Future Generation Computer Systems 161 (2024) 390–403

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

A dynamic multi-objective evolutionary algorithm with variable stepsize and dual prediction strategies

Hu Peng [a,b,*], Chen Pi [a], Jianpeng Xiong [a], Debin Fan [a], Fanfan Shen [c]

[a] School of Computer and Big Data Science, Jiujiang University, Jiujiang 332005, PR China
[b] Jiujiang Key Laboratory of Digital Technology, Jiujiang 332005, PR China
[c] School of Computer Science, Nanjing Audit University, Nanjing 211815, PR China

ARTICLE INFO

Keywords:
Dynamic multi-objective optimization
Variable stepsize
Dual prediction strategies

ABSTRACT

The prediction strategy is a key method for solving dynamic multi-objective optimization problems (DMOPs), particularly the commonly used linear prediction strategy, which has an advantage in solving problems with regular changes. However, using the linear prediction strategy may have limited advantages in addressing problems with complex changes, as it may result in the loss of population diversity. To tackle this issue, this paper proposes a dynamic multi-objective optimization algorithm with variable stepsize and dual prediction strategies (VSDPS), which aims to maintain population diversity while making predictions. When an environmental change is detected, the variable stepsize is first calculated. The stepsize of the nondominated solutions is expressed by the centroid of the population, while the stepsize of the dominated solutions is determined by the centroids of the clustered subpopulations. Then, the dual prediction strategies combine an improved linear prediction strategy with a dynamic particle swarm prediction strategy to track the new Pareto-optimal front (PF) or Pareto-optimal set (PS). The improved linear prediction strategy aims to enhance the convergence of the population, while the dynamic particle swarm prediction strategy focuses on preserving the diversity of the population. There have also been some improvements made in the static optimization phase, which are advantageous for both population convergence and diversity. VSDPS is compared with six state-of-the-art dynamic multi-objective evolutionary algorithms (DMOEAs) on 28 test instances. The experimental results demonstrate that VSDPS outperforms the compared algorithms in most instances.

1. Introduction

There are some special metaheuristic algorithms that allow to determine simultaneously the optimized value of the decision variables as well as the optimized number of decision variables, such as [1–3]. Specifically, such algorithms allow solving a constrained nonlinear programming problem (CNLPP) defined over a design space of varying dimension (and with a variable number of constraint functions as well as decision variables). Currently, these kinds of algorithms are widely exploited in different fields for various aims, such as reverse engineering, geometrical modelling or medical images reconstruction. Multi-objective optimization problems (MOPs) [4–6] involve the interaction and conflict between two or more objectives. Numerous studies have been conducted on MOPs, particularly in the field of evolutionary computing. Typically, MOPs are addressed in a static environment where the objective functions and constraints remain unchanged. However, MOPs are not always stationary, as they might contain multiple parameters, constraints, or conflicting objectives that change over time. These types of problems are referred to as dynamic multi-objective optimization problems (DMOPs) [7–9]. There are many applications of DMOPs in daily life, such as greenhouse control [10], machine learning [11], weapon selection [12] and unmanned aerial vehicle online path planning [13]. Due to the practical implications of DMOPs, there has been a growing research interest in dynamic multi-objective evolutionary algorithms (DMOEAs) in recent years.

The prediction strategy [14–16], which has gained popularity among researchers for solving DMOPs, focuses on using historical data or time series of past solutions to predict the population after environmental changes. Among the prediction strategies, the linear prediction strategy is commonly used to track the Pareto-optimal front (PF) or Pareto-optimal set (PS) in the new environment. These strategies include a first-order difference model-based (MOEA/D-FD) [17], Kalman filter prediction (MOEA/D-KF) [18], a population prediction strategy (PPS) [19], and a prediction strategy based on center points and knee points (CKPS) [20]. Typically, these algorithms are effective

[*] Corresponding author at: School of Computer and Big Data Science, Jiujiang University, Jiujiang 332005, PR China.
E-mail address: hu_peng@whu.edu.cn (H. Peng).

https://doi.org/10.1016/j.future.2024.07.028
Received 16 December 2023; Received in revised form 2 June 2024; Accepted 13 July 2024
Available online 19 July 2024

for linear environmental changes, but less so for nonlinear changes. The reason for this is that if only a linear prediction strategy is used, the predicted population may lack diversity. Currently, most algorithms employ mutation operators or randomly generate individuals to enhance population diversity. In dynamic nondominated sorting genetic algorithm (DNSGA-II) [21], both a mutation operator and randomly generated individuals are employed to enhance population diversity. In the effect of diversity maintenance on prediction (DMS) [22], a random diversity maintenance strategy is proposed to improve population diversity. In combining a hybrid prediction strategy and a mutation strategy (HPPCM) [23], a precision controlled mutation operator is used to enhance population diversity.

There are some clustering methods that are already used for solving DMOPs, such as niche prediction strategy (PCPB) [24], manifold clustering-based prediction (MCP) [25], multidirectional prediction (MDP) [26] and so on. The clustering method is typically utilized to assist algorithms in making predictions. In general, the population is divided into several clusters, and then individuals in each cluster are predicted using different methods to improve the accuracy of the prediction.

The generational gap between two environmental changes in dynamic environments is typically minimal. Dynamic multi-objective optimization (DMO) is often regarded as a hybrid of a static multi-objective optimizer and a change response by many scholars. Currently, most algorithms focus on the change response part, while few concentrate on the static multi-objective optimizer part. Static multi-objective optimizers are typically chosen from multi-objective evolutionary algorithms (MOEAs). The most commonly used MOEAs include multi-objective optimization problems with complicated pareto sets (MOEA/D-DE) [27], a fast and elitist multi-objective genetic algorithm (NSGA-II) [28], and a regularity model-based multi-objective estimation of distribution algorithm (RM-MEDA) [29]. However, due to the time-varying nature of DMOPs, some excellent MOEAs are not as effective when used as static multi-objective optimizers. Therefore, a suitable static multi-objective optimizer is crucial for solving DMOPs.

Based on these considerations, it is important to maintain both the rapid convergence of the population to the current $\mathrm{PS}_t/\mathrm{PF}_t$ and the diversity of the population when making predictions. Therefore, a dynamic multi-objective optimization algorithm with variable stepsize and dual prediction strategies (VSDPS) is proposed to track the new $\mathrm{PS}_t/\mathrm{PF}_t$, where variable stepsize is used for the solutions when making the dual prediction strategies. In addition, there are some improvements based on MOEA/D-DE, which is advantageous for both population convergence and diversity. Experimental results show that VSDPS outperforms the other algorithms in most instances. The main contributions of this paper can be summarized as follows:

(1) Variable stepsize is used to improve prediction accuracy, where the centroid of the population determines the stepsize of the nondominated solutions, and the centroids of the subpopulations after clustering determine the stepsize of the dominated solutions.
(2) The dual prediction strategies are proposed, where the improved linear prediction strategy is employed to enhance the population convergence, and the dynamic particle swarm prediction strategy aims at maintaining the population diversity.
(3) The static multi-objective optimization algorithm MOEA/D-DE is improved by incorporating the genetic algorithm (GA) operator and variants of the differential evolution (DE) operator to monitor the current PS/PF more quickly along with the DE/rand/1 operator.

The rest of this paper is organized as follows. In Section 2, background and motivation are introduced. In Section 3, the core strategies of VSDPS are presented. Section 4 provides a brief introduction to the comparison algorithms, test suites, performance indicators, and experimental setup. In Section 5, a series of experimental results are presented and analyzed. Finally, Section 6 concludes this paper and discusses some future work.

2. Preliminary

2.1. Dynamic multi-objective optimization problems

DMOPs refer to problems with multiple objectives that must be met within a range of constraints, where the constraints and/or objectives may change over time. Taking the minimization problem as an example, the mathematical expression for DMOPs is as follows [30,31]:

$$\begin{cases} min \quad \mathbf{F}(x,t) = \left(f_1(x,t), f_2(x,t), \dots, f_m(x,t)\right) \\ s.t. \quad g_i(x,t) \le 0, i \in 1,2,\dots,p \\ \qquad h_j(x,t) = 0, i \in 1,2,\dots,q \\ \qquad x \in \Omega_x, \quad t \in \Omega_t \end{cases} \tag{1}$$

where x is a vector of decision variables. $\mathbf{F}$ is a set of objective functions that require minimization at time t and m is the dimension of the objective space. The functions g and h denote the inequality constraint and the equation constraint at time t, respectively. $\Omega_x \in R^n$ is the feasible decision space and $\Omega_t \in R$ is the space of time variables. The discrete-time variable t is defined as follows [32]:

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

where n_t is the severity of environmental change, τ is the number of evolutionary iterations, and τ_t is the frequency of environmental change.

Definition 1 (*Dynamic Pareto Dominance*)**.** At time t, a decision vector x_2 is considered to Pareto dominate decision vector x_1 in the decision space Ω_x, denoted as $x_2 \prec_t x_1$, if and only if [33]:

$$\begin{cases} \forall i \in \{1,\dots,m\}, f_i\left(x_2,t\right) \le f_i\left(x_1,t\right) \\ \exists j \in \{1,\dots,m\}, f_j\left(x_2,t\right) < f_j\left(x_1,t\right) \end{cases} \tag{3}$$

Definition 2 (*Dynamic Pareto-optimal Solution*)**.** At time t, if a solution x is not dominated by other solutions in the decision space, it is called a Pareto-optimal solution [34].

Definition 3 (*Dynamic Pareto-optimal Set, PS_t*)**.** At time t, a collection of all Pareto-optimal solutions is called the Dynamic Pareto-optimal Set, which can be defined as follows [35]:

$$\mathrm{PS}_t = \left\{x \in \Omega_x \mid \nexists y \in \Omega_x, y \prec_t x\right\} \tag{4}$$

Definition 4 (*Dynamic Pareto-optimal Front, PF_t*)**.** At time t, a collection of all nondominated solutions with respect to the objective space constitutes the Dynamic Pareto-optimal Front, which can be defined as follows [36]:

$$\mathrm{PF}_t = \left\{\mathbf{F}(x,t) \mid x \in \mathrm{PS}_t\right\} \tag{5}$$

Based on the dynamic changes of the PS_t and PF_t, Farina et al. [37] classified DMOPs into four different types.

Type I : PS_t changes over time and PF_t is fixed.
Type II : Both PS_t and PF_t change over time.
Type III : PS_t is fixed and PF_t changes over time.
Type IV : Both PS_t and PF_t remain fixed.

In DMOPs, most researchers concentrate on the first three types of changes, and the experiments conducted by VSDPS do as well.

2.2. Dynamic multi-objective evolutionary algorithm

Environmental change detection, environmental change response, and static multi-objective optimizer are important components of DMOAEs, which constitute the general framework of DMOEAs. The pseudo-code of DMOEAs is presented in Algorithm 1.

Algorithm 1 The framework of DMOEAs

```
Initialize the time step $t = 0$;
while the termination criterion is not met do
    if an environmental change is detected then
        Re-evaluate all detectors;
        Obtain the initial population at time $t + 1$ by using a response mechanism;
        $t = t + 1$;
    else
        Update the population by the MOEA;
    end if
end while
```

2.2.1. Environmental change response

In order to quickly track the PS_t/PF_t after detecting environmental changes, it is necessary to respond to the changes. Existing environmental change response approaches can be broadly classified into five categories, which are prediction-based approaches, diversity introduction based approaches, diversity maintenance based approaches, memory-based approaches, and multi-population based approaches.

(1) Prediction-based approaches: Environmental changes are sometimes predictable. Utilizing historical information to forecast environmental changes enables algorithms to adapt to these changes swiftly. Zhou et al. [19] proposed a population prediction strategy that initializes the entire population by combining the estimated manifold and the predicted center point. Tan et al. [18] proposed a strategy based on the Kalman filter (KF) model, which combines KF predictions with a random re-initialization method to generate the initial population at the next moment. In recent years, there has been a growing trend of integrating particle swarm optimization (PSO) algorithms with prediction-based approaches to forecast population in new environments. Liu et al. [38] proposed a cooperative particle swarm optimization with reference point-based prediction strategy, in which multiple particle swarms co-evolve to track the new PS_t/PF_t. Wang et al. [39] proposed a particle swarm prediction strategy and a prediction adjustment strategy. The particle swarm prediction strategy is employed to forecast the population after environmental changes, while the prediction adjustment strategy is employed to improve the accuracy of the predicted solutions. While prediction-based approaches are effective for DMOPs, they may be less effective for nonlinear or complex problems.

(2) Diversity introduction based approaches: In the process of tracking PS_t/PF_t, it is impossible to detect changes when the optimization algorithm converges to one region of the problem landscape while the landscape changes in the other region, leading to a lack of diversity in the algorithm. The lack of diversity will make it difficult for the algorithm to find the global optimum. Therefore, it is vital to incorporate diversity into the optimization process. Currently, the primary methods utilized to introduce diversity are mutation [21,40] and random initialization [21,41]. Although diversity introduction based approaches are easy to implement, they are blind and may mislead the population. In addition, it may not work effectively when the environmental changes are random, severe, or rapid.

(3) Diversity maintenance based approaches: Diversity maintenance based approaches aim to preserve population diversity throughout the optimization process. Zeng et al. [42] proposed a dynamic orthogonal multi-objective evolutionary algorithm (DMOEA), which directly uses the PS from the previous environment as the initial population for the next environment. The diversity maintenance based approaches are simple to implement. The approaches do not explicitly respond to the new environment, but they adapt to the new environment with their inherent approaches. Therefore, they can only be applied to DMOPs with less environmental changes, and they may not be effective when the environment changes dramatically.

(4) Memory-based approaches: Due to the periodicity of some DMOPs, historical information can help DMOEAs respond to the environmental changes quickly. Azzouz et al. [43] proposed a hybrid population management strategy incorporating memory, local search, and random strategies. Liang et al. [44] proposed a dynamic strategy based on the hybrid of the memory and prediction strategies, which evaluates the resemblance between the current environmental change and the historical environmental change to decide which response to employ. Memory-based approaches have obvious advantages in solving periodic problems. When dealing with non-periodic problems, the individuals stored in the memory may not track the frontier effectively, potentially leading the population in the wrong direction. Information stored in memory may also be redundant, and retrieving information from memory requires a significant amount of computation.

(5) Multi-population based approaches: The multi-population based approaches involve using multiple strategies to generate subpopulations that explore different areas. Liu et al. [40] proposed a multi-population particle swarm optimization algorithm (CMPSODMO), where each subpopulation optimizes a specific objective function. Xu et al. [45] proposed the environment sensitivity-based cooperative co-evolutionary algorithms (DNSGAII-CO and DMOPSO-CO), which divide the decision variables into two components and adopt two populations to cooperatively optimize the two components. Zhang et al. [46] designed prediction nondominated set, sampling strategy, and shrinking strategy, which can respond to environmental changes effectively. The advantage of multi-population based approaches is that they can preserve the diversity of the population efficiently, but the number of subpopulations is a problem worth exploring.

Prediction-based approaches are frequently employed, with the linear prediction strategy being particularly popular. However, relying on linear prediction strategy solely may result in a lack of population diversity. Currently, most algorithms utilize diversity introduction based approaches or diversity maintenance based approaches to enhance or preserve population diversity. VSDPS blends prediction and diversity, aiming to maintain population diversity while making predictions.

2.2.2. Static multi-objective optimizer

When the environment remains constant, the population must be optimized using a static multi-objective optimizer. An efficient static multi-objective optimizer can help track the PS_t/PF_t quickly and accurately before the next environmental change. Therefore, the selection of a static multi-objective optimizer also has a certain impact on the performance of DMOEAs. At present, the widely used static multi-objective optimizers include RM-MEDA [29], MOEA/D-DE [27], and so on. In some papers, new methods for static optimization are proposed, such as a layered prediction and subspace-based diversity maintenance strategy (LPSDM) [47], hybrid environmental change responses strategy (MDMEA-HCR) [33]. There are also some improvements in the static multi-objective optimizer in VSDPS.

