

Novel Prediction Strategies for Dynamic Multiobjective Optimization

Qingyang Zhang, Shengxiang Yang[✉], Senior Member, IEEE, Shouyong Jiang, Ronggui Wang, and Xiaoli Li[✉]

Abstract—This paper proposes a new prediction-based dynamic multiobjective optimization (PBDMO) method, which combines a new prediction-based reaction mechanism and a popular regularity model-based multiobjective estimation of distribution algorithm (RM-MEDA) for solving dynamic multiobjective optimization problems. Whenever a change is detected, PBDMO reacts effectively to it by generating three subpopulations based on different strategies. The first subpopulation is created by moving nondominated individuals using a simple linear prediction model with different step sizes. The second subpopulation consists of some individuals generated by a novel sampling strategy to improve population convergence as well as distribution. The third subpopulation comprises some individuals generated using a shrinking strategy based on the probability distribution of variables. These subpopulations are tailored to form a population for the new environment. The experimental results carried out on a variety of bi- and three-objective benchmark functions demonstrate that the proposed technique has competitive performance compared with some state-of-the-art algorithms.

Index Terms—Dynamic multiobjective optimization, nondominated sorting, prediction-based reaction, probability distribution.

I. INTRODUCTION

MANY real-world optimization problems, such as scheduling [36], planning [37], constrained optimization [39], electromagnetic micromirrors [44], dynamic subset sum [46], weapon selection [47], and

Manuscript received December 24, 2018; revised April 26, 2019; accepted June 1, 2019. Date of publication June 13, 2019; date of current version March 31, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61673331, Grant 61873006, Grant 61473034, and Grant 61673053, in part by the Shenzhen Peacock Plan under Grant KQTD2016112514355531, in part by the Beijing Science and Technology Major Project under Grant Z18110003118012, and in part by the National Key Research and Development Project under Grant 2018YFC1602704 and Grant 2018YFB1702704. (*Corresponding author: Shengxiang Yang*.)

Q. Zhang is with the School of Computer Science and Technology, Jiangsu Normal University, Xuzhou, China (e-mail: hgd_qy@126.com).

S. Yang is with the School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, U.K., and also with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: syang@dmu.ac.uk).

S. Jiang is with the School of Computer Science, University of Lincoln, Lincoln LN6 7TS, U.K. (e-mail: math4neu@gmail.com).

R. Wang is with the School of Computer and Information, Hefei University of Technology, Hefei 230009, China (e-mail: wangrgui@hfut.edu.cn).

X. Li is with the Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China (e-mail: lixiaolibjut@bjut.edu.cn).

This paper has supplementary downloadable material available at <http://ieeexplore.ieee.org>, provided by the author.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TEVC.2019.2922834

machine learning [40], [41], require to be solved in dynamic or uncertain environments [42]. This kind of problems are called dynamic multiobjective optimization problems (DMOPs), which are characterized by some changes that may occur in the Pareto-optimal set (POS) or Pareto-optimal front (POF), objective functions, constraints, or other parameters. Sometimes, the changes between two consecutive moments are not significant and have some similarities. It is not necessary to restart the relevant programs. Therefore, the study on DMOPs is meaningful in solving practical applications efficiently. Recently, DMOPs is one of the hot topics that has attracted increasing attention in the field of evolutionary computation.

Although there may exist different classes of dynamic optimization problems according to [14] and [43], this paper considers the following mathematical form of DMOPs:

$$\min_{x \in \Omega} F(x, t) = (f_1(x, t), f_2(x, t), \dots, f_m(x, t))^T \quad (1)$$

where $\Omega = \prod_{i=1}^D [L_i, U_i] \subset R^D$ is the feasible area of the decision space, and F consists of m time-varying objective functions. $x = (x_1, x_2, \dots, x_D)$ defines the decision vector involving D variables, and t is the time instant of the problem. L_i and U_i represent the lower and upper bounds of the i th variable x_i , respectively.

Different from single optimization problems (SOPs), multiobjective optimization problems (MOPs) usually involves at least two conflicting optimization objectives that need to be optimized simultaneously. Due to multiobjectivity, the goal of solving MOPs is not to find a single optimal solution but to find a set of tradeoff solutions. When an MOP involves time-dependent components, it can be regarded as a DMOP. Compared with stationary MOPs, DMOPs are more difficult due to the nature of dynamics, which may cause the change of POS in the decision space or POF in the objective space over time. DMOPs are frequently reported in real-world applications like engineering [38] and water distribution systems [45], and there is a great need of effective approaches to solve them. Due to great importance and economic relevance, there are an increasing number of research interests in dynamic optimization [10], [15], [34].

To solve DMOPs, most of the existing dynamic multiobjective evolutionary algorithms (DMOEAs) directly adopt the techniques from static multiobjective evolutionary algorithms (MOEAs), assuming that a DMOP can be divided into a sequence of MOPs over time. That is, solving a DMOP can be regarded as adopting MOEAs to solve a sequence of correlated MOPs and obtain corresponding

POSSs or POFs [48]. However, an efficient DMOEA should not only have a good MOEA, but also have a good reaction mechanism, which is able to keep a good balance between convergence and diversity over time [10]. Since the changes are unknown in advance, a DMOEA may not have enough time for searching optimal solutions before a change occurs. So, it is necessary for a DMOEA to have the fast convergence performance. Meanwhile, whenever a change occurs, previously obtained solutions may not remain optimal for the new environment [50]. So, how to address diversity loss is also important. In addition, change detection techniques are needed since they can prevent the algorithm from being misled in the optimization process.

Based on above considerations, this paper proposes a new prediction-based DMOEA, called PBDMO, which combines several strategies to generate a high-quality population for new environments. First, a new prediction strategy is proposed to relocate nondominated solutions of the previous environment diversely around the estimated location of the new POF after a change occurs. Second, a classification strategy is proposed to classify decision variables into principal and nonprincipal groups in order to effectively generate well-diversified solutions. Third, the probability distribution of decision variables is used to generate another group of good solutions. Solutions generated from these three strategies undergo nondominated sorting and then form a population for the new environment. The experimental results indicate that PBDMO has promising tracking ability for time-varying POFs or POSSs. Overall, the main contributions of this paper include the following.

- 1) New strategies to relocate nondominated solutions for previous environments well to the new POF.
- 2) Classification-based sampling to enhance population diversity.
- 3) Probability-based space shrinkage to generate high-quality solutions.
- 4) Systematic studies that demonstrate the effectiveness of the proposed strategies.

As a result, the proposed PBDMO has two main advantages over the considered state-of-the-art algorithms as demonstrated by extensive empirical studies. First, PBDMO is able to generate a well-diversified and well-positioned population so as to track the moving POF closely and with good solution distribution. Second, PBDMO is more robust than the other algorithms as its performance does not fluctuate widely across a variety of problems with diverse dynamic characteristics.

The rest of this paper is outlined as follows. Section II gives the basic framework and related work about DMOEAs. Section III describes the proposed PBDMO algorithm in detail. In Section IV, the performance of the proposed algorithm is validated and analyzed based on a comprehensive set of benchmark functions. Section V presents further analysis on the proposed algorithm. Section VI concludes this paper.

II. RELATED WORK

In recent years, much effort has been devoted to designing efficient DMOEAs. A widely used framework of DMOEAs in the literature is shown in Algorithm 1. As shown in

Algorithm 1 Basic Framework of DMOEA

```

1: Initialize time instance  $t = 0$ ;
2: Generate an initial population  $Pop_t$ ;
3: while The termination criterion is not satisfied do
4:   Detect change;
5:   if Change detected then
6:     Optimize the population using DMOEA;
7:   else
8:     Optimize the population using MOEA;
9:   end if
10: end while

```

this framework, the whole procedure of solving DMOPs contains two main components: 1) change detection and 2) multiobjective algorithms, including DMOEAs and MOEAs.

A. Change Detection

Change detection is an important part in dynamic optimization since it identifies whether the change has appeared or not and determines whether a response should be made [9]. Changes are detected by either re-evaluating solutions [11], [12], [33] or checking population statistical information [20]. Each method has its advantages and disadvantages for different DMOPs. Most existing studies focus on the former, which is easy to implement but likely sensitive to noise. In contrast, the latter is robust to noise, but it needs some additional parameters.

B. Dynamic Multiobjective Optimization Algorithms

Due to the frequency or severity of change, there may be limited computational resources available for evolution before another change arrives, and changes may occur in various forms. This provides different levels of challenges. So, high-quality algorithms are needed to cope with the change if detected. A good dynamic multiobjective algorithm should consider two important elements: 1) diversity and 2) convergence. The former helps the algorithm avoid local optima whereas the latter determines whether the algorithm tracks changes rapidly. How to design an algorithm that can balance the diversity and convergence for tracking the uncertain POF or POS closely is always a formidable task. Most existing algorithms can be categorized into the following four different classes: 1) diversity-based algorithms; 2) memory-based algorithms; 3) multipopulation-based algorithms; and 4) prediction-based algorithms.

Diversity-based algorithms aim to increase/maintain population diversity in the event of environmental changes. Many diversity maintenance methods have been proposed in recent years. Hyper-mutation and random immigration are effective approaches to maintain population and there have been applied to the nondominated sorting genetic algorithm II (NSGA-II) [7], resulting in two dynamic variants, i.e., DNSGA-II-A and DNSGA-II-B for DMOPs. However, these approaches seem ineffective in complex dynamic environments, particularly when changes are not predictable [28].

Memory-based algorithms transfer knowledge of previous environments to new environments. Generally, some promising solutions are stored in memory and reused after a change

is detected. In [10], a co-evolutionary multiobjective algorithm hybridizes competitive and cooperative mechanisms to solve the DMOPs, and the out-of-date archived solutions are replaced by an external population. Azzouz *et al.* [2] proposed an adaptive hybrid population management strategy using memory, local search, and random strategies. The main idea is that according to the change severity, the number of memory and random solutions can be tuned when facing changing environments. Jiang and Yang [14] used a steady-state manner to respond to changes. The main feature of this method is that when a change occurs, a portion of recorded outdated solutions with good distribution are reused based on the information collected from previous environments. The memory-based algorithms may perform effectively in problems with periodical changing environments.

Multipopulation-based algorithms use several subpopulations to explore different regions in the feasible space for diversity maintenance [3]. This type of algorithm performs effectively in problems with multipeak problems. Goh and Tan [10] introduced multipopulation strategies to tackle changes in multiobjective dynamic environments and observed good performance.

Prediction-based algorithms refer to the reuse of information collected from previous environments and POS/POF prediction for future environments. Compared with other algorithms mentioned before, prediction-based algorithms can accommodate the changes in advance, which may be the reason for popularity in DMOEAs. For example, in [22], Kalman filter is combined with evolutionary algorithms to solve DMOPs. Population prediction strategy (PPS) proposed by Zhou *et al.* [34] is used to predict the whole population. The univariate auto-regression (AR) model was used to predict the next locations of the centroid and manifold of the whole population based on the information of previous environments. In [19], [24], and [35], other prediction techniques were proposed by combining some special points obtained from previous environments, such as knee points, center points, and boundary points, with certain search methods so as to predict the new POS/POF accurately.

C. Multiobjective Optimization Algorithms

MOEAs are an important part or the core of solving MOPs. Technically, any existing MOEAs can be directly applied to evolve the population for the period that the environment stays unchanged. Existing DMOEAs show promising performance in solving different problems, but most of them are designed by considering different machine learning models [22], [34]. The internal properties of decision variables are usually neglected, which will affect the accuracy and quality of discovered solutions. Motivated by this, this paper proposes three strategies that take the distribution and classification of variables into account to generate high-quality populations for new environments. The proposed PBDMO algorithm is described in detail below.

III. PROPOSED PBDMO ALGORITHM

The basic framework of the PBDMO algorithm is presented in Algorithm 2. As shown, the proposed algorithm consists of

Algorithm 2 Overall Framework of PBDMO

```

1: Initialize parameter settings for the algorithm;
2: Initialize and evaluate population ( $Pop_0$ ) and set  $Iter := 0$ ;
3: while the termination criterion is not met do
4:   if change detected then
5:     Generate the first sub-population ( $SubPop_1$ ) resulting from
       predicting a new non-dominated set;
6:     Generate the second sub-population ( $SubPop_2$ ) by the
       proposed sampling strategy;
7:     Generate the third sub-population ( $SubPop_3$ ) according to
       Algorithm 5;
8:     Merge these sub-populations  $MixPop := SubPop_1 \cup$ 
        $SubPop_2 \cup SubPop_3$ ;
9:     Obtain a population from  $MixPop$  by non-dominated
       sorting;
10:    else
11:      Optimize the population using RM-MEDA;
12:    end if
13:     $Iter := Iter + 1$ ;
14: end while

```

three key strategies to obtain a new population in response to changes. The main idea of this algorithm is to use history information, variable relationship, and variable probability distribution to generate a population that approximates the new POS/POF as much as possible. Like other predicted algorithms, our hypothesis is that there is sort of similarity between two consecutive changes. These strategies are described as follows.

