CIT. Journal of Computing and Information Technology, Vol. 34, No. 1, March 2026, 1–22
doi: 10.20532/cit.2026.1006008

A Dynamic Multi-objective Optimization Algorithm for Black-Box Optimization in Dynamic Irregular Environment

Sanyi Li[1], Shuang Liu[2], Wenjie Hou[1] and Yuanyuan Zhang[3]

[1]School of Electrical and Information Engineering, Zhengzhou University of Light Industry, Zhengzhou, China
[2]College of Mechanical and Electrical Engineering, Linzhou College of Architectural Technology, Anyang Henan, China
[3]College of Mechanical and Marine Engineering, Beibu Gulf University, Qinzhou Guangxi, China

This paper proposes a novel initial population prediction algorithm (IPPA) for solving irregular black-box dynamic multi-objective optimization problems. In the phase of environmental change detection, IPPA find representative solutions based on a clustering algorithm and form environment vector to judge whether the environment has changed. In the population initialization stage, the initial population is generated by three mechanisms. First, a feedforward neural network (FNN) is trained with historical population information, and part of the initial population is generated by FNN. Second, when the new environment is similar to the last environment, part of the elite population in the last environment is retained. Finally, some populations are randomly generated to maintain population diversity. Since IPPA does not require the use of explicit objective functions and recent solutions, it can solve black-box dynamic multi-objective optimization problems with drastic and irregularly changing Pareto set. The proposed algorithm is compared with other state-of-the-art algorithms on a number of benchmark problems. Experimental results show that the proposed IPPA algorithm is effective for irregular black-box dynamic multi-objective optimization problems.

ACM CCS (2012) Classification: Theory of computation → Design and analysis of algorithms → Algorithm design techniques → Dynamic programming

Computing methodologies → Artificial intelligence → Planning and scheduling → Planning under uncertainty

Computing methodologies → Machine learning → Machine learning approaches → Neural networks

Keywords: dynamic multi-objective optimization, pareto Set, neural network, prediction, irregular environment

1. Introduction

Many real-world multi-objective optimization problems (MOPs) often contain uncertain and dynamic factors, whose decision variables, parameters, or even the number of objective functions may change over time. These kinds of MOPs are usually considered as dynamic multi-objective optimization problems (DMOPs). Because of the universality of DMOPs, it attracts more and more interests in many fields, such as scheduling [1], [2], [3], routing [4], [5], design [6], [7] and flow industry [8], [9]. In this paper, we mainly consider the following black-box DMOPs whose parameters of objective functions are unknown:

$$\begin{cases} \min & F(x,t) = \left(f_1(x,t),\ldots,f_m(x,t)\right)^T \\ \text{s.t.} & x \in \prod_{i=1}^{n}\left[a_i,b_i\right] \end{cases} \tag{1}$$

where $F(x, t)$ is the m-dimensional objective function vector and $f_i(x, t)$ is the i-th objective function; t is the time step; $x = (x_1, x_2, \ldots, x_n)^T$ is an n-dimensional decision variable vector and $\prod_{i=1}^{n}\left[a_i,b_i\right] \subset R^n$ is the decision space. The DMOP defined in (1) can be regarded as a series of static MOPs switched over time. As the optimization problem changes, Pareto optimal solutions (POS) and the Pareto optimal front (POF) change accordingly. Therefore, the goal of solving DMOPs is to quickly track the changing

POF (POS). When dealing with DMOPs, traditional static multi-objective optimization algorithms (SMOA) are insufficient [10]. When the environment changes, it is difficult to quickly converge to the new POF. Therefore, some researchers tried to solve this problem by increasing the diversity of population, such as adopting various strategies of mutation, immigration and randomization [10], [11], [12]. However, when the environment changes dramatically, simply increasing population diversity cannot quickly track the changing POF.

Some researchers focus their attention on population initialization after environmental changes, trying to obtain excellent initial population by exploiting historical information [13], [14], [15]. If the environment changes are regular, the movement of POS can be modeled based on historical information. When an environment change is detected, the new POS can be predicted through the model, thus enabling the population to rapidly track the moving POS [16]. For example, Xie *et al.* [17] utilized the information of decision variable classification to guide the search process for improving convergence speed. Wang *et al.* [18] built the grey prediction model by using the centroid point of each cluster to generate the initial population. Cao *et al.* [19] employed the support vector regression (SVR) to conduct nonlinear prediction. Although the methods based on prediction models have a good effect on solving DMOPs with regular environmental changes, the prediction models of the POS cannot be established when the environment changes irregularly.

Different from prediction-based algorithms, if the environment changes periodically, some researchers build a memory pool to store the non-dominated solutions of current POS and reuse these solutions when encountering a similar environment in the future. This kind of method is often called memory-based strategies. The reuse of non-dominated solutions in memory pool can improve the convergence speed of the population. For example, Sahmoud *et al.* [20] presented a new hybrid strategy by integrating the memory concept with the NSGA-II algorithm. Zheng *et al.* [21] used the memory strategy to provide valuable information to the prediction model to predict the POS of the new environment more accurately. Koo *et al.* [22] introduced a new memory technique to exploit any periodicity in the dynamic problem. But if the environment changes aperiodically, these methods cannot determine whether the new environment is the same as a historical environment.

For DMOPs, black-box problems are equally challenging. The defining characteristic of black-box problems is that we can only obtain the outputs corresponding to given inputs, without having direct knowledge of the specific mapping relationship between inputs and outputs. In practice, there are numerous instances of black-box problems. Take, for example, the case of wastewater treatment processes, where input variables such as aeration rate and internal recirculation rate are commonly considered, while various effluent water quality parameters and energy consumption parameters are regarded as the outputs. Due to the involvement of numerous microbial reactions and complex physicochemical processes in wastewater treatment, we cannot directly discern mapping relationships between inputs and outputs. Consequently, wastewater treatment processes represent a typical black-box DMOP. Most dynamic multi-objective optimization algorithms require prior knowledge of the explicit expression of objective functions, and as a result, they are not well-suited to address black-box DMOPs.

In previous work, we proposed a population prediction algorithm based on modular neural network (PA-MNN) [23] to solve irregular DMOPs. When a new environment has been encountered previously, initial solutions generated by modular neural network will have similar distribution characteristics as the final solutions that were obtained in the same environment last time. However, PA-MNN requires knowledge of the specific mapping relation between the decision variables and the objective function when setting environment vectors, so it cannot solve black-box DMOPs.

To address the shortcomings of PA-MNN, this paper proposes an initial population prediction algorithm (IPPA) for irregular black-box DMOPs. In the population initialization stage, the initial population is generated by three mechanisms. First, a feedforward neural network (FNN) is trained with historical population and environment vector information. When an environment change is detected, part of the initial populations is generated by FNN. Second, when the new environment is similar to the last environment, part of the elite population in the last environment is retained. Finally, some populations are randomly generated to maintain population diversity. The main contributions of this paper are summarized as follows:

1. This paper proposes a novel setting method of environment variables for aperiodic irregular black-box DMOPs, which can effectively detect environment changes and determine whether the new environment is the same as a historical environment.
2. For an aperiodic irregular black-box DMOP, the initial population generated by IPPA will have similar distribution characteristics as the final solutions that were obtained in the same environment last time.
3. The computational complexity of the proposed algorithm is not high, so it can be applied in the situations requiring high speed.

The rest of this paper is organized as follows: Section 2 presents some related work. Section 3 introduces the details of the proposed IPPA. Section 4 introduces the test problems, parameter settings and performance metrics used in the experiments. Section 5 provides the experimental results and analysis. Section 6 concludes this paper and suggests future research topics.

2. Background

This section presents a general dynamic multi-objective evolutionary algorithm (DMOEA) framework. The framework is shown in Algorithm 1.

As shown in Algorithm 1, the main research points of dynamic multi-objective optimization are environmental change detection mechanism, change response, and population optimization. Environmental change detection methods are mainly divided into two categories: reevaluating solutions [24] and checking population statistical information [25]. Both methods can detect environmental changes but cannot determine whether the new environment has ever existed, which leads to historical information not being reused effectively. For this problem, we proposed an environmental vector setting method in [23]. By setting an environment vector, it can be judged whether the environment has changed and whether the environment has ever been encountered, which can realize the reuse of population history information. However, the method needs to know the exact objective function, so it cannot be used in black-box DMOPs. Therefore, this paper proposes an environment vector setting method based on clustering a algorithm for irregular black-box DMOPs. The setting of environment vectors and the process of population initialization will be described in detail in Section 3.

Algorithm 1. A general DMOEA framework.

Set the time step $t = 0$.
Initial population P_I.
while *termination conditions are not met* **do**
    if *environmental change is detected*
        Set $t = t + 1$.
        Change reaction phase: increase population diversity or reinitialize the population.
    else
        Optimize the population with SMOEA for one generation.
    end
end while

2.1. Clustering Algorithm

The clustering algorithm is based on the Euclidean distance formula, and it determines the appropriate centroids based on the distance between the centroids of each class cluster. Suppose there is data set $X = \{x_i \mid i = 1,2,\ldots,n\}$ and the distance between any objects is represented by Euclidean distance, as follows:

$$d(x_i, x_j) = \sqrt{(x_i - x_j)^T (x_i - x_j)} \quad (2)$$

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a representative unsupervised machine learning method. It requires setting the radius of the neighborhood (*Eps*) and the density threshold of the neighborhood (*MinPts*). DBSCAN can divide the dataset into multiple class clusters by *Eps* and *MinPts*. When the data set changes, the appropriate centroids need to be redetermined to form new class clusters. A detailed description of the clustering can be found in [26] and will not be repeated in this paper.

2.2. Improved Feedforward Neural Network

In machine learning, a neural network is a basic algorithm, with the structure of feedforward neural network shown in Figure 1. This paper uses the improved variable learning rate neural network (INN) when generating the initial population. When using the feedforward neural network, it is necessary to continuously tune the parameters to find the appropriate learning rate. Therefore, many scholars have proposed the adaptive variable learning rate neural network. In [27], the authors proposed an improved variable learning rate algorithm, which can determine the change in step size by the difference of the cost function of two iterations. This algorithm has the advantage of fast convergence and low probability of oscillation.

The weight matrix of the hidden and output layers is denoted by b_{jk}, and the initial learning rate matrix is denoted by η, with a value range of [0,1]. $\eta(n)$ represents the learning rate at the n-th iteration, and $E(n)$ represents the global error at the n-th iteration. Sde is the step size adjustment coefficient. The dynamic adjustment rule for improving the overall learning rate is as follows:

$$\eta_{(n+1)} = \eta_{(n)} + \lg\left(E_{(n)} - E_{(n-1)} + 1\right) \times Sde \quad (3)$$

If the error $E(n)$ of the n-th iteration is less than the error of the $(n-1)$-th iteration, it indicates that the system has converged. In this case, the learning rate increment is positive and the $(n+1)$-th learning rate is equal to the current learning rate $\eta(n)$ plus a positive value, which

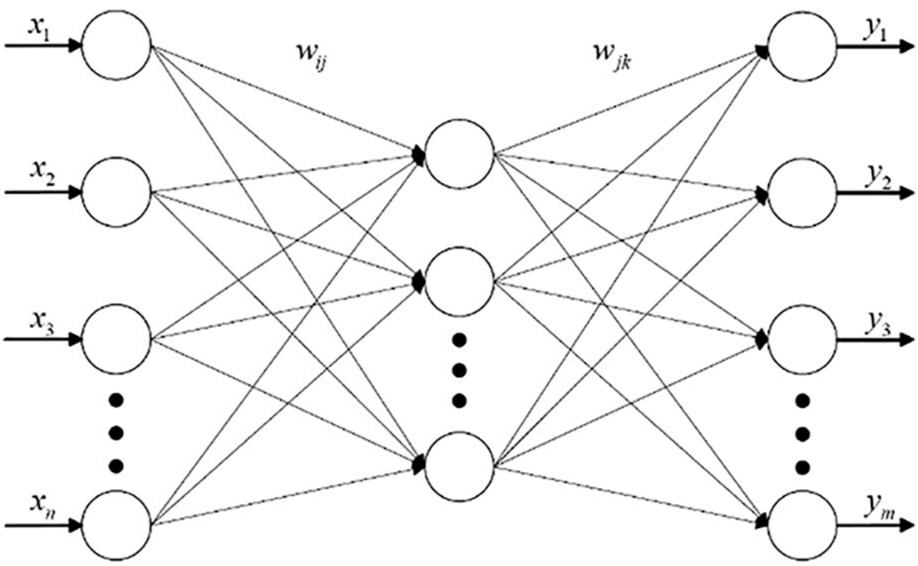


Figure 1. Structure of a feedforward neural network.

increases the learning rate. This way, the learning rate of b_{jk} is adjusted repeatedly until the global error is less than the target value.

3. Details of the IPPA

The population generated by IPPA consists of three parts: the first part is generated by feedforward neural network when the current environment is similar to that encountered before; the second part is generated by elite maintenance strategy to improve the convergence rate of the population; the third part is randomly generated by the algorithm to maintain population diversity.