2.3. Motivation

In order to solve DMOPs effectively, it is important to meet the following two requirements as much as possible. If the environment changes, the algorithm must be able to detect the change sensitively and respond to it effectively. If the environment does not change, the algorithm must quickly track the PS_t/PF_t of the current environment.

At the same time, it is also crucial to maintain a balance between convergence and diversity. The linear prediction strategy has an advantage in solving problems with regular changes but has a limited advantage in problems with complex changes. The reason is that relying on the linear prediction strategy solely can result in a loss of population diversity. Most algorithms utilize a mutation operator or randomly generate individuals to enhance population diversity. In VSDPS, the methods mentioned above are not utilized. However, population diversity is maintained when making predictions. After detecting the environmental change, VSDPS is used to track the new $\mathrm{PS}_t/\mathrm{PF}_t$. Firstly, the stepsize of the solutions will be calculated. The centroid of the population determines the stepsize of the nondominated solutions, while the centroids of the subpopulations after clustering determine the stepsize of the dominated solutions. Then, the dual prediction strategies, which combine an improved linear prediction strategy with a dynamic particle swarm prediction strategy, are employed to adapt to the change. Within the dual prediction strategies, the improved linear prediction strategy aims to enhance the convergence of the population, while the dynamic particle swarm prediction strategy focuses on preserving the diversity of the population.

At present, many researchers consider the DMOEAs to be a combination of change response and static multi-objective optimizer. DMOEAs using such a two-stage model usually focus on the change response. However, the phase of change response typically lasts only a few generations, and there is no assurance that the diversity of the population can be adequately restored within such a brief period. Many DMOEAs have utilized the decomposition-based framework as a static multi-objective optimizer, among which the most commonly used is MOEA/D-DE, such as an adaptive response mechanism selection strategy (MOEA/D-ARMS) [48], a new prediction strategy using gaussian mixture model (MOEA/D-GMM) [15], and so on. Since MOEA/D-DE only uses the DE/best/1 operator to generate offspring, it may not maintain the diversity effectively. In VSDPS, the MOEA/D-DE is improved by incorporating the GA operator as well as variants of DE operator, aiming to track the current $\mathrm{PS}_t/\mathrm{PF}_t$ more quickly along with DE/rand/1 operator.

3. The proposed VSDPS

3.1. The framework of VSDPS

In DMOEAs, the change response is significant, and a good change response can facilitate the rapid convergence of the population to the $\mathrm{PS}_t/\mathrm{PF}_t$ in the new environment. Additionally, the selection of a static multi-objective optimizer also impacts the convergence speed of the population. A suitable static multi-objective optimizer can effectively maintain both the convergence and diversity of the population. Therefore, VSDPS proposed variable stepsize and dual prediction strategies to track the new $\mathrm{PS}_t/\mathrm{PF}_t$, maintaining the diversity of the population while making predictions. It also enhances the MOEA/D-DE, enabling it to adapt to the new environment and track the $\mathrm{PS}_t/\mathrm{PF}_t$ more rapidly. The overall framework of VSDPS is presented in Algorithm 2.

As shown in line 3 of Algorithm 2, environmental change detection is initially conducted through a re-initialization method. In each iteration, 20% of the population is randomly chosen as detectors to determine if their function values have altered. If altered, it indicates an environmental change. When an environmental change is detected, the response to the change is initiated, which involves variable stepsize and dual prediction strategies. The variable stepsize is utilized to enhance the accuracy of predictions. Nondominated solutions and dominated solutions have different stepsizes. The dual prediction strategies consist of an improved linear prediction strategy and a dynamic particle swarm prediction strategy. The improved linear prediction strategy aims to enhance population convergence, while the dynamic particle swarm prediction strategy is employed to preserve population diversity.

Initially, the variable stepsizes of the solutions are calculated. The stepsize of the nondominated solutions is based on the centroid of the population, while the stepsize of the dominated solutions is based on the centroids of the subpopulations. Then, the population is sorted in ascending order based on the Chebyshev values of the individuals. The top 60% of the population is predicted using the improved linear prediction strategy, while the dynamic particle swarm prediction strategy is applied to the remaining individuals. There have been some enhancements made to MOEA/D-DE, attributed to the influence of the static multi-objective optimizer on the overall performance of the algorithm.

Algorithm 2 The framework of VSDPS

```
Input: Population at time $t$ $P^t$, population at time $t-1$ $P^{t-1}$, number
    of population $N$;
Output: Population at time $t+1$ $P^{t+1}$;
 1: Initialize the population $P^t$, time step $t = 0$;
 2: while the termination criterion is not met do
 3:   if change detected then
 4:     $(\Delta C^t, V)$ =Variable stepsize($P^t$, $P^{t-1}$, $t$);
 5:     $P^{t+1}$ =Dual prediction strategies($P^t$, $\Delta C^t$, $V$, $t$);
 6:     $t = t + 1$;
 7:   else
 8:     Update the population by the improved MOEA/D-DE;
 9:   end if
10: end while
```

3.2. Variable stepsize

The stepsize is crucial in the prediction strategy, with the population centroid as the most commonly used moving trend. The variable stepsize is proposed in VSDPS, with the stepsize of nondominated solutions being related to the population centroid, and the stepsize of dominated solutions being associated with the centroids of the subpopulations after clustering. Algorithm 3 outlines the specific steps of the variable stepsize.

The centroid of the population is widely used to predict population trends and guide the direction of population evolution. Assume that $POS_t = \{x^t\} = \{x_1^t, x_2^t, \ldots, x_n^t\}$ represents all individuals at time t, where $i = 1, 2, 3, \ldots, n$ is the dimension of the decision variable. $|POS_t|$ represents the cardinality of all individuals at time t. The formula for the stepsize vector $V = \{V_1, V_2, \ldots, V_n\}$ of the population at time $t+1$ is as follows:

$$V_i = C_i^t - C_i^{t-1} \tag{6}$$

$$C_i^t = \frac{\sum_{x_t \in POS_t} x_i^t}{|POS_t|} \tag{7}$$

where C^{t-1} is the centroid of the population in the previous environment.

For the dominated solutions, a clustering method is employed. The number of clusters, denoted as K, varies with the intensity of the environmental change to determine the stepsize. This clustering method involves selecting representative individuals and dividing the population into K clusters based on their Euclidean distance to the representative individuals. In order to preserve the diversity of the population, the farthest first selection method [49] is employed to choose representative individuals.

The multidirectional prediction approach for dynamic multiobjective optimization problems (MDP) [26] is used to update the value of K. First, it defines an indicator that reflects the severity of the environmental change, $\delta(t)$, which is represented by the degree of change in the objective function value of N individuals before and after

Algorithm 3 Variable stepsize

```
Input: Population at time $t$ $P^t$, population at time $t-1$ $P^{t-1}$, current time step $t$;
Output: The stepsize of the population at time $t$ $V$, the stepsize of each cluster at time $t$ $\Delta C^t$;
Calculate the degree of the environmental change $\delta$ according to formula (8);
Use the farthest first selection method[49] to select $K$ individuals as representative individuals;
Divide the population into $K$ clusters according to the Euclidean distance between each individual and representative individuals;
Calculate the centroid $C^t = \{C_1^t, C_2^t, \cdots, C_K^t\}$ of each cluster and find the position of the representative individuals at the previous time $C^{t-1} = \{C_1^{t-1}, C_2^{t-1}, \cdots, C_K^{t-1}\}$ from $P_{t-1}$;
Calculate the stepsize of each cluster $\Delta C^t = \{\Delta C_1^t, \Delta C_2^t, \cdots, \Delta C_K^t\}$ using formula (10) and (11);
Calculate the stepsize of the population $V = \{V_1, V_2, \cdots, V_n\}$ using formula (6) and (7);
```

the environmental change. The formula for computing $\delta(t)$ is as follows:

$$\delta(t) = \frac{1}{M}\frac{1}{N}\sum_{j=1}^{M}\sum_{i=1}^{N}|\Delta f_j(i) - \mu_j(g)| \tag{8}$$

where $\Delta f_j(i) = [(f_j(i,g) - f_j(i,g-1))/(u_j(g) - l_j(g))]$ and $\mu_j(i) = (1/N)\sum_{i=1}^{N}|[(f_j(i,g) - f_j(i,g-1))/(u_j(g) - l_j(g))]|$. Here, $f_j(i,g)$ and $(f_j(i,g-1))$ are the j_{th} objective value of individual i in the g_{th} iteration and $(g-1)_{th}$ iteration. $u_j(g)$ and $l_j(g)$ represent the maximum and minimum on the j_{th} objective at the g_{th} iteration, respectively.

According to the severity of the environmental change defined above, the number of representative individuals K can be calculated. The value of K is determined as follows,

$$K = K_1 + \lceil \delta(t) * (K_2 - K_1) \rceil \tag{9}$$

where K_1 and K_2 are the minimum and maximum of K respectively. The sizes of K_1 and K_2 are the same as those set in [26]. They are $K_1 = M + 1$ and $K_2 = 3 * M$ respectively. M is the number of the objective functions. As the objective function changes, the value of K_1 and K_2 will also change. The PS center combines the extreme point of the corresponding PF as the initial representative individuals, that is, the size of the initial representative individuals K is $M + 1$. After that, the value of K will never be less than $M + 1$. A too large or small value of K_2 will both pose a negative influence on the performance of the algorithm. In [26], the value of K_2 is tested, and the experimental results show that the maximum of K is $3 * M$. Since the criterion for assessing environmental changes is based on detecting a change in the objective function values of 20% of the individuals in the population, the calculated $\Delta f_j(i)$ and $u_j(g)$ may also change. Consequently, the $\delta(t)$ and the K value may be affected.

Once the population has been classified into clusters according to the Euclidean distance, the next step is to compute the centroids of each cluster $\Delta C^t = \{\Delta C_1^t, \Delta C_2^t, \ldots, \Delta C_K^t\}$. The formula for calculating the centroids of each cluster is shown below,

$$\Delta C_j = C_j^t - C_j^{t-1} \tag{10}$$

$$C_j^t = \frac{\sum_{x_i \in POS_j^t} x_i^t}{|POS_j^t|} \tag{11}$$

where C_j^t refers to the centroid of j_{th} cluster, C_j^{t-1} represents the position of the representative individual of j_{th} cluster at time $t-1$. POS_j^t represents all the individuals in the j_{th} cluster at time t. $|POS_j^t|$ represents the cardinality of individuals in the j_{th} cluster at time t. Fig. 1 shows the stepsizes of the subpopulations after clustering.

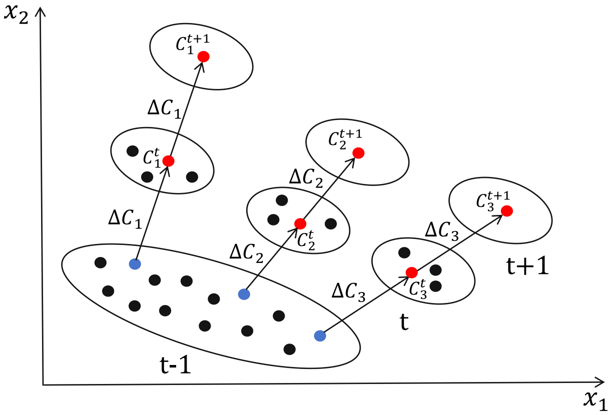

Fig. 1. Take $N = 14$, $M = 2$, $K = 3$ as an example. The red dots represent the centroids of the subpopulations, while the blue dots represent the individuals closest to the centroids of the subpopulations in the population at time $t-1$. The differences between them represent the stepsizes of the subpopulation centroids.

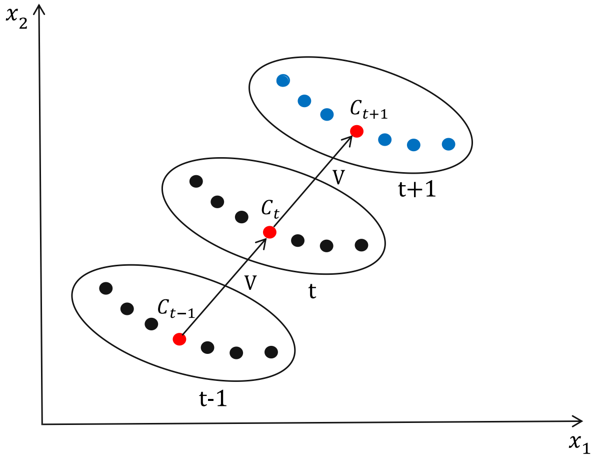

Fig. 2. Take $M = 2$ as an example, the implementation process of the improved linear prediction strategy, where the red dot represents the current population centroid.

3.3. Dual prediction strategies

The pseudo-code of the dual prediction strategies is shown in Algorithm 4. It mainly consists of two parts, one is the improved linear prediction strategy, and the other is the dynamic particle swarm prediction strategy. The population is sorted in ascending order based on the Chebyshev values of the individuals. The top 60% of the population is predicted using the improved linear prediction strategy, while the dynamic particle swarm prediction strategy is applied to the remaining individuals. In this paper, the Chebyshev values are used to represent the fitness values of individuals to distinguish the strengths and weaknesses of individuals. Individuals with low Chebyshev values predicted more accurately, and vice versa.

3.3.1. Improved linear prediction strategy

The linear prediction strategy [17,23] is commonly utilized for solving DMOPs. It typically involves using the difference of population centroids between two adjacent environmental changes to represent the direction of population movement and guide the evolution of the population. The linear prediction strategy provides a rough estimate of environmental change with limited accuracy. To enhance prediction accuracy, an improved linear prediction strategy is proposed, building upon the existing linear prediction strategy, In this new strategy, the stepsize is variable, as detailed in Section 3.2.

Algorithm 4 Dual prediction strategies

```
Input: Population at time $t$ $P^t$, current time step $t$, the stepsize of the
    population at time $t$ $V$, the stepsize of each cluster at time $t$ $\Delta C^t$;
Output: New environment initial population $P^{t+1}$;
$P^{t+1} = \emptyset$;
for $i = 1 : NP$ do
    if $x_i^t$ is ranked lower than 60% of the individuals in $P^t$ then
        if $x_i^t$ is a nondominated solution then
            Calculate $y$ by formula (12);
        else
            Calculate $y$ by formula (13);
        end if
    else
        if $x_i^t$ is a nondominated solution then
            Calculate $y$ by formula (14) (16);
        else
            Calculate $y$ by formula (15) (16);
        end if
    end if
    $P^{t+1} = P^{t+1} \cup y$;
end for
```

If an individual x is a nondominated individual, the evolutionary direction of x in the decision space is determined by the centroid of the population. The prediction process of x is depicted in Fig. 2 and its position at time $t+1$ can be computed by

$$x^{t+1} = x^t + V \tag{12}$$

where $V = \{V_1, V_2, \dots, V_n\}$ is the moving trend of the population, as introduced in formula (6). x^t and x^{t+1} are the solutions at time t and $t+1$ respectively.

If an individual x is a dominated individual, the evolutionary direction vector of the individual in the decision space is determined by the movement trend of the centroid of the subpopulation to which the individual belongs. The number of subpopulations varies with the intensity of the environmental change. The stepsize is calculated by the formula (10), which is discussed in detail in Section 3.2. The formula for the predicted solutions at time $t+1$ is as follows:

$$x^{t+1} = x^t + \Delta C_i \tag{13}$$

where ΔC_i is the moving direction of the ith cluster. x^t and x^{t+1} are the solution at time t and $t+1$.

