A Dynamic Multi-objective Evolutionary Algorithm Using Adaptive Reference Vector and Linear Prediction

Jinhua Zheng[a,b,c], Qishuang Wu[a,b], Juan Zou[a,b], Shengxiang Yang[a,d], Yaru Hu[a,b]

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

Abstract

Responding to environmental changes quickly is a very key component in solving dynamic multi-objective optimization problems (DMOPs). Most existing methods perform well on predicting individuals, but exist some difficulties in improving the accuracy of the predicted population. This paper proposes an approach that predicting the population based on the adjusted reference vector (RVCP) combined with a multi-objective evolutionary algorithm to solve DMOPs. First, the nondominated set is predicted by a linear prediction strategy, which can relocate elite solutions to track the true Pareto set (POS) in the new environment. Second, an adaptive reference-vector-based adjustment strategy is introduced based on the number of nondominated solutions. Then the population in the new environmention is predicted in terms of the adjusted reference vectors, which can track the POS and/or the true Pareto front (POF) more accurately. Finally, a noise-based individual expansion strategy is applied, which can generate variation individuals to keep the population in good diversity. To prove the effectiveness of RVCP, it is compared with five popular dynamic multi-objective evolutionary algorithms (DMOEAs) on twelve test instances with different dynamic characteristics. The experimental results show that RVCP has certain advantages in dealing with DMOPs.

Keywords: Dynamic multi-objective optimization, evolutionary algorithms, prediction, reference vector.

---

*Co-first author: Qishuang Wu, Jinhua Zheng (These authors have contributed equally to this work.)
Corresponding author: Qishuang Wu
Email address: wuqishuang1@foxmail.com (Qishuang Wu)

1. Introduction

As an extension of multi-objective optimization problems (MOPs), dynamic multi-objective optimization problems (DMOPs) [1] involve multiple conflicting objectives (e.g., variable parameters, constraints, etc.) that change over time. This kind of optimization problems has attracted extensive interest due to their many real-world applications. For instance, with the continuous change of power demand, a power generator needs a scheduling algorithm [2] to continuously generate optimal actions to supply energy under constantly changing environments of fluctuating power demand. (i.e., minimizing power generation cost and maximizing generators' stability).

DMOPs need to be solved effectively to meet the needs of real-world applications. There are various types of DMOPs in the world. Without loss of generality, the mathematical form for DMOPs has been defined as follows:

$$\begin{aligned} &\min F(x,t) = (f_1(x,t), f_2(x,t), \ldots, f_m(x,t))\,, \\ &\text{subject to} \;:\quad x \in \Omega_x, t \in \Omega_t, \end{aligned} \tag{1}$$

where $x = (x_1, x_2, ..., x_n)$ is an n-dimensional decision variable vector in the decision space Ω_x; t represents the discrete time instance in the time space Ω_t; $f_i(x,t)$ is an objective function that changes over time t; m is the number of objectives and $F(x,t)$ is the objective function vector in the objective space at time instance t.

Contrast to static multi-objective optimization algorithms (SMOAs) [3], dynamic multi-objective evolutionary algorithms (DMOEAs) usually need to take actions to track the Pareto-optimal set (POS) or Pareto-optimal front (POF) with changes after a environmental change [4] occured, while SMOAs are directly used for MOPs without the environmental change. In DMOEAs, the time instance t is calculated as follows [5]:

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

where n_t represents the number of different steps at unit time; τ signifies the generation counter, and τ_t is the number of generations as the objective function remains the same. The severity and frequency of changes are dependent on n_t and τ_t, respectively.

In the contemporary world, various DMOPs are derived from real-world industrial applications problems, such as industrial electricity scheduling [2][6] and fuel supplementation [7] and distribution systems [8]. In general, DMOEAs usually consist of three key components: change detection, an underlying multi-objective evolutionary algorithm (MOEA) and change response [9].

Recent research indicates that many existing algorithms improve diversity and/or convergence of the population to deal with DMOPs. These algorithms can be divided into three categories:diversity maintenance methods, prediction-based methods, and memory-based methods. The introductions of the part methods are listed in Table 1.

The diversity maintenance methods [10] preserve the diversity of the population during the evolution course, which helps track the POF after the environmental change occurs. Many techniques have been applied to increase the diversity of the population. For example, Liu and Ding [11] put forward a strategy based on reference points to increase population diversity that divides the population into several subpopulations depending on reference points, and the center of the subpopulation is used to predict the center of the corresponding subpopulation in the new environment. Deb and Karthik [10] proposed the DNSGA-II, which has two different versions: DNSGA-II-A and DNSGA-II-B. DNSGA-II-A mainly replaces some old solutions with randomly produced individuals. In contrast, DNSGA-II-B strengthens diversity by using mutated solutions to replace a part of the individuals in the current population. However, giving strength to the diversity sometimes may be difficult to find the POF quickly and accurately.

Prediction-based strategies are effective methods in solving DMOPs. A kind of popular prediction strategy proposed by Li et al. [12] can record the information from past historical experience, then use the special points to analyze the historical information and select the appropriate model based on the historical high-quality solutions to predict the population at the next moment after the environment changes. For example, Zhou et al. [13] proposed a population-based prediction method (PPS) that applied an autoregressive model to predict the POS in the decision space. As long as the POS at the next moment is predicted, the POS manifold moves immediately to the new POS accordingly. Furthermore, Jiang et al. [14] introduced the knee point-based transfer learning method, which generates a high-quality initial population by using estimated knee points in the whole decision space based on transfer learning. However, this method has high computational costs, as it requires a long training time, which is the main obstacle for certain DMOPs.

In addition, some studies divide the entire population into a number of clusters to predict new individuals that are used to solve the time-varying DMOPs well. Rong et al. [19] put forward a multimodel prediction approach in which four prediction models are adopted to predict the POS in the decision space. However, the primary shortcoming of the predictors may lead to a negative influence on the subsequent optimization, which is not enough to deal with dynamic multiobjectivity in a time-varying environment.

Another method [21] combines dual prediction methods based on the quantile, not only tracking the POS by quantile in the decision space, but using quantile to predict

Table 1: The advantage and disadvantage of traditional DMOEAs and other novel methods.

Types of improved methods	Representative Algorithm	Advantage	Disadvantage
Diversity maintenance methods	DNSGA-II [10] DMS [15] DSS [16]	The principle is simple and easy to achieve.	1. Responding to environmental changes randomly may mislead the evolution direction of population; 2. The fitness values of introduced individuals usually are small and convergence speed is slow.
Prediction-based methods	PPS [13] CKPS [17] MOEAD-KF [18] MMP [19] PBDMO [20] NQDPEA [21]	1. The prediction mechanism will be very effective when the prediction results are reliable enough; 2. Learning the distributing characteristics and obtaining a more accurate POS distribution fast.	1. The wrong prediction may be misleading evolution search; 2. Expensive calculation complexity and time complexity.
Memory-based methods	dMOEA/D-L_p [22] DTAEA [23] dCOEA [24]	It is suitable for solving DMOPs with cyclical changes.	1. The problem with irregular change can not well handled; 2. The efficiency will be greatly reduced when the information of the dynamic environment is limited.
Response mechanism based on special model	KT-MOEA/D [14] SVM-DMOEA [25]	Many effective methods are introduced to solve DMOPs	1. High computational costs, as it requires a long training time; 2. The calculation time of different strategy is also different, so the running time is uncertain when the models are uncertain.

the POF in the objective space and then mapping back to the decision space to obtain solutions [26]. Unlike other prediction methods, this method uses information in the decision space and makes full use of the information in the objective space. Moreover, searching in the objective space [27] means a more promising method to find better candidate solutions so as to keep the diversity of the population. Additionally, a method has been put forward to predict the solutions in the kernelized autoencoding model [28], which takes the dynamicity of the objective space and decision space into consideration.

For periodic problems, the memory-based method achieves good performance. Ruan et al. [15] used a memory strategy to predict the position of individuals and accelerate the convergence of the population under environmental changes. Azzouz et al. [29] developed an adaptive memory-based strategy, which makes full use of previous optimal solutions effectively to handle the environment's dynamicity. In short, the memory-based method has the advantage of dealing with periodic problems.

In DMOEAs, the core difficulty is tracking the dynamic POS/POF to respond to the changing environment [30]. For example, some methods may misuse prediction strategies based on inferior information to track individuals in the new environment, which sometimes causes mistakes. In order to avoid this situation, it is advisable to find high-quality individuals that could converge to the POF or/and POS for the next new environmental change. In view of the above characteristics about DMOEAs, a few studies have predicted the individuals by adjusted reference vectors. It is worth noting that the work about reference vectors adjustment in this article is not an improved version based on the research of the reference vector method to deal with DMOPs. They are two quite different mechanisms for solving different types of MOPs. Thus, we propose a dynamic multi-objective evolutionary algorithm using adaptive reference vector and linear prediction (RVCP), which combines three methods to find high-quality individuals for new environments.

(1) The linear prediction model is used to predict the partial population in the new environment, which guides the corresponding nondominated individuals to find the location of the POS.

(2) A reference vector-based adjustment strategy [31] is introduced, which can lead the population towards the POF according to the degree of difficulty of the problem, thereby accelerating the convergence of the population.

(3) A noise-based individual expansion strategy is applied, which can introduce diverse individuals by adding disturbance into the population.

(4) RVCP is compared with five state-of-the-art DMOEAs on twelve different benchmark problems, which is proved to be competitive.

The remainder of this paper is organized as follows: Section 2 introduces the preliminaries and related work. The core strategies of RVCP are presented in Section 3. Section 4 analyzes a series of experimental results in detail with other compared algorithms. Finally, section 5 concludes the paper.

2. Preliminaries

2.1. Dynamic Multi-objective Optimization Problem

$Definition$ 1: **Decision Vector Domination**

At time t, a decision vector x_1 is pareto-dominated by another vector x_2 in the decision space, denoted by $x_2 \succ_t x_1$, if and only if:

$$\begin{cases} \forall i = 1, \ldots, m, \quad f_i(x_2, t) \leq f_i(x_1, t), \\ \exists i = 1, \ldots, m, \quad f_i(x_2, t) < f_i(x_1, t). \end{cases} \tag{3}$$

$Definition$ 2: **Pareto Set, POS**

If a decision vector x^* at time t is suitable for:

$$POS_t = \{x^* \mid \nexists x, x \succ_t x^*\}, \tag{4}$$

then x^* is called the dynamic Pareto-optimal solution, and the set of dynamic Pareto-optimal solutions is called the POS.

$Definition$ 3: **Pareto Front, POF**

At time t, the pareto front (POF) is the corresponding objective vectors of the POS, which satisfies :

$$POF_t = \{F(x^*, t) \mid x^* \in POS_t\}.$$

$Definition$ 4: **Types of DMOPs**

DMOPs may change in a variety of forms (e.g., decision variable changes in the decision space; objective variable changes in the objective space, constraints, etc.). The four types of test problems used in this paper are summarized in Table 2.

Table 2: Different types of DMOPs

Types	POS changes	POF changes	Test problems
Type I	YES	NO	FDA1, FDA4, JY1, JY6
Type II	YES	YES	FDA2, FDA3, JY2, JY3, JY4, JY7
Type III	NO	YES	JY5, JY8
Type IV	NO	NO	–

These four types may occur when the environment changes. However, we generally pay more attention to changes in the top three types. In type IV, the DMOP may have changes in the fitness landscape but without affecting the POS or POF.

2.2. A general framework of dynamic multi-objective evolutionary algorithm

The framework of the dynamic multi-objective evolutionary algorithm is as follows:

(1) Initialize a population; set related parameters.

(2) Detect the environmental change [32]: If an environmental change is detected, the response mechanism is triggered to respond. If no changes are detected, static optimization algorithms are implemented directly.

(3) Apply static optimization algorithms: RM-MEDA [33] is applied to the DMOEA.

(4) Respond to the change: The mechanism is used to respond to environmental changes. There are various response strategies: re-initialization of the population, individual mutation, prediction, population memory methods.

(5) Terminate generation: If the stopping conditions are satisfied, then the optimization algorithm is terminated.

3. Proposed approach

In DMOEAs, the linear prediction strategy is currently one of the most practical and popular prediction strategies. The center point is a special point that can be used to represent the distribution of a group of data to some extent, which may lead a series of elitist individuals closer to the POS. Therefore, we use the center point as the moving direction of the nondominated solutions in previous two population. Then individuals follow the movement direction of the center point to the corresponding location under the environment changes.