3.1. Change Detection

There are a variety of commonly adopted environmental detection methods, but most of them fail to detect whether the new environment has been experienced before. When IPPA encounters the same environment, it reuses historical information to place the initial solutions closer to the expected solutions. Therefore, this paper sets up an environment vector, which cannot only detect whether the environment has changed, but also detect whether the new environment was encountered before.

First, we generate a set of random solutions within the decision space and perform non-dominated sorting on these solutions. Subsequently, when setting the environmental vector, we employ the *k*-means clustering algorithm to classify the non-dominated solutions into three classes. The clustering center points of each category are k_1, k_2 and k_3. For the decision variable x, each dimension of x has its upper and lower boundaries. If each dimension of x takes the upper bound value, then the point ub is formed. If each dimension of x takes lower bound value, then the point lb is formed. The target function values corresponding to these five points constitute the environment vector. For the DMOPs described by Equation 1, the environment vector at time t can be set as:

$$E(t) = (F(k_1,t), F(k_2,t), F(k_3,t), F(ub,t), F(lb,t)) \quad (4)$$

The value of environmental vector $E(t)$ will change with the change of the environment, and each environment uniquely corresponds to an environmental vector. Through testing, it was found that when the number of cluster centers is set to three, it is possible to capture the characteristics of solutions across different distribution regions within the decision space. When combined with the information of two boundary points, the resulting environmental vector can fully capture the diversity and variations of the environment. If the number of clusters is excessively small, it will lead to insufficient environmental discrimination. Conversely, an excessively large number of clusters will increase redundant computations. Therefore, the adoption of a 5-point configuration scheme can better balance discrimination and computational efficiency, thereby enabling IPPA to achieve optimal performance. After setting the environment vector, it can be used to detect environmental changes and determine whether the new environment is the same as a certain historical environment. Assume that the value of the environmental vector the last time was E_l and that the current value is E_c, then an environment change can be detected with the following formula:

$$\Delta E = \|E_l - E_c\| > \delta \quad (5)$$

where δ is the threshold that is set according to the problem. In order to effectively detect environmental changes, the value of δ in the experimental part of this paper is set to 0. The process to determine whether the new environment is the same as a certain historical environment is described in Section 3.3.

3.2. Training of INN

This paper uses the improved feedforward neural network (part 2.2). When an environmental change is detected, it is necessary to train the INN and save the relevant parameters. The input vector of INN in the IPPA algorithm is the environmental vector. Because the environment vector consists of the target function values corresponding to the five points, the dimension of the input vector of INN is $5 \times m$. Assume that the number of solutions is N and the dimension of decision variable is n. The number of output variables is $N \times n$. We can use the empirical formula to determine the number of neurons in the hidden layer, which is given by:

$$h = \sqrt{i + j} + a \tag{6}$$

where i is the number of nodes in the input layer, j is the number of nodes in the output layer, and a is the adjustment constant between 1 and 10.

In DMOP1, an environment with time $t = 1$ is selected and the standard solution in this environment is used to test the training effect of the neural network. The training sample is the population obtained before the environmental change. Among them, the number of training samples is 100, and the number of verification samples is 30. Assume that the decision vector is 20-dimensional; the target vector is 2-dimensional; the number of initialization individuals generated by INN is 100. From Equation 6, the value of h is between 46 and 55. Figure 2 shows the RMSE curve when the number of neurons in the hidden layer is 50. To ensure that the model converges to the target error, the number of neural network training times is set to 50. The RMSE can be less than 0.01 when the INN is trained to 15 steps, so we can set the number of training steps to 20. According to Table 1, when the number of neurons in the hidden layer is 50, the RMSE is the smallest.

Assuming that the environment vectors corresponding to all historical environments are divided into p classes by the DBSCAN clustering algorithm, there are p clustering centers M_i, $i = 1, 2, ..., p$. Each class corresponds to a neural network. The value of the environment vector last time (referring to the last stable environment before an environmental change occurs) was E_l. Now an environment change is detected and the solutions form the $N \times n$ dimensional vector Y. Calculate the minimum Euclidean distance between E_l and all cluster centers $O = \min\| E_l - M_i \| = \| E_l - M_j \|$, $i = 1, 2, ..., p$. If $O < Eps$, where Eps is the radius of the neighborhood in part 2.1, then take E_l as the input vector and Y as the output vector to train the j-th neural network (this training is performed by fine-tuning based on the original network parameters). If $O > Eps$, the clustering algorithm forms a new class and generates a corresponding neural network, the E_l and Y are used to train the new neural network.

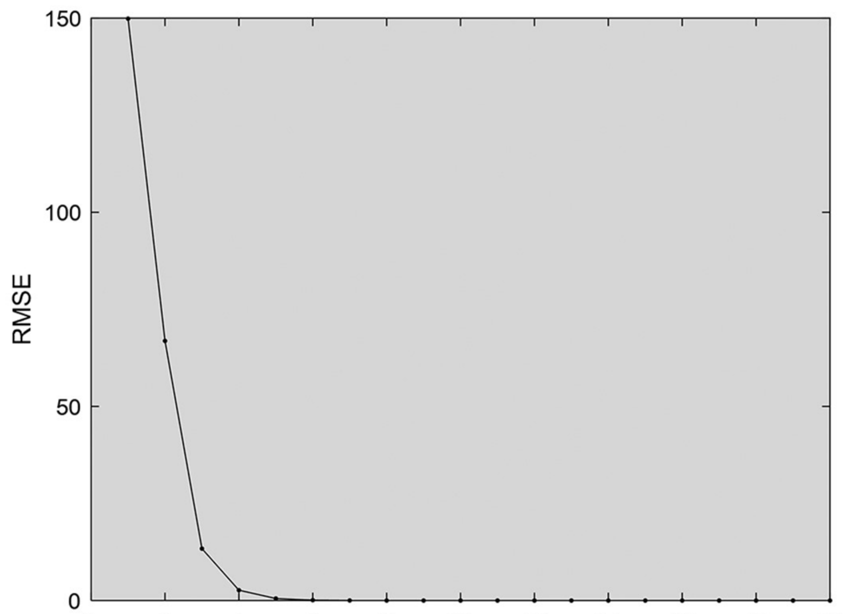


Figure 2. The training RMSE curve with 50 hidden neurons.

Table 1. Mean testing RMSE of 50 experiments with different number of hidden neurons.

Number of Hidden neurons	46	47	48	49	50	51	52	53	54	55
RMSE (10^{-2})	2.99	2.93	2.89	2.85	2.82	2.87	2.94	2.98	3.12	3.21

3.3. Predicting the Non-dominated Set by INN

Assuming that the current value of environment vector is E_c, and the environment vector values corresponding to all historical environments are divided into p classes by the DBSCAN clustering algorithm, there are p clustering centers M_i, $i = 1, 2, ..., p$. Each class corresponds to a neural network. When using DBSCAN, *Eps* and *MinPts* need to be set in advance.

Since there are widespread cases where an environment is not similar to any other environment, set *MinPts* = 1. The value of *Eps* is determined according to the following equation:

$$Eps = \frac{d\left(E_{\max}, E_{\min}\right)}{N_E} \tag{7}$$

where $E_{\max}$ and $E_{\min}$ are the upper and lower bounds of the environment vector $E(t)$. N_E is the maximum number of classes of environments. We classify all environments into at most N_E categories. $d(E_{\max}, E_{\min})$ can be calculated based on Equation 2.

When an environmental change is detected, we need to determine whether the current environment is similar to a historical environment. Calculate the minimum Euclidean distance between Ec and all cluster centers $O' = \min \| E_c - M_i \| = \| E_c - M_j \|$, $i = 1, 2, ..., p$. If $O' < Eps$, that means the new environment belongs to the j-th class and the new population can be predicted by the j-th neural network. At this point, the prediction relies on the historical mapping of similar environments. Even if there is a slight deviation in the predicted solutions, they can still be concentrated in the neighborhood of the POF. This is attributed to the fact that cluster centers reflect the core distribution regions of the POF, and boundary points define the effective range of the decision space, thereby avoiding the blind search caused by random initialization. If $O' > Eps$, the new environment will generate a new class in the neural network training stage (part 3.2), but the new population still can be predicted by the j-th neural network. Take the current value of environment vector Ec as the input of the j-th neural network and the population can be achieved from the output. The population generated by the neural network is called P_{INN}. Its structure diagram is shown in Figure 3. To sum up, the IPPA algorithm will find the category that best matches the new environment and use the corresponding INN to generate the P_{INN}.

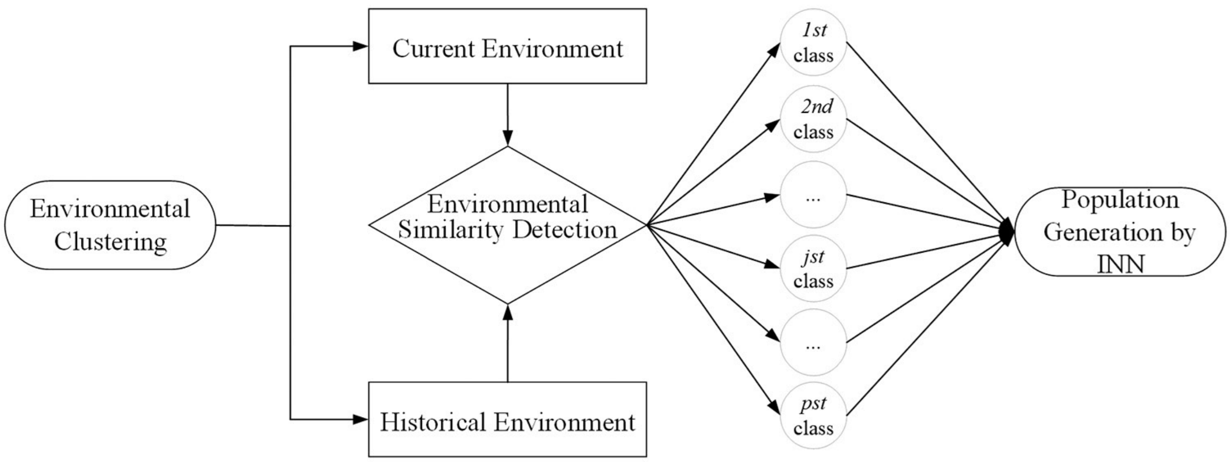


Figure 3. The algorithm structure schematic.

3.4. Elite Maintenance and Diversity Maintenance Strategy

When the new environment is similar to the previous environment, the non-dominant elite solutions before the environmental change has a certain guidance to the generation of the initial population after the environmental change, so we use the elite maintenance strategy to improve the convergence speed of the population. Assume the population size is N, to achieve an appropriate balance in the number of individuals generated by the elite strategy, different numbers of individuals were assigned to the experiment. It was observed that algorithm performance was optimized when P_E is $0.2N$. The $0.2N$ elite individuals are selected through non-dominated sorting and crowded distance calculation, prioritizing non-dominated solutions with larger crowded distances to ensure both convergence and diversity. Diversity maintenance strategies are pivotal for addressing DMOPs. In a fixed environment, populations tend to gradually converge, resulting in a loss of diversity. Within the context of dynamic multi-objective optimization, striking a balance is critical when implementing random strategies. If the population size generated by a random strategy is excessively small, the intended effect of the random strategy may be compromised. Conversely, an overly large population from such a strategy can lead to the random strategy becoming disproportionately dominant, potentially overshadowing the contributions of other strategies. Drawing on empirical insights, we therefore determined the size of the randomly generated population to achieve an optimal balance: specifically, the number of individuals in the randomly generated population P_R is set to $0.2N$.

3.5. Overall Framework of IPPA

This section introduces the overall algorithm framework of IPPA. The framework is shown in Algorithm 2.

Before using IPPA to solve DMOPs, some parameters need to be set first: the population size N; the number of decision variables n; upper bound and lower bound of decision variables $u = (u_1, u_2, ..., u_n)$, $l = (l_1, l_2, ..., l_n)$; the number of objective functions m; environment vector $E(t)$; threshold of environment change δ. The setting method of $E(t)$ and δ are described in part 3.1. Parameter setting of INN is described in part 3.2.

In Algorithm 2, it is first detected whether the environment has changed (line 2). If the environment changes, the INN is trained with the population before the environment changes and its parameters are stored (line 3), then form the population P_{INN} based on the INN (line 5−9), P_E based on the elite maintenance strategy (line 10−11) and P_R based on the random algorithm (line 12). The initial population P_I is generated from P_{INN}, P_E and P_R (line 12). Lines 5−12 are the change reaction steps.

4. Test Instances and Performance Metrics

4.1. Test Problems

The benchmarks used in this paper are the dMOP test suite [29] and FDA test suite [30]. In these experiments, there are three targets for both FDA4 and FDA5. Since IPPA mainly solves the problem of dMOP with irregular environmental changes, we make t in these experiments change irregularly with time (irregular change refers to no fixed cycle and no clear trend of environmental parameter t, which is realized by randomly generating change nodes) to achieve the effect of random environmental changes.