A. Predicting the Nondominated Set

From the statistical point of view, the geometric center is an important characteristic and can be used to represent the changing trend of population to some extent. Here, we compute the moving direction of the center point of the last two consecutive populations and use it to predict the position of the nondominated members of current population in the new environment.

Suppose that Pc_t is the centroid of population (Pop_t) and Pos_t is the nondominated set of Pop_t at the time t . Pc_t can be calculated as follows:

$$Pc_t = \frac{\sum_{x_t \in Pop_t} x_t}{|Pop_t|} \quad (2)$$

where $|Pop_t|$ is the population size, $x_t = (x_t^1, x_t^2, \dots, x_t^D)$ denotes the decision vector of a solution at time t . The moving direction (Dir_t) of the center point at time t is calculated by

$$Dir_t = Pc_t - Pc_{t-1}. \quad (3)$$

Then, the new position of members in Pos_t at time $t+1$ can be obtained by Dir_t and Pos_t according to the following formula:

$$Pos_{t+1} = Pos_t + Dir_t \times rs \quad (4)$$

where rs refers to the moving step-size along the moving direction Dir_t . Here, three different values of rs (i.e., 0.5, 1.0, and 1.5) can be used, which represents a small, medium, and large movement of Pos_t , respectively. Fig. 1 illustrates the prediction process.

As shown in Fig. 1, Pc_t and Pc_{t-1} (black points) are utilized to obtain Dir_t . Pos_t moves to three different regions described

Algorithm 3 Predicting the Nondominated Set

- 1: Retrieve the populations Pop_t and Pop_{t-1} at time t and $t-1$, respectively;
- 2: Calculate the population centers according to Eq. (2);
- 3: Predict the moving direction according to Eq. (3);
- 4: Generate three sub-populations Pos_{t+1}^1 , Pos_{t+1}^2 and Pos_{t+1}^3 using Eq. (4) with different rs values;
- 5: Save the sub-populations to $SubPop_1$.

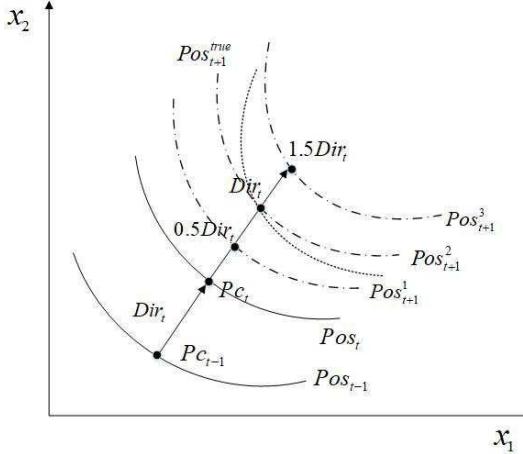


Fig. 1. Illustration of predicting nondominated solutions.

by Pos_{t+1}^1 , Pos_{t+1}^2 , and Pos_{t+1}^3 using the suggested rs values. A combination of these three solution predictions is more likely to approximate the true POS of the population (Pos_{t+1}^{true}) at time $t+1$. Algorithm 3 provides the implementation of this prediction strategy.

However, two questions arise here—how many step-sizes and how big the step-sizes should be used. The three step-sizes and their values (0.5, 1.0, and 1.5) are used due to two reasons. First, the three step-sizes classify possible changes into three levels, which is a common practice in fuzzy systems [23]. Here, $rs = 1$ suggests the use of the same step-size as for the previous environment, meaning that the change is similar to the previous change (medium changes). $rs = 0.5$ and $rs = 1.5$ mean small and large changes, respectively, and they are chosen not only for simplicity but also by sensitivity analysis as will be detailed later. It is also possible to use much smaller or bigger values for rs to represent small or large changes, but this would betray our assumption that there exists similarities between two consecutive changes. Second, three levels of changes are considered for the sake of computational efficiency, although we recognize that the prediction can be improved with more detailed classification. The three step sizes allow to predict a possible range of the new POS and increase the probability of finding new solutions. Sensitivity analysis of these rs values can be found later in Section V.

B. Novel Sampling Strategy

Introducing some sampling points from the search space into the new population can potentially increase the population diversity [10], [14]. However, the contribution of sampling points to convergence is often uncertain. Here, we

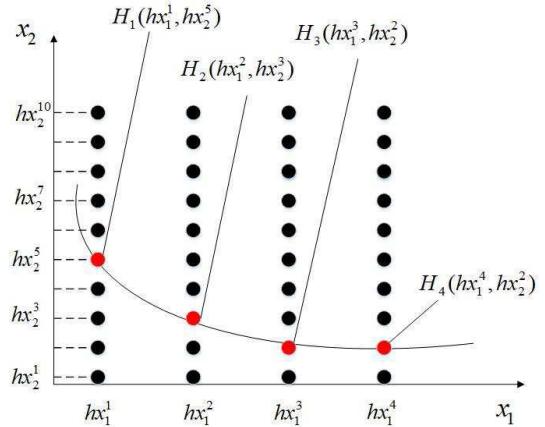


Fig. 2. Illustration of the sampling strategy in the 2-D space.

develop a novel sampling strategy that potentially enhances the convergence in addition to diversity.

To aid effective sampling, the decision vector is divided into two subvectors: 1) the principal and 2) nonprincipal variables. Nonprincipal variables can be assumed to depend on principal ones. The assumption holds to some degree as, from the mathematical point of view, objective functions present the dependency between them even if there is no dependency between variables.

The determination of principal variables is not difficult. To ease the task, we just need to identify the most principal variable and regard the others as nonprincipal variables. The identification uses a very simple approach, that is, the most principal variable is the one having the largest standard deviation (the first encountered is chosen if there is a tie) in the nondominated set. Let x_d be the identified most principal variable, N_1 evenly spaced points are sampled for x_d . For any nonprincipal variable x_k ($k \in \{1, \dots, D\} \setminus \{d\}$), N_2 sampling points are obtained by

$$hx_k^i = L_k + i \times \frac{U_k - L_k}{N_2}, \quad (i = 1, \dots, N_2) \quad (5)$$

where L_k and U_k are the lower and upper bounds of x_k , respectively. This results in a total of $N_1 \times N_2$ sampling points $\bar{x} = (x_d^i, hx_k^j)$, for $i = 1, \dots, N_1$ and $j = 1, \dots, N_2$.

After that, all the sampled points are evaluated and nondominated solutions are identified by nondominated sorting. These nondominated solutions form the second subpopulation in response to the environmental change. Fig. 2 illustrates the proposed sampling strategy in the 2-D space, where four points are uniformly sampled from the principal variable x_1 , nine points from the nonprincipal variable x_2 , and nondominated points are marked in red. Algorithm 4 shows the steps of creating a subpopulation by the proposed sampling strategy.

The setting of N_1 and N_2 is problem-specific. For example, it is expected to use a large number of sampling points for problems with a large number of variables and a small number of sampling points for problems with a few variables. Setting large values for them should help generate high-quality subpopulation but would lead to high-computational costs. Therefore, for simplicity, N_2 is recommended to couple with the number of variables, e.g., $N_2 = D$, and N_1 is set to a

Algorithm 4 Sampling Strategy

- 1: Sample uniformly points for the first variable according to Eq. (5);
- 2: For each sampled value of the first variable, generate a set of uniformly distributed points of the other variables according to Eq. (5);
- 3: Evaluate these solutions and select the non-dominated solutions using non-dominated sorting;
- 4: Save the non-dominated solutions to $SubPop_2$.

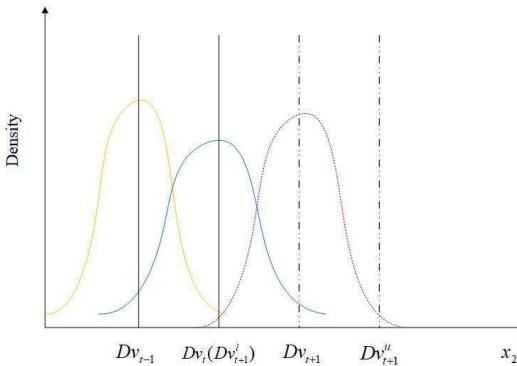


Fig. 3. Example of variable distribution and predicted variable bounds.

value such that the total number of sampling points generated from this strategy is less than half of the population, i.e., $N_1 N_2 = N_1 D \leq N/2$.

C. Shrinking Strategy

The chance of generating high-quality solutions improves if we can estimate to some extent where the optimal solutions probably reside. Here, we propose a shrinking strategy to refine the likely optimal range of variables, which is essentially a reduction of the decision space, based on the probability distribution of variables. The strategy is mainly applied to non-principal variables mentioned in Section III-B as the optimal range of principal variables is highly uncertain and an inaccurate shrinkage may generate more negative impact on principal variables than on nonprincipal variables. The range shrinking strategy works for a given nonprincipal variable x_i as follows. First, the maximum probable value of x_i at time t , denoted Dv_t , is calculated using the probability estimation strategy [17] on the corresponding nondominated set. Second, the maximum probable value of x_i for a new environment Dv_{t+1} at time $t+1$ can be predicted by using the last two consecutive Dv_{t-1} and Dv_t as follows:

$$Dv_{t+1} = Dv_t + (Dv_t - Dv_{t-1}). \quad (6)$$

Then, the upper bound of x_i at time $t+1$ is estimated as

$$Dv_{t+1}^u = Dv_{t+1} + (Dv_{t+1} - Dv_t). \quad (7)$$

For simplicity, the lower bound of x_i , denoted Dv_{t+1}^l , at time $t+1$ is made identical to Dv_t . Fig. 3 illustrates the construction of a shrank range for nonprincipal variable x_2 . At the end, the estimated range of x_i for the new environment is $[Dv_{t+1}^l, Dv_{t+1}^u]$. Combining the estimated range for all nonprincipal variables and the original range of the

Algorithm 5 Shrinking Strategy

- 1: Find the populations (Pop_t and Pop_{t-1}) at time t and $t-1$, respectively;
- 2: Compute the maximum probable value for each non-principal variable x_i at time $t-1$ and t using the probability density estimation strategy [17];
- 3: Estimate the new maximum probable value for each x_i at time $t+1$ by Eq. (6);
- 4: Calculate the bounds of x_i by Eq. (7);
- 5: Create a sub-population $SubPop_3$ by sampling from the shrank decision space.

principal variable, we have a shrank decision space $\Omega = [L_1, U_1] \cup \prod_{i=2}^D [Dv_{t+1}^l, Dv_{t+1}^u] \subset R^D$, from which another sub-population ($SubPop_3$) of solutions will be uniformly sampled. The implementation of this strategy is shown in Algorithm 5.

IV. EXPERIMENTAL STUDY**A. Test Instances**

The performance of the proposed approach is examined on a recently proposed DF test suite [15], which has nine bi-objective and five tri-objective benchmark functions with diverse properties, such as variable linkage, disconnectivity, irregular POF shapes, and time-dependent geometries. The DF test suite consists of functions taken from other test suites, e.g., FDA [8], dMOP [10], ZJJZ [33], and JY [13]. The time instance t used in the test suite is given as $t = (1/n_t)\lfloor(\tau/\tau_t)\rfloor$, where n_t , τ , and τ_t are the severity of change, the number of iterations, and the frequency of change, respectively. The dimension (D) of benchmark functions is set to 10. The definition of these test instances can be found in [15].

B. Performance Indicators

Performance indicators play an important role in assessing the performance of algorithms. In this paper, we adopts the following three performance indicators.

1) *Mean Inverted Generational Distance*: The mean inverted generational distance (MIGD), a widely adopted measure [30], [32], [34], is mainly employed to measure the convergence and diversity of solutions obtained by an algorithm. Suppose that POF_t^* is a set of uniformly distributed solutions in the true POF and POF_t^{ob} is a POF approximation, at time t , IGD can be calculated as

$$IGD(POF_t^*, POF_t^{ob}) = \frac{\sum_{g \in POF_t^*} d(g, POF_t^{ob})}{|POF_t^*|} \quad (8)$$

where $d(g, POF_t^{ob})$ refers to the minimum Euclidian distance between g and the points in POF_t^{ob} , and $|POF_t^*|$ is the number of solutions in POF_t^* . Then, the MIGD can be computed as

$$MIGD = \frac{\sum_{t \in T} IGD(POF_t^*, POF_t^{ob})}{|T|} \quad (9)$$

where T is a set of discrete time points and $|T|$ is the total number of changes in a run.