3.1. Linear prediction strategy (LPS)

Prediction methods only collect high-quality historical information to predict populations. In the new environment, the previous center points and knowledge are stored to initialize the population, leading the final population to approach the true POF. The details are shown in Algorithm 1.

Therefore we only predict nondominated individuals using the method of feed-forward center points. The method of calculating the center point in this study uses the central point to predict the new individuals. Let Pc_t be the center of the nondominated set Pop_t. $\mid Pop_t \mid$ is the number of nondominated solutions selected by the nondominated sorting algorithm at the end of time t. The equation for calculating the center point is as follows:

$$\mathrm{Pc}_t = \frac{\sum_{x_t \in \mathrm{Pop}_t} x_t}{\left|\mathrm{Pop}_t\right|}. \tag{5}$$

As a result, at time t+1, historical center point information from the last two moments has already been stored in the center point archive C. Then the movement direction ΔPc_t to the new position of the changed center point at time step t+1 can be estimated by the population center points at time t-1 and t. The equation for calculating movement direction is

$$\Delta Pc_t = |Pc_t - Pc_{t-1}|, \tag{6}$$

where ΔPc_t denotes the Euclidean distance between the center points Pc_t at time t and Pc_{t-1} at time $t-1$.

According to ΔPc_t, as long as the environment changes, the strategy will lead nondominated individuals to the new position around the POS. For each x_t, the moving direction [34] should be able to guide the population in good diversity toward promising search regions. The new location of each x_t in the decision space is generated as follows:

$$x_{t+1} = x_t + \Delta Pc_t, \tag{7}$$

where $x_t \in Pop_t$. The x_{t+1} is the predicted solution after (t+ 1) environmental changes.

It is worth noting that boundary check operator is applied to detect whether the variable of the predicted solution is within the given boundary of the decision space. If not, it will be corrected. The boundary detection equation is as follows:

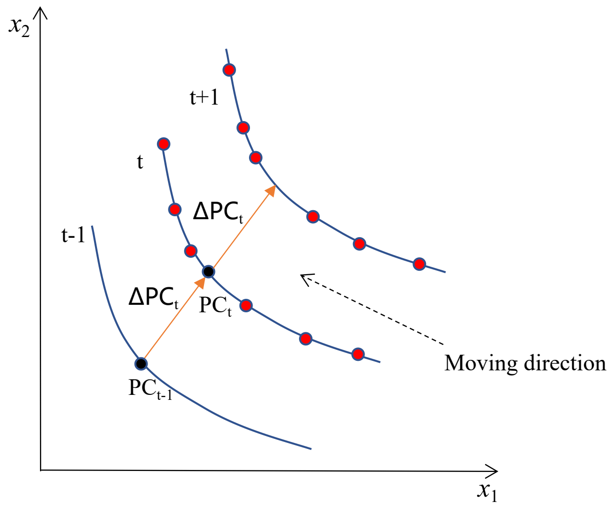

Figure 1: Linear prediction on the decision space.

$$y_i = \begin{cases} x_i & \text{if } l_i \le x_i \le u_i \\ \text{random}(l_i, 0.5(l + u_i)) & \text{if } x_i < l_i \\ \text{random}(0.5(l_i + u_i), u_i) & \text{if } x_i > u_i \end{cases}, \tag{8}$$

where random (a, b) is a random function that can generate a random value between a and b, $i = 1, ..., n$. n denotes the dimensions of the decision space; l and u are the minimum and maximum boundaries of the ith decision space, respectively. Figure. 1

Algorithm 1 : Linear prediction strategy

Input: The whole population of time t is POP_t; the number of the nondominated solutions in the POS at time t is POS_t; the population size is N and the C represents the archive set of the center point.

Output: $POP_{predicted}$

1: POS_t = fast-nondominated-sort (POP_t);
2: Calculate the center point of the POS_t in each dimension according to equation 5, and put it into the set C;
3: Calculate the moving direction of the individuals at time t according to equation 6;
4: Generate new predicted individuals using equation 7 and perform boundary check 8;
5: Save $POP_{predicted}$ and return $POP_{predicted}$.

illustrates the prediction procedure, which mainly uses the center point in black dots to lead the nondominated individuals in red dots to the new POS in the decision space.

3.2. Predict the population based on adaptive reference-vector adjustment strategy (RAS)

To achieve more accurate predictions, this strategy proposes a specific center vector and adjusts the other reference vectors. This method is originally used to solve the high-dimensional problems, but we found that it is an effective way to lead individuals toowards the promising area by adjusted reference vectors. Furthermore, the reference-vector-based adjustment direction depends on the difficulty of test instances, then the adjusted reference vector lead the population. Finally, the linear prediction model is applied to lead the optimized individuals to track the POF in new environments.

Here, uniform reference vectors are applied to generate initial reference vectors. In [35], N represents the number of reference vectors, which is calculated as follows:

$$H = \begin{pmatrix} Q + p - 1 \\ p \end{pmatrix}, \tag{9}$$

where p is the number of divisions at each objective axis predefined by the user and Q denotes the divisions on each dimension.

$$\begin{cases} R_i = \left(r_i^1, r_i^2, \cdots, r_i^m\right), \\ r_i^j \in \left\{\frac{0}{p}, \frac{1}{p}, \cdots, \frac{p}{p}\right\}, \sum_{j=1}^{m} r_i^j = 1, \end{cases} \tag{10}$$

where i =1, 2, ..., H, m denotes the number of objectives, and R represents the collection of reference points.

Our strategy uses a center vector to adjust the surrounding reference vectors. According to the difficulty of test instances, the reference vector is adjusted closer or farther with the center vector adaptively. Under the same optimization generation number, when the number of nondominated individuals in the first layer is sufficient (more than $N/5$), the position of the POS can be predicted quickly by the center point of the first layer of the nondominated solutions more precisely. Under this circumstance, it is promising to move the reference vector away from the central vector, so individuals surrunding the boundary point can be tracked —— to guarantee better distribution of the population.

However, when the test problem to be solved is complicated, the number of nondominated individuals is relatively small (less than $N/5$). Therefore, the number of individuals moving with the movement direction of the center point is insufficient, and the population cannot converge as soon as possible. When the POFs of the test instances are more difficult to track, it is promising to make the reference vector adjust near the center vector in this approach.

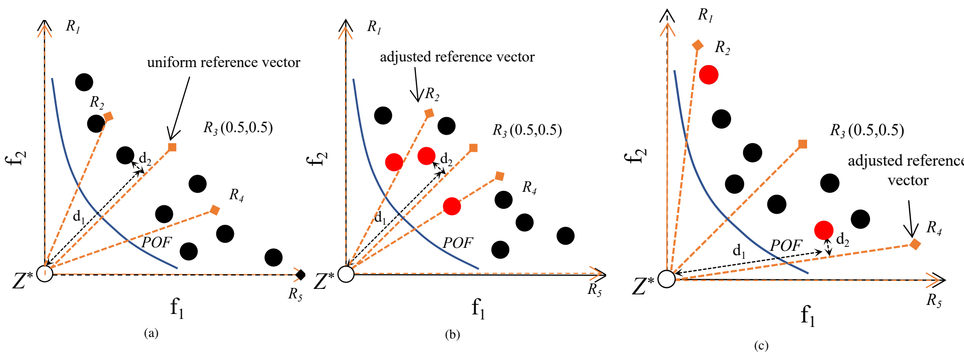

Figure 2: In case (a), the uniform reference vectors are presented in orange dotted lines. In case (b), when the question to be solved is complicated, the reference vectors are closer to the center vectors depending on their similarity, and the individuals selected by using the PBI method are in red. In case (c), when the question to be solved is easy, the reference vectors are farther from the center vectors depending on their similarity, and the individuals selected by the PBI method are in red. R_2 is the specific center vector.

An Adjusting Function (AF) is applied to adjust the initial uniform reference vectors as follows:

$$W_i = \frac{W_i \cdot a_i + C_v \cdot (1 - a_i)}{|W_i \cdot a_i + C_v \cdot (1 - a_i)|}, \tag{11}$$

where $i = 1, 2, . . ., N$, and W_i signifies the ith reference vector. C_v denotes a definite center vector, which can be defined as $C_v = (\underbrace{1/2, \ldots, 1/2}_{M})$ where M means the number of objectives; | | is used to calculates the norm; a_i is the adjusting degree for the ith reference vector, which is formulated as follows:

$$a_i = S_i^k, \tag{12}$$

where $S_i \in [0, 1]$ denotes the degree of similarity between the ith reference vector and the center vector, and k is a control parameter. A positive k means W_i moves closer to C_v, whereas a negative k indicates that W_i will move farther away from C_v. The less distance from W_i to C_v, the higher the degree of similarity (the value of S_i is closer to 0).

$$S_i = \frac{|\{|W_i - C_v|\} - \min\{|W_i - C_v|\}|}{|\max\{|W_i - C_v|\} - \min\{|W_i - C_v|\}|}, i \in \{1, \ldots, m\} \tag{13}$$

After the reference vector is updated according to the similarity of the center vector, we calculate the $g(x|R_i)$ value of each individual based on the PBI method [36]. The PBI method decomposes a multi-objective optimization problem into several singleobjective optimization problems, and its definition is as follows:

$$\begin{cases} \text{minimize } g\left(x \mid R_i\right) = d_1\left(x \mid R_i\right) + \theta d_2\left(x \mid R_i\right), \\ \text{s.t. } x \in \Omega, \end{cases} \tag{14}$$

where

$$d_1\left(x \mid R_i\right) = \frac{\left\|\left(F(x) - z^*\right)^T R_i\right\|}{\|R_i\|}, \tag{15}$$

$$d_2\left(x \mid R_i\right) = \left\|F(x) - \left(z^* + d_1\left(x \mid R_i\right) \times \frac{R_i}{\|R_i\|}\right)\right\|, \tag{16}$$

where $z^* = (Minf_1(\cdot), ..., Minf_m(\cdot))$ is the ideal point.

Algorithm 2 : Predict the population based on adaptive reference-vector adjustment strategy

Input: The population of time t is POP_t; C represents the archive set of the center point
Output: POP_{rvc}

Generate uniformly distributed reference vectors $W = (R_1, R_2, ..., R_N)$ by equation 10;
Calculate the number of solutions in the first layer of nondominated set, ND;
// Update reference vector
for R_i in R **do**
    Calculate the similarity of each reference vector R_i and center vector C_v, then update R_i;
end for
// Utilize PBI method
for R_i in R **do**
    for x_j in POP_t **do**
        Compute g($x_j|R_i$) for x_j according to R_i;
    end for
    The individual x_j with the minimum g($x_j|R_i$) is assigned to R_i;
    Delete the individual x_j from POP_t;
end for
Crossover two individuals on ajacement reference vectors to obtain POP_t;
Calculate the global center point and the direction using equations 5 and 6;
Generate new predicted individuals POP_{rvc} and perform boundary check equation 8;
Return POP_{rvc}

Then sort all individuals according to the g($x|R_i$) value. For each adjusted reference vector, an individual with the smallest g($x|R_i$) value is selected. Finally, we crossover the two individuals of two adjacent reference vectors to obtain a new population POP_t. The obtained individuals will either be as close to the center vector as possible or will be far from the center vector. Figure 2 depicts how to select one individual for every reference vector according to the PBI method. Finally, we attempt to map all obtained solutions back to the decision space and predict new solutions. Calculate the center point of the POP_t and the movement direction of the POP_t by equation 5 and 6. These generated individuals are stored to POP_{rvc}, and the algorithm returns POP_{rvc}. Here we still use a linear prediction strategy in the first strategy, which is beneficial to compensate for the insufficiency of the prediction and guide the population to produce excellent solutions. The linear prediction strategy in this method plays a similar role as the first strategy, which makes use of the moving direction of the center point so that predicted solutions close to the new POS. The difference is that the center point at t and t-1 are calculated by the entire population in this strategy. The individuals obtained through the adjusted reference vector, in this case, the new solutions obtained by the prediction strategy will be close to the true POF.

3.3. Noise-based individual expansion strategy (NES)

The first two strategies focus on predicting the individuals in the new environment to improve the convergence speed. In order to exploit the diversity of the population, introducing noise variation positively impacts the global diversity of the algorithm, which makes algorithms have better robustness. The noise-based individual expansion strategy is used to produce the different individuals. Several diverse solutions are generated to enhance the population's diversity. The detailed procedures are described in Algorithm 4. After each environmental change, individuals with a specific number can be obtained by Eq.17:

$$\hat{x}_j = x_j^{\min} + \alpha \left(x_j^{\max} - x_j^{\min} \right), \text{ for } j = 1, \dots, n, \tag{17}$$

where $x_j^{\min}$ and $x_j^{\max}$ are the lower and upper boundaries of the decision space, respectively; and α denotes a random variable associated with a uniform distribution.