3.3.2. Dynamic particle swarm prediction strategy

In recent years, PSO has been used as a strategy to solve DMOPs in several papers [35,38,39]. In most cases, multiple particle swarms are used to co-evolve the population. Each particle swarm corresponds to a specific function and optimizes its respective objective function. Instead of utilizing the previously described method, the dynamic particle swarm prediction strategy establishes a connection between PSO and the centroid. There are some changes in the setting of the inertia weight w and the non-negative constants r_1 and r_2 compared to PSO, which will be discussed in Section 4. In the formula of PSO, particle swarm always tracks two extreme values, one is the historical individual near optimal location **Pbest**, and the other is the historical population near optimal location **Gbest**. To align with the characteristics of DMOPs, the dynamic particle swarm prediction strategy links the historical population near optimal location **Pbest** and population historical near optimal position **Gbest** with the direction of population movement. In the following, the selection of **Pbest** and **Gbest** and how they relate to the movement direction of the population will be explained in detail.

First, an introduction to the **Gbest** selection is provided. To select **Gbest**, the nondominated solutions of the most recent moments are stored in an archive, following the first in and first out (FIFO) principle. The number of the nondominated solutions in the archive is determined by the size of the archive. The size of the archive can be found in Section 4.4. The solutions in the archive are sorted using the nondominated sorting. Based on their crowding distance, the individuals with the lowest nondominated rank are further sorted from largest to smallest. Lastly, a single individual is chosen at random from the first half of the sorted individuals is designated as the **Gbest**. **Gbest** needs to be re-selected each time the environmental changes.

Next, the selection of **Pbest** is introduced. The initial stage involves identifying the cluster to which the point is assigned. Then, perform nondominated sorting on the individuals within the cluster. Finally, an individual is randomly selected from the lowest nondominated rank as **Pbest**.

In the dynamic particle swarm prediction strategy, the stepsize of the nondominated and dominated individuals differs, similar to linear prediction. If an individual is a nondominated individual, the formula for calculating the velocity of the individual at the next time step, denoted as Vel^{t+1}, can be expressed as follows:

$$Vel^{t+1} = w * Vel^t + r_1 * (Pbest + V - x^t) + r_2 * (Gbest + V - x^t) \tag{14}$$

where w represents the inertia weight. r_1 and r_2 are the learning factors. V is the moving direction of the population, as introduced in the formula (6).

If an individual is a dominated individual, the velocity Vel^{t+1} of the individual x^{t+1} at the next time is calculated by

$$Vel^{t+1} = w * Vel^t + r_1 * (Pbest + \Delta C_i - x^t) + r_2 * (Gbest + V - x^t) \tag{15}$$

where ΔC_i represents the direction of movement of the cluster to which the individual belongs, computed by the formula (10).

After calculating the next time velocity Vel^{t+1} of the individual, the prediction formula for the position of the individual at the next time $t+1$ is as follows:

$$x^{t+1} = x^t + Vel^{t+1} \tag{16}$$

where x^t is the solution at time period t, x^{t+1} is the predicted solution at time period $t+1$.

3.4. Improved MOEA/D-DE

A static multi-objective optimizer also plays an indispensable role in DMOEAs. A suitable static multi-objective optimizer can enable the algorithm to track the changed $\mathrm{PS}_t/\mathrm{PF}_t$ as soon as possible after the environment changes. An enhanced version of MOEA/D-DE [27], known as improved MOEA/D-DE, is the static multi-objective optimizer utilized in VSDPS.

The pseudo-code of the improved MOEA/D-DE is shown in Algorithm 5. In the improved MOEA/D-DE, the framework of MOEA/D-DE is still utilized, but with some modifications in the operator usage. The only operator used in MOEA/D-DE is DE/rand/1, while in the improved MOEA/D-DE, the DE/current-to-lbest/1, DE/lbest/2, and GA operators are also used. The control number 0.4 in line 4 is based on the experimental results, which can be found in the supplementary material. If the control number is greater than 0.4 (line 4), the DE/rand/1 operator will be used to generate a new individual; otherwise, other operators will be used. The DE/rand/1 operator is an evolutionary operator known for its powerful search capability. The conditions in lines 7, 9 and 11 take the remainder of 3 and then select the corresponding operator. These three operators are equal in position and interchangeable, with each having its own characteristics. If the result is 0 (line 7), the GA operator will be used to generate the new individual. GA operator generates offspring by utilizing a polynomial mutation operator along with simulated binary crossover [50], resulting in good diversity and a powerful local search capability. If the result is 1 (line 9), the DE/lbest/2 operator is used to generate the new individual. DE/lbest/2 operator, which exhibits

Algorithm 5 Improved MOEA/D-DE

```
Input: Population at time $t$ $P^t$, weight vector set $\lambda$, ideal point $z^*$,
    environmental change frequency $\tau_t$, static optimization times $t_s$;
Output: Evolutionary population of the next generation $P^t_{g+1}$, static
    optimization times $t_s$;
 1: Determine the neighbors of each weight vector in the set of weight
    vectors $\lambda$ , which are stored in the set $B$;
 2: for $i = 1 : |NP|$ do
 3:   $P = B(i)$;
 4:   if $rand > 0.4$ then
 5:     The individual $X_i$ and a randomly selected individual from the
        neighbor $B(i)$ is used to generate the individual $y$ using the
        DE/rand/1 operator;
 6:   else
 7:     if $mod(i, 3) == 0$ then
 8:       The individual $X_i$ and a randomly selected individual from
          the neighbor $B(i)$ are used to generate the individual $y$ using
          the GA operator;
 9:     else if $mod(i, 3) == 1$ then
10:       The individual $X_i$ with its neighbor $B(i)$ best individual and
          two randomly selected individuals are used to generate the
          individual $y$ using the DE/lbest/2 operator;
11:     else
12:       The individual $X_i$ with its neighbor $B(i)$ best individual and
          two randomly selected individuals are used to generate the
          individual $y$ using the DE/current-to-lbest/1 operator;
13:     end if
14:   end if
15:   Boundary check and boundary correction for y;
16:   The ideal point $z^*$ is updated using the objective value $FV_y$ of $y$
      ;
17:   if $g^{te}\left(y \mid \lambda_i, z^*\right) < g^{te}\left(x_i \mid \lambda_i, z^*\right)$ then
18:     In the population $P^t$, the $x_i = y, FV_i = FV_y$;
19:   end if
20: end for
21: $P^t_{g+1} = P^t$;
22: $t_s = t_s + 1$;
```

good convergence, is an improved version of DE. If the result is 2 (line 11), the DE/current-to-lbest/1 operator is used to generate the new individual. DE/current-to-lbest/1 is primarily utilized to balance population diversity and convergence. The DE/current-to-lbest/1 operator is implemented as follows:

$$V_i = X_i + F \cdot (X_{lbest,B_i} - X_i) + F \cdot (X_{r_1,B_i} - X_{r_2,B_i}) \tag{17}$$

where V_i represents the offspring, X_i denotes the parent individual, X_{lbest,B_i} denotes the locally near optimal individual within X_i neighbors, X_{r_1,B_i} and X_{r_2,B_i} are the random individuals within X_i neighbors, respectively. F is the scale factor, which is set to 0.5.

The DE/lbest/2 operator is implemented as follows:

$$V_i = X_{lbest,B_i} + F \cdot (X_{r_1,B_i} - X_{r_2,B_i}) + F \cdot (X_{r_3,B_i} - X_{r_4,B_i}) \tag{18}$$

where V_i represents the offspring, X_{lbest,B_i} denotes the locally near optimal individual within X_i neighbors, X_{r_1,B_i}, X_{r_2,B_i}, X_{r_3,B_i} and X_{r_4,B_i} are the random individuals within X_i neighbors, respectively. F is the scale factor, which is set to 0.5.

4. Experimental setup

This section provides an overview of the test instances, comparison algorithms, performance metrics, and parameter settings utilized in the experiments.

4.1. Test instances

The proposed algorithm was tested on four benchmark suites, including the FDA [37] benchmark suite, dMOP [51] benchmark suite, F [19] benchmark suite, and DF [52] benchmark suite. The modified hypervolume difference (MHV) [53] and the modified inverted generational distance (MIGD) [19] are utilized to evaluate the performance of VSDPS and the comparison algorithms. The FDA, dMOP, and F are classic benchmark suites in DMOPs, which include Type I, II, and III. The FDA benchmark suite is commonly used to evaluate DMOEAs and has been expanded to create the dMOP benchmark suite. The F benchmark suite considers the nonlinear linkage between the decision variables, where dynamic changes are sharp and irregular. The DF benchmark suite was presented in CEC2018 [52], containing multiple complex environmental changes, such as time-varying PS$_t$/PF$_t$ flow shapes, irregular PF shapes, disconnections, and more. These challenges make it more challenging for DMOEAs.

4.2. Comparison algorithms

Six state-of-the-art algorithms are employed as comparison algorithms, which are MOEA/D-KF [18], SGEA [54], MOEA/D-FD [17], MOEA/D-DM [55], MOEA/D-IEC [56], and MOEA/D-ARMS [48]. A brief introduction to the comparison algorithms is described as follows.

(1) MOEA/D-KF: It uses Kalman filter prediction in decision space to estimate the population in the new environment. Additionally, a random re-initialization technique is utilized to enhance the overall performance of the algorithm.

(2) SGEA: It uses a steady-state and generational evolutionary algorithm to track the changing PS$_t$/PF$_t$ in the dynamic environment. When the environmental change is detected, two different methods are used to re-initialize the population. To reduce population loss, some outdated solutions with good distribution are being reused. Other solutions are initiated using the prediction approach, aiming to relocate multiple solutions around the new PS$_t$/PF$_t$.

(3) MOEA/D-FD: It forecasts the new locations of a certain number of Pareto-optimal solutions using a first-order difference model. In addition, a portion of the existing Pareto-optimal solutions are kept for the new population to preserve diversity.

(4) MOEA/D-DM: It constructs a differential model using the recent history of centroid positions to predict the population in the new environment. The projected solutions, along with some conserved solutions, form a new population.

(5) MOEA/D-IEC: It employs a U-test mechanism to evaluate decision variables and subsequently categorizes each individual into two parts, which are macro changing decision and micro changing decision. This mechanism establishes an effective information exchange in a dynamic environment to generate a good response.

(6) MOEA/D-ARMS: It can dynamically choose the most effective response mechanism from a pool of four representative options based on their recent performance. An evaluation strategy is implemented to reward the response mechanism. Additionally, a probability-based method is used to determine the appropriate response mechanism for generating a new solution.

4.3. Performance metrics

Performance evaluation indicators play an important role in assessing the performance of DMOEAs. Currently, a wide range of evaluation indicators have been proposed to assess algorithm performance from different perspectives. The evaluation metrics used in VSDPS are mean hypervolume (MHV) and mean inverted generational distance (MIGD).

Table 1
The statistics of the MIGD values obtained by VSDPS and the six other state-of-the-art algorithms were run 30 times independently on FDA, dMOP, and F test suites. The final two rows display the results of the Wilcoxon rank sum test and the Friedman test, respectively, with the best results highlighted.

