4.10e-4)	8.7452e-4 (2.50e-4)	**1.5475e-4 (2.67e-5)**
	20	**3.9710e-4 (5.62e-5)**	7.6084e-1 (6.62e-2)	7.3291e-4 (9.48e-5)	5.2916e-4 (1.47e-4)	1.1090e-3 (2.42e-4)	**1.9590e-4 (3.40e-5)**
JY6	5	2.1115e+2 (3.92e+0)	1.0276e+2 (5.40e+0)	1.3592e+2 (3.41e+1)	**4.5842e+0 (1.57e+0)**	1.8410e+2 (8.55e+1)	**7.6213e+1 (7.52e+0)**
	10	1.6285e+2 (2.77e+0)	9.2926e+1 (5.06e+0)	6.0299e+1 (7.17e+0)	**4.5561e+0 (1.35e+0)**	9.2046e+1 (3.69e+1)	**4.5478e+1 (2.77e+0)**
	20	1.0269e+2 (1.76e+0)	7.7956e+1 (1.68e+0)	3.3226e+1 (1.11e+1)	**1.5462e+0 (4.33e-1)**	4.1472e+1 (1.25e+1)	**5.9184e+0 (5.95e+0)**

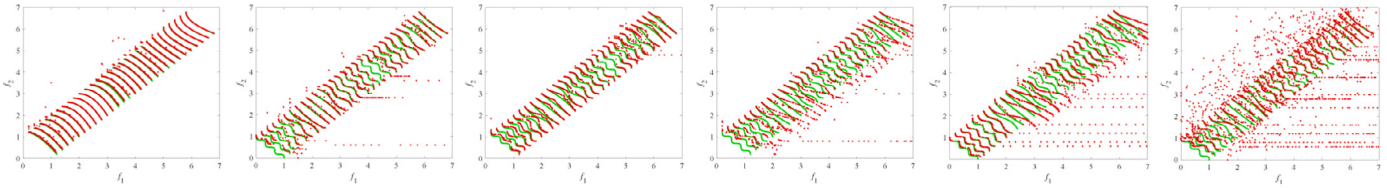
Table 7
MHVD indicators of six algorithms on JY test suites.

Problem	n_t	DNSGA-II	PPS	PMS	PBDMO	DVA	HPPCM
JY1	5	3.8760e-1 (5.69e-2)	1.4449e-1 (1.09e-2)	**5.1969e-1 (7.55e-3)**	3.3443e-1 (6.01e-3)	2.2141e-1 (3.18e-2)	**4.9176e-1 (1.36e-2)**
	10	4.1066e-1 (3.84e-2)	1.5353e-1 (1.33e-2)	**5.2806e-1 (1.21e-2)**	**5.6677e-1 (1.18e-2)**	4.6090e-1 (1.05e-2)	4.9530e-1 (1.37e-2)
	20	2.5530e-1 (1.14e-1)	1.2783e-2 (8.09e-3)	**4.6417e-1 (1.44e-2)**	**5.6379e-1 (9.99e-3)**	4.5837e-1 (9.25e-3)	4.1699e-1 (1.25e-2)
JY2	5	5.8378e-1 (1.13e-3)	1.6663e-1 (3.15e-2)	**6.3716e-1 (2.81e-3)**	3.6664e-1 (7.18e-3)	3.0853e-1 (3.00e-2)	**6.5956e-1 (1.64e-3)**
	10	5.8499e-1 (1.23e-3)	1.8063e-1 (3.91e-2)	**6.3828e-1 (4.60e-3)**	6.3333e-1 (4.51e-3)	6.0098e-1 (1.11e-2)	**6.5781e-1 (3.27e-3)**
	20	6.0784e-1 (1.72e-3)	1.3755e-1 (5.20e-3)	**6.4168e-1 (5.05e-3)**	6.3238e-1 (6.77e-3)	6.0120e-1 (7.55e-3)	**6.0201e-1 (5.11e-4)**
JY3	5	5.6486e-1 (1.06e-1)	1.5347e-1 (6.18e-3)	**6.8314e-1 (6.79e-3)**	6.4781e-1 (1.31e-2)	5.4812e-1 (1.05e-1)	**6.8723e-1 (1.34e-3)**
	10	5.8760e-1 (9.66e-2)	1.5721e-1 (2.06e-2)	**6.8130e-1 (3.35e-3)**	6.6289e-1 (1.65e-2)	5.3007e-1 (9.20e-2)	**6.8761e-1 (1.20e-3)**
	20	4.4448e-1 (2.30e-2)	8.4381e-2 (8.50e-3)	5.2916e-1 (1.00e-1)	5.4181e-1 (1.22e-1)	**5.9560e-1 (9.40e-2)**	**6.6430e-1 (1.78e-2)**
JY4	5	2.7658e-1 (3.91e-2)	2.2002e-1 (3.06e-2)	**6.6260e-1 (1.99e-2)**	4.2896e-1 (1.00e-2)	3.2048e-1 (5.04e-2)	**6.8190e-1 (1.10e-2)**
	10	2.8406e-1 (2.40e-2)	2.4521e-1 (3.22e-2)	6.3603e-1 (2.66e-2)	**7.0483e-1 (4.08e-2)**	4.9167e-1 (4.34e-2)	**6.7490e-1 (1.44e-2)**
	20	1.2453e-1 (3.31e-2)	6.7864e-2 (5.21e-3)	4.4530e-1 (1.96e-1)	**6.8475e-1 (2.90e-2)**	**5.1808e-1 (5.83e-2)**	4.0076e-1 (1.72e-1)
JY5	5	6.1579e-1 (3.59e-4)	4.5731e-1 (1.83e-2)	6.1267e-1 (6.93e-4)	**6.2172e-1 (7.18e-3)**	6.1418e-1 (4.44e-3)	**6.1881e-1 (8.325e-4)**
	10	6.1593e-1 (3.48e-4)	4.6794e-1 (2.02e-2)	6.1265e-1 (4.58e-4)	**7.3916e-1 (4.88e-3)**	**7.3279e-1 (3.62e-3)**	6.1885e-1 (1.42e-4)
	20	6.1843e-1 (2.90e-1)	1.6355e-1 (4.22e-2)	6.0365e-1 (2.82e-1)	**7.3820e-1 (5.25e-3)**	**7.3277e-1 (3.14e-3)**	6.2226e-1(2.92e-1)
JY6	5	6.6516e-3 (4.35e-4)	1.4273e-2 (4.04e-3)	**4.6479e-2 (4.52e-3)**	4.0929e-2 (5.43e-3)	1.0706e-3 (3.03e-3)	**1.0330e-1 (1.76e-2)**
	10	6.9938e-3 (4.23e-4)	1.3226e-2 (2.61e-3)	5.1077e-2 (6.30e-3)	**1.1704e-1 (3.66e-2)**	6.0286e-2 (6.30e-2)	**8.6256e-2 (1.93e-2)**
	20	3.7044e-2 (3.54e-2)	2.0000e-2 (4.47e-2)	5.4615e-2 (4.75e-2)	**9.4662e-2 (1.92e-2)**	6.6037e-2 (5.88e-2)	**7.3611e-2 (5.06e-2)**