Algorithm 3 : Noise-based individual expansion strategy

Input: the current population, P_{mixed}; the population size N;
Output: the expanded population, P_3;
1: Set $P_{new} = \emptyset$;
2: $u \leftarrow$rand(0, 1) %sample u from $U(0, 1)$;
3: $L = u * N$;
4: **for** $i = 1 : L$ **do**
5: $x \leftarrow$ Randomly generate an individual using equation 17 ;
6: Put x into P_{new};
7: **end for**
8: P_3= $P_{mixed} \cup P_{new}$;

Specifically, for random noise, 10% of the individuals are randomly chosen from the mixed set P_{mixed} of $POP_{predicted}$ and POP_{rvc}, then a number u is generated from a uniform distribution $U(0, 1)$. Multiple the number of the population N and u to obtain an integer L, then generate L individuals using equation(17).

3.4. Overall Algorithm

First, the population is randomly initialized and the initial time is set to 0. The center point set of the objective function value and the sub-strategies sets are set to empty (Line 1 of algorithm 4). When the environmental change is detected, the center point of nondominated solutions at time t is calculated, and the center point is stored in PC_t (Lines 3-4 of algorithm 4). The linear prediction strategy generates new solutions $POP_{predicted}$ based on historical information.

Algorithm 4 : The overall framework of the algorithm

Input: N (population size), τ_t (the number of iterations where t remains fixed), n_t (the number of distinct steps at unit time), the stopping criterion
Output: The final population, P;
1: Initialize population POP, $t = 0$;
2: **while** the stopping criterion is not met **do**
3: **if** change is detected **then**
4: PC_t←Calculate the center point of POP at time t;
5: // the following sub-strategies in RVCP (change response mechanism) are the Linear prediction strategy (line 6), the guiding individual-based prediction strategy (line 7) and noise-based individual expansion strategy (line 9) respectively.
6: $POP_{predicted}$ ← **Linear prediction strategy** (); // refer to Algorithm 1
7: POP_{rvc} ← **Reference-vector-based Prediction** (); // refer to Algorithm 2
8: Incorporate POP_{rvc} and $POP_{predicted}$ into $Main_{pop}$:= $POP_{rvc} \bigcup POP_{predicted}$;
9: P_3 ← **Noise-based individual expansion strategy**(); // refer to Algorithm 3
10: P ←Obtain a new population from P_3 by Environmental selection;
11: **else**
12: Optimize the population using RM-MEDA;
13: **end if**
14: Iter := Iter + 1;
15: **end while**

Then the number of individuals in the first layer of nondominated solutions is used to determine the reference-vector-based adjustment direction(Line 6 of algorithm 4). Suppose the number of nondominated solutions accounts for less than $N/5$. In that case, the reference vector will get closer to the specific center vector so that individuals will be produced near the center vector to make up for the insufficiency of solutions produced by algorithm 1. On the contrary, if the number of nondominated solutions is more than $N/5$, the reference vector will be farther from the specific center vector and near the boundary. After updating the reference vectors, the PBI method is applied to bind the population to the adjusted reference vectors. A crossover operator is applied to mating two individuals with adjacent vectors one by one, and then a linear prediction strategy 1 is applied to obtain the new population POP_{rvc} (Line 7 of algorithm 4). $POP_{predicted}$ and POP_{rvc} are combined and choose individuals randomly from the mixed population and generate some variation individuals, then new individuals set $P3$ is produced by the noise-based individual expansion strategy. Finally, the algorithm will use the nondominated sorting of NSGA-II called ND-Selection [37] to select N solutions in $P3$ and return P. The new population is formed by selecting the solution with the lower rank when comparing two solutions of different nondominated levels and preferring the solution located in a lesser crowded region between two solutions having the same rank. The main purpose of this selection strategy is to allow the individuals that dominate others and contribute more to the maintenance of the population diversity. If there is no environmental change, the population is optimized by RM-MEDA (Line 12 of algorithm 4). This algorithm terminates when the stopping criterion is satisfied and the final number of obtained individuals in both cases is N. The parameter settings about the experiments are shown in Table 3.

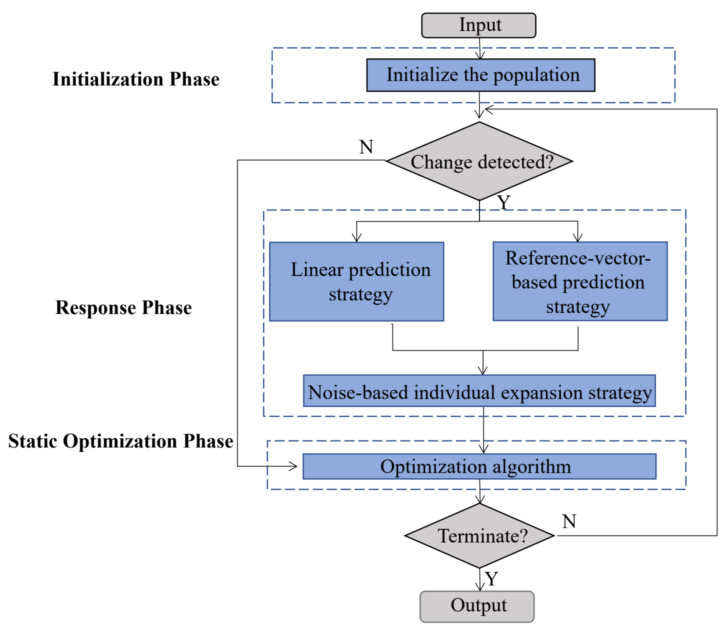

Figure 3: General framework of RVCP.

The basic framework of the RVCP algorithm is presented in Fig 3. The algorithm uses three key strategies to obtain a new population in response to environmental changes. The static optimization method is employed regardless of whether the environmental changes occur. Here RM-MEDA is applied as the underlying MOEA optimizer.

Table 3: Parameter settings of the experiments

Number of decision variables, n	10 for all test problems
Population size	100 for all test problems
Severity of change	$n_t = 10$
Frequency of change	$\tau_t = 5, 10, 20$
p in 9	24 and 9 for bi- and tri-objective problems
θ in 14	1/6

3.5. Computational complexity of the compared algorithms and RVCP

To demonstrate computational resources of the compared algorithms and RVCP, the computational complexity of each optimization algorithm and RVCP are analyzed as follows:

Set M, N, D to be the dimension of the objective space, population size, and the dimensions of decision variables, respectively.

(1) CKPS: The computational resource in CKPS is mainly spent in the process of predicting the new individuals by using the feed-forward center points, which requires $O(MN^2)$ computational complexity. Therefore, the CKPS's computational complexity is $O(MN^2)$.

(2) PBDMO: The computational complexity of PBDMO is composed of three parts: prediction, sampling and shrinking. The sampling takes the most time complexity $O(MN^2)$, which is because it needs to separate variables into principle and non-principle variables. Therefore, the overall computational complexity of PBDMO is $O(MN^2)$.

(3) PPS: PPS chooses RM-MEDA as the static optimizer. In RM-MEDA, the computational complexity of RM-MEDA is determined by modeling, reproduction and the selection operator. The modeling cost is $O(DN)$. The reproduction spends $O(DK)$ and K is the number of clusters. The manifold prediction costs $O(DN^2)$ computational complexity. Therefore, the overall computational complexity is $O(DN^2)$.

(4) DMS: The computational complexity of DMS is mainly determined by three parts. The prediction strategy in DMS requires $O(MN^2)$ computational complexity, and the gradual search strategy takes $O(N)$ computational complexity. Then computational complexity of the overall framework of DMS is mainly spent on the nondominated sort. Therefore, the overall computational complexity is $O(MN^2)$.

(5) HPPCM: The computational complexity of the center point prediction in HPPCM is $O(DN)$; the guiding individual-based prediction strategy costs $O(DN^2)$ computational complexity, so the two parts take $O(DN^2)$ computational complexity. The third strategy, precision controllable mutation strategy needs $O(N)$. Therefore, the computational complexity of HPPCM is $O(DN^2)$.

(6) RVCP: The overall computational complexity of RVCP is mainly determined by the the change response mechanism. The change response mechanisms includes three parts: the time complexity of calculating the nondominated set in algorithms 1 is $O(MN^2)$, because all of the individuals in population are nondominated with respect to each other at worst; and the time complexity of boundary check of Line 13 in algorithm 1 is O (MD); The complexity of the adaptive reference-vector adjustment strategy module requires $O(MN^2)$, there is a key operation that needs to calculate similarity value of each individual and bind the reference vector with the minimum value for each individual, which takes $O(MN^2)$; the noise-based individual expansion strategy takes $O(N)$ computational complexity. The computational complexity of the overall framework of RVCP is mainly spent on the nondominated sorting and adjusting the reference vectors. Because the largest time complexity is $O(MN^2)$ in these three operators, the overall complexity of RVCP is $O(MN^2)$.

In general, it can be found that the computational complexity values of RVCP are not very significant with other compared algorithms, which is because most algorithms use the nondominated sorting. When our population objective number is M, the population size is N. For nondominated sorting, the first internal loop is N times, because each individual in the population can become the member of at most one Pareto front; the second internal loop is that each individual can perform at most (N- 1) comparison, and each dominated check needs at most M comparisons , so the operator requires the overall $O(MN^2)$ calculation.

4. Experimental study

This section introduces the benchmark problems, performance metrics, parameter settings and analysis of experimental results. To verify the effectiveness of the dynamic multi-objective evolution optimization processes, the experimental results of five state-of-the-art dynamic multi-objective algorithms are compared. All experiments are performed on the MATLAB platform.

4.1. Test Instances

We used 12 benchmarks, i.e. FDA1-FDA4 [1] and JY1-JY8 [38]. JY1-JY8, whose properties are nonmonotonic or/and time-varying or/and random in type change. The second group of test questions is FDA1-FDA4, whose characteristics are nonconvexity or/and disconnectedness or/and being time-dependent.

4.2. Comparison algorithms

Five popular DMOEAs with different parameter setting principles were compared to RVCP. They are the prediction strategies for dynamic multi-objective optimization (PBDMO)[20] algorithm, population prediction strategy (PPS) algorithm [13], a prediction strategy based on center points and knee points (CKPS) [17] algorithm, a hybrid prediction strategy and a mutation strategy (HPPCM) [39] algorithm and the diversity maintenance mechanism (DMS) [15] algorithm.

PBDMO is a prediction-based algorithm [20]. PPS [13] and DMS [15] are representative algorithms of metaheuristics. CKPS [17] uses center points and knee points to predict the population in different ways. HPPCM [39] is the latest algorithm in the field of DMOEA, which take advantage of the prediction strategy and the controllable mutation strategy to respond to the environmental changes. HPPCM and RVCP use different mechanisms for dealing with DMOPs. HPPCM proposed an individual-based prediction strategy to guide the population. In contrast, RVCP uses adjustable reference vectors to optimize individuals and predict the population for responding to the change.

4.3. Performance Indicators

There are many indicators applied to evaluate the performance of the algorithm, like Inverted Generational Distance (IGD) [40][41] and Hypervolume Difference [42]. Two indicators [43] were used to evaluate the performance of the six algorithms and another indicator (MSP) [24] is applied to the performance of the algorithms' each components.

1) Mean Inverted Generational Distance (MIGD): The inverted generational distance (IGD) is widely adopted to evaluate the performance of the algorithms [16], which could investigate the performance of algorithms deeply. Suppose POF_t is a set of uniformly distributed solutions of the POF in time t and P_t is a POF approximation at time t. The definition of IGD is as follows:

$$\mathrm{IGD}\left(\mathrm{POF}_t, \mathrm{P}_t\right) = \frac{\sum_{v \in \mathrm{POF}_t} d\left(v, \mathrm{P}_t\right)}{|\mathrm{POF}_t|}, \tag{18}$$

where $d(v, P_t) = \min_{u \in P_t} \sqrt{\sum_{j=1}^{m} \left(f_j^v - f_j^u \right)^2}$ denotes the minimum Euclidian distance between v and the obtained points in P_t, and $|POF_t|$ is the cardinality of solutions in POF_t .