Problem	$(n_t,\ \tau_t)$	MOEA/D-KF	SGEA	MOEA/D-FD	MOEA/D-DM	MOEA/D-IEC	MOEA/D-ARMS	VSDPS
FDA1	(10,10)	1.5538e−2 (1.52e−3) −	1.5690e−2 (2.73e−3) −	1.0138e−2 (5.04e−4) −	1.2784e−2 (8.46e−4) −	7.9339e−3 (5.31e−4) −	1.0167e−2 (5.76e−4) −	6.7156e−3 (3.52e−4)
	(10,20)	6.4134e−3 (1.86e−4) −	9.7410e−3 (1.90e−3) −	6.0139e−3 (1.64e−4) −	6.3922e−3 (1.83e−4) −	5.2603e−3 (1.49e−4) −	6.0033e−3 (1.29e−4) −	4.8273e−3 (9.14e−5)
	(10,30)	5.0587e−3 (1.09e−4) −	8.0176e−3 (1.83e−3) −	5.0178e−3 (8.34e−5) −	5.1870e−3 (1.24e−4) −	4.6852e−3 (8.61e−5) −	4.9967e−3 (5.70e−5) −	4.2604e−3 (2.63e−5)
FDA2	(10,10)	1.1455e−2 (1.26e−3) −	1.0314e−2 (1.16e−3) −	8.5453e−3 (5.97e−4) −	6.3565e−3 (2.28e−4) −	5.4648e−3 (8.08e−5) +	6.2787e−3 (1.32e−4) −	5.6774e−3 (9.38e−5)
	(10,20)	5.6189e−3 (8.78e−5) −	7.4634e−3 (7.46e−4) −	5.7063e−3 (2.67e−4) −	5.0014e−3 (6.95e−5) −	4.7000e−3 (3.70e−5) +	4.9467e−3 (3.30e−5) −	4.7736e−3 (6.37e−5)
	(10,30)	4.7889e−3 (5.16e−5) −	6.4986e−3 (5.28e−4) −	4.8135e−3 (9.09e−5) −	4.6197e−3 (5.46e−5) −	4.4403e−3 (2.47e−5) −	4.5804e−3 (2.32e−5) −	4.4253e−3 (1.82e−5)
FDA3	(10,10)	7.7660e−2 (4.24e−3) −	4.8174e−2 (5.53e−3) +	4.3006e−2 (2.90e−3) +	4.9055e−2 (2.37e−3) +	5.7025e−2 (3.77e−3) −	3.2405e−2 (3.42e−3) +	5.1645e−2 (1.97e−3)
	(10,20)	4.3615e−2 (1.51e−3) −	3.9345e−2 (3.59e−3) −	2.0421e−2 (1.54e−3) +	2.1454e−2 (1.92e−3) +	3.7868e−2 (2.28e−3) −	1.4248e−2 (1.44e−3) +	3.0933e−2 (2.19e−3)
	(10,30)	3.0894e−2 (9.80e−4) −	3.6803e−2 (3.79e−3) −	1.3254e−2 (1.24e−3) +	1.3713e−2 (1.23e−3) +	2.9269e−2 (2.09e−3) −	9.7966e−3 (5.80e−4) +	2.3974e−2 (1.88e−3)
FDA4	(10,10)	9.8523e−2 (1.55e−3) −	1.2093e−1 (6.36e−3) −	9.6714e−2 (3.61e−3) −	1.0006e−1 (3.79e−3) −	8.3563e−2 (9.61e−4) =	8.6407e−2 (1.04e−3) −	8.3816e−2 (1.13e−3)
	(10,20)	8.3249e−2 (7.32e−4) −	8.8334e−2 (2.43e−3) −	8.3287e−2 (8.38e−4) −	8.4352e−2 (1.13e−3) −	7.8218e−2 (3.26e−4) −	7.9121e−2 (4.05e−4) −	7.7464e−2 (3.61e−4)
	(10,30)	7.9357e−2 (2.98e−4) −	8.1170e−2 (9.67e−4) −	8.0392e−2 (5.58e−4) −	8.0616e−2 (6.39e−4) −	7.6916e−2 (2.29e−4) −	7.7380e−2 (3.20e−4) −	7.6130e−2 (2.56e−4)
FDA5	(10,10)	3.0296e−1 (2.27e−2) −	1.8952e−1 (5.97e−3) +	2.9616e−1 (5.43e−2) −	3.2223e−1 (3.71e−2) −	2.6876e−1 (3.10e−2) −	2.3902e−1 (3.31e−2) =	2.2518e−1 (1.28e−2)
	(10,20)	2.4665e−1 (3.04e−2) −	1.4239e−1 (3.40e−3) +	2.2992e−1 (3.89e−2) −	2.6172e−1 (3.55e−2) −	2.2711e−1 (3.97e−2) −	2.0109e−1 (3.04e−2) −	1.6716e−1 (7.63e−3)
	(10,30)	2.2009e−1 (2.62e−2) −	1.3165e−1 (2.30e−3) +	2.0739e−1 (2.91e−2) −	2.4264e−1 (3.39e−2) −	2.1231e−1 (3.83e−2) −	1.7749e−1 (3.42e−2) −	1.5108e−1 (6.00e−3)
DMOP1	(10,10)	2.5124e−2 (2.10e−3) −	4.4182e−2 (3.15e−2) −	4.8688e−3 (1.15e−4) −	5.0586e−3 (1.45e−4) −	5.1025e−3 (1.42e−4) −	6.1838e−3 (1.63e−4) −	4.6848e−3 (9.43e−5)
	(10,20)	7.9683e−3 (3.62e−4) −	3.0405e−2 (1.78e−2) −	4.2518e−3 (6.38e−5) −	4.3618e−3 (7.86e−5) −	4.2802e−3 (5.64e−5) −	4.7178e−3 (8.32e−5) −	4.0159e−3 (2.26e−5)
	(10,30)	5.5618e−3 (1.05e−4) −	3.1360e−2 (2.72e−2) −	4.0242e−3 (3.59e−5) −	4.0890e−3 (3.19e−5) −	4.0187e−3 (2.12e−5) −	4.2796e−3 (4.36e−5) −	3.9025e−3 (1.35e−5)
DMOP2	(10,10)	1.6787e−2 (1.22e−3) −	1.7655e−2 (2.93e−3) −	1.1695e−2 (7.53e−4) −	1.4467e−2 (1.29e−3) −	8.8634e−3 (4.45e−4) −	1.2081e−2 (6.78e−4) −	7.8703e−3 (4.76e−4)
	(10,20)	6.6571e−3 (2.04e−4) −	1.0046e−2 (1.21e−3) −	6.4262e−3 (2.07e−4) −	6.8655e−3 (2.77e−4) −	5.4557e−3 (1.15e−4) −	6.4373e−3 (1.59e−4) −	4.9971e−3 (7.93e−5)
	(10,30)	5.0651e−3 (7.94e−5) −	8.3391e−3 (1.53e−3) −	5.1738e−3 (9.32e−5) −	5.3352e−3 (1.43e−4) −	4.6874e−3 (4.75e−5) −	5.1242e−3 (6.47e−5) −	4.2618e−3 (3.68e−5)
DMOP3	(10,10)	1.3806e−2 (1.20e−3) −	1.7157e−2 (1.49e−3) −	1.1951e−2 (1.05e−3) −	1.5002e−2 (1.88e−3) −	7.1848e−3 (4.10e−4) =	8.1205e−3 (3.28e−4) −	7.3556e−3 (4.64e−4)
	(10,20)	6.6121e−3 (1.71e−4) −	8.9177e−3 (9.56e−4) −	6.0942e−3 (2.15e−4) −	6.5390e−3 (3.05e−4) −	5.0915e−3 (1.12e−4) −	5.3833e−3 (1.01e−4) −	4.9016e−3 (9.46e−5)
	(10,30)	5.1244e−3 (8.98e−5) −	6.9018e−3 (4.67e−4) −	4.9963e−3 (1.06e−4) −	5.1254e−3 (1.08e−4) −	4.6373e−3 (9.04e−5) −	4.7084e−3 (5.52e−5) −	4.3227e−3 (3.85e−5)
F5	(10,10)	1.0791e+0 (5.34e−2) −	1.2980e+0 (2.01e−1) −	1.2151e+0 (2.07e−1) −	1.0176e+0 (5.74e−2) −	1.1191e+0 (9.02e−2) −	9.5468e−1 (2.19e−2) −	9.1923e−1 (1.87e−2)
	(10,20)	9.1161e−1 (1.00e−2) −	9.9107e−1 (6.11e−2) −	9.3024e−1 (1.65e−2) −	9.0509e−1 (8.77e−3) −	8.8864e−1 (7.71e−3) =	8.9242e−1 (7.20e−3) −	8.8990e−1 (8.92e−3)
	(10,30)	8.8735e−1 (6.27e−3) −	9.5351e−1 (1.88e−2) −	8.9863e−1 (8.01e−3) −	8.8687e−1 (5.54e−3) −	8.6576e−1 (2.03e−3) +	8.8343e−1 (2.58e−3) −	8.8117e−1 (2.80e−3)
F6	(10,10)	8.6631e−1 (8.35e−2) −	6.8166e−1 (1.07e−2) −	6.9023e−1 (2.31e−2) −	6.5396e−1 (1.28e−2) −	6.6697e−1 (2.14e−2) −	6.3874e−1 (8.94e−3) −	6.1779e−1 (5.50e−3)
	(10,20)	6.4218e−1 (2.14e−2) −	6.6492e−1 (8.16e−3) −	6.2760e−1 (6.30e−3) −	6.2615e−1 (4.84e−3) −	5.9331e−1 (3.49e−3) +	6.1497e−1 (3.57e−3) −	6.0999e−1 (3.80e−3)
	(10,30)	6.1657e−1 (4.55e−3) =	6.6237e−1 (5.51e−3) −	6.1942e−1 (4.92e−3) −	6.2014e−1 (4.65e−3) −	5.8301e−1 (2.25e−3) +	6.1324e−1 (2.99e−3) =	6.1377e−1 (3.93e−3)
F7	(10,10)	8.0797e−1 (2.15e−2) −	7.3825e−1 (1.03e−2) −	7.6510e−1 (2.37e−2) −	7.7019e−1 (3.43e−2) −	7.5861e−1 (1.12e−2) −	7.3299e−1 (6.22e−3) −	7.1969e−1 (4.03e−3)
	(10,20)	7.1600e−1 (3.00e−3) −	7.1646e−1 (3.72e−3) −	7.1570e−1 (5.04e−3) −	7.1325e−1 (2.86e−3) −	7.0580e−1 (1.79e−3) =	7.0767e−1 (2.63e−3) −	7.0449e−1 (2.29e−3)
	(10,30)	7.0627e−1 (2.38e−3) −	7.1258e−1 (4.10e−3) −	7.0773e−1 (2.97e−3) −	7.0539e−1 (3.02e−3) −	6.9766e−1 (1.47e−3) +	7.0241e−1 (2.33e−3) =	7.0109e−1 (2.46e−3)
F8	(10,10)	1.5100e−1 (8.18e−3) −	2.0300e−1 (1.42e−2) −	9.8090e−2 (2.16e−3) −	1.0126e−1 (2.49e−3) −	8.9033e−2 (1.26e−3) +	9.9768e−2 (1.47e−3) −	9.1328e−2 (1.79e−3)
	(10,20)	9.6025e−2 (1.96e−3) −	1.8359e−1 (1.13e−2) −	8.6281e−2 (1.36e−3) −	8.7517e−2 (1.27e−3) −	8.0614e−2 (6.90e−4) +	8.4656e−2 (7.01e−4) −	8.2221e−2 (6.29e−4)
	(10,30)	8.5757e−2 (8.97e−4) −	1.7106e−1 (9.84e−3) −	8.2321e−2 (7.73e−4) −	8.3562e−2 (8.68e−4) −	7.8389e−2 (3.82e−4) =	8.0522e−2 (4.34e−4) −	7.8685e−2 (4.94e−4)
F9	(10,10)	7.7423e−1 (1.33e−1) −	1.5724e+0 (1.05e−1) −	1.4669e+0 (4.13e−1) −	7.0078e−1 (2.88e−1) =	1.0740e+0 (3.49e−1) −	3.9016e−1 (2.19e−2) +	6.1206e−1 (2.08e−1)
	(10,20)	4.4141e−1 (5.36e−2) −	1.2065e+0 (1.87e−1) −	7.7307e−1 (3.14e−1) −	3.5400e−1 (3.42e−2) +	4.6694e−1 (7.61e−2) −	3.2029e−1 (7.18e−3) +	3.9462e−1 (6.06e−2)
	(10,30)	3.4169e−1 (3.79e−2) =	8.3615e−1 (1.92e−1) −	4.6294e−1 (1.28e−1) −	3.2163e−1 (9.20e−3) +	3.5629e−1 (4.10e−2) =	3.1215e−1 (3.47e−3) +	3.4303e−1 (2.65e−2)
F10	(10,10)	2.0579e+0 (1.04e−1) −	2.0209e+0 (1.29e−1) −	2.3709e+0 (2.04e−1) −	2.0302e+0 (1.09e−1) −	2.1803e+0 (1.53e−1) −	1.8722e+0 (3.93e−2) −	1.8262e+0 (3.35e−2)
	(10,20)	1.7772e+0 (2.07e−2) +	1.8362e+0 (4.23e−2) −	1.8540e+0 (5.01e−2) −	1.7778e+0 (2.92e−2) +	1.7867e+0 (2.24e−2) =	1.7358e+0 (1.08e−2) +	1.7949e+0 (2.59e−2)
	(10,30)	1.7355e+0 (1.52e−2) −	1.8090e+0 (7.96e−2) −	1.7473e+0 (1.98e−2) −	1.7306e+0 (1.31e−2) −	1.7312e+0 (1.85e−2) −	1.7110e+0 (6.36e−3) −	1.7066e+0 (8.19e−3)
+/ − / =		1/39/2	4/38/0	3/39/0	6/35/1	8/27/7	7/32/3	
Average ranking		5.31	6.14	4.56	4.58	2.77	2.77	1.86

4.3.1. Mean inverted generational distance (MIGD)

Inverted generational distance (IGD) [57] is a comprehensive performance evaluation indicator. It primarily assesses the performance of the algorithm by calculating the minimum distance sum between the approximate PF$_t$ and the true PF$_t$ obtained by the algorithm. The IGD metric is defined as follows:

$$IGD\left(P_t, P_t^*\right) = \frac{\sum_{v \in P_t^*} d(v, P_t)}{\left|P_t^*\right|} \tag{19}$$

where P$_t^*$ is a collection of Pareto-optimal solutions obtained by a uniform sampling at time t, P$_t$ is collection of solutions approximate to P$_t^*$ obtained by the algorithm, d(v, P$_t$) denotes the minimum Euclidean distance between a point v and points in P$_t^*$, and |PF$_t$| is the cardinality of PF$_t$. However, IGD cannot measure the performance of DMOEAs at a run. Therefore, mean inverted generational distance (MIGD) [53] is proposed,

$$MIGD = \frac{\sum_{t \in T} IGD\left(P_t, P_t^*\right)}{|T|} \tag{20}$$

herein, $t \in T = \{1, 2, \ldots\}$ is a set of discrete time points in a run, and $|T|$ denotes the cardinality of T. The smaller the MIGD and IGD, the better the convergence and distribution of the algorithm.

4.3.2. Mean hypervolume (MHV)

The hypervolume (HV) [58] is also a comprehensive performance evaluation indicator. It indicates the hypervolume of an area dominated by the obtained PF$_t$, and it is defined as follows:

$$HV = Volume(\cup_{i=1}^{|PF_t|} V_i) \tag{21}$$

where PF$_t$ represents the leading edge of the Pareto-optimal solution obtained by the algorithm at timestep t, and V_i is the super volume formed between individuals on PF$_t$ and the reference point. Mean hypervolume (MHV) [19] is upon on the improvement of hypervolume (HV), and its mean value can evaluate the performance of the algorithm more accurately. MHV is defined as the average of the HV values in a single run. The formula for MHV is as follows:

$$MHV = \frac{\sum_{t \in T} HV(i, V_i)}{|T|} \tag{22}$$

where T represents a set of discrete time points in a run. The larger the MHV value, the better the algorithm performs.

4.4. General parameter settings

In order to ensure the fairness of the experimental procedure, the parameter settings of the comparison algorithms used in the experiments are based on the original paper. The following are some critical parameters for the experiment.

(1) For all test instances, the population size N is set to 100, and the dimension D of the decision variable is set to 10.
(2) For environmental parameters, the severity of change n_t is set to 10, which is commonly used in the experiments of DMOEAs. The frequency of change τ_t is set to 10, 20, and 30, indicating high, medium, and low environmental change frequency, respectively.
(3) The total number of iterations for all algorithms is set to 30τ_t + 50, where no environmental change occurs in the initial 50 iterations, ensuring that at least 30 environmental changes occur during a single run.
(4) All the algorithms in the experiment independently run 30 times on the test instances to obtain the results.

Table 2
The statistics of the MHV values obtained by VSDPS and the six other state-of-the-art algorithms were run 30 times independently on FDA, dMOP, and F test suites. The final two rows display the results of the Wilcoxon rank sum test and the Friedman test, respectively, with the best results highlighted.

