Handling Dynamic Multi-objective Optimization Environments via Layered Prediction and Subspace-based Diversity Maintenance

Yaru Hu, Jinhua Zheng, Shouyong Jiang, Shengxiang Yang, Juan Zou

***Abstract*—In this paper, we propose an evolutionary algorithm based on layered prediction (LP) and subspace-based diversity maintenance (SDM) for handling dynamic multi-objective optimization environments. The LP strategy takes into account different levels of progress by different individuals in evolution and historical information to predict the population in the event of environmental changes for a prompt change response. The SDM strategy identifies gaps in population distribution and employs a gap filling technique to increase population diversity. SDM further guides rational population reproduction with a subspace-based probability model to maintain the balance between population diversity and convergence in every generation of evolution regardless of environmental changes. The proposed algorithm has been extensively studied through comparison with five state-of-the-art algorithms on a variety of test problems, demonstrating its effectiveness in dealing with dynamic multi-objective optimization problems.**

***Index Terms*—layered prediction, gap filling, subspace-based diversity maintenance, change response, dynamic multi-objective optimization.**

I. INTRODUCTION

EVOLUTIONARY algorithms (EAs) are a kind of random search approach for optimization. EAs have been successful in dealing with a wide range of optimization problems [1], [2]. In the real world, optimization problems from various application areas usually have multiple objective functions that are in conflict with each other. These problems are called multi-objective optimization problems (MOPs). Sometimes, MOPs are not stationary as they have variable objective functions and/or related constraint parameters that change over time. These kinds of MOPs are called dynamic MOPs (DMOPs).

EAs are important approaches to dynamic multi-objective optimization (DMO) for DMOPs in various application areas, including scheduling [3], network [4] and control [5]. They are able to track the time-varying Pareto-optimal set (POS) and/or Pareto-optimal front (POF) of DMOPs due to environmental changes [6]. This makes EAs advantageous over conventional DMO approaches for complex DMOPs. For example, in control sciences, EAs can handle online DMOPs with multiple time-varying objective functions, for which traditional optimal control can be difficult [7], [8], [9].

Conventional EAs are effective in solving static MOPs [10] [11]. However, they are reported to encounter difficulties for solving DMOPs unless they are properly modified [1]. This can be explained by two main reasons: **a)** Conventional EAs are not designed for approximating stationary POF and/or POS in a significantly short period of time (say 10/20 generations), which is often the case in DMO; **b)** Conventional EAs assume there is no environmental change during the optimization course, so it does not have any change response mechanisms to rescue possible diversity loss caused by environmental changes [12], [13], [14]. In other words, good DMOEAs need to address the balance between population convergence and diversity (i.e., fast convergence while preserving the diversity of solutions [15]). Consequently, much effort has been made to render EAs capable of handling DMOPs in recent years. This leads to a number of effective DMOEAs [16], [17], [18].

In dynamic environments [8], there is a short period of time (i.e., between two consecutive environmental changes) during which DMOPs stay stationary. This observation has inspired many researchers to consider DMO as a combination of dynamic handling and static multi-objective optimization [19], [20]. That is, when an environmental change occurs, a response is made to the change by either rational population re-initialization or population adjustment, followed by conventional static optimization until the arrival of the next change. This two-stage model has been popular in the DMO literature for developing effective DMOEAs. DMOEAs using this two-stage model often focus primarily on change response since many good algorithms are available for static multi-objective optimization. As a result, the above DMOEAs rely mainly on the change response stage to recover diversity loss in population. However, the change response stage often lasts only for a few generations (around 10 generations or equivalently 1000 function evaluations [20], [21]), and there is no guarantee that population diversity can be soundly recovered in such a short time. This concern motivates us to deal with dynamic environments differently.

The work [22] for dynamic single-objective optimization problems sends a clear message that maintaining population diversity all the time rather than doing so only when changes

Yaru Hu, Jinhua Zheng and Juan Zou are with the Key Laboratory of Intelligent Computing and Information Processing, Ministry of Education, Xiangtan University, Xiangtan 411105, China (e-mail: huyaru1199@gmail.com, jhzheng@xtu.edu.cn, zoujuan@xtu.edu.cn).

Shouyong Jiang is with the Department of Computer Science, University of Aberdeen, AB24 3FX, U.K. (email: math4neu@gmail.com).

Shengxiang Yang is with the Centre for Computational Intelligence (CCI), School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, U.K. (email: syang@dmu.ac.uk)

This work was supported by the research projects: the National Natural Science Foundation of China under Grant Nos. 62176228 and 61876164; the Natural Science Foundation of Hunan Province under Grant No. 2020JJ4590; the Education Department Major Project of Hunan Province under Grant No. 17A212. (*Corresponding author: Jinhua Zheng and Shouyong Jiang*).

occur is helpful for addressing environmental changes. We believe this could also be useful for DMO. Therefore, we devise a subspace-based diversity maintenance strategy in this paper, which keeps an eye on population diversity at all times regardless of environmental changes, to alleviate the situation that population diversity may not be fully recovered in a short period of change response. Additionally, we also recognize the fact that population members could have different levels of progress in evolution, which can be exploited with historical information to predict population effectively. Accordingly, a layered prediction strategy is developed, which sufficiently uses the advantage of solutions on different levels of progress and improves the accuracy of the prediction direction to some extent. These two strategies form the base of our approach to DMOPs. **The main contributions of the proposed algorithm are summarized as follows:**

i) A layered prediction strategy for change response. This strategy considers the evolutionary difference among population members and guides the relocation of fittest for a new environment by historical population information. The less fit members are updated based on the adaptability of the fittest in the new environment.

ii) A gap filling approach. It fills gaps in population distribution, thereby maintaining good diversity all the time. This strategy aims to alleviate the situation that population diversity may not be fully recovered in a short period of change response.

iii) Probability-based population reproduction. It uses population distribution to build a subspace-based probability distribution for mating selection. This mating selection strategy improves population diversity and convergence. Additionally, it requires fewer parameters than other mating selection approaches (e.g., MOEA/D-DE [23]).

The rest of this paper is organized as follows. We provide some useful definitions for this paper in Section II. Then, section III provides a review of existing studies on DMOPs. Section IV presents the proposed approach. Section V describes the experimental setting. Experiment results are presented and analyzed in Section VI. Finally, Section VII concludes this paper and discusses future research directions.

II. Basic definitions

A. Dynamic Multi-objective Optimization Problem

Without loss of generality, minimization of DMOPs considered here is unequivocally defined [1] as follows:

$$\begin{cases} \min F(x,t) = [f_1(x,t), f_2(x,t), \dots, f_m(x,t)]^T, \\ s.t.\ x \in \Omega, \end{cases} \tag{1}$$

where m is the number of objectives; $x = [x_1, ..., x_n]^T$ ($x \in \mathcal{R}^n$) is a decision variable vector consisting of n attributes and $\Omega \in \mathcal{R}^n$ is the feasible decision space. The objective vector $F(x,t) \in \mathcal{R}^m$ consists of m real-valued objective functions, each of which f_i is the ith objective function out of m objectives. $\mathcal{R}^m$ is the objective space. $t \in \{0, 1, 2, ...\}$ represents time or environment variable.

B. Related Definitions

In this section, we provide some useful definitions for this paper. We first specify K uniformly distributed weight vectors,

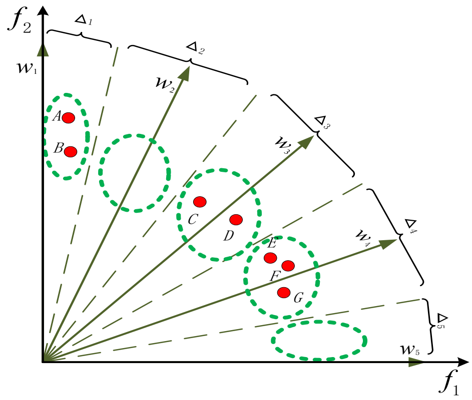


Fig. 1: Illustration of gaps in population distribution, which takes a K = 5 subspaces as an example.

i.e., $\mathcal{W} = \{w_1, ..., w_i, ..., w_K\}$. $w_i \in \mathcal{R}^m$ represents the i-th weight vector in $\mathcal{W}$. The weight vector generation method of [11] is employed. The following definitions are used in the proposed algorithm. Note that Definition 1 has been mentioned in the literature [24].

Definition 1: The vector set $\mathcal{W}$ divides $\mathcal{R}^m$ into K subspaces, i.e., $\Delta_1, ...\Delta_i..., \Delta_K$ in Fig. 1. In particular, a subspace Δ_i, where $i \in 1, ..., K$, is defined as

$$\Delta_i = \{F(x,t) \in \mathcal{R}^m | \langle F(x,t), w_i \rangle \leq \langle F(x,t), w_j \rangle\}, \tag{2}$$

where $j \in 1, \dots, K$ and $\langle F(x,t), w \rangle$ is the perpendicular distance from $F(x,t)$ to the weight vector w, in which $w \in \mathcal{W}$. After the setup of subspaces, each solution of the population is associated with a unique subspace according to its position in $\mathcal{R}^m$ (a random subspace is selected if the solution happens to be at the intersection of multiple subspaces).

Definition 2: For the weight vector set $\mathcal{W}$, the proximity between any two vectors w_i and w_j is defined by

$$\beta_{i,j} = 1 - \frac{||w_i - w_j||}{\sum_{k=1}^{K} ||w_i - w_k||}, \tag{3}$$

where $||w_i - w_j||$ is the Euclidean distance between vectors w_i and w_j in $\mathcal{R}^m$.

III. Related Work

There have been a number of contributions made to several important aspects of DMO, including dynamism classification [25], [26], test problems [27], [28], performance measures [20], [29], and algorithm design [30]. Among these, algorithm designs have received the highest attention in the last 10 years. In what follows, we review different types of DMOEAs available in the literature in great detail.

1) **Prediction-based approach:** In some cases, environmental changes present a regular pattern, e.g., they are somewhat predictable [25]. This predictable dynamism can be best handled by models [18], [31], [32] that predict the new POF and/or POS upon an environmental change. This has inspired the development of various prediction-based strategies [33], [34], [35]. However, existing prediction-based strategies have some open issues, such as poor performance in the early stages of search and strict underlying assumptions. For example, it is found in numerous studies [16], [19] that convergence is slow

during the early evolutionary process due to little history information available. Many prediction-based strategies assume that any two consecutive changes are similar and linear in the decision space [12], [36].

2) **Memory-based approach:** Some DMOPs may exhibit periodic environmental changes over time in the objective spaces and/or the decision space. In this situation, historical POS stored in a memory pool can help the current population to respond quickly to the new environment. Memory-based approaches have been introduced in a few studies, such as memory-based the nondominated sorting genetic algorithm-II (NSGA-II) [37] and memory enhanced evolutionary algorithms [38]. However, memory-based methods may only apply to periodic DMOPs. For non-periodic DMOPs, the memory strategies may be inefficient as historical solutions may not fit any future dynamic environment.
3) **Diversity-based approach**: One of the biggest issues after an environmental change is diversity loss [8], [39], [40]. Therefore, how to maintain good diversity over time has been the focus of a number of DMOEAs. It is reported that diversity can be increased after an environmental change by introducing random individuals or by hyper-mutating some of the existing individuals [8]. Although these approaches are effective for handling the diversity loss, if such individuals are introduced in excess, it may cause an imbalance between population convergence and diversity.
4) **Hybrid approach:** Considering that each of the aforementioned categories has advantages and weaknesses, a series of hybrid strategies [29] have been proposed to solve DMOPs. Peng et al. put forward novel prediction and memory strategies [29], which includes three parts: the previously found elite solutions, exploration operator based on population evolutionary direction and generating guide individuals. Lianga presented a hybrid of memory and prediction strategies [41], which identifies whether an environmental change is similar to historical ones and then applies memory or prediction strategy to respond to change. However, hybrid approaches [1], [29] are not fully explored and there is room for improvement.
5) **Multipopulation approach:** Multipopulation approaches [42], [43] are another class of DMOEAs, where multiple subpopulations can cooperate with and/or compete against each other to evolve in dynamic environments. Goh et al. proposed a competitive-cooperative coevolutionary paradigm [13], which incorporates both competitive and cooperative mechanisms to facilitate adaptive problem decomposition. Despite effectiveness for DMOPs, this kind of algorithm needs a careful setting of the number of subpopulations, which is challenging.