4.2. Performance Metrics

There are many methods to evaluate the algorithm such as the inverted generational distance (IGD), Hypervolume difference (HVD) and Schott's spacing metric (SP). We first introduce these three performance metrics for static algorithms and then modify them to those for dynamic algorithms: mean IGD (MIGD), mean HVD (MHVD) and mean SP (MSP) [31]. In actual operation, we take the average value and mean square deviation of performance indicators for repeated experiments. This method can avoid the chance of a single experiment and make the experimental data more convincing.

Algorithm 2. Overall framework of IPPA.

1. Initialization: Number of time change, $t = 0$; optimized time recorder, $T_r = 0$; total optimization time, T_{max}; environment detection vector, E; random initial population, P_I; population size, N.
2. **while** $T_r \leq T_{max}$
3. **if** *an environment change is detected (part 3.1)* **then**
4. Train the INN (improved variable learning rate neural network bound to the category of the previous environment) with the population and environment vector at the time before environmental change. (part 3.2)
5. Calculate the distance between the current environment vector and the historical environment vectors.
6. **if** *the current environment belongs to a certain class (assume the j-th class) of historical environment (part 3.3)*
7. Produce population P_{INN} with the j-th INN. (part 3.3)
8. **else**
9. Find the class that most closely matches the new environment and produce population P_{INN} with the corresponding INN. (part 3.3)
10. **end**
11. Generate population P_E with elite maintenance strategy. (part 3.4).
12. Randomly generate population P_R (part 3.5).
13. Combine P_E, P_{INN} and P_R to form population P_{all}. Select N individuals from P_{all} based on non-dominated sorting and crowded distance sorting. The N individuals represent the initial population P_I.
14. Optimize the population P_I with the static optimization algorithm IMOPEO-PLM [28] for one step.
15. **else**
16. Optimize the population P_I with the static optimization algorithm IMOPEO-PLM [28] for one step.
17. **end**
18. **end**
19. The non-dominated solutions in P_I are the best solutions find so far and stop.

Assume P^{t*} is a set of Pareto optimal points that uniformly distributed in the POF. P^t is a set of points near the POF. IGD is defined as:

$$\mathrm{IGD}\left(P^t, P^{t*}\right) = \frac{\sum_{v \in P^{t*}} d\left(v, P^t\right)}{\left|P^{t*}\right|} \tag{8}$$

where $d\left(v, P^t\right) = \min_{u \in P^t} \|F(v) - F(u)\|$ is the distance between v and P^t; $|P^{t*}|$ is the number of points in P^{t*}. The IGD is a comprehensive evaluation index that can measure both diversity and convergence.

HVD is defined as:

$$\mathrm{HVD}(P^t, P^{t*}) = \mathrm{HV}(P^{t*}) - \mathrm{HV}(P^t) \tag{9}$$

where HV(S) is the hypervolume of a set S. The reference point for the computation of hypervolume is $(z_1 + 0.5, z_2 + 0.5, ..., z_m + 0.5)$, where z_j is the maximum value of the j-th objective of the true POF and m is the number of objectives.

SP is defined as:

$$\mathrm{SP}\left(P^t\right) = \frac{1}{\left|P^t\right| - 1} \sum_{i=1}^{\left|P^t\right|} \left(Di - \overline{D}\right)^2 \tag{10}$$

where D_i is the Euclidean distance between the i-th member in P^t, and its nearest member in P^t; $\overline{D}$ is the average value of D_i, SP measures how evenly the solutions in $|P^t|$ are distributed.

The MIGD, MHVD and MSP are defined as the mean value of IGD, GD and SP in some time steps for a DMOP respectively

$$\mathrm{MIGD} = \frac{1}{|T|}\sum_{t\in T}\mathrm{IGD}\left(P^t, P^{t*}\right) \quad (11)$$

$$\mathrm{MHVD} = \frac{1}{|T|}\sum_{t\in T}\mathrm{HVD}\left(P^t, P^{t*}\right) \quad (12)$$

$$\mathrm{MSP} = \frac{1}{|T|}\sum_{t\in T}\mathrm{SP}\left(P^t\right) \quad (13)$$

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

5. Experimental Results

This section presents the experimental results of IPPA and four contrasting algorithms in the dMOP test suite and FDA test suite. The averages of their evaluation indicators are presented in the form of a table. In the table of statistical results, the best results of each problem are marked in bold. To evaluate whether the performance advantage of the best algorithm, compared to other algorithms, has statistical significance across all test instances, we conducted the Wilcoxon rank-sum test at a significance level of 0.05. Here, the symbol "+" indicates that the algorithm performs significantly better than the compared algorithm in the corresponding test instance. Both POS and POF of dMOP2 and FDA5 change, so dMOP2 and FDA5 are challenging. We plot the distribution of initial and final populations for each algorithm on dMOP2 and FDA5. The main purpose of these experiments is to compare the dynamic optimization effects of algorithms, so their static optimization algorithms all use IMOPEO-PLM.

5.1. Compared Strategies and Parameter Settings

The three main parts in the experiment are as follows:

Change detection: Detects whether the environment changes based on the changes that occur in the environment vector.

Change reaction: When the environment is changed, the initial population will be generated using one of the following algorithms:

1. a novel initial population prediction algorithm (IPPA), which is proposed in this paper;
2. combining a hybrid prediction strategy and a precision controllable mutation strategy (HPPCM) [32], which coordinates the center point-based prediction and the guiding individual-based prediction;
3. a hybrid of memory and prediction strategies (MOEA/D-HMPS) [33], which is the prediction based on the previous two consecutive population centers;
4. an improved memory prediction strategy (MOEA/D-MP) [21], adopts a sensor-based method to detect the environment change and find a similar one in history to reuse the information of it in the prediction process;
5. an algorithm based on grey prediction model GM(1,1) (GM-DMOP) [18], which is a predictive method based on grey prediction model.

Static optimization: the extremal optimization algorithm IMOPEO-PLM [38] is chosen to optimize the population because of its speed.

The problem and algorithm parameters are set as follows:

1. The severity of environmental change n_τ is set to 10 and 20. The population size $N = 100$.
2. We ran each algorithm 20 times for each test problem. The environment changes every 20 seconds. In order to avoid the situation that some environments are encountered many times in an experiment, while some environments cannot be encountered, we added constraints to t: for the experiment of $n_\tau = 10$, when $1 \le t \le 20$, t traverses $1 \sim 20$, and so on; for the experiment of $n_\tau = 20$, when $1 \le t \le 40$, t traverses $1 \sim 40$, and so on. For the FDA and dMOP test suites, the environment changed 80 times, so we set $n_\tau = 10$ and 20, which means each environment value was encountered 4 and 2 times respectively.
3. According to the properties of the dMOP test suite and FDA test suite, threshold δ to determine whether the environment has changed is set to be 0.

4. Parameters in IPPA: the number of training steps is 20; the number of neurons in the hidden layer is 50; the number of training times of the neural network is 50; the value of MinPts is 1; the value of *Eps* is 0.1; the value of N_E is 10; the population size generated by the neural network is *N*; the population size generated by elite maintenance strategy is 0.2*N*; the population size generated by random strategy is 0.2*N*.
5. Parameters in HPPCM: the algebra Δt of autonomous evolution is set to 2; the parameter *r* of the Random(*r*) function is set to 2.
6. Parameters in MOEA/D-HMPS: the threshold ε of change identification is set to 10^{-4}; the maximum memory size of the two archive sets D and C was set to 50.
7. Parameters in MOEA/D-MP: the environmental change detection method adopts the method proposed in this paper.
8. Parameters in GM-DMOP: when the environment changes, 50% of the population members are generated using the prediction strategy based on centroid points; 30% and 20% of population members are chosen from previous generation and randomly generated in the search space, respectively.

5.2. Computational Complexity Analysis

Complexity can reflect the computational cost of an algorithm and thus the response speed of the algorithm. For IPPA, the number of calculations of function as follows: in the change detection step (line 2 of Algorithm 2), it is 10; in the neural network training step (line 3 of Algorithm 2), it is $10T_{train}$ (where T_{train} is the number of neural network training times, and we set T_{train} to 50); in the environmental comparison step (line 4 of Algorithm 2), it is $5N_E+1$ (where N_E takes a value between 10 and 20 in this paper); in the initial solution generation step (lines 5−12 of Algorithm 2), it is 1.4*N*.

The number of calculations of function for the dynamic algorithm of IPPA is $10T_{train}+5N_E+1.4N+11$. For HPPCM, it is $(2n+3)N+5$. For MOEA/D-HMPS, it is $(n+1)N+5$. For MOEA/D-MP, it is $(n+1)N+5$. For GM-DMOP, it is $4N+5$.

Based on the rough calculation results, the complexities of IPPA, MOEA/D-HMPS, and MOEA/D-MP are similar, while HPPCM has the highest complexity and GM-DMOP has the lowest. From this, the neural network training does not excessively increase the computational complexity of IPPA, and the complexity of IPPA is at a good level.

5.3. Comparison with Algorithms without Memory Strategy on Modified Tests

This section compares and analyzes the experimental results of IPPA, HPPCM, and GM-DMOP. All the tests were run 20 times and the mean and standard deviation values for IGD, HVD and SP are shown in Tables 2–4 respectively. For each test, the environment is changed 80 times, where there are four cycles when $n_\tau=10$ and two cycles when $n_\tau=20$. A cycle starts right after an environmental change and ends just before the next environmental change (for instance, the period between $t=1$ and $t=20$ constitutes one cycle when $n_\tau=10$). The problems with $n_\tau=10$ and 20 encounter each environment value 4 and 2 times respectively, so it can be divided into four phases and two phases respectively.

Table 2 shows that when $21\leq t\leq 40$ ($n_\tau=10$), $41\leq t\leq 60$ ($n_\tau=10$), $61\leq t\leq 80$ ($n_\tau=10$) and $41\leq t\leq 80$ ($n_\tau=20$), the performance of IPPA is not much different from other algorithms on dMOP1 and FDA2. In complex problems with drastic changes in POF such as dMOP2 and FDA5, the IGD value of IPPA was significantly lower than that of HPPCM, and the SP value was significantly lower than that of GM-DMOP. Moreover, in 20 repeated experiments, IPPA maintained a stable advantage in most scenarios. On FDA4 and FDA5, HPPCM and GM-DMOP performed much worse than IPPA. IGD of IPPA in cycles 2 to 4 ($21\leq t\leq 80$, $n_\tau=10$) and in cycle 2 ($41\leq t\leq 80$, $n_\tau=20$) changes slightly. The characteristics of the data in Table 3 are generally consistent with those in Table 2. On dMOP1 and FDA2, the performance of HPPCM and GM-DMOP are similar to that of IPPA. Other than that, the performance of HPPCM and the performance of GM-DMOP are poor. In Table 4, it is important to note that the value of SP of GM-DMOP showed a large fluctuation on FDA3.

Table 2. Mean and SD values of IGD indicator obtained by three algorithms.

