

Multidirectional Prediction Approach for Dynamic Multi-Objective Optimization Problems

Miao Rong, Dunwei Gong*, *Member, IEEE*, Yong Zhang, Yaochu Jin, *Fellow, IEEE*,
and Witold Pedrycz, *Fellow, IEEE*

Abstract—Various real-world multi-objective optimization problems are dynamic, requiring evolutionary algorithms to be able to rapidly track the moving Pareto front of an optimization problem once an environmental change occurs. To this end, several methods have been developed to predict the new location of the moving Pareto set so that the population can be reinitialized around the predicted location. In this paper, we present a multidirectional prediction strategy to enhance the performance of evolutionary algorithms in solving a dynamic multi-objective optimization problem. To more accurately predict the moving location of the Pareto set, the population is clustered into a number of representative groups by a proposed classification strategy, where the number of clusters is adapted according to the intensity of the environmental change. To examine the performance of the developed algorithm, the proposed prediction strategy is compared with four state-of-the-art prediction methods under the framework of particle swarm optimization as well as five popular evolutionary algorithms for dynamic multi-objective optimization. Our experimental results demonstrate that the proposed algorithm can effectively tackle dynamic multi-objective optimization problems.

Index Terms—dynamic multi-objective optimization, multi-direction prediction, representative individual, adaptation.

I. INTRODUCTION

Dynamic multi-objective optimization problems (DMOPs) has found an increasing number of applications in both engineering [1]–[6] and science [7]–[9]. Compared with dynamic single objective optimization problems [10], DMOPs pose a bigger challenge to an optimizer in tracking the moving Pareto set (PS) or Pareto front (PF) in a changing environment [11]–[14].

Evolutionary algorithms (EAs) have been recognized as one of the most powerful optimization algorithms [15], [16]. The first attempt to solve DMOPs using EAs can be traced back to the 1960s and solving DMOPs has become popular over the last decade [17]. Several surveys of dynamic multi-objective evolutionary algorithms ((d)MOEAs) can be found in the literature, e.g., [18]–[21].

*Corresponding author: D.W. Gong. Email: dwgong@vip.163.com.

M. Rong, D.W. Gong, and Y. Zhang are with the School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, 221006 China. D.W. Gong is also with School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao, 266061 China.

Y. Jin is with the Department of Computer Science, University of Surrey, Guildford, GU2 7XH, United Kingdom. He is also with the Department of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China. Email: yaochu.jin@surrey.ac.uk.

W. Pedrycz is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, T6G 1H7, Canada. Email: w-pedrycz@ualberta.ca.

As the change of the environment may exhibit some predictable patterns [13], it is feasible to learn these patterns based on historical information collected during the search so as to predict the new location of the Pareto set [11]. Numerous studies have been reported following this idea to exploit the predictability of dynamic environments [21]. In this paper, we focus on the prediction method with the purpose of generating a new population around the predicted PS due to its effectiveness evidenced in solving a variety of DMOPs. First, a multiple-direction prediction approach is proposed to deal with DMOPs. Second, a method to adjust the number of representative individuals is developed to strike a balance between convergence and the computational complexity. Third, a framework for incorporating the multi-direction prediction method into a particle swarm optimization (PSO) algorithm is presented and its effectiveness is empirically verified. Finally, in the experimental part, the proposed prediction method is compared with four commonly used prediction strategies as well as four popular algorithms for addressing DMOPs. Furthermore, to investigate the efficiency of the proposed clustering strategy, the adaptive number of clusters is compared with the fixed number under dynamic environment. The assumption that two consecutive changes are similar is taken into consideration. In comparison to the previous work [22], we have adjusted the method, proposed new test instances, enriched the comparison experiments, and set three pairs of frequency and severity values in experiments to investigate the performance of all compared algorithms.

The paper is organized as follows. Section II provides a review of previous work on DMOPs, in particular those predicting the center of the PS. The proposed prediction strategy based on the representative individuals is detailed in Section III. Section IV presents a general framework for incorporating the proposed prediction strategy into a PSO algorithm. The experimental settings are provided in Section V, followed by the experimental results and discussions in Section VI. Section VII concludes the paper.

II. RELATED WORK

DMOPs can be described in terms of various dynamisms, like most research papers [16], [23] and Farina et al. ever classified DMOPs with respect to the change in the decision and the objective spaces over time [23]. In this paper, we

mainly consider the following DMOP:

$$\begin{cases} \min_{x \in \Omega} F(x, t) = (f_1(x, t), f_2(x, t), \dots, f_M(x, t))^T \\ \text{s. t. } x \in \Omega = \prod_{i=1}^n [a_i, b_i] \end{cases} \quad (1)$$

where $x = (x_1, \dots, x_n)^T$ is the decision variable in the decision space, Ω , and n is the dimension of Ω . $t \in T = \{1, 2, \dots\}$ refers to the time scale. $F(x, t) : R^n \times T \rightarrow R^M$ consists of M objectives varying over time with the i -th objective being $f_i(x, t)$, $i = 1, \dots, M$. R^M denotes the objective space.

A. Prediction-based Methods

A variety of prediction-based methods for tracking moving optima have been proposed, which can be largely classified into two categories, namely, individual-based prediction [24] and population-based prediction [4], [25]–[27]. In the former approach, the moving trajectory of each individual in the population is predicted, whereas in the latter approach, one single prediction model is employed for the entire population. Linear regression models [24], [28], autoregressive models (AR models) [26], [29], or Kalman Filter-based models [30] have been used for prediction.

Zhou et al. [24] proposed an individual-based prediction method, RPS, which builds up a prediction model for each individual at time t . The model is built by relating an individual to its parent, defined to be the closest neighbouring individual at $t - 1$. Accordingly, once an environmental change has been detected, a new location will be predicted for each individual using its prediction model. As this method builds a prediction model for each individual, it suffers from a large computational complexity.

By contrast, a population-based prediction method, termed PPS, was proposed in [26], where only one time series model, an AR model, is employed for predicting the whole population. This method assumes that the PS can be described by a PS center (an abstract point) and a PS manifold, and a prediction model is built for the PS center only. Once the new PS center has been predicted, the PS manifold will be directly shifted to the new PS model. This method exhibits a good convergence performance and is computationally efficient. However, for an optimization problem with a non-translational PS when the environment changes, PPS does not work well.

Inspired by PPS, a few other methods have been proposed. For example, Peng et al. [25] proposed a method combining the prediction of the PS center with a memory strategy. Wu et al. [27] proposed a directed search strategy (denoted as PPS-Lin), which predicts the PS center using a linear model, and creates new candidates between the predicted PS and the current PS.

B. PS Center-based Prediction Methods

Due to its outstanding performances, the idea of predicting the geometrical center of the PS of a DMOP has attracted increasing interest. For example, in [4], [25]–[27], the positions of the PS center in previous environments are employed to predict the location of the PS center in the new environment.

Individuals are then generated around the predicted PS center to initialize the population for the new environment. The PS center at time t is defined as follows:

$$C_i^t = \frac{1}{|PS^t|} \sum_{i=1}^{|PS^t|} x_i^t \quad (2)$$

where PS^t is the PS at time t , $|PS^t|$ refers to the number of solutions in PS^t , and x_i^t means the i -th solution in PS^t . Fig. 1 illustrates the change of the PS and its center of the first two decision variables of FDA1 [13]. Since this problem has a translational PS, the change of the PS center is able to fully describe the variation of the PS, if a manifold and its center is used to describe the PS as done in [26]. It works well particularly for a class of DMOPs, where the change of the PS can be captured by the center of the PS.

According to [31], Jiang and Yang has pointed out that the dynamic nature in the decision variable space can be captured by the function vector, S , in a DMOP model and $S(x_I) = S(x_{II} - g(x_I))$. Therefore, if we make some modifications of $g(x_I)$, we can obtain various changing patterns of the PS. Fun7 in Table I exhibits such behavior whose PS changes over time as follows:

$$PS : x_1 \in [0, 1], x_i = G(t) \cdot x_1, i = 2, \dots, n \quad (3)$$

herein, n is the dimension of the variable space.

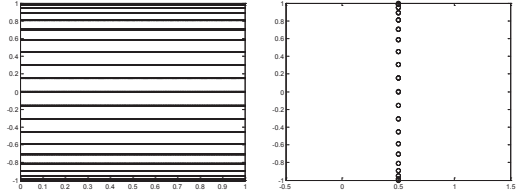


Fig. 1. The change of the PS over time (left panel) and its center (right panel) of the first two decision variables of FDA1 over time.

Fig. 2 depicts the change of the PS and its center of Fun7, where the PS rotates along the origin of the coordinate system. Although the PS center changes in a similar manner as that of FDA1, the whole PS change is much different from that of FDA1. As a result, predicting the PS center only is not able to adequately capture the change of the PS and generating individuals around the predicted PS center is not efficient in approximating the whole PS in the new environment. It is of

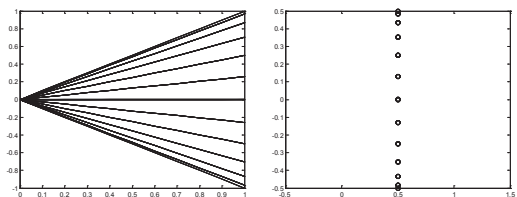


Fig. 2. The change of the PS (left panel) and its center (right panel) of the first two decision variables of Fun7 over time.

great interest to note that x_I in $g(x_I)$ is governed in the form

$$g(x_I) = G(t) \cdot (x_I - l) \quad (4)$$

herein, l is an array composed of n_I real numbers (n_I is the dimension of x_I), and l is an adjustable parameter. Then the

crossing locations of PSs at different time instances will vary. To investigate the condition that PS rotates with not only the origin of the coordinate system, we design another variant (Fun8) of FDA1.

If a DMOP has an adjustable parameter of $[0, 0.5]$, its another variant (Fun8 is such) in Fig. 3 (described in Table I), will occur, whose PS rotates around the PS center over time as follows:

$$PS : x_1 \in [0, 1], x_i = G(t) \cdot (x_1 - 0.5), i = 2, \dots, n \quad (5)$$

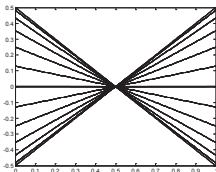


Fig. 3. A PS rotating along the PS center.

Clearly, predicting the PS center will fail to predict the change of the PS of such DMOPs.

To summarize, predicting the center of the PS for accelerating convergence of an EA for solving DMOPs may become inefficient or even completely fail when the change of the PS cannot be sufficiently captured by the change of the PS center. It is therefore necessary to develop new EAs that are able to deal with a wide range of DMOPs.