(a) HPCM (b) DVA (c PBDMO (d) PMS (e) PPS (f) DNSGA-II

Fig. 9. Scatter plots of the population obtained by HPPCM and the other peer algorithms (mean MIGD value) on F9 for time period *t* from 0 to 30. The green line is POF in different environments. Red dots represent solutions obtained by five algorithms.



(a) HPPCM (b) DVA (c) PBDMO (d) PMS (e) PPS (f) DNSGA-II

Fig. 10. Scatter plots of the population obtained by HPPCM and the other peer algorithms (mean MIGD value) on JY3 for time period *t* from 0 to 30. The green line is POF in different environments. Red dots represent solutions obtained by five algorithms.

Table 8
MIGD and MHVD indicators of MO_Ring_PSO_SCD and MOAGDE on dMOP, FDA, ZJZ and JY test suites.

Problem	n_t	Metric	MO_Ring_PSO_SCD	MOAGDE	HPPCM	Metric	MO_Ring_PSO_SCD	MOAGDE	HPPCM
dMOP1	5	MIGD	8.6988e+1 (5.54e+0)	**4.4140e+1 (5.09e+0)**	**4.5702e-5 (1.32e-6)**	MHVD	0.0000e+0 (0.00e+0)	0.0000e+0 (0.00e+0)	**6.5324e-1 (4.84e-3)**
	10		7.7450e+1 (3.21e+1)	**5.4282e+1 (2.96e+1)**	**4.6014e-5 (6.60e-7)**		0.0000e+0 (0.00e+0)	0.0000e+0 (0.00e+0)	**6.6298e-1 (1.38e-4)**
	20		9.6132e+1 (4.75e+0)	**4.8751e+1 (4.67e+0)**	**4.6434e-5 (9.91e-7)**		0.0000e+0 (0.00e+0)	0.0000e+0 (0.00e+0)	**5.5968e-1 (1.37e-4)**
dMOP2	5	MIGD	2.7802e+1 (1.52e+0)	**6.6768e+0 (6.92e-1)**	**1.1559e-2 (6.18e-3)**	MHVD	**3.6092e-2 (5.83e-3)**	7.8985e-3 (6.02e-3)	**5.9216e-1 (2.49e-2)**
	10		3.8227e+1 (1.46e+0)	**2.2593e+1 (1.64e+1)**	**4.7595e-4 (6.81e-5)**		**3.1719e-2 (3.29e-3)**	5.3396e-3 (2.65e-3)	**6.4074e-1 (1.63e-3)**
	20		6.2810e-1 (6.57e-2)	**1.0448e+1 (7.07e-1)**	**8.8123e-5 (4.57e-6)**		**5.8307e-2 (7.73e-3)**	1.8788e-3 (2.16e-3)	**5.5165e-1 (5.76e-4)**
dMOP3	5	MIGD	**2.0591e-1 (2.17e-2)**	2.3949e-1 (3.96e-2)	**3.6396e-2 (2.59e-3)**	MHVD	3.3471e-1 (1.22e-2)	**3.5203e-1 (1.09e-2)**	**6.5427e-1 (2.51e-2)**
	10		**2.1139e-1 (1.65e-2)**	2.4190e-1 (2.83e-2)	**1.4950e-2 (1.04e-3)**		3.3696e-1 (6.08e-3)	**3.5350e-1 (4.85e-3)**	**6.9971e-1 (1.41e-2)**
	20		**2.0347e-1 (2.15e-2)**	2.2930e-1 (2.11e-2)	**1.3868e-2 (1.59e-3)**		**3.3781e-1 (1.24e-2)**	3.2835e-1 (9.69e-3)	**7.0601e-1 (1.76e-2)**
FDA1	5	MIGD	**1.5715e+2 (5.11e+0)**	4.5257e+1 (4.62e+0)	**2.1367e-2 (4.16e-3)**	MHVD	**3.3670e-2 (4.58e-3)**	2.1382e-4 (6.05e-4)	**7.6226e-1 (3.23e-2)**
	10		2.2132e+2 (3.78e+0)	**1.1914e+2 (9.01e+1)**	**1.3117e-3 (1.00e-4)**		**2.9976e-2 (5.57e-3)**	0.0000e+0 (0.00e+0)	**8.2845e-1 (4.47e-3)**
	20		**2.0776e+0 (1.89e-1)**	5.8625e+1 (5.18e+0)	**1.6656e-4 (1.95e-5)**		**5.3284e-2 (5.51e-3)**	0.0000e+0 (0.00e+0)	**8.5571e-1 (1.21e-3)**
FDA2	5	MIGD	6.7112e-1 (3.65e-2)	**1.2194e-1 (1.04e-2)**	**9.9827e-5 (3.57e-5)**	MHVD	3.8078e-1 (8.14e-3)	**7.8881e-1 (8.12e-3)**	**1.1660e+0 (3.63e-3)**
	10		5.9862e-1 (4.99e-2)	**3.1520e-1 (2.36e-1)**	**7.4949e-5 (1.78e-5)**		3.8165e-1 (1.11e-2)	**7.8686e-1 (1.20e-2)**	**1.1623e+0 (1.82e-3)**
	20		2.1278e-1 (1.23e-2)	**3.4863e-2 (1.92e-3)**	**7.8743e-5 (1.81e-5)**		5.6576e-1 (1.25e-2)	**9.4569e-1 (6.27e-3)**	**1.1832e+0 (6.97e-4)**
FDA3	5	MIGD	**2.9236e+0 (2.55e-1)**	5.3657e+1 (5.47e+0)	**3.0841e-2 (4.35e-2)**	MHVD	**6.4803e-2 (5.57e-3)**	0.0000e+0 (0.00e+0)	**7.2262e-1 (1.17e-1)**
	10		**2.9880e+0 (2.43e-1)**	4.0369e+1 (3.19e+1)	**3.2073e-3 (7.67e-4)**		**6.7832e-2 (7.12e-3)**	0.0000e+0 (0.00e+0)	**8.7351e+0 (1.17e-2)**
	20		**3.6634e+0 (2.49e-1)**	7.6481e+1 (6.43e+0)	**2.8730e-4 (2.53e-5)**		**3.4721e-2 (6.94e-3)**	0.0000e+0 (0.00e+0)	**1.1690e+0 (1.95e-3)**
FDA4	5	MIGD	2.8942e+1 (9.25e-1)	**2.8174e+1 (1.79e+0)**	**1.8237e+1 (2.00e-2)**	MHVD	6.4597e-3 (1.89e-3)	**1.1970e-2 (2.37e-3)**	**3.4571e-1 (1.70e-2)**
	10		4.0032e+1 (1.53e+0)	**4.0164e+1 (1.89e+0)**	**1.9376e+1 (2.68e-2)**		7.2237e-3 (1.56e-3)	**1.2403e-2 (2.95e-3)**	**3.6837e-1 (4.07e-3)**
	20		**5.9050e-1 (3.42e-2)**	7.4481e-1 (3.71e-2)	**1.2907e+1 (2.56e-2)**		1.0634e-2 (1.58e-3)	**1.8828e-2 (3.18e-3)**	**6.4364e-1 (5.24e-3)**
JY1	5	MIGD	1.0618e+2 (5.58e+0)	**4.0607e+1 (2.22e+0)**	**1.2319e-1 (9.17e-3)**	MHVD	**1.7747e-3 (1.70e-3)**	0.0000e+0 (0.00e+0)	**4.9176e-1 (1.36e-2)**
	10		1.4931e+2 (3.81e+0)	**8.5989e+1 (5.13e+1)**	**3.7127e-2 (4.20e-3)**		**3.0774e-3 (2.29e-3)**	0.0000e+0 (0.00e+0)	**4.9530e-1 (1.37e-2)**
	20		**2.6292e+0 (1.60e-1)**	5.3640e+1 (3.39e+0)	**1.9575e-3 (3.71e-5)**		**3.4933e-3 (1.88e-3)**	0.0000e+0 (0.00e+0)	**4.1699e-1 (1.25e-2)**