Aside from the aforementioned approaches, there are other methods to improve the convergence rate and enhance diversity. For instance, Jiang et al. proposed a steady-state and generational evolutionary algorithm [20], which responds to changes in a steady-state and generational manner. Wu et al. [44] presented a directed search strategy that consists of two mechanisms for improving the performance of multi-objective evolutionary algorithms in changing environments. Helbig et al. [45] proposed to apply vector evaluated particle swarm optimization for DMOPs.

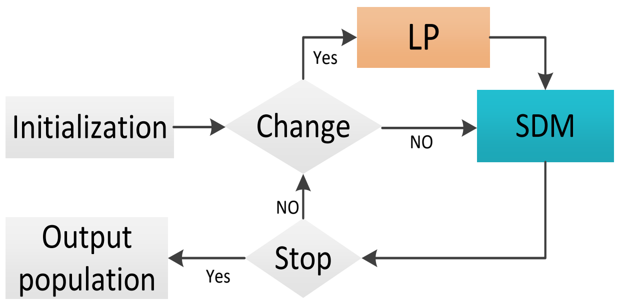


Fig. 2: Flow chart of LPSDM.

IV. Proposed Algorithm

The overall framework of the proposed algorithm is shown in Fig. 2 (pseducode can be found in the supplementary material). The algorithm starts with population and parameter initialization. In each generation, change detection is done by the approach in [1], which assumes an environmental change if the objective values of random detectors are different before and after reevaluation. If an environmental change has been detected, the algorithm executes a layered prediction (LP) strategy. Then, the algorithm proceeds to a subspace-based diversity maintenance (SDM) strategy, which aims to produce high-quality offspring regardless of changes and select for the next generation a population with a good balance between population diversity and convergence. The above procedure will be repeated until the stopping condition is met.We detail the proposed two novel strategies in the following subsections.

A. Layered Prediction Strategy

Algorithm 1 Layered prediction (LP) strategy.

Input: current population, $\mathcal{P}$, current time instant, t; archive, $\mathcal{V}$; the three subpopulations $Sub1$, $Sub2$ and $Sub3$; the centroid of $\mathcal{V}$ at time t, C_t; the centroid of $\mathcal{V}$ at time $t-1$, C_{t-1}.
Output: updated population $\mathcal{P}$.
1: Classify $\mathcal{P}$ into three subpopulations, i.e. $Sub1$, $Sub2$ and $Sub3$, by nondominated sorting;
2: Update each individual of $Sub1$ by Eq. (4) and then re-evaluate $Sub1$;
3: Compute C_{Sub1} and C_{Sub2};
4: Update each individual of $Sub2$ by Eq. (5);
5: Hypermutate $Sub3$ as DNSGA-II-B [8];
6: $\mathcal{P} \leftarrow Sub1 \cup Sub2 \cup Sub3$;
7: $t = t + 1$, $C_{t-1} = C_t$.

When an environmental change is detected, it is important to take immediate actions to handle the new environment. We propose a layered prediction (LP) strategy for this purpose. The basic idea behind LP is to predict roughly the new POS and/or POF after an environmental change. Algorithm 1 provides the implementation of this prediction strategy.

As the name suggests, LP classifies the population (size of N) into different layers and moves each layer in a promising direction, hopefully, towards the new POS/POF. We use the fast nondominated sorting method [10] to obtain a number of nondomination fronts, i.e., L_1, L_2, $\ldots$, by which the

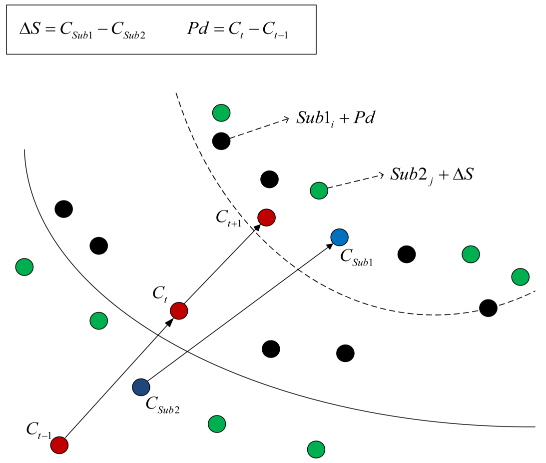


Fig. 3: Illustration of individual update in $Sub1$ and $Sub2$.

population can be classified into three groups, namely $Sub1$, $Sub2$ and $Sub3$. $Sub1$ is exactly L_1, i.e. the nondominated set of the whole population. $Sub2$ selects $0.5(N - |L_1|)$ individuals starting from L_2 onward, and $Sub3$ consists of the remaining unselected individuals. As a result, a three-layer hierarchical prediction model is created. The above population partition approach may result in $Sub2$ and $Sub3$ being empty if the whole population is a nondominated set, particularly for many-objective optimization. In this case, the strategy in [12] can be employed to alleviate the issue.

Each individual x_t in $Sub1$ is predicted to have a better location at time $t+1$

$$x_{t+1} = x_t + (C_t - C_{t-1}), \tag{4}$$

where C_t denotes the centroid of the nondominated set (i.e., archive $\mathcal{V}$ or equivalently the nondominated front L_1) from the population at time t. Eq. (4) assumes that the current environmental change is roughly similar to the last change in terms of the change magnitude and the direction of the moving POS.

Each individual x_t in $Sub2$ is predicted by:

$$x_{t+1} = x_t + C_{Sub1} - C_{Sub2}, \tag{5}$$

where C_{Sub1} is the centroid of the nondominated set of the updated $Sub1$; C_{Sub2} is the centroid of $Sub2$.

$Sub3$ is updated by employing the hypermutation operation from [8], [46], in which each individual x_t in $Sub3$ is regenerated by:

$$x_{t+1} = x_t + \bar{\delta}\Delta_{max}, \tag{6}$$

where $\bar{\delta}$ is the perturbance factor; Δ_{max} is the maximum permissible perturbance that is a fixed value. The more information for $\bar{\delta}$ and Δ_{max} can refer to [46].

Fig. 3 illustrates the prediction of $Sub1$ and $Sub2$. As shown in Fig. 3, C_t and C_{t-1} (orange points) are utilized to predict the individuals of $Sub1$. Then, $Sub1$ is re-evaluated and the centroid of its nondominated set can be used to guide the prediction of $Sub2$.

B. Subspace-based Diversity Maintenance

Diversity maintenance is remarkably important in DMO. Here, we propose a subspace-based diversity maintenance (SDM) strategy to avoid diversity loss in every generation of evolution regardless of environmental changes. SDM decomposes the objective space into K subspaces by Eq. 2 and then identifies gaps in population distribution by checking if there are subspaces that have no population individuals. SDM then conducts reproduction differently for subspaces with and without gaps. A gap filling approach is introduced to generate a portion of offspring for subspaces with gaps, and a probability-based approach using subspace information is employed to generate the remaining portion of offspring population. In what follows, we discuss these two novel approaches in more detail for diversity maintenance in the course of population reproduction.

Algorithm 2 The Subspace-based Diversity Maintenance (SDM) Strategy.

Input: current population, $\mathcal{P}$; archive, $\mathcal{V}$; population size, N.
Output: $\mathcal{P}$

Save the nondominated individuals of $\mathcal{P}$ to $\mathcal{V}$ and set $\mathcal{Q} \leftarrow \emptyset$;
Associate each individual of $\mathcal{P}$ with a unique subspace according to the minimum distance from the individual to the weight vectors;
▷ % Gap filling %
Identify the set $\mathcal{E}$ of subspaces that have no associated individuals;
for all $e \in \mathcal{E}$ **do**
  Generate a new individual y according to Eq. 7 and add y to $\mathcal{Q}$;
end for
Update $\mathcal{V}$ with $\mathcal{Q}$;
▷ % Probability-based reproduction %
while $|\mathcal{Q}| < N)$ **do**
  if $rnd(0,1) \leq 0.5$ **then**
    Choose a random individual x from $\mathcal{V}$;
    Set to $s1$ the subspace that contains x;
  else
    Choose randomly a subspace $s1$;
    Select the best individual from $s1$ as x;
  end if
  Find the first element larger than a random value $rnd(0,1)$ in the $s1$-th row of Φ in Eq. (10) and denote the corresponding index as $s2$;
  Select randomly an individual y from the $s2$-th subspace;
  Generate a new individual p by DE: $p = x + \gamma * (y - z)$, where z is a random individual from $\mathcal{P}$;
  Add p to $\mathcal{Q}$;
end while
Update $\mathcal{V}$ with the last $(N - |\mathcal{E}|)$ individuals of $\mathcal{Q}$;
Obtain a new population $\mathcal{P}$ from $\mathcal{P} \cup \mathcal{Q}$ by the environmental selection method of NSGA-II [10];

1) Gap Filling: This subsection is devoted to removing the gaps the population has in the objective space. Gaps in population distribution can be devastating as they may cause incomplete approximation of the whole POF. This issue is amplified in DMO due to insufficient time for diversity recovery. What is worse, the negative effect of these gaps can be propagated to future environments. For this reason, this paper proposes to address gaps in every generation of evolution regardless of environmental changes.

We identify gaps with the aid of weight vectors which can partition the objective space uniformly into K subspaces (see Fig. 1 for example). We obtain these subspaces by Definition 1. Accordingly, the population is mapped into these subspaces. A gap is found when a subspace has no population individuals, and a new individual is created to hopefully fill the gap. Note that, to help find boundary solutions, we consider an edging subspace (whose weight vector is along an objective axis) as a gap if it has only one individual. Suppose subspace s is a gap, we produce an individual y in this subspace by

$$y = x^b + \bar{F} * (x^r - x^b), \tag{7}$$

where b is the closest gap-free subspace to s. r is another gap-free subspace different from s and b. The determination of r depends on whether s is an intermediate gap or not. If s is an intermediate gap, ideally r is expected to be opposite to b such that w_s lies between w_b and w_r. This can be done by randomly choosing r from the gap-free subspaces such that the angle between $w_b - w_s$ and $w_r - w_s$ is bigger than $\pi - 1/m$ (m is the number of objectives). In the case that s is not an intermediate gap, i.e., it is an edging gap having only one solution, r is randomly chosen such that the angle between $w_b - w_s$ and $w_r - w_s$ is smaller than $1/m$, ensuring b is geometrically between s and r. x^b and x^r are the best individuals from the subspace b and r, respectively. The coefficient $\bar{F}$ is computed by

$$\bar{F} = \begin{cases} \theta & \text{if } s \text{ is not an edging subspace,} \\ -\theta & \text{otherwise,} \end{cases} \tag{8}$$

where

$$\theta = \left|\frac{w_b - w_s}{w_r - w_b}\right|. \tag{9}$$

The gap filling procedure is presented in the pink box of Algorithm 2.