Problem	$(n_t,\ \tau_t)$	MOEA/D-KF	SGEA	MOEA/D-FD	MOEA/D-DM	MOEA/D-IEC	MOEA/D-ARMS	VSDPS
FDA1	(10,10)	7.0154e–1 (1.88e–3) –	7.0275e–1 (1.74e–3) –	7.0944e–1 (7.45e–4) –	7.0542e–1 (1.29e–3) –	7.1276e–1 (7.81e–4) –	7.0935e–1 (7.86e–4) –	7.1494e–1 (5.06e–4)
	(10,20)	7.1496e–1 (2.88e–4) –	7.1168e–1 (1.22e–3) –	7.1574e–1 (2.41e–4) –	7.1507e–1 (3.00e–4) –	7.1691e–1 (2.61e–4) –	7.1573e–1 (1.97e–4) –	7.1784e–1 (1.51e–4)
	(10,30)	7.1727e–1 (1.76e–4) –	7.1443e–1 (1.18e–3) –	7.1742e–1 (1.30e–4) –	7.1705e–1 (2.17e–4) –	7.1794e–1 (1.61e–4) –	7.1740e–1 (8.91e–5) –	7.1899e–1 (4.63e–5)
FDA2	(10,10)	5.2084e–1 (1.00e–3) –	5.2280e–1 (1.08e–3) –	5.2403e–1 (8.24e–4) –	5.2691e–1 (3.43e–4) –	5.2854e–1 (1.37e–4) +	5.2721e–1 (1.83e–4) –	5.2831e–1 (1.57e–4)
	(10,20)	5.2824e–1 (1.38e–4) –	5.2714e–1 (6.42e–4) –	5.2822e–1 (3.62e–4) –	5.2916e–1 (1.37e–4) –	5.2986e–1 (6.86e–5) +	5.2936e–1 (6.71e–5) –	5.2980e–1 (1.08e–4)
	(10,30)	5.2963e–1 (7.20e–5) –	5.2858e–1 (4.10e–4) –	5.2964e–1 (1.42e–4) –	5.2993e–1 (1.17e–4) –	5.3040e–1 (5.49e–5) –	5.3008e–1 (5.15e–5) –	5.3049e–1 (4.23e–5)
FDA3	(10,10)	5.1540e–1 (4.54e–3) –	5.4893e–1 (4.28e–3) +	5.5048e–1 (3.26e–3) +	5.4379e–1 (2.71e–3) –	5.3930e–1 (3.66e–3) –	5.6163e–1 (3.52e–3) +	5.4663e–1 (1.98e–3)
	(10,20)	5.5265e–1 (1.40e–3) –	5.5864e–1 (3.17e–3) –	5.7465e–1 (1.54e–3) +	5.7355e–1 (2.09e–3) +	5.6007e–1 (2.02e–3) –	5.8118e–1 (1.43e–3) +	5.6684e–1 (1.98e–3)
	(10,30)	5.6540e–1 (8.25e–4) –	5.6169e–1 (3.33e–3) –	5.8206e–1 (1.42e–3) +	5.8171e–1 (1.28e–3) +	5.6833e–1 (1.83e–3) –	5.8613e–1 (5.79e–4) +	5.7359e–1 (1.85e–3)
FDA4	(10,10)	4.7384e–1 (3.53e–3) –	4.5243e–1 (1.14e–2) –	4.8292e–1 (6.78e–3) –	4.7482e–1 (8.10e–3) –	5.0707e–1 (2.17e–3) –	5.0264e–1 (2.38e–3) –	5.1489e–1 (2.40e–3)
	(10,20)	5.0577e–1 (1.88e–3) –	5.2258e–1 (2.65e–3) –	5.1250e–1 (1.87e–3) –	5.0936e–1 (2.13e–3) –	5.1943e–1 (1.22e–3) –	5.1679e–1 (1.02e–3) –	5.2521e–1 (7.46e–4)
	(10,30)	5.1558e–1 (8.63e–4) –	5.3903e–1 (1.08e–3) +	5.1970e–1 (1.09e–3) –	5.1944e–1 (1.28e–3) –	5.2315e–1 (5.95e–4) –	5.2167e–1 (6.63e–4) –	5.2970e–1 (4.14e–4)
FDA5	(10,10)	4.6163e–1 (7.03e–3) –	4.5760e–1 (7.58e–3) –	4.6255e–1 (1.41e–2) –	4.5287e–1 (9.90e–3) –	4.9086e–1 (6.77e–3) –	4.9468e–1 (7.72e–3) –	5.0730e–1 (2.95e–3)
	(10,20)	4.9693e–1 (6.36e–3) –	5.2389e–1 (2.52e–3) –	4.9987e–1 (8.52e–3) –	4.9148e–1 (7.24e–3) –	5.1009e–1 (7.96e–3) –	5.1332e–1 (6.04e–3) –	5.2671e–1 (1.24e–3)
	(10,30)	5.0845e–1 (5.00e–3) –	5.3973e–1 (1.86e–3) +	5.1019e–1 (6.23e–3) –	5.0264e–1 (6.77e–3) –	5.1494e–1 (7.09e–3) –	5.1978e–1 (6.55e–3) –	5.3180e–1 (8.04e–4)
DMOP1	(10,10)	4.9127e–1 (3.38e–3) –	4.8626e–1 (2.99e–2) –	5.2497e–1 (2.49e–4) –	5.2454e–1 (3.16e–4) –	5.2461e–1 (2.61e–4) –	5.2242e–1 (3.24e–4) –	5.2546e–1 (1.99e–4)
	(10,20)	5.1884e–1 (6.57e–4) –	4.9985e–1 (1.75e–2) –	5.2619e–1 (1.41e–4) –	5.2590e–1 (1.85e–4) –	5.2615e–1 (1.35e–4) –	5.2508e–1 (1.81e–4) –	5.2670e–1 (6.78e–5)
	(10,30)	5.2322e–1 (1.99e–4) –	4.9923e–1 (2.59e–2) –	5.2671e–1 (9.22e–5) –	5.2652e–1 (6.96e–5) –	5.2674e–1 (4.25e–5) –	5.2598e–1 (1.13e–4) –	5.2707e–1 (4.38e–5)
DMOP2	(10,10)	5.0490e–1 (1.87e–3) –	5.0592e–1 (2.57e–3) –	5.1350e–1 (1.12e–3) –	5.0927e–1 (1.99e–3) –	5.1820e–1 (6.95e–4) –	5.1258e–1 (1.09e–3) –	5.1984e–1 (7.79e–4)
	(10,20)	5.2136e–1 (3.82e–4) –	5.1777e–1 (1.50e–3) –	5.2200e–1 (3.19e–4) –	5.2120e–1 (4.46e–4) –	5.2375e–1 (1.93e–4) –	5.2192e–1 (2.88e–4) –	5.2460e–1 (1.54e–4)
	(10,30)	5.2428e–1 (1.51e–4) –	5.2084e–1 (1.46e–3) –	5.2412e–1 (1.66e–4) –	5.2380e–1 (2.73e–4) –	5.2511e–1 (9.83e–5) –	5.2426e–1 (1.26e–4) –	5.2611e–1 (8.05e–5)
DMOP3	(10,10)	7.0386e–1 (1.62e–3) –	7.0041e–1 (1.78e–3) –	7.0680e–1 (1.57e–3) –	7.0226e–1 (2.74e–3) –	7.1388e–1 (6.27e–4) =	7.1243e–1 (4.95e–4) –	7.1393e–1 (7.14e–4)
	(10,20)	7.1467e–1 (2.55e–4) –	7.1250e–1 (7.54e–4) –	7.1569e–1 (3.36e–4) –	7.1489e–1 (4.84e–4) –	7.1724e–1 (1.84e–4) –	7.1678e–1 (1.56e–4) –	7.1773e–1 (1.50e–4)
	(10,30)	7.1717e–1 (1.52e–4) –	7.1539e–1 (3.71e–4) –	7.1754e–1 (1.63e–4) –	7.1723e–1 (1.92e–4) –	7.1807e–1 (1.66e–4) –	7.1797e–1 (9.88e–5) –	7.1885e–1 (7.28e–5)
F5	(10,10)	2.3881e–1 (1.37e–2) –	2.1881e–1 (1.50e–2) –	1.9994e–1 (4.21e–2) –	2.6330e–1 (1.87e–2) –	2.1064e–1 (2.35e–2) –	2.7622e–1 (9.44e–3) –	2.9951e–1 (9.14e–3)
	(10,20)	3.0711e–1 (5.99e–3) –	2.6676e–1 (1.10e–2) –	3.0452e–1 (6.93e–3) –	3.1347e–1 (5.76e–3) –	3.0958e–1 (3.88e–3) –	3.1949e–1 (4.31e–3) =	3.2076e–1 (4.57e–3)
	(10,30)	3.2509e–1 (3.01e–3) –	2.7869e–1 (9.60e–3) –	3.2061e–1 (5.57e–3) –	3.2622e–1 (3.89e–3) –	3.2535e–1 (1.21e–3) –	3.2987e–1 (9.96e–4) –	3.3267e–1 (7.80e–4)
F6	(10,10)	1.7460e–1 (1.42e–2) –	2.1980e–1 (5.70e–3) –	1.9604e–1 (1.83e–2) –	2.3127e–1 (8.25e–3) –	2.0602e–1 (1.23e–2) –	2.3698e–1 (3.52e–3) –	2.5488e–1 (3.09e–3)
	(10,20)	2.4810e–1 (5.77e–3) –	2.3524e–1 (2.84e–3) –	2.5111e–1 (4.60e–3) –	2.5820e–1 (2.14e–3) –	2.5542e–1 (2.70e–3) –	2.6054e–1 (1.03e–3) –	2.6234e–1 (9.53e–4)
	(10,30)	2.6181e–1 (1.53e–3) –	2.3995e–1 (2.19e–3) –	2.6168e–1 (1.16e–3) –	2.6336e–1 (1.35e–3) –	2.6327e–1 (6.35e–4) –	2.6504e–1 (8.13e–4) –	2.6689e–1 (4.77e–4)
F7	(10,10)	1.6630e–1 (1.25e–2) –	2.1439e–1 (3.46e–3) –	2.0498e–1 (6.10e–3) –	2.0913e–1 (4.76e–3) –	2.0617e–1 (5.66e–3) –	2.1525e–1 (2.39e–3) –	2.2793e–1 (1.75e–3)
	(10,20)	2.2681e–1 (1.32e–3) –	2.2849e–1 (1.97e–3) –	2.3152e–1 (1.15e–3) –	2.3282e–1 (8.70e–4) –	2.3257e–1 (1.23e–3) –	2.3507e–1 (5.39e–4) –	2.3682e–1 (5.09e–4)
	(10,30)	2.3637e–1 (4.02e–4) –	2.3226e–1 (1.99e–3) –	2.3716e–1 (4.59e–4) –	2.3737e–1 (6.75e–4) –	2.3814e–1 (5.14e–4) –	2.3917e–1 (2.00e–4) –	2.4078e–1 (1.64e–4)
F8	(10,10)	3.7741e–1 (1.00e–2) –	3.7544e–1 (1.18e–2) –	4.7671e–1 (4.48e–3) –	4.6772e–1 (4.87e–3) –	4.8820e–1 (2.40e–3) –	4.7088e–1 (2.43e–3) –	4.9207e–1 (3.13e–3)
	(10,20)	4.7166e–1 (3.77e–3) –	4.0911e–1 (1.12e–2) –	5.0136e–1 (1.94e–3) –	4.9727e–1 (2.07e–3) –	5.0633e–1 (1.73e–3) +	4.9888e–1 (1.76e–3) –	5.0517e–1 (1.73e–3)
	(10,30)	4.9316e–1 (1.95e–3) –	4.2777e–1 (7.70e–3) –	5.1042e–1 (1.28e–3) –	5.0726e–1 (1.64e–3) –	5.1202e–1 (1.04e–3) –	5.0796e–1 (8.68e–4) –	5.1646e–1 (1.16e–3)
F9	(10,10)	2.4867e–1 (1.94e–2) –	2.2526e–1 (1.18e–2) –	1.8565e–1 (3.85e–2) –	2.7206e–1 (3.47e–2) –	2.0571e–1 (2.62e–2) –	3.2505e–1 (7.64e–3) –	3.3262e–1 (1.87e–2)
	(10,20)	3.5471e–1 (1.14e–2) –	2.6874e–1 (1.57e–2) –	3.2627e–1 (2.62e–2) –	3.7221e–1 (9.78e–3) –	3.5481e–1 (1.03e–2) –	3.8347e–1 (3.48e–3) +	3.7779e–1 (7.18e–3)
	(10,30)	3.8621e–1 (4.81e–3) –	3.0198e–1 (1.80e–2) –	3.7597e–1 (1.45e–2) –	3.8910e–1 (5.47e–3) –	3.8544e–1 (4.08e–3) –	3.9551e–1 (9.51e–4) –	3.9793e–1 (1.37e–3)
F10	(10,10)	1.7580e–1 (1.98e–2) –	1.9175e–1 (1.83e–2) –	1.2513e–1 (2.58e–2) –	1.9980e–1 (2.89e–2) –	1.6040e–1 (1.93e–2) –	2.3040e–1 (9.93e–3) –	2.5126e–1 (1.23e–2)
	(10,20)	2.7517e–1 (4.23e–3) –	2.5397e–1 (7.25e–3) –	2.6673e–1 (5.33e–3) –	2.8661e–1 (3.57e–3) –	2.7983e–1 (5.48e–3) –	2.9491e–1 (3.73e–3) =	2.9397e–1 (3.70e–3)
	(10,30)	3.0372e–1 (3.81e–3) –	2.6493e–1 (8.01e–3) –	2.9478e–1 (3.63e–3) –	3.0290e–1 (3.13e–3) –	3.0576e–1 (3.58e–3) –	3.1066e–1 (3.35e–3) –	3.1403e–1 (2.55e–3)
+/ – / =		0/42/0	3/39/0	3/39/0	2/40/0	3/38/1	4/36/2	
Average ranking		5.65	5.90	4.48	4.27	3.33	2.85	1.51

(5) The parameters in VSDPS are set as follows: In the dual prediction strategies, as shown in Section 3.3, the improved linear prediction is 60% and the dynamic particle swarm prediction is 40%. The capacity of the archive used to store the nondominant individuals to obtain the Gbest in Section 3.3 is 150. The inherent weight, w, is set to 0.5. The learning factors, r_1 and r_2, are random numbers between 0 and 1. The parameter settings of the GA operator in Section 3.4 is the same as that in NSGA-II [28].

5. Experimental results and analysis

In this section, six state-of-the-art algorithms were compared with our proposed algorithm VSDPS on four test suites comprising a total of twenty-eight test instances, which include FDA, dMOP, F, and DF test suites. To evaluate the effect of change frequency on the performance of the algorithms, each test instance corresponds to three types of change frequency, where τ_t = 10, 20, and 30, and n_t is set to 10. In all three cases, each test instance ran 30 times independently in a single run. The statistics of the MIGD and MHV values for all compared algorithms are collected in Tables 1–4. In these tables, the values in each small grid represent the average value and standard deviation. The statistics following (+), (–), and (=) indicate that the algorithm performed better, worse, and similarly compared to VSDPS according to the Wilcoxon rank sum test at a significance level of 0.05. The best result in each test instance is highlighted. Figure S-I and Figure S-II show the trends of the IGD values for seven algorithms on several representative test instances during the first 30 time periods. Figure S-I and Figure S-II can be found in the supplementary material. In the end, the results of the Friedman average ranking of seven algorithms on all test instances in three different frequency environmental changes are listed in Table 5.

5.1. Results on FDA, dMOP and F test suites

In Table 1, seven algorithms are compared on the FDA, dMOP, and F benchmark suites for the MIGD metric. The three test suites consist of a total of fourteen test instances, with each test instance corresponding to three different frequency environmental changes. Therefore, each algorithm encounters 42 cases. Among the 42 cases, VSDPS has 20 cases with the best performance. In the statistical results of the Friedman test, the ranking result of VSDPS is 1.86, which surpasses the comparison algorithms, demonstrating its exceptional performance. In some cases, although VSDPS does not perform the best, it performs the second best. It is even close to being the best performing algorithm in some instances. For example, in FDA4 and DMOP3, the performance of VSDPS is close to that of MOEA/D-IEC.

In Table 2, the performance of VSDPS is similar to that of Table 1. Among the 42 cases, VSDPS has 32 cases with the best performance. In the statistical results of the Friedman test, the ranking result of VSDPS is 1.51, indicating superior performance compared to the other algorithms. Hence, the outstanding performance of VSDPS is evident. Although VSDPS does not perform the best on FDA2 and FDA3, it ranks the second best and third best respectively.

To summarize, VSDPS performs well on FDA, dMOP, and F test suites due to its variable stepsize and dual prediction strategies. It can be seen from the experimental results in Tables 1 and 2 that VSDPS performs better than other algorithms, which demonstrates that variable stepsize and dual prediction strategies allow VSDPS to swiftly

Table 3
The statistics of the MIGD values obtained by VSDPS and the six other state-of-the-art algorithms were run 30 times independently on the DF test suite. The final two rows display the results of the Wilcoxon rank sum test and the Friedman test, respectively, with the best results highlighted.