Problem	Algorithm	$n_\tau = 10$				$n_\tau = 20$	
		$1 \le t \le 20$ Mean(SD)	$21 \le t \le 40$ Mean(SD)	$41 \le t \le 60$ Mean(SD)	$61 \le t \le 80$ Mean(SD)	$1 \le t \le 40$ Mean(SD)	$41 \le t \le 80$ Mean(SD)
dMOP1	IPPA	**0.0047(0.0004)**	**0.0044(0.0000)**	**0.0043(0.0000)**	**0.0043(0.0000)**	**0.0135(0.0125)**	**0.0044(0.0000)**
	HPPCM	0.0667(0.0718)+	**0.0044(0.0000)**	0.0044(0.0000)	0.0044(0.0000)	0.0592(0.0705)+	**0.0044(0.0000)**
	GM-DMOP	0.2750(0.0312)+	0.0045(0.0001)	0.0045(0.0001)	0.0045(0.0000)	0.0334(0.0064)+	0.0046(0.0002)
dMOP2	IPPA	**0.0053(0.0011)**	**0.0044(0.0000)**	**0.0044(0.0000)**	**0.0044(0.0000)**	**0.0057(0.0004)**	**0.0044(0.0000)**
	HPPCM	0.0410(0.0386)+	0.0191(0.0104)+	0.0132(0.0016)+	0.0121(0.0020)+	0.0373(0.0144)+	0.0164(0.0036)+
	GM-DMOP	0.2961(0.0751)+	0.0972(0.1031)+	0.0514(0.0524)+	0.1040(0.1159)+	0.0306(0.0176)+	0.00118(0.0017)+
dMOP3	IPPA	**0.0054(0.0005)**	**0.0044(0.0000)**	**0.0044(0.0000)**	**0.0043(0.0000)**	**0.0059(0.0013)**	**0.0044(0.0000)**
	HPPCM	0.0280(0.0153)+	0.0123(0.0053)+	0.0151(0.0099)+	0.0211(0.0118)+	0.0300(0.0224)+	0.0156(0.0125)+
	GM-DMOP	0.0087(0.0018)+	0.0068(0.0002)+	0.0064(0.0002)+	0.0068(0.0010)+	0.0327(0.0073)+	0.0321(0.0063)+
FDA1	IPPA	**0.0070(0.0020)**	**0.0044(0.0000)**	**0.0044(0.0000)**	**0.0044(0.0000)**	**0.0050(0.0003)**	**0.0043(0.0000)**
	HPPCM	0.0177(0.0153)+	0.0065(0.0001)+	0.0068(0.0003)+	0.0068(0.0007)+	0.0359(0.0206)+	0.0319(0.0036)+
	GM-DMOP	0.0134(0.0050)+	0.0080(0.0007)+	0.0074(0.0005)+	0.0078(0.0006)+	0.0103(0.0017)+	0.0091(0.0014)+
FDA2	IPPA	**0.0048(0.0004)**	0.0045(0.0001)	**0.0044(0.0000)**	**0.0044(0.0000)**	**0.0058(0.0016)**	**0.0044(0.0000)**
	HPPCM	0.0150(0.0051)+	**0.0044(0.0000)**	**0.0044(0.0000)**	**0.0044(0.0000)**	0.0098(0.0034)+	**0.0044(0.0000)**
	GM-DMOP	0.0278(0.0095)+	0.0047(0.0003)	0.0045(0.0000)	0.0045(0.0000)	0.0202(0.0046)+	0.0045(0.0000)
FDA3	IPPA	**0.0029(0.0002)**	**0.0028(0.0001)**	**0.0025(0.0000)**	**0.0028(0.0000)**	**0.0029(0.0001)**	**0.0026(0.0001)**
	HPPCM	0.0567(0.0684)+	0.0066(0.0038)+	0.0079(0.0052)+	0.0215(0.0169)+	0.0079(0.0029)+	0.0080(0.0030)+
	GM-DMOP	0.0138(0.0119)+	0.0041(0.0003)+	0.0052(0.0017)+	0.0040(0.0001)+	0.0183(0.0098)+	0.0115(0.0048)+
FDA4	IPPA	**0.0776(0.0091)**	**0.0634(0.0003)**	**0.0630(0.0008)**	**0.0630(0.0006)**	**0.0707(0.0013)**	**0.0627(0.0004)**
	HPPCM	0.1811(0.0403)+	0.1476(0.0165)+	0.1440(0.0015)+	0.1368(0.0111)+	0.2869(0.1052)+	0.2593(0.0933)+
	GM-DMOP	0.1668(0.0097)+	0.1176(0.0116)+	0.1128(0.0100)+	0.1404(0.0145)+	0.1599(0.0272)+	0.1693(0.0301)+
FDA5	IPPA	**0.1648(0.0084)**	**0.1334(0.0059)**	**0.1291(0.0030)**	**0.1224(0.0059)**	**0.1499(0.0034)**	**0.1265(0.0016)**
	HPPCM	0.2792(0.0188)+	0.2224(0.0244)+	0.2313(0.0355)+	0.2440(0.0178)+	0.2793(0.0626)+	0.3140(0.0479)+
	GM-DMOP	0.2181(0.0064)+	0.2054(0.0100)+	0.2106(0.0054)+	0.2003(0.00968)+	0.2346(0.0124)+	0.2104(0.0044)+

Table 3. Mean and SD values of HVD indicator obtained by three algorithms.

Problem	Algorithm	$n_\tau = 10$				$n_\tau = 20$	
		$1 \le t \le 20$ Mean(SD)	$21 \le t \le 40$ Mean(SD)	$41 \le t \le 60$ Mean(SD)	$61 \le t \le 80$ Mean(SD)	$1 \le t \le 40$ Mean(SD)	$41 \le t \le 80$ Mean(SD)
dMOP1	IPPA	**0.0723(0.0021)**	**0.0703(0.0019)**	0.0709(0.0016)	0.0705(0.0009)	**0.0730(0.0028)**	**0.0705(0.0007)**
	HPPCM	0.0821(0.0054)+	0.0705(0.0009)	0.0714(0.0003)	0.0715(0.0007)	0.0768(0.0029)	0.0713(0.0009)
	GM-DMOP	0.0982(0.0084)+	0.0709(0.0018)	**0.0698(0.0018)**	**0.0692(0.0005)**	0.0763(0.0020)	0.0713(0.0008)
dMOP2	IPPA	**0.0715(0.0015)**	**0.0706(0.0006)**	**0.0694(0.0003)**	**0.0704(0.0003)**	**0.0714(0.0002)**	**0.0705(0.0004)**
	HPPCM	0.0875(0.0087)+	0.0821(0.0063)+	0.0795(0.0012)+	0.0804(0.0014)+	0.0888(0.0059)+	0.0823(0.0022)+
	GM-DMOP	0.1333(0.0286)+	0.1092(0.0336)+	0.1036(0.0352)+	0.1252(0.0621)+	0.0853(0.0029)+	0.0780(0.0009)+
dMOP3	IPPA	**0.0858(0.0010)**	**0.0843(0.0003)**	**0.0842(0.0006)**	**0.0845(0.0001)**	**0.0854(0.0000)**	**0.0845(0.0006)**
	HPPCM	0.0974(0.0033)+	0.0905(0.0050)	0.0943(0.0084)+	0.0975(0.0089)+	0.1036(0.0166)+	0.0940(0.0112)+
	GM-DMOP	0.0890(0.0036)+	0.0867(0.0009)	0.0858(0.0004)	0.0873(0.0010)	0.1061(0.0066)+	0.1081(0.0063)+
FDA1	IPPA	**0.0877(0.0022)**	**0.0839(0.0004)**	**0.0843(0.0003)**	**0.0844(0.0004)**	**0.0850(0.0003)**	**0.0844(0.0002)**
	HPPCM	0.0994(0.0178)+	0.0869(0.0010)	0.0867(0.0006)	0.0868(0.0012)	0.1115(0.0180)+	0.1063(0.0018)+
	GM-DMOP	0.0926(0.0040)+	0.0877(0.0015)	0.0867(0.0006)	0.0862(0.0013)	0.0889(0.0012)	0.0877(0.0005)
FDA2	IPPA	**0.0710(0.0012)**	0.0716(0.0002)	**0.0706(0.0006)**	0.0703(0.0002)	**0.0722(0.0017)**	0.0712(0.0002)
	HPPCM	0.0801(0.0025)+	0.0716(0.0006)	0.0710(0.0007)	0.0711(0.0011)	0.0755(0.0030)	0.0706(0.0004)
	GM-DMOP	0.0862(0.0032)+	**0.0704(0.0017)**	0.0720(0.0014)	**0.0700(0.0025)**	0.0812(0.0007)+	**0.0702(0.0004)**
FDA3	IPPA	**0.0748(0.0014)**	**0.0734(0.0007)**	**0.0716(0.0008)**	**0.0735(0.0010)**	**0.0747(0.0009)**	**0.0724(0.0004)**
	HPPCM	0.0963(0.0099)+	0.0869(0.0050)+	0.0862 (0.0103)+	0.0920(0.0121)+	0.0858(0.0038)+	0.0890(0.0041)+
	GM-DMOP	0.0898(0.0069)+	0.0792(0.0013)	0.0831(0.0039)+	0.0817(0.0010)+	0.0984(0.0129)+	0.0925(0.0090)+
FDA4	IPPA	**0.0238(0.0049)**	**0.0177(0.0003)**	**0.0174(0.0004)**	**0.0160(0.0005)**	**0.0217(0.0005)**	**0.0166(0.0002)**
	HPPCM	0.0650(0.0111)+	0.0597(0.0083)+	0.0575(0.0020)+	0.0540(0.0048)+	0.0762(0.0165)+	0.0714(0.0149)+
	GM-DMOP	0.0605(0.0041)+	0.0453(0.0064)+	0.0435(0.0064)+	0.0544(0.0062)+	0.0571(0.0102)+	0.0591(0.0063)+
FDA5	IPPA	**0.0641(0.0017)**	**0.0557(0.0018)**	**0.0532(0.0026)**	**0.0503(0.0019)**	**0.0600(0.0005)**	**0.0525(0.0022)**
	HPPCM	0.0938(0.0082)+	0.0851(0.0069)+	0.0838(0.0075)+	0.0897(0.0063)+	0.0906(0.0078)+	0.0963(0.0055)+
	GM-DMOP	0.0816(0.0011)+	0.0760(0.0016)+	0.0802(0.0021)+	0.0757(0.0012)+	0.0843(0.0024)+	0.0795(0.0023)+

Table 4. Mean and SD values of SP indicator obtained by three algorithms.

Problem	Algorithm	$n_\tau = 10$				$n_\tau = 20$	
		$1 \le t \le 20$ Mean(SD)	$21 \le t \le 40$ Mean(SD)	$41 \le t \le 60$ Mean(SD)	$61 \le t \le 80$ Mean(SD)	$1 \le t \le 40$ Mean(SD)	$41 \le t \le 80$ Mean(SD)
dMOP1	IPPA	**0.0065(0.0001)**	**0.0064(0.0001)**	**0.0062(0.0001)**	**0.0062(0.0000)**	**0.0071(0.0007)**	**0.0063(0.0000)**
	HPPCM	0.0122(0.0036)+	0.0065(0.0001)	0.0065(0.0000)	0.0064(0.0001)	0.0111(0.0035)+	0.0065(0.0001)
	GM-DMOP	0.0289(0.0175)+	0.0070(0.0003)	0.0069(0.0003)	0.0068(0.0000)	0.0096(0.0008)+	0.0073(0.0006)+
dMOP2	IPPA	**0.0069(0.0007)**	**0.0061(0.0001)**	**0.0063(0.0001)**	**0.0064(0.0000)**	**0.0067(0.0001)**	**0.0061(0.0000)**
	HPPCM	0.0177(0.0040)+	0.0161(0.0022)+	0.0145(0.0015)+	0.0144(0.0015)+	0.0205(0.0030)+	0.0147(0.0010)+
	GM-DMOP	0.0414(0.0099)+	0.0264(0.0128)+	0.0222(0.0133)+	0.0291(0.0209)+	0.0168(0.0024)+	0.0138(0.0010)+
dMOP3	IPPA	**0.0073(0.0012)**	**0.0063(0.0001)**	**0.0063(0.0001)**	**0.0063(0.0001)**	**0.0072(0.0013)**	**0.0062(0.0001)**
	HPPCM	0.0159(0.0013)+	0.0121(0.0030)+	0.0142(0.0051)+	0.0146(0.0053)+	0.0171(0.0056)+	0.0120(0.0054)+
	GM-DMOP	0.0100(0.0011)+	0.0085(0.0004)+	0.0081(0.0003)+	0.0083(0.0011)+	0.0203(0.0036)+	0.0231(0.0027)+
FDA1	IPPA	**0.0087(0.0024)**	**0.0061(0.0001)**	**0.0064(0.0000)**	**0.0064(0.0001)**	**0.0064(0.0004)**	**0.0062(0.0000)**
	HPPCM	0.0120(0.0040)+	0.0082(0.0004)+	0.0086(0.0003)+	0.0090(0.0014)+	0.0208(0.0062)+	0.0234(0.0005)+
	GM-DMOP	0.0120(0.0019)+	0.0097(0.0009)+	0.0091(0.0008)+	0.0095(0.0004)+	0.0114(0.0015)+	0.0359(0.0358)+
FDA2	IPPA	**0.0067(0.0005)**	**0.0061(0.0000)**	**0.0059(0.0000)**	**0.0059(0.0000)**	**0.0069(0.0004)**	**0.0062(0.0000)**
	HPPCM	0.0092(0.0004)+	0.0063(0.0000)	0.0064(0.0001)	0.0065(0.0000)+	0.0080(0.0006)+	0.0063(0.0000)
	GM-DMOP	0.0113(0.0022)+	0.0067(0.0001)	0.0066(0.0001)+	0.0067(0.0001)+	0.0097(0.0006)+	0.0067(0.0000)
FDA3	IPPA	**0.0084(0.0005)**	**0.0070(0.0000)**	**0.0070(0.0003)**	**0.0068(0.0000)**	**0.0085(0.0008)**	**0.0067(0.0001)**
	HPPCM	0.0233(0.0065)+	0.0243(0.0070)+	0.0215(0.0079)+	0.0206(0.0103)+	0.0211(0.0034)+	0.0238(0.0051)+
	GM-DMOP	0.0202(0.0017)+	0.0710(0.0329)+	0.0169(0.0014)+	0.0144(0.0016)+	0.0745(0.0646)+	0.0472(0.0234)+
FDA4	IPPA	**0.3776(0.0321)**	**0.3212(0.0042)**	**0.3234(0.0023)**	**0.3181(0.0048)**	**0.3407(0.0045)**	**0.3189(0.0030)**
	HPPCM	0.4333(0.0484)+	0.4066(0.0100)+	0.4047(0.0072)+	0.4016(0.0129)+	0.4771(0.0534)+	0.4931(0.0765)+
	GM-DMOP	0.4301(0.0164)+	0.4011(0.0167)+	0.4126(0.0195)+	0.4786(0.0454)+	0.4504(0.0286)+	0.4562(0.0306)+
FDA5	IPPA	**0.6522(0.0186)**	**0.5668(0.0218)**	**0.5665(0.0090)**	**0.5638(0.0260)**	**0.6227(0.0069)**	**0.5533(0.0034)**
	HPPCM	0.7252(0.0132)+	0.6798(0.0208)+	0.6803(0.0257)+	0.6911(0.0305)+	0.6955(0.0263)+	0.7177(0.0261)+
	GM-DMOP	0.7144(0.0264)+	0.6759(0.0206)+	0.6873(0.0141)+	0.7076(0.0318)+	0.6887(0.0134)+	0.6845(0.0180)+