2) *Mean Schott's Spacing Metric*: The Schott's spacing (SP) metric [10], [21], [25] aims to evaluate the distribution of POF_t^{ob} . The following gives the expression formula:

$$\text{SP}(\text{POF}_t^{ob}) = \sqrt{\frac{1}{|\text{POF}_t^{ob}| - 1} \left(\sum_{i=1}^{|\text{POF}_t^{ob}|} (D_i - \bar{D})^2 \right)} \quad (10)$$

where D_i is the Euclidean distance between the i th point in POF_t^{ob} and its nearest point in POF_t^{ob} . \bar{D} represents the average value of D_i . The mean Schott's spacing metric (MSP) can be defined as follows:

$$\text{MSP} = \frac{\sum_{t \in T} \text{SP}(\text{POF}_t^{ob})}{|T|}. \quad (11)$$

3) *Hypervolume Metric*: Hypervolume (HV) [34] is a widely used indicator in multiobjective optimization. The calculation of HV requires a reference point, which is usually defined as a vector dominated by any points in the true POF. A larger HV value indicates a better approximation. HV is defined as follows:

$$\text{HV}_t = \text{HV}(\text{POF}_t^{ob}) \quad (12)$$

where $\text{HV}(\text{POF}_t^{ob})$ refers to the HV [27] of set POF_t^{ob} . The reference point for the computation of HV is $(z_j + 0.5, j = 1, \dots, m)$, where z_j is the maximum value of the j th objective of true POF. The mean HV (MHV) can be calculated as follows:

$$\text{MHV} = \frac{\sum_{t \in T} \text{HV}_t}{|T|}. \quad (13)$$

C. Compared Algorithms

In order to validate the performance of our proposed method, several existing approaches are selected as the comparison algorithms. A brief description of these algorithms and their parameter settings is given as follows.

1) *Population Prediction Strategy (PPS)*: The main idea of PPS is dividing the optimal solutions into two parts: 1) population center and 2) manifold. An AR model is adopted to predict the next population center based on a time series of history population centers. Similarly, history manifolds are also used to predict the new manifold. Then, a new population will be generated based on the predicted population center and manifold [34].

2) *Dynamic Version of NSGA-II (DNSGA-II)*: DNSGA-II is a dynamic version of NSGA-II by making some changes to the original NSGA-II. Two versions are available: 1) NSGA-II-A and 2) NSGA-II-B. The former is formed by replacing a small portion of the population with random solutions, and it is suitable for solving dynamic problems with severe changes. The latter is similar to the former except that the replacement uses mutated solutions of existing individuals, and works well in solving problems with small changes. Both of them are used in this paper [7].

3) *First-Order Difference Model-Based MOEA/D*: First-order difference model-based MOEA/D (MOEA/D-FD) [5] utilizes history information to predict the location of the new POS after a change is detected. The new population is composed of two kinds of solutions: 1) the old solutions and 2) the predicted ones. The movement of population centroid defines a predicted direction. To make the new population diversified, evenly distributed individuals selected from the previous population are used in the prediction.

D. Parameter Settings

The parameters of the MOEAs considered in the experiment were referenced from their original papers. Some key parameters in these algorithms were set as follows.

1) *Population Size*: The population size (N) in all the algorithms was set to 100. Around 1000 points were uniformly sampled from the true POF for computing the performance metrics in both bi- and three-objective cases.

2) *Other Parameters*: All the parameters in the compared algorithms used the same settings as in their original studies.

a) *Parameters in PBDMO*: The second sampling strategy obtains $N_1 N_2$ sampling points from $L_1 + \{0.2, 0.4, 0.6, 0.8\} * (U_1 - L_1)$, and the size of SubPop_3 was set to $0.3N$.

3) *Stopping Criterion and the Number of Executions*: Each algorithm terminates after a prespecified number of generations and should cover all possible changes. To minimize the effect of static optimization, we gave 50 generations for each algorithm before the first change occurs. The total number of generations was set to $3n_t \tau_t + 50$, which ensures there are $3n_t$ changes during the evolution. Additionally, each algorithm was executed 20 independent times on each test instance.

4) *Change Detection*: For all the algorithms, a maximum number of 10% population members are re-evaluated for change detection.

E. Experimental Results

To study the impact of the change frequency on algorithms' ability in dynamic environments, the severity of change (n_t) was fixed to 10, and the frequency of change (τ_t) was set to 5, 10, and 20, respectively. The statistical results are presented in Tables I–III, where the best values obtained by one of the five algorithms are highlighted in bold face.

Table I lists the MIGD values of all the algorithms. It can be seen that PBDMO obtains the best results on the majority of 14 test instances. For three functions, DF2, DF11, and DF12, the difference between PBDMO and the best method is very small. The considerably small IGD values of PBDMO suggest the POF approximations are close to the true POF, showing its excellent tracking ability. This demonstrates that the proposed prediction technique can generate good populations for varying environments. Even if there exist different levels of change severity, PBDMO can still achieve better performance than other algorithms in these test instances.

The MHV values for the algorithms are presented in Table II. One can see that the MHV results are highly consistent with the MIGD ones. PBDMO has significantly better

TABLE I
MEAN AND STANDARD DEVIATION VALUES OF MIGD OBTAINED BY FIVE ALGORITHMS

Fun.	(τ_t, n_t)	MOEA/D-FD	DNSGA-II-A	DNSGA-II-B	PPS	PBDMO
DF1	(5,10)	1.3377e-2(2.4536e-4)	1.2815e-1(1.0888e-2)	4.0948e-1(6.1782e-2)	6.9803e-1(1.4463e-1)	6.5907e-3(4.1274e-4)
	(10,10)	9.5248e-3(1.5828e-4)	5.8375e-2(5.4323e-3)	8.0500e-2(7.9357e-3)	3.4628e-1(4.2348e-2)	5.5687e-3(2.1958e-4)
	(20,10)	7.3724e-3(8.2575e-5)	1.4867e-2(5.9269e-4)	1.3998e-2(5.3880e-4)	1.3402e-1(2.0495e-2)	5.0667e-3(1.5401e-4)
DF2	(5,10)	1.9599e-2(7.3670e-4)	8.6463e-2(6.0977e-3)	3.2018e-1(3.6745e-2)	4.9000e-1(1.1130e-1)	7.8786e-2(5.7042e-3)
	(10,10)	1.0959e-2(1.7838e-4)	4.3680e-2(8.1074e-3)	1.2025e-1(9.6012e-3)	2.2890e-1(3.7999e-2)	3.4739e-2(3.3543e-3)
	(20,10)	7.5063e-3(8.4108e-5)	1.4701e-2(3.0683e-3)	3.9591e-2(6.1428e-3)	8.3623e-2(1.3538e-2)	1.1780e-2(1.1659e-3)
DF3	(5,10)	4.7593e-2(1.3488e-2)	3.9299e-1(4.4728e-2)	3.5008e-1(4.5990e-2)	4.4112e-1(1.5743e-1)	1.6751e-2(6.7187e-3)
	(10,10)	3.3634e-2(1.1575e-3)	2.8130e-1(3.0150e-2)	2.8233e-1(3.0376e-2)	2.0300e-1(1.8595e-1)	1.1915e-2(5.9480e-3)
	(20,10)	2.6130e-2(1.6133e-3)	2.3815e-1(2.1797e-2)	2.3486e-1(2.6841e-2)	9.4226e-2(1.2283e-1)	9.0913e-3(5.7123e-3)
DF4	(5,10)	1.1985e-1(2.1803e-3)	1.1382e-1(6.8154e-3)	1.0811e-1(6.3892e-3)	3.1145e-1(5.9685e-2)	7.4666e-2(3.8800e-3)
	(10,10)	1.0672e-1(1.3873e-3)	7.6643e-2(3.2921e-3)	7.5365e-2(3.0515e-3)	1.1851e-1(1.0833e-2)	7.1674e-2(3.0848e-3)
	(20,10)	9.9324e-2(1.4062e-3)	6.2710e-2(1.6365e-3)	6.2451e-2(1.3417e-3)	8.2339e-2(6.6108e-3)	7.0900e-2(2.9866e-3)
DF5	(5,10)	1.9425e-2(4.9644e-4)	2.6918e-1(3.6061e-2)	3.3045e-1(3.5508e-2)	1.2102e+0(2.3246e-1)	8.4182e-3(1.0122e-3)
	(10,10)	1.4655e-2(2.4786e-4)	5.8123e-2(8.4188e-3)	5.1666e-2(3.9707e-3)	3.4648e-1(1.2439e-1)	6.5586e-3(7.7474e-4)
	(20,10)	1.1270e-2(1.6130e-4)	1.3655e-2(3.8585e-4)	1.3388e-2(3.9506e-4)	7.4278e-2(2.0116e-2)	5.6538e-3(6.2324e-4)
DF6	(5,10)	5.2019e+0(6.2903e-1)	7.2179e+0(4.2587e-1)	1.4465e+1(6.6272e-1)	1.0649e+1(1.0992e+0)	5.8358e-1(3.7318e-2)
	(10,10)	4.8674e+0(5.4767e-1)	3.4813e+0(2.2456e-1)	9.0723e+0(7.3310e-1)	7.1312e+0(8.8867e-1)	4.8638e-1(1.8615e-2)
	(20,10)	4.8855e+0(6.8135e-1)	1.3120e+0(2.9491e-1)	5.9070e+0(8.5389e-1)	3.6264e+0(7.1611e-1)	4.0620e-1(2.5600e-2)
DF7	(5,10)	1.1113e-1(1.2668e-2)	2.7914e-1(5.3170e-2)	3.5985e-1(7.8006e-2)	1.7931e-1(7.6929e-2)	1.1882e-2(6.2114e-4)
	(10,10)	9.8690e-2(1.3214e-2)	2.3844e-1(3.6387e-2)	2.9375e-1(7.4635e-2)	6.0637e-2(1.6953e-2)	9.6751e-3(3.2085e-4)
	(20,10)	6.3632e-2(1.0840e-2)	1.9767e-1(3.6864e-2)	1.9669e-1(3.4827e-2)	2.2886e-2(6.0030e-3)	8.3712e-3(1.9506e-4)
DF8	(5,10)	3.5901e-2(2.1273e-3)	1.4797e-2(1.0676e-3)	1.6316e-2(3.5859e-3)	3.0128e-2(1.9313e-3)	7.6445e-3(5.4946e-4)
	(10,10)	2.9565e-2(1.8211e-3)	1.1639e-2(7.7879e-4)	1.1367e-2(5.2164e-4)	1.5201e-2(1.3426e-3)	6.9286e-3(4.4440e-4)
	(20,10)	2.5319e-2(1.6577e-3)	1.0156e-2(4.4830e-4)	9.8779e-3(3.1351e-4)	9.4965e-3(6.1726e-4)	6.1997e-3(2.2351e-4)
DF9	(5,10)	1.9072e-1(4.2276e-2)	6.1517e-1(6.7178e-2)	1.0992e+0(1.0106e-1)	1.1433e+0(1.7869e-1)	9.1211e-2(8.0318e-3)
	(10,10)	8.9055e-2(1.3005e-2)	3.4057e-1(3.8154e-2)	7.1330e-1(6.4474e-2)	5.3145e-1(1.1461e-1)	5.7369e-2(5.3842e-3)
	(20,10)	4.2832e-2(8.1064e-3)	1.9834e-1(3.4209e-2)	3.7067e-1(3.3289e-2)	1.8059e-1(6.8860e-2)	3.5463e-2(2.6413e-3)
DF10	(5,10)	1.8897e-1(3.8967e-2)	1.2030e-1(9.8663e-3)	1.2124e-1(1.1081e-2)	2.4241e-1(1.9645e-2)	9.4127e-2(6.5802e-3)
	(10,10)	1.8135e-1(4.3232e-2)	1.1333e-1(7.8492e-3)	1.1381e-1(7.0951e-3)	1.8068e-1(8.8626e-3)	9.4208e-2(3.5294e-3)
	(20,10)	1.5021e-1(3.0491e-2)	1.0559e-1(6.8413e-3)	1.0267e-1(4.0909e-3)	1.4429e-1(6.5544e-3)	9.6191e-2(3.8677e-3)
DF11	(5,10)	6.3920e-1(5.3081e-4)	6.6776e-1(1.8531e-3)	6.6716e-1(2.1127e-3)	6.6790e-1(3.3456e-3)	6.4447e-1(1.8045e-3)
	(10,10)	6.3739e-1(3.0483e-4)	6.5708e-1(1.3716e-3)	6.5622e-1(1.2129e-3)	6.5537e-1(2.4397e-3)	6.4437e-1(1.2550e-3)
	(20,10)	6.3606e-1(1.8611e-4)	6.4820e-1(1.0055e-3)	6.4791e-1(1.1038e-3)	6.4848e-1(1.3920e-3)	6.4218e-1(1.1303e-3)
DF12	(5,10)	9.2972e-1(2.4916e-2)	5.5678e-1(3.0835e-2)	5.3806e-1(3.0700e-2)	3.2578e-1(2.3880e-2)	2.6127e-1(1.1686e-2)
	(10,10)	9.4800e-1(2.3178e-2)	4.9613e-1(3.5419e-2)	4.7212e-1(2.9421e-2)	3.0639e-1(8.3670e-3)	3.1241e-1(8.9745e-3)
	(20,10)	9.4829e-1(1.6749e-2)	4.2393e-1(1.4396e-2)	4.2978e-1(2.5053e-2)	3.2805e-1(7.4974e-3)	3.4361e-1(9.0832e-3)
DF13	(5,10)	2.4607e-1(4.8671e-2)	6.1352e-1(6.7508e-2)	9.4700e-1(8.5191e-2)	8.1252e-1(9.3628e-2)	1.3442e-1(5.0895e-3)
	(10,10)	2.2604e-1(6.4744e-3)	3.1991e-1(2.1590e-2)	3.3640e-1(2.2135e-2)	4.1562e-1(3.9580e-2)	1.3735e-1(5.1994e-3)
	(20,10)	2.3506e-1(5.7118e-2)	1.4396e-1(4.7138e-3)	1.4076e-1(5.5952e-3)	2.4416e-1(9.3604e-3)	1.4238e-1(4.3516e-3)
DF14	(5,10)	1.2430e-1(5.6420e-3)	2.9820e-1(5.3468e-2)	4.1963e-1(1.9996e-1)	4.0924e-1(8.3124e-2)	4.8251e-2(1.3401e-3)
	(10,10)	1.2360e-1(3.8031e-3)	1.8185e-1(6.1739e-2)	1.7098e-1(1.2090e-1)	1.5354e-1(2.6930e-2)	4.8188e-2(1.3258e-3)
	(20,10)	1.2265e-1(3.6703e-3)	8.6828e-2(2.8908e-2)	9.0886e-2(2.5838e-2)	7.4931e-2(5.1931e-3)	4.9236e-2(1.4073e-3)