In DMOEAs, it is difficult to evaluate which algorithm is better in the dynamic environment by merely using IGD. The mean inverted generational distance (MIGD) is modified from IGD and is applied to assess the convergence and diversity of the obtained solutions. When the MIGD value is smaller, the algorithm's performance is better. MIGD is calculated as:

$$\mathrm{MIGD} = \frac{\sum_{t \in T} \mathrm{IGD}\left(\mathrm{POF}_t, \mathrm{P}_t\right)}{|T|}, \tag{19}$$

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

2) Hypervolume Metric (MHV): The hypervolume (HV) is used to compare the gap between the hypervolume of the true POF and the obtained POF. Suppose POF_t is a set of uniformly Pareto optimal solutions of the P_t at time t and P_t is a POF approximation at time t. HV is defined as follows:

$$\mathrm{HV}\left(\mathrm{POF_t}, \mathrm{P_t}\right) = \mathrm{HV}\left(\mathrm{POF_t}\right) - \mathrm{HV}\left(\mathrm{P_t}\right), \tag{20}$$

where $\mathrm{HV}(S)$ refers to the HV of the set S.

MHV denotes the average value of HV in several time steps in an environmental change, which is a comprehensive indicator to evaluate convergence and distribution. When the MHV value is greater, the algorithm's performance is better. The calculation of the MHV can be expressed as follows:

$$MHV = \frac{1}{|T|} \sum_{t \in T} HV\left(PF_t, P_t\right), \tag{21}$$

where T is a set of discrete time points over a run and $|T|$ is the cardinality of T. The reference point for computing hypervolume indicates $(Z_{t1} + 0.5, Z_{t2} + 0.5, ..., Z_{tM} + 0.5)$, where M represents the number of objectives, and Z_{ti} is the maximum value of the ith objective of the true POF at time t.

3) Mean Schott's Spacing Metric (MSP): The Schott's spacing (SP) metric [24] [44] aims to evaluate the distribution of the obtained POF_t^{ob}. The metric is given as

$$\mathrm{SP}\left(\mathrm{POF}_t^{ob}\right) = \sqrt{\frac{1}{\left|\mathrm{POF}_t^{ob}\right| - 1} \left(\sum_{i=1}^{\left|\mathrm{POF}_t^{ob}\right|} \left(D_i - \bar{D}\right) \right)}, \tag{22}$$

where D_i is the Euclidean distance between the ith point in POF_t^{ob} and its nearest point in POF_t^{ob} . $\overline{D}$ represents the average value of D_i. When the MSP value is smaller, the algorithm's performance is better. The mean Schott's spacing metric (MSP) can be defined as follows:

$$\mathrm{MSP} = \frac{\sum_{t \in T} \mathrm{SP}\left(\mathrm{POF}_t^{ob}\right)}{|T|}. \tag{23}$$

4.4. Experimental results and analysis

All experiments were run 20 times on all test problems. The values of MIGD, MHV and standard deviations were computed, and the standard deviation is bracketed after the values of MIGD and MHV. To make the results of the experiment more distinct in comparing these algorithms, the best performance in each test issue is bold and the background color is gray to emphasize. In the experiments, τ_t is set to 5, 10 and 20; n_t is fixed to 10. Each test problem has three change types and there were 36 results for MIGD and MHV values as shown in Table 4 to 7. The results were tested by the Wilcoxon rank-sum test with a significance level of 0.05. [45]. The test is performed to investigate the results obtained by different algorithms effectively.

4.4.1. Results on FDA problems

Table 4 shows the MIGD results for six algorithms. RVCP performed well in some cases. With the increase of τ_t on FDA1, RVCP performed slightly worse than HPPCM, while RVCP has stable tracking ability and the standard deviation is pretty small. The MIGD of HPPCM is considerable on FDA1, which indicates the hybrid prediction strategy and the mutation strategy make the HPPCM converge faster.

Table 4: MIGD indicators of six algorithms on FDA1-4

Problem	(τ_t, n_t)	CKPS	PBDMO	DMS	PPS	HPPCM	RVCP
FDA1	(5,10)	9.6801e-3 (2.08e-3)	1.4347e-1 (3.08e-2)	1.5848e+1 (2.70e+0)	1.7743e-2 (2.66e-2)	**5.2062e-3 (1.15e-3)**	4.0993e-2 (2.52e-2)
	(10,10)	5.6559e-3 (7.50e-5)	6.6535e-3 (2.98e-4)	7.4939e+0 (1.40e+0)	7.3945e-1 (1.23e-1)	**2.2972e-3 (4.67e-4)**	2.0332e-2 (3.11e-2)
	(20,10)	**2.6233e-4 (9.25e-6)**	3.7592e-3 (1.47e-4)	3.4164e-1 (1.44e-1)	7.4740e-4 (3.74e-5)	6.4409e-4 (4.09e-5)	6.2641e-4 (3.22e-4)
FDA2	(5,10)	2.4421e-3 (3.73e-4)	7.0839e-2 (1.43e-2)	5.2985e-3 (5.73e-4)	6.2856e-3 (2.41e-3)	2.3413e-3 (1.19e-4)	**2.0609e-3 (1.09e-4)**
	(10,10)	1.7592e-3 (5.40e-5)	1.7688e-3 (2.70e-5)	1.7191e-3 (1.62e-4)	2.1340e-3 (3.98e-4)	1.7270e-3 (4.55e-5)	**1.6731e-3 (2.81e-5)**
	(20,10)	1.6335e-3 (4.50e-5)	1.7705e-3 (1.50e-5)	1.6219e-3 (2.78e-5)	1.6453e-3 (1.23e-5)	1.6111e-3 (1.35e-5)	**1.5842e-3 (2.46e-5)**
FDA3	(5,10)	4.8017e-2 (1.34e-2)	6.4535e-2 (1.24e-2)	5.2997e-1 (4.53e-2)	2.5131e-1 (9.31e-2)	3.5794e-2 (8.21e-3)	**1.6434e-2 (5.39e-3)**
	(10,10)	8.4209e-3 (2.29e-3)	1.5820e-2 (2.97e-3)	3.9974e-1 (1.49e-1)	5.1032e-2 (6.28e-2)	**7.0785e-3 (3.36e-3)**	9.0120e-3 (2.27e-3)
	(20,10)	7.4444e-3 (3.91e-3)	5.3706e-3 (2.51e-3)	3.6844e-2 (1.22e-2)	7.8238e-3 (1.56e-3)	**7.1625e-4 (1.38e-4)**	6.1133e-3 (9.62e-4)
FDA4	(5,10)	8.7012e+0 (1.63e-1)	1.2884e+1 (1.68e+0)	1.0050e+1 (2.02e+0)	1.1484e+1 (4.84e-1)	**7.4933e+0 (3.32e-2)**	5.2785e+1 (8.34e+0)
	(10,10)	1.0134e+1 (6.42e-2)	9.8406e+0 (2.72e-2)	1.2981e+1 (3.43e+0)	2.3842e+1 (2.50e-1)	**9.0652e+0 (2.88e-2)**	1.1477e+1 (7.16e-2)
	(20,10)	1.0267e+1 (7.52e-2)	9.8170e+0 (1.30e-2)	**9.3704e+0 (1.92e-1)**	1.0997e+1 (1.54e-1)	1.0064e+1 (1.80e-2)	1.1407e+1 (8.30e-2)

RVCP has advantages over the other algorithms on FDA2. Compared with CKPS, RVCP adopts a reference-vector-adjustment strategy to accelerate the convergence speed

Table 5: MHV indicators of six algorithms on FDA1-4

Problem	(τ_t, n_t)	CKPS	PBDMO	DMS	PPS	HPPCM	RVCP
FDA1	(5,10)	7.5448e-1 (1.35e-2)	5.5308e-1 (3.51e-2)	1.8422e-1 (2.48e-2)	7.6953e-1 (1.94e-2)	7.8528e-1 (5.09e-3)	7.5907e-1 (6.66e-3)
	(10,10)	8.5909e-1 (7.94e-4)	7.9381e-1 (8.49e-4)	4.9482e-1 (5.65e-2)	8.4967e-1 (5.70e-4)	8.4706e-1 (7.50e-4)	8.2971e-1 (3.10e-3)
	(20,10)	8.2891e-1 (3.81e-4)	7.9383e-1 (8.05e-4)	4.8791e-1 (5.44e-2)	8.3905e-1 (9.79e-4)	8.4686e-1 (9.62e-4)	8.4911e-1 (5.35e-3)
FDA2	(5,10)	1.1265e+0 (2.04e-3)	8.8676e-1 (7.65e-2)	1.1115e+0 (6.69e-3)	1.0973e+0 (1.28e-2)	1.1241e+0 (4.94e-3)	1.1351e+0 (2.70e-3)
	(10,10)	1.1311e+0 (4.87e-4)	1.1386e+0 (6.94e-5)	1.1296e+0 (3.62e-4)	1.1338e+0 (5.20e-4)	1.1381e+0 (3.56e-4)	1.1427e+0 (5.12e-4)
	(20,10)	8.5891e-1 (3.81e-4)	7.9383e-1 (8.05e-4)	4.8791e-1 (5.44e-2)	8.4905e-1 (9.79e-4)	8.4686e-1 (9.62e-4)	1.1340e+0 (7.43e-4)
FDA3	(5,10)	5.6902e-1 (5.32e-2)	5.9167e-1 (9.72e-2)	2.8183e-1 (2.47e-2)	5.1192e-1 (1.75e-1)	6.7449e-1 (6.34e-2)	6.9249e-1 (2.29e-2)
	(10,10)	8.7633e-1 (9.38e-3)	9.7838e-1 (8.03e-3)	7.1757e-1 (1.75e-2)	8.1916e-1 (2.36e-3)	1.5176e+0 (7.76e-3)	7.8142e-1 (6.44e-3)
	(20,10)	8.7203e-1 (4.84e-3)	9.8192e-1 (4.81e-3)	7.3095e-1 (1.25e-2)	8.1105e-1 (1.16e-2)	8.4900e-1 (4.66e-3)	8.3621e-1 (8.32e-3)
FDA4	(5,10)	3.1454e-1 (1.96e-2)	1.1807e-1 (1.28e-2)	8.9565e-2 (4.89e-3)	1.6075e-1 (2.21e-2)	3.8766e-1 (8.35e-3)	1.8916e-1 (8.07e-3)
	(10,10)	3.7631e-1 (2.67e-3)	3.0133e-1 (2.88e-3)	2.0667e-1 (9.93e-3)	2.5612e-1 (8.12e-3)	4.0338e-1 (1.81e-3)	2.4539e-1 (5.44e-3)
	(20,10)	3.8259e-1 (3.35e-3)	3.0058e-1 (3.48e-3)	2.0358e-1 (1.65e-2)	2.5411e-1 (7.98e-3)	4.0561e-1 (1.51e-3)	3.2618e-1 (6.15e-3)

and the distribution, which makes RVCP perform better than CKPS no matter how frequent the environmental change is. HPPCM was slightly inferior to RVCP on FDA2, but it was superior to PBDMO and PPS on this test. The reason is that the POS of FDA2 remains fixed and HPPCM's spontaneous evolution strategy can track the moving direction well when the environment changed.

RVCP has advantages over the other algorithms on FDA2. Compared with CKPS, RVCP adopts a reference-vector-adjustment strategy to accelerate the algorithm's convergence speed and the distribution, which likely makes RVCP perform better than CKPS no matter how frequent the environmental change is. The prediction method and gradual search for DMS make effects on finding the varying POSs of FDA2, indicating some well-distributed solutions produced by DMS can track time-varying decision variables on FDA2 over time. In FDA3, the environmental change not only shifts the POS but influences the density points distribution on the POF. Faced with the characteristic of FDA3, when the environment changes quickly, RVCP's adaptive strategy demonstrated its superiority compared to the other algorithms. Both HPPCM and CKPS perform better than RVCP when τ_t= 10 and 20, and RVCP performs steadily to cope with random changes in FDA3. In FDA4 RVCP loses to other algorithms, which means the performance still needs to be improved.

In order to evaluate the accuracy of RVCP, we record the MHV value to judge the convergence and distribution of the algorithm in Table 5. RVCP can get the better results in solving some test problems. Due to the complex shape of some POF, RVCP could not obtain the best result, but still better than other algorithms a lot.