JY2	5	MIGD	1.1411e+2 (4.07e+0)	**4.4030e+1 (4.33e+0)**	**2.5051e-2 (3.51e-3)**	MHVD	**3.4020e-3 (1.36e-3)**	0.0000e+0 (0.00e+0)	**6.5956e-1 (1.64e-3)**
	10		1.6009e+2 (7.49e+0)	**9.3595e+1 (5.87e+1)**	**1.6070e-2 (1.31e-3)**		**2.7404e-3 (2.31e-3)**	0.0000e+0 (0.00e+0)	**6.5781e-1 (3.27e-3)**
	20		**2.6875e+0 (2.24e-1)**	5.4005e+1 (3.20e+0)	**7.1653e-4 (5.84e-5)**		0.0000e+0 (0.00e+0)	0.0000e+0 (0.00e+0)	**6.0201e-1 (5.11e-4)**
JY3	5	MIGD	**2.1436e+0 (1.04e-1)**	5.3994e+0 (2.76e-1)	**3.0423e-4 (2.85e-5)**	MHVD	**1.1770e-6 (3.33e-6)**	0.0000e+0 (0.00e+0)	**6.8723e-1 (1.34e-3)**
	10		**2.2362e+0 (6.53e-2)**	4.3152e+0 (1.80e+0)	**1.6858e-2 (1.92e-2)**		**0.0000e+0 (0.00e+0)**	0.0000e+0 (0.00e+0)	**6.8761e-1 (1.20e-3)**
	20		**2.1538e+0 (5.12e-2)**	5.6907e+0 (4.65e-1)	**2.6206e-2 (2.06e-2)**		**7.0052e-3 (3.43e-3)**	0.0000e+0 (0.00e+0)	**6.6430e-1 (1.78e-2)**
JY4	5	MIGD	1.0678e+2 (5.21e+0)	**4.3052e+1 (2.39e+0)**	**2.0372e-1 (3.34e-2)**	MHVD	**3.7119e-3 (2.43e-3)**	0.0000e+0 (0.00e+0)	**6.8190e-1 (1.10e-2)**
	10		1.5053e+2 (3.80e+0)	**8.6698e+1 (5.17e+1)**	**8.9616e-2 (7.77e-3)**		**4.8337e-3 (1.71e-3)**	3.8353e-5 (9.39e-5)	**6.7490e-1 (1.44e-2)**
	20		**2.6140e+0 (1.15e-1)**	5.6037e+1 (3.41e+0)	**5.0983e-2 (2.74e-3)**		**7.0052e-3 (3.43e-3)**	0.0000e+0 (0.00e+0)	**4.0076e-1 (1.72e-1)**
JY5	5	MIGD	8.1503e+0 (2.65e-1)	**4.6559e+0 (2.08e-1)**	**1.3597e-4 (2.73e-5)**	MHVD	1.1998e-3 (1.77e-3)	**7.9276e-3 (4.52e-3)**	**6.1881e-1 (8.325e-4)**
	10		8.4790e+0 (2.83e-1)	**6.2306e+0 (1.69e+0)**	**1.5475e-4 (2.67e-5)**		1.4880e-3 (2.80e-3)	**6.7500e-3 (1.90e-3)**	**6.1885e-1 (1.42e-4)**
	20		6.3566e+0 (3.61e-1)	**3.3963e+0 (2.48e-1)**	**1.9590e-4 (3.40e-5)**		1.6139e-3 (1.48e-3)	**1.1006e-2 (4.75e-3)**	**6.2226e-1(2.92e-1)**
JY6	5	MIGD	1.6784e+3 (5.07e+1)	**9.7185e+2 (6.46e+1)**	**7.6213e+1 (7.52e+0)**	MHVD	0.0000e+0 (0.00e+0)	0.0000e+0 (0.00e+0)	**1.0330e-1 (1.76e-2)**
	10		2.2314e+3 (7.00e+1)	**1.5085e+3 (6.33e+2)**	**4.5478e+1 (2.77e+0)**		0.0000e+0 (0.00e+0)	0.0000e+0 (0.00e+0)	**8.6256e-2 (1.93e-2)**
	20		**1.9874e+2 (5.66e+0)**	1.1790e+3 (3.73e+1)	**5.9184e+0 (5.95e+0)**		0.0000e+0 (0.00e+0)	0.0000e+0 (0.00e+0)	**7.3611e-2 (5.06e-2)**
F5	5	MIGD	3.6349e+3 (1.10e+2)	**5.2821e+1 (1.61e+2)**	**1.6701e-1 (2.26e-2)**	MHVD	7.0733e-2 (1.79e-3)	**1.1299e-1 (3.09e-2)**	**4.5286e-1 (2.70e-2)**
	10		3.1162e+3 (2.56e+2)	**4.6274e+2 (1.30e+3)**	**1.1356e-2 (1.23e-3)**		7.0702e-2 (7.56e-4)	**1.3025e-1 (2.63e-2)**	**7.4704e-1 (2.57e-3)**
	20		2.6889e+3 (1.44e+2)	**1.1099e+2 (3.25e+2)**	**1.3241e-3 (7.99e-5)**		7.8271e-2 (4.29e-3)	**1.2279e-1 (1.99e-2)**	**6.8198e-1 (9.34e-3)**
F6	5	MIGD	7.2451e+1 (5.18e+0)	**3.2611e+0 (2.13e+0)**	**2.2993e-1 (1.17e-1)**	MHVD	0.0000e+0 (0.00e+0)	**4.0593e-2 (1.68e-2)**	**4.0176e-1 (1.25e-2)**
	10		8.1990e+1 (2.97e+0)	**1.2119e+1 (2.68e+1)**	**1.1371e-2 (1.30e-3)**		0.0000e+0 (0.00e+0)	**5.1861e-2 (1.28e-2)**	**7.4125e-1 (9.28e-4)**
	20		8.6598e+1 (4.02e+0)	**2.6244e+0 (1.42e+0)**	**1.3848e-3 (5.50e-5)**		0.0000e+0 (0.00e+0)	**4.0376e-2 (1.64e-2)**	**6.7407e-1 (6.28e-3)**
F7	5	MIGD	5.2379e+4 (7.28e+2)	**3.8377e+0 (1.33e+0)**	**1.3541e-1 (1.74e-2)**	MHVD	0.0000e+0 (0.00e+0)	**3.1979e-2 (1.38e-2)**	**4.8398e-1 (3.03e-2)**
	10		5.9177e+4 (3.47e+3)	**6.2107e+3 (1.75e+4)**	**9.4294e-3 (1.10e-3)**		0.0000e+0 (0.00e+0)	**2.8995e-2 (9.39e-3)**	**7.4730e-1 (2.33e-3)**
	20		6.0987e+4 (8.04e+2)	**6.5400e+0 (5.93e+0)**	**1.3802e-3 (9.97e-5)**		0.0000e+0 (0.00e+0)	**3.9885e-2 (1.99e-2)**	**7.0832e-1 (4.68e-3)**
F8	5	MIGD	**3.1255e+0 (1.93e-1)**	7.4457e+0 (8.13e-1)	**2.0990e-2 (3.95e-4)**	MHVD	**1.5202e-2 (3.76e-3)**	3.2420e-6 (9.17e-6)	**3.4080e-1 (7.63e-3)**
	10		**4.0381e+0 (3.21e-1)**	9.5666e+0 (1.36e+0)	**9.6343e-3 (1.27e-4)**		**1.1644e-2 (4.20e-3)**	0.0000e+0 (0.00e+0)	**5.7613e-1 (1.94e-3)**
	20		**3.6030e+0 (2.36e-1)**	7.5893e+0 (8.74e-1)	**8.0578e-3 (1.04e-4)**		**1.5679e-2 (4.45e-3)**	8.6047e-6 (2.28e-5)	**5.0549e-1 (4.04e-3)**
F9	5	MIGD	3.6501e+4 (2.27e+3)	**6.8845e+0 (1.52e+0)**	**2.5725e-2 (1.47e-2)**	MHVD	**5.5727e-2 (2.93e-3)**	2.0813e-2 (1.16e-2)	**4.5454e-1 (2.46e-2)**
	10		1.3048e+5 (2.92e+3)	**5.3283e+0 (1.41e+0)**	**8.8778e-3 (6.99e-3)**		**5.6954e-2 (2.43e-3)**	4.4357e-2 (1.97e-2)	**6.7845e-1 (1.87e-2)**
	20		2.9955e+5 (2.76e+3)	**4.1242e+0 (1.74e+0)**	**7.3260e-5 (8.05e-6)**		0.0000e+0 (0.00e+0)	0.0000e+0 (0.00e+0)	**6.9266e-1 (2.50e-2)**