Problem	$(n_t,\ \tau_t)$	MOEA/D-KF	SGEA	MOEA/D-FD	MOEA/D-DM	MOEA/D-IEC	MOEA/D-ARMS	VSDPS
DF1	(10,10)	1.6726e−2 (1.01e−3) −	1.6342e−2 (2.83e−3) −	1.3583e−2 (1.26e−3) −	1.5443e−2 (1.44e−3) −	7.9245e−3 (3.72e−4) +	9.5250e−3 (4.31e−4) −	8.6874e−3 (4.75e−4)
	(10,20)	6.8553e−3 (2.40e−4) −	9.5147e−3 (8.40e−4) −	6.2932e−3 (1.77e−4) −	6.8370e−3 (2.58e−4) −	5.1755e−3 (1.58e−4) −	5.6777e−3 (1.35e−4) −	4.9646e−3 (7.44e−5)
	(10,30)	5.0396e−3 (8.55e−5) −	7.9529e−3 (1.51e−3) −	5.0334e−3 (1.05e−4) −	5.1526e−3 (1.24e−4) −	4.4809e−3 (5.27e−5) −	4.7463e−3 (5.44e−5) −	4.2440e−3 (5.03e−5)
DF2	(10,10)	5.4303e−2 (6.23e−3) −	1.2657e−1 (1.26e−2) −	7.1567e−2 (6.20e−3) −	8.0298e−2 (8.91e−3) −	3.2801e−2 (2.76e−3) =	3.2774e−2 (4.63e−3) =	3.2330e−2 (5.77e−3)
	(10,20)	1.7187e−2 (2.54e−3) −	1.0896e−1 (5.39e−3) −	2.4260e−2 (2.31e−3) −	2.5995e−2 (3.52e−3) −	9.0514e−3 (8.37e−4) −	1.0472e−2 (1.10e−3) −	7.4018e−3 (2.93e−4)
	(10,30)	7.9432e−3 (5.70e−4) −	1.0791e−1 (8.44e−3) −	1.0780e−2 (7.53e−4) −	1.1599e−2 (5.87e−4) −	5.6156e−3 (1.28e−4) −	6.1398e−3 (2.25e−4) −	5.1519e−3 (7.80e−5)
DF3	(10,10)	2.4656e−2 (8.39e−3) −	2.6566e−1 (2.16e−2) −	1.1706e−2 (2.60e−3) −	1.4097e−2 (3.58e−3) −	7.7477e−3 (2.02e−3) =	1.3803e−2 (3.48e−3) −	8.2088e−3 (1.48e−3)
	(10,20)	9.4581e−3 (2.68e−4) −	2.2426e−1 (1.59e−2) −	7.5464e−3 (3.48e−4) −	8.2869e−3 (3.77e−4) −	5.8633e−3 (1.35e−4) =	7.9539e−3 (2.50e−4) −	5.8656e−3 (1.00e−4)
	(10,30)	6.6385e−3 (1.64e−4) −	2.2909e−1 (3.09e−2) −	6.0832e−3 (1.64e−4) −	6.5084e−3 (2.01e−4) −	5.0022e−3 (7.19e−5) +	6.1470e−3 (9.63e−5) −	5.0553e−3 (5.78e−5)
DF4	(10,10)	8.5450e−2 (1.96e−3) −	7.5582e−2 (2.46e−3) +	8.5373e−2 (1.20e−3) −	8.7264e−2 (1.32e−3) −	1.3935e−1 (5.37e−3) −	8.4960e−2 (8.72e−4) −	8.2006e−2 (4.89e−4)
	(10,20)	8.1453e−2 (8.04e−4) −	7.4468e−2 (2.20e−3) +	8.2224e−2 (7.90e−4) −	8.3162e−2 (6.92e−4) −	1.1961e−1 (3.13e−3) −	8.1987e−2 (6.75e−4) −	7.9251e−2 (3.83e−4)
	(10,30)	8.1560e−2 (4.81e−4) −	8.3097e−2 (3.11e−3) −	8.1326e−2 (9.18e−4) −	8.1727e−2 (7.60e−4) −	1.0995e−1 (2.60e−3) −	8.1348e−2 (7.76e−4) −	7.8443e−2 (3.87e−4)
DF5	(10,10)	1.4101e−2 (7.18e−4) −	1.4423e−2 (5.21e−4) −	1.0404e−2 (6.77e−4) −	1.2783e−2 (8.02e−4) −	6.4751e−2 (5.20e−3) −	1.0621e−2 (4.93e−4) −	7.4512e−3 (3.27e−4)
	(10,20)	6.4421e−3 (1.86e−4) −	8.2306e−3 (2.05e−4) −	6.1414e−3 (1.61e−4) −	6.6714e−3 (2.61e−4) −	4.4365e−2 (1.68e−3) −	6.0640e−3 (7.18e−5) −	4.8886e−3 (5.44e−5)
	(10,30)	5.1285e−3 (7.64e−5) −	6.6969e−3 (2.28e−4) −	5.0363e−3 (5.74e−5) −	5.2709e−3 (1.27e−4) −	3.7673e−2 (9.07e−4) −	5.0183e−3 (5.52e−5) −	4.3238e−3 (3.19e−5)
DF6	(10,10)	3.0839e+0 (1.22e+0) +	1.6853e+0 (6.75e−1) +	1.1230e+0 (5.22e−1) +	2.3523e+0 (1.22e+0) +	2.2698e+0 (1.42e+0) +	2.1907e+0 (6.96e−1) +	3.9139e+0 (1.20e+0)
	(10,20)	1.6846e+0 (9.71e−1) +	1.1576e+0 (6.27e−1) +	8.2880e−1 (5.01e−1) +	1.1223e+0 (4.48e−1) +	1.9622e+0 (6.26e−1) +	8.0165e−1 (6.41e−1) +	3.9062e+0 (1.49e+0)
	(10,30)	1.9488e+0 (8.46e−1) +	1.0858e+0 (5.01e−1) +	7.2606e−1 (4.11e−1) +	1.0022e+0 (7.65e−1) +	8.0919e−1 (9.88e−1) +	8.3594e−1 (5.28e−1) +	2.7760e+0 (1.06e+0)
DF7	(10,10)	2.7420e−2 (2.34e−2) +	3.2413e−1 (7.92e−2) −	3.2558e−2 (2.74e−2) +	2.7338e−2 (1.63e−2) +	4.8647e−2 (1.15e−2) −	2.3660e−2 (2.26e−2) +	3.5551e−2 (3.06e−2)
	(10,20)	1.5147e−2 (6.46e−3) =	2.1715e−1 (6.03e−2) −	1.3780e−2 (3.41e−4) =	1.4175e−2 (7.18e−4) =	4.0253e−2 (1.12e−2) −	3.0226e−2 (2.42e−2) −	1.8878e−2 (1.57e−2)
	(10,30)	1.6700e−2 (1.41e−2) =	2.3580e−1 (7.38e−2) −	1.4559e−2 (7.67e−3) =	1.6224e−2 (1.25e−2) =	3.6797e−2 (6.05e−3) −	2.1259e−2 (1.99e−2) +	2.5855e−2 (2.17e−2)
DF8	(10,10)	1.3888e−1 (3.12e−3) −	1.6024e−1 (1.21e−3) −	1.3103e−1 (1.74e−3) +	1.3009e−1 (1.49e−3) +	2.4287e−1 (1.27e−2) −	1.2586e−1 (1.18e−3) +	1.3586e−1 (2.18e−3)
	(10,20)	1.2304e−1 (1.34e−3) =	1.5781e−1 (9.38e−4) −	1.2381e−1 (3.71e−4) =	1.2346e−1 (4.52e−4) =	2.1197e−1 (8.03e−3) −	1.2095e−1 (5.89e−4) +	1.2414e−1 (1.12e−3)
	(10,30)	1.2163e−1 (4.49e−4) −	1.5633e−1 (1.01e−3) −	1.2133e−1 (4.74e−4) =	1.2142e−1 (3.66e−4) −	1.9610e−1 (4.12e−3) −	1.1947e−1 (1.86e−4) +	1.2101e−1 (4.00e−4)
DF9	(10,10)	2.1120e+0 (6.96e−3) −	2.2657e+0 (3.43e−2) −	2.1453e+0 (2.80e−2) −	2.1307e+0 (1.25e−2) −	2.1352e+0 (7.82e−3) −	2.0951e+0 (3.54e−3) =	2.1000e+0 (1.12e−2)
	(10,20)	2.0822e+0 (1.26e−3) −	2.1324e+0 (8.99e−3) −	2.0888e+0 (4.66e−3) −	2.0857e+0 (3.88e−3) −	2.1075e+0 (2.93e−3) −	2.0795e+0 (9.62e−4) =	2.0788e+0 (1.70e−3)
	(10,30)	2.0778e+0 (9.15e−4) +	2.1082e+0 (3.94e−3) −	2.0791e+0 (1.27e−3) −	2.0801e+0 (1.47e−3) −	2.0980e+0 (1.89e−3) −	2.0756e+0 (8.32e−4) =	2.0790e+0 (8.43e−3)
DF10	(10,10)	2.4000e−1 (1.52e−2) −	8.0725e−2 (8.83e−3) +	1.6029e−1 (1.34e−2) =	1.6374e−1 (1.49e−2) =	1.8219e−1 (1.86e−2) −	1.8481e−1 (1.83e−2) −	1.6016e−1 (1.75e−2)
	(10,20)	1.9539e−1 (1.19e−2) −	7.2548e−2 (5.82e−3) +	1.4654e−1 (1.03e−2) =	1.4594e−1 (7.12e−3) =	1.7666e−1 (6.34e−2) =	1.5652e−1 (1.49e−2) =	1.5380e−1 (8.91e−3)
	(10,30)	1.8361e−1 (1.76e−2) −	6.7994e−2 (5.16e−3) +	1.3554e−1 (4.47e−3) =	1.3896e−1 (5.26e−3) =	1.5540e−1 (3.91e−2) −	1.4314e−1 (7.23e−3) −	1.3874e−1 (1.54e−2)
DF11	(10,10)	8.7359e−2 (1.21e−3) −	1.2121e−1 (1.96e−2) −	8.2295e−2 (1.12e−3) −	8.0493e−2 (1.25e−3) −	1.0288e−1 (8.34e−4) −	7.7729e−2 (4.21e−4) −	7.6939e−2 (4.72e−4)
	(10,20)	7.7894e−2 (5.57e−4) −	9.2271e−2 (6.48e−3) −	7.6394e−2 (5.74e−4) −	7.5774e−2 (4.13e−4) −	9.8722e−2 (3.13e−4) −	7.3649e−2 (2.12e−4) −	7.2723e−2 (3.01e−4)
	(10,30)	7.5138e−2 (2.97e−4) −	8.4960e−2 (9.59e−3) −	7.4492e−2 (3.42e−4) −	7.3974e−2 (2.94e−4) −	9.7072e−2 (2.81e−4) −	7.2441e−2 (2.17e−4) −	7.1940e−2 (1.58e−4)
DF12	(10,10)	2.9030e−1 (4.16e−3) −	5.3335e−1 (4.28e−2) −	2.8242e−1 (1.54e−3) =	2.8290e−1 (1.21e−3) =	2.7631e−1 (9.04e−4) +	2.7949e−1 (1.04e−3) +	2.8307e−1 (1.37e−3)
	(10,20)	2.7733e−1 (9.01e−4) −	4.2063e−1 (7.63e−2) −	2.7744e−1 (7.01e−4) −	2.7814e−1 (9.61e−4) −	2.7378e−1 (3.68e−4) +	2.7527e−1 (4.85e−4) =	2.7546e−1 (4.89e−4)
	(10,30)	2.7488e−1 (3.93e−4) +	3.9528e−1 (1.03e−1) −	2.7601e−1 (6.25e−4) −	2.7630e−1 (5.69e−4) −	2.7321e−1 (2.98e−4) +	2.7390e−1 (3.44e−4) +	2.7541e−1 (4.14e−4)
DF13	(10,10)	3.0703e−1 (7.51e−3) +	1.9185e−1 (4.06e−3) +	3.0547e−1 (5.59e−3) +	3.0345e−1 (7.11e−3) +	3.1894e−1 (4.21e−3) −	3.1236e−1 (5.04e−3) =	3.1298e−1 (2.83e−3)
	(10,20)	3.2032e−1 (5.28e−3) =	1.7254e−1 (2.12e−3) +	3.1392e−1 (4.02e−3) +	3.1145e−1 (6.08e−3) +	3.3037e−1 (4.74e−3) −	3.2622e−1 (3.58e−3) −	3.2333e−1 (1.85e−3)
	(10,30)	3.3027e−1 (3.83e−3) =	1.6791e−1 (2.48e−3) +	3.1779e−1 (4.59e−3) +	3.1654e−1 (3.66e−3) +	3.3328e−1 (2.11e−3) −	3.3001e−1 (3.12e−3) =	3.3016e−1 (1.76e−3)
DF14	(10,10)	6.8112e−2 (1.27e−3) −	9.3128e−1 (1.78e−1) −	6.3937e−2 (1.45e−3) −	6.5697e−2 (1.37e−3) −	6.2598e−2 (6.96e−4) −	6.1323e−2 (7.30e−4) −	5.8809e−2 (4.13e−4)
	(10,20)	6.0037e−2 (5.84e−4) −	7.3791e−1 (8.90e−2) −	5.8786e−2 (5.05e−4) −	5.9304e−2 (8.48e−4) −	5.8863e−2 (1.81e−4) −	5.7232e−2 (1.61e−4) −	5.6108e−2 (4.24e−4)
	(10,30)	5.7564e−2 (3.64e−4) −	7.3515e−1 (1.58e−1) −	5.7408e−2 (3.49e−4) −	5.7262e−2 (3.58e−4) −	5.7881e−2 (3.37e−4) −	5.6276e−2 (3.21e−4) −	5.5456e−2 (2.18e−4)
+/ − / =		7/30/5	11/31/0	8/26/8	8/27/7	8/30/4	10/25/7	
Average ranking		4.75	5.38	3.38	4.06	4.7262	2.93	2.77

monitor new PS/PF when the environment changes. The shortcoming of VSDPS is that the performance on FDA3 is not ideal. FDA3 belongs to the second type of DMOPs. When the environment changes, both PS_t and PF_t will change, which increases the difficulty of finding new PS_t/PF_t. MOEA/D-ARMS performs the best on FDA3. It uses a mechanism response pool that can adaptively change the probability of each strategy being selected next time in response to environmental changes. Therefore, it can be relatively accurate to predict the population after environmental changes. VSDPS cannot control the prediction as effectively as MOEA/D-ARMS, and therefore, it does not effectively track the new PS_t/PF_t on FDA3.

In order to comprehensively and intuitively compare the performance of seven algorithms on FDA, dMOP, and F test suites, this paper selected several representative test instances and described the trends of the IGD for the seven different algorithms in the first 30 time periods. In Figure S-I in the supplementary material, it can be seen that VSDPS responds more stably to the changes and recovers faster on most of the test instances, thereby achieving higher convergence performance compared to other algorithms. Although the IGD values of VSDPS on DMOP1 are not consistently the lowest, the overall performance of VSDPS is the most stable compared to the other six algorithms.

5.2. Results on DF test suite

The DF test suite consists of fourteen test instances, with each test instance corresponding to three different frequency environmental changes. Therefore, each algorithm encounters 42 cases. The statistical results of MIGD and MHV on the DF test suite are shown in Tables 3 and 4.

As indicated in Table 3, VSDPS demonstrates superior MIGD values compared to the other six algorithms on most of the test instances. VSDPS has the best performance in 16 out of the 42 cases. In the statistical results of the Friedman test, the ranking result of VSDPS is 2.77, which outperforms the compared algorithms. In some cases, although VSDPS is not the best, it performs the second best, such as DF1, DF3, DF4, and DF9. Regarding the nine bi-objective benchmark instances DF1-DF9, VSDPS performs the first or second best in most cases, indicating that VSDPS is particularly well-suited for bi-objective problems with diverse properties. However, the performance of VSDPS on DF6 is poor. For five tri-objective benchmark functions DF10-DF14, VSDPS achieved significantly better results on DF11 and DF14 compared to the six comparison algorithms. On DF10 and DF12, there is little difference between VSDPS and the best comparison algorithm. However, VSDPS performs worse on DF13.

Table 4 shows the MHV values of the seven algorithms. It is found that the MHV results are highly consistent with the MIGD results on DF test suite. Out of the 42 cases, VSDPS has the best performance in 19 cases. In the statistical results of the Friedman test, the ranking result of VSDPS is 2.77, which outperforms the compared algorithms. Regarding the nine bi-objective benchmark functions DF1-DF9, VSDPS performs the first or second best in most cases. VSDPS has poor performance on DF6. For five tri-objective benchmark functions DF10-DF14, VSDPS has significantly better results than all the comparison algorithms on DF11 and DF14. There is little difference between VSDPS and the best comparison algorithm on DF10 and DF12. VSDPS has performed poorly on DF13.

To summarize, all algorithms performed worse on the DF test suite than on the FDA, dMOP, and F test suites because the DF test suite was

Table 4
The statistics of the MHV values obtained by VSDPS and the six other state-of-the-art algorithms were run 30 times independently on the DF test suite. The final two rows display the results of the Wilcoxon rank sum test and the Friedman test, respectively, with the best results highlighted.