III. THE PROPOSED MULTIDIRECTIONAL PREDICTION STRATEGY

In this section, we propose a new evolutionary algorithm for solving DMOPs, which aims to generate a population close to the true PS, and to overcome the weaknesses of the prediction approaches mentioned above. The main idea is to introduce multiple models for predicting the new location of the PS once an environmental change occurs. To achieve this target, multiple representative individuals that can adequately describe the shape and the diversity of the PS are first stored at each time instant. Once an environmental change has been detected, the evolutionary trajectories are estimated using these representative individuals in the previous two time instants. Finally, new candidates are generated along the predicted new location of the PS, hoping to produce a good guess of the position of the new PS, thereby speeding up convergence of the algorithm responding to the environmental change.

A. Selection of the Representative Individuals by Adaptive Clustering

In this section, we propose a computationally efficient clustering method for selecting representative individuals. Unlike the method proposed in [32], and that presented in [17] targeting at evenly segmenting the PS by using adaptive and K-means clustering methods, respectively, at the cost of a high computational complexity, the proposed cluster strategy selects a number of individuals which can describe the location and the diversity of a Pareto set at different time instances,

recognized as representative individuals. The selection process is inspired by the cell division. It first selects a number of initial representative individuals, followed by classifying individuals in the current approximated PS into a number of clusters according to their distances to these representative individuals.

If the representative individuals are not sufficient, the cluster with the largest radius, which is defined as the largest distance between individuals and the representative individual in this cluster, will be selected, and the individual with the largest distance is regarded as a newly added representative individual. Following that, individuals in the current approximated PS are re-classified into their corresponding clusters. This selection and division process will continue until all the required representative individuals have been found. The proposed strategy saves computation as a result of avoiding repeatedly calculating and comparing distances between all the individuals.

Due to the high efficiency of employing the calculated PS center to guide the evolution of a population, it is desirable to retain the advantage of the PS center. Besides, since extreme points are the farthest on one dimension in the decision space, they can make a good description of the location of the PS. To this end, we consider the PS center combined with extreme points of the corresponding PF as the initial representative individuals. Let the numbers of objectives and the required representative individuals be M and K , respectively. The current approximate PS is denoted as PS^t with a size of N , whereas the true PS is labelled as ps^t . If $M + 1$ is less than K , then the steps of the selection method can be described as Algorithm 1.

During the selection and the division phases, after each new representative individual is selected, all the other individuals are re-categorized, leading to a number of uniform PS segments. Therefore, these representative individuals have good performances in distribution and convergence to describe the diversity and the location of the PS. The total computation complexity of the above process is $O(NK)$, which is very computationally efficient. Herein, the number of representative individuals, K , adapts with the severity of the environmental change as discussed in Section IV-A. The analysis of the complexity is presented in Section IV-C1. To easily understand the above process, we also provide an example in supplementary materials.

B. A Multidirectional Prediction Strategy

We term a multidirectional prediction strategy, which uses multiple representative solutions for prediction, rather than the PS center only, MDP strategy. The MDP strategy constructs multiple time series models based on historical information provided by the representative individuals in the previous two environments to predict a number of evolutionary directions (trajectories). In this way, the movement or dynamics of the PS can be predicted sufficiently.

The details of the prediction method are given as follows. Let $C^t = \{c_1^t, c_2^t, \dots, c_{K'}^t\}$ and $C^{t-1} = \{c_1^{t-1}, c_2^{t-1}, \dots, c_{K''}^{t-1}\}$ be two representative individual sets at time instances t and $t - 1$, respectively. For an individual c_i^t ($i = 1, \dots, K'$) in C^t , the proposed strategy first seeks its

Algorithm 1 the pseudo code of selecting representative individuals

Input: the current approximate PS, PS^t ; the number of representative individuals, K ; the number of objectives, M ;
Output: a set of representative individuals, C ;
 1: Label the center of PS^t and all the M extreme solutions of the PF as the initial representative individuals, and include them into a set, C ;
 2: Calculate the Euclidean distance between an individual, x_i , in PS^t and each of the individuals in C , and categorize x_i into its corresponding cluster, $L_j(x_i)$, $j = 1, 2, \dots, |L|$ is the label of a category;
 3: **if** $|C| > K$ **then**
 4: Output $|C|$ representative individuals, and stop selecting;
 5: **else**
 6: **for** each cluster, L_j , $j = 1, 2, \dots, |L|$ **do**
 7: Identify the furthest individual of L_j , denoted as h_j ;
 8: **end for**
 9: Select the furthest individual among all h_j , labelled as *point*, and include it into C ;
 10: Go to Step 2;
 11: **end if**
 12: **return** C ;

nearest representative individual in set C^{t-1} , denoted as c_j^{t-1} , which is regarded as the parent of c_i^t . Herein, the Euclidean distance is measured. Then, the evolutionary direction of individual c_i^t is estimated based on the movement from c_j^{t-1} to the current position as follows.

$$\Delta c_i^t = c_i^t - c_j^{t-1} \quad (6)$$

From (6), each representative individual has its own evolutionary direction determined by the individual and its parent. Furthermore, all these directions are beneficial to building the scenario of the new PS. Since the representative individuals obtained by the method proposed in Subsection III-A have a good performance in depicting the PS, the multiple directions are likely to precisely show the change trend of the PS.

C. Generation of the Predicted Individuals

We attempt to generate a number of individuals around the predicted PS to achieve a rapid response to the environmental change in the evolutionary search. These predicted individuals are also expected to help the population converge to the new true PF with good distribution. Once an environmental change is detected, the following process is carried out.

Let the current population be P^t of size N_P , the current optimal set be PS^t of size N , the representative individual set at time t be $C^t = \{c_1^t, \dots, c_K^t\}$, and the predicted evolutionary direction set be $\Delta C = \{\Delta c_1^t, \dots, \Delta c_K^t\}$. For an individual, x_i^{t+1} , which belongs to the j -th cluster, its new candidate, x_i^{t+1} , can be generated as follows:

$$x_i^{t+1} = x_i^t + \Delta c_j^t + \varepsilon^t \quad (7)$$

where $i = 1, 2, \dots, K$, and $\varepsilon^t \sim N(0, \sigma^t)$ is a random number created from a normal distribution of mean 0 and variance σ^t , which is calculated according to the following expression:

$$\sigma^t = \frac{1}{K} \sum_{i=1}^K |\Delta c_i^t| \quad (8)$$

Following the above process, N new candidates can be generated to predict the new location of the PS. To improve the diversity of the population, the rest $N_P - N$ individuals are

randomly generated in the decision space. These solutions together constitute the initial population for evolutionary search in the new environment, which are expected to help rapidly locate a well-distributed PS in the new environment.

IV. AN MDP BASED PARTICLE SWARM OPTIMIZER

In principle, the MDP strategy can be embedded in any population-based meta-heuristics for dynamic multi-objective optimization problems. Without loss of generality, we embed MDP into a PSO framework for dynamic multi-objective optimization to verify the effectiveness of MDP for dynamic optimization. Before giving the framework of the whole procedure, a strategy for adaptively determining the number of representative individuals is first presented to further enhance the performance.

A. Adaptive Election of Representative Individuals

The optimal number of representative solutions for predicting the new locations of the Pareto optimal solutions in the PS may depend on the shape of the Pareto front and the severity of the change in the PS when there is an environmental change. In the following, we propose a mechanism for adaptively determining the number of representative solutions. To this end, we define a measure for reflecting the severity of environmental changes, $\delta(t)$, which is indicated by the degree of the change in the objective values of N individuals before and after the environment change:

$$\delta(t) = \frac{1}{M} \frac{1}{N} \sum_{j=1}^M \sum_{i=1}^N |\Delta f_j(Ar_i) - \mu_j(g)| \quad (9)$$

where $\Delta f_j(Ar_i) = \frac{f_j(Ar_i, g) - f_j(Ar_i, g-1)}{u_j(g) - l_j(g)}$ and $\mu_j(g) = \frac{1}{N} \sum_{i=1}^N \left| \frac{f_j(Ar_i, g) - f_j(Ar_i, g-1)}{u_j(g) - l_j(g)} \right|$. Herein, $\Delta f_j(Ar_i)$ is the j -th objective value of individual i in the archive at the g -th iteration, $u_j(g)$ and $l_j(g)$ refer to the maximal and the minimal values on the j -th objective at the g -th iteration, respectively.

Based on the severity of the environmental change defined above, the number of representative individuals, K , can be calculated. The basic idea is that the more severe the environment change is, the more representative individuals should be chosen, and vice versa. More specifically, the value of K is determined as follows:

$$K = \lceil K_1 + \delta(t) * (K_2 - K_1) \rceil \quad (10)$$

where K_1 and K_2 are the lower and upper limits of K . According to Section III-A, we set $K_1 = M + 1$. A too large or small value of K_2 will both pose a negative influence on the performance of the algorithm. By realizing some preliminary experiments shown in the supplementary materials, we finally set $K_2 = 3M$.

B. The Proposed PSO Framework

The PSO framework incorporated with the MDP strategy for dynamic multi-objective optimization is presented as Algorithm 2. First, the swarm is initialized by randomly generating N_P

particles in the search space (**Lines 2 to 6**). With respect to change detection, according to [33], an environmental change can be detected by reevaluating the location of \vec{p}_g at each iteration. When the current fitness value of \vec{p}_g varies from its stored one at the last iteration, it is recognized that an environmental change occurs (**Lines 11 to 12**). If a change occurs, recall the change response procedure described as Algorithm 3, generate an initial swarm for the PSO optimizer (**Line 13**), and the current time instance comes to $t + 1$ (**Line 14**); otherwise, optimize the t -th MOP by using the PSO optimizer for one generation, at the beginning of which period, the personal best particle (**Line 20**), the global best particle (**Line 21**) are updated. Then, the location and the velocity of each particle are updated (**Line 22**). This process is described in Appendix. If the number of non-dominated solutions is larger than the size of archive N (**Line 29**), the crowding distance [34] is calculated for all the $|Ar|$ particles, and N individuals with the largest crowding distance are retained. Following that, include all individuals in Ar into ps^t . When the stopping criterion has been met, the program is terminated and the PS is output; otherwise, go to **Line 10**.

As shown in Algorithm 3, in case that there is an environmental change, the fitness values of individuals at the last iteration are reevaluated under the new environment, and the degree of the change is obtained according to formula 9 (**Line 4**). Next, K representative individuals are chosen (**Line 6**) to build up K evolutionary trajectories (**Line 7**), which are used to generate N new candidates as a part of the initial population for the new environment combined with $(N_P - N)$ individuals randomly generated in the initial population (**Line 8**). Finally, complete boundary checks for individuals in the swarm.