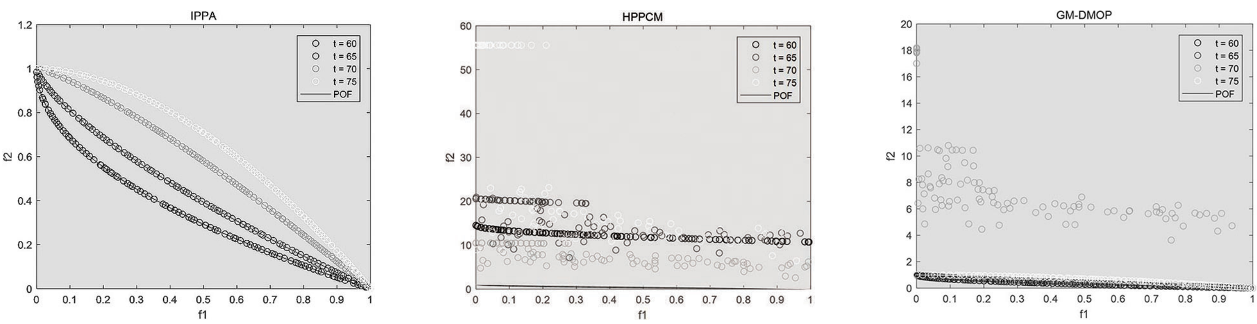
Table 5 shows the performance metrics for the initial populations of the three algorithms on dMOP2 ($n_\tau = 10$) and FDA5 ($n_\tau = 10$). On dMOP2 ($n_\tau = 10$), all three performance metrics of IPPA are significantly better than those of the other two algorithms. The value of IGD of HPPCM is larger compared to that of GM-DMOP. The HVD values of HPPCM and GM-DMOP in FDA4 and FDA5 are 2−3 times those of IPPA.HPPCM and GM-DMOP lack historical environment memory and only rely on recent solution guidance to generate the initial population. When the environment switches from a low-dimensional POF (FDA1) to a high-dimensional POF (FDA5 with 3 objectives), the recent solutions deviate significantly from the POF of the new environment, leading to substantial hypervolume (HV) loss of the final population (reflected by high HVD values). In contrast, IPPA learns the POF characteristics of multiple environments through INN. Even when switching to a high-dimensional environment, its initial population can still be close to the optimal front, resulting in lower HVD values.

The value of SP of HPPCM is smaller compared to the value of SP of GM-DMOP at $t = 60$ and $t = 65$, and the value of SP for GM-DMOP is smaller at $t = 70$ and $t = 75$. Similar to the data characteristics on dMOP2 ($n_\tau = 10$), all three performance metrics of IPPA are significantly better than the three performance metrics of the other two algorithms on FDA5 ($n_\tau = 10$). We compare HPPCM with GM-DMOP. The value of IGD of GM-DMOP is smaller, the value of SP of HPPCM is smaller, and the value of HVD of HPPCM is similar to that of GM-DMOP.

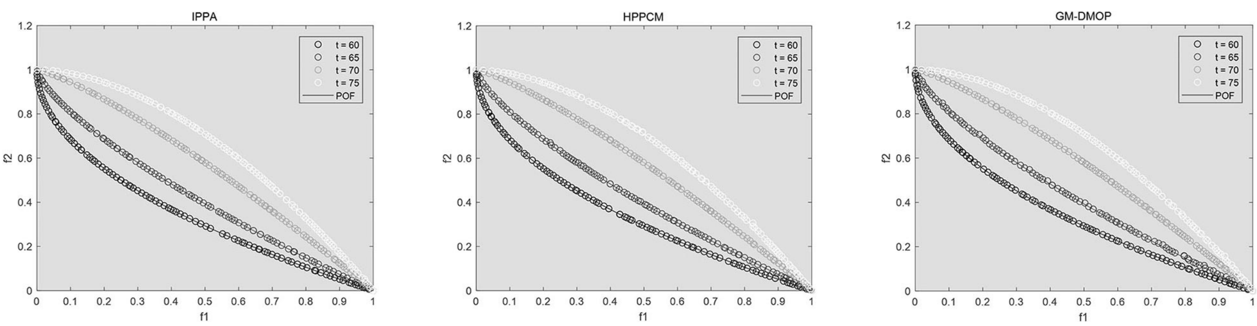
Figure 4 shows the initial and final populations of IPPA, HPPCM and GM-DMOP at $t = 60$, 65, 70, and 75 on dMOP2 ($n_\tau = 10$). The final population of all three algorithms is close to POF. However, for the initial population, only IPPA is close to POF. The initial population of GM-DMOP is closer to POF than that of HPPCM.

Table 5. The value of three performance metrics and their SD obtained by three algorithms on dMOP2 ($n_\tau = 10$) and FDA5 ($n_\tau = 10$).

Algorithm	dMOP2 (n_τ = 10)				FDA5 (n_τ = 10)	
	t = 60 IGD(SD)	t = 65 IGD(SD)	t = 70 IGD(SD)	t = 75 IGD(SD)	t = 60 IGD(SD)	t = 70 IGD(SD)
IPPA	**0.0971(0.0786)**	**0.0476(0.0314)**	**0.0552(0.0638)**	**0.0633(0.0806)**	**0.4973(0.1747)**	**0.3249(0.1264)**
HPPCM	4.6869(0.7339)+	4.7153(1.1270)+	6.6857(5.0639)+	6.3135(2.9728)+	2.9008(1.6508)+	1.7385(1.1038)+
GM-DMOP	3.1512(2.7305)+	3.2436(1.4539)+	3.3732(0.2297)+	1.8034(1.6230)+	1.4124(0.8991)+	0.7675(0.4237)+
	HVD(SD)	HVD(SD)	HVD(SD)	HVD(SD)	HVD(SD)	HVD(SD)
IPPA	**0.0276(0.1192)**	**0.0463(0.0443)**	**0.0983(0.0608)**	**0.0025(0.0726)**	**0.0676(0.0413)**	**0.0765(0.0323)**
HPPCM	0.2671(0.0073)+	0.4217(0.2353)+	0.2534(0.0049)+	0.2500(0.0000)+	0.1319(0.0008)+	0.1307(0.0041)+
GM-DMOP	0.4509(0.3065)+	0.4914(0.3197)+	0.2548(0.0068)+	0.2384(0.0479)+	0.1326(0.0006)+	0.1120(0.0240)+
	SP(SD)	SP(SD)	SP(SD)	SP(SD)	SP(SD)	SP(SD)
IPPA	**0.0061(0.0004)**	**0.0118(0.0072)**	**0.0133(0.0092)**	**0.0100(0.0051)**	**1.0257(0.2992)**	**0.7450(0.1390)**
HPPCM	1.6802(0.5581)+	2.7358(2.3262)+	6.4181(5.2542)+	8.3994(1.4394)+	1.6550(0.3721)+	2.3201(1.7200)+
GM-DMOP	2.7935(2.1605)+	3.1808(2.0506)+	4.1886(2.6325)+	3.2035(1.8123)+	2.3968(0.4983)+	2.5166(1.3068)+



(a) The initial populations at t = 60, 65, 70, 75.



(b) The final populations at t = 60, 65, 70, 75.

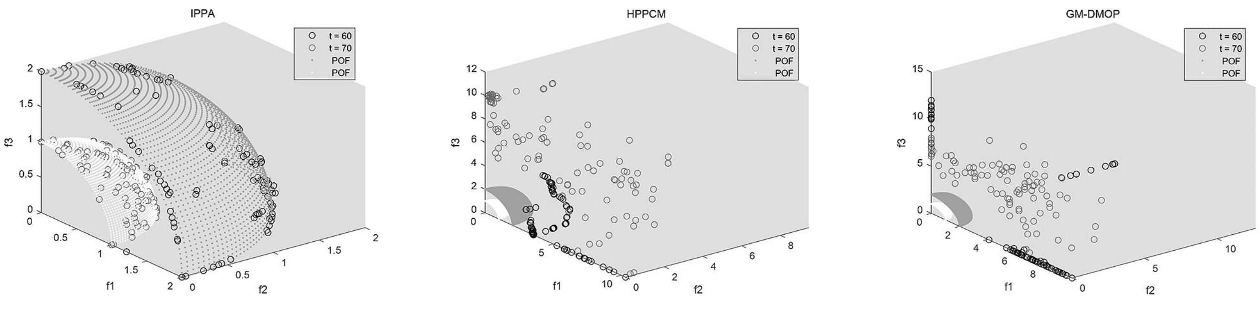
Figure 4. The initial and final populations at t = 60, 65, 70, 75 obtained by IPPA, HPPCM and GM-DMOP on dMOP2 (n_τ = 10).

Figure 5 shows the initial and final populations of IPPA, HPPCM and GM-DMOP at $t=60$, 70 on FDA5 ($n_\tau=10$). Both the initial and final populations of IPPA are closest to POF. For the initial population, both HPPCM, and GM-DMOP are far from the POF. For the final population, GM-DMOP is closer to the POF than HPPCM.

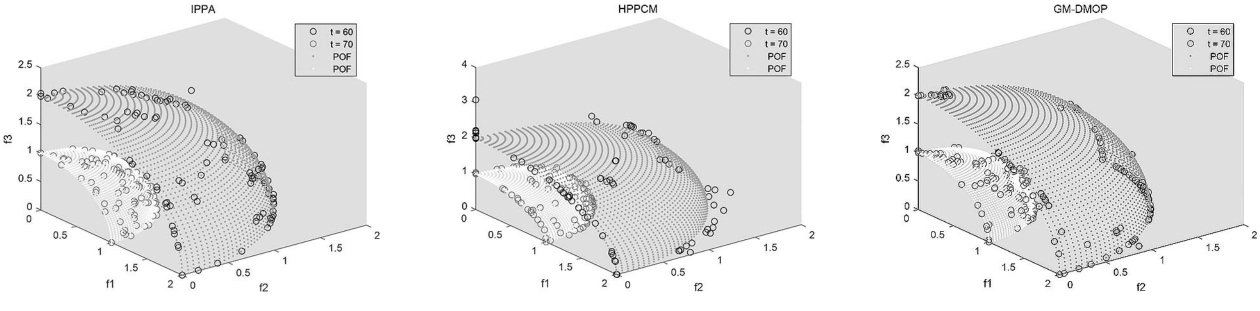
IPPA uses the INN, which needs to be trained in the first cycle of each test. The performance of IPPA gradually improves from the start of the second cycle. For IPPA, it performs much better than other algorithms when encountering similar environments (after the first cycle). There is little variation in the performance of IPPA in cycles 2 to 4, so the predictive accuracy of the initial population is already high when IPPA encounters a certain environment for the second time. Both HPPCM and GM-DMOP are guided by recent solutions to generate initial populations of new environments. Although the value in the experiment is randomly changed, this kind of method can still play a certain optimization role. For dMOP1 and FDA2, the POS is fixed, so the prediction mechanisms of HPPCM and GM-DMOP are able to generate better initial populations. FDA4 and FDA5 are the more difficult tests. Compared to other tests, HPPCM and GM-DMOP perform poorly in these two tests, which is a challenge for them. It can be seen from the results in Table 5 that the performance metrics of the initial population of IPPA are significantly better than those of the other two algorithms. Since HPPCM and GM-DMOP have no memory mechanism, the performance metrics of their initial population are worse when the environment changes irregularly. As can be seen from Figures 4 and 5, the initial population of IPPA is very close to POF. The population of IPPA converges much faster than that of HPPCM and GM-DMOP.

5.4. Comparison with Algorithms with Memory Strategy on Modified Tests

This section compares and analyzes the experimental results of IPPA, MOEA/D-HMPS, and MOEA/D-MP. The experimental method is the same as part 5.4. Both MOEA/D-HMPS and MOEA/D-MP have memory strategies. They all use the center point of the historical population to guide the generation of the initial population.