2) Probability-based Reproduction: Subspace information can not only be useful for gap filling but also provide a good way to select mating pools for reproduction. Mating selection requires at least two parent individuals to be determined for mating. While one parent can be randomly chosen from either the current population or a well-maintained external archive, other parents have to be carefully chosen so that their mating can produce a fitter individual than them. In this paper, the second parent is chosen by a cumulative probability matrix that takes into the proximity between any two subspaces, which is defined in Eq. 10:

$$\Phi = \begin{pmatrix} \Phi_{1,1} & \cdots & \Phi_{1,j} & \cdots & \Phi_{1,K} \\ \Phi_{2,1} & \cdots & \Phi_{2,j} & \cdots & \Phi_{2,K} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \Phi_{K,1} & \cdots & \Phi_{K,j} & \cdots & \Phi_{K,K} \end{pmatrix}, \tag{10}$$

where $\Phi_{i,j}$ is a cumulative probability for an individual in the i-th subspace to select a mate from the 1st to j-th subspace. $\Phi_{i,j}$ is calculated by $\Phi_{i,j} = \sum_{r=1}^{j} \beta_{i,r}$ where $\beta_{i,r}$ is defined in Eq. 3. $\Phi_{i,j}$ defines a roulette wheel strategy to select mating parents: the closer a subspace to the i-th subspace, the higher probability an individual chosen from this subspace as a mating candidate.

The probability-based reproduction procedure for filling up an offspring population $\mathcal{Q}$ is presented in the blue box of Algorithm 2. This procedure produces $(N - |\mathcal{E}|)$ offspring individuals, provided that the gap filling strategy has generated $|\mathcal{E}|$ individuals already. To produce a new individual p, we first determine a parent x and its subspace index $s1$. As shown in lines 9–15 of Algorithm 2, x is selected from either the archive $\mathcal{V}$ or the current population $\mathcal{P}$. Choosing an individual as x from $\mathcal{V}$ or the best from a selected subspace helps enhance population convergence. Once knowing the subspace index of x, we can easily determine the subspace index $s2$ of another mating parent y by Eq. 10 (line 16 of Algorithm 2). For simplicity, y is randomly chosen from the current population individuals of subspace index $s2$. The new individual p can be created by the DE operator [23] with the mating parents x and y plus another random population individual z (line 18 of Algorithm 2). Then, p is added to $\mathcal{Q}$.

When an offspring population $\mathcal{Q}$ is created, it is used for archive update (i.e., removing dominated members from $\mathcal{V}$ and add nondominated ones from $\mathcal{Q}$ to $\mathcal{V}$). $\mathcal{Q}$ and $\mathcal{P}$ are also combined to form a new population $\mathcal{P}$ for the next generation of evolution, and this is done using the environmental selection approach of NSGA-II [10] (line 22 of Algorithm 2).

C. Computational Complexity of One Generation of LPSDM

In the loop (Fig. 2) of each generation, computational resources are mainly consumed by the layered prediction strategy (LP) and subspace-based diversity maintenance (SD-M, including gap filling and probability-based reproduction). In the LP strategy, the most computational intensive part is population partition by nondominated sorting, which has the time complexity of $O(N^2)$ (N is population size), and individual re-initialization for different subpopulations is of linear complexity, i.e., $O(N)$. In SDM, gap filling first needs to associate the population with K subspaces in the objective space, which takes $O(KN)$ computations. Then, it identifies and fills gaps over K subspaces in the objective space, requiring $O(K)$ computations. The mating selection requires the calculation of a cumulative probability matrix in Eq. 10, which has the time complexity of $O(K^2)$. The environmental selection procedure (line 22 of Algorithm 2) spends $O(N^2)$ computations on elitist preservation. Since K is smaller than N, the overall computational complexity of LPSDM for one generational cycle is therefore $O(N^2)$.

V. Experimental Settings

A. Benchmark Problems

Test problems with dynamic changes play an important role in assessing and analyzing the performance of DMOEAs. We use multiple different benchmark test suites, including FDA [25], dMOP [13] and JY [27], to systematically study the performance of the proposed algorithm. FDA is one of the earliest test suitess for performance evaluation in DMO. The dMOP test suite is an upgraded version of the FDA test suite. The JY test suite is recently proposed to cover a number of rarely considered characteristics, including mixed POFs, nonmonotonic and time-varying relationships between variables, and dynamic problem types.

B. Performance Measures

Many performance measures have been introduced to assess the performance of algorithms on convergence and diversity in dynamic environments, and among these, the inverted generational distance (IGD) [19], hypervolume difference (HVD) [20], generational distance (GD) [29] and Schott's spacing

TABLE I: Parameter settings

Number of decision variables, n	10 for all test problems
Population size	100 for all test problems
Probability that parents are selected from the archive	0.5
Decomposition method	Tchebycheff [11]
Differential evolution	$CR = 1.0$ and $\gamma = 0.2$ like [33]
Polynomial mutation	$P_m = 0.05$
Crossover probability	$P_c = 0.8$
Dynamic setting	Severity of changes, $n_t = 5, 10, 30$ Frequency of change, $\tau_t = 10, 20, 30$
Number of changes	$num = 50$
Number of generations	$\tau_t * num$

(SP)[20] are popular ones. These measures have been redefined for dynamic multi-objective optimization [18], [34], [36] and are respectively denoted as MIGD, MHVD, MGD and MSP. Note that MIGD and MHVD can measure the convergence and diversity of DMOEAs in a compact manner whereas MGD (MSP) can only measure convergence (diversity). These measures are detailed in the supplementary material.

C. Compared Algorithms

In this empirical study, we demonstrate our approach by comparing it with five popular DMOEAs of different types. They are the MOEA based on decomposition (MOEA/D) [11], population prediction strategy (PPS) [19], dynamic non-dominated sorting genetic algorithm II (DNSGA-II) [8], a mixture-of-experts prediction (MoE) [18] and steady-state and generational EA (SGEA) [20]. These algorithms have different change reaction mechanisms, and they are briefed in the supplementary material.

D. Parameter Settings

The selected DMOEAs have algorithm-specific parameters, and we used their default settings as in their original papers. Like [1], [18], ten solutions are randomly selected and re-evaluated for detecting changes in environment. K is set as $K = 20$ in the proposed algorithm based on sensitivity analysis which is detailed later. Some key parameters summarized in Table I are briefly explained as follows:

1) The number of decision variables is set as $n = 10$ for all the test problems with a population size of $N = 100$. Noting that $N = 105$ when $m = 3$ for MOEA/D and MoE.
2) The proposed algorithm uses for reproduction differential evolution (DE) [15] whose crossover rate is set as $CR = 1.0$ and scaling factor is assigned as $\gamma = 0.2$. MOEA/D, DNSGA-II and SGEA use the simulation binary crossover and polynomial mutation (SBX) [15], [26]. The crossover probability is $p_c = 0.8$ and the mutation probability is $p_m = 0.05$.
3) Dynamic environments are simulated with $num = 50$ environmental changes. The severity of change (n_t) is set as 5, 10, and 30. The frequency of change (τ_t) is set as 10, 20, and 30.
4) The total number of generations is set as $\tau_t * num$ for all the algorithms on all the test problems. Each algorithm ran independently 20 times on each problem.

VI. Experimental Results And Discussion

The proposed algorithm is compared with five state-of-the-art algorithms, and the resulting MIGD and MHVD values are presented in Tables II and III, respectively, where the best values obtained by one of the five algorithms are highlighted in bold face. The MGD and MSP values of the five algorithms can be found in Tables VI and VII of the supplementary material. To indicate the significance of differences between algorithms, the Wilcoxon rank-sum test [47] is performed at the 5% significance level. Besides, all algorithms from tables are implemented in C++. The following subsections present key comparison results and detailed performance analysis on a number of selected test problems. Additional results (e.g. impact of severity of change) and more discussion (e.g. results on FDA3, FDA3, dMOP) are reported in the supplementary file.

A. Results on FDA1, FDA4, JY1-JY9

The average and standard deviation values for different measures show that LPSDM obtains the best results on the majority of instances, implying that it offers better approximations to the changing POS/POF than the other compared algorithms in most cases as shown in Tables II, III and Tables VI, VII of the supplementary material. The following presents the results for each test problem in detail.

FDA1: Tables II, III show that LPSDM obtains the best results on FDA1. This may be because LPSDM maintains the best diversity and convergence over time than the other compared algorithms, as shown in Tables VI and VII of the supplementary material. Note that MoE and PPS obtain large MIGD and MHVD values, showing their poor convergence (see Table VI of the supplementary material) and distribution (indicated by large MSP values in Table VII of the supplementary material) for this problem.

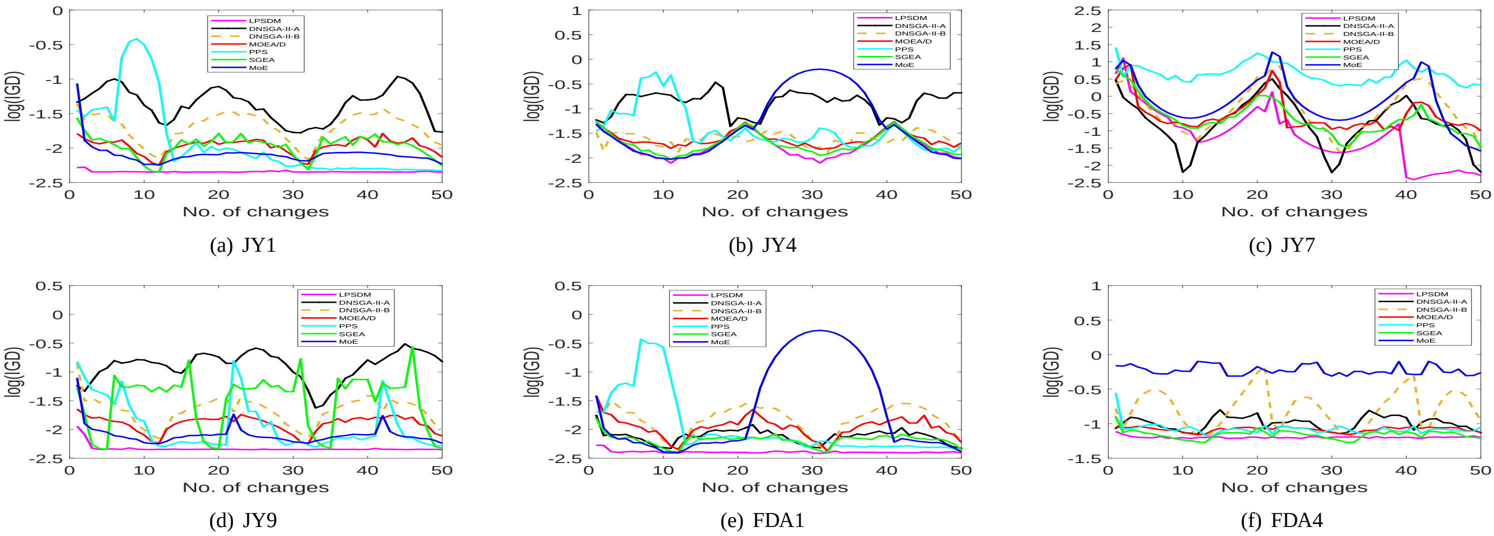
FDA4: Although the distribution obtained by LPSDM is less uniform than SGEA for the slow-changing environment as shown in Table VII of the supplementary material, the convergence of LPSDM significantly outperforms the other compared algorithms, i.e., its MGD values are the smallest. Consequently, LPSDM obtains the best overall performance on FDA4 (see Tables II, III). Note that the DNSGA-II-A, DNSGA-II-B and PPS do not perform well in MIGD and MHVD, and their large MGD values (see Table VI of the supplementary material) imply poor convergence for this problem.