C. Complexity of the MDP Framework

1) *Analysis of the Proposed Selection Process*: Since the complexity of calculating the distance between each pair of individuals in the PS is much higher than that of comparing these distances, the complexity of the proposed method is determined by that of calculating all the distances. Taking the worst case into consideration, only one individual is chosen to generate a cluster each time. First, the calculation of the distance between each of $(N - M)$ individuals and each of M representative individuals costs $O((N - M)M)$. Then, it takes $O((N - M - 1)(M + 1))$ to obtain the $(M + 1)$ -th cluster, and $O((N - K)K)$ to achieve the K -th cluster.

The total computational complexity of the above process is $O((N - M)(K - M))$. Since $N \gg K > M$, the computational complexity of the proposed method can be simplified as $O(NK)$, which is very computationally efficient. This can be attributed to the fact that it seeks representative individuals with a good diversity resulting from the segmentation style of a cluster.

2) Analysis of MDP framework over One Generation:

In Algorithm 2, the computing overhead is $O(MN_P^2)$ for the first loop (**Lines 2-5**). For the second loop (**Lines 7-35**), the computation complexity is mainly concerned with fitness calculation. At the first stage (Lines 8-12), it takes $O(MN_D)$. Then (**Line 14**), it costs $O(NK)$ for Change Response as discussed in Section IV-C1. Following that (Lines

Algorithm 2 Framework of MDP

Input: the swarm size, N_P ; the archive size, N ; the number of particles used for change detection, N_D ; the maximal number of iterations, g_{max} ;
Output: an approximate Pareto optimal set Op ;
1: $t = 0, g = 0$; %Initialize the swarm, pop
2: **for** $i = 1$ to N_P **do**
3: initialize \vec{x}_i, \vec{v}_i ;
4: calculate $f(\vec{x}_i)$;
5: $\vec{p}_i \leftarrow \vec{x}_i$; $f(\vec{p}_i) \leftarrow f(\vec{x}_i)$;
6: **end for**
7: **repeat**
8: Randomly Select N_D individuals;
9: $\vec{p}_g \leftarrow \arg \min_{\vec{p}_i} \{f(\vec{p}_i) | i = 1, 2, \dots, N_D\}$;
10: $f(\vec{p}_g) \leftarrow \min(f(\vec{p}_i)); f_{record} \leftarrow f(\vec{p}_g)$;
11: $g \leftarrow g + 1$; %Detect and respond to an environmental change
12: calculate $f(\vec{p}_g)$;
13: **if** $f(\vec{p}_g) \neq f_{record}$ **then** %Re-initialize the swarm, pop
14: Change Response (Algorithm 3);
15: $t \leftarrow t + 1$;
16: **for** $i = 1$ to N_P **do**
17: $\vec{x}_i = pop_i$;
18: **end for**
19: **else**
20: **for** $i = 1$ to N_P **do**
21: $pbest_i \leftarrow get_pbest()$;
22: $gbest_i \leftarrow get_gbest()$;
23: $\vec{x}_i \leftarrow update$;
24: **end for**
25: **end if**
26: **for** $i = 1$ to N_P **do**
27: calculate $f(\vec{x}_i)$;
28: $Ar \leftarrow nondominate(pop)$;
29: **end for**
30: **if** $|Ar| > N$ **then**
31: Retain N individuals with the largest crowding distances;
32: $ps^t \leftarrow Ar$;
33: Update \vec{p}_i and \vec{p}_g ;
34: **end if**
35: **until** $g > g_{max}$
36: $Op \leftarrow ps$ and stop the algorithm; % Output the obtained PS
37: **return** Op ;

Algorithm 3 Change Response in MDP

Input: the time instance, t ; the approximate PS at t and $t - 1$, ps_t and ps_{t-1} , respectively; the size of ps_t , N ; the swarm size, N_P ;
Output: an initial swarm, pop ;
1: **for** $i = 1$ to N **do**
2: calculate $f(\vec{x}_i)$;
3: **end for**
4: Calculate the degree of the change, δ , according to formula 9;
5: Calculate the number of representative individuals, K , according to formula 10;
6: $C^t \leftarrow select(ps^t, K)$;
7: $\Delta_C \leftarrow predict(C^t, C^{t-1}, K)$;
8: $pop = generate(ps^t, \Delta_C, \epsilon, N, N_P)$;
9: Do boundary check for individuals in pop ;
10: **return** pop ;

16-29), the searching process completed by the optimizer spends $O(MN_P^2)$ on fitness computation and on average $O(N^2 \lg N)$ [35], [36] on elitist preservation. As a result, the computational complexity of the MDP framework is $O(MN_P^2)$ or $O(N^2 \lg N)$, depending on whichever is larger. Jiang and Yang have pointed out that under fast-changing environment, the complexity would reach $O(N^2 \lg N)$ [36].

V. EXPERIMENTAL SETTINGS

A. Algorithms under Comparison

To evaluate the performances of the proposed MDP strategy in dynamic multi-objective optimization, we compare it with three popular dynamic multi-objective evolutionary algorithms ((d)MOEAs), dynamic nondominated sorting genetic algorithm II (dNSGA-II) [37], steady-state and generational evolutionary algorithm (SGEA) [36], and dynamic multi-objective

particle swarm optimization (DMOPSO) [38], and one classic multiobjective evolutionary algorithm, MOEA based on decomposition (MOEA/D) [39]. It should be emphasised that by comparison between DMOPSO and MDP, we can see clearly whether the proposed prediction strategy, MDP, or the standard PSO, contribute more to the good performance since MDP is incorporated into DMOPSO. Besides, to investigate the efficiency of the proposed clustering method in Section III-A, we also take a K-means clustering-based method (denoted as Kmeans) proposed by Halder et al [17] into consideration for comparative study. Moreover, four popular change response methods, PPS [26], PPS-Lin(in Section II-A), RPS [24], and RIS [24], are incorporated into PSO due to useful properties that PSO has demonstrated in addressing DMOPs [40], [41] to complete comparative of the proposed MDP. The settings of those four strategies are presented in the supplementary materials.

Similar to [16] and [36], the experiments are conducted for various combinations of change severity levels and frequencies, i.e., $(n_t, \tau_t)=(5,10)$, $(10,10)$, and $(20,10)$, in order to study the influence of the change frequency on algorithms' ability in dynamic environments. Each algorithm is run independently 20 times on each test instance, and the algorithm is stopped when $t > 200$, $t > 100$, and $t > 50$ according to the above settings of the frequency. At each generation, 5% solutions randomly selected from the PS are re-evaluated for detecting environment changes (i.e., $N_D = 5\% * N$) [23], [30]. In this work, the dimensionality of the decision space of the test problems is set as $n = 10$ as Farina and Deb has done [23]. The swarm size is $N_P = 150$, and the archive size (the PS size) is $N = 100$ like the settings in [22] and [35].

B. Test Instances

We compare the algorithms on 11 test problems with two or three objectives, among which there are three newly proposed problems. They are FDA1-5 proposed by Farina et al. [23], a variant of FDA1 without an adjustable parameter (Fun7), a variant of FDA2 with the adjustable parameter of [0.5] (Fun8), another variant of FDA2 (Fun9), F8 proposed by Zhou et al. [26], JY5 proposed by Jiang and Yang [16], and GTA1a proposed by Gee et al. [13]. The details of the three proposed test problems are listed in Table I, whose types can be analyzed according to [23]. The reason behind employing the two variants of FDA1, Fun7, and Fun8, lies in that they correspond to the two occasions, namely, rotating with the origin of the coordinate system and the PS center. When adding the adjustable parameter for $g(x_I)$, we can adjust the cross location of PSs at different time and investigate the usefulness of the developed method.

In Table I, both PF and PS of all the three test cases change over time. A difference lies in that the PSs of Fun7 and Fun9 rotate with the original center of the coordinate system, whereas that of Fun8 rotates along the PS centroid. During their changes, the shapes of the PSs remain unchanged.

C. The Performance Metric

For assessing the performance of convergence and diversity of solutions obtained by different prediction strategies, Zhou

TABLE I
THREE NEWLY PROPOSED TEST INSTANCES

Instance	Definition
Fun7 (Type II)	$\begin{cases} f_1 = x_1 \\ f_2 = g \cdot h \\ g = 1 + H + \sum_{i=2}^n (x_i - G(t) \cdot x_1)^2 \\ h(f_1, g) = 1 - \sqrt{\left(\frac{f_1}{g}\right)} \\ G(t) = \sin(t \cdot \pi/12) \\ H(t) = 0.75 \sin(0.5\pi t) + 1.25, t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_t} \right\rfloor \\ \text{where : } x_i \in [0, 1], \forall i = 1, 2, \dots, n \end{cases}$
Fun8 (Type II)	$\begin{cases} f_1 = x_1 \\ f_2 = g \cdot h \\ g = 1 + \sum_{i=2}^n (x_i - G(t) \cdot (x_1 - 0.5))^2 \\ G(t) = \sin(t \cdot \pi/12) \\ h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^{H(t)-1} \\ H(t) = 0.75 + 0.7 \sin(0.5\pi t), t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_t} \right\rfloor \\ \text{where : } x_1 \in [0, 1]; x_i \in [-1, 1], i = 2, \dots, n \end{cases}$
Fun9 (Type II)	$\begin{cases} f_1 = x_1 \\ f_2 = g \cdot h \\ g = 1 + \sum_{i=2}^n (x_i - G(t) \cdot x_1)^2 \\ G(t) = \sin(t \cdot \pi/12) \\ h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^{H(t)} \\ H(t) = 1 + \sin(0.5\pi t), t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_t} \right\rfloor \\ \text{where : } x_1 \in [0, 1]; x_i \in [-1, 1], i = 2, \dots, n \end{cases}$

et al. [26] introduced the MIGD metric, where the average IGD value of a period of time is calculated. We present more details of the MIGD metric in the supplementary material.

Since the IGD metric depends mainly on the closeness, distribution, and coverage of an approximation to the true PF, the SP and HVD metrics combined with MIGD can help us deeply and comprehensively understand the performances of each algorithm [36]. The Spacing Metric (SP) [42] is employed to compare the distribution throughout the optima obtained by all the six (d)MOEAs. The HVD metric [36] is the difference between hypervolume of the obtained PS and the true PS to measure the convergence of the obtained PS.

Furthermore, to qualify whether there is significance difference between solutions produced by the proposed prediction method and the one such as [26], the t-test at the 0.05 significance level is engaged. If its value is smaller than 5%, then one concludes that a significant difference between the two exists; this observation is labeled as "+" after the value. Otherwise, there is a "-" after the result value.

VI. EXPERIMENTAL RESULTS AND DISCUSSIONS

In this section, we first observe the SP, HVD, and MIGD performances of all the five popular (d)MOEAs under all three different pairs of frequency and severity, followed by further discussions on the IGD values at each time instant in the initial and final populations obtained by the proposed prediction strategy and the other four compared strategies with the frequency of 10. Due to limited space, the average MIGD values of the initial populations and the final (after 30 generations) populations obtained for the four compared prediction strategies over 20 runs are investigated in supplementary materials. The reason for comparing the MIGD value

obtained after 30 generations is to investigate whether the prediction strategies can improve the optimizer, or whether the given computational cost is enough to tackle a stationary multi-objective optimization problem.