5.3. Results on JYs

Tables 6 and 7 imply the statistical data (MIGDand MHVD) of each algorithm running 20 times on the JY test suite (JY1-JY6). JY1- JY6 sums up to 6 test instances. For each problem, they correspond to 3 cases where the value of n_t differs, so there are 18 cases altogether for each algorithm. Among 18 cases for each algorithm, HPPCM obtains the best MIGD results in 8 cases and the second-best ones in 8 cases. Table 6 displays that HPPCM receives the best MHVD results in 9 cases and the second-best ones in 4 cases. As far as MIGD is concerned, there are 8 cases PMDMO performs best. As far as MHVD is concerned, it obtains the best MHVD results in 8 cases. Judging from the indicators of MIGD and MHVD, the convergence and distribution performance of HPPCM and PBDMO on the JY test suite is 50/50.

The overall performance of HPPCM on the JY test suite is much better than that of DNSGA-II and PPS. There are nonlinear, time-varying and nonmonotonic links among JY's variables. Among them, JY6 is a multi-peak problem that changes with time. Not only the local optimal value will change with time, but the POS will also move dynamically. This proves that HPPCM has the potential to solve such complex problems.

Except for the JY1 test case, the overall performance of HPPCM is much better than that of PMS. JY1 is a type I problem, and POF is fixed. As mentioned earlier, HPPCM has a better ability to track changing POS (or POF), and the statistical data at this time is not the best. When n_t = 20, the DVA method is better than HPPMS on JY1, JY3, JY4 and JY5. It means when the environmental change is slight, for solving those problems which are nonlinear, time-varying and nonmonotonic linkage between variables, DVA is more competitive.