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

Problem	(τ_t, n_t)	CKPS	PBDMO	DMS	PPS	HPPCM	RVCP
JY1	(5,10)	3.5686e-1 (4.86e-2)	4.3868e-1 (7.92e-3)	4.2747e-1 (1.28e-1)	4.3894e-1 (3.37e-1)	2.3899e-1 (7.42e-2)	4.7469e-2 (1.59e-2)
	(10,10)	6.6545e-2 (4.63e-2)	1.2593e-1 (1.14e-2)	1.2007e-1 (1.82e-2)	6.6283e-1 (4.66e-1)	1.2220e-1 (1.56e-2)	2.0216e-2 (1.23e-2)
	(20,10)	1.8333e-1 (2.56e-2)	3.2801e-2 (7.35e-3)	5.8370e-2 (7.34e-3)	9.7583e-1 (3.85e-1)	5.7565e-2 (2.24e-3)	1.8441e-2 (2.35e-3)
JY2	(5,10)	6.5175e-2 (1.38e-2)	4.0780e-1 (1.56e-3)	8.7639e+0 (2.50e+0)	2.3255e-2 (1.20e-2)	5.7379e-2 (1.30e-2)	5.6800e-2 (2.41e-2)
	(10,10)	5.1877e-2 (2.08e-2)	3.2821e-2 (8.79e-4)	2.0484e+0 (3.67e-1)	1.3158e-2 (1.34e-2)	9.8232e-2 (3.69e-4)	5.5091e-3 (2.30e-3)
	(20,10)	1.5679e-2 (1.65e-3)	6.3025e-3 (2.60e-4)	4.4261e-2 (2.67e-2)	1.7693e-1 (1.69e-1)	1.7320e-3 (1.88e-4)	6.5236e-4 (1.87e-4)
JY3	(5,10)	8.9306e-2 (8.70e-3)	4.0966e-2 (3.50e-3)	8.9033e-3 (1.76e-3)	8.5445e-2 (2.18e-2)	1.1813e-2 (6.55e-3)	8.6008e-3 (7.91e-4)
	(10,10)	8.4504e-2 (1.21e-2)	1.5763e-3 (4.58e-4)	1.0800e-3 (3.31e-4)	6.4962e-2 (1.34e-2)	3.3951e-3 (1.31e-3)	8.3579e-3 (1.19e-3)
	(20,10)	7.4294e-2 (1.56e-2)	1.0011e-3 (2.58e-4)	6.9241e-4 (1.67e-4)	5.4365e-2 (2.51e-2)	2.0273e-3 (2.49e-3)	4.6702e-3 (1.43e-3)
JY4	(5,10)	2.6381e-1 (1.39e-1)	1.9230e-1 (8.64e-2)	3.3366e+0 (1.75e+0)	1.2538e-1 (1.22e-1)	8.1637e-2 (2.19e-2)	6.1296e-3 (1.81e-3)
	(10,10)	6.9842e-1 (1.95e-1)	3.0125e-2 (4.49e-3)	1.6572e-1 (8.65e-2)	1.5893e-1 (1.16e-1)	2.8447e-2 (6.56e-3)	3.9017e-3 (1.52e-3)
	(20,10)	4.6347e-1 (1.60e-1)	2.6241e-2 (2.48e-3)	2.7826e-2 (2.37e-3)	7.4622e-1 (1.64e-1)	1.4360e-2 (1.96e-3)	5.9747e-3 (2.55e-3)
JY5	(5,10)	7.4613e-2 (1.23e-2)	2.8143e-2 (3.89e-2)	7.2979e-3 (1.64e-3)	3.8826e-2 (1.39e-2)	8.1088e-3 (2.86e-3)	1.8387e-3 (5.57e-4)
	(10,10)	8.2113e-3 (1.81e-3)	4.5845e-4 (9.12e-5)	1.3168e-3 (2.58e-4)	1.4427e-2 (1.28e-2)	1.1329e-3 (3.60e-4)	1.4476e-3 (4.75e-4)
	(20,10)	2.5729e-3 (1.05e-3)	3.5681e-4 (7.60e-5)	5.6688e-4 (1.20e-4)	7.2173e-3 (7.47e-3)	2.5192e-4 (4.92e-5)	2.4752e-3 (1.27e-3)
JY6	(5,10)	4.1052e+1 (1.29e+1)	4.1400e+0 (7.06e-1)	1.1706e+2 (2.88e+1)	1.7111e+1 (2.22e+0)	7.8239e+0 (4.76e-1)	3.9907e+0 (4.94e-1)
	(10,10)	1.9847e+1 (2.33e+0)	4.1947e+0 (4.19e-1)	5.2525e+1 (5.94e+0)	1.1535e+1 (1.53e+0)	6.3655e+0 (3.18e-1)	3.3212e+0 (4.29e-1)
	(20,10)	1.2629e+1 (1.03e+0)	4.0356e+0 (5.28e-1)	2.7128e+1 (1.52e+0)	6.3008e+0 (6.76e-1)	4.6263e+0 (2.20e-1)	2.0661e+0 (1.42e-1)
JY7	(5,10)	1.2980e+1(2.17e+1)	1.2069e+1 (9.60e+0)	2.2408e+2 (1.67e+1)	8.7518e+1 (7.80e+1)	1.1660e+2 (1.67e+1)	1.0958e+1 (6.90e+0)
	(10,10)	3.5410e+0 (2.00e+1)	7.5319e+0 (8.73e-1)	2.3004e+2 (2.88e+1)	1.2801e+1 (7.91e+0)	9.3718e+1 (1.85e+1)	3.3159e+0 (4.46e+0)
	(20,10)	4.1103e+0(3.62e+1)	7.6062e+0 (1.23e+0)	1.6840e+2 (1.13e+1)	3.9418e+0 (1.39e+0)	1.8745e+1 (6.80e+0)	3.6132e+0 (6.58e+0)
JY8	(5,10)	1.0951e-1 (1.73e-2)	2.6478e-2 (5.01e-3)	3.3614e-2 (7.85e-3)	5.2253e-2 (1.90e-2)	3.5807e-2 (6.51e-3)	2.3051e-2 (7.46e-4)
	(10,10)	2.2983e-2 (1.94e-3)	2.3690e-2 (2.47e-4)	2.4424e-2 (1.81e-3)	4.3924e-2 (1.10e-2)	1.9936e-2 (5.26e-4)	1.9457e-2 (4.83e-4)
	(20,10)	1.8232e-2 (3.34e-4)	1.8162e-2 (1.84e-4)	2.1147e-2 (2.54e-4)	3.0783e-2 (1.01e-2)	1.7297e-2 (6.24e-5)	1.9352e-2 (7.92e-4)

4.4.2. Results on JY problems

The JY test suite has been used to assess the comprehensive performance of DMOEAs [46] [47]. Here, each test problem corresponds to the three cases where the value of τ_t equals 5, 10 and 20, respectively. Tables 6 and 7 show that RVCP has smaller MIGD values and larger MHV values on JY1-JY8 than the other algorithms. Next we analyze the reasons for this result.

The POS of JY1 presents periodical changes, while its POF is fixed and contains the convex and concave portion of the figure, leading to difficulty in tracking the true POF. RVCP has the ability to track the varying POS well and acquired the best MIGD results on JY1, indicating that RVCP can track the POS stably. JY2 belongs to type II, and both of its POF and POS varies over time. Similar to JY1, it has concave and convex portions of the POF. RVCP performs better than the other algorithms on JY2 (except $\tau_t = 5$), benefiting from the proposed linear prediction strategy the reference-vector-based adjustment strategy. When $\tau_t = 5$, the effect of PPS is better except RVCP, which means that PPS is able to predict the population well when the environment changes fast. Since the POS of JY2 changes over time in a weakly-nonlinear pattern, both the linear prediction model proposed by us and HPPCM can improve the tracking ability of the algorithm.

Table 7: MHV indicators of six algorithms on JY test suites

Problem	(τ_t, n_t)	CKPS	PBDMO	DMS	PPS	HPPCM	RVCP
JY1	(5,10)	1.5713e-1 (1.46e-2)	2.0583e-1 (1.41e-2)	1.4136e-1 (4.33e-2)	2.3998e-1 (7.28e-2)	2.3756e-1 (3.57e-2)	5.1952e-1 (2.35e-2)
	(10,10)	2.4090e-1 (1.94e-2)	4.9750e-1 (2.41e-2)	4.1654e-1 (1.23e-2)	9.3466e-2 (2.10e-2)	4.5430e-1 (4.35e-3)	5.9493e-1 (1.46e-2)
	(20,10)	2.3640e-1 (7.99e-3)	4.9817e-1 (1.29e-2)	4.2233e-1 (8.35e-3)	9.5915e-2 (1.10e-2)	4.5011e-1 (1.01e-2)	5.5741e-1 (2.37e-2)
JY2	(5,10)	4.4956e-1 (1.79e-2)	2.8379e-1 (4.28e-2)	1.0945e-1 (5.66e-3)	5.4673e-1 (4.02e-2)	4.2622e-1 (2.50e-2)	5.9173e-1 (7.82e-3)
	(10,10)	6.3817e-1 (3.02e-3)	6.0332e-1 (1.80e-3)	5.1335e-1 (1.83e-2)	5.8324e-1 (1.65e-2)	6.4437e-1 (3.93e-3)	6.6359e-1 (6.31e-3)
	(20,10)	6.3603e-1 (3.61e-3)	6.0251e-1 (1.64e-3)	5.0617e-1 (1.13e-2)	6.0034e-1 (6.72e-2)	6.4299e-1 (1.54e-3)	6.8382e-1 (1.02e-3)
JY3	(5,10)	3.3058e-1 (2.55e-2)	3.9151e-1 (1.22e-2)	5.8610e-1 (2.10e-2)	3.6738e-1 (2.92e-2)	5.5676e-1 (4.06e-2)	6.0497e-1 (1.84e-3)
	(10,10)	5.1964e-1 (7.84e-3)	5.7980e-1 (2.75e-3)	5.9655e-1 (8.68e-4)	4.2170e-1 (4.33e-2)	6.1324e-1 (3.20e-3)	5.9496e-1 (3.50e-3)
	(20,10)	5.2245e-1 (6.91e-3)	6.8038e-1 (1.49e-3)	7.5838e-1 (1.12e-2)	4.9254e-1 (9.75e-2)	6.8170e-1 (4.42e-3)	6.2845e-1 (4.60e-3)
JY4	(5,10)	3.9881e-1 (3.84e-2)	4.7336e-1 (4.36e-2)	1.7297e-1 (7.23e-2)	5.3480e-1 (4.14e-2)	5.0990e-1 (3.12e-2)	7.3099e-1 (2.22e-2)
	(10,10)	4.5675e-1 (1.26e-2)	6.0441e-1 (1.82e-2)	5.7323e-1 (6.32e-3)	1.1733e-1 (1.59e-2)	6.6550e-1 (5.87e-3)	7.5245e-1 (1.41e-2)
	(20,10)	4.5483e-1 (1.36e-2)	6.0564e-1 (1.59e-2)	5.6809e-1 (6.88e-3)	1.0585e-1 (1.74e-2)	6.5445e-1 (8.34e-3)	7.5526e-1 (1.36e-2)
JY5	(5,10)	3.9577e-1 (3.33e-2)	5.1856e-1 (1.03e-1)	6.1850e-1 (9.19e-3)	5.1701e-1 (6.76e-3)	6.3412e-1 (8.30e-3)	6.6676e-1 (5.31e-3)
	(10,10)	6.5258e-1 (9.00e-3)	1.0119e+0 (1.13e-2)	6.9036e-1 (2.00e-3)	6.5518e-1 (7.90e-3)	6.8346e-1 (1.06e-3)	6.7202e-1 (2.93e-3)
	(20,10)	6.4726e-1 (5.86e-3)	7.2072e-1 (2.43e-4)	6.8944e-1 (1.78e-3)	6.4537e-1 (6.25e-3)	6.8293e-1 (7.42e-4)	6.7357e-1 (4.01e-3)
JY6	(5,10)	4.5433e-3 (4.57e-3)	5.6470e-2 (1.10e-2)	2.6471e-3 (3.72e-3)	5.2999e-3 (2.62e-3)	3.3481e-3 (2.62e-3)	1.3755e-1 (4.84e-3)
	(10,10)	4.0295e-2 (7.96e-3)	6.9783e-2 (5.98e-3)	6.7763e-3 (2.65e-3)	4.0925e-2 (1.16e-2)	5.3857e-2 (6.28e-3)	1.1821e-1 (5.56e-3)
	(20,10)	4.1694e-2 (6.34e-3)	6.9799e-2 (7.04e-3)	8.3233e-3 (4.85e-3)	3.8946e-2 (9.55e-3)	5.0573e-2 (7.57e-3)	1.8371e-1 (4.59e-3)
JY7	(5,10)	8.8864e-2 (9.07e-3)	2.7194e-1 (7.45e-2)	9.9766e-3 (1.39e-2)	1.2570e-1 (6.98e-2)	4.9879e-2 (3.21e-2)	3.6580e-1 (2.01e-2)
	(10,10)	5.1492e-1 (5.24e-2)	3.6824e-1 (1.55e-2)	1.4340e-1 (2.35e-2)	5.9375e-1 (1.10e-2)	5.7025e-1 (5.11e-2)	6.0879e-1 (4.80e-2)
	(20,10)	3.8869e-1 (4.12e-2)	3.5472e-1 (5.39e-3)	1.3692e-1 (1.74e-2)	6.8221e-1 (2.05e-2)	5.4776e-1 (7.04e-2)	7.8410e-1 (6.97e-3)
JY8	(5,10)	4.6804e-1 (3.69e-2)	7.2148e-1 (1.14e-2)	7.0354e-1 (1.56e-2)	6.7281e-1 (2.04e-2)	6.8699e-1 (6.48e-3)	7.1694e-1 (3.93e-3)
	(10,10)	7.2809e-1 (8.33e-3)	7.0892e-1 (4.06e-4)	7.3325e-1 (6.78e-4)	6.6521e-1 (2.43e-2)	7.1135e-1 (2.57e-4)	7.3778e-1 (1.71e-3)
	(20,10)	7.2914e-1 (3.26e-3)	7.4895e-1 (5.11e-4)	7.8268e-1 (7.89e-4)	7.0241e-1 (1.48e-2)	8.5151e-1 (2.38e-4)	7.4171e-1 (1.28e-3)

JY3 belongs to type II, and its variables' links are relatively close but not monotonous. RVCP ranks first on JY3 when τ_t=5. The results obtained by RVCP are slightly worse than the other algorithms in the other two cases.