The SP, HVD, and MIGD values after 30 generations obtained by the four compared (d)MOEAs and the Kmeans-based clustering method (Kmeans) are presented in Tables II, III, and IV. Since the initial populations obtained by the other three algorithms are generated randomly, their initial populations are no longer estimated.

A. Combined with Popular (d)MOEAs

Tables II, III, and IV list the mean and the standard deviation values of SP, HVD, and MIGD metrics obtained by MDP and the other five popular (d)MOEAs over 20 runs with 200, 100, and 50 environmental changes, respectively.

Table II shows that, (1) MDP obtains the best results for the majority of test instances in terms of the SP metric, indicating that MDP achieves solutions with a better performance in uniformly distributing along the true PF than the compared algorithms. (2) MDP mainly loses good distribution for FDA2 and FDA3, when the environment changes rapidly (i.e., $\tau_t = 5$ or 10), for which dNSGA-II shows a superior performance in distribution. (3) The SP value of dNSGA-II for FDA3 when $\tau_t = 10$ is slightly smaller than that of MDP, the t -test value indicates that there is no significant difference between these two solution sets. (4) For Fun7, SGEA achieves slightly better SP values than MDP when $\tau_t = 5$ and 10, implying that for an optimization problem with a rotating PS, strategies with the aim of improving the performance of MDP in distribution are expected. (5) For complex dynamic problems like F8, JY5, and GTA1a, optima obtained by MOEAD usually exhibit better diversity. Compared to that, our method, MDP, should incorporate some strategy to improve the diversity of optimal solutions when dealing with complex DMOPs.

It is worth noting that compared to K-means, whose number of clustering centers is fixed with a large computational complexity of $O(N_I^2k)$, where N_I represents the number of individuals to be clustered, and k means the number of clusters. In this paper, $N_I = N$ and $k = K$. The proposed clustering method with an adaptive number of representative individuals is suitable for dynamic scenarios, even though it shows a poor performance in distribution for FDA4, and Fun7 with a rapid environment change. For the other instances like FDA2, FDA5, and GTA1a, MDP achieves solutions with a remarkably better performance in distribution than Kmeans. For complex problems such as F8, JY5, and GTA1a, MDP obtains better HVD results than DMOPSO no matter what the frequency is. For some complex cases with both changed PSs and PFs, like FDA5, JY5, and GTA1a, when the environmental changing speed is fast (the frequency value is 5), MDP always achieves a set of optima with noticeably better convergence than those produced by DMOPSO. This indicates that after adding the strategy proposed in Section III, the convergence speed of DMOPSO can be greatly improved. When the changing speed is low with the frequency value of 20, SGEA obtains better results than MDP in some cases like FDA1, FDA5,

and F8, suggesting that for DMOPs with translational PSs, when the changing speed is not high, the prediction method employing only the centroid individual is sufficient to well trace the change.

From Table III, (1) MDP performs well in terms of convergence for most test instances, especially those with rotating PSs, suggesting that the prediction strategy incorporated into MDP is effective. (2) For Fun7 with all the frequencies of the environment change, MDP achieves a better performance in terms of the HVD indicator than the compared algorithms, especially K-means. (3) SGEA outperforms MDP on FDA2 and FDA5, no matter what the change frequency is, suggesting that improving the performance of MDP in convergence is of necessity for DMOPs with changing PFs. (4) MDP greatly improves the convergence of DMOPSO in almost all test cases with any changing frequency, suggesting that by incorporating the proposed prediction strategy into it, PSO can trace dynamic environments with better convergence.

The MIGD metric measures the performance of a solution set in terms of both convergence and distribution. As a result, if an algorithm achieves solutions with a poor performance in diversity, it is possible that it exhibits a good MIGD value on account of a superior performance in convergence. For example, SGEA has an average SP value of 3.5744E-2 for FDA5 with 50 environment changes, which is worse than 2.9009E-2 obtained by MDP, whereas the former obtains an average HVD value of 1.3933E+0, 10% smaller than MDP. Under this circumstance, SGEA still achieves a better MIGD value than MDP (2.3052E-1 versus 2.9917E-1). For FDA2, SGEA obtains the best HVD value, and dNSGA-II, DMOPSO, and MOEA/D also obtain better results than MDP when the environment changes rapidly, implying that a large gap is left to improve the performance of MDP when tackling DMOPs with changing PFs.

The following observations are drawn from Table IV. (1) MDP obtains the best results for the majority of test instances in terms of the MIGD metric, and loses mainly for DMOPs with either changing PFs or three objectives, like FDA2 and FDA4, where SGEA and DMOPSO are the best, respectively. (2) SGEA loses its superiority for FDA2 with the change frequency of 5 due to its inferior performance in distribution. Similarly, SGEA is down by DMOPSO for FDA4 when the change frequency is 10. (3) For FDA4, K-means with $\tau_t = 5$ achieves solutions with a better SP value than MDP, MDP is still remarkably competitive in terms of the MIGD metric resulted from its superb performance in convergence (its HVD value under this scenario is extremely better than that of K-means, 1.1303E+0 versus 3.3443E+0). The reason lies in that MDP can make full use of historical information in previous environments to predict new locations and initialize the population with candidates generated around the predicted locations.

From Tables III and IV, approximately consistent conclusions can be drawn from the HVD and MIGD indicators. Although MDP achieves comparatively good SP values for FDA2, FDA4, and FDA5 at a small change frequency, like dNSGA-II, SGEA, and DMOPSO, its MIGD values are slightly worse than those of the three compared algorithms due to

TABLE II
THE MEAN AND STANDARD DEVIATION VALUES OF SP OBTAINED BY SIX ALGORITHMS.

	MOEAD	dNSGA-II	SGEA	Kmeans	DMOPSO	MDP
FDA1	5,10	7.0675E-1(8.2266E-3)+	3.2997E-2(3.6136E-3)+	9.7019E-3(7.4175E-3)+	8.7711E-2(7.7764E-3)+	1.5757E-1(3.5951E-1)+
	10,10	6.7935E-1(8.4396E-3)+	1.3263E-2(3.7334E-3)+	7.0437E-3(7.7707E-3)+	5.9607E-2(7.4129E-3)+	6.5079E-2(1.5608E-2)+
	20,10	3.1430E-2(9.8322E-3)+	6.3180E-3(3.8221E-3)+	3.1796E-3(8.1414E-3)	3.6239E-2(9.3936E-3)+	7.0969E-3(3.4293E-1)+
FDA2	5,10	2.0975E-2(9.9283E-2)+	8.0399E-3(8.2560E-3)	1.2019E-2(2.4161E-2)+	8.1142E-2(9.9281E-2)+	9.6236E-2(3.4584E-1)+
	10,10	1.5325E-2(9.2506E-2)+	5.4151E-3(7.5477E-3)	8.2437E-3(2.2816E-2)+	6.6778E-2(9.3822E-2)+	6.9997E-2(2.4298E-1)+
	20,10	1.1788E-2(9.4693E-2)+	5.1621E-3(1.7778E-2)+	5.0835E-3(1.0215E-2)	4.0729E-2(1.0204E-1)+	8.1830E-2(2.2150E-1)+
FDA3	5,10	9.0322E-2(3.2991E-3)+	1.8453E-2(5.9298E-3)	3.9870E-2(8.2666E-3)+	3.8476E-2(3.1915E-3)+	3.7810E-2(1.2650E-2)+
	10,10	4.5941E-2(5.6344E-3)+	1.1454E-2(5.1407E-3)	2.2608E-2(1.4364E-2)+	2.1323E-2(4.1214E-3)+	2.7770E-2(1.2617E-2)+
	20,10	3.3888E-2(7.0476E-3)+	8.9359E-3(4.9990E-3)	2.1213E-2(1.3160E-2)+	2.7185E-2(5.5576E-3)+	1.8156E-2(1.2900E-2)+
FDA4	5,10	2.0322E-1(3.2991E-3)+	1.1908E-1(1.8506E-2)+	8.5146E-2(1.1478E-1)+	8.4027E-2(2.1295E-2)	2.1412E-1(3.9458E-3)+
	10,10	1.8223E-2(2.7133E-2)+	8.9110E-2(3.4180E-2)+	4.7341E-2(1.3894E-1)+	5.1323E-2(4.1214E-3)+	1.1577E-1(7.9568E-3)+
	20,10	1.1887E-2(1.0413E-2)+	6.2056E-2(1.9014E-2)+	2.9553E-2(1.5676E-1)+	3.3840E-2(1.5310E-2)+	8.2942E-2(9.0622E-3)+
FDA5	5,10	1.5655E-1(3.6627E-2)+	2.1614E-1(3.8940E-2)+	8.6090E-2(6.9641E-2)+	9.0929E-2(4.1573E-2)+	1.3294E-1(8.9648E-2)+
	10,10	1.0596E-1(3.5620E-2)+	1.2277E-1(5.8813E-2)+	5.6581E-2(6.2166E-2)+	6.0942E-1(4.1514E-2)+	7.3193E-2(9.6022E-2)+
	20,10	1.0573E-1(3.7701E-2)+	8.1706E-2(3.4297E-2)+	3.5744E-2(6.6338E-2)+	5.0812E-1(3.7563E-3)+	5.3761E-1(9.2300E-2)+
Fun7	5,10	9.1099E-3(6.1099E-3)	1.3464E-1(3.1762E-3)+	1.8950E-2(1.0409E-2)+	2.6114E-2(7.4638E-3)+	7.5046E-2(8.4605E-3)+
	10,10	1.7154E-2(6.7154E-3)	9.3569E-2(2.7081E-3)+	1.6876E-2(8.7322E-3)+	1.5070E-2(6.3842E-3)+	4.2393E-2(9.1089E-3)+
	20,10	1.1133E-2(9.1133E-3)	7.3804E-2(2.8826E-3)+	1.3974E-2(7.8408E-3)+	1.4449E-2(6.2017E-3)+	1.2732E-2(8.0063E-3)+
Fun9	5,10	1.2218E-2(2.2984E-2)+	7.1010E-2(2.1485E-3)+	2.0538E-2(9.9909E-3)+	3.1878E-2(1.2358E-1)+	3.9826E-2(2.0048E-2)+
	10,10	7.0620E-1(2.0741E-2)+	4.0076E-2(1.7074E-3)+	1.2210E-2(8.6199E-3)+	1.1874E-2(3.7788E-2)+	1.9323E-2(1.8874E-2)+
	20,10	4.1834E-1(1.3987E-1)+	9.1553E-3(1.7439E-3)+	8.5274E-2(9.1854E-3)+	1.1225E-2(6.8735E-1)+	9.8958E-3(1.3706E-1)+
F8	5,10	4.5565E-2(2.4974E-3)	1.7870E-1(6.0491E-2)+	4.2739E-1(2.9811E-1)+	6.8712E-2(6.7253E-3)+	1.9260E-1(3.6125E-2)+
	10,10	4.5713E-2(2.2674E-3)	1.7827E-1(5.6005E-2)+	4.6281E-1(3.0678E-1)+	7.0987E-2(8.1535E-3)+	1.9395E-1(3.6902E-2)+
	20,10	4.5302E-2(2.5580E-3)	1.6414E-1(5.2122E-2)+	4.7788E-1(2.9465E-1)+	7.3657E-2(6.0413E-3)+	1.9323E-1(3.6886E-2)+
JY5	5,10	5.9601E-3(2.4516E-3)	2.5896E-2(7.3772E-3)+	3.5657E-2(3.4906E-2)+	3.4393E-2(4.7848E-3)+	5.7412E-2(1.5631E-2)+
	10,10	5.6382E-3(5.1967E-3)	2.5985E-2(1.1630E-2)+	3.3260E-2(3.7234E-2)+	3.2543E-2(6.3022E-3)+	5.7651E-2(1.7771E-2)+
	20,10	5.9522E-3(2.1649E-3)	2.6677E-2(7.7950E-3)+	3.5888E-2(2.5699E-2)+	3.4519E-2(6.7784E-3)+	5.3897E-2(1.6895E-2)+
GTA1a	5,10	2.6962E-3(1.2747E-3)	1.0035E-2(3.8316E-3)+	2.0420E-2(1.1369E-2)+	1.3127E-2(3.4619E-3)+	2.5565E-2(5.8286E-3)+
	10,10	2.6438E-3(8.0269E-4)	1.0034E-2(2.9576E-3)+	1.9906E-2(1.1890E-2)+	1.3704E-2(2.0954E-3)+	2.5758E-2(3.8861E-3)+
	20,10	2.6170E-3(1.1258E-3)	1.0233E-2(3.8166E-3)+	2.0136E-2(1.0811E-2)+	1.4872E-2(4.2360E-3)+	2.8270E-2(5.8707E-3)+