Table 9
Performance comparison of different HPPCM variants on MIGD for $(n_t, \tau_t) = (10,30)$.

Problem	HPPCM-V1	HPPCM-V2	HPPCM-V3	HPPCM
dMOP1	**4.6390e-5 (1.54e-6)**	6.9656e-5 (1.45e-6)	4.7265e-5 (3.05e-6)	**4.6014e-5 (6.60e-7)**
dMOP2	6.6974e-4 (1.65e-4)	9.4119e-3 (2.57e-3)	**4.9201e-4 (1.20e-4)**	**4.7595e-4 (6.81e-5)**
dMOP3	3.6307e-2 (4.53e-3)	**1.8731e-2 (8.78e-4)**	2.8251e-2 (4.20e-3)	**1.4950e-2 (1.04e-3)**
FDA1	2.3073e-3 (7.37e-4)	7.0876e-2 (1.20e-2)	**1.4067e-3 (3.38e-4)**	**1.3117e-3 (1.00e-4)**
FDA2	**1.1011e-4 (1.99e-5)**	1.2659e-4 (3.05e-5)	1.6775e-4 (4.74e-5)	**7.4949e-5 (1.78e-5)**
FDA3	5.1796e-3 (1.45e-3)	**2.7174e-2 (4.59e-3)**	3.4872e-3 (1.09e-3)	**3.2073e-3 (7.67e-4)**
FDA4	1.9642e+1 (3.17e-3)	**1.9606e+1 (4.11e-3)**	1.9612e+1 (3.13e-3)	**1.9376e+1 (2.68e-2)**
F5	2.4656e-2 (4.45e-3)	5.5388e-2 (1.66e-2)	**1.4263e-2 (2.94e-3)**	**1.1356e-2 (1.23e-3)**
F6	2.3031e-2 (5.01e-3)	7.0582e-2 (1.11e-2)	**1.3035e-2 (1.34e-3)**	**1.1371e-2 (1.30e-3)**
F7	**1.2444e-2 (2.35e-3)**	6.1026e-2 (1.04e-2)	8.6800e-2 (2.37e-3)	**9.4294e-3 (1.10e-3)**
F8	**1.0292e-2 (4.27e-4)**	2.7610e-2 (9.83e-4)	9.8335e-3 (2.09e-4)	**9.6343e-3 (1.27e-4)**
F9	4.4175e-2 (1.43e-4)	6.8077e-2 (1.26e-4)	**2.8492e-2 (1.07e-4)**	**8.8778e-3 (6.99e-3)**
JY1	4.5901e-1 (1.03e-1)	**3.8378e-1 (3.70e-2)**	2.2765e-1 (7.70e-2)	**3.7127e-2 (4.20e-3)**
JY2	1.8027e-2 (5.80e-3)	**1.7001e-1 (4.00e-2)**	6.2010e-1 (1.68e-3)	**1.6070e-2 (1.31e-3)**
JY3	3.2946e-2 (2.30e-2)	**3.1377e-2 (2.25e-2)**	3.9064e-2 (1.87e-2)	**1.6858e-2 (1.92e-2)**
JY4	4.0538e-1 (5.41e-3)	1.8298e-1 (3.49e-2)	**8.9717e-2 (3.47e-2)**	**8.9616e-2 (7.77e-3)**
JY5	2.9507e-3 (5.59e-4)	4.2406e-3 (1.67e-3)	**1.2621e-3 (3.99e-4)**	**1.5475e-4 (2.67e-5)**
JY6	**5.7893e+1 (4.97e+0)**	9.3416e+1 (6.39e+0)	7.0075e+1 (3.74e+1)	**4.5478e+1 (2.77e+0)**

5.4. Other comparative experiments

In addition to the tabular format, we also provide the evolution curve of the IGD value under the first 30 time periods of most test cases. We find from Fig. 5 that for most of the test problems, the MIGD curve of HPPCM is relatively flat and as low as close to the coordinate axis. It suggests that compared with other algorithms, HPPCM responds more stably to changes and recovers faster. The only exception is dMOP2. PPS performed poorly because of insufficient historical information accumulation before $t = 6$, but it performed best after $t = 16$. Thence comprehensively, the average IGD (MIGD) value was worse than HPPCM under the 30 time periods.

By observing Fig. 5, it is not difficult to find that the MIGD value of the algorithm fluctuates greatly on most problems, such as dMOP2, dMOP3 and FDA3. Nevertheless, compared with other algorithms, HPPCM has the smallest fluctuation range and more stable performance. To understand the tracking ability of these algorithms intuitively, we also plot the approximate POF obtained by each algorithm on dMOP2, dMOP3, FDA1, F9, JY3, for $(n_t, \tau_t) = (10,30)$, in Figs. 6–10. Compared with the data table and the index value change curve, the approximate POF graph can more intuitively reflect the excellent tracking ability of HPPCM. Figs. 6–10 clearly shows that in addition to HPPCM, the approximate POF obtained by other algorithms on most test cases cannot accurately fit the true POF. It shows that HPPCM has a powerful ability to respond to environmental changes. For the two fixed POF test cases, FDA2 and dMOP3, HPPCM has the defect that it is relatively difficult for some solutions (including boundary solutions) to converge. Therefore, it also affects the MIGD and MHVD index values that reflect convergence on these issues.

In our previous work, the comparison algorithm used is always the classic DMOEA, such as DNSGA-II, PPS, etc. We found that multi-modal MOEAs mainly design operators in the decision space, such as the dominance relationship in the decision space and the special crowding distance [49]. Dynamic MOEAs are also usually operated in the decision space, such as the analysis of decision variables [45], the crowding distance ranking in the decision space [38]. It is precisely because of this commonality that we finally added a powerful and another newest multi-modal MOP algorithm as a competitor. They are MO_Ring_PSO_SCD [49] and MOAGDE [50] respectively. The MIGD and MHVD indicators of MO_Ring_PSO_SCD and MOAGDE on dMOP, FDA, ZJZ and JY test suites can be found in Table 8. Obviously, HPPCM wins over these two no matter judging by MIGD metric and MHVD metric.