MHV values than other algorithms in most of the problems, but it seems slightly inefficient on DF2, DF10, and DF11. DNSGA-II-B has a marginal advantage over the others on DF8 and DF13 with slow changes. Nevertheless, the MHV metric further indicates the ability of PBDMO to solve dynamic problems.

As indicated in Table III, PBDMO has the best solution distribution on three bi-objective problems, i.e., DF1, DF5, and DF7, but seems to have worse MSP values than the others in a majority of the test cases, although the difference is not significant. MOEA/D-FD is clearly the winner

as it obtains the best MSP values in most cases. We wonder why MOEA/D-FD can obtain a good solution distribution where the others can not, and we postulate that this is mainly attributed to different distribution mechanisms used in these algorithms. MOEA/D-FD is a decomposition-based algorithm that converts a multiobjective problem into a number of single-objective subproblems. The weights used in aggregation functions are generated evenly, and it has been widely reported that the Tchebycheff approach is able to provide a good distribution of solutions. In contrast, the other MOEAs do not have this mechanism, and their environmental selection for solution

TABLE II
MEAN AND STANDARD DEVIATION VALUES OF MHV OBTAINED BY FIVE ALGORITHMS

Fun.	(τ_t, n_t)	MOEA/D-FD	DNSGA-II-A	DNSGA-II-B	PPS	PBDMO
DF1	(5,10)	1.6516e+0(6.5843e-4)	1.3696e+0(2.4852e-2)	9.1882e-1(6.4536e-2)	6.8631e-1(7.5822e-2)	1.6658e+0(1.7047e-3)
	(10,10)	1.6611e+0(5.2987e-4)	1.5346e+0(1.3090e-2)	1.4842e+0(1.8139e-2)	9.8814e-1(6.3972e-2)	1.6697e+0(9.0635e-4)
	(20,10)	1.6661e+0(2.4812e-4)	1.6452e+0(1.5474e-3)	1.6477e+0(1.5756e-3)	1.3672e+0(4.3413e-2)	1.6722e+0(5.5826e-4)
DF2	(5,10)	1.8708e+0(1.9581e-3)	1.7222e+0(1.2664e-2)	1.2547e+0(5.7322e-2)	1.0348e+0(1.2583e-1)	1.6923e+0(1.3314e-2)
	(10,10)	1.8921e+0(6.0890e-4)	1.8225e+0(1.2896e-2)	1.6348e+0(1.7309e-2)	1.4193e+0(6.3314e-2)	1.8115e+0(9.9833e-3)
	(20,10)	1.8995e+0(1.6125e-4)	1.8852e+0(4.6767e-3)	1.8196e+0(1.1975e-2)	1.7249e+0(2.8389e-2)	1.8867e+0(3.1529e-3)
DF3	(5,10)	1.5275e+0(4.6860e-2)	8.9547e-1(5.9539e-2)	9.5311e-1(5.2584e-2)	7.7731e-1(2.4483e-1)	1.5959e+0(1.5695e-2)
	(10,10)	1.5607e+0(4.5365e-3)	1.0740e+0(3.0154e-2)	1.0751e+0(3.0565e-2)	1.2360e+0(2.9366e-1)	1.6119e+0(1.4275e-2)
	(20,10)	1.5788e+0(9.6241e-3)	1.1298e+0(2.2927e-2)	1.1362e+0(2.7652e-2)	1.4436e+0(2.2044e-1)	1.6217e+0(1.2137e-2)
DF4	(5,10)	7.3076e+0(9.6274e-3)	7.1814e+0(3.5696e-2)	7.2088e+0(3.6012e-2)	6.1180e+0(2.6430e-2)	7.5365e+0(2.5637e-2)
	(10,10)	7.3990e+0(7.2713e-3)	7.4117e+0(1.7434e-2)	7.4162e+0(1.8028e-2)	7.2172e+0(7.9157e-2)	7.5843e+0(1.7300e-2)
	(20,10)	7.4611e+0(2.8467e-3)	7.5463e+0(6.0602e-3)	7.5452e+0(4.5830e-3)	7.5133e+0(3.7989e-2)	7.5991e+0(1.4942e-2)
DF5	(5,10)	1.7112e+0(9.7075e-4)	1.1864e+0(6.7654e-2)	1.0833e+0(6.2101e-2)	5.4753e-1(7.8645e-2)	1.7307e+0(2.2870e-3)
	(10,10)	1.7189e+0(5.0967e-4)	1.6400e+0(1.6283e-2)	1.6524e+0(7.5657e-3)	1.1029e+0(1.5098e-1)	1.7351e+0(1.4837e-3)
	(20,10)	1.7258e+0(3.9364e-4)	1.7229e+0(6.9872e-4)	1.7234e+0(7.1480e-4)	1.5988e+0(4.0938e-2)	1.7373e+0(1.1746e-3)
DF6	(5,10)	1.1390e+0(1.7883e-2)	2.9137e-1(5.4332e-2)	2.7458e-1(6.2074e-2)	4.3247e-3(6.0419e-3)	1.1778e+0(1.1648e-2)
	(10,10)	1.2269e+0(1.0211e-2)	5.2056e-1(2.6817e-2)	5.3018e-1(3.9555e-2)	3.6533e-2(3.6828e-2)	1.2654e+0(1.3376e-2)
	(20,10)	1.2913e+0(2.7400e-2)	1.0085e+0(2.1741e-2)	1.0105e+0(2.0603e-2)	2.3910e-1(1.3016e-1)	1.3303e+0(1.5288e-2)
DF7	(5,10)	3.3392e+0(1.0704e-2)	2.8390e+0(9.8682e-2)	2.6654e+0(1.9815e-1)	2.9273e+0(1.8053e-1)	3.4573e+0(2.9521e-3)
	(10,10)	3.3653e+0(1.2781e-2)	2.9564e+0(9.2309e-2)	2.8426e+0(1.6107e-1)	3.2852e+0(5.1348e-2)	3.4663e+0(1.6305e-3)
	(20,10)	3.4092e+0(8.0006e-3)	3.0963e+0(6.1092e-2)	3.0983e+0(5.0842e-2)	3.4167e+0(1.8423e-2)	3.4723e+0(1.7622e-3)
DF8	(5,10)	1.7669e+0(1.1943e-3)	1.7554e+0(1.0993e-2)	1.7596e+0(1.6025e-2)	1.7061e+0(7.2741e-3)	1.7675e+0(2.4454e-3)
	(10,10)	1.7738e+0(5.4330e-4)	1.7750e+0(9.9699e-3)	1.7784e+0(5.8430e-3)	1.7529e+0(5.0560e-3)	1.7737e+0(2.8306e-3)
	(20,10)	1.7780e+0(4.1528e-4)	1.7832e+0(5.6077e-3)	1.7843e+0(4.6002e-3)	1.7730e+0(2.5959e-3)	1.7778e+0(2.6290e-3)
DF9	(5,10)	1.3914e+0(4.9198e-2)	6.9188e-1(5.3580e-2)	5.3459e-1(6.5295e-2)	4.1101e-1(8.2987e-2)	1.5126e+0(1.9448e-2)
	(10,10)	1.5458e+0(2.3001e-2)	1.1163e+0(3.7140e-2)	1.0127e+0(2.9063e-2)	8.4195e-1(1.2188e-1)	1.5906e+0(1.3155e-2)
	(20,10)	1.6225e+0(1.7026e-2)	1.3676e+0(4.3861e-2)	1.2423e+0(1.7356e-2)	1.3623e+0(8.9762e-2)	1.6412e+0(7.3492e-3)
DF10	(5,10)	1.3154e+0(4.2911e-2)	1.5009e+0(8.11129e-3)	1.4997e+0(7.9058e-3)	9.1583e-1(4.5043e-2)	1.2803e+0(2.1705e-2)
	(10,10)	1.3573e+0(1.2131e-2)	1.5146e+0(5.3260e-3)	1.5135e+0(7.5550e-3)	1.0907e+0(2.3924e-2)	1.3261e+0(1.7679e-2)
	(20,10)	1.3720e+0(1.3164e-2)	1.5244e+0(4.3825e-3)	1.5240e+0(6.2009e-3)	1.2220e+0(2.2523e-2)	1.3495e+0(1.0716e-2)
DF11	(5,10)	3.0723e-1(1.4172e-3)	3.6714e-1(1.5235e-3)	3.6656e-1(1.9629e-3)	3.2919e-1(9.5921e-3)	3.6852e-1(1.9356e-3)
	(10,10)	3.0959e-1(1.4227e-3)	3.7202e-1(1.3716e-3)	3.7310e-1(1.4625e-3)	3.4979e-1(5.1911e-3)	3.7023e-1(3.2374e-3)
	(20,10)	3.1158e-1(1.1737e-3)	3.7812e-1(1.2551e-3)	3.7810e-1(1.2905e-3)	3.6147e-1(2.8344e-3)	3.7400e-1(1.8596e-3)
DF12	(5,10)	3.3943e+0(1.3892e-2)	3.3787e+0(3.6948e-2)	3.4122e+0(2.1612e-2)	2.8107e+0(1.2448e-1)	3.4573e+0(2.7184e-2)
	(10,10)	3.4085e+0(3.7436e-2)	3.4463e+0(2.1655e-2)	3.4559e+0(1.7019e-2)	3.1837e+0(4.5101e-2)	3.4687e+0(2.4142e-2)
	(20,10)	3.4354e+0(9.2232e-3)	3.4883e+0(8.2943e-3)	3.4879e+0(1.1489e-2)	3.3608e+0(1.8850e-2)	3.4920e+0(2.1832e-2)
DF13	(5,10)	6.5504e+0(3.6332e-2)	4.5769e+0(2.8836e-1)	3.5882e+0(1.4940e-1)	3.2579e+0(3.3401e-1)	7.1577e+0(2.9770e-2)
	(10,10)	6.6744e+0(1.0012e-2)	6.1246e+0(1.1789e-1)	6.0397e+0(1.2553e-1)	5.5350e+0(2.6095e-1)	7.1296e+0(2.5025e-2)
	(20,10)	6.7295e+0(4.7829e-2)	7.1971e+0(2.7990e-2)	7.2175e+0(2.8750e-2)	6.5402e+0(6.1670e-2)	7.0894e+0(3.4532e-2)
DF14	(5,10)	9.1267e-1(6.8322e-3)	6.7574e-1(7.3178e-2)	6.1144e-1(7.0561e-2)	4.7579e-1(6.6427e-2)	1.0794e+0(2.0743e-3)
	(10,10)	9.1551e-1(5.5030e-3)	8.7668e-1(9.3532e-2)	9.2252e-1(1.0947e-1)	8.7833e-1(5.1356e-2)	1.0807e+0(1.6116e-3)
	(20,10)	9.1749e-1(4.9592e-3)	1.0262e+0(4.5703e-2)	1.0198e+0(4.2483e-2)	1.0320e+0(9.6874e-3)	1.0798e+0(1.7808e-3)

distribution maintenance may not be able to generate solutions as well-organized as in MOEA/D. Keeping a well-organized set of solutions is particularly more difficult for these nondecomposition MOEAs in three-objective problems. Moreover, a good solution distribution does not necessarily guarantee a good approximation to the POF. While MOEA/D-FD wins on solution distribution, it loses to PBDMO on the other two measures, i.e., MIGD and MHV, which are more reliable to distinguish between algorithms in terms of the overall performance.