TABLE III
THE MEAN AND STANDARD DEVIATION VALUES OF HVD OBTAINED BY SIX ALGORITHMS.

	MOEAD	dNSGA-II	SGEA	Kmeans	DMOPSO	MDP
FDA1	5,10	7.5968E-1(1.9934E-1)+	9.0814E-1(1.0144E-1)+	8.6303E-2(1.8776E-1)	7.6927E-1(1.8696E-1)+	6.8303E-1(2.4698E-1)+
	10,10	2.1766E-1(5.8494E-1)+	1.6916E-1(2.8816E-1)+	4.1653E-2(5.5150E-1)	1.1995E-1(5.4385E-1)+	1.4024E-1(6.2808E-1)+
	20,10	1.9104E-1(1.9362E-1)+	4.3031E-2(1.0550E-1)+	3.7108E-2(1.8176E-1)	6.9461E-2(1.8078E-1)+	4.6297E-2(2.3765E-1)+
FDA2	5,10	1.5672E-1(4.6104E-2)+	4.3603E-2(1.1109E-1)+	2.1468E-2(1.1811E-1)	2.4487E-1(1.7804E-1)+	9.8441E-2(7.0174E-1)+
	10,10	6.3088E-2(1.2030E-1)+	2.1588E-2(3.2266E-1)+	1.4051E-2(3.7220E-1)	7.4659E-2(5.2418E-1)+	3.1763E-2(2.2898E-1)+
	20,10	2.8670E-2(1.7362E-1)+	1.1884E-2(1.1062E-1)+	1.1633E-2(1.2295E-1)	5.5203E-2(1.7417E-1)+	2.1546E-2(2.1821E-1)+
FDA3	5,10	1.5832E+0(2.3557E-3)+	2.4767E+0(1.7009E-2)+	9.6924E+1(1.1737E-1)+	6.4936E+0(1.6073E-2)+	1.2979E+0(4.7163E-2)+
	10,10	1.1745E+0(7.0309E-3)+	1.9407E+0(5.7737E-2)+	9.2047E+1(3.6284E-1)+	1.9440E+0(5.7631E-2)+	7.2690E-1(1.4187E-1)+
	20,10	9.5840E-1(2.2629E-3)+	1.4315E-1(2.6451E-2)+	8.5400E-1(1.1795E-1)+	6.4634E-1(2.4022E-2)+	5.8829E-1(4.7095E-2)+
FDA4	5,10	3.9182E+0(1.0274E-2)+	3.3928E+0(1.8614E-2)+	1.3612E+0(2.9535E-2)+	3.3443E+0(2.0847E-2)+	1.4065E+0(2.5226E-2)+
	10,10	1.1744E+0(3.1348E-2)+	1.0128E+0(5.7983E-2)+	2.7085E+1(8.8438E-2)+	1.0044E+0(6.2143E-2)+	1.0208E-1(7.4858E-2)
	20,10	3.9147E-1(1.0132E-2)+	3.3944E-1(1.9914E-2)+	1.3657E-1(3.0204E-2)+	7.3449E-1(2.1685E-2)+	9.4177E-2(2.5979E-2)+
FDA5	5,10	7.3289E+0(9.8879E-2)+	6.8257E+0(6.9449E-2)+	3.2193E+0(2.8440E-2)+	2.8164E+0(6.8993E-2)+	3.2020E+0(4.4064E-03)+
	10,10	4.9810E-2(3.4257E-3)+	5.5465E+0(1.8732E-1)+	1.7871E+0(7.7884E-2)+	1.8847E+0(1.9409E-1)+	1.6728E+0(6.0011E-2)+
	20,10	2.7423E+0(1.0620E-1)+	2.5700E+0(3.0409E-3)+	1.3933E+0(2.1852E-2)	1.9456E+0(7.3062E-2)+	1.9209E+0(3.0703E-2)+
Fun7	5,10	1.9729E+0(1.0716E-1)+	1.5002E+0(3.1154E-2)+	1.8606E+1(5.8141E-2)+	2.3362E-1(3.3504E-1)+	7.9109E-1(4.3556E-1)+
	10,10	9.9459E-1(3.0177E-1)+	7.3634E-1(7.8159E-2)+	1.0238E-1(1.5707E-1)+	9.6628E-2(1.1651E-1)+	2.3520E-1(2.1487E-1)+
	20,10	5.6741E-1(9.6689E-2)+	4.3392E-1(2.5036E-2)+	8.3606E-2(4.1684E-1)+	7.7918E-2(1.1248E-1)+	5.3193E-2(4.4860E-1)+
Fun9	5,10	5.5298E+0(2.4027E-1)+	4.3230E+0(2.9095E-1)+	3.6045E+0(2.7870E-1)+	3.6045E+0(4.3807E-1)+	3.3855E+0(2.7632E-1)+
	10,10	2.4755E+0(6.9599E-1)+	2.0928E+0(8.0907E-1)+	1.1510E+0(8.0658E-1)+	1.4100E+0(2.6608E-1)+	1.1934E+0(7.8625E-1)+
	20,10	1.0798E+0(2.3246E-1)+	1.6132E+0(2.7138E-1)+	8.1772E-1(2.8174E-1)+	8.3352E-1(2.8787E-1)+	9.9938E-1(2.6378E-1)+
F8	5,10	3.6895E-1(1.5956E-2)+	2.8519E-1(1.6455E-2)+	2.6086E-1(1.3675E-2)+	2.7616E-1(1.3311E-2)+	3.4654E-1(1.8550E-2)+
	10,10	3.6909E-1(1.5695E-2)+	4.3888E-1(1.4509E-2)+	1.6580E-1(2.9677E-2)	2.7426E-1(1.3396E-2)+	3.5832E-1(1.8324E-2)+
	20,10	3.6821E-1(1.6202E-2)+	3.9240E-1(1.9500E-2)+	1.5300E-1(2.3014E-2)	2.6604E-1(1.4117E-2)+	2.7160E-1(1.79540E-2)+
JY5	5,10	4.8505E-1(2.7027E-3)+	2.4563E-1(3.8416E-2)	4.8453E-1(3.3356E-3)+	4.8385E-1(3.3770E-3)+	4.3804E-1(1.8012E-2)+
	10,10	4.8855E-1(2.8139E-3)+	2.4859E-1(3.9808E-2)	4.8331E-1(6.1294E-3)+	4.8277E-1(4.0807E-3)+	4.3888E-1(1.4509E-2)+
	20,10	5.3818E-1(2.7168E-3)+	4.6510E-1(1.9445E-2)+	5.1768E-1(1.0853E-2)+	5.2292E-1(6.5555E-2)+	3.8247E-1(1.8855E-2)+
GTA1a	5,10	5.3531E-1(6.8633E-2)+	6.2578E-1(4.9125E-3)+	3.4461E-1(4.8014E-2)+	6.3820E-1(4.5327E-3)+	5.7625E-1(8.1383E-3)+
	10,10	5.3881E-1(6.9508E-2)+	6.2568E-1(4.7690E-3)+	3.0355E-1(3.1047E-2)+	6.3644E-1(4.0251E-3)+	5.7662E-1(6.9616E-3)+
	20,10	5.3081E-1(6.6053E-2)+	3.4511E-1(7.5111E-2)+	2.8542E-1(6.2952E-2)	6.3070E-1(5.5722E-3)+	5.7126E-1(7.6411E-3)+

its inferior performance in convergence. Generally, MDP has promising performances in both convergence and diversity for almost all the test instances.

Note that neither MOEA/D nor any (d)MOEA is as good as MDP for the two modified test instances, despite that some optimizers indeed obtain solutions with a better performance in distribution than MDP. Taking Fun7 as an example, when $\tau_t = 5$, MOEA/D achieves a smaller average SP value of 9.1099E-2, whereas its average HVD value is approximately double that of MDP. This also corresponds to the results of the last three complex cases, even though optima obtained by

MOEAD have good distribution in the variable space, their convergence is inferior to other methods. This explains why the MIGD values of MOEAD are not as good as the SP values, suggesting that it is hard for MOEAD to tackle changes well within limited number of generations. It is also worth emphasizing that for Fun9, SGEA achieves a better average MIGD value than MOEA/D, dNSGA-II, and K-means, since both its SP and HVD values are better than those of the other four algorithms. This might be attributed to the PS center-based prediction strategy. However, most results assessed in terms of these three metrics obtained by MDP outperform

TABLE IV
THE MEAN AND STANDARD DEVIATION VALUES OF MIGD OBTAINED BY SIX ALGORITHMS.