JY1: JY1 mainly tests the convergence speed and reactivity of the algorithm. The performance of LPSDM is the best, as shown in Tables II and III. This may be because both the convergence or distribution uniformity of LPSDM have significantly small variations. Besides, the performance of LPSDM is more steady than the other compared algorithms across the whole evolution process, as shown in Fig. 4(a).

TABLE II: Mean and standard deviation values of MIGD obtained by seven algorithms

Problems	(n_t, τ_t)	LPSDM	DNSGA-II-A	DNSGA-II-B	SGEA	MOEA/D	PPS	MoE
FDA1	(10, 10)	**4.14e-3(5.90e-5)**	1.83e-2(5.44e-4)‡	1.03e-1(1.75e-2)‡	1.29e-2(9.92e-4)‡	3.19e-2(3.03e-3)‡	8.24e-2(6.49e-2)‡	1.13e-1(7.68e-4)‡
	(10, 20)	**4.05e-3(1.25e-5)**	7.76e-3(1.75e-4)‡	1.75e-2(7.32e-4)‡	6.63e-3(3.33e-4)‡	1.19e-2(4.89e-4)‡	3.91e-2(1.70e-2)‡	1.07e-1(3.20e-4)‡
	(10, 30)	**4.00e-3(9.57e-6)**	6.11e-3(4.66e-5)‡	9.00e-3(4.33e-4)‡	5.25e-3(1.46e-4)‡	8.25e-3(1.55e-4)‡	7.71e-3(7.05e-4)‡	1.05e-1(3.86e-5)‡
FDA4	(10, 10)	**6.27e-2(8.16e-5)**	1.37e-1(2.12e-3)‡	4.70e-1(2.46e-2)‡	9.97e-2(4.46e-3)‡	1.10e-1(1.89e-3)‡	1.03e-1(3.77e-3)‡	7.51e-1(1.17e-3)‡
	(10, 20)	**6.35e-2(5.00e-5)**	1.03e-1(2.30e-3)‡	1.96e-1(8.16e-3)‡	7.04e-2(1.82e-3)‡	8.27e-2(2.98e-4)‡	8.59e-2(2.26e-3)‡	7.46e-1(9.68e-5)‡
	(10, 30)	**6.20e-2(1.89e-4)**	8.93e-2(2.30e-3)‡	1.03e-1(2.43e-3)‡	6.24e-2(1.04e-3)†	7.58e-2(1.50e-4)‡	8.04e-2(1.10e-3)‡	7.46e-1(4.03e-5)‡
JY1	(10, 10)	**4.62e-3(7.50e-5)**	2.05e-1(4.12e-2)‡	1.10e-1(1.20e-2)‡	2.41e-2(1.56e-3)‡	2.27e-2(8.16e-4)‡	3.50e-2(6.00e-3)‡	2.43e-2(4.24e-3)‡
	(10, 20)	**4.54e-3(1.73e-5)**	4.75e-2(8.25e-3)‡	2.27e-2(8.40e-4)‡	1.13e-2(5.05e-4)‡	1.09e-2(1.89e-4)‡	3.85e-2(2.57e-2)‡	9.36e-3(6.17e-4)‡
	(10, 30)	**4.52e-3(1.25e-5)**	2.65e-2(7.64e-3)‡	1.18e-2(2.46e-4)‡	7.89e-3(4.01e-4)‡	7.95e-3(1.21e-4)‡	8.36e-3(2.13e-3)‡	7.01e-3(2.43e-4)‡
JY2	(10, 10)	**5.02e-2(5.00e-5)**	5.71e-2(6.33e-4)‡	9.94e-2(6.07e-3)‡	5.66e-2(1.21e-3)‡	5.64e-2(6.35e-4)‡	9.41e-2(5.93e-2)‡	1.68e-1(1.63e-3)‡
	(10, 20)	**5.02e-2(1.04e-5)**	5.22e-2(1.80e-4)‡	7.21e-2(7.11e-4)‡	5.14e-2(4.93e-4)‡	5.18e-2(3.69e-4)‡	6.21e-2(1.95e-2)‡	1.64e-1(1.46e-4)‡
	(10, 30)	5.01e-2(5.00e-5)	5.17e-2(7.58e-5)‡	6.18e-2(2.41e-4)‡	**5.00e-2(1.17e-4)**	5.09e-2(0.00e+0)‡	5.60e-2(7.10e-3)‡	1.63e-1(7.22e-5)‡
JY3	(10, 10)	3.15e-1(0.00e+0)	**2.92e-1(1.26e-2)**	3.15e-1(1.39e-3)‡	3.47e-1(1.43e-2)‡	3.29e-1(1.86e-2)‡	3.39e-1(1.25e-2)‡	3.16e-1(6.93e-3)†
	(10, 20)	3.11e-1(4.12e-3)	**2.79e-1(4.60e-3)**	3.14e-1(9.68e-4)†	3.40e-1(9.95e-3)‡	3.21e-1(4.00e-3)‡	3.42e-1(7.55e-3)‡	3.12e-1(1.88e-3)†
	(10, 30)	3.10e-1(1.41e-3)	**2.81e-1(1.28e-2)**	3.13e-1(6.27e-4)†	3.38e-1(9.23e-3)‡	3.18e-1(7.89e-3)‡	3.14e-1(5.00e-4)‡	3.13e-1(2.77e-3)†
JY4	(10, 10)	**1.98e-2(1.25e-4)**	1.94e-1(1.51e-2)‡	1.30e-1(8.84e-3)‡	2.83e-2(9.20e-4)‡	3.42e-2(5.67e-4)‡	1.00e-1(3.09e-2)‡	1.51e-1(1.78e-3)‡
	(10, 20)	**2.01e-2(1.00e-4)**	1.50e-1(6.60e-3)‡	2.20e-2(5.69e-4)‡	2.19e-2(4.23e-4)†	2.54e-2(2.63e-4)‡	8.04e-2(2.20e-2)‡	1.36e-1(2.86e-4)‡
	(10, 30)	**2.01e-2(5.77e-5)**	1.28e-1(7.63e-3)‡	2.53e-2(3.07e-4)‡	2.08e-2(3.31e-4)†	2.31e-2(2.87e-4)‡	5.96e-2(3.38e-2)‡	1.35e-1(8.84e-5)‡
JY5	(10, 10)	**4.20e-3(5.00e-6)**	3.06e-2(3.62e-4)‡	6.37e-3(3.24e-4)‡	5.04e-3(4.17e-4)‡	9.97e-3(1.84e-3)‡	7.24e-3(2.15e-4)‡	9.89e-3(4.54e-4)‡
	(10, 20)	**4.21e-3(5.77e-6)**	3.02e-2(3.69e-4)‡	5.06e-3(4.88e-5)‡	4.36e-3(2.44e-4)†	7.58e-3(2.59e-4)‡	5.13e-3(1.73e-5)‡	7.59e-3(1.99e-4)‡
	(10, 30)	4.20e-3(1.00e-5)	3.02e-2(5.27e-4)‡	4.90e-3(3.28e-5)‡	**4.17e-3(1.47e-5)**	6.53e-3(3.26e-5)‡	4.87e-3(1.61e-4)‡	7.12e-3(1.70e-4)‡
JY6	(10, 10)	**3.72e-2(2.72e-3)**	1.70e+0(1.24e-1)‡	3.21e+0(1.81e-1)‡	5.34e-1(4.15e-2)‡	4.15e-1(3.81e-2)‡	2.67e+0(4.69e-2)‡	1.64e+0(1.35e-1)‡
	(10, 20)	**1.80e-2(1.50e-2)**	7.35e-1(5.02e-2)‡	7.26e-1(4.20e-2)‡	1.80e-1(6.50e-3)‡	1.93e-1(3.12e-2)‡	2.12e+0(8.09e-2)‡	9.45e-1(1.01e-1)‡
	(10, 30)	**1.93e-2(1.32e-2)**	4.38e-1(1.26e-1)‡	2.74e-1(1.53e-2)‡	9.26e-2(7.75e-3)‡	8.43e-2(1.90e-2)‡	1.87e+0(7.22e-2)‡	5.86e-1(4.90e-2)‡
JY7	(10, 10)	**5.63e-1(1.47e-1)**	2.07e+0(3.11e-1)‡	7.76e+0(1.01e+0)‡	1.02e+0(3.27e-1)‡	1.84e+0(1.64e-1)‡	9.36e+0(1.22e+0)‡	2.69e+0(5.09e-1)‡
	(10, 20)	**4.30e-1(3.52e-1)**	5.22e-1(2.66e-1)‡	1.17e+0(4.15e-1)‡	4.78e-1(2.04e-1)‡	8.67e-1(4.55e-1)‡	5.88e+0(1.28e+0)‡	2.27e+0(8.53e-1)‡
	(10, 30)	7.97e-1(2.03e-1)	**1.16e-1(1.30e-2)**	1.04e+0(1.33e-1)‡	3.69e-1(2.49e-1)	1.00e+0(5.36e-2)‡	1.72e+0(2.36e+0)‡	2.20e+0(6.58e-1)‡
JY8	(10, 10)	1.08e-2(5.67e-4)	8.59e-2(4.74e-2)‡	1.20e-2(1.56e-3)‡	3.69e+0(0.00e+0)‡	3.04e-2(7.16e-3)‡	**9.41e-3(3.65e-4)**	2.24e-2(1.77e-3)‡
	(10, 20)	1.11e-2(5.67e-4)	1.48e-1(7.25e-2)‡	1.15e-2(5.15e-4)†	3.69e+0(0.00e+0)‡	2.02e-2(2.86e-3)‡	**8.70e-3(2.99e-4)**	1.75e-2(9.14e-4)‡
	(10, 30)	1.16e-2(6.00e-4)	1.38e-1(5.98e-2)‡	2.29e-2(2.03e-4)‡	3.69e+0(0.00e+0)‡	1.66e-2(7.22e-4)‡	**8.44e-3(3.77e-5)**	1.64e-2(7.65e-4)‡
JY9	(10, 10)	**6.45e-3(1.71e-3)**	4.71e-1(2.10e-1)‡	7.87e-2(4.79e-3)‡	3.94e-1(5.02e-2)‡	3.18e-2(1.02e-2)‡	2.61e-1(2.26e-1)‡	2.32e-2(4.19e-3)‡
	(10, 20)	**4.71e-3(6.18e-5)**	1.52e-1(5.62e-2)‡	2.57e-2(1.08e-3)‡	4.87e-2(3.19e-3)‡	1.31e-2(4.50e-4)‡	2.12e-2(2.77e-3)‡	9.26e-3(6.48e-4)‡
	(10, 30)	**4.61e-3(3.55e-5)**	7.22e-2(1.99e-2)‡	1.37e-2(6.25e-4)‡	3.44e-2(6.84e-4)‡	9.63e-3(7.05e-4)‡	9.97e-3(8.05e-4)‡	7.03e-3(1.67e-4)‡

‡ and † indicate LPSDM performs significantly better than and equivalently to the corresponding algorithm, respectively.



(a) JY1 (b) JY4 (c) JY7

(d) JY9 (e) FDA1 (f) FDA4

Fig. 4: Evolution curves of average IGD values for problems with n_t =10 and τ_t=20.

In conclusion, the proposed algorithm's excellent response to dynamic changes compared with the other algorithms.