It can also be observed from the three measures that the frequency of change has a significant effect on algorithms' performance, and the effect decreases when environmental changes become slow. For three-objective problems, frequent changes cause these algorithms poor POF approximations, as indicated by their large MIGD and MHV values in Tables I and II, respectively. Overall, PBDMO seems less sensitive to the frequency of change, as can be seen from its gradual improvement on the three measures when τ_t increases from 5 to 20. On the other hand, drastic changes with respect

TABLE III
MEAN AND STANDARD DEVIATION VALUES OF MSP OBTAINED BY FIVE ALGORITHMS

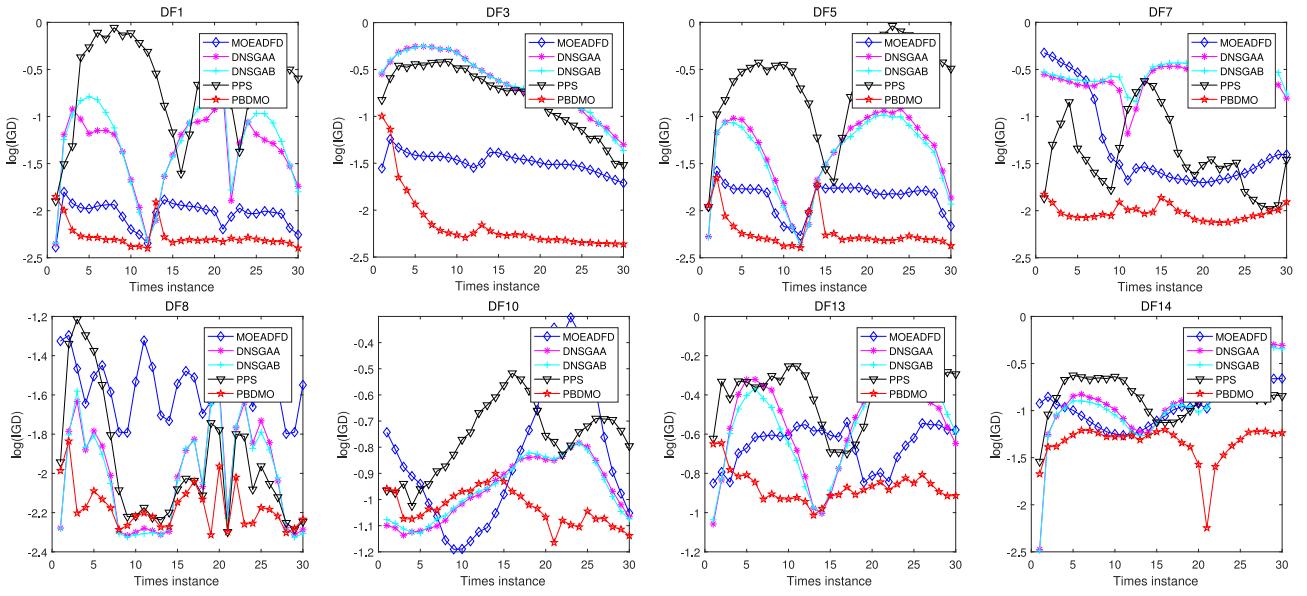
Fun.	(τ_t, n_t)	MOEA/D-FD	DNSGA-II-A	DNSGA-II-B	PPS	PBDMO
DF1	(5,10)	9.4029e-3(1.2133e-4)	1.7703e-2(1.7181e-3)	2.0796e-2(1.5152e-3)	1.8140e-1(6.3630e-2)	5.6170e-3(1.7735e-3)
	(10,10)	8.2100e-3(1.0882e-4)	1.1334e-2(5.3655e-4)	1.2189e-2(6.2419e-4)	1.0821e-1(4.0601e-2)	6.3333e-3(5.1830e-3)
	(20,10)	6.7905e-3(7.4139e-5)	6.9871e-3(2.6810e-4)	6.6445e-3(1.8039e-4)	4.1261e-2(1.2272e-2)	5.4449e-3(2.5565e-3)
DF2	(5,10)	1.2244e-2(3.3624e-4)	2.1784e-2(2.3646e-3)	1.8267e-2(1.1643e-3)	2.0255e-1(1.2132e-1)	1.0268e-1(1.6850e-2)
	(10,10)	1.1602e-2(1.7518e-4)	1.0616e-2(1.3175e-3)	1.0573e-2(8.3015e-4)	6.9423e-2(2.3812e-2)	1.4402e-1(2.1345e-2)
	(20,10)	1.0556e-2(1.1006e-4)	5.7611e-3(4.6626e-4)	7.0121e-3(7.9818e-4)	4.1121e-2(1.1214e-2)	1.6359e-1(2.4889e-2)
DF3	(5,10)	1.4426e-2(1.1430e-3)	2.4481e-2(2.7895e-3)	2.4954e-2(3.5040e-3)	1.8456e+0(1.2621e+0)	2.8983e-2(1.3026e-2)
	(10,10)	1.1976e-2(4.4941e-4)	1.3914e-2(2.2553e-3)	1.2964e-2(3.1566e-3)	4.1912e-1(2.1708e-1)	1.8212e-2(1.1031e-2)
	(20,10)	1.1856e-2(3.7984e-4)	1.0616e-2(2.2160e-3)	1.1350e-2(3.7708e-3)	2.4571e-1(2.5992e-1)	1.5371e-2(1.1000e-2)
DF4	(5,10)	1.0157e-1(1.4191e-2)	7.4020e-2(2.0187e-2)	5.7903e-2(1.1662e-2)	2.6126e+0(2.1431e+0)	9.0780e-2(6.7615e-2)
	(10,10)	9.3139e-2(1.1817e-2)	6.2118e-2(2.6318e-2)	5.0066e-2(1.0350e-2)	3.1168e-1(2.4894e-1)	5.6830e-2(4.7220e-2)
	(20,10)	8.2449e-2(4.5007e-3)	3.5093e-2(1.0986e-2)	3.8985e-2(1.3480e-2)	8.8500e-2(4.9333e-2)	3.5199e-2(2.3189e-2)
DF5	(5,10)	1.1198e-2(1.8037e-4)	3.4964e-2(4.2185e-3)	3.6819e-2(3.9059e-3)	1.1687e+0(5.6705e-1)	1.0467e-2(4.3650e-3)
	(10,10)	1.0239e-2(1.2387e-4)	1.4127e-2(1.2385e-3)	1.3429e-2(1.5939e-3)	3.2706e-1(1.9465e-1)	9.2734e-3(5.2298e-3)
	(20,10)	9.2490e-3(8.5697e-5)	7.6441e-3(5.9021e-4)	7.4691e-3(3.4571e-4)	7.7818e-2(2.5240e-2)	6.5864e-3(3.5960e-3)
DF6	(5,10)	2.6212e-1(5.3312e-2)	1.1787e+0(1.0268e-1)	1.3071e+0(7.5035e-2)	1.3386e+2(4.9621e+1)	2.2673e+0(4.3123e-1)
	(10,10)	1.6089e-1(2.4948e-2)	5.5187e-1(5.1233e-2)	7.2333e-1(4.4059e-2)	6.2461e+1(3.1967e+1)	1.4760e+0(3.5868e-1)
	(20,10)	1.4070e-1(1.5598e-2)	1.5202e-1(2.1032e-2)	2.5682e-1(2.6554e-2)	3.6419e+1(4.9424e+1)	5.7453e-1(1.7213e-1)
DF7	(5,10)	2.7018e-2(1.0289e-3)	2.9406e-2(7.0437e-3)	2.0716e-2(1.3698e-2)	2.7441e-2(4.1252e-3)	9.3158e-3(4.8611e-4)
	(10,10)	2.7250e-2(1.0137e-3)	2.2504e-2(6.8737e-3)	1.7675e-2(5.7338e-3)	1.7617e-2(2.5212e-3)	7.5165e-3(2.1666e-4)
	(20,10)	2.9657e-2(8.1921e-4)	1.9081e-2(4.2596e-3)	1.8154e-2(4.4887e-3)	1.0236e-2(9.0154e-4)	6.9462e-3(1.1344e-4)
DF8	(5,10)	1.9365e-2(1.6946e-3)	3.2735e-2(9.4409e-3)	2.4851e-2(7.4129e-3)	1.2316e-1(3.4514e-2)	4.1530e-2(7.9194e-3)
	(10,10)	1.9045e-2(8.0073e-4)	1.8928e-2(4.8021e-3)	1.5466e-2(4.5403e-3)	8.6782e-2(3.8698e-2)	3.5292e-2(7.7929e-3)
	(20,10)	1.8939e-2(7.6444e-4)	1.0759e-2(2.7644e-3)	1.0098e-2(2.6119e-3)	5.5292e-2(3.1136e-2)	2.3675e-2(7.5871e-3)
DF9	(5,10)	1.8427e-2(3.8016e-3)	8.9926e-2(1.5814e-2)	8.4824e-2(7.9560e-3)	1.3426e+0(4.3716e-1)	1.2553e-1(2.9933e-2)
	(10,10)	1.2072e-2(1.3294e-3)	4.5782e-2(8.2607e-3)	4.7653e-2(5.1128e-3)	5.3145e-1(1.1461e-1)	7.6180e-2(2.2360e-2)
	(20,10)	1.0017e-2(5.4045e-4)	2.6050e-2(5.4019e-3)	2.9432e-2(4.8425e-3)	2.5234e-1(1.0440e-1)	4.5956e-2(1.3125e-2)
DF10	(5,10)	3.5061e-2(1.5860e-2)	8.8583e-2(1.8228e-2)	9.3216e-2(1.5020e-2)	1.2555e+0(3.4768e-1)	3.1673e-1(4.4141e-2)
	(10,10)	3.8637e-2(1.7612e-2)	8.2117e-2(9.3678e-3)	8.4987e-2(1.5811e-2)	8.7872e-1(2.1905e-1)	3.1638e-1(4.2849e-2)
	(20,10)	3.2465e-2(1.1605e-2)	7.8996e-2(1.1703e-2)	7.9783e-2(1.3565e-2)	8.1298e-1(3.8463e-1)	3.2513e-1(3.6829e-2)
DF11	(5,10)	2.5439e-2(5.4454e-4)	5.6303e-2(9.8620e-4)	5.6371e-2(1.1788e-3)	6.1775e-2(5.4680e-3)	5.2637e-2(2.5637e-3)
	(10,10)	2.4045e-2(4.0488e-4)	5.3544e-2(1.0143e-3)	5.3701e-2(9.8587e-4)	5.4945e-2(3.4220e-3)	5.1533e-2(2.6103e-3)
	(20,10)	2.2054e-2(3.0127e-4)	5.2136e-2(8.3359e-4)	5.2020e-2(1.0533e-3)	5.2460e-2(2.4097e-3)	5.1727e-2(3.4494e-3)
DF12	(5,10)	1.4776e-2(7.6499e-3)	8.6057e-2(7.8278e-3)	9.3901e-2(1.0390e-2)	1.1784e+0(2.3349e-1)	4.3098e-1(4.4652e-2)
	(10,10)	1.3523e-2(1.0268e-2)	9.5582e-2(9.2223e-3)	9.3844e-2(1.0187e-2)	8.0796e-1(1.6230e-1)	3.9178e-1(3.2336e-2)
	(20,10)	1.3224e-2(5.8471e-3)	9.7327e-2(1.3674e-2)	9.1248e-2(1.4752e-2)	5.8093e-1(9.6952e-2)	3.7705e-1(3.7903e-3)
DF13	(5,10)	1.3927e-1(5.5487e-3)	2.4586e-1(2.0846e-2)	2.8807e-1(1.8919e-2)	4.5108e+0(1.0008e+0)	1.9493e-1(6.0771e-2)
	(10,10)	1.5208e-1(3.2609e-3)	1.6352e-1(7.8961e-3)	1.6093e-1(6.8360e-3)	2.5276e+0(9.1521e-1)	1.6532e-1(1.8369e-2)
	(20,10)	1.5664e-1(2.1860e-3)	1.1088e-1(3.7440e-3)	1.1143e-1(2.6600e-3)	8.6459e-1(3.4169e-1)	1.7024e-1(1.8961e-2)
DF14	(5,10)	1.6250e-2(7.4238e-4)	9.6528e-2(2.3475e-2)	1.1271e-1(4.5214e-2)	8.7825e-1(2.7034e-1)	4.5133e-2(5.4089e-3)
	(10,10)	1.6416e-2(4.3668e-4)	1.4251e-1(7.0052e-2)	1.0148e-1(5.4245e-2)	3.1317e-1(1.2279e-1)	4.3046e-2(4.1334e-3)
	(20,10)	1.6048e-2(4.4702e-4)	1.2831e-1(8.1860e-2)	1.5916e-1(9.1897e-2)	1.1005e-1(1.7845e-2)	4.0813e-2(3.1191e-3)

to the measures for DNSGA-II-A, DNSGA-II-B, PPS, and MOEA/D-FD are observed in most of the test instances with the variation of frequencies.