		MOEAD	gNSGA-II	SGEA	Kmeans	DMOPSO	MDP
FDA1	5,10	5.8581E-1(1.2949E-1)+	6.8087E-1(1.0458E-1)+	4.3646E-2(1.1581E-1)+	1.2149E-1(1.1839E-2)+	5.6083E-1(1.4380E-2)+	1.3309E-2(1.8065E-2)
	10,10	5.15295E-1(1.2962E-1)+	5.7834E-2(1.1658E-1)+	2.6032E-2(6.4880E-2)+	5.1627E-2(1.7388E-2)+	9.4374E-2(2.6460E-2)+	1.0102E-2(1.3811E-2)
	20,10	5.2748E-2(1.2481E-1)+	3.5542E-2(9.8937E-2)+	8.0430E-3(2.0764E-2)	1.0156E-2(1.2238E-2)+	4.8487E-2(3.3211E-2)+	9.6083E-3(1.4380E-2)
FDA2	5,10	9.6858E-2(2.5313E-2)+	3.8301E-2(7.5366E-1)+	2.6321E-2(1.0697E-1)+	6.1180E-2(1.0478E-1)+	6.4186E-1(2.0210E-1)+	1.6267E-2(1.1007E-1)
	10,10	4.2270E-2(2.5447E-2)+	2.5465E-2(7.6954E-1)+	1.0040E-2(1.1254E-1)	2.7028E-2(1.4621E-2)+	2.2537E-2(9.1265E-2)+	1.3710E-2(3.5475E-2)-
	20,10	2.3943E-2(1.0305E-1)+	8.7558E-3(7.2101E-1)-	8.5996E-3(1.1138E-1)	1.0940E-2(1.0432E-1)+	1.3527E-2(8.9855E-2)+	8.8031E-3(1.0177E-2)-
FDA3	5,10	2.4296E-1(1.6177E-2)+	7.3660E-1(5.1949E-1)+	8.4503E-2(9.1298E-2)+	1.9368E-1(9.9420E-3)+	3.6439E-1(2.1745E-2)+	5.1770E-2(1.4827E-3)
	10,10	1.3910E-1(9.9393E-2)+	7.3414E-1(5.1398E-1)+	6.3653E-2(9.9025E-2)+	6.0840E-2(1.0498E-1)+	1.8321E-2(1.8549E-2)+	3.1696E-2(1.3688E-3)
	20,10	5.1476E-2(1.0821E-2)+	7.1278E-2(5.0873E-2)+	4.3336E-2(8.4422E-2)+	4.9416E-2(1.2182E-2)+	4.4914E-2(1.5001E-2)+	2.1412E-2(1.5756E-3)
FDA4	5,10	9.4521E-1(3.1081E-3)+	8.1086E-1(1.8507E-1)+	4.7261E-1(3.8865E-2)	5.4260E-1(8.9028E-3)+	5.5174E-1(2.2030E-2)+	5.4178E-1(8.4546E-3)+
	10,10	4.4506E-1(3.1627E-3)+	7.9145E-1(1.6980E-1)+	2.6989E-1(3.6746E-2)+	2.4351E-1(1.1127E-2)+	2.5114E-1(2.1645E-2)+	1.4123E-1(1.1863E-2)
	20,10	2.4400E-1(3.3575E-3)+	3.8834E-1(2.0401E-1)+	1.6530E-1(3.2698E-2)+	1.4225E-1(8.8014E-3)+	1.0056E-1(2.4897E-2)	1.1298E-1(7.7567E-3)-
FDA5	5,10	1.5598E+0(2.5792E-1)+	1.5326E+0(5.0293E-1)+	7.3654E-1(4.3482E-1)+	8.5443E-1(2.5794E-1)+	6.6165E-1(3.0073E-1)+	6.0996E-1(2.4361E-1)
	10,10	6.4232E-1(8.0181E-3)+	1.0381E+0(5.4510E-1)+	4.3562E-1(4.2892E-1)+	5.4994E-1(2.6446E-1)+	4.5288E-1(3.3488E-1)+	4.0009E-1(2.5226E-1)
	20,10	4.1842E-1(2.5212E-1)+	6.5228E-1(5.2509E-1)+	2.3052E-1(4.2109E-1)	4.8620E-1(2.6455E-1)+	4.0866E-1(2.7541E-1)+	2.9917E-1(2.4795E-1)+
Fun7	5,10	5.9886E-1(1.7818E-2)+	7.0062E-1(1.5300E-1)+	4.2209E-1(2.2319E-1)+	1.0693E-1(7.5694E-2)+	9.0566E-2(2.0221E-2)+	5.6616E-2(1.4004E-2)
	10,10	2.9253E-1(2.5704E-1)+	3.9641E-1(1.4028E-1)+	1.9102E-1(2.1256E-1)+	7.8384E-2(7.6385E-2)+	8.9923E-2(1.7468E-2)+	2.6506E-2(2.0049E-2)
	20,10	9.4915E-2(1.1028E-2)+	1.9391E-1(1.4463E-1)+	6.5790E-2(2.0155E-1)+	7.4637E-2(7.6830E-2)+	8.1126E-2(1.7814E-2)+	2.0589E-2(1.0728E-2)
Fun9	5,10	6.1981E-2(7.8512E-2)+	3.5107E-1(7.1842E-2)+	2.0712E-1(8.3201E-2)+	3.8651E-1(5.6648E-3)+	3.3505E-1(3.2251E-2)+	3.9669E-2(2.3494E-2)
	10,10	4.0313E-2(7.1148E-2)+	1.4431E-1(7.0747E-2)+	1.5215E-1(8.2641E-2)+	2.3255E-1(1.2278E-1)+	8.5539E-2(3.2620E-2)+	2.6285E-2(1.8475E-2)
	20,10	3.8392E-2(7.1122E-2)+	8.3988E-2(7.106E-2)+	5.2842E-2(3.3549E-2)+	8.2127E-2(6.3344E-1)+	6.2891E-2(8.8536E-2)+	2.0917E-2(1.5688E-2)
F8	5,10	9.9896E-1(3.1972E-4)+	1.2536E+0(2.8063E-1)+	1.0280E+0(1.1875E-2)+	1.0157E+0(5.4602E-3)+	1.1304E+0(3.4385E-2)+	4.3596E-1(1.1911E-2)
	10,10	9.9896E-1(3.2497E-4)+	1.9053E+0(2.9657E-1)+	1.0453E+0(2.2877E-2)+	1.0184E+0(7.4097E-3)+	1.1312E+0(3.9520E-2)+	4.3658E-1(1.5001E-2)
	20,10	9.9893E-1(3.0033E-4)+	1.2305E+0(2.7107E-1)+	1.0668E+0(2.8077E-2)+	1.0208E+0(6.0626E-3)+	1.1356E+0(4.2817E-2)+	4.3791E-1(1.4093E-2)
JYS	5,10	9.7839E-3(1.6861E-3)	2.8854E-1(6.3592E-2)+	1.0448E-2(1.7678E-3)+	1.1486E-2(1.7421E-3)+	1.0759E-1(1.6447E-2)+	3.9897E-2(1.1302E-2)+
	10,10	4.1506E-2(9.9728E-3)+	2.8772E-1(6.7315E-2)+	1.3526E-2(3.4434E-3)+	1.4320E-2(1.2600E-2)+	1.1197E-1(1.2995E-2)+	1.0030E-2(1.7344E-3)
	20,10	1.9534E-1(1.6341E-3)+	2.6910E-1(6.4125E-2)+	2.1987E-2(6.7049E-3)	9.9190E-2(4.0730E-2)+	1.1480E-1(1.4819E-2)+	5.6271E-2(1.2810E-2)+
GTA1a	5,10	1.3396E-1(2.9993E-3)-	3.3313E-1(6.4633E-2)+	2.3788E-1(4.6759E-2)+	1.9554E-1(5.7502E-3)+	2.4135E-2(5.4884E-3)+	1.1372E-1(6.6791E-2)
	10,10	3.0828E-1(6.6984E-2)+	3.3821E-1(8.0017E-2)+	2.7473E-1(3.0984E-2)+	1.9071E-1(4.6416E-3)+	2.5209E-1(5.2024E-3)+	1.3798E-1(3.4731E-3)
	20,10	2.3385E-1(6.9262E-2)+	3.1203E-1(7.1881E-2)+	1.1785E-1(6.5354E-2)+	1.5280E-1(3.9476E-3)+	1.7302E-1(3.9531E-3)+	9.6270E-2(6.3600E-3)

those obtained by SGEA, implying the superiority and the adaptability of MDP for various changes of the PS.

Finally, since the SP, HVD, and MIGD values obtained by MDP are smaller than those of K-means for most scenarios, we can conclude that the strategy of adaptively adjusting the number of clusters is more suitable for DMOPs than that of the fixed number of clusters.

B. Further Discussion

To further understand the performances of the proposed MDP, we take a closer look into the tracking dynamics of the prediction strategies under comparisons. We first compare the IGD values obtained at each time by all the five prediction strategies. Then, some initial populations obtained by all the compared strategies, as well as the final populations after 30 generations by MDP for each newly designed DMOP are presented.

Fig.4 depicts the IGD profile averaged over 20 runs of MDP, PPS-Lin, RPS, RIS, and PPS for the eight test problems designed in this work. We can make the following observations from Fig.4.

First, when $t \geq 21$, MDP considerably outperforms the other four strategies for almost all the test instances in terms of the IGD value. In addition, when $t \geq 21$, the IGD values of solutions obtained by MDP slightly fluctuate at a low level, indicating that the proposed method is able to track the environmental change efficiently and effectively.

Second, prediction strategies based on the PS center perform unsatisfactory since their average IGD values are much larger than those of the other three prediction methods, and are even larger than the values obtained by RIS under some circumstances. These results indicate that the PS center-based prediction strategies are unsuitable for a DMOP whose PS rotates. For Fun7 with a PS rotating around the origin of

the coordinate system, RPS achieves better results than RIS. However, for Fun8, RPS achieves as poor results as RIS, suggesting that for a DMOP whose PS rotates around the PS center, a prediction strategy in which each individual seeks its own evolutionary direction according to the direction of its nearest neighbor in the previous environments cannot accurately predict.

Similar conclusions can be drawn from Figures 5 and 6, which present the initial populations at some time instants with the lowest IGD values among 20 runs obtained by MDP and compared prediction strategies with the frequency of 10. Moreover, Fig. 7 depicts the final populations on some classic test cases and the newly designed DMOPs with the lowest IGD values obtained by MDP among 20 runs when the frequency is 10.

The results indicate that the proposed MDP has achieved consistently better results than the compared state-of-the-art methods for a DMOP with a rotating PS as the environment changes and is not worse than any compared prediction strategy. Overall, MDP, RPS, PPS-Lin and PPS have better performances than RIS for DMOPs with linearly correlated decision variables. Therefore, the proposed strategy, MDP, is highly competitive when tackling both simple and complex problems, especially those with the rotating PSs.