5.5. Component analysis

The proposed framework in this paper has three key components and it has shown competence. Here, we use ablation studies to verify the effectiveness of each component separately. The ablation experiment is similar to the ”controlled variable method.” Suppose that a certain system used A, B and C, and has achieved good results. We could keep two of the three and remove one to verify whether the deleted one plays a role in the entire system. In this subsection, we modify HPPCM to three variants. The first variant, HPPCM-V1, only discards the center point-based prediction strategy we proposed and retains the guiding individual-based prediction strategy and the precision controllable mutation strategy. It's not difficult to find that HPPCM-V1 is designed to verify whether the center point-based prediction strategy plays a decisive role in HPPCM. The guiding individual-based prediction strategy is switched off to study its importance to the framework in this paper, and this variant is called HPPCM-V2. Similarly, we also deactivate the precision controllable mutation strategy to study the role it plays in our proposed algorithm, resulting in another variant named HPPCM-V3.

These three variants are all compared with the original HPPCM, and their MIGD values on eighteen test instances for $(n_t, \tau_t) = (10,30)$ are presented in Table 9. It is evident from the results of the experimental comparison that the overall performance of the original version of HPPCM is better than other variants. This also reflects from the side that each component is indispensable for the overall performance of HPPCM. Removing any of them causes somewhat performance degradation in general. Thus, it is a good way to combine them in one algorithm, as done by HPPCM. It is not surprising to see that HPPCM-V1 performs worse than HPPCM. The reason may be due to the fact that the center point-based prediction strategy in HPPCM would generate many non-dominated solutions closer to new optima. The lack of this strategy in HPPCM-V1 is not good for population convergence. For those periodic problems, the evolution direction at a certain time step may be completely opposite to the direction of the previous time step. The guiding individual-based prediction strategy can accurately predict the evolution direction in the current environment, thereby adjusting the area where the predicted solution is generated. This is why the performance of HPPCM-V2 after removing this strategy has significantly decreased on periodic problems such as dMOP1, dMOP2, F5, F6, F7, F9. The precision controllable mutation strategy proposed in this paper is contributed to ensuring the diversity of our algorithm. Thus the lack of this strategy in HPPCM-V3 will definitely make the performance of indicators worse than HPPCM.

Table 10
Average rankings and adjusted p-values obtained of each algorithm through Friedman test on all test problems (HPPCM is the control method whose average ranking is 1.3889).

Algorithm	Avg Ranking	p-value	p-Hochberg	p-Holm	p-Bonf
DNSGA-II	4.777800	0.000200	0.000333	0.000800	0.001000
PPS	4.888900	0.000022	0.000111	0.000111	0.000111
PMS	2.666700	0.001000	0.001250	0.002000	0.005000
PBDMO	3.833300	0.004700	0.004700	0.004700	0.023500
DVA	3.444400	0.000200	0.000333	0.001000	0.000600

Table 11
Average rankings and adjusted p-values obtained of multi-model MOEAs through Friedman test on all test problems (HPPCM is the control method whose average ranking is 1.0).

Algorithm	Avg Ranking	p-value	p-Hochberg	p-Holm	p-Bonf
MO_Ring_PSO_SCD	2.722200	0.000022	0.000022	0.000022	0.000044
MOAGDE	2.277800	0.000022	0.000022	0.000044	0.000044

Table 12
Average rankings and adjusted p-values obtained of different HPPCM variants through Friedman test on all test problems (HPPCM is the control method whose average ranking is 1.0).

Algorithm	Avg Ranking	p-value	p-Hochberg	p-Holm	p-Bonf
HPPCM-V1	2.889000	0.000022	0.000022	0.000066	0.000066
HPPCM-V2	3.444000	0.000022	0.000022	0.000044	0.000066
HPPCM-V3	2.667000	0.000022	0.000022	0.000022	0.000066

5.6. Statistical analysis

One of the most frequent situations for statistical tests is in multi-data analysis, where the experimental results achieved by multiple algorithms in computational intelligence are required to be tested. We have the result of every algorithm for each test problem, but we are not allowed to make a pairwise comparison for each pair of algorithms without losing the control on the Family-Wise Error Rate (FWER) [51]. Statistical tests can perform two classes of analysis: pairwise comparisons and multiple comparisons. Both belong to non-parametric tests, and they are more robust and less sensitive to dirty data. Pairwise statistical procedures compare a pair of algorithms performance in several benchmarks. Therefore, multiple comparisons tests should be used to carry out a comparison that involves more than two algorithms. In this paper, a multiple comparisons test known as Friedman test [52] is performed to investigate the results obtained by different algorithms, with $(n_t, \tau_t) = (10{,}30)$. In the Friedman test, two hypotheses, the null hypothesis H_0 and the alternative hypothesis H_1 are defined. The null hypothesis is a statement of no significant differences between algorithms, whereas the alternative hypothesis represents the presence of a significant difference between algorithms [52]. One more critical term in the Friedman test is the p-value, which is the probability of obtaining a result at least as extreme as the one that was actually observed, assuming that H_0 is true [53]. We compute the p-value of every hypothesis using a normal approximation for a statistic that depends on the non-parametric test used. However, as we previously stated, singular p-values should not be used for multiple comparisons due to the loss of control over the FWER, so an adjustment is made to obtain the adjusted p-values [54]. Several post hoc procedures are available for the computation of the adjusted p-values. We finally choose Holm, Hochberg and Bonferroni [55].

To scientifically assess the performance of HPPCM and other state-of-the-art dynamic MOEAs, the Friedman test is applied to the obtained results, and the associated p-value is 0.005, indicating that there exit significant differences over the whole multiple comparisons among all algorithms. The average rankings of each algorithm on all test instances are listed in Table 10. Table 10 shows that our HPPCM gets the best average ranking (1.3889) outperforming PMS (2.6667), which are followed by DVA (3.4444) and PBDMO (3.8333), while DNSGA-II (4.7778) and PPS (4.8889) get worse rankings compared to their counterparts. As we can see in this table, the Friedman test shows a significant improvement of HPPCM over PMS, DVA, PBDMO, DNSGA-II and PPS for all considered post hoc procedures, with a level of significance of 0.05. The proposed algorithm is significantly better than other competitors with a level of significance of 0.005 for the Holm and Hochberg tests. In comparison with multi-modal MOEAs, It is shown by Tables 11 that HPPCM gets the best average ranking (1.0) outperforming MOAGDE (2.2778) followed by MO_Ring_PSO_SCD (2.7222). From the table, HPPCM is far better than MOAGDE and MO_Ring_PSO_SCD for all considered post hoc procedures, with a level of significance of 0.00005.