Problem	$(n_t,\ \tau_t)$	MOEA/D-KF	SGEA	MOEA/D-FD	MOEA/D-DM	MOEA/D-IEC	MOEA/D-ARMS	VSDPS
DF1	(10,10)	5.0520e−1 (1.56e−3) −	5.0818e−1 (2.72e−3) −	5.1084e−1 (1.86e−3) −	5.0796e−1 (2.17e−3) −	5.1969e−1 (5.81e−4) +	5.1674e−1 (6.75e−4) −	5.1853e−1 (7.43e−4)
	(10,20)	5.2113e−1 (4.27e−4) −	5.1889e−1 (9.44e−4) −	5.2224e−1 (2.44e−4) −	5.2128e−1 (4.50e−4) −	5.2424e−1 (2.65e−4) −	5.2327e−1 (2.43e−4) −	5.2469e−1 (1.29e−4)
	(10,30)	5.2438e−1 (1.71e−4) −	5.2178e−1 (1.59e−3) −	5.2446e−1 (1.93e−4) −	5.2422e−1 (2.33e−4) −	5.2555e−1 (1.12e−4) −	5.2499e−1 (1.16e−4) −	5.2617e−1 (9.40e−5)
DF2	(10,10)	6.5361e−1 (7.02e−3) −	5.8760e−1 (1.33e−2) −	6.3306e−1 (7.04e−3) −	6.2083e−1 (1.03e−2) −	6.7623e−1 (4.12e−3) −	6.8115e−1 (4.07e−3) =	6.8127e−1 (5.09e−3)
	(10,20)	6.9985e−1 (2.67e−3) −	6.0428e−1 (6.45e−3) −	6.8872e−1 (2.08e−3) −	6.8777e−1 (2.95e−3) −	7.1058e−1 (1.19e−3) −	7.0916e−1 (8.88e−4) −	7.1360e−1 (4.23e−4)
	(10,30)	7.1252e−1 (8.67e−4) −	6.0530e−1 (9.02e−3) −	7.0790e−1 (1.21e−3) −	7.0660e−1 (9.24e−4) −	7.1616e−1 (2.30e−4) −	7.1553e−1 (3.38e−4) −	7.1728e−1 (1.40e−4)
DF3	(10,10)	4.6003e−1 (1.31e−2) −	2.6929e−1 (1.25e−2) −	4.8006e−1 (4.23e−3) −	4.7648e−1 (5.71e−3) −	4.8688e−1 (3.46e−3) =	4.7665e−1 (5.74e−3) −	4.8616e−1 (2.55e−3)
	(10,20)	4.8335e−1 (3.38e−4) −	2.9708e−1 (1.03e−2) −	4.8674e−1 (5.45e−4) −	4.8559e−1 (6.59e−4) −	4.9006e−1 (2.06e−4) =	4.8602e−1 (4.33e−4) −	4.9014e−1 (2.10e−4)
	(10,30)	4.8665e−1 (4.82e−3) −	2.9345e−1 (1.86e−2) −	4.8833e−1 (4.22e−3) −	4.8760e−1 (4.23e−3) −	4.9079e−1 (3.49e−3) =	4.8852e−1 (3.52e−3) −	4.9091e−1 (3.08e−3)
DF4	(10,10)	2.8131e−1 (1.73e−4) −	2.8366e−1 (1.98e−4) −	2.8350e−1 (1.13e−4) −	2.8330e−1 (1.56e−4) −	2.7898e−1 (3.58e−4) −	2.8339e−1 (8.75e−5) −	2.8449e−1 (6.20e−5)
	(10,20)	2.8411e−1 (5.85e−5) −	2.8413e−1 (2.32e−4) −	2.8436e−1 (8.64e−5) −	2.8424e−1 (5.25e−5) −	2.8108e−1 (2.54e−4) −	2.8443e−1 (3.95e−5) −	2.8499e−1 (2.31e−5)
	(10,30)	2.8456e−1 (5.32e−5) −	2.8447e−1 (2.15e−4) −	2.8466e−1 (4.20e−5) −	2.8459e−1 (4.62e−5) −	2.8212e−1 (1.37e−4) −	2.8476e−1 (3.23e−5) −	2.8514e−1 (2.16e−5)
DF5	(10,10)	1.2498e−1 (5.80e−4) −	1.2599e−1 (2.65e−4) −	1.2596e−1 (4.39e−4) −	1.2534e−1 (4.25e−4) −	1.2087e−1 (7.18e−4) −	1.2642e−1 (4.08e−4) −	1.2720e−1 (2.64e−4)
	(10,20)	1.2796e−1 (1.27e−4) −	1.2796e−1 (1.01e−4) −	1.2800e−1 (9.54e−5) −	1.2780e−1 (1.55e−4) −	1.2465e−1 (1.59e−4) −	1.2825e−1 (6.25e−5) −	1.2861e−1 (4.35e−5)
	(10,30)	1.2851e−1 (4.79e−5) −	1.2832e−1 (2.45e−4) −	1.2849e−1 (4.71e−5) −	1.2838e−1 (6.47e−5) −	1.2571e−1 (8.42e−5) −	1.2863e−1 (2.75e−5) −	1.2885e−1 (2.35e−5)
DF6	(10,10)	1.8663e−1 (1.24e−1) −	3.6668e−1 (7.86e−2) +	5.5727e−1 (5.92e−2) +	3.1721e−1 (1.21e−1) +	3.2717e−1 (1.83e−1) =	3.1468e−1 (1.07e−1) +	2.3492e−1 (1.12e−1)
	(10,20)	3.1066e−1 (1.88e−1) +	4.6211e−1 (1.01e−1) +	6.3471e−1 (4.30e−2) +	4.1361e−1 (5.67e−2) +	3.0223e−1 (1.50e−1) +	4.5485e−1 (1.53e−1) +	8.8972e−2 (9.62e−2)
	(10,30)	2.2795e−1 (1.41e−1) =	4.9127e−1 (9.68e−2) +	5.7189e−1 (9.96e−2) +	4.6576e−1 (1.72e−1) +	4.9884e−1 (2.23e−1) +	4.7259e−1 (1.46e−1) +	1.9149e−1 (1.62e−1)
DF7	(10,10)	1.5124e−1 (2.60e−3) =	1.2276e−1 (1.15e−2) −	1.5053e−1 (2.99e−3) +	1.5096e−1 (1.90e−3) +	1.4953e−1 (1.54e−3) −	1.5167e−1 (2.64e−3) =	1.5040e−1 (3.40e−3)
	(10,20)	1.5274e−1 (8.91e−4) =	1.3413e−1 (3.64e−3) −	1.5289e−1 (7.16e−5) =	1.5284e−1 (1.98e−4) =	1.5100e−1 (1.48e−3) −	1.5102e−1 (2.84e−3) −	1.5237e−1 (1.84e−3)
	(10,30)	1.5262e−1 (1.71e−3) =	1.3056e−1 (1.06e−2) −	1.5285e−1 (1.11e−3) =	1.5267e−1 (1.53e−3) =	1.5151e−1 (8.74e−4) =	1.5212e−1 (2.37e−3) =	1.5161e−1 (2.56e−3)
DF8	(10,10)	8.0415e−2 (1.74e−4) −	7.9396e−2 (1.68e−4) −	8.0730e−2 (1.66e−4) +	8.0758e−2 (1.24e−4) +	7.4899e−2 (6.89e−4) −	8.1122e−2 (1.26e−4) +	8.0588e−2 (1.53e−4)
	(10,20)	8.1503e−2 (7.43e−5) +	7.9559e−2 (1.42e−4) −	8.1428e−2 (4.79e−5) =	8.1475e−2 (6.95e−5) =	7.7134e−2 (5.48e−4) −	8.1682e−2 (5.37e−5) +	8.1357e−2 (1.14e−4)
	(10,30)	8.1803e−2 (4.23e−5) +	7.9766e−2 (9.08e−5) −	8.1747e−2 (5.91e−5) +	8.1731e−2 (5.19e−5) +	7.8144e−2 (3.02e−4) −	8.1909e−2 (3.25e−5) +	8.1663e−2 (6.86e−5)
DF9	(10,10)	3.1297e−2 (1.37e−3) =	2.1806e−2 (4.47e−3) −	3.1806e−2 (1.53e−3) =	3.1921e−2 (1.10e−3) =	3.1420e−2 (1.21e−3) =	3.1969e−2 (1.88e−3) +	3.0086e−2 (3.89e−3)
	(10,20)	3.4850e−2 (3.98e−4) =	2.4812e−2 (3.25e−3) −	3.3966e−2 (7.29e−4) −	3.3343e−2 (2.90e−3) =	3.3289e−2 (2.28e−3) =	3.5042e−2 (1.94e−4) +	3.4186e−2 (1.73e−3)
	(10,30)	3.5243e−2 (8.50e−4) =	2.3600e−2 (5.68e−3) −	3.5252e−2 (2.93e−4) =	3.4879e−2 (1.36e−3) =	3.5090e−2 (2.47e−4) =	3.5569e−2 (3.43e−4) +	3.2326e−2 (6.01e−3)
DF10	(10,10)	4.6327e−1 (1.01e−2) −	6.4927e−1 (5.96e−3) +	6.3584e−1 (4.83e−3) +	6.2725e−1 (7.07e−3) +	5.7446e−1 (1.91e−2) −	5.9663e−1 (1.47e−2) −	6.1925e−1 (1.43e−2)
	(10,20)	5.4422e−1 (1.72e−2) −	6.5563e−1 (3.40e−3) +	6.4400e−1 (1.69e−3) +	6.3944e−1 (3.16e−3) +	6.0937e−1 (3.07e−2) =	6.3064e−1 (6.29e−3) +	6.1525e−1 (1.54e−2)
	(10,30)	5.9339e−1 (7.93e−3) −	6.5945e−1 (2.75e−3) +	6.4575e−1 (1.54e−3) =	6.4318e−1 (1.58e−3) −	6.2897e−1 (1.73e−2) −	6.3922e−1 (3.46e−3) −	6.4358e−1 (7.15e−3)
DF11	(10,10)	2.7773e−1 (1.27e−3) −	2.6835e−1 (6.91e−3) −	2.8176e−1 (2.30e−3) −	2.8325e−1 (2.34e−3) −	2.8493e−1 (8.83e−4) −	2.8644e−1 (5.34e−4) −	2.8895e−1 (4.90e−4)
	(10,20)	2.8559e−1 (6.66e−4) −	2.7865e−1 (3.71e−3) −	2.8748e−1 (1.27e−3) −	2.8808e−1 (1.30e−3) −	2.8821e−1 (3.31e−4) −	2.8958e−1 (3.05e−4) −	2.9131e−1 (3.14e−4)
	(10,30)	2.8787e−1 (2.93e−4) −	2.8264e−1 (4.80e−3) −	2.8953e−1 (5.16e−4) −	2.8990e−1 (5.40e−4) −	2.8970e−1 (2.57e−4) −	2.9076e−1 (3.53e−4) −	2.9237e−1 (2.00e−4)
DF12	(10,10)	8.8280e−1 (3.37e−3) −	8.5294e−1 (6.01e−3) −	9.0475e−1 (6.69e−4) =	9.0421e−1 (5.33e−4) −	9.0544e−1 (3.81e−4) +	9.0500e−1 (4.02e−4) =	9.0502e−1 (4.89e−4)
	(10,20)	9.0117e−1 (8.42e−4) −	8.6855e−1 (1.59e−2) −	9.0683e−1 (3.65e−4) =	9.0660e−1 (5.01e−4) −	9.0746e−1 (2.55e−4) +	9.0745e−1 (2.48e−4) +	9.0711e−1 (2.50e−4)
	(10,30)	9.0485e−1 (4.30e−4) −	8.7877e−1 (2.31e−2) −	9.0751e−1 (2.32e−4) −	9.0742e−1 (4.00e−4) −	9.0827e−1 (1.69e−4) −	9.0814e−1 (2.57e−4) −	9.0864e−1 (1.88e−4)
DF13	(10,10)	4.2089e−1 (1.87e−2) −	6.0230e−1 (3.95e−3) +	5.0859e−1 (1.21e−2) −	4.7391e−1 (1.64e−2) −	5.1901e−1 (1.20e−2) −	5.0660e−1 (9.82e−3) −	5.6168e−1 (4.93e−3)
	(10,20)	5.0351e−1 (1.04e−2) −	6.4230e−1 (2.05e−3) +	5.4644e−1 (5.46e−3) −	5.3096e−1 (5.80e−3) −	5.5071e−1 (4.81e−3) −	5.4367e−1 (5.67e−3) −	5.7723e−1 (3.97e−3)
	(10,30)	5.4061e−1 (1.23e−2) −	6.5482e−1 (2.08e−3) +	5.6004e−1 (5.65e−3) −	5.5261e−1 (5.48e−3) −	5.6616e−1 (3.73e−3) −	5.5797e−1 (3.62e−3) −	5.8542e−1 (2.20e−3)
DF14	(10,10)	5.5650e−1 (2.21e−3) −	7.3109e−2 (3.82e−2) −	5.7088e−1 (1.96e−3) −	5.6521e−1 (2.64e−3) −	5.7726e−1 (1.56e−3) −	5.7386e−1 (1.29e−3) −	5.8281e−1 (8.74e−4)
	(10,20)	5.7568e−1 (1.21e−3) −	8.0065e−2 (2.33e−2) −	5.8175e−1 (8.93e−4) −	5.8025e−1 (1.45e−3) −	5.8322e−1 (5.96e−4) −	5.8089e−1 (5.67e−4) −	5.8391e−1 (7.33e−4)
	(10,30)	5.8032e−1 (8.15e−4) −	9.9428e−2 (3.69e−2) −	5.8340e−1 (7.32e−4) =	5.8333e−1 (7.45e−4) =	5.8506e−1 (4.40e−4) +	5.8184e−1 (6.12e−4) −	5.8347e−1 (6.63e−4)
+/ − / =		3/32/7	9/33/0	8/25/9	8/27/7	6/27/9	11/27/4	
Average ranking		5.06	5.27	3.36	4.31	4.06	3.17	2.77

adapted from these suites, making the test instances more challenging. However, VSDPS still outperforms other algorithms. As can be seen in Figure S-I in the supplementary material, in most cases, the IGD of VSDPS is at the lowest level, which indicates that VSDPS can track PS_t/PF_t quickly after the environment changes, demonstrating that the dual prediction strategies are effective in solving DMOPs.

To enable a more thorough and intuitive comparison of the performance of the seven algorithms on the DF test suite, this paper selected a few representative test instances from the DF test suite and described the trends of the IGD for each algorithm over the first 30 time periods. In Figure S-II in the supplementary material, it can be seen that VSDPS responds to changes more stably and recovers faster on DF2, DF3, and DF5 compared to other algorithms except MOEA/D-FD. VSDPS presents a behavior similar to MOEA/D-FD. Although the IGD values of VSDPS on DF9 are not consistently the lowest, they exhibit less fluctuation than the other six algorithms. Therefore, the overall performance of VSDPS is the most stable compared to the others.

5.3. Effectiveness analysis of different strategies

The VSDPS has three key strategies and shows capability. In order to validate the effectiveness of the strategies, this subsection will analyse the effectiveness of the three main strategies using ablation experiments. The strategy variants of VSDPS are set as follows.

(1) VSDPS-S: Compared to VSDPS, this variant does not use the variable stepsize, it only uses a fixed stepsize(the difference between current centroid and previous centroid) to respond to the environmental changes.
(2) VSDPS-P: This variant replaces the dual prediction strategies with linear prediction strategy. It is mainly used to verify the effectiveness of the dual prediction strategies.
(3) VSDPS-O: This variant replaces improved MOEA/D-DE used in VSDPS with the MOEA/D-DE [27]. It is mainly used to verify the effectiveness of improved MOEA/D-DE.

The experimental results of these variants on FDA, dMOP, F, and DF test suites are shown in Table S-I and Table S-II, respectively. Tables S-I and S-II can be found in the supplementary material. These tables clearly show that VSDPS achieves the best results, demonstrating the effectiveness of the strategies in VSDPS.

5.4. Results for the ratio of dual prediction strategies

To assess the influence of the ratio between the improved linear prediction strategy and the dynamic particle swarm prediction strategy in the dual prediction strategies on the algorithm, we modified this ratio in the algorithm and carried out an experiment. In this experiment, five sets of ratios and VSDPS were used. In these five groups of ratios, the proportion of the improved linear prediction strategy and the dynamic particle swarm prediction strategy are 100% and 0%, 80% and 20%, 40% and 60%, 20% and 80%, 0% and 100%, which are recorded as VSDPS-P1, VSDPS-P2, VSDPS-P3, VSDPS-P4 and VSDPS-P5, respectively. The five variants are compared with VSDPS when $n_t = 10$, $\tau_t = 10$. The statistical results of MIGD and MHV on FDA, dMOP, F, and DF test suites can be found in Table S-III and Table S-IV in the supplementary material. As can be seen from the experimental results, each ratio is advantageous in some instances, with VSDPS performing the best when the improved linear prediction strategy is 60%, and the dynamic particle swarm prediction strategy is 40%.