VII. CONCLUSIONS

We have proposed a multi-directional prediction strategy to enhance the performances of population-based metaheuristics in solving dynamic multi-objective optimization problems. Every time when an environmental change occurs, a part of the population is re-initialized by the prediction strategy, and the rest will be filled with randomly generated individuals. The proposed strategy uses multiple directions determined by multiple representative solutions coming from the previous environments to predict the new location of Pareto optimal

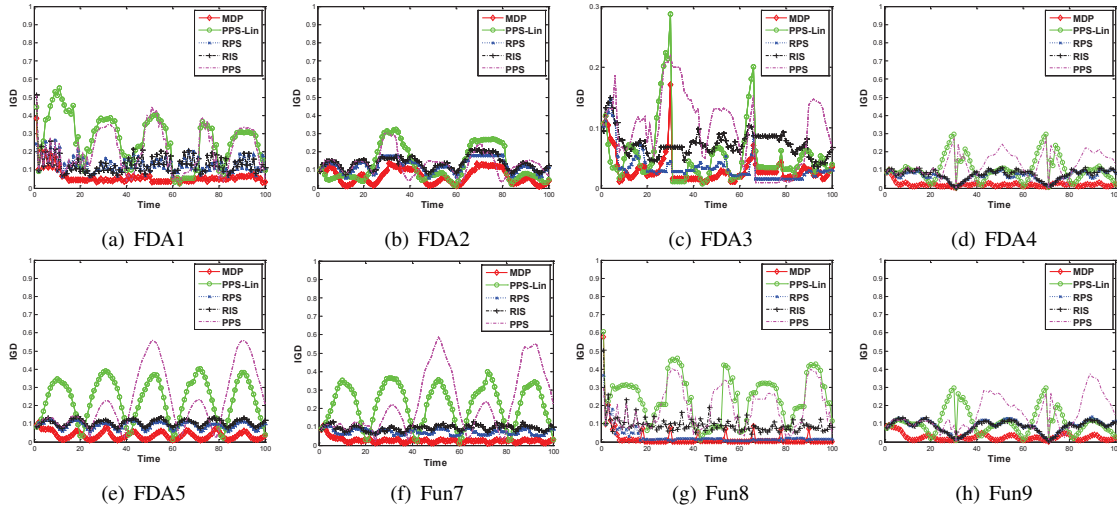


Fig. 4. The average IGD values over 20 runs versus the time scale when the frequency is 10.

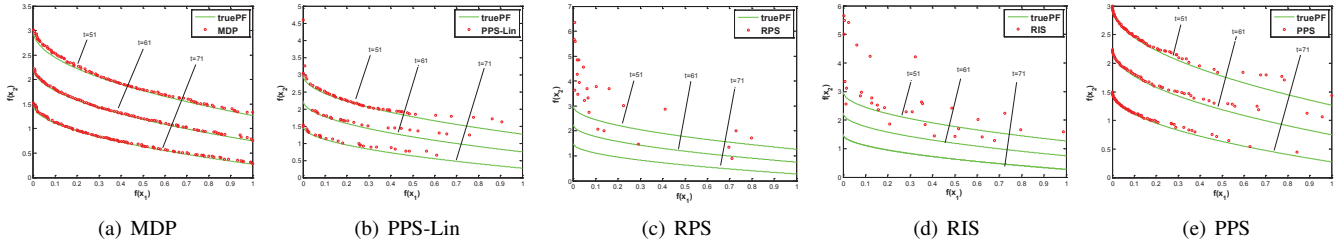


Fig. 5. The initial populations with the lowest IGD values among 20 runs on Fun7 at $t=51, 61,$ and $71,$ respectively when the frequency is 10.

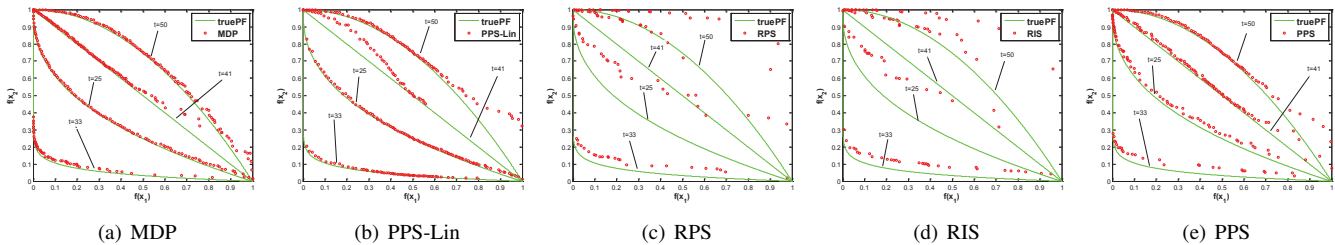


Fig. 6. The initial populations with the lowest IGD values among 20 runs on Fun8 at $t=25, 33, 41,$ and $50,$ respectively when the frequency is 10.

solutions in the decision space, around which candidates are generated to re-initialize the population whenever there is an environmental change. To best describe the shape of the Pareto set in the new environment, the number of predicted individuals is adjusted according to an estimated dynamic changes of the environment. Additionally, a new clustering method is presented in MDP, with the purpose of describing the shape of the PS with a low computing overhead.

To test the effectiveness of the proposed strategy, three new test functions are designed by modifying existing test problems for dynamic multi-objective optimization. As demonstrated in the experimental studies, the proposed strategy is very competitive compared to the number of the existing state-of-the-art prediction strategies, as well as five popular (d)MOEAs, and clearly outperforms its peers especially for those having rotating PSs when the environment changes. To better understand the usefulness of the proposed prediction strategy, the results obtained by standard PSO are investigated and compared. The results reveal that MDP greatly improves the

performance of standard PSO in addressing DMOPs. It should also be pointed out that even though MDP can obtain a set of dominated optima, the distribution of these optima are expected to be increased when dealing with complex DMOPs.

It is worth noting that various environmental changes in real-world applications are irregular, which makes the proposed method difficult to predict the new location of the Pareto set when the environment changes. With this regard, as suggested in [43], it might be desirable to seek solutions with slowly changing for an environmental change, which are known as robust solutions over time. It is of great interest to strike a good balance between tracking a changing Pareto front as fast as possible and seeking optimal solutions with slow changes when the environment changes.

Moreover, we only consider the situations when two consecutive changes are similar and linear in the decision space in this paper. More investigations are required in the future to make the proposed method applicable to more practical cases. What if we combine online learning methods with prediction

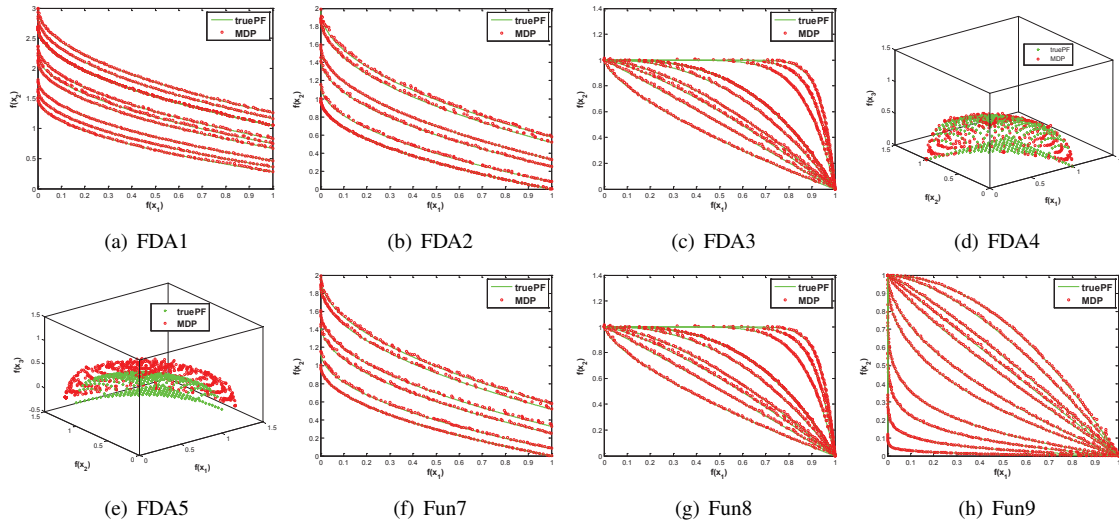


Fig. 7. The final populations obtained by MDP with the lowest IGD values among 20 runs on some classic and the newly designed scenarios when the frequency is 10.

for dealing with complex dynamic optimization problems? Herein, online learning methods aim at investigating the fundamental changing shapes rooted in history and providing useful guidance for prediction. The prediction models and the optimizers are selected automatically corresponding to the changing shapes with purpose of making a more accurate estimation for further evolutions. From the perspective of dynamic optimization, an initial population which is close to or even covers the true optimal set under the new environment is expected to be obtained. Additionally, the complexity may lie in the step change of the PS, discontinuous points existing on the PS, multiple changing ways of the PS with time (more than translation or rotation), with missing information in history, changes in the dimension of the variable space, or with varying number of objectives. These aspects are not rare in practice. Furthermore, for practical dynamic problems, how to provide an optimal solution based on a set of optimal ones should be taken into account. The decision makers' priorities should also be considered.

ACKNOWLEDGMENT

This work was jointly supported by National Natural Science Foundation of China (No. 61473299, 61573361, and 61773384), National Key R&D Program of China (No. 2018YFB1003802-01), and National Basic Research Program of China (973 Program) (No. 2014CB046306-2).