JY2: JY2's objective functions oscillate among several optimization modes, and its POS changes over time. Tables II and III show LPSDM is significantly better than the other compared algorithms for $\tau_t \leq 20$ and it is slightly worse than SGEA when $\tau_t = 30$. As observed from small MGD values (see Table VI of supplementary material), the MGD values of LPSDM, SGEA and MOEA/D are not significantly different from each other on JY2 and outperform the other algorithms. Additionally, the distribution of LPSDM substantially outperforms the other algorithms (indicated by MSP values in Table VII of the supplementary material). The performance of LPSDM is very stable regardless of change frequency on

TABLE III: Mean and standard deviation values of MHVD obtained by seven algorithms

Problems	(n_t, τ_t)	LPSDM	DNSGA-II-A	DNSGA-II-B	SGEA	MOEA/D	PPS	MoE
FDA1	(10, 10)	**6.49e-3(2.68e-4)**	4.09e-2(1.30e-3)‡	1.89e-1(1.67e-2)‡	2.89e-2(4.68e-4)‡	6.34e-2(2.84e-3)‡	1.60e-1(1.17e-1)‡	2.31e-1(2.38e-3)‡
	(10, 20)	**6.14e-3(4.12e-5)**	1.64e-2(5.17e-4)‡	4.05e-2(1.70e-3)‡	1.51e-2(5.72e-4)‡	2.86e-2(6.92e-4)‡	9.26e-2(4.27e-2)‡	2.13e-1(1.71e-3)‡
	(10, 30)	**6.00e-3(3.77e-5)**	1.18e-2(2.17e-4)‡	2.04e-2(6.65e-4)‡	1.10e-2(4.32e-4)‡	1.80e-2(5.19e-4)‡	1.60e-2(1.43e-3)‡	2.08e-1(4.34e-4)‡
FDA4	(10, 10)	**1.28e-1(1.63e-3)**	1.16e+0(1.42e-1)‡	2.66e+0(3.72e-1)‡	3.65e-1(3.83e-2)‡	5.04e-1(1.33e-1)‡	5.05e-1(4.71e-2)‡	1.15e+0(7.30e-2)‡
	(10, 20)	**1.31e-1(4.34e-3)**	5.69e-1(1.29e-1)‡	6.32e-1(3.49e-2)‡	2.12e-1(1.78e-2)‡	1.63e-1(5.18e-3)‡	2.71e-1(1.04e-2)‡	9.47e-1(2.62e-2)‡
	(10, 30)	**1.24e-1(1.91e-3)**	4.07e-1(6.21e-2)‡	3.09e-1(1.79e-2)‡	1.79e-1(9.14e-3)‡	1.35e-1(1.16e-2)	2.51e-1(2.00e-2)‡	9.08e-1(1.36e-2)‡
JY1	(10, 10)	**3.81e-3(9.05e-5)**	4.84e-1(1.32e-1)‡	2.02e-1(1.22e-2)‡	2.52e-2(1.50e-3)‡	2.15e-2(2.13e-3)‡	6.85e-2(1.37e-2)‡	6.77e-2(2.05e-2)‡
	(10, 20)	**3.67e-3(6.94e-5)**	6.39e-2(1.78e-2)‡	2.77e-2(8.60e-4)‡	1.08e-2(2.14e-4)‡	9.54e-3(1.37e-3)‡	9.16e-2(7.59e-2)‡	1.39e-2(3.24e-3)‡
	(10, 30)	**3.55e-3(5.67e-5)**	3.11e-2(1.16e-2)‡	1.18e-2(3.05e-4)‡	7.82e-3(6.12e-4)‡	6.95e-3(6.60e-4)‡	1.12e-2(3.94e-3)‡	7.09e-3(6.82e-4)‡
JY2	(10, 10)	**6.48e-3(1.35e-4)**	5.06e-2(3.12e-3)‡	2.24e-1(2.50e-2)‡	2.95e-2(2.24e-3)‡	3.08e-2(1.55e-3)‡	1.65e-1(2.18e-1)‡	3.07e-1(5.61e-3)‡
	(10, 20)	**6.13e-3(3.46e-5)**	2.46e-2(3.65e-4)‡	3.31e-2(7.12e-4)‡	1.36e-2(7.96e-4)‡	1.28e-2(1.13e-3)‡	4.16e-2(4.69e-2)‡	2.85e-1(5.94e-4)‡
	(10, 30)	**6.04e-3(3.69e-5)**	2.23e-2(3.44e-4)‡	1.65e-2(2.92e-4)‡	9.33e-3(2.06e-4)‡	8.78e-3(3.77e-5)‡	2.45e-2(1.55e-2)‡	2.81e-1(4.88e-4)‡
JY3	(10, 10)	2.91e-1(1.31e-1)	**9.52e-2(1.32e-1)**	3.86e-1(5.81e-3)‡	4.11e-1(9.52e-3)‡	3.84e-1(1.41e-3)‡	5.57e-1(6.19e-2)‡	2.40e-1(1.96e-1)
	(10, 20)	1.13e-1(1.61e-1)	**2.80e-2(6.35e-3)**	3.61e-1(7.99e-4)‡	3.67e-1(4.76e-2)‡	3.58e-1(2.82e-2)‡	4.26e-1(4.17e-2)‡	9.34e-2(1.12e-1)
	(10, 30)	**1.62e-2(1.52e-2)**	7.93e-2(1.36e-1)‡	3.56e-1(1.88e-3)‡	3.27e-1(1.22e-1)‡	3.32e-1(6.79e-2)‡	3.73e-1(7.39e-3)‡	2.00e-1(1.70e-1)‡
JY4	(10, 10)	**2.12e-2(1.50e-4)**	1.09e+1(1.66e+0)‡	2.72e-1(2.62e-2)‡	4.78e-2(1.64e-3)‡	5.84e-2(1.69e-3)‡	2.55e-1(1.01e-1)‡	3.19e-1(4.80e-3)‡
	(10, 20)	**2.07e-2(2.50e-4)**	1.64e+1(7.63e-1)‡	4.94e-2(7.28e-4)‡	2.85e-2(5.76e-4)‡	3.16e-2(6.55e-4)‡	1.89e-1(6.40e-2)‡	2.71e-1(8.06e-4)‡
	(10, 30)	**2.01e-2(1.50e-4)**	1.84e+1(4.39e-1)‡	2.96e-2(8.15e-4)‡	2.40e-2(7.31e-4)‡	2.57e-2(8.18e-4)‡	1.39e-1(9.54e-2)‡	2.62e-1(4.50e-4)‡
JY5	(10, 10)	**4.62e-3(5.00e-6)**	4.38e-2(7.04e-4)‡	8.77e-3(4.49e-4)‡	6.40e-3(7.78e-4)‡	8.65e-3(7.46e-4)‡	1.23e-2(4.61e-4)‡	1.92e-2(2.54e-3)‡
	(10, 20)	**4.69e-3(5.00e-6)**	4.24e-2(8.86e-4)‡	5.76e-3(1.48e-4)‡	5.07e-3(3.67e-4)†	6.23e-3(1.12e-4)‡	7.77e-3(3.20e-5)‡	1.10e-2(7.89e-4)‡
	(10, 30)	4.73e-3(1.00e-5)	4.22e-2(5.02e-4)‡	5.30e-3(6.23e-5)‡	**4.72e-3(5.31e-5)**	5.28e-3(2.06e-5)‡	7.04e-3(3.32e-4)‡	8.59e-3(4.16e-4)‡
JY6	(10, 10)	**8.37e-2(1.36e-2)**	1.20e+1(1.46e+0)‡	3.19e+1(3.67e+0)‡	1.78e+0(2.30e-1)‡	1.26e+0(2.13e-1)‡	2.59e+1(1.52e+0)‡	1.42e+1(1.08e+0)‡
	(10, 20)	**4.10e-2(4.86e-2)**	3.97e+0(7.10e-1)‡	3.23e+0(3.65e-1)‡	3.44e-1(2.32e-2)‡	2.80e-1(4.86e-2)‡	1.79e+1(8.99e-1)‡	6.95e+0(8.81e-1)‡
	(10, 30)	**3.90e-2(3.29e-2)**	1.97e+0(8.95e-1)‡	6.58e-1(6.33e-2)‡	1.35e-1(1.67e-2)‡	1.28e-1(9.60e-3)‡	1.46e+1(8.96e-1)‡	3.72e+0(4.86e-1)‡
JY7	(10, 10)	**4.84e+0(8.40e-1)**	3.07e+1(8.52e+0)‡	2.63e+2(5.45e+1)‡	7.74e+0(2.75e+0)‡	1.34e+1(4.84e+0)‡	5.02e+2(8.07e+1)‡	8.86e+1(2.88e+1)‡
	(10, 20)	6.49e+0(8.52e+0)	5.04e+0(4.55e+0)	1.02e+1(6.70e+0)‡	**2.89e+0(2.03e+0)**	1.48e+1(1.36e+1)‡	2.41e+2(6.96e+1)‡	5.68e+1(2.99e+1)‡
	(10, 30)	9.57e+0(4.70e+0)	**4.35e-1(5.72e-2)**	1.86e+0(1.79e+0)	3.51e+0(3.94e+0)	1.26e+1(5.67e+0)‡	6.16e+1(8.23e+1)‡	4.87e+1(2.50e+1)‡
JY8	(10, 10)	**1.14e-1(5.00e-4)**	3.82e-1(1.96e-1)‡	1.37e-1(3.44e-3)‡	1.63e+0(4.43e-2)‡	1.19e-1(2.06e-3)†	1.33e-1(5.05e-3)‡	1.65e-1(2.69e-2)‡
	(10, 20)	**1.13e-1(0.00e+0)**	2.86e-1(1.58e-1)‡	1.22e-1(1.74e-3)†	1.59e+0(2.66e-2)‡	1.14e-1(3.00e-3)†	1.23e-1(1.00e-3)‡	1.20e-1(2.26e-3)‡
	(10, 30)	**1.14e-1(5.00e-4)**	3.12e-1(2.08e-1)‡	1.19e-1(1.19e-3)†	1.63e+0(4.59e-2)‡	1.16e-1(2.87e-3)†	1.20e-1(5.00e-4)‡	1.20e-1(6.64e-3)‡
JY9	(10, 10)	**8.81e-3(4.94e-3)**	2.31e+0(2.22e+0)‡	1.38e-1(1.45e-2)‡	2.82e+0(6.75e-1)‡	2.98e-2(4.76e-3)‡	1.60e+0(1.67e+0)‡	6.92e-2(2.44e-2)‡
	(10, 20)	**3.97e-3(1.17e-4)**	3.55e-1(1.70e-1)‡	3.28e-2(1.43e-3)‡	4.06e-2(7.42e-3)‡	1.16e-2(1.30e-3)‡	3.92e-2(8.74e-3)‡	1.41e-2(3.48e-3)‡
	(10, 30)	**3.80e-3(1.12e-4)**	1.16e-1(4.02e-2)‡	1.46e-2(6.01e-4)‡	1.40e-2(1.15e-3)‡	9.10e-3(1.76e-3)‡	1.50e-2(1.63e-3)‡	7.04e-3(2.42e-4)‡