Apart from tabular presentation, we provide the evolution curves of the average IGD values for most of the test instances in Fig. 4. It can be clearly seen that, compared with other algorithms, PBDMO responds to changes more stably and recovers faster for most of the test problems, thereby obtaining higher convergence performance. The only exception is DF10, where PBDMO performs slightly worse for the first half of time points, but it is the best for the second half. On

DF8, the overall performance of PBDMO is better than the others, although no noticeable difference can be seen for some changes.

Another observation is that the IGD values for the algorithms fluctuate widely on most problems, such as DF1, DF7, DF13, and DF14. Despite that, PBDMO performs steadier compared with other algorithms. For a graphical view of these algorithms' tracking ability, we also plot their POF approximations for DF3, DF5, DF7, and DF8 in Fig. 5. It evidently shows that PBDMO is very capable of tackling environmental changes, but may have convergence issues for boundary

Fig. 4. Mean IGD curves for different problems with $n_t=10$ and $\tau_t=10$.TABLE IV
PERFORMANCE COMPARISON OF DIFFERENT PBDMO VARIANTS ON HV

Fun.	(n_t, τ_t)	PBDMO-V1	PBDMO-V2	PBDMO-V3	PBDMO
DF1	(10,10)	1.6682e+0(1.2132e-3)	1.6160e+0(2.8525e-2)	1.6695e+0(1.0817e-3)	1.6697e+0(9.0635e-4)
DF2	(10,10)	1.8016e+0(1.0739e-2)	1.7817e+0(9.9569e-3)	1.8568e+0(6.8272e-3)	1.8115e+0(9.9833e-3)
DF3	(10,10)	1.6044e+0(1.9625e-2)	1.5918e+0(4.8685e-2)	1.5987e+0(2.4591e-2)	1.1915e-2(5.9480e-3)
DF4	(10,10)	7.5725e+0(2.1481e-2)	7.5833e+0(1.1889e-2)	7.5733e+0(1.4687e-2)	7.5843e+0(1.7300e-2)
DF5	(10,10)	1.7344e+0(1.1335e-3)	1.6980e+0(2.0602e-2)	1.7347e+0(1.6068e-3)	1.7351e+0(1.4837e-3)
DF6	(10,10)	1.2028e+0(2.6215e-2)	1.0249e+0(7.2468e-2)	1.1642e+0(4.6191e-2)	1.4760e+0(3.5868e-1)
DF7	(10,10)	3.4648e+0(2.6535e-3)	3.3476e+0(8.2642e-3)	3.4650e+0(2.4352e-3)	3.4663e+0(1.6305e-3)
DF8	(10,10)	1.7726e+0(3.3650e-3)	1.7721e+0(2.8138e-3)	1.7724e+0(2.8019e-3)	1.7737e+0(2.8306e-3)
DF9	(10,10)	1.5739e+0(1.1293e-2)	1.5111e+0(3.6460e-2)	1.5778e+0(1.3413e-2)	1.5906e+0(1.3155e-2)
DF10	(10,10)	1.3018e+0(1.6169e-2)	1.3178e+0(1.6352e-2)	1.3076e+0(1.2369e-2)	1.3261e+0(1.7679e-2)
DF11	(10,10)	3.7281e-1(2.1821e-3)	3.6782e-1(3.0625e-3)	3.6999e-1(2.8957e-3)	3.7023e-1(3.2374e-3)
DF12	(10,10)	3.4503e+0(2.3129e-2)	3.4322e+0(2.3948e-2)	3.3718e+0(2.2646e-2)	3.4687e+0(2.4142e-2)
DF13	(10,10)	7.1564e+0(2.8673e-2)	7.0159e+0(4.3686e-2)	7.0947e+0(3.6120e-2)	7.1296e+0(2.5025e-2)
DF14	(10,10)	1.0812e+0(2.0968e-3)	1.0769e+0(1.7892e-3)	1.0801e+0(2.0273e-3)	1.0807e+0(1.6116e-3)

individuals. The less converged boundary solutions in PBDMO may affect the performance indicators.

Computational time for all the algorithms is reported in the supplementary material due to page limit. It is found that PBDMO requires longer computational time than all the other algorithms except PPS. This is acceptable given the fact PBDMO generally obtains better results as mentioned above.

V. MORE EXPERIMENTAL ANALYSIS

A. Component Analysis

The proposed method has three key components. However, the role that each component plays in handling dynamics remains unclear. Here, we modify PBDMO to study each component separately. Specifically, a one-step-size prediction strategy replaces the three-step-size strategy in PBDMO for predicting the nondominated set. That is, a single step-size value ($rs = 1$) is used, and this variant is called PBDMOV1.

In other words, PBDMOV1 uses a very common prediction strategy similar to most existing prediction-based algorithms. A comparison between PBDMO and PBDMOV1 should illustrate the unique advantage of the three step-size strategy. The sampling strategy is switched off to study its importance to PBDMO, and this variant is named PBDMOV2. Similarly, we also deactivate the shrinking strategy to study the role it plays in PBDMO, resulting in another variant called PBDMOV3.

These three variants are all compared with the original PBDMO, and their MHV values are presented in Table IV. The corresponding MIGD and MSP values can be found in the supplementary material. The following details the influence of each component.

1) *Predicting Nondominated Set*: It is not difficult to see that PBDMO performs better than PBDMOV1 on bi-objective cases, but it is outperformed by PBDMOV1 on three-objective problems according the MIGD, MHV, and MSP values. The reason may be due to the fact that the three-step-size strategy

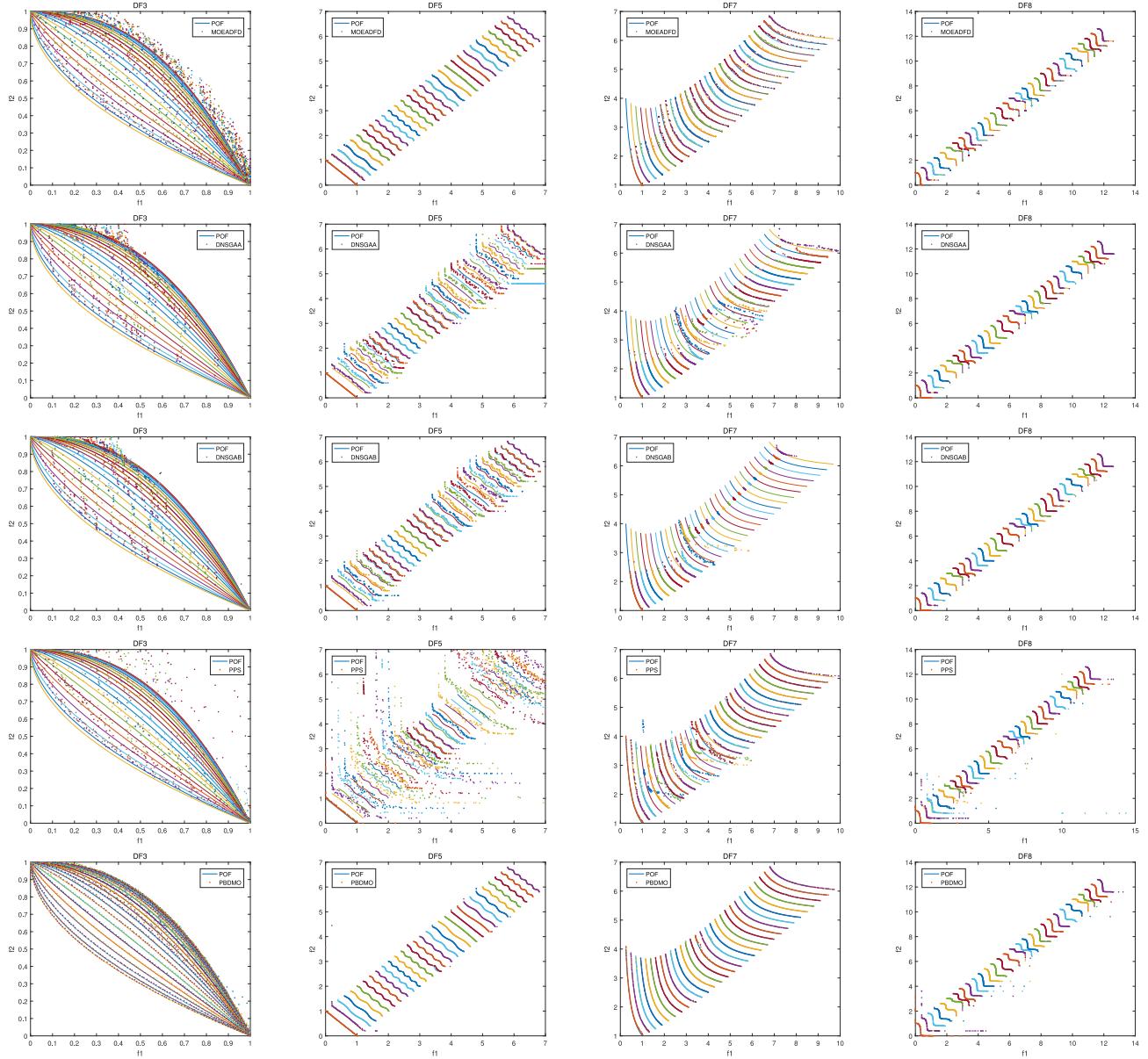


Fig. 5. POF approximations of five algorithms for different problems with $n_t = 10$ and $\tau_t = 10$.

in PBDMO would generate more boundary individuals than the one-step-size strategy in PBDMOV1. Boundary individuals are more likely to be nondominated solutions than intermediate individuals. Too much nondominated boundary solutions is not good for population diversity. Despite that, the three-step-size strategy brings more benefits than the one-step-size strategy for the majority of the problems.

2) *Sampling Strategy*: All three measures show that PBDMO performs significantly better than PBDMOV2 on almost all test cases, although PBDMOV2 has slightly better solution distribution on DF2, DF8, DF10, and DF12 than PBDMO as suggested by the MSP measure. Thus, the proposed sampling strategy clearly enhances the initial population for each new environment, and can efficiently improve the performance of the algorithm (see Figs. 2 and 3 in the supplementary material).

3) *Shrinking Strategy*: It can be observed from the results of the three indicators that PBDMO is better than PBDMOV3 in almost all the problems. This demonstrates the use of space shrinkage indeed helps improve the prediction of population for changing environments. This is not surprising because shrinking the decision space based on the probability distribution of variables improves the chance of creating promising solutions, which in return improves the quality of the initial population for each new environment.

The comparison between the three variants and the original PBDMO shows that each component is indispensable for the high performance of PBDMO. 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 PBDMO.