(a) The initial populations at t = 60, 70.



(b) The final populations at t = 60, 70.

Figure 5. The initial and final populations at t = 60, 70 obtained by IPPA, HPPCM and GM-DMOP on FDA5 (n_τ = 10).

Table 6 show that the performance of IPPA is better other algorithms in the first period. Compared with other algorithms, the performance of MOEA/D-HMPS is worst on FDA2. The performance of MOEA/D-MP is only good on dMOP1 and FDA2, and its performance is the worst in other tests. The characteristics of the data in Tables 7 and 8 are generally consistent with those in Table 6. It should be stressed that the value of SP of MOEA/D-HMPS and MOEA/D-MP are much higher on FDA3-5 than IPPA.

Table 9 shows the performance metrics of the initial population of the three algorithms on dMOP2 ($n_\tau = 10$) and FDA5 ($n_\tau = 10$). The three performance metrics of IPPA are obviously better than the three performance metrics of the other two algorithms. On dMOP2, the three performance metrics of MOEA/D-HMPS are better than those of MOEA/D-MP. In FDA3-5, the SP values of MOEA/D-HMPS are much higher than those of IPPA. MOEA/D-HMPS relies on historical population center point prediction. In environments with discontinuous POF changes such as FDA3-5, the center points fail to cover the extreme value regions of the multi-objective space, resulting in an uneven distribution of the initial population (reflected by high SP values). In contrast, IPPA constructs environmental vectors through "clustering centers + boundary solutions", which can fully characterize both the distribution range and core features of the environment. Consequently, the initial population predicted by INN naturally covers all regions of the POF, leading to lower SP values.

When t = 60, the HVD of MOEA/D-HMPS is similar to that of MOEA/D-MP. When t = 70, the HVD of MOEA/D-HMPS is larger than that of MOEA/D-MP.

Table 6. Mean and SD values of IGD indicator obtained by three algorithms.

Problem	Algorithm	$n_\tau = 10$				$n_\tau = 20$	
		$1 \le t \le 20$ Mean(SD)	$21 \le t \le 40$ Mean(SD)	$41 \le t \le 60$ Mean(SD)	$61 \le t \le 80$ Mean(SD)	$1 \le t \le 40$ Mean(SD)	$41 \le t \le 80$ Mean(SD)
dMOP1	IPPA	**0.0047(0.0004)**	**0.0044(0.0000)**	**0.0043(0.0000)**	**0.0043(0.0000)**	0.0135(0.0125)	**0.0044(0.0000)**
	MOEA/D-HMPS	0.0704(0.0402)+	0.0061(0.0010)+	0.0044(0.0001)	**0.0043(0.0000)**	0.0672(0.0276)+	0.0052(0.0001)+
	MOEA/D-MP	0.0140(0.0068)+	0.0044(0.0000)	0.0044(0.0000)	0.0044(0.0000)	**0.0130(0.0075)**	**0.0044(0.0000)**
dMOP2	IPPA	**0.0053(0.0011)**	**0.0044(0.0000)**	**0.0044(0.0000)**	**0.0044(0.0000)**	**0.0057(0.0004)**	**0.0044(0.0000)**
	MOEA/D-HMPS	0.0208(0.0084)+	0.0051(0.0001)+	**0.0044(0.0001)**	**0.0044(0.0000)**	0.0200(0.0031)+	0.0048(0.0001)
	MOEA/D-MP	0.0975(0.0586)+	0.0735(0.0791)+	0.0527(0.0603)+	0.0507(0.0518)+	0.0238(0.0164)+	0.0105(0.0003)+
dMOP3	IPPA	**0.0054(0.0005)**	**0.0044(0.0000)**	**0.0044(0.0000)**	**0.0043(0.0000)**	**0.0059(0.0013)**	**0.0044(0.0000)**
	MOEA/D-HMPS	0.0154(0.0046)+	0.0051(0.0004)+	**0.0044(0.0001)**	**0.0043(0.0000)**	0.0100(0.0013)+	0.0046(0.0000)
	MOEA/D-MP	0.0109(0.0020)+	0.0136(0.0052)+	0.0323(0.0330)+	0.0108(0.0036)+	0.0130(0.0065)+	0.0298(0.0163)+
FDA1	IPPA	**0.0070(0.0020)**	**0.0044(0.0000)**	**0.0044(0.0000)**	**0.0044(0.0000)**	**0.0050(0.0003)**	**0.0043(0.0000)**
	MOEA/D-HMPS	0.0220(0.0192)	0.0049(0.0004)	**0.0044(0.0000)**	**0.0044(0.0000)**	0.0438(0.0418)+	0.0055(0.0013)+
	MOEA/D-MP	0.0110(0.0052)+	0.0066(0.0004)+	0.0066(0.0007)+	0.0063(0.0005)+	0.0247(0.0240)+	0.0113(0.0054)+
FDA2	IPPA	**0.0048(0.0004)**	**0.0045(0.0001)**	**0.0044(0.0000)**	**0.0044(0.0000)**	**0.0058(0.0016)**	**0.0044(0.0000)**
	MOEA/D-HMPS	0.0389(0.0023)+	0.0095(0.0007)+	0.0056(0.0002)+	0.0051(0.0001)+	0.0683(0.0114)+	0.0110(0.0019)+
	MOEA/D-MP	0.0117(0.0044)+	**0.0045(0.0001)**	**0.0044(0.0000)**	**0.0044(0.0000)**	0.0078(0.0015)+	**0.0044(0.0000)**
FDA3	IPPA	**0.0029(0.0002)**	**0.0028(0.0001)**	**0.0025(0.0000)**	**0.0028(0.0000)**	**0.0029(0.0001)**	**0.0026(0.0001)**
	MOEA/D-HMPS	0.1183(0.0687)+	0.0093(0.0039)+	0.0044(0.0016)+	0.0030(0.0001)	0.0099(0.0030)+	0.0030(0.0000)
	MOEA/D-MP	0.0345(0.0170)+	0.0147(0.0113)+	0.0427(0.0396)+	0.0103(0.0048)+	0.0049(0.0005)+	0.0039(0.0002)+
FDA4	IPPA	**0.0776(0.0091)**	**0.0634(0.0003)**	**0.0630(0.0008)**	**0.0630(0.0006)**	**0.0707(0.0013)**	**0.0627(0.0004)**
	MOEA/D-HMPS	0.1831(0.0516)+	0.0649(0.0010)	**0.0630(0.0004)**	**0.0630(0.0005)**	0.2238(0.0356)+	0.0669(0.0030)
	MOEA/D-MP	0.1792(0.0280)+	0.1715(0.0142)+	0.1620(0.0252)+	0.1607(0.0230)+	0.2047(0.0374)+	0.1883(0.0073)+
FDA5	IPPA	**0.1648(0.0084)**	**0.1334(0.0059)**	**0.1291(0.0030)**	**0.1224(0.0059)**	**0.1499(0.0034)**	**0.1265(0.0016)**
	MOEA/D-HMPS	0.2316(0.0074)+	0.1681(0.0078)+	0.1595(0.0025)+	0.1550(0.0123)+	0.2084(0.0102)+	0.1589(0.0060)+
	MOEA/D-MP	0.2954(0.0897)+	0.2715(0.0587)+	0.2049(0.0154)+	0.2110(0.0033)+	0.2419(0.0135)+	0.2312(0.0066)+

Table 7. Mean and SD values of HVD indicator obtained by three algorithms.

Problem	Algorithm	$n_\tau = 10$				$n_\tau = 20$	
		$1 \leq t \leq 20$ Mean(SD)	$21 \leq t \leq 40$ Mean(SD)	$41 \leq t \leq 60$ Mean(SD)	$61 \leq t \leq 80$ Mean(SD)	$1 \leq t \leq 40$ Mean(SD)	$41 \leq t \leq 80$ Mean(SD)
dMOP1	IPPA	**0.0723(0.0021)**	**0.0703(0.0019)**	0.0709(0.0016)	0.0705(0.0009)	**0.0730(0.0028)**	0.0705(0.0007)
	MOEA/D-HMPS	0.1083(0.0051)+	0.0724(0.0009)	0.0709(0.0010)	**0.0679(0.0016)**	0.1161(0.0078)+	0.0710(0.0002)
	MOEA/D-MP	0.0772(0.0054)	0.0714(0.0010)	**0.0692(0.0015)**	0.0722(0.0014)	0.0746(0.0019)	**0.0703(0.0008)**
dMOP2	IPPA	**0.0715(0.0015)**	**0.0706(0.0006)**	**0.0694(0.0003)**	0.0704(0.0003)	**0.0714(0.0002)**	**0.0705(0.0004)**
	MOEA/D-HMPS	0.0855(0.0012)+	0.0713(0.0006)	0.0702(0.0007)	**0.0669(0.0009)**	0.0846(0.0009)+	**0.0705(0.0003)**
	MOEA/D-MP	0.1211(0.0262)+	0.0952(0.0187)+	0.1048(0.0405)+	0.1025(0.0332)+	0.0805(0.0035)+	0.0789(0.0010)+
dMOP3	IPPA	**0.0858(0.0010)**	**0.0843(0.0003)**	**0.0842(0.0006)**	**0.0845(0.0001)**	**0.0854(0.0000)**	**0.0845(0.0006)**
	MOEA/D-HMPS	0.0936(0.0044)+	0.0852(0.0004)	0.0848(0.0007)	0.0846(0.0003)	0.0888(0.0010)	0.0848(0.0002)
	MOEA/D-MP	0.0890(0.0020)	0.0921(0.0041)+	0.0994(0.0163)+	0.0885(0.0012)	0.0900(0.0036)	0.1039(0.0121)+
FDA1	IPPA	**0.0877(0.0022)**	**0.0839(0.0004)**	**0.0843(0.0003)**	**0.0844(0.0004)**	**0.0850(0.0003)**	**0.0844(0.0002)**
	MOEA/D-HMPS	0.0982(0.0143)+	0.0850(0.0005)	0.0848(0.0002)	0.0847(0.0002)	0.1170(0.0350)+	0.0852(0.0006)
	MOEA/D-MP	0.0902(0.0043)	0.0867(0.0006)	0.0860(0.0006)	0.0866(0.0012)	0.1043(0.0237)+	0.0897(0.0044)
FDA2	IPPA	**0.0710(0.0012)**	0.0716(0.0002)	0.0706(0.0006)	**0.0703(0.0002)**	**0.0722(0.0017)**	0.0712(0.0002)
	MOEA/D-HMPS	0.1198(0.0042)+	0.0768(0.0016)	0.0715(0.0009)	0.0710(0.0020)	0.1409(0.0083)+	0.0817(0.0023)+
	MOEA/D-MP	0.0772(0.0029)	**0.0694(0.0006)**	**0.0703(0.0009)**	0.0710(0.0018)	0.0735(0.0016)	**0.0703(0.0001)**
FDA3	IPPA	**0.0748(0.0014)**	**0.0734(0.0007)**	**0.0716(0.0008)**	**0.0735(0.0010)**	**0.0747(0.0009)**	**0.0724(0.0004)**
	MOEA/D-HMPS	0.1577(0.0262)+	0.0895(0.0012)+	0.0795(0.0029)+	0.0757(0.0021)	0.0883(0.0045)+	0.0739(0.0002)
	MOEA/D-MP	0.1087(0.0160)+	0.0992(0.0139)+	0.1061(0.0198)+	0.0935(0.0108)+	0.0850(0.0015)+	0.0800(0.0005)+
FDA4	IPPA	**0.0238(0.0049)**	**0.0177(0.0003)**	0.0174(0.0004)	**0.0160(0.0005)**	**0.0217(0.0005)**	**0.0166(0.0002)**
	MOEA/D-HMPS	0.0566(0.0011)+	0.0193(0.0004)	**0.0172(0.0002)**	0.0174(0.0004)	0.0684(0.0038)+	0.0206(0.0020)+
	MOEA/D-MP	0.0641(0.0026)+	0.0668(0.0038)+	0.0639(0.0083)+	0.0646(0.0073)+	0.0746(0.0059)+	0.0736(0.0022)+
FDA5	IPPA	**0.0641(0.0017)**	**0.0557(0.0018)**	0.0532(0.0026)	0.0503(0.0019)	**0.0600(0.0005)**	0.0525(0.0022)
	MOEA/D-HMPS	0.0850(0.0023)+	0.0567(0.0023)	**0.0506(0.0013)**	**0.0486(0.0010)**	0.0754(0.0017)+	**0.0494(0.0011)**
	MOEA/D-MP	0.0857(0.0151)+	0.0902(0.0085)+	0.0781(0.0050)+	0.0791(0.0007)+	0.0881(0.0020)+	0.0855(0.0012)+

Table 8. Mean and SD values of SP indicator obtained by three algorithms.