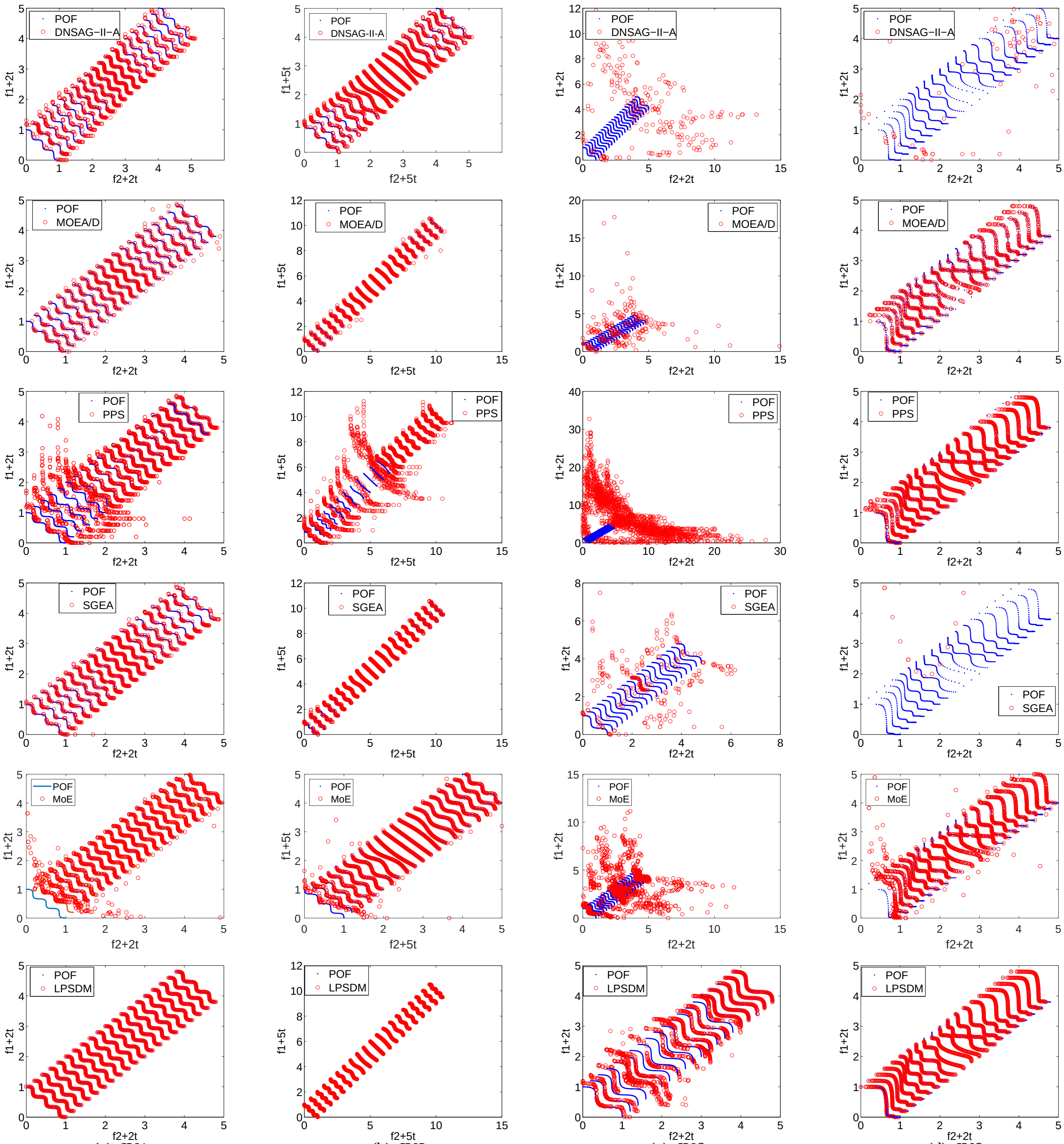
‡ and † indicate LPSDM performs significantly better than and equivalently to the corresponding algorithm, respectively.

MIGD, MHVD, MGD and MSP. To sum up, LPSDM can better overcome the problems' oscillation compared with the other algorithms.

JY3: In JY3, the density of solutions changes over time, and the dependency between variables is non-monotonic and becomes increasingly complicated as time precedes. Table VI of the supplementary material shows that LPSDM has the best performance in convergence among all the algorithms under $\tau_t = 20$ and $\tau_t = 30$. LPSDM is relatively worse than DNSGA-II-A, DNSGA-II-B, SGEA and MOEA/D on MGD and MSP when $\tau_t = 10$. For Table VII of the supplementary material, LPSDM shows convergence improvement when $\tau_t = 20$, but its distribution performance remains poor. Tables II and III show that DNSGA-II-A has the best results than LPSDM when $\tau_t = 10$ and $\tau_t = 20$, indicating that although DNSGA-II-A is not best on MGD and MSP, it can still obtain the best MIGD and MHVD values. The inconsistency between these four measures obtained by DNSGA-II-A and LPSDM suggests that, on the one hand, the approximation obtained by LPSDM covers poorly the whole POF for JY3. On the other hand, DNSGA-II-A may not be well converged, which affects the calculation of MHVD.

JY4: JY4 has a time-changing number of disconnected POF segments, which poses challenges for algorithms to search for the whole POF. Although the MSP measure shows that the solution distribution of LPSDM is not as good as that of MOEA/D, LPSDM can still be very well on JY4, as indicated by the smallest MIGD (Table II) and MHVD (Table III) values. This demonstrates that LPSDM is capable of dealing with disconnected POF. Because LPSDM has a strong convergence ability and it overcomes the ineffective solutions distribution no matter how changes the number of disconnected POF segments over time [27].

JY5: JY5 has a simple POF, which changes from convex geometry to concave geometry and vice versa. For $\tau_t = 10$, LPSDM obtains better MSP and MGD values than the other algorithms (see Tables VI, VII of the supplementary material). However, when $\tau_t = 20$ or $\tau_t = 30$, the low MGD values suggest MOEA/D converges better than the other algorithms, and small MSP suggests SGEA obtains well-diversified solutions across the POF. Despite good MSP values, SGEA is not comparable with LPSDM in terms of the overall performance since the MHVD or MIGD values of LPSDM are the lowest when $\tau_t = 20$ and $\tau_t = 10$. This is because SGEA's convergence is not good (indicated by poor MGD values when $\tau_t = 20$ and $\tau_t = 30$). Though MOEA/D has good convergence for JY5, its solution distribution across the POF is unsatisfactory (see the poor SP values of MOEA/D). LPSDM significantly outperforms the other algorithms by a clear margin in terms of MIGD and MHVD. With an increase in the frequency of change, LPSDM does not improve prominently but still can find the best solutions for JY5, as indicated by its MIGD and



(a) JY1 (b) JY2 (c) JY6 (d) JY8

Fig. 5: Obtained POFs for five problems with $n_t = 10$ and $\tau_t = 10$.

MHVD. Notably, the performance of LPSDM is statistically equal to SGEA with the best performance when $\tau_t = 30$. Overall, LPSDM can overcome the influence on the POF changes from convex geometry to concave geometry because it can balance the population distribution and convergence.

JY6: JY6 is a multimodal problem, where not only the number of local optima changes over time but also the POS is dynamically shifted. From Tables II, III, we see the LPSDM performs significantly better than the other algorithms on JY6 no matter what the frequency of change is, indicating that LPSDM is very effective for this kind of DMOPs. This may be because the distribution and convergence of LPSDM (see Tables VI, VII of the supplementary material) significantly outperform the other algorithms resulting in owning the good overall performance of LPSDM on JY6.

JY7: JY7 is similar to JY6 but more difficult than JY6. It takes into account the dynamic POS shift and overall POF shape and adds drastic changes on multi-modality, which increases the difficulty for algorithms to jump out of local optima. For this problem, LPSDM does not defeat the other competitors in all cases. Especially, LPSDM has very competitive convergence performance (i.e., relatively small MGD) for high frequencies of change, but its convergence degrades significantly for a lower frequency of change, leading to the worst MGD value (see Table VI of the supplementary material). Nevertheless, the diversity of solutions obtained by LPSDM is significantly better than the other algorithms for the three frequencies of change. Tables II and III show that the MIGD and MHVD of LPSDM are not good. This suggests an imbalance exists between convergence and diversity in LPSDM. Therefore, the convergence of LPSDM should be enhanced to solve JY7 well.

JY8: JY8 features the time-varying overall POF shape, in which the geometry and the number of mixed segments of the POF change over time. The values of LPSDM on MGD and MSP are smaller than the other algorithms, implying that LPSDM has good convergence and presents diversified solutions. However, the other two measures MIGD and MHVD, obtained by different algorithms, are inconsistent with each other. For example, the MIGD obtained by LPSDM is inferior to PPS, but the MHVD is the smallest. This is likely due to the fact that MHVD is sensitive to knee solutions in the population [48], suggesting LPSDM seems effective for identifying knee points. Besides, LPSDM is also shown, by the smallest MSP values, to have very good spacing between the obtained solutions. Its MIGD values are not the best but clearly much better DNSGA-II-A, DNSGA-II-B, SGEA, MoE and MOEA/D. MSP and MIGD collectively indicate that the coverage of solutions by LPSDM may need to be improved. PPS also has a good overall performance for this problem.

JY9: JY9 is a problem that cyclically switches from type I (i.e., the POS changes over time while the POF remains stationary) to type II (i.e., both the POF and POS change over time), then to type III (i.e., the POF changes over time while the POS remains stationary), which is a novel DMOP in DMO. From Tables II, III and Tables VI, VII of the supplementary material, we find that LPSDM has consistently shown the best performance regardless of different τ_t. For all measures, LPSDM is significantly better than the other algorithms since the gap filling approach of SDM can improve the diversity loss adaptively due to time-changing problem types.

Besides, it can also be observed from the results of the four used measures that the frequency of change has a significant effect on algorithms' performance, and the effect decreases as environmental changes become less frequent. Nevertheless, whatever the frequency of change is, LPSDM is the best for most of the test instances. Alongside, Table VIII of the supplementary material presents information regarding the p-value of the Wilcoxon rank-sum test [49] for the considered DMOPs, in which the specific parameters of p-value are set according to [49]. P-values obtained by five algorithms on JY1-JY9, FDA1 and FDA4 have been shown in Table VIII of the supplementary material that LPSDM is the best performing algorithm, indicating that LPSDM is strongly robust throughout the whole evolutionary process.

Apart from tabular presentation, we provide evolution curves of the average IGD values for six test instances in Fig. 4. It can be clearly seen that, compared with the other algorithms, LPSDM responds to changes more stably and recovers faster for most of the test problems, thereby obtaining better approximations to the moving POF. The IGD curve of LPSDM fluctuates for JY7 in a few environmental changes, but it is better than that of the other algorithms. In summary, we conclude that LPSDM is very capable of maintaining good diversity and fast convergence while tracking the moving POF/POS. For a graphical view of the algorithms' tracking ability, we also plot their obtained POFs of JY1, JY2, JY6 and JY8 over 20 time windows in Fig. 5. Fig. 5 shows that LPSDM is very capable of tracking environmental changes and handles JY6 much better than the others.

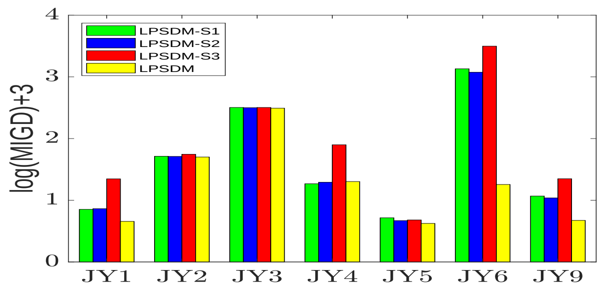


Fig. 6: MIGD values obtained by different components.