TABLE V
PBDMO WITH DIFFERENT STEP-SIZE VALUES

S	L
PBDMO-S1(0.1,1,1.5)	PBDMO-L1(0.5,1,1.1)
PBDMO-S2(0.3,1,1.5)	PBDMO-L2(0.5,1,1.3)
PBDMO (0.5,1,1.5)	PBDMO (0.5,1,1.5)
PBDMO-S3(0.7,1,1.5)	PBDMO-L3 (0.5,1,1.7)

B. Influence of rs Values

The three-step-size strategy for predicting nondominated set uses three different step-sizes, and they are set to (0.5, 1, and 1.5) in PBDMO. Here, we wonder whether other rs values will enable PBDMO significantly better performance than (0.5, 1, and 1.5), particularly when changing the value for the small or large step-size. Thus, we carried out experiments on two groups of rs settings. The first group (S) uses different values for the small step-size whereas the second group (L) has several values for the large step-size. The influence of the medium step-size is not investigated as we assume it is for moderate cases where two consecutive environmental changes are similar. A step-size of 1 is a very common setting for moderate cases in many prediction-based algorithms. Table V details these two groups of settings.

The experimental results are shown in Table VI for groups S and L. It is seen from the S group comparison that the optimal value for the small step-size is problem dependent: small values favor some problems while slightly large values favor others. A comparison between PBDMO and each S column of Table VI demonstrates that the setting (0.5, 1, and 1.5) helps PBDMO to win slightly more competitions. A similar observation can be made for the L group from Table VI. Thus, this experiment supports the decision made on choosing a step-size vector of (0.5, 1, and 1.5).

C. Influence of $Subpop_3$ Size

The size of $Subpop_3$ is another important parameter that can affect the performance of PBDMO. So, it is varied with an increment of 0.1 from 0.1 to 0.5 times of the population size, to study its potential influences. Other parameters are kept the same as in the previous section. The results are given in Table VII. It is clear that no single optimal $Subpop_3$ size exists for all the problems but instead the optimal $Subpop_3$ sizes are quite divergent. While some problems (e.g., DF1) are not very sensitive to the size of $Subpop_3$, other problems (e.g., DF2 and DF3) show widely fluctuated performance over the tested sizes and obtained the best performance when the size of $Subpop_3$ is around $0.3N$. It is also clear that although PBDMO with $0.3N$ is not the best on some cases, it is very close to the best ones. Therefore, $0.3N$ is the most preferred subpopulation size for $Subpop_3$ in PBDMO.

D. Influence of the Severity of Change n_t

To study whether different change severity levels bring significant influence to the performance of the algorithm, experiments were carried out with τ_t fixed to 10, and n_t set to 5, 10, and 20. The simulation results can be found in the

supplementary material. We observe that all algorithms are sensitive to n_t . Although various n_t bring different difficulties to the problem, the PBDMO can still obtain promising results with respective other compared algorithms. In total, PBDMO has more robust performance in solving problems with different changing levels.

E. Comparison With State-of-the-Arts

A transfer learning-based DMOEA (TrDMOEA) has been recently developed and shown promising performance on DMOPs [49]. Here, we compare PBDMO with this state-of-the-art to demonstrate the effectiveness of the proposed strategies. Table VIII shows the MIGD values of the two algorithms for some selected problems, and more results can be found in the supplementary material. It is clear that PBDMO is more effective than TrDMOEA for DF problems, showing complex learning models do not necessarily guarantee high performance. Note that, PBDMO is also computationally cheaper than TrDMOEA.

VI. DISCUSSION

We would like to have more discussion here about the advantages and limitations of the three components of the proposed PBDMO method. First, the three-step-size strategy for predicting nondominated solutions increases the population diversity, which allows the population to explore widely for the new environment and helps to improve the search performance of dynamic multiobjective algorithms. However, the improvement comes at the expense of complexity increase because it generates more solutions than an one-step-size strategy does. Besides, it is likely to generate too many near-boundary individuals, as shown in the experimental results, which may not be beneficial to global search if most of these near-boundary individuals are nondominated. Therefore, this strategy could be improved by controlling the number of near-boundary members effectively.

Second, the sampling technique generates solutions by classifying decision variables into two groups, which can be very helpful for generating well-distributed solutions in the population. The subpopulation from this strategy additionally has convergence advantages as the solutions have been filtered by nondominated sorting. This provides us a new way of efficient sampling in dynamic environments. However, the strategy heavily depends on the classification. This paper assumes there is a single most principal variable, and this assumption may not hold for generalization. The sampling strategy needs improvements on the classification and robustness when the principal variable is misidentified or there are many principal variables.

Third, the shrinking strategy aids efficient sampling, thereby improving the quality of the initial population for each new environment. However, it sometimes fails to work effectively, particularly when facing cases where two consecutive POSs has low similarities. This results in poor estimate of the range of nonprincipal variables. The failure has been observed for DF11 and DF12, for which the POS is very unpredictable. Therefore, further improvement should be made on this strategy to shrink the decision space more effectively.

TABLE VI
PERFORMANCE COMPARISON OF PBDMO VARIANTS ON MIGD FOR $(n_t, \tau_t) = (10, 10)$

Fun.	PBDMO-S1	PBDMO-S2	PBDMO-S3	PBDMO-L1	PBDMO-L2	PBDMO-L3	PBDMO
DF1	5.6079e-3(2.4725e-4)	5.5702e-3(2.0919e-4)	5.6000e-3(2.7930e-4)	5.5808e-3(2.5028e-4)	5.5468e-3(2.5822e-4)	5.6625e-3(2.6767e-4)	5.5687e-3(2.1958e-4)
DF2	3.4915e-2(2.9066e-3)	3.4334e-2(2.6436e-3)	3.5540e-2(3.7890e-3)	3.6489e-2(2.6309e-3)	3.5532e-2(3.3191e-3)	3.5437e-2(3.2281e-3)	3.4739e-2(3.3543e-3)
DF3	1.2290e-2(5.8988e-3)	1.5808e-2(8.7778e-3)	1.0472e-2(4.2792e-3)	1.1792e-2(5.9125e-3)	1.3886e-2(5.9125e-3)	1.4803e-2(7.7206e-3)	1.1915e-2(5.9480e-3)
DF4	7.1063e-2(2.9154e-3)	7.1892e-2(3.0600e-3)	7.2797e-2(2.0516e-3)	7.1668e-2(2.8895e-3)	7.2264e-2(2.7956e-3)	7.2878e-2(3.8344e-3)	7.1674e-2(3.0848e-3)
DF5	6.9555e-3(6.1395e-4)	6.6668e-3(7.5173e-4)	6.0804e-3(4.5956e-4)	6.5099e-3(7.2399e-4)	6.6233e-3(6.2820e-4)	6.8986e-3(8.2222e-4)	6.5586e-3(7.7474e-4)
DF6	5.0504e-1(3.2438e-2)	4.8624e-1(3.0402e-2)	4.8618e-1(3.1674e-2)	4.7871e-1(2.5561e-2)	4.8658e-1(2.5979e-2)	4.9288e-1(2.6383e-2)	4.8638e-1(1.8615e-2)
DF7	9.5643e-3(3.5670e-4)	9.4518e-3(3.8272e-4)	1.0010e-2(4.5394e-4)	9.8773e-3(3.9498e-4)	9.8305e-3(3.4775e-4)	9.5416e-3(4.3798e-4)	9.6751e-3(3.2085e-4)
DF8	6.9131e-3(3.7459e-4)	6.9973e-3(4.2898e-4)	6.9241e-3(3.3819e-4)	6.9457e-3(3.5700e-4)	6.8571e-3(4.0086e-4)	6.9004e-3(4.1787e-4)	6.9286e-3(4.4440e-4)
DF9	6.3067e-2(5.1197e-3)	5.8210e-2(3.9409e-3)	5.7400e-2(4.8239e-3)	5.8596e-2(4.5538e-3)	5.9777e-2(2.4007e-3)	5.7012e-2(5.3177e-3)	5.7369e-2(5.3842e-3)
‡/†/≈	5/4/0	6/3/0	5/4/0	5/4/0	7/2/0	5/4/0	-

‡, † and ≈ indicate that the performance of PBDMO is better than, worse than and similar to that of the corresponding algorithm, respectively.

TABLE VII
PERFORMANCE COMPARISON OF PBDMO WITH DIFFERENT SIZES OF Subpop3 ON MIGD

Fun.	(n_t, τ_t)	0.1N	0.2N	0.3N	0.4N	0.5N
DF1	(10, 10)	5.5492e-3(2.1402e-4)	5.5196e-3(1.7355e-4)	5.5687e-3(2.1958e-4)	5.4825e-3(2.4577e-4)	5.5309e-3(2.3957e-4)
DF2	(10, 10)	3.5695e-2(3.0669e-3)	3.8211e-2(4.9469e-3)	3.4739e-2(3.3543e-3)	3.4793e-2(5.0706e-3)	3.4334e-2(2.7482e-3)
DF3	(10, 10)	1.3438e-2(8.5500e-3)	1.2753e-2(8.8312e-3)	1.1915e-2(5.9480e-3)	1.4692e-2(8.1973e-3)	1.4423e-2(7.6144e-3)
DF4	(10, 10)	7.1355e-2(2.2803e-3)	7.1857e-2(3.9415e-3)	7.1674e-2(3.0848e-3)	7.4075e-2(2.9373e-3)	7.2064e-2(3.3974e-3)
DF5	(10, 10)	6.9264e-3(6.7986e-4)	6.5961e-3(7.1678e-4)	6.5586e-3(7.7474e-4)	6.7354e-3(6.8402e-4)	6.6764e-3(5.7173e-4)
DF6	(10, 10)	4.7630e-1(2.1540e-2)	4.9751e-1(2.0212e-2)	4.8638e-1(1.8615e-2)	4.8249e-1(2.4461e-2)	4.8560e-1(3.0765e-2)
DF7	(10, 10)	9.7227e-3(3.4707e-4)	9.6342e-3(3.6384e-4)	9.6751e-3(3.2085e-4)	9.7179e-3(4.1769e-4)	9.6443e-3(4.2604e-4)
DF8	(10, 10)	7.0284e-3(3.4306e-4)	6.9530e-3(3.4787e-4)	6.9286e-3(4.4440e-4)	6.9241e-3(3.9296e-4)	7.0286e-3(3.2845e-4)
DF9	(10, 10)	5.7761e-2(4.0234e-3)	5.8695e-2(5.5071e-3)	5.7369e-2(5.3842e-3)	5.7467e-2(4.2334e-3)	5.8601e-2(3.7503e-3)

TABLE VIII
PERFORMANCE COMPARISON BETWEEN PBDMO AND TRDMOEA FOR SELECTED PROBLEMS ON MIGD

Fun.	(n_t, τ_t)	TrDMOEA	PBDMO
DF1	(10, 10)	1.1933e-1(5.6046e-2)	5.5687e-3(2.1958e-4)
DF3	(10, 10)	4.9757e-2(3.5121e-2)	1.1915e-2(5.9480e-3)
DF5	(10, 10)	2.3408e-2(2.9032e-3)	6.5586e-3(7.7474e-4)
DF7	(10, 10)	4.0571e-2(3.4438e-3)	9.6751e-3(3.2085e-4)
DF9	(10, 10)	5.9994e-2(6.9909e-3)	5.7369e-2(5.3842e-3)

VII. CONCLUSION

In this paper, a new dynamism handling algorithm, called PBDMO, is proposed for solving multiobjective problems with time-varying characteristics. This approach consists of three main components, that is, a three step-size strategy for predicting the nondominated set, a sampling strategy, and a shrinking strategy. These components are important for creating a good initial population, enhancing either diversity or convergence, when a change occurs in the environment.

A recent test suite of 14 benchmark functions with different characteristics is employed to assess the performance of our algorithm. The experimental comparisons indicate that the proposed method performs much better than the other algorithms considered in this paper on most of the test problems. This demonstrates PBDMO has a good tracking ability and responds fast to environmental changes. Besides, the role that each component of the proposed algorithm plays in handling dynamics is also investigated properly, showing that the combination of these components indeed is better than using each only.

Although PBDMO has shown a good tracking ability and search performance on the tested functions, there are still several relevant issues to be addressed in the future. For example, developing prediction strategies that can better deal with near boundary solutions is an important future work. It is also important to develop more efficient strategies that can adaptively classify principle and nonprincipal decision variables according to the characteristics of the problem being handled. How to improve the shrinking strategy in PBDMO is also an interesting future work.