In addition, Tables 12 displays the average ranking of all HPPCM variants and the adjusted p-value with the application of three post hoc methods. Not surprisingly, HPPCM obtains the best average ranking (1.0) outperforming HPPCM-V3 (2.667) followed by HPPCM-V1 (2.889), while HPPCM-V3 (3.444) get the worse rankings compared to other variants. From the table, HPPCM is far better than its three variants for all considered post hoc procedures, with a level of significance = 0.00001. It shows the effect of the combination of the three sub-strategies is considerable to improve the algorithm's performance comprehensively. However, the ranking of the three variants also reflects the difference in their contribution to HPPCM. In particular, HPPCM's hybrid prediction strategy is more effective than the mutation strategy. Among them, the hybrid prediction strategy's second strategy (guiding individual-based prediction strategy) may be the most effective. The second strategy that improves the overall performance is its first strategy (center point-based prediction strategy). The role of the precision controllable mutation strategy is relatively minimal.

6. Conclusion

Responding quickly to the new environment and converging to the new POS while maintaining good distribution is the standard for evaluating the performance of a DMOEA. This paper proposes a new change response mechanism called HPPCM, consisting of a hybrid prediction strategy and a precision controllable mutation strategy. The hybrid prediction strategy focuses on convergence and responds quickly to predictable changes. The mutation strategy responds to unpredictable changes well and increases the diversity exploration of the population. Experimental results on several test suits have demonstrated that HPPCM combined with the RM-MEDA framework have obtained a set of well-distributed solutions which could converge to the POF or POS of most DMOPs in the new environment efficiently.

HPPCM has shown promising performance on various benchmark problems, but there is still room for further improvement. Our current actual algorithm version is not able to address combinatorial problems, HPPCM can only handle continuous problems. In the center point-based prediction strategy, the moving step size (α) can be determined by detecting the intensity of environmental changes. If the intensity of new environmental changes detected is the same as the last one, then the moving step size (α) can be set to be 1.0 definitely. If the former is much larger than the latter, then α will be r more than 1.0, like 1.5. Otherwise, we can set α less than 1, like 0.5. In this way, the solutions predicted by the center point-based prediction strategy will be more close to the new optima.

Declaration of Competing Interest

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service, and company that could be construed as influencing the position presented in, or the review of, the manuscript entitled “Combining a Hybrid Prediction Strategy and a Mutation Strategy for Dynamic Multiobjective Optimization.”

Acknowledgements

The authors wish to thank the support of the National Natural Science Foundation of China (Grant No. 61876164, 61772178), The MOEA Key Laboratory of Intelligent Computing and Information Processing, the Science and Technology Plan Project of Hunan Province (Grant No. 2016TP1020), the Provinces and Cities Joint Foundation Project (Grant No. 2017JJ4001), Science and Technology Planning Project of Guangdong Province of China (Grant No. 2017B010111005), the Hunan province science and technology project funds (Grant No. 2018TP1036).

References

[1] M. Farina, K. Deb, P. Amato, Dynamic multiobjective optimization problems: test cases, approximations, and applications, IEEE Trans. Evol. Comput. 8 (5) (2004) 425–442.

[2] Z. Zhang, Multiobjective optimization immune algorithm in dynamic environments and its application to greenhouse control, Appl. Soft Comput. 8 (2) (2008) 959–971.

[3] F.J. Gil-Gala, M.R. Sierra, C. Menca, R. Varela, Genetic programming with local search to evolve priority rules for scheduling jobs on a machine with time-varying capacity, Swarm Evol. Comput. (2021) 100944.

[4] R. Roy, J. Mehnen, Dynamic multi-objective optimisation for machining gradient materials, CIRP Ann. 57 (1) (2008) 429–432.

[5] L. Huang, I.H. Suh, A. Abraham, Dynamic multi-objective optimization based on membrane computing for control of time-varying unstable plants, Inf. Sci. 181 (11) (2011) 2370–2391.

[6] S. Jiang, S. Yang, Evolutionary dynamic multiobjective optimization: benchmarks and algorithm comparisons, IEEE Trans. Cybern. 47 (1) (2016) 198–211.

[7] L. Chen, L. Ding, X. Du, Genetic algorithm with particle filter for dynamic optimization problems, in: 2011 3rd International Conference on Computer Research and Development, vol. 1, IEEE, 2011, pp. 452–457.

[8] Y. Tian, R. Cheng, X. Zhang, F. Cheng, Y. Jin, An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility, IEEE Trans. Evol. Comput. 22 (4) (2017) 609–622.

[9] Y. Liu, N. Zhu, M. Li, Solving many-objective optimization problems by a Pareto-based evolutionary algorithm with preprocessing and a penalty mechanism, IEEE Trans. Cybern. (2020) 1–10.

[10] Y. Liu, Y. Hu, N. Zhu, K. Li, J. Zou, M. Li, A decomposition-based multiobjective evolutionary algorithm with weights updated adaptively, Inf. Sci. 572 (2021) 343–377.

[11] Q. Zhang, S. Yang, S. Jiang, R. Wang, X. Li, Novel prediction strategies for dynamic multiobjective optimization, IEEE Trans. Evol. Comput. 24 (2) (2019) 260–274.

[12] T.T. Nguyen, S. Yang, J. Branke, Evolutionary dynamic optimization: a survey of the state of the art, Swarm Evol. Comput. 6 (2012) 1–24.

[13] P. Rakshit, A. Konar, S. Das, Noisy evolutionary optimization algorithms–a comprehensive survey, Swarm Evol. Comput. 33 (2017) 18–45.

[14] C.A.C. Coello, G.T. Pulido, M.S. Lechuga, Handling multiple objectives with particle swarm optimization, IEEE Trans. Evol. Comput. 8 (3) (2004) 256–279.

[15] K. Deb, S. Karthik, et al., Dynamic multi-objective optimization and decision-making using modified NSGA-II: a case study on hydro-thermal power scheduling, in: Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization, Springer, 2007, pp. 803–817.

[16] C.-K. Goh, K.C. Tan, A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization, IEEE Trans. Evol. Comput. 13 (1) (2008) 103–127.

[17] A. Zhou, Y. Jin, Q. Zhang, A population prediction strategy for evolutionary dynamic multiobjective optimization, IEEE Trans. Cybern 44 (1) (2013) 40–53.

[18] A. Muruganantham, K.C. Tan, P. Vadakkepat, Evolutionary dynamic multiobjective optimization via kalman filter prediction, IEEE Trans. Cybern. 46 (12) (2015) 2862–2873.