B. *Study of Different Components of LPSDM*

This subsection is devoted to studying the role different components playing in LPSDM for dynamic multi-objective optimization. LPSDM has two key components: the layered prediction strategy for change response and the subspace-based strategy for diversity maintenance. The subspace-based strategy has two subcomponents: gap filling approach and probability-based production. To deeply examine the role that each component plays in dynamic optimization, we adapt the original LPSDM into three variants. The first variant (LPSDM-S1) replaces the probability-based approach with a normal DE reproduction. The second variant (LPSDM-S2) discards the gap filling approach of LPSDM. The third variant, LPSDM-S3, does not have the layered prediction strategy and randomly initializes a population when an environmental change is detected. These three variants are compared with the original LPSDM on seven problems with the setting of $(\tau_t, n_t) = (20, 10)$. The MIGD and MHVD values, obtained by different LPSDM variants, are presented in Fig. 6, and Fig. 3 of the supplementary material.

The experimental results show that the LPSDM performs significantly better than the three variants on most JY problems except JY4 in terms of two measures, implying all key components are crucial to the high performance of LPSDM on different problems. The inconsistency between MIGD and MHVD for JY4 may be due to the irregular shape of the POF, which affects the calculation of MHVD greatly [48]. In most cases, LPSDM-S1 is significantly worse than LPSDM on the MIGD and MHVD, suggesting that the probability-based approach is important for LPSDM to maintain population diversity. Similarly, LPSDM-S2 obtains worse MIGD and MHVD values than LPSDM, indicating the gap filling approach enhances considerably population distribution, which in turn results in better competitiveness for LPSDM. Without the layered prediction strategy, LPSDM-S3 is almost defeated by LPSDM on every considered performance indicator. This highlights the effectiveness of the proposed layered prediction

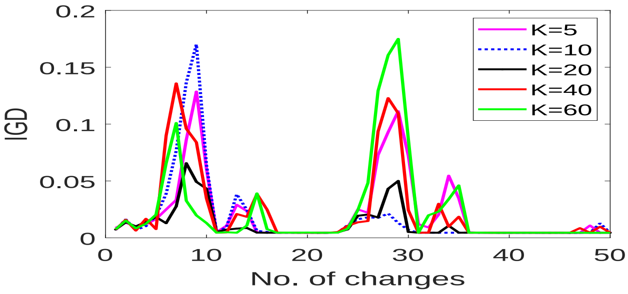


Fig. 7: The average IGD values of LPSDM with different K values on JY6.

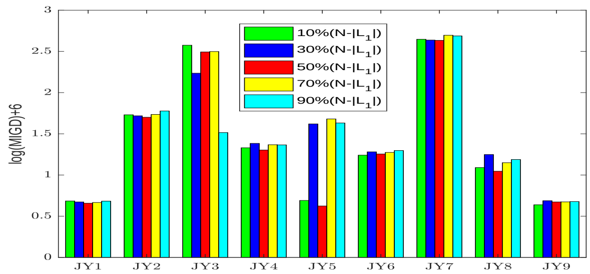


Fig. 8: Performance comparison of LPSDM with different sizes of $Sub2$ on MIGD with $n_t = 10$ and $\tau_t = 20$, where N is size of the population and L_1 is the size of first nondominated level.

strategy in dealing with dynamic environments. Overall, the analysis reveals that the LP approach contributes most to LPSDM, followed by gap filling and probability-based reproduction which are almost equivalently important to LPSDM.

C. Influence of K Values

To study whether the proposed algorithm is sensitive to the value of K, we test LPSDM with varying $K \in \{5, 10, 20, 40, 60\}$, fixed $\tau_t = 30$ and $n_t = 10$. The experimental result for JY6 is shown in Fig. 7. It shows K is a key parameter affecting the performance of LPSDM. LPSDM has significant performance degradation for either too big or too small values of K as the IGD value fluctuates widely for extreme cases. The reasons are as follows. A small value of K means a small number of subspaces and each subspace can have a large number of individuals on average. The chance that a subspace has zero individuals is little. In other words, the gap filling approach does not help in this case. On the contrary, a large value of K creates too many subspaces, and it is likely that a significant number of subspaces can be empty. In this case, the gap filling approach will fill many gaps and the population can be dramatically changed, leading to reduced convergence. In addition, large K increases computations in population association.

Therefore, it is desirable to choose K in between the extremes. This is demonstrated in Fig. 7 where $K = 20$ (20% of the population size) appears to be the best choice. We thus suggest K is reasonably near 20% of the population size.

D. Influence of the size of $Sub2$

The size of $Sub2$ is a key parameter that can affect the performance of LPSDM. Therefore, LPSDM is tested on different sizes of $Sub2$ varying from $10\%(N - |L_1|)$ to $90\%(N - |L_1|)$. Fig. 8 shows that LPSDM has the best performance for most of the selected problems when the size of $Sub2$ is $50\%(N - |L_1|)$. For the other problems, such as JY5-6 and JY9, a size of $50\%(N - |L_1|)$ for $Sub2$ also helps LPSDM obtain very small MIGD values. This demonstrates the robustness and effectiveness of LPSDM with the choice of $50\%(N - |L_1|)$ for the size of $Sub2$.

E. Comparison with other prediction models

To study the advantages and disadvantages between the LP strategy and other prediction models, we select some prediction models, including KF [1] and PPS [19]. Here, the LP strategy in LPSDM is replaced with PPS and KF. The experimental results for JY1-JY5 and JY9 are given in Fig. 4 of the supplementary material. The results of MIGD show that LPSDM is the best among the three SDM variants. So it is clear that LP is more competitive than KF and PPS. In addition, LP is simple and computationally efficient in comparison with the complex prediction methods KF and PPS. In conclusion, the LP strategy is more suitable for the dynamic environments tested in this paper than PPS and KF.

F. Comparison with other diversity strategy

To fairly study the advantages and disadvantages of the SDM strategy, the response strategy to the change in DNSGA-II is replaced with the LP strategy, in which 20% of the offspring set is randomly generated in each generation, denoted as LP+NSGAII-RND in Fig. 5 of the supplementary material. The experimental results for FDA1, dMOP1, dMOP2, JY1, JY4 and JY8 are given, which shows that LPSDM is the best among the two LP variants. So it is clear that SDM can improve diversity loss in every generation of evolution regardless of environmental changes and alleviate the situation that population diversity may not be fully recovered in a short period of change response. Consequently, the SDM strategy can maintain population diversity in the optimization process well.

G. Discussion

We have developed a novel algorithm, i.e., LPSDM, for DMO, and our extensive empirical studies have demonstrated the promise of LPSDM for a large variety of benchmark DMOPs studied in this paper. Specifically, LPSDM has shown the best performance on DMOPs without strong variable linkages, e.g., most of the FDA, dMOP and JY1, JY5, and with simple diversity-resistant schemes (e.g., JY6), and with mixed changes (e.g., JY2, JY4, JY9). In some patterns of changes during the evolution, such as mixed POF of convexity-concavity that changes over time, a problem that changes between different types, time-varying relationships between variables, changes are slight and/or smooth, LPSDM has the best performance compared with some state-of-the-art algorithms, which suggests that LPSDM tracks the moving POSs/POFs effectively and efficiently in these DMOPs. Accordingly, the conclusion that LPSDM can work well on DMOPs with different characteristics during periodical environments has been proved, especially for DMOPs with mixed changes, weak variable linkages, and simple diversity-resistant schemes. To summarize, our extensive empirical studies have unveiled two main reasons behind the high performance of LPSDM:

i) LP reinitializes the population by recognizing that population members have different levels of progress in evolution, enabling fast convergence to the new POF when an environment changes.
ii) SDM maintains diversity rigorously throughout evolution, addressing diversity loss due to environmental changes. Meanwhile, it also provides a good way of select mating pools for reproduction, which is to produce high-quality offspring, improve population diversity and speed up population convergence.

However, like other algorithms, LPSDM has some drawbacks too. One drawback is that LPSDM struggles to handle dynamic variable dependencies that gradually become more severe, e.g., JY3 and FDA3. Another drawback comes from the inefficiency of LPSDM for handling problems with severe changes in multimodality while dynamic shifts of the position of POF, like JY7. This is because these problems increase the difficulty for the population to jump out of local optima. The POFs with time-varying mixed segments (see results on JY8) may challenge LPSDM.

Besides, the focus of this work is mainly on the theoretical study of algorithm design, which will lay a solid foundation for the study of DMO, and practitioners can benefit from the point of view of algorithm design. However, we also recognize the need for real-world evidence for further justifying LPSDM. Linking theory to practice is important but can be difficult for LPSDM at its current immature stage. One reason is that LPSDM is designed to suit well-defined DMOPs. Real-world applications are often poorly presented and much effort is required for proper problem formalism, which often needs expert knowledge [50], [51], [52]. Another reason is that even if a well-defined DMOP has been available, in most cases, problem-specific knowledge is needed for initialization of candidate solutions [51], which is different from the commonly-adopted random initialization strategy in many evolutionary algorithms including the proposed one. Therefore, problem-specific versions of LPSDM have to be developed in order for real-world application. This is beyond the scope of our current work but can be interesting for our future work.

In the proposed algorithms, the choice of 3 groups is justified as follows. First, the selection of the number of groups is determined by the number of operations developed for change response. In this work, we consider three commonly used operations for population reinitialization in response to an environmental change, namely linear prediction, guided mutation, and random generation. We aim to use these operations collectively through our layered prediction strategy rather than focus on one of them as done in the literature. This thus leads to the need for three groups. Second, the three operations in the three groups can strike a balance between population diversity and convergence. The time-series prediction that uses historical search information enables faster convergence. The guided mutation handles well small environmental changes whereas random generation is effective for severe environmental changes [8]. We believe the combination of these three operations maximizes robustness to various dynamic environments. Third, the choice of three groups also has efficiency benefits since more groups will involve more computations in group allocation. Lastly, the number of groups cannot be too big, provided that the population size is often limited, and each group should have an appropriate number of individuals in order to anticipated group intelligence. Therefore, three groups are used in our approach, which has performed well as anticipated.

VII. Conclusion

This paper has proposed a novel DMOEA to deal with DMOPs. The algorithm employs a layered prediction strategy to respond to environmental changes rapidly. This strategy recognizes that population members may progress differently in evolution and this information can be useful for dynamic optimization if it is properly exploited. Thus, in response to environmental changes, this strategy first relocates the fittest. After relocation, the adaptability of these previously fittest individuals in the new environment is assessed to support the re-initialization of a portion of the remaining individuals. Additionally, the algorithm introduces a subspace-based strategy to maintain population diversity in every generation of evolution. This can largely prevent the population from diversity loss during the search, especially when an environmental change occurs.

The proposed algorithm is extensively justified by a thorough comparison with five state-of-the-art algorithms on a wide range of test problems. The comparison shows that our algorithm achieves better performance in most of the dynamic environments used in this paper. Additional sensitivity analysis and discussion reveal the robustness of the proposed algorithm.

In our future work, we will explore the possibility of combining the proposed framework and other popular static EAs for better performance. We will also develop dynamic self-adaptation throughout the run based on population information and environmental changes to handle dynamic environments. Besides, we will investigate the practicability of the proposed algorithms toward real-life problems.

References

[1] 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.

[2] Y. Liu, J. Liu, and Y. Jin, “Surrogate-assisted multipopulation particle swarm optimizer for high-dimensional expensive optimization,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–14, 2021.

[3] L. Xue, P. Wang, H. Cheng, P. Zeng, and H. Yu, “Event-driven dynamic job shop scheduling execution based on improved genetic algorithm and ontology,” in *Chinese Automation Congress (CAC)*, 2017, pp. 6430–6435.