Table 5
The average ranking of the seven algorithms obtained by the Friedman test for the combination of environment parameters with varying τ_t and unchanged n_t. The average rankings of MIGD and MHV are shown.

Algorithm	n_t&τ_t(10,10)		n_t&τ_t(10,20)		n_t&τ_t(10,30)	
	Ranking1(MIGD)	Ranking2(MHV)	Ranking1(MIGD)	Ranking2(MHV)	Ranking1(MIGD)	Ranking2(MHV)
MOEA/D-KF	5.46	5.63	4.95	5.27	4.75	4.93
SGEA	5.32	5.23	5.57	5.68	6.14	5.86
MOEA/D-FD	3.93	3.91	4.13	3.93	3.91	3.89
MOEA/D-DM	4.32	4.33	4.27	4.18	4.39	4.34
MOEA/D-IEC	3.88	3.79	3.77	3.71	3.64	3.59
MOEA/D-ARMS	2.79	2.91	2.95	3.25	2.86	3.11
VSDPS	**2.30**	**2.20**	**2.38**	**1.98**	**2.30**	**2.29**

5.5. Influence of variable stepsize

In order to assess the effectiveness of the variable stepsize, several stepsize related variants are used in this subsection for experimental comparison with VSDPS. These variants are referred to as VSDPS-S1, VSDPS-S2, VSDPS-S3, and VSDPS-S4. In VSDPS-S1, the stepsize in each prediction strategy is only related to the centroid of the population. In VSDPS-S2, the stepsize in each prediction strategy is only related to the centroids of the subpopulations. In VSDPS-S3, the stepsize in the improved prediction strategy is only related to the centroid of the population, whereas the stepsize in the dynamic particle swarm prediction strategy is related to the centroid of the population and the centroids of the subpopulations. In VSDPS-S4, the stepsize in the improved prediction strategy is linked to the centroid of the population and the centroids of subpopulations, whereas the stepsize in the dynamic particle swarm prediction strategy is associated with the centroids of the subpopulations. The experimental results of these variants on FDA, dMOP, F, and DF test suites are shown in Table S-V and Table S-VI in the supplementary material. These tables clearly show that VSDPS achieves the best results, demonstrating the effectiveness of the variable stepsize.

5.6. Influence of different values of control number in static optimization

In order to assess the effectiveness of the control number in static optimization, several values of control number are used in this subsection for experimental comparison with VSDPS. These variants are referred to as VSDPS-O1, VSDPS-O2, VSDPS-O3, VSDPS-O4, and VSDPS-O5. The control number in them are 0, 0.2, 0.6, 0.8, 1.0, respectively. The experimental results are shown in Table S-VII and Table S-VIII in the supplementary material. These tables clearly show VSDPS achieves the best results, demonstrating that 0.4 is the most preferred control number in static optimization.

5.7. Influence of τ_t

In order to verify the influence of τ_t on algorithms, this paper conducted the Friedman test for each algorithm, where n_t kept at 10 and τ_t was set to 10, 20, and 30, respectively. The average ranking of all algorithms on MHV and MIGD across all test instances is shown in Table 5. It can be observed from Table 5 that the algorithm is sensitive to changes in τ_t, which will increase the difficulty of the instance. However, compared to the comparison algorithms, VSDPS still performs the best. That is, VSDPS has a significant advantage over the comparison algorithms at a significance level of less than 0.05.

6. Conclusion

In this paper, a dynamic multi-objective evolutionary algorithm with dual prediction strategies, called VSDPS, was proposed for solving DMOPs. When an environmental change is detected, dual prediction strategies that combine improved linear prediction and dynamic particle swarm prediction are used to respond to the change. In addition, the nondominated solutions and the dominated solutions adopt different stepsize during the prediction process. In the static optimization stage, the improved MOEA/D-DE is used as the static multi-objective optimizer, which incorporates the GA operator as well as variants of DE along with the widely used DE/rand/1 operator.

VSDPS was compared with six state-of-the-art DMOEAs on 28 DMOPs, including FDA, dMOP, F, and DF test suites. The experimental results show that VSDPS outperforms the compared algorithms. It can effectively track the changing PS_t/PF_t and strike a good balance between population diversity and convergence. When τ_t changes, VSDPS still achieves satisfactory results, demonstrating improved robustness.

Although VSDPS performed well on most test instances, there is still ample room for improvement, especially in the DF and F test suites. This paper only employs prediction strategies to address environmental changes; however, some changes are unpredictable. In the future, we need to find techniques for dealing with unpredictable changes. The source code of VSDPS can be downloaded from Hu Peng's homepage: https://whuph.github.io/index.html.

CRediT authorship contribution statement

Hu Peng: Writing – original draft, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization. **Chen Pi:** Writing – original draft, Validation, Software, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. **Jianpeng Xiong:** Software, Project administration. **Debin Fan:** Supervision, Project administration. **Fanfan Shen:** Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Hu peng reports financial support was provided by National Natural Science Foundation of China. Hu peng reports financial support was provided by Science and Technology Plan Projects of Jiangxi Provincial Education Department. Fanfan Shen reports financial support was provided by Basic Science (Natural Science) Research Project of Colleges and Universities in Jiangsu Province. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors are unable or have chosen not to specify which data has been used.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62266024), the Science and Technology Plan Projects of Jiangxi Provincial Education Department (GJJ2201906) and the Basic Science (Natural Science) Research Project of Colleges and Universities in Jiangsu Province (22KJA520004).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.future.2024.07.028.

References

[1] Y. Audoux, M. Montemurro, J. Pailhès, Non-uniform rational basis spline hypersurfaces for metamodelling, Comput. Methods Appl. Mech. Engrg. 364 (2020) 112918.

[2] G. Costa, M. Montemurro, J. Pailhès, A general hybrid optimization strategy for curve fitting in the non-uniform rational basis spline framework, J. Optim. Theory Appl. 176 (2018) 225–251.

[3] G. Bertolino, M. Montemurro, N. Perry, F. Pourroy, An efficient hybrid optimization strategy for surface reconstruction, in: Computer Graphics Forum, Vol. 40, (6) 2021, pp. 215–241.

[4] Y. Gong, K. Bian, F. Hao, Y. Sun, Y. Wu, Dependent tasks offloading in mobile edge computing: A multi-objective evolutionary optimization strategy, Future Gener. Comput. Syst. 148 (2023) 314–325.

[5] H. Peng, C. Wang, Y. Han, W. Xiao, X. Zhou, Z. Wu, Micro multi-strategy multi-objective artificial bee colony algorithm for microgrid energy optimization, Future Gener. Comput. Syst. 131 (2022) 59–74.

[6] H. Peng, F. Kong, Q. Zhang, Micro multiobjective evolutionary algorithm with piecewise strategy for embedded-processor-based industrial optimization, IEEE Trans. Cybern. (2023) 1–12.

[7] F. Luna, P.H. Zapata-Cano, J.C. Gonzalez-Macias, J.F. Valenzuela-Valdés, Approaching the cell switch-off problem in 5G ultra-dense networks with dynamic multi-objective optimization, Future Gener. Comput. Syst. 110 (2020) 876–891.

[8] J. Zheng, F. Zhou, J. Zou, S. Yang, Y. Hu, A dynamic multi-objective optimization based on a hybrid of pivot points prediction and diversity strategies, Swarm Evol. Comput. 78 (2023) 101284.

[9] T. Macias-Escobar, L. Cruz-Reyes, H. Fraire, B. Dorronsoro, Plane Separation: A method to solve dynamic multi-objective optimization problems with incorporated preferences, Future Gener. Comput. Syst. 110 (2020) 864–875.

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

[11] S. Liu, Q. Lin, K.-C. Wong, L. Ma, C.A. Coello Coello, D. Gong, A novel multi-objective evolutionary algorithm with dynamic decomposition strategy, Swarm Evol. Comput. 48 (2019) 182–200.

[12] J. Xiong, Z. Zhou, K. Tian, T. Liao, J. Shi, A multi-objective approach for weapon selection and planning problems in dynamic environments, J. Ind. Manage. Optim. 13 (3) (2017) 1189–1211.

[13] X. Peng, D. Xu, F. Zhang, UAV online path planning based on dynamic multiobjective evolutionary algorithm, in: Proceedings of the 30th Chinese Control Conference, 2011, pp. 5424–5429.

[14] Y. Wu, Y. Jin, X. Liu, A directed search strategy for evolutionary dynamic multiobjective optimization, Soft Comput. 19 (2015) 3221–3235.

[15] F. Wang, F. Liao, Y. Li, H. Wang, A new prediction strategy for dynamic multi-objective optimization using Gaussian Mixture Model, Inform. Sci. 580 (2021) 331–351.

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

[17] L. Cao, L. Xu, E. Goodman, H. Li, A First-Order Difference Model-Based Evolutionary Dynamic Multiobjective Optimization, Springer, 2017, pp. 644–655.

[18] A. Muruganantham, K.C. Tan, P. Vadakkepat, Evolutionary dynamic multiobjective optimization via kalman filter prediction, IEEE Trans. Cybern. 46 (12) (2016) 2862–2873.

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

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

[21] K. Deb, U.B. Rao N, 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.

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

[23] Y. Chen, J. Zou, Y. Liu, S. Yang, J. Zheng, W. Huang, Combining a hybrid prediction strategy and a mutation strategy for dynamic multiobjective optimization, Swarm Evol. Comput. 70 (2022) 101041.

[24] J. Zheng, B. Zhang, J. Zou, S. Yang, Y. Hu, A dynamic multi-objective evolutionary algorithm based on Niche prediction strategy, Appl. Soft Comput. 142 (2023) 110359.

[25] L. Yan, W. Qi, A. Qin, S. Yang, D. Gong, B. Qu, J. Liang, Manifold clustering-based prediction for dynamic multiobjective optimization, Swarm Evol. Comput. 77 (2023) 101254.

[26] M. Rong, D. Gong, Y. Zhang, Y. Jin, W. Pedrycz, Multidirectional prediction approach for dynamic multiobjective optimization problems, IEEE Trans. Cybern. 49 (9) (2019) 3362–3374.

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

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

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

[30] R. Azzouz, S. Bechikh, L. Ben Said, Dynamic multi-objective optimization using evolutionary algorithms: a survey, Recent Adv. Evol. Multi-Obj. Optim. (2017) 31–70.

[31] S. Jiang, J. Zou, S. Yang, X. Yao, Evolutionary dynamic multi-objective optimisation: A survey, ACM Comput. Surv. 55 (4) (2022) 1–47.

[32] C. Raquel, 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.

[33] H. Peng, C. Mei, S. Zhang, Z. Luo, Q. Zhang, Z. Wu, Multi-strategy dynamic multi-objective evolutionary algorithm with hybrid environmental change responses, Swarm Evol. Comput. 82 (2023) 101356.

[34] Z. Liu, H. Wang, A data augmentation based Kriging-assisted reference vector guided evolutionary algorithm for expensive dynamic multi-objective optimization, Swarm Evol. Comput. 75 (2022) 101173.

[35] P. Wang, Y. Ma, M. Wang, A dynamic multi-objective optimization evolutionary algorithm based on particle swarm prediction strategy and prediction adjustment strategy, Swarm Evol. Comput. 75 (2022) 101–164.

[36] M. Jiang, Z. Huang, L. Qiu, W. Huang, G.G. Yen, Transfer learning-based dynamic multiobjective optimization algorithms, IEEE Trans. Evol. Comput. 22 (4) (2017) 501–514.

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

[38] X.-F. Liu, Y.-R. Zhou, X. Yu, Cooperative particle swarm optimization with reference-point-based prediction strategy for dynamic multiobjective optimization, Appl. Soft Comput. 87 (2020) 105–988.

[39] P. Wang, Y. Ma, M. Wang, A dynamic multi-objective optimization evolutionary algorithm based on particle swarm prediction strategy and prediction adjustment strategy, Swarm Evol. Comput. 75 (2022) 101164.

[40] R. Liu, J. Li, C. Mu, L. Jiao, et al., A coevolutionary technique based on multi-swarm particle swarm optimization for dynamic multi-objective optimization, European J. Oper. Res. 261 (3) (2017) 1028–1051.

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

[42] S.-Y. Zeng, G. Chen, L. Zheng, H. Shi, H. de Garis, L. Ding, L. Kang, A dynamic multi-objective evolutionary algorithm based on an orthogonal design, in: 2006 IEEE International Conference on Evolutionary Computation, 2006, pp. 573–580.

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

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

[45] B. Xu, Y. Zhang, D. Gong, Y. Guo, M. Rong, Environment sensitivity-based cooperative co-evolutionary algorithms for dynamic multi-objective optimization, IEEE/ACM Trans. Comput. Biol. Bioinform. 15 (6) (2017) 1877–1890.

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

[47] Y. Hu, J. Zheng, S. Jiang, S. Yang, J. Zou, Handling dynamic multiobjective optimization environments via layered prediction and subspace-based diversity maintenance, IEEE Trans. Cybern. 53 (4) (2021) 2572–2585.

[48] L. Chen, H. Wang, D. Pan, H. Wang, W. Gan, D. Wang, T. Zhu, Dynamic multiobjective evolutionary algorithm with adaptive response mechanism selection strategy, Knowl.-Based Syst. 246 (2022) 108–691.

[49] C. Chen, L.-Y. Tseng, An improved version of the multiple trajectory search for real value multi-objective optimization problems, Eng. Optim. 46 (10) (2014) 1430–1445.

[50] R. Agrawal, K. Deb, R. Agrawal, Simulated binary crossover for continuous search space, Complex Systems 9 (1994) 115–148.

[51] C.-K. Goh, K.C. Tan, A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization, IEEE Trans. Evol. Comput. 13 (1) (2009) 103–127.

[52] S. Jiang, S. Yang, X. Yao, K.C. Tan, M. Kaiser, N. Krasnogor, Benchmark problems for CEC2018 competition on dynamic multiobjective optimisation, in: IEEE Congress on Evolutionary Computation (CEC), 2018, pp. 1–18.

[53] R. Chen, K. Li, X. Yao, Dynamic multiobjectives optimization with a changing number of objectives, IEEE Trans. Evol. Comput. 22 (1) (2017) 157–171.

[54] S. Jiang, S. Yang, A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization, IEEE Trans. Evol. Comput. 21 (1) (2017) 65–82.

[55] L. Cao, L. Xu, E.D. Goodman, H. Li, Decomposition-based evolutionary dynamic multiobjective optimization using a difference model, Appl. Soft Comput. 76 (2019) 473–490.

[56] Y. Hu, J. Zheng, J. Zou, S. Yang, J. Ou, R. Wang, A dynamic multi-objective evolutionary algorithm based on intensity of environmental change, Inform. Sci. 523 (2020) 49–62.

[57] Z.-M. Gu, G.-G. Wang, Improving NSGA-III algorithms with information feedback models for large-scale many-objective optimization, Future Gener. Comput. Syst. 107 (2020) 49–69.

[58] P. Paknejad, R. Khorsand, M. Ramezanpour, Chaotic improved PICEA-g-based multi-objective optimization for workflow scheduling in cloud environment, Future Gener. Comput. Syst. 117 (2021) 12–28.

Hu Peng received the B.S. degree in computer science and technology from Hunan Normal University, Changsha, China in 2004, the M.S. degrees in computer technology from the Huazhong University of Science and Technology, Wuhan, China in 2008, and the Ph.D. degree from Wuhan University, Wuhan, China in 2016. He is currently an associate professor with the School of Computer and Big Data Science, Jiujiang University. His research interests include evolutionary algorithms and their applications. Dr. Peng served as a Referee for over 20 international journals, such as the IEEE Transactions on Cybernetics, Information Sciences, Swarm and Evolutionary Computation, Knowledge-Based Systems, International Journal of Bio-Inspired Computation, Applied Soft Computing, and Soft Computing.