[19] L. Cao, L. Xu, E.D. Goodman, C. Bao, S. Zhu, Evolutionary dynamic multiobjective optimization assisted by a support vector regression predictor, IEEE Trans. Evol. Comput. 24 (2) (2019) 305–319.

[20] Q. Zhang, A. Zhou, Y. Jin, RM-MEDA: a regularity model-based multiobjective estimation of distribution algorithm, IEEE Trans. Evol. Comput. 12 (1) (2008) 41–63.

[21] Q. Zhao, B. Yan, Y. Shi, M. Middendorf, Evolutionary dynamic multiobjective optimization via learning from historical search process, IEEE Trans. Cybern. (2021).

[22] S. Jiang, S. Yang, A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization, IEEE Trans. Evol. Comput. 21 (1) (2016) 65–82.

[23] M. Rong, D. Gong, W. Pedrycz, L. Wang, A multimodel prediction method for dynamic multiobjective evolutionary optimization, IEEE Trans. Evol. Comput. 24 (2) (2019) 290–304.

[24] D. Chen, F. Zou, R. Lu, X. Wang, A hybrid fuzzy inference prediction strategy for dynamic multi-objective optimization, Swarm Evol. Comput. 43 (2018) 147–165.

[25] Z. Liang, S. Zheng, Z. Zhu, S. Yang, Hybrid of memory and prediction strategies for dynamic multiobjective optimization, Inf. Sci. 485 (2019) 200–218.

[26] J. Zou, Q. Li, S. Yang, H. Bai, J. Zheng, A prediction strategy based on center points and knee points for evolutionary dynamic multi-objective optimization, Appl. Soft Comput. 61 (2017) 806–818.

[27] Q. Li, J. Zou, S. Yang, J. Zheng, G. Ruan, A predictive strategy based on special points for evolutionary dynamic multi-objective optimization, Soft Comput. 23 (11) (2019) 3723–3739.

[28] M. Rong, D. Gong, Y. Zhang, Y. Jin, W. Pedrycz, Multidirectional prediction approach for dynamic multiobjective optimization problems, IEEE Trans. Cybern. 49 (9) (2018) 3362–3374.

[29] I. Hatzakis, D. Wallace, Dynamic multi-objective optimization with evolutionary algorithms: a forward-looking approach, in: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, 2006, pp. 1201–1208.

[30] R. Rambabu, P. Vadakkepat, K.C. Tan, M. Jiang, A mixture-of-experts prediction framework for evolutionary dynamic multiobjective optimization, IEEE Trans. Cybern. 50 (12) (2019) 5099–5112.

[31] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, E. Tsang, Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization, in: Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization, Springer, 2007, pp. 832–846.

[32] S. Biswas, S. Das, S. Kundu, G.R. Patra, Utilizing time-linkage property in dops: an information sharing based artificial bee colony algorithm for tracking multiple optima in uncertain environments, Soft Comput. 18 (6) (2014) 1199–1212.

[33] R. Shang, L. Jiao, Y. Ren, L. Li, L. Wang, Quantum immune clonal coevolutionary algorithm for dynamic multiobjective optimization, Soft Comput. 18 (4) (2014) 743–756.

[34] F. Vavak, T.C. Fogarty, K. Jukes, A genetic algorithm with variable range of local search for tracking changing environments, in: International Conference on Parallel Problem Solving from Nature, Springer, 1996, pp. 376–385.

[35] Y.G. Woldesenbet, G.G. Yen, Dynamic evolutionary algorithm with variable relocation, IEEE Trans. Evol. Comput. 13 (3) (2009) 500–513.

[36] M. Cámara, J. Ortega, F. de Toro, A single front genetic algorithm for parallel multi-objective optimization in dynamic environments, Neurocomputing 72 (16–18) (2009) 3570–3579.

[37] G. Ruan, G. Yu, J. Zheng, J. Zou, S. Yang, The effect of diversity maintenance on prediction in dynamic multi-objective optimization, Appl. Soft Comput. 58 (2017) 631–647.

[38] K. Zhang, Z. Xu, S. Xie, G.G. Yen, Evolution strategy-based many-objective evolutionary algorithm through vector equilibrium, IEEE Trans. Cybern. (2020).

[39] K. Deb, R.B. Agrawal, et al., Simulated binary crossover for continuous search space, Complex Syst. 9 (2) (1995) 115–148.

[40] N. Kambhatla, T.K. Leen, Dimension reduction by local principal component analysis, Neural Comput. 9 (7) (1997) 1493–1516.

[41] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.

[42] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results, Evol. Comput. 8 (2) (2000) 173–195.

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

[44] Z. Peng, J. Zheng, J. Zou, M. Liu, Novel prediction and memory strategies for dynamic multiobjective optimization, Soft Comput. 19 (9) (2015) 2633–2653.

[45] J. Zheng, Y. Zhou, J. Zou, S. Yang, J. Ou, Y. Hu, A prediction strategy based on decision variable analysis for dynamic multi-objective optimization, Swarm Evol. Comput. 60 (2021) 100786.

[46] M.D. McKay, R.J. Beckman, W.J. Conover, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics 42 (1) (2000) 55–61.

[47] Y. Yuan, H. Xu, B. Wang, B. Zhang, X. Yao, Balancing convergence and diversity in decomposition-based many-objective optimizers, IEEE Trans. Evol. Comput. 20 (2) (2015) 180–198.

[48] H. Li, Q. Zhang, Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II, IEEE Trans. Evol. Comput. 13 (2) (2008) 284–302.

[49] C. Yue, B. Qu, J. Liang, A multiobjective particle swarm optimizer using ring topology for solving multimodal multiobjective problems, IEEE Trans. Evol. Comput. 22 (5) (2017) 805–817.

[50] S. Duman, M. Akbel, H.T. Kahraman, Development of the multi-objective adaptive guided differential evolution and optimization of the MO-ACOPF for wind/PV/tidal energy sources, Appl. Soft Comput. 112 (2021) 107814.

[51] J. Carrasco, S. García, M. Rueda, S. Das, F. Herrera, Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: practical guidelines and a critical review, Swarm Evol. Comput. 54 (2020) 100665.

[52] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18.

[53] V.K. Patel, V.J. Savsani, A multi-objective improved teaching–learning based optimization algorithm (MO-ITLBO), Inf. Sci. 357 (2016) 182–200.

[54] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power, Inf. Sci. 180 (10) (2010) 2044–2064.

[55] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18.