Problem	Algorithm	$n_\tau = 10$				$n_\tau = 20$	
		$1 \leq t \leq 20$ Mean(SD)	$21 \leq t \leq 40$ Mean(SD)	$41 \leq t \leq 60$ Mean(SD)	$61 \leq t \leq 80$ Mean(SD)	$1 \leq t \leq 40$ Mean(SD)	$41 \leq t \leq 80$ Mean(SD)
dMOP1	IPPA	**0.0065(0.0001)**	**0.0064(0.0001)**	0.0062(0.0001)	0.0062(0.0000)	**0.0071(0.0007)**	**0.0063(0.0000)**
	MOEA/D-HMPS	0.0314(0.0019)+	0.0176(0.0023)+	**0.0060(0.0001)**	**0.0061(0.0001)**	0.1188(0.1224)+	0.0073(0.0001)+
	MOEA/D-MP	0.0106(0.0020)+	**0.0064(0.0000)**	0.0064(0.0002)	0.0064(0.0000)	0.0085(0.0005)+	0.0065(0.0002)
dMOP2	IPPA	**0.0069(0.0007)**	**0.0061(0.0001)**	**0.0063(0.0001)**	0.0064(0.0000)	**0.0067(0.0001)**	0.0061(0.0000)
	MOEA/D-HMPS	0.0173(0.0029)+	0.0063(0.0002)	0.0060(0.0002)	**0.0062(0.0002)**	0.0163(0.0007)+	**0.0059(0.0000)**
	MOEA/D-MP	0.0276(0.0076)+	0.0206(0.0076)+	0.0222(0.0164)+	0.0232(0.0127)+	0.0141(0.0019)+	0.0463(0.0488)+
dMOP3	IPPA	**0.0073(0.0012)**	0.0063(0.0001)	0.0063(0.0001)	0.0063(0.0001)	**0.0072(0.0013)**	0.0062(0.0001)
	MOEA/D-HMPS	0.0144(0.0026)	**0.0062(0.0004)**	**0.0060(0.0001)**	**0.0060(0.0001)**	0.0460(0.0483)+	**0.0059(0.0000)**
	MOEA/D-MP	0.0117(0.0014)+	0.0144(0.0033)+	0.0159(0.0075)+	0.0124(0.0032)+	0.0109(0.0017)+	0.0188(0.0076)+
FDA1	IPPA	**0.0087(0.0024)**	**0.0061(0.0001)**	0.0064(0.0000)	0.0064(0.0001)	**0.0064(0.0004)**	**0.0062(0.0000)**
	MOEA/D-HMPS	0.0165(0.0091)+	0.0063(0.0003)	**0.0061(0.0001)**	**0.0062(0.0001)**	0.0225(0.0131)+	0.0069(0.0015)
	MOEA/D-MP	0.0209(0.0025)+	0.0083(0.0005)+	0.0082(0.0008)+	0.0080(0.0007)+	0.0171(0.0114)+	0.0117(0.0033)+
FDA2	IPPA	**0.0067(0.0005)**	**0.0061(0.0000)**	**0.0059(0.0000)**	**0.0059(0.0000)**	**0.0069(0.0004)**	**0.0062(0.0000)**
	MOEA/D-HMPS	0.0245 (0.0020)+	0.0100(0.0002)+	0.0069(0.0006)+	0.0066(0.0002)+	0.0346(0.0044)+	0.0126(0.0013)+
	MOEA/D-MP	0.0098(0.0015)+	0.0067(0.0002)	0.0067(0.0001)+	0.0066(0.0001)+	0.0078(0.0003)+	0.0066(0.0000)
FDA3	IPPA	**0.0084(0.0005)**	**0.0070(0.0000)**	**0.0070(0.0003)**	**0.0068(0.0000)**	**0.0085(0.0008)**	**0.0067(0.0001)**
	MOEA/D-HMPS	0.0451(0.0049)+	0.0795(0.0832)+	0.0152(0.0043)+	0.0108(0.0023)+	0.0228(0.0031)+	0.0085(0.0001)+
	MOEA/D-MP	0.0309(0.0054)+	0.0268(0.0058)+	0.0497(0.0282)+	0.0248(0.0068)+	0.0174(0.0010)+	0.0162(0.0004)+
FDA4	IPPA	**0.3776(0.0321)**	0.3212(0.0042)	0.3234(0.0023)	0.3181(0.0048)	**0.3407(0.0045)**	**0.3189(0.0030)**
	MOEA/D-HMPS	0.4146(0.0204)+	**0.3171(0.0028)**	**0.3147(0.0026)**	**0.3136(0.0044)**	0.4590(0.0296)+	0.3203(0.0029)
	MOEA/D-MP	0.4799(0.0572)+	0.4271(0.0266)+	0.4338(0.0435)+	0.4210(0.0202)+	0.4818(0.0238)+	0.4844(0.0301)+
FDA5	IPPA	**0.6522(0.0186)**	**0.5668(0.0218)**	**0.5665(0.0090)**	**0.5638(0.0260)**	**0.6227(0.0069)**	**0.5533(0.0034)**
	MOEA/D-HMPS	0.6962(0.0263)	0.6651(0.0035)+	0.6415(0.0295)+	0.6317(0.0247)+	0.6816(0.0151)+	0.6274(0.0077)+
	MOEA/D-MP	0.7982(0.0403)+	0.7508(0.0751)+	0.7046(0.0250)+	0.6871(0.0200)+	0.7328(0.0467)+	0.6992(0.0262)+

Table 9. The value of three performance metrics and their SD obtained by three algorithms on dMOP2 (n_τ=10) and FDA5 (n_τ=10).

Algorithm	dMOP2 (n_τ = 10)				FDA5 (n_τ = 10)	
	t = 60 IGD(SD)	t = 65 IGD(SD)	t = 70 IGD(SD)	t = 75 IGD(SD)	t = 60 IGD(SD)	t = 70 IGD(SD)
IPPA	**0.0971(0.0786)**	**0.0476(0.0314)**	**0.0552(0.0638)**	**0.0633(0.0806)**	**0.4973(0.1747)**	**0.3249(0.1264)**
MOEA/D-HMPS	0.1498(0.0201)+	0.1892(0.0576)+	0.1082(0.0714)+	0.1367(0.0144)+	1.0118(0.6168)+	0.5329(0.2776)+
MOEA/D-MP	8.3964(1.5510)+	5.6061(5.3114)+	0.7994(0.6521)+	2.5452(1.2926)+	2.1578(0.9428)+	0.7820(0.2943)+
	HVD(SD)	HVD(SD)	HVD(SD)	HVD(SD)	HVD(SD)	HVD(SD)
IPPA	**0.0276(0.1192)**	**0.0463(0.0443)**	**0.0983(0.0608)**	**0.0025(0.0726)**	**0.0676(0.0413)**	**0.0765(0.0323)**
MOEA/D-HMPS	0.0505(0.0344)+	0.0532(0.1171)+	0.1309(0.0859)+	0.0579(0.0306)+	0.1324(0.1489)+	0.1463(0.2190)+
MOEA/D-MP	0.5793(0.2238)+	0.3757(0.1656)+	0.3893(0.1643)+	0.2772(0.0090)+	0.1330(0.0010)+	0.0905(0.0079)+
	SP(SD)	SP(SD)	SP(SD)	SP(SD)	SP(SD)	SP(SD)
IPPA	**0.0061(0.0004)**	**0.0118(0.0072)**	**0.0133(0.0092)**	**0.0100(0.0051)**	1.0257(0.2992)+	**0.7450(0.1390)**
MOEA/D-HMPS	0.8052(1.1285)+	0.4055(0.5610)+	0.0891(0.0108)+	0.4511(0.6258)+	**0.6743(0.1091)**	0.7502(0.3299)
MOEA/D-MP	3.7744(0.2839)+	2.3624(1.0112)+	2.4209(1.9888)+	4.1548(1.7122)+	2.9552(1.4949)+	1.2267(0.8315)+

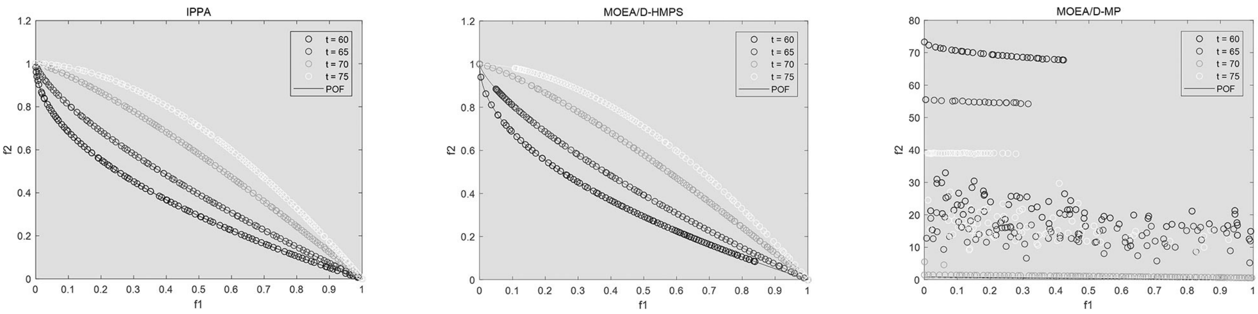
Figure 6 shows the initial and final populations of IPPA, MOEA/D-HMPS and MOEA/D-MP at t = 60, 65, 70, and 75 on dMOP2 (n_τ = 10). For the initial population, only IPPA is close to POF. MOEA/D-HMPS is closer to the POF than the initial population of MOEA/D-MP. The final population of all three algorithms is close to the POF.

Figure 7 shows the initial and final populations of IPPA, MOEA/D-HMPS and MOEA/D-MP at t=60, 70 on FDA5 (n_τ=10). Both the initial and final populations of IPPA are closest to POF. For the initial population, both MOEA/D-HMPS and MOEA/D-MP are far from the POF. For the final population, both IPPA and MOEA/D-HMPS are close to POF, and MOEA/D-MP being the worst.

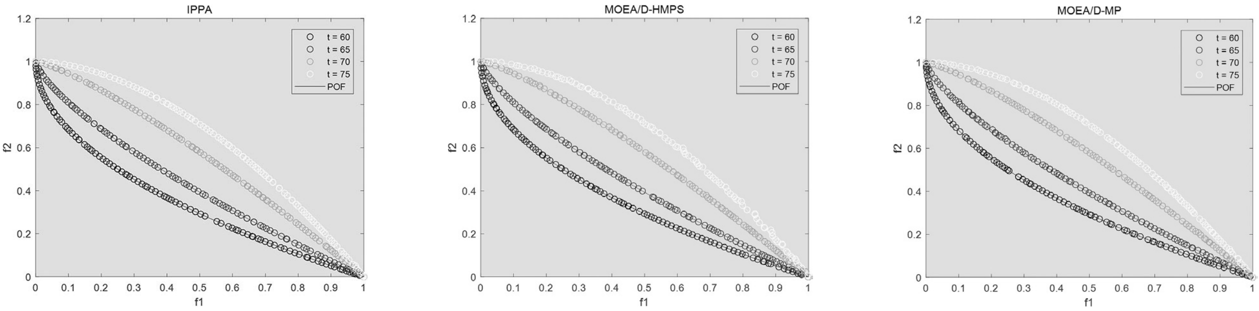
IPPA, MOEA/D-HMPS, and MOEA/D-MP all have memory strategies, so they perform better after the second cycle than the first. MOEA/D-HMPS and MOEA/D-MP can also use the population information before the environmental change to guide the generation of the initial population of the new environment. This forecasting mechanism sometimes works, but they perform worse than IPPA in the first cycle. MOEA/D-MP performs well in the easier tests but performs worst in the harder tests (FDA4 and FDA5). It can be seen from the results in Table 9 that the performance metrics of the initial population of IPPA are significantly better than those of the other two algorithms, and it can also be seen from Figures 6 and 7 that the initial population of IPPA is very close to the POF. The population convergence speed of IPPA is much faster than that of MOEA/D-HMPS and MOEA/D-MP.

6. Conclusion

This paper proposes a new initial population prediction algorithm (IPPA), designed to address black-box DMOPs with drastic and irregularly environmental changes. The experimental results show that when the test is relatively simple, IPPA performs better than other algorithms or has little difference. When the tests are more complex (FDA4 and FDA5), IPPA significantly outperforms most of the other algorithms.

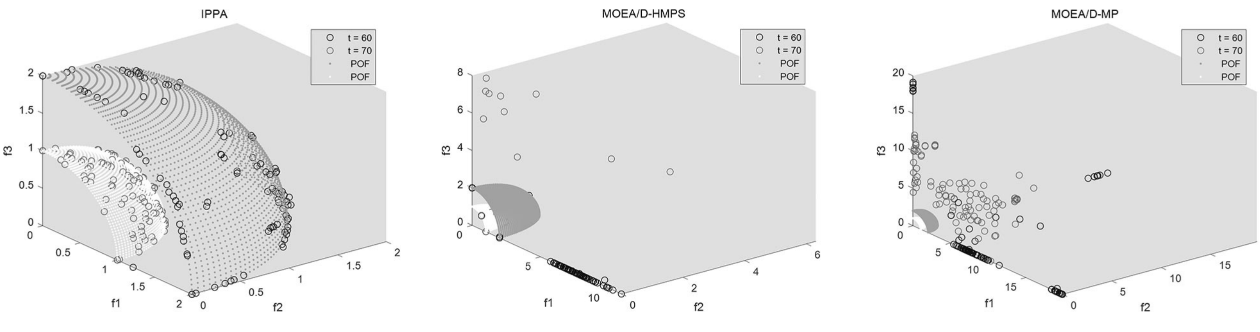


(a) The initial populations at t = 60, 65, 70, 75.