[4] E. Q. Wu, Z. Tang, Y. Yao, X.-Y. Qiu, P.-Y. Deng, P. Xiong, A. Song, L.-M. Zhu, and M. Zhou, “Scalable gamma-driven multilayer network for brain workload detection through functional near-infrared spectroscopy,” *IEEE Transactions on Cybernetics*, pp. 1–15, 2021.

[5] J. Guo, W. Zhang, C. Liu, and L. Chu, “Adaptive model predictive control strategy of hybrid electric bus based on soc programming,” in *IEEE Information Technology, Networking, Electronic and Automation Control Conference*, 2016, pp. 796–799.

[6] M. Jiang, Z. Wang, H. Hong, and G. G. Yen, “Knee point-based imbalanced transfer learning for dynamic multiobjective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 1, pp. 117–129, 2021.

[7] A. Novikov, A. Levykin, and E. Novikov, “(m, k)-methods for control theory problems,” in *2019 15th International Asian School-Seminar Optimization Problems of Complex Systems (OPCS)*, 2019, pp. 120–124.

[8] K. Deb, U. B. N. Rao, and S. Karthik, “Dynamic multi-objective optimization and decision-making using modified nsga-ii: a case study on hydro-thermal power scheduling,” in *International Conference on Evolutionary Multi-criterion Optimization*, 2007, pp. 803–817.

[9] Y. Zhang, S. Wang, and J. Shi, “Robust fault detection and estimation of sensor fault for closed-loop control systems,” in *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2020, pp. 1155–1160.

[10] 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.

[11] 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.

[12] M. Rong, D. Gong, Y. Zhang, Y. Jin, and W. Pedrycz, “Multidirectional prediction approach for dynamic multiobjective optimization problems,” *IEEE Transactions on Cybernetics*, vol. 49, no. 9, pp. 3362–3374, 2019.

[13] C. Goh and K. C. Tan, “A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 1, pp. 103–127, 2009.

[14] Y. Hu, J. Ou, J. Zheng, J. Zou, S. Yang, and G. Ruan, “Solving dynamic multi-objective problems with an evolutionary multi-directional search approach,” *Knowledge-Based Systems*, vol. 194, p. 105175, 2020.

[15] W. Koo, C.-K. Goh, and K. Tan, “A predictive gradient strategy for multiobjective evolutionary algorithms in a fast changing environment,” *Memetic Computing*, vol. 2, pp. 87–110, 06 2010.

[16] L. Cao, L. Xu, E. D. Goodman, C. Bao, and S. Zhu, “Evolutionary dynamic multiobjective optimization assisted by a support vector regression predictor,” *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 305–319, 2020.

[17] M. Jiang, Z. Huang, L. Qiu, W. Huang, and G. G. Yen, “Transfer learning-based dynamic multiobjective optimization algorithms,” *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 4, pp. 501–514, 2018.

[18] R. Rambabu, P. Vadakkepat, K. C. Tan, and M. Jiang, “A mixture-of-experts prediction framework for evolutionary dynamic multiobjective optimization,” *IEEE Transactions on Cybernetics*, pp. 1–14, 2019.

[19] 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.

[20] 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.

[21] S. B. Gee, K. C. Tan, and C. Alippi, “Solving multiobjective optimization problems in unknown dynamic environments: An inverse modeling approach,” *IEEE Transactions on Cybernetics*, vol. 47, no. 12, pp. 4223–4234, 2017.

[22] S. Yang, “Genetic algorithms with elitism-based immigrants for changing optimization problems,” *Lecture Notes in Computer Science*, vol. 4448, pp. 627–636, 2007.

[23] H. Li and Q. Zhang, “Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.

[24] R. Chen, K. Li, and X. Yao, “Dynamic multiobjectives optimization with a changing number of objectives,” *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 157–171, 2018.

[25] 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.

[26] Q. Chen, J. Ding, S. Yang, and T. Chai, “A novel evolutionary algorithm for dynamic constrained multiobjective optimization problems,” *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 4, pp. 792–806, 2020.

[27] 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.

[28] 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.

[29] 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.

[30] R. Azzouz, S. Bechikh, and L. B. Said, “A dynamic multi-objective evolutionary algorithm using a change severity-based adaptive population management strategy,” *Soft Computing*, vol. 21, no. 4, pp. 885–906, 2017.

[31] M. Rong, D. Gong, W. Pedrycz, and L. Wang, “A multimodel prediction method for dynamic multiobjective evolutionary optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 290–304, 2020.

[32] L. Cao, L. Xu, E. D. Goodman, and H. Li, “Decomposition-based evolutionary dynamic multiobjective optimization using a difference model,” *Applied Soft Computing*, vol. 76, pp. 473–490, 2019.

[33] Y. Hu, J. Zheng, J. Zou, J. Ou, and R. Wang, “A dynamic multi-objective evolutionary algorithm based on intensity of environmental change,” *Information Sciences*, vol. 523, 03 2020.

[34] Q. Li, J. Zou, S. Yang, J. Zheng, and G. Ruan, “A predictive strategy based on special points for evolutionary dynamic multi-objective optimization,” *Soft Computing*, vol. 23, no. 11, pp. 3723–3739, 2019.

[35] J. Zou, Q. Li, S. Yang, H. Bai, and J. Zheng, “A prediction strategy based on center points and knee points for evolutionary dynamic multi-objective optimization,” *Applied Soft Computing*, vol. 61, pp. 806–818, 2017.

[36] Q. Zhang, S. Yang, S. Jiang, R. Wang, and X. Li, “Novel prediction strategies for dynamic multiobjective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 260–274, 2020.

[37] S. Sahmoud and H. R. Topcuoglu, “A memory-based nsga-ii algorithm for dynamic multi-objective optimization problems,” in *European Conference on the Applications of Evolutionary Computation*, vol. 9598, 2016, pp. 296–310.

[38] J. Branke, “Memory enhanced evolutionary algorithms for changing optimization problems,” in *Proceedings of the Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, vol. 3, 1999, pp. 1875–1882.

[39] G. Ruan, G. Yu, J. Zheng, J. Zou, and S. Yang, “The effect of diversity maintenance on prediction in dynamic multi-objective optimization,” *Applied Soft Computing*, vol. 58, pp. 631–647, 2017.

[40] J. Ou, J. Zheng, G. Ruan, Y. Hu, J. Zou, M. Li, S. Yang, and X. Tan, “A pareto-based evolutionary algorithm using decomposition and truncation for dynamic multi-objective optimization,” *Applied Soft Computing*, vol. 85, p. 105673, 2019.

[41] Z. Liang, S. Zheng, Z. Zhu, and S. Yang, “Hybrid of memory and prediction strategies for dynamic multiobjective optimization,” *Information Sciences*, vol. 485, pp. 200–218, 2019.

[42] R. Liu, J. Li, J. Fan, C. Mu, and L. Jiao, “A coevolutionary technique based on multi-swarm particle swarm optimization for dynamic multi-objective optimization,” *European Journal of Operational Research*, vol. 261, no. 3, pp. 1028–1051, 2017.

[43] D. Gong, B. Xu, Y. Zhang, Y. Guo, and S. Yang, “A similarity-based cooperative co-evolutionary algorithm for dynamic interval multiobjective optimization problems,” *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 1, pp. 142–156, 2020.

[44] Y. Wu, W. Jin, and X. Liu, “A directed search strategy for evolutionary dynamic multiobjective optimization,” *Soft Computing*, vol. 19, no. 11, pp. 3221–3235, 2015.

[45] M. Greeff and A. P. Engelbrecht, “Solving dynamic multi-objective problems with vector evaluated particle swarm optimisation,” in *IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, 2008, pp. 2917–2924.

[46] K. Deb and M. Goyal, “A combined genetic adaptive search (geneas) for engineering design,” *Computer Science and Informatics*, vol. 26, 08 1999.

[47] F. Wilcoxon, “Individual comparisons by ranking methods,” in *Breakthroughs in Statistics*, 1992, pp. 196–202.

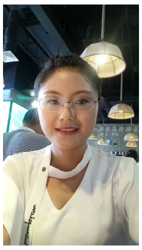
[48] S. Jiang, S. Yang, and M. Li, “On the use of hypervolume for diversity measurement of pareto front approximations,” in *IEEE Symposium Series on Computational Intelligence (SSCI)*, 2016, pp. 1–8.

[49] S. Biswas, S. Das, P. N. Suganthan, and C. A. C. Coello, “Evolutionary multiobjective optimization in dynamic environments: A set of novel benchmark functions,” in *IEEE Congress on Evolutionary Computation (CEC)*, 2014, pp. 3192–3199.

[50] B. Lokeshgupta and S. Sivasubramani, “Multi-objective dynamic economic and emission dispatch with demand side management,” *International Journal of Electrical Power & Energy Systems*, vol. 97, pp. 334–343, 2018.

[51] 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, 2017.

[52] J. Qiao and W. Zhang, "Dynamic multi-objective optimization control for wastewater treatment process," *Neural Computing and Applications*, vol. 29, no. 11, pp. 1261–1271, 2018.



Yaru Hu received the B.Sc. degree in Computer Science and Engineering from the College of Information Engineering from Zhengzhou University of Aeronautics, Zhengzhou, China in 2016.

she is currently pursuing the Ph.D. degree in mathematics with the Department of Mathematics and Computational Science, China University of Xiangtan University, Xiangtan, China.

Her current research interests include evolutionary computation, multi-objective optimization, and dynamic optimization.

Jinhua Zheng received the B.Sc. degree in computer software and the M.Sc. degree in computer application from the Nanjing University of Science and Technology, Nanjing, China, in 1986 and 1989, respectively, and the Ph.D. degree in control theory and control engineering from Central South University, Changsha, China, in 2000. received the Ph.D. degree in control theory and control engineering from Central South University, Changsha, China, in 2000.

Since 2000, he has been a Professor with the College of Information Engineering, Xiangtan University, China. He has over 300 research publications, including two academic monograph and five textbooks. His current research interests include evolutionary computation, evolutionary multi-objective optimization, robust optimization, and evolutionary algorithms in real-world applications.

Shouyong Jiang received the B.Sc. degree in information and computation science and the M.Sc. degree in control theory and control engineering from in Northeastern University, China in 2011 and 2013, respectively, and the Ph.D. degree in computer science from De Montfort University, UK in 2017. His current research interests include AI optimisation, evolutionary computation, and machine learning.

Shengxiang Yang received the B.Sc. and M.Sc. degrees in automatic control and the Ph.D. degree in systems engineering from Northeastern University, Shenyang, China in 1993, 1996, and 1999, respectively.

He is currently a Professor in Computational Intelligence and Director of the Centre for Computational Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester, U.K. He has over 210 publications. His current research interests include evolutionary and genetic algorithms, swarm intelligence, computational intelligence in dynamic and uncertain environments, artificial neural networks for scheduling, and relevant real-world applications.

Prof. Yang is the Chair of the Task Force on Evolutionary Computation in Dynamic and Uncertain Environments, under the Evolutionary Computation Technical Committee of the IEEE Computational Intelligence Society and the Founding Chair of the Task Force on Intelligent Network Systems, under the Intelligent Systems Applications Technical Committee of the IEEE Computational Intelligence Society.

Juan Zou received the B.Sc. degree in Computer Applications Technology and the M.Sc. degree in Computer Science and Engineering from the Xiangtan University of Science and Technology, Xiangtan, China, in 2001 and 2005, respectively, and the Ph.D. degree in mathematics from Xiangtan University, Xiangtan, China, in 2014.

She is currently a Professor of Computational Intelligence in the College of Computer and Cyberspace Security of Xiangtan University, Xiangtan, China. She has over 50 publications and one monograph. Her current research interests include multi-objective optimization, swarm intelligence, artificial neural networks, and related applications.