REFERENCES

- [1] S. Bera, P. Gupta, and S. Misra, "D2s: Dynamic demand scheduling in smart grid using optimal portfolio selection strategy," *IEEE Transactions on Smart Grid*, vol. 6, no. 3, pp. 1434–1442, 2014.
- [2] M. Ismail, M. Kashef, E. Serpedin, and K. Qaraqe, "On balancing energy efficiency for network operators and mobile users in dynamic planning," *Communications Magazine IEEE*, vol. 53, no. 11, pp. 158–165, 2015.
- [3] P. A. Navrátil, H. Childs, D. S. Fussell, and C. Lin, "Exploring the spectrum of dynamic scheduling algorithms for scalable distributed-memoryray tracing," *IEEE Transactions on Visualization & Computer Graphics*, vol. 20, no. 6, pp. 893–906, 2013.
- [4] D. J. Wang, F. Liu, and Y. Jin, "A multi-objective evolutionary algorithm guided by directed search for dynamic scheduling," *Computers & Operations Research*, vol. 79, pp. 279–290, 2016.
- [5] X. Yan, B. Cai, B. Ning, and W. Shangguan, "Moving horizon optimization of dynamic trajectory planning for high-speed train operation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 5, pp. 1258–1270, 2016.
- [6] A. Yazici, G. Kirlik, O. Parlaktuna, and A. Sipahioglu, "A dynamic path planning approach for multirobot sensor-based coverage considering energy constraints," *IEEE Transactions on Cybernetics*, vol. 44, no. 3, pp. 305–314, 2014.
- [7] G. Feng, L. Yuan, X. Zhang, and Z. Qian, "Dynamic adjustment of hidden node parameters for extreme learning machine," *IEEE Transactions on Cybernetics*, vol. 45, no. 2, pp. 279–288, 2015.
- [8] D. J. Yu, J. Hu, Q. M. Li, and Z. M. Tang, "Constructing query-driven dynamic machine learning model with application to protein-ligand binding sites prediction," *IEEE Transactions on Nanobioscience*, vol. 14, no. 1, pp. 44–57, 2015.
- [9] R. Zhang, Y. Lan, G. B. Huang, Z. B. Xu, and Y. C. Soh, "Dynamic extreme learning machine and its approximation capability," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 2054–2065, 2013.
- [10] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments—a survey," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 3, pp. 303–317, 2005.
- [11] Y. Bravo, G. Luque, and E. Alba, "Global memory schemes for dynamic optimization," *Natural Computing*, vol. 15, no. 2, pp. 319–333, 2016.
- [12] L. T. Bui, Z. Michalewicz, E. Parkinson, and M. B. Abello, "Adaptation in dynamic environments: A case study in mission planning," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 2, pp. 190–209, 2012.
- [13] S. B. Gee, K. C. Tan, and H. A. Abbass, "A benchmark test suite for dynamic evolutionary multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 47, no. 2, pp. 461–472, 2017.
- [14] W. F. Leong and G. G. Yen, "Pso-based multiobjective optimization with dynamic population size and adaptive local archives," *IEEE Transactions on Systems Man & Cybernetics Part B Cybernetics*, vol. 38, no. 5, pp. 1270–1293, 2008.
- [15] S. Das, A. Mandal, and R. Mukherjee, "An adaptive differential evolution algorithm for global optimization in dynamic environments," *IEEE Transactions on Cybernetics*, vol. 44, no. 6, pp. 966–978, 2014.
- [16] S. Jiang and S. Yang, "Evolutionary dynamic multiobjective optimization: Benchmarks and algorithm comparisons," *IEEE Transactions on Cybernetics*, vol. 47, no. 1, pp. 198–211, 2017.
- [17] U. Halder, S. Das, and D. Maity, "A cluster-based differential evolution algorithm with external archive for optimization in dynamic environments," *IEEE Transactions on Cybernetics*, vol. 43, no. 3, pp. 881–897, 2013.
- [18] B. Xu, Y. Zhang, D. Gong, Y. Guo, and M. Rong, "Environment sensitivity-based cooperative co-evolutionary algorithms

- for dynamic multi-objective optimization," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pp. 1–14, 2018. [Online]. Available: 10.1109/TCBB.2017.2652453
- [19] M. Helbig and A. P. Engelbrecht, "Analysing the performance of dynamic multi-objective optimisation algorithms," in *IEEE Congress on Evolutionary Computation*. IEEE, 2013, pp. 1531–1539.
- [20] T. T. Nguyen, S. Yang, and J. Branke, "Evolutionary dynamic optimization: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 6, pp. 1–24, 2012.
- [21] C. Raquel and X. Yao, "Dynamic multi-objective optimization: a survey of the state-of-the-art," in *Evolutionary Computation for Dynamic Optimization Problems*. Springer, 2013, pp. 85–106.
- [22] M. Rong, D. W. Gong, and Y. Zhang, "A multi-direction prediction approach for dynamic multi-objective optimization," in *International Conference on Intelligent Computing*. Springer, 2016, pp. 629–636.
- [23] M. Farina, K. Deb, and P. Amato, "Dynamic multiobjective optimization problems: test cases, approximations, and applications," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 5, pp. 425–442, 2004.
- [24] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, "Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization," in *Evolutionary Multi-Criterion Optimization*. Springer, 2007, pp. 832–846.
- [25] Z. Peng, J. Zheng, J. Zou, and M. Liu, "Novel prediction and memory strategies for dynamic multiobjective optimization," *Soft Computing*, vol. 19, no. 9, pp. 2633–2653, 2015.
- [26] A. Zhou, Y. Jin, and Q. Zhang, "A population prediction strategy for evolutionary dynamic multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 44, no. 1, pp. 40–53, 2014.
- [27] Y. Wu, Y. Jin, and X. Liu, "A directed search strategy for evolutionary dynamic multiobjective optimization," *Soft Computing*, vol. 19, no. 11, pp. 3221–3235, 2015.
- [28] Z. Li, H. Chen, Z. Xie, C. Chen, and A. Sallam, "Dynamic multiobjective optimization algorithm based on average distance linear prediction model," *The Scientific World Journal*, vol. 2014, no. 1, pp. 1–9, 2013.
- [29] I. Hatzakis and 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*. ACM, 2006, pp. 1201–1208.
- [30] A. Muruganantham, K. C. Tan, and P. Vadakkepat, "Evolutionary dynamic multiobjective optimization via kalman filter prediction," *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 2862–2873, 2016.
- [31] S. Jiang and S. Yang, "A framework of scalable dynamic test problems for dynamic multi-objective optimization," in *IEEE Symposium on Computational Intelligence in Dynamic & Uncertain Environments*, 2014, pp. 32–39.
- [32] S. Jiang and S. Yang, "An improved multiobjective optimization evolutionary algorithm based on decomposition for complex pareto fronts," *IEEE Transactions on Cybernetics*, vol. 46, no. 2, pp. 421–437, 2015.
- [33] L. Liu, S. Yang, and D. Wang, "Particle swarm optimization with composite particles in dynamic environments," *IEEE Transactions on Systems Man & Cybernetics: Part B Cybernetics*, vol. 40, no. 6, pp. 1634–1648, 2010.
- [34] J. Zheng, *Multi-objective Evolutionary Optimization and Its Applications*. Science Press, 2007.
- [35] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948.
- [36] S. Jiang and S. Yang, "A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 1, pp. 65–82, 2017.
- [37] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [38] X. Li, J. Branke, and M. Kirley, "On performance metrics and particle swarm methods for dynamic multiobjective optimization problems," in *IEEE Congress on Evolutionary Computation*. IEEE, 2007, pp. 576–583.
- [39] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [40] Y. Zhang, D. W. Gong, and J. H. Zhang, "Robot path planning in uncertain environment using multi-objective particle swarm optimization," *Neurocomputing*, vol. 103, no. 2, pp. 172–185, 2013.
- [41] K. R. Harrison, B. M. Ombuki-Berman, and A. P. Engelbrecht, "Dynamic multi-objective optimization using charged vector evaluated particle swarm optimization," in *IEEE Congress on Evolutionary Computation*. IEEE, 2014, pp. 1929–1936.

- [42] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.
- [43] Y. Jin, B. Sendhoff, and X. Yao, "A framework for finding robust optimal solutions over time," *Memetic Computing*, vol. 5, no. 1, pp. 3–18, 2013.



Miao Rong is currently a PhD student in control theory and control engineering of the School of Information and Electronic Engineering, China University of Mining and Technology. Her research interests include intelligence optimization and data mining.



Dunwei Gong (M'14) received the Ph.D. degree in control theory and control engineering from China University of Mining and Technology in 1999. He is a professor in the School of Information and Electrical Engineering, China University of Mining and Technology. His main research interest includes intelligence optimization and control.



Yong Zhang received the M.S. and Ph.D. degrees in control theory and control engineering from China University of Mining and Technology in 2006 and 2009, respectively. He is currently with the School of Information and Electronic Engineering, China University of Mining and Technology. His research interests include intelligence optimization and data mining.



Yaochu Jin (M'98-SM'02-F'16) received the B.Sc., M.Sc., and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 1988, 1991, and 1996 respectively, and the Dr.-Ing. degree from Ruhr University Bochum, Germany, in 2001.

He is a Professor in Computational Intelligence, Department of Computer Science, University of Surrey, Guildford, U.K., where he heads the Nature Inspired Computing and Engineering Group. He is also a Finland Distinguished Professor and a Changjiang Distinguished Visiting Professor appointed. He has

(co)authored over 200 peer-reviewed journal and conference papers and been granted eight patents on evolutionary optimization.

He is the Editor-in-Chief of the *IEEE Transactions on Cognitive and Developmental Systems* and *Complex & Intelligent Systems*. He is also an Associate Editor or Editorial Board Member of the *IEEE Transactions on Evolutionary Computation*, *IEEE Transactions on Cybernetics*, *IEEE Transactions on Nanobioscience*, *Evolutionary Computation*, *BioSystems*, *Soft Computing*, and *Natural Computing*.

Dr Jin is an IEEE Distinguished Lecturer (2017-2019). He is the recipient of the 2014 and 2016 IEEE Computational Intelligence Magazine Outstanding Paper Award, the 2017 IEEE Transactions on Evolutionary Computation Outstanding Paper Award, and the Best Paper Award of the 2010 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology. He is a Fellow of IEEE.



Witold Pedrycz (F'98) received the M.Sc, Ph.D, and DSci degrees from the Silesian University of Technology, Glwice, Poland, in 1977, 1980, and 1984, respectively. He is currently Professor and Canada Research Chair (CRC) in Computational Intelligence in the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada. He is also with the Systems Research Institute of the Polish Academy of Sciences, Warsaw, Poland. In 2009 Dr. Pedrycz was elected a foreign member of the Polish Academy of Sciences. In 2012 he was

elected a Fellow of the Royal Society of Canada. Witold Pedrycz has been a member of numerous program committees of IEEE conferences in the area of fuzzy sets and neurocomputing. In 2007 he received a prestigious Norbert Wiener award from the IEEE Systems, Man, and Cybernetics Society. He is a recipient of the IEEE Canada Computer Engineering Medal, a Cajastur Prize for Soft Computing from the European Centre for Soft Computing, a Killam Prize, and a Fuzzy Pioneer Award from the IEEE Computational Intelligence Society.

His main research directions involve Computational Intelligence, fuzzy modeling and Granular Computing, knowledge discovery and data science, fuzzy control, pattern recognition, knowledge-based neural networks, relational computing, and Software Engineering. He has published numerous papers in this area. He is also an author of 17 research monographs and edited volumes covering various aspects of Computational Intelligence, data mining, and Software Engineering.

Dr. Pedrycz is vigorously involved in editorial activities. He is an Editor-in-Chief of *Information Sciences*, Editor-in-Chief of *WIREs Data Mining and Knowledge Discovery* (Wiley), and *Int. J. of Granular Computing* (Springer). He serves on an Advisory Board of *IEEE Transactions on Fuzzy Systems* and is a member of a number of editorial boards of international journals.