REFERENCES

- [1] M. B. Abello, L. T. Bui, and Z. Michalewicz, "An adaptive approach for solving dynamic scheduling with time-varying number of tasks—Part II," in *Proc. IEEE Congr. Evol. Comput.*, 2011, pp. 1711–1718.
- [2] R. Azzouz, S. Bechikh, and L. B. Said, "A dynamic multi-objective evolutionary algorithm using a change severity-based adaptive population management strategy," *Soft Comput.*, vol. 21, no. 4, pp. 885–906, 2015.
- [3] J. Branke, T. Kaussler, C. Smidt, and H. Schmeck, "A multi-population approach to dynamic optimization problems," in *Evolutionary Design and Manufacture*. London, U.K.: Springer, 2000, pp. 299–307.
- [4] L. T. Bui and Z. Michalewicz, "An evolutionary multi-objective approach for dynamic mission planning," in *Proc. IEEE Congr. Evol. Comput.*, 2010, pp. 1–8.
- [5] L. L. Cao, L. H. Xu, E. D. Goodman, and H. Li, "A first-order difference model-based evolutionary dynamic multiobjective optimization," in *Proc. Conf. Simulat. Evol. Learn.*, 2017, pp. 644–655.
- [6] C. Cruz, J. R. González, and D. A. Pelta, "Optimization in dynamic environments: A survey on problems, methods and measures," *Soft Comput.*, vol. 15, no. 7, pp. 1427–1448, 2011.
- [7] K. Deb, U. V. B. Rao, and S. Karthik, "Dynamic multi-objective optimization and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling," in *Proc. 4th Int. Conf. Evol. Multi Criterion Optim.*, vol. 4403, 2007, pp. 803–817.
- [8] M. Farina, K. Deb, and P. Amato, "Dynamic multiobjective optimization problems: Test cases, approximations, and applications," *IEEE Trans. Evol. Comput.*, vol. 8, no. 5, pp. 425–442, Oct. 2004.

- [9] S. Jiang, M. Kaiser, S. Yang, S. Kollias, and N. Krasnogor, "A scalable test suite for continuous dynamic multiobjective optimization," *IEEE Trans. Cybern.*, to be published. doi: 10.1109/TCYB.2019.2896021.
- [10] C.-K. Goh and K. C. Tan, "A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 1, pp. 103–127, Feb. 2009.
- [11] M. Greeff and A. P. Engelbrecht, "Solving dynamic multi-objective problems with vector evaluated particle swarm optimisation," in *Proc. IEEE Congr. Evol. Comput.*, 2008, pp. 2922–2929.
- [12] A. Isaacs, V. Puttige, T. Ray, W. Smith, and S. Anavatti, "Development of a memetic algorithm for dynamic multi-objective optimization and its applications for online neural network modeling of UAVs," in *Proc. Int. Joint Conf. Neural Netw.*, 2008, pp. 548–554.
- [13] S. Jiang and S. Yang, "Evolutionary dynamic multiobjective optimization: Benchmarks and algorithm comparisons," *IEEE Trans. Cybern.*, vol. 47, no. 1, pp. 198–211, Jan. 2017.
- [14] S. Jiang and S. Yang, "A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 65–82, Feb. 2017.
- [15] S. Jiang, S. Yang, X. Yao, K. Tan, M. Kaiser, and N. Krasnogor, "Benchmark functions for the CEC'2018 competition on dynamic multiobjective optimization," Newcastle Univ., Newcastle upon Tyne, U.K., Rep., Jan. 2018.
- [16] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environment—A survey," *IEEE Trans. Evol. Comput.*, vol. 9, no. 3, pp. 303–317, Jun. 2005.
- [17] Z. Zhou, G. Si, Y. Zhang, and K. Zheng, "Robust clustering by identifying the vein of clusters based on kernel density estimation," *Knowledge Based Syst.*, vol. 159, no. 1, pp. 309–320, 2018.
- [18] K. Kim, R. B. McKay, and B.-R. Moon, "Multiobjective evolutionary algorithms for dynamic social network clustering," in *Proc. Genet. Evol. Comput. Conf.*, 2011, pp. 1179–1186.
- [19] Q. Li, J. Zhou, S. Yang, J. Zheng, and G. Ruan, "A predictive strategy based on special points for evolutionary dynamic multi-objective optimization," *Soft Comput.*, vol. 23, no. 11, pp. 3723–3739, Jun. 2019.
- [20] C.-A. Liu and Y. Wang, "New evolutionary algorithm for dynamic multiobjective optimization problems," in *Proc. Evol. Comput. Theory Algorithms*, vol. 4221, 2006, pp. 889–892.
- [21] R. C. Liu, Y. Y. Chen, W. P. Ma, C. H. Mu, and L. C. Jiao, "A novel cooperative coevolutionary dynamic multi-objective optimization algorithm using a new predictive model," *Soft Comput.*, vol. 18, no. 10, pp. 1913–1929, 2014.
- [22] A. Muruganantham, K. C. Tan, and P. Vadakkepat, "Evolutionary dynamic multiobjective optimization via Kalman filter prediction," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2862–2873, Dec. 2016.
- [23] D. Nauck, K. F. Klawonn, and K. Rudolf, *Foundations of Neuro-Fuzzy Systems*. Chichester, U.K.: Wiley, 1997.
- [24] G. Ruan, G. Yu, J. Zheng, J. Zou, and S. Yang, "The effect of diversity maintenance on prediction in dynamic multi-objective optimization," *Appl. Soft Comput.*, vol. 58, pp. 631–647, Sep. 2017.
- [25] J. R. Schott, *Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization*. Cambridge, MA, USA: Massachusetts Inst. Technol., 1995.
- [26] J. Tang, S. Alam, C. Lokan, and H. A. Abbass, "A multi-objective evolutionary method for dynamic airspace re-sectorization using sectors clipping and similarities," in *Proc. IEEE Congr. Evol. Comput.*, 2012, pp. 3565–3572.
- [27] Y. Tian, R. Cheng, X. Y. Zhang, F. Cheng, and Y. Jin, "An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 609–622, Aug. 2018.
- [28] S. Jiang *et al.*, "An empirical study of dynamic triobjective optimisation problems," in *Proc. IEEE Congr. Evol. Comput.*, 2018, pp. 1–8.
- [29] P. P.-Y. Wu, D. Campbell, and T. Merz, "Multi-objective four-dimensional vehicle motion planning in large dynamic environments," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 3, pp. 621–634, Jun. 2011.
- [30] Y. Wu, J. Yao, and X. Liu, "A directed search strategy for evolutionary dynamic multiobjective optimization," *Soft Comput.*, vol. 19, no. 11, pp. 3221–3235, 2015.
- [31] S. Zeng, S. Chen, J. Zhao, A. Zhou, Z. Li, and H. Jing, "Dynamic constrained multi-objective model for solving constrained optimization problem," in *Proc. IEEE Congr. Evol. Comput.*, 2011, pp. 2041–2046.
- [32] Z. Zhang, "Multiobjective optimization immune algorithm in dynamic environments and its application to greenhouse control," *Appl. Soft Comput.*, vol. 8, no. 2, pp. 959–971, 2008.
- [33] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, "Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization," in *Proc. 4th Int. Conf. Evol. Multi Criterion Optim.*, vol. 4403, 2007, pp. 832–846.
- [34] A. Zhou, Y. Jin, and Q. Zhang, "A population prediction strategy for evolutionary dynamic multiobjective optimization," *IEEE Trans. Cybern.*, vol. 44, no. 1, pp. 40–53, Jan. 2014.
- [35] J. Zou, Q. Li, S. Yang, B. Hu, and J. Zheng, "A prediction strategy based on center points and knee points for evolutionary dynamic multi-objective optimization," *Appl. Soft Comput.*, vol. 61, pp. 806–818, Dec. 2017.
- [36] S. Nguyen, M. Zhang, M. Johnston, and K. C. Tan, "Automatic design of scheduling policies for dynamic multi-objective job shop scheduling via cooperative coevolution genetic programming," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 193–208, Apr. 2014.
- [37] X.-H. Yan, B.-G. Cai, B. Ning, and W. Shangguan, "Moving horizon optimization of dynamic trajectory planning for high-speed train operation," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 5, pp. 1258–1270, May 2016.
- [38] P. A. Navrátil, H. Childs, D. S. Fussell, and C. Lin, "Exploring the spectrum of dynamic scheduling algorithms for scalable distributed-memory ray tracing," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 6, pp. 893–906, Jun. 2014.
- [39] A. Yazici, G. Kirlik, O. Parlaktuna, and A. Sipahioglu, "A dynamic path planning approach for multirobot sensor-based coverage considering energy constraints," *IEEE Trans. Cybern.*, vol. 44, no. 3, pp. 305–314, Mar. 2014.
- [40] G. Feng, L. Yuan, X. Zhang, and Z. Qian, "Dynamic adjustment of hidden node parameters for extreme learning machine," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 279–288, Feb. 2015.
- [41] D. J. Yu, J. Hu, Q. M. Li, Z. M. Tang, J. Y. Yang, and H. B. Shen, "Constructing query-driven dynamic machine learning model with application to protein-ligand binding sites prediction," *IEEE Trans. Nanobiosci.*, vol. 14, no. 1, pp. 45–58, Jan. 2015.
- [42] Y. Bravo, G. Luque, and E. Alba, "Global memory schemes for dynamic optimization," *Nat. Comput.*, vol. 15, no. 2, pp. 319–333, 2016.
- [43] S. B. Gee, K. C. Tan, and H. A. Abbass, "A benchmark test suite for dynamic evolutionary multiobjective optimization," *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 461–472, Feb. 2017.
- [44] F. Pieri and A. Cilea, "A fast multiobjective optimization strategy for single-axis electromagnetic MOEMS micromirrors," *Micromachines*, vol. 9, no. 1, pp. 1–14, 2018.
- [45] F. Rahmani, K. Behzadian, and A. Ardeshir, "Rehabilitation of a water distribution system using sequential multiobjective optimization models," *J. Water Resources Plan. Manag.*, vol. 142, no. 5, 2015, Art. no. C4015003.
- [46] P. Rohlfschagen and X. Yao, "Dynamic combinatorial optimization problems: An analysis of the subset sum problem," *Soft Comput.*, vol. 15, no. 9, pp. 1723–1734, 2015.
- [47] J. Xiong, Z. Zhou, K. Tian, T. Liao, and J. Shi, "A multi-objective approach for weapon selection and planning problems in dynamic environments," *J. Ind. Manag. Optim.*, vol. 13, no. 3, pp. 1189–1211, 2017.
- [48] X. Yu, Y. Jin, K. Tang, and X. Yao, "Robust optimization over time—A new perspective on dynamic optimization problems," in *Proc. IEEE Congr. Evol. Comput.*, 2010, pp. 3998–4003.
- [49] M. Jiang, Z. Q. Huang, L. M. Qiu, W. Z. Huang, and G. G. Yen, "Transfer learning-based dynamic multiobjective optimization algorithms," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 501–514, Aug. 2018.
- [50] M. Mavrovouniotis, C. H. Li, and S. Yang, "A survey of swarm intelligence for dynamic optimization: Algorithms and applications," *Swarm Evol. Comput.*, vol. 33, pp. 1–17, Apr. 2017.



Qingyang Zhang received the Ph.D. degree from the School of Computer and Information, Hefei University of Technology, Hefei, China, in 2019.

He is currently a Lecturer with the School of Computer Science and Technology, Jiangsu Normal University, Xuzhou, China. Current research interests include dynamic optimization, evolutionary computation, and image processing.



Shengxiang Yang (M'00–SM'14) received the Ph.D. degree from Northeastern University, Shenyang, China, in 1999.

He is currently a Professor of Computational Intelligence and the Director of the Centre for Computational Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester, U.K. He has over 270 publications with over 8500 Google Scholar citations and an H -index of 49. His current research interests include evolutionary computation, swarm intelligence, artificial neural networks, data mining and data stream mining, and relevant real-world applications.

Dr. Yang serves as an Associate Editor/Editorial Board Member of a number of international journals, such as the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the IEEE TRANSACTIONS ON CYBERNETICS, *Information Sciences*, *Enterprise Information Systems*, and *Soft Computing*.



Ronggui Wang received the Ph.D. degree in computer science from the Hefei University of Technology, Hefei, China, in 2005.

He is currently a Professor with the School of Computer and Information, Hefei University of Technology. His current research interests include digital image processing, artificial intelligence, and data mining.



Shouyong Jiang received the Ph.D. degree in computer science from De Montfort University, Leicester, U.K., in 2017.

He is currently a Lecturer with the School of Computer Science, University of Lincoln, Lincoln, U.K. His current research interests include computational modeling, network analysis, evolutionary computation, and dynamic optimization.



Xiaoli Li received the Ph.D. degree in control science and engineering from Northeastern University, Shenyang, China, in 2000.

He is currently a Professor with the Faculty of Information Technology, Beijing University of Technology, Beijing, China. His current research interests include complex system modeling, optimization and control, and intelligent control.