Unlike JY3, JY4 has little variable linkage, and its POF is discontinuous and changes complicatedly with varied time. RVCP has the relatively leading position, indicating the effective performance of RVCP in solving DMOPs with a time-varying POF.

The POSs of JY5 and JY8 do not change with varied time, and the geometries of their POFs vary from convex to concave. RVCP performed less ideal than on both of them, while PBDMO, DMS, and HPPCM can achieve considerable data. One reason may be that the prediction strategies of these three comparative algorithms can predict the fixed POSs of the problems. Especially DMS, the combination of three response mechanism, the central point prediction strategy, the gradual search strategy and the random individual method captures the fixed position of the POSs. All six algorithms obtain similar MIGD results on JY8. In general, RVCP has the best MIGD value when τ_t=5, whereas PBDMO and HPPCM have better values when τ_t=10 and τ_t=20. Both PBDMO and HPPCM have different moving direction settings and can track the irregular POF.

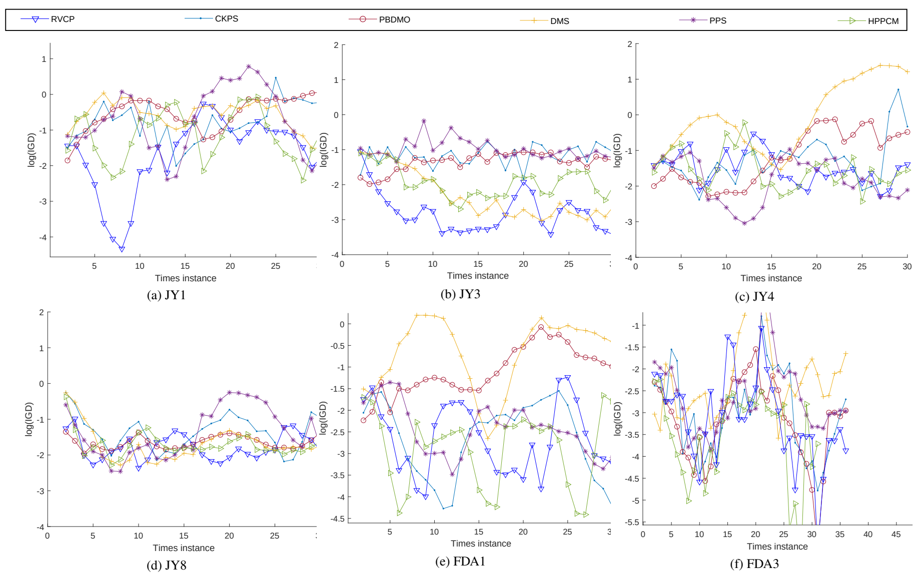

Figure 4: Evolution curve of MIGD values of six algorithms on representative test cases for time period t from 0 to 30, with (n_t, τ_t)=(10,10)

Different from JY5, both JY6 and JY7 are multimodal instances, and there are POSs shifting dynamically. And the local optima geometry parts in JY7 whose POF can be concave or convex with varied environmental changes, remains the same. RVCP has the most stable tracking abilities to find the POF of JY7, which may benefit from its predicted individuals covering the global area of POF. As a whole, RVCP can cope with three change types of DMOPs even in complex DMOP like JY2 and JY4, the MIGD and MHV results imply that RVCP is better than the other five algorithms in convergence or diversity.

We charted the evolutionary curve of IGD over the first 30 environmental changes for three instances. RVCP has a stable curve change for most of the time, signifying that the response strategy performed excellently in tracking the POS or POF. For JY8, although the IGD value is slightly higher for some periods, the curve is steadier. As has been observed, RVCP can respond to environmental changes in a more stable way.

4.4.3. Comparisons of different percentage settings in adaptive reference-vector adjustment strategy

When the number of the nondominated solutions is suffient, the shape of the POF can be described precisely by the predicted individuals of the first layer. When the number of nondominated solutions is insuffient, the predicted individuals of the first

Table 8: The MIGD values of RVCP on test instances with different percentage settings when (n_t, τ_t)=(10,10).

Problem	10%N	20%N	30%N	40% N	50% N
JY1	2.1746e-2 (7.23e-3)	2.0216e-2 (1.23e-2)	1.2889e-2 (2.24e-3)	1.7790e-2 (5.92e-3)	1.8150e-2 (1.32e-2)
JY2	4.7581e-3 (1.31e-3)	5.5091e-3 (2.30e-3)	4.1300e-3 (1.40e-3)	4.7090e-3 (1.35e-3)	4.5342e-3 (2.61e-3)
JY3	7.7087e-3 (1.00e-3)	8.3579e-3 (1.19e-3)	7.6189e-3 (7.81e-4)	6.9584e-3 (4.50e-4)	8.0246e-3 (6.70e-4)
JY4	8.4322e-3 (4.79e-3)	3.9017e-3 (1.52e-3)	5.6992e-3 (1.65e-3)	7.3584e-3 (4.82e-3)	4.6274e-3 (1.60e-3)
JY5	1.6515e-3 (1.24e-4)	1.4476e-3 (4.75e-4)	1.7136e-3 (2.05e-4)	1.5978e-3 (8.48e-4)	1.5398e-3 (3.75e-4)
JY6	3.4357e+0 (5.76e-1)	3.3212e+0 (4.29e-1)	3.3597e+0 (2.19e-1)	3.4523e+0 (6.48e-1)	3.4207e+0 (3.31e-1)
JY7	1.7369e+0 (1.89e+0)	3.3159e+0 (4.46e+0)	1.6579e+0 (1.58e+0)	1.0871e+0 (9.22e-1)	4.8680e+0 (2.36e+0)
JY8	2.0122e-2 (3.17e-4)	1.9457e-2 (4.83e-4)	1.9609e-2 (7.66e-4)	1.9668e-2 (7.77e-4)	1.9511e-2 (5.41e-4)
FDA1	1.0503e-2 (1.09e-2)	2.0332e-2 (3.11e-2)	1.7587e-2 (6.17e-3)	1.2937e-2 (1.15e-2)	1.1517e-2 (1.57e-2)
FDA2	1.6811e-3 (2.61e-5)	1.6731e-3 (2.81e-5)	1.6569e-3 (3.34e-5)	1.6542e-3 (2.04e-5)	1.6514e-3 (3.91e-5)
FDA3	1.2758e-2 (1.87e-3)	1.0664e-2 (2.71e-3)	1.1510e-2 (3.48e-3)	1.1340e-2 (1.51e-3)	1.1722e-2 (2.50e-3)
FDA4	1.1603e+1 (8.09e-2)	1.1550e+1 (2.53e-2)	1.1562e+1 (9.25e-2)	1.1613e+1 (8.57e-2)	1.1583e+1 (5.07e-2)

layer have difficulty tracking the POF. Thus, the reference vectors are adjusted adaptively according to the number of nondominated solutions. To express the influence of the critical value of the number of nondominated solutions, the mean and standard deviations of the MIGD indicator in 20 independent runs under the different percentage settings when (n_t, τ_t)=(10,10) are given in Table 8. As shown in Table 8, the 20% percentage setting achieves the best results compared to other percentage settings in the majority of test instances. Thus, we adopted the 20% percentage setting for RVCP. When the number of nondominated solutions is more than 20% of the population, the benchmark to be solved is regarded as an easy problem. When the number of nondominated solutions is less than 20% of the population, the benchmark to be solved is regarded as a complicated problem.

4.4.4. Comparisons of different adjustment degree settings in adaptive reference-vector adjustment strategy

The control parameter k is used to decide whether the reference vector is adjusted closer to or farther from the center vector. A positive k means the reference vector moves closer to C_v, whereas a negative k indicates the reference vector moves farther from C_v. Different adjustment degree settings of the reference vector are discussed in section 4.4.4. These control parameter settings are compared with the control parameter setting (0.5 or -0.5), and their MIGD values on seven test instances for (n_t,

Table 9: The MIGD values of RVCP on test instances with different adjustment degree settings when (n_t, τ_t)=(10,10).

Problem	(0.5 or -0.3)	(0.5 or -0.4)	(0.5 or -0.5)	(0.5 or -0.6)	(0.5 or -0.7)
JY1	3.2672e-2 (1.09e-2)	2.3270e-2 (3.86e-3)	2.0216e-2 (1.23e-2)	2.4405e-2 (1.12e-2)	2.4261e-2 (1.06e-2)
JY2	5.6549e-3 (1.68e-3)	5.7165e-3 (2.61e-3)	5.5091e-3 (2.30e-3)	4.8684e-3 (8.72e-4)	4.6099e-3 (6.20e-4)
JY3	7.9850e-3 (5.76e-4)	7.7266e-3 (7.76e-4)	8.3579e-3 (1.19e-3)	7.9058e-3 (6.78e-4)	8.6946e-3 (3.56e-4)
JY4	1.2187e-2 (1.52e-2)	7.2755e-3 (3.75e-3)	3.9017e-3 (1.52e-3)	9.9051e-3 (6.78e-3)	7.6992e-3 (2.33e-3)
JY5	8.2835e-4 (1.60e-4)	6.5149e-4 (1.55e-4)	1.4476e-3 (4.75e-4)	1.1323e-3 (2.43e-4)	7.5994e-4 (6.48e-5)
JY6	3.6742e+0 (1.69e-1)	3.3065e+0 (2.09e-1)	3.3212e+0 (4.29e-1)	3.5015e+0 (4.00e-1)	3.4473e+0 (7.95e-2)
JY7	1.1778e+0 (8.22e-1)	5.2077e-1 (4.63e-1)	3.3159e+0 (4.46e+0)	8.6614e-1 (9.64e-1)	2.6467e+0 (1.48e+0)
JY8	1.9556e-2 (1.45e-4)	1.9866e-2 (3.05e-4)	1.9457e-2 (4.83e-4)	1.9693e-2 (1.21e-4)	2.0290e-2 (4.79e-4)
FDA1	1.6655e-2 (1.13e-2)	2.6156e-2 (2.08e-2)	2.0332e-2 (3.11e-2)	1.2751e-2 (8.68e-3)	1.2618e-2 (7.37e-3)
FDA2	1.8306e-3 (1.51e-5)	1.8317e-3 (2.01e-5)	1.6731e-3 (2.81e-5)	1.8296e-3 (1.00e-5)	1.8519e-3 (2.41e-5)
FDA3	1.1254e-2 (1.00e-3)	1.2180e-2 (2.13e-3)	1.0664e-2 (2.71e-3)	1.2482e-2 (2.84e-3)	1.1261e-2 (2.36e-3)
FDA4	1.1791e+1 (7.59e-2)	1.1793e+1 (7.15e-2)	1.1550e+1 (2.53e-2)	1.1767e+1 (4.64e-2)	1.1797e+1 (2.93e-2)

Table 10: The MIGD values of RVCP on test instances with different adjustment degree settings when (n_t, τ_t)=(10,10).