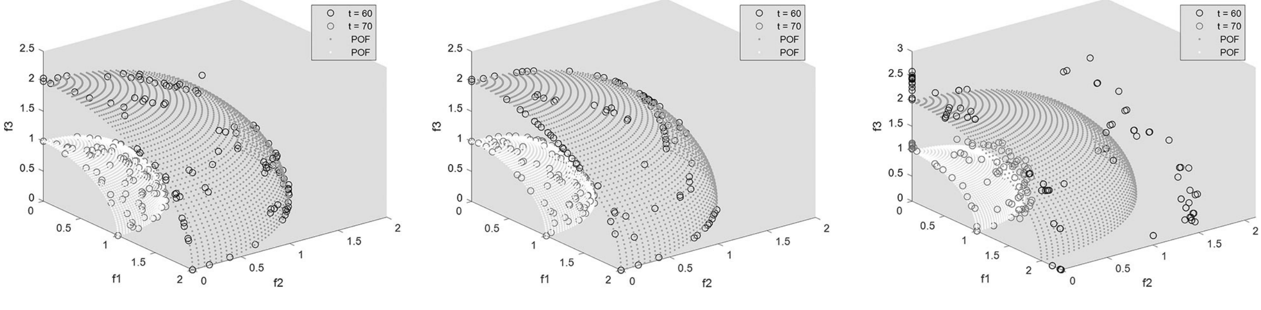


(b) The final populations at t = 60, 65, 70, 75.

Figure 6. The initial and final populations at t = 60, 65, 70, 75 obtained by IPPA, MOEA/D-HMPS and MOEA/D-MP on dMOP2 ($n_\tau = 10$).



(a) The initial populations at t = 60, 70.



(b) The final populations at t = 60, 70.

Figure 7. The initial and final populations at t = 60, 70 obtained by IPPA, MOEA/D-HMPS and MOEA/D-MP on FDA5($n_\tau = 10$).

Although IPPA has many advantages, there are still some limitations:

1. The elite maintenance strategy is limited in solving irregular dynamic multi-objective optimization problems, so there is still a lot of room for improvement in the performance of IPPA in the first cycle.
2. The performance of IPPA in the first cycle is much worse than in the second. The historical population information used when training the INN can affect the training performance. If the historical population information differs greatly from the distribution of the POS, the prediction performance of IPPA will be reduced.
3. For different practical problems, the optimal value of N_E varies, and it is difficult to determine this optimal value.

Declaration of Competing Interests

The authors declare no conflict of interest.

Funding

This work was supported by the National Natural Science Foundation of China (62203402); the National Natural Science Foundation of China (62103378); Science and Technology Project of Henan Province (232102321034); and the Science and technology project of Henan Province(202102310284); and the Innovation incubation program (2020ZCKJ211).

Data Availability

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

References

[1] L. Wei *et al.*, "A multi-objective Migrating Birds Optimization Algorithm Based on Game Theory for Dynamic Flexible Job Shop Scheduling Problem", *Expert Systems with Applications*, vol. 227, p. 120268, 2023. http://dx.doi.org/10.1016/j.eswa.2023.120268

[2] Y. Li *et al.*, "Process Scheduling for Prefabricated Construction Based on Multi-objective Optimization Algorithm", *Automation in Construction*, vol. 168, p. 105809, 2024, http://dx.doi.org/10.1016/j.autcon.2024.105809

[3] Y. Dong *et al.*, "Multi-Objective Optimization of Energy-Efficient Multi-Stage, Multi-Level Assembly Job Shop Scheduling", *Applied Sciences*, vol. 14, no. 19, p. 8712, 2024. http://dx.doi.org/10.3390/app14198712

[4] C. Qi, "Multi-Objective Optimization-Based Algorithm for Selecting the Optimal Path of Rural Multi-Temperature Zone Cold Chain Dynamic Logistics Intermodal Transportation", *Int J Comput Intell Syst*, vol. 17, no. 1, p. 224, 2024. http://dx.doi.org/10.1007/s44196-024-00616-3

[5] X. Zong *et al.*, "Multi-Objective Unmanned Vehicle Path Planning Based on Whale Optimization Algorithm", *Int. J. Mod. Phys. C*, p. 2550005, 2024. http://dx.doi.org/10.1142/S0129183125500056

[6] H. Yazdani *et al.*, "Techno-Economic and Environmental Design of Hybrid Energy Systems Using Multi-Objective Optimization and Multi-Criteria Decision Making Methods", *Energy Conversion and Management*, vol. 282, p. 116873, 2023. http://dx.doi.org/10.1016/j.enconman.2023.116873

[7] S. Chen *et al.*, "Multi-Objective Optimization Design for a Double-Direction Liquid Heating System-Based Cell-to-Chassis Battery Module", *International Journal of Heat and Mass Transfer*, vol. 183, p. 122184, 2022. http://dx.doi.org/10.1016/j.ijheatmasstransfer.2021.122184

[8] "Optimization and Efficiency Improvement of Robot-Based Industrial Production Process", *IJNDES*, vol. 8, no. 2, 2024. http://dx.doi.org/10.25236/IJNDES.2024.080214

[9] H. Song *et al.*, "Multi-Objective Battery Energy Storage Optimization for Virtual Power Plant Applications", *Applied Energy*, vol. 352, p. 121860, 2023. http://dx.doi.org/10.1016/j.apenergy.2023.121860

[10] J. Zou *et al.*, "A Prediction Strategy Based on Center Points and Knee Points for Evolutionary Dynamic Multi-Objective Optimization", *Applied Soft Computing*, vol. 61, pp. 806–818, 2017. http://dx.doi.org/10.1016/j.asoc.2017.08.004

[11] S. Jiang and S. Yang, "A Steady-State and Generational Evolutionary Algorithm for Dynamic Multiobjective Optimization", *IEEE Trans. Evol. Computat.*, vol. 21, no. 1, pp. 65–82, 2017. http://dx.doi.org/10.1109/TEVC.2016.2574621

[12] K. Wang *et al.*, "Multimodal Optimization via Dynamically Hybrid Niching Differential Evo-

lution", *Knowledge-Based Systems*, vol. 238, p. 107972, 2022.
http://dx.doi.org/10.1016/j.knosys.2021.107972

[13] Z. Hu *et al.*, "A Niching Backtracking Search Algorithm with Adaptive Local Search for Multimodal Multiobjective Optimization", *Swarm and Evolutionary Computation*, vol. 69, p. 101031, 2022.
http://dx.doi.org/10.1016/j.swevo.2022.101031

[14] X.-F. Liu *et al.*, "Cooperative Particle Swarm Optimization with Reference-Point-Based Prediction Strategy for Dynamic Multiobjective Optimization", *Applied Soft Computing*, vol. 87, p. 105988, 2020.
http://dx.doi.org/10.1016/j.asoc.2019.105988

[15] X. Sun *et al.*, "Enhanced NSGA-II with Evolving Directions Prediction for Interval Multi-Objective Optimization", *Swarm and Evolutionary Computation*, vol. 49, pp. 124–133, 2019.
http://dx.doi.org/10.1016/j.swevo.2019.05.009

[16] N. Kouka *et al.*, "A Change Severity Degree-Based Dynamic Multi-Objective Optimization Algorithm with Adaptive Response Strategy", *Information Sciences*, vol. 677, p. 120794, 2024.
http://dx.doi.org/10.1016/j.ins.2024.120794

[17] H. Xie *et al.*, "A Decision Variable Classification-Based Cooperative Coevolutionary Algorithm for Dynamic Multiobjective Optimization", *Information Sciences*, vol. 560, pp. 307–330, 2021.
http://dx.doi.org/10.1016/j.ins.2021.01.021

[18] C. Wang *et al.*, "A Grey Prediction-Based Evolutionary Algorithm for Dynamic Multiobjective Optimization", *Swarm and Evolutionary Computation*, vol. 56, p. 100695, 2020.
http://dx.doi.org/10.1016/j.swevo.2020.100695

[19] L. Cao *et al.*, "Evolutionary Dynamic Multiobjective Optimization Assisted by a Support Vector Regression Predictor", *IEEE Trans. Evol. Computat.*, vol. 24, no. 2, pp. 305–319, 2020.
http://dx.doi.org/10.1109/TEVC.2019.2925722

[20] S. Sahmoud and H. R. Topcuoglu, "A Memory-Based NSGA-II Algorithm for Dynamic Multi-Objective Optimization Problems", *Lecture Notes in Computer Science*, vol. 9598, pp. 296–310, 2016.
http://dx.doi.org/10.1007/978-3-319-31153-1_20

[21] J. Zheng *et al.*, "An Improved Memory Prediction Strategy for Dynamic Multiobjective Optimization", in *Proceedings of the 2020 5th International Conference on Computational Intelligence and Applications (ICCIA)*, 2020, pp. 166–171.
http://dx.doi.org/10.1109/ICCIA49625.2020.00039

[22] W. T. Koo *et al.*, "A Predictive Gradient Strategy for Multiobjective Evolutionary Algorithms in a Fast Changing Environment", *Memetic Comp.*, vol. 2, no. 2, pp. 87–110, 2010.
http://dx.doi.org/10.1007/s12293-009-0026-7

[23] S. Li *et al.*, "A Modular Neural Network-Based Population Prediction Strategy for Evolutionary Dynamic Multi-Objective Optimization", *Swarm and Evolutionary Computation*, vol. 62, p. 100829, 2021.
http://dx.doi.org/10.1016/j.swevo.2020.100829

[24] H. Zille *et al.*, "Dynamic Distance Minimization Problems for Dynamic Multi-Objective Optimization", in *Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC), IEEE*, 2017, pp. 952–959.
http://dx.doi.org/10.1109/CEC.2017.7969411

[25] C. Liu and Y. Wang, "New Evolutionary Algorithm for Dynamic Multiobjective Optimization Problems", in *Advances in Natural Computation*, vol. 4221, L. Jiao, L. Wang, X. Gao, J. Liu, and F. Wu, Eds., in Lecture Notes in Computer Science, vol. 4221., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 889–892.
http://dx.doi.org/10.1007/11881070_117

[26] S. U. Rehman *et al.*, "DBSCAN: Past, Present and Future", in *Proceedings of the 5th International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014), IEEE*, 2014, pp. 232–238.
http://dx.doi.org/10.1109/ICADIWT.2014.6814687

[27] S. Chen *et al.*, "Research on Wind Turbine Fault Prediction Based on Improved Variable Learning Rate Neural Network", *Heilongjiang Electric Power*, vol. 43, no. 2, pp. 99–102, 2021.
http://dx.doi.org/10.13625/j.cnki.hljep.2021.02.002

[28] G.-Q. Zeng *et al.*, "An Improved Multi-Objective Population-Based Extremal Optimization Algorithm with Polynomial Mutation", *Information Sciences*, vol. 330, pp. 49–73, 2016.
http://dx.doi.org/10.1016/j.ins.2015.10.010

[29] C.-K. Goh and K. C. Tan, "A Competitive-Cooperative Coevolutionary Paradigm for Dynamic Multiobjective Optimization", *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 1, pp. 103–127, 2009.
http://dx.doi.org/10.1109/TEVC.2008.920671

[30] M. Farina *et al.*, "Dynamic Multiobjective Optimization Problems: Test Cases, Approximations, and Applications", *IEEE Trans. Evol. Computat.*, vol. 8, no. 5, pp. 425–442, 2004.
http://dx.doi.org/10.1109/TEVC.2004.831456

[31] F. Wang *et al.*, "A New Prediction Strategy for Dynamic Multi-Objective Optimization Using Gaussian Mixture Model", *Information Sciences*, vol. 580, pp. 331–351, 2021.
http://dx.doi.org/10.1016/j.ins.2021.08.065

[32] Y. Chen *et al.*, "Combining a Hybrid Prediction Strategy and a Mutation Strategy for Dynamic Multiobjective Optimization", *Swarm and Evolutionary Computation*, vol. 70, p. 101041, 2022.
http://dx.doi.org/10.1016/j.swevo.2022.101041

[33] Z. Liang *et al.*, "Hybrid of Memory and Prediction Strategies for Dynamic Multiobjective Optimization", *Information Sciences*, vol. 485, pp. 200–218, 2019.
http://dx.doi.org/10.1016/j.ins.2019.01.066

Received: March 2025
Revised: February 2