Problem	(0.3 or -0.5)	(0.4 or -0.5)	(0.5 or -0.5)	(0.6 or -0.5)	(0.7 or -0.5)
JY1	2.7439e-2 (1.14e-2)	1.7898e-2 (1.02e-2)	2.0216e-2 (1.23e-2)	2.5278e-2 (1.50e-2)	1.8728e-2 (4.38e-3)
JY2	5.0885e-3 (2.33e-3)	4.0358e-3 (7.48e-4)	5.5091e-3 (2.30e-3)	5.6162e-3 (2.13e-3)	6.2616e-3 (1.99e-3)
JY3	8.0613e-3 (7.14e-4)	7.9663e-3 (2.95e-4)	8.3579e-3 (1.19e-3)	8.4080e-3 (7.90e-4)	7.8487e-3 (4.62e-4)
JY4	1.2854e-2 (1.58e-2)	9.0065e-3 (2.37e-3)	3.9017e-3 (1.52e-3)	7.9721e-3 (3.09e-3)	1.5364e-2 (1.15e-2)
JY5	7.4350e-4 (1.84e-4)	8.6176e-4 (3.98e-4)	1.4476e-3 (4.75e-4)	9.3493e-4 (3.87e-4)	1.3250e-3 (9.10e-4)
JY6	3.6366e+0 (3.93e-1)	3.3806e+0 (2.04e-1)	3.3212e+0 (4.29e-1)	3.4875e+0 (1.63e-1)	3.5162e+0 (3.35e-1)
JY7	2.9005e+0 (3.65e+0)	6.2552e-1 (6.12e-1)	3.3159e+0 (4.46e+0)	2.1694e+0 (1.20e+0)	2.5209e+0 (1.04e+0)
JY8	1.9878e-2 (4.68e-4)	1.9810e-2 (1.25e-4)	1.9457e-2 (4.83e-4)	1.9715e-2 (3.90e-4)	1.9702e-2 (2.08e-4)
FDA1	1.7714e-2 (1.46e-2)	1.7988e-2 (8.54e-3)	2.0332e-2 (3.11e-2)	2.6327e-2 (1.61e-2)	1.9406e-2 (9.95e-3)
FDA2	1.8303e-3 (1.58e-5)	1.8254e-3 (1.55e-5)	1.6731e-3 (2.81e-5)	1.8425e-3 (2.48e-5)	1.8368e-3 (1.80e-5)
FDA3	1.1575e-2 (1.12e-3)	1.1991e-2 (1.53e-3)	1.0664e-2 (2.71e-3)	1.2270e-2 (1.53e-3)	1.2311e-2 (2.26e-3)
FDA4	1.1791e+1 (4.94e-2)	1.1804e+1 (7.53e-2)	1.1550e+1 (2.53e-2)	1.1848e+1 (5.40e-2)	1.1752e+1 (5.97e-2)

τ_t)=(10,10) are presented in Tables 9 and 10. It is evident from the results of the experimental comparison that the overall performance of the adjustment degree (0.5 or -0.5) of the adaptive reference-vector adjustment strategy is better than other degree settings.

As shown in Table 9, the control parameter setting (0.5 or -0.5) achieves the best results compared to other control parameter settings in the majority of test instances. In JY6, the control parameter setting (0.5 or -0.5) test results are still close to the best one.

As shown in Table 10, the control parameter setting (0.5 or -0.5) obtains better results in most test instances. In JY1 and JY2, the parameter setting (0.5 or -0.5) is close to other parameter settings. In most test instances, the control parameter setting (0.5 or -0.5) obtains better result. In general, we adopted the parameter setting (0.5 or

Table 11: The Mean and Standard Deviation values of the MIGD metric obtained by RVCP variants when (n_t, τ_t)=(10,10).

Problem	RVCP-V1	RVCP-V2	RVCP-V3	RVCP
JY1	7.3904e-1 (2.84e-1)	2.4763e-1 (3.10e-2)	2.3130e-2 (4.26e-3)	2.0216e-2 (1.23e-2)
JY3	2.9688e-3 (3.47e-3)	3.9007e-3 (1.06e-3)	1.1227e-2 (8.94e-3)	8.3579e-3 (1.19e-3)
JY5	2.4175e-2 (2.88e-2)	1.3741e-2 (1.34e-2)	1.1373e-2 (1.34e-2)	1.4476e-3 (4.75e-4)
JY8	8.6009e-2 (5.79e-2)	4.0788e-2 (7.89e-4)	3.1921e-2 (2.15e-3)	1.9457e-2 (4.83e-4)
FDA1	5.3761e-1 (1.60e-1)	6.5889e-1 (5.73e-1)	7.1883e-3 (1.81e-1)	2.0332e-2 (3.11e-2)
FDA2	1.4437e-2 (5.98e-3)	3.7664e-3 (2.40e-4)	9.7299e-3 (1.28e-2)	1.6731e-3 (2.81e-5)
FDA3	2.1490e-2 (3.49e-2)	2.4304e-1 (1.22e+0)	1.2860e-2 (2.70e-2)	9.0120e-3 (2.27e-3)
FDA4	3.8659e+1 (1.66e+1)	5.0451e+1 (3.41e-1)	1.6328e+1 (7.30e-1)	1.1477e+1 (7.16e-2)
p-VALUE	3.00e-02	3.39e-02	4.47e-03	

The results in grey backgroud indicates that the performance of the corresponding algorithm ranks the best, at the 5% significance level under the Friedman test.

-0.5) as the control parameter setting of *k*.

4.4.5. *Influence of different parts of RVCP*

In order to study the different components (i.e. LPS, RAS and NES) of RVCP, the ablation experiments are applied to verify the effectiveness of the components, which is similar with the ”controlled variable method. We could keep two of the three and remove one to verify whether the deleted one makes effects in the entire system. RVCP algorithm has been transformed into three critical versions. The first version (RVCP-V1) only discards LPS methods to update population. The second (RVCP-V2) does not contain the RAS method. In the second variant (RVCP-V2), the RAS methods is replaced with the individuals using the randomly initialization to generate individuals. Similarly, we also deactivate noise-based individual expansion strategy to study the role it plays in the proposed algorithm, resulting in another variant named RVCP-V3.

From Table 12 and Table 11 , it is observed that RVCP performs better than the three variants on test instances. Compared with RVCP-V3, RVCP-V1 and RVCP-V2 does not perform as well as RVCP on test issues, suggesting that the LPS and RAS mechanism have a greater contribution to algorithm performance.

4.4.6. *Statistical analysis*

To show the significant difference among the six algorithms from the perspective of statistical, a multiple comparisons test such as Friedman is used in combination with a post-hoc analysis test such as Holm adjustment. This will make comparisons more robust and statistically consistent.

Table 12: The Mean and Standard Deviation values of the MSP metric obtained by RVCP variants when (n_t, τ_t)=(10,10).

Problem	RVCP-V1	RVCP-V2	RVCP-V3	RVCP
FDA1	2.2759e+0 (1.83e-1)	1.3776e+0 (5.48e-1)	1.2652e+0 (5.49e-2)	1.0791e+0 (8.58e-2)
FDA2	1.2494e+0 (3.46e-1)	1.2594e+0 (4.27e-1)	1.2736e+0 (3.99e-1)	1.1449e+0 (2.82e-1)
FDA3	1.1502e+0 (5.42e-2)	1.7185e+0 (4.54e-1)	1.2664e+0 (2.15e-1)	1.0319e+0 (4.45e-2)
FDA4	7.3171e+0 (3.68e-1)	1.5785e+0 (4.00e-1)	9.5784e-1 (3.81e-2)	1.2114e+0 (1.62e-1)
JY1	1.0232e+0 (1.77e-1)	1.5481e+0 (3.34e-1)	9.2054e-1 (8.26e-2)	1.0796e+0 (1.82e-1)
JY2	1.4904e+0 (2.83e-1)	1.5747e+0 (5.84e-1)	1.4306e+0 (3.16e-1)	1.2716e+0 (1.65e-1)
JY3	1.3863e+0 (1.96e-1)	1.7771e+0 (2.16e-1)	1.4474e+0 (2.08e-1)	1.0720e+0 (6.04e-2)
JY4	1.8647e+0 (1.92e-1)	1.7774e+0 (2.59e-1)	1.9022e+0 (2.16e-1)	1.4946e+0 (2.78e-1)
JY5	1.2707e+0 (2.11e-1)	1.1638e+0 (7.25e-1)	7.2422e-1 (7.37e-2)	1.0000e+0 (1.31e-13)
JY6	1.7481e+0 (4.40e-1)	1.4684e+0 (3.95e-1)	1.8596e+0 (2.20e-1)	1.0009e+0 (2.99e-3)
JY7	2.1275e+0 (5.11e-2)	1.8940e+0 (9.63e-2)	2.1008e+0 (4.47e-2)	1.6102e+0 (2.42e-1)
JY8	1.5050e+0 (8.63e-2)	1.3958e+0 (4.33e-1)	1.4925e+0 (7.65e-2)	1.0705e+0 (1.01e-1)
p-value	2.09e-02	3.90e-03	8.33e-02	

The results in grey backgroud indicates that the performance of the corresponding algorithm ranks the best, at the 5% significance level under the Friedman test.

In the Friedman test, *p*-Value represents the test value of all the algorithms. The Friedman test with the correction methods at a significance level 0.05 are adopted to perform statistical analysis on the experimental results. We choose Benjamini-Hochberg and Holm correction [48] to adjust *p*-values.

As shown in the Table 13, RVCP obtains the minimum average ranking, which indicates the robustness and superiority of its performance further. At the same time, the Benjamini-Hochberg and Holm correction are used to obtain the adjusted *p*-values. To perform quantitative analysis, the effect size [49]is applied to measures the extent to which the first algorithm outperforms the second one. Therefore, non-parametric effect size statistic test is conducted on the MIGD results obtained by RVCP and the

Table 13: Average rankings and adjusted *p*-values obtained of each algorithm through Friedman test on all test problems (RVCP is the control method whose average rank smallest, 2.1389).

Algorithm	CKPS	PBDMO	DMS	PPS	HPPCM
Avg Ranking	3.8611	3.3611	4.3889	4.6944	2.5556
p-value	0.0027	0.0077	6.33e-05	3.06e-06	0.0455
Holm	0.0081	0.0154	0.0002532	1.53e-05	0.0455
Benjamini-Hochberg	0.0045	0.009625	0.00015825	1.53e-05	0.0455

Table 14: EFFECT SIZE STATISTIC OF MIGD VALUES OBTAINED BY RVCP AND THE COMPARED ALGORITHMS

RVCP vs	(n_t, τ_t)	CKPS	PBDMO	DMS	PPS	HPPCM
FDA1	(5, 10)	-0.66 (Medium)	**0.88 (Large)**	**0.97 (Large)**	0.41 (Medium)	0.71 (Medium)
	(10, 10)	0.32 (Medium)	0.30 (Medium)	**0.97 (Large)**	**0.97 (Large)**	-0.38 (Medium)
	(20, 10)	-0.62 (Medium)	**0.99 (Large)**	**0.86 (Large)**	0.26 (Medium)	0.04 (Small)
FDA2	(5, 10)	0.57 (Medium)	**0.96 (Large)**	**0.97 (Large)**	0.78 (Medium)	0.78 (Medium)
	(10, 10)	0.71 (Medium)	**0.87 (Large)**	0.19 (Small)	0.63 (Medium)	0.58 (Medium)
	(20, 10)	0.56 (Medium)	**0.98 (Large)**	0.58 (Medium)	**0.84 (Large)**	0.56 (Medium)
FDA3	(5, 10)	**0.84 (Large)**	**0.93 (Large)**	**0.99 (Large)**	**0.87 (Large)**	**0.81 (Large)**
	(10, 10)	-0.13 (Small)	0.79 (Medium)	**0.88 (Large)**	0.43 (Medium)	-0.32 (Medium)
	(20, 10)	0.23 (Medium)	-0.19 (Small)	**0.87 (Large)**	0.55 (Medium)	**-0.97 (Large)**
FDA4	(5, 10)	**-0.97 (Large)**	**-0.96 (Large)**	**-0.96 (Large)**	**-0.96 (Large)**	**-0.97 (Large)**
	(10, 10)	**-0.99 (Large)**	**-1.00 (Large)**	0.30 (Medium)	**1.00 (Large)**	**-1.00 (Large)**
	(20, 10)	**-0.99 (Large)**	**-1.00 (Large)**	**-0.99 (Large)**	**-0.86 (Large)**	**-1.00 (Large)**
JY1	(5, 10)	**0.97 (Large)**	**1.00 (Large)**	**0.9 (Large)**	0.08 (Small)	**0.87 (Large)**
	(10, 10)	0.56 (Medium)	**0.98 (Large)**	**0.95 (Large)**	0.70 (Medium)	**0.96 (Large)**
	(20, 10)	**0.98 (Large)**	**0.80 (Large)**	**0.96 (Large)**	**0.87 (Large)**	**0.99 (Large)**
JY2	(5, 10)	0.21 (Small)	**1.00 (Large)**	**0.93 (Large)**	-0.66 (Medium)	0.01 (Small)
	(10, 10)	**0.84 (Large)**	**0.99 (Large)**	**0.97 (Large)**	0.37 (Medium)	**1.00 (Large)**
	(20, 10)	**1.00 (Large)**	**1.00 (Large)**	0.76 (Medium)	0.59 (Medium)	**0.94 (Large)**
JY3	(5, 10)	**1.00 (Large)**	**0.99 (Large)**	0.11 (Small)	**0.93 (Large)**	0.33 (Medium)
	(10, 10)	**0.98 (Large)**	**-0.97 (Large)**	**-0.97 (Large)**	**0.95 (Large)**	**-0.89 (Large)**
	(20, 10)	**0.95 (Large)**	**-0.87 (Large)**	**-0.89 (Large)**	**0.81 (Large)**	-0.55 (Medium)
JY4	(5, 10)	**0.80 (Large)**	**0.84 (Large)**	**0.80 (Large)**	0.57 (Medium)	**0.92 (Large)**
	(10, 10)	**0.93 (Large)**	**0.97 (Large)**	**0.80 (Large)**	0.69 (Medium)	**0.93 (Large)**
	(20, 10)	**0.90 (Large)**	**0.97 (Large)**	**0.98 (Large)**	**0.95 (Large)**	**0.88 (Large)**
JY5	(5, 10)	**0.97 (Large)**	0.43 (Medium)	**0.91 (Large)**	**0.89 (Large)**	**0.84 (Large)**
	(10, 10)	**0.93 (Large)**	**-0.82 (Large)**	-0.17 (Small)	0.58 (Medium)	-0.35 (Medium)
	(20, 10)	0.04 (Small)	-0.76 (Medium)	-0.73 (Small)	0.4 (Medium)	-0.78 (Medium)
JY6	(5, 10)	**0.90 (Large)**	0.12 (Small)	**0.94 (Large)**	**0.97 (Large)**	**0.97 (Large)**
	(10, 10)	**0.98 (Large)**	0.72 (Medium)	**0.99 (Large)**	**0.96 (Large)**	**0.97 (Large)**
	(20, 10)	**0.99 (Large)**	**0.93 (Large)**	**1.00 (Large)**	**0.97 (Large)**	**0.99 (Large)**
JY7	(5, 10)	0.06 (Small)	0.07 (Small)	**0.99 (Large)**	0.57 (Medium)	**0.97 (Large)**
	(10, 10)	0.01 (Small)	0.55 (Medium)	**0.98 (Large)**	0.59 (Medium)	**0.96 (Large)**
	(20, 10)	0.01 (Small)	0.39 (Medium)	**0.99 (Large)**	0.03 (Small)	0.75 (Medium)
JY8	(5, 10)	**0.96 (Large)**	0.43 (Medium)	0.69 (Medium)	0.74 (Medium)	**0.81 (Large)**
	(10, 10)	0.78 (Medium)	**0.98 (Large)**	**0.88 (Large)**	**0.84 (Large)**	0.43 (Medium)
	(20, 10)	-0.68 (Medium)	-0.72 (Medium)	**0.84 (Large)**	0.62 (Medium)	**-0.88 (Large)**

compared algorithms, to check the magnitude of the differences. The effect size in terms of MIGD values are calculated and the results are presented in Table 14. The results of the effect size are usually considered as small, medium and large when the effect size value is less than 0.2, greater than 0.2 but less than 0.8, greater than 0.8. The results in bold indicates that the performance of the corresponding algorithm ranks the better.

5. Conclusion

In this paper, the reference-vector-based adjustmen prediction approach is proposed to solve DMOPs. The RVCP algorithm consists of three key components: the linear prediction strategy (LPS), reference-vector-based strategy (RAS) and noise-based individual expansion strategy (NES), which serves as the dynamic response strategy in finding the POF and boundary solutions in the new environment. To keep a fast convergence speed, LPS strategy is used to generate solutions in a new environment by using linear prediction method, and the RAS strategy is applied to track more accurate individuals through the adjusted reference vectors adaptively. Finally, the NES method is used to expand the individuals for improving the diversity. These three strategies are important for generating a good initial population, which enhances either diversity or convergence upon a new change occurs in the environment. To examine the performance of the proposed algorithm, there are 12 benchmark functions with various characteristics employed to estimate the performance of RVCP. The experimental results show that the performance of RVCP is better than the other five comparison algorithms on most test issues, indicating that RVCP is therefore suitable to solve DMOPs with dynamic characteristics.

For better performance, several extensions are possible for future work:

- The DMOEAs have demonstrated the ability to solve DMOPs, which is a prospective research problem. Simultaneously, new dynamic real-problems test instances [50] and performance metrics are needed to evaluate the performance of the algorithm.

- We will continue to study other prediction models and apply them into RVCP to solve the real-world problems, such as Kalman filter prediction model [51]. In the future, we will try to modify RVCP for solving dynamic multi-objective optimal problems.

6. Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (Grant No. 62176228 and Grant No. 61876164), in part by the Research Foundation of Education Bureau of Hunan Province, China (Grant No. 21A0444), and in part by the Science and Technology Plan Project of Hunan Province (Grant No. 2018TP1036).

References

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

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

[3] S. Jiang, Evolutionary algorithms for static and dynamic multiobjective optimization (2016).

[4] H. Richter, Change detection in dynamic fitness landscapes: An immunological approach, in: 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), IEEE, 2009, pp. 719–724.

[5] S. Jiang, S. Yang, X. Yao, K. C. Tan, M. Kaiser, N. Krasnogor, Benchmark functions for the cec'2018 competition on dynamic multiobjective optimization, Tech. rep., Newcastle University (2018).

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

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

[8] F. Rahmani, K. Behzadian, A. Ardeshir, Rehabilitation of a water distribution system using sequential multiobjective optimization models, Journal of Water Resources Planning and Management 142 (5) (2016) C4015003.

[9] L. Chen, H. Wang, D. Pan, H. Wang, W. Gan, D. Wang, T. Zhu, Dynamic multiobjective evolutionary algorithm with adaptive response mechanism selection strategy, Knowledge-Based Systems (2022) 108691.

[10] 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, Springer, 2007, pp. 803–817.

[11] Q. Liu, J. Ding, Reference vector based multidirectional prediction for evolutionary dynamic multiobjective optimization, in: 2019 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2019, pp. 1081–1087.

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

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

[14] M. Jiang, Z. Wang, H. Hong, G. G. Yen, Knee point-based imbalanced transfer learning for dynamic multiobjective optimization, IEEE Transactions on Evolutionary Computation 25 (1) (2020) 117–129.

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

[16] Y. Wu, Y. Jin, X. Liu, A directed search strategy for evolutionary dynamic multiobjective optimization, Soft Computing 19 (11) (2015) 3221–3235.

[17] 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.

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

[19] M. Rong, D. Gong, W. Pedrycz, L. Wang, A multimodel prediction method for dynamic multiobjective evolutionary optimization, IEEE Transactions on Evolutionary Computation 24 (2) (2019) 290–304.

[20] 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.

[21] H. Sun, A. Cao, Z. Hu, X. Li, Z. Zhao, A novel quantile-guided dual prediction strategies for dynamic multi-objective optimization, Information Sciences 579 (2021) 751–775.

[22] X. Xu, Y. Tan, W. Zheng, S. Li, Memory-enhanced dynamic multi-objective evolutionary algorithm based on lp decomposition, Applied Sciences 8 (9) (2018) 1673.

[23] R. Chen, K. Li, X. Yao, Dynamic multiobjectives optimization with a changing number of objectives, IEEE Transactions on Evolutionary Computation 22 (1) (2017) 157–171.

[24] C.-K. Goh, K. C. Tan, A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization, IEEE Transactions on Evolutionary Computation 13 (1) (2008) 103–127.

[25] M. Jiang, W. Hu, L. Qiu, M. Shi, K. C. Tan, Solving dynamic multi-objective optimization problems via support vector machine, in: 2018 Tenth International Conference on Advanced Computational Intelligence (ICACI), IEEE, 2018, pp. 819–824.

[26] S. B. Gee, K. C. Tan, C. Alippi, Solving multiobjective optimization problems in unknown dynamic environments: An inverse modeling approach, IEEE Trans. Cybern. 47 (12) (2016) 4223–4234.

[27] R. Cheng, Y. Jin, K. Narukawa, B. Sendhoff, A multiobjective evolutionary algorithm using gaussian process-based inverse modeling, IEEE Trans. Evol. Comput. 19 (6) (2015) 838–856.

[28] W. Zhou, L. Feng, K. C. Tan, M. Jiang, Y. Liu, Evolutionary search with multiview prediction for dynamic multi-objective optimization, IEEE Transactions on Evolutionary Computation (2021).

[29] R. Azzouz, S. Bechikh, L. B. Said, A dynamic multi-objective evolutionary algorithm using a change severity-based adaptive population management strategy, Soft Computing 21 (4) (2017) 885–906.

[30] D. Yazdani, R. Cheng, D. Yazdani, J. Branke, Y. Jin, X. Yao, A survey of evolutionary continuous dynamic optimization over two decades—part a, IEEE Transactions on Evolutionary Computation 25 (4) (2021) 609–629.

[31] Z. Liang, W. Hou, X. Huang, Z. Zhu, Two new reference vector adaptation strategies for many-objective evolutionary algorithms, Information Sciences 483 (2019) 332–349.

[32] L. Feng, W. Zhou, W. Liu, Y.-S. Ong, K. C. Tan, Solving dynamic multiobjective problem via autoencoding evolutionary search, IEEE Transactions on Cybernetics (2020).

[33] 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.

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

[35] H.-L. Liu, F. Gu, Q. Zhang, Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems, IEEE transactions on evolutionary computation 18 (3) (2013) 450–455.

[36] I. Das, J. E. Dennis, Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems, SIAM journal on optimization 8 (3) (1998) 631–657.

[37] 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.

[38] S. Jiang, S. Yang, Evolutionary dynamic multiobjective optimization: Benchmarks and algorithm comparisons, IEEE transactions on cybernetics 47 (1) (2016) 198–211.

[39] 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 and Evolutionary Computation (2022) 101041.

[40] W. K. Mashwani, A. Salhi, Multiobjective evolutionary algorithm based on multimethod with dynamic resources allocation, Applied Soft Computing 39 (2016) 292–309.

[41] W. K. Mashwani, A. Salhi, O. Yeniay, H. Hussian, M. A. Jan, Hybrid non-dominated sorting genetic algorithm with adaptive operators selection, Applied Soft Computing 56 (2017) 1–18.

[42] M. Jiang, Z. Huang, L. Qiu, W. Huang, G. G. Yen, Transfer learning-based dynamic multiobjective optimization algorithms, IEEE Transactions on Evolutionary Computation 22 (4) (2017) 501–514.

[43] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, V. G. Da Fonseca, Performance assessment of multiobjective optimizers: An analysis and review, IEEE Transactions on evolutionary computation 7 (2) (2003) 117–132.

[44] R. Liu, Y. Chen, W. Ma, C. Mu, L. Jiao, A novel cooperative coevolutionary dynamic multi-objective optimization algorithm using a new predictive model (2014).

[45] M. Hollander, D. A. Wolfe, E. Chicken, Nonparametric statistical methods, Vol. 751, John Wiley & Sons, 2013.

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

[47] Y. Wu, L. Shi, X. Liu, A new dynamic strategy for dynamic multi-objective optimization, Information Sciences 529 (2020) 116–131.

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

[49] G. M. Sullivan, R. Feinn, Using effect size—or why the p value is not enough, Journal of graduate medical education 4 (3) (2012) 279–282.

[50] K. Shi, P. Wu, M. Liu, Research on path planning method of forging handling robot based on combined strategy, in: 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA), IEEE, 2021, pp. 292–295.

[51] K. K. Singh, S. Kumar, P. Dixit, M. K. Bajpai, Kalman filter based short term prediction model for covid-19 spread, Applied Intelligence 51 (5) (2021) 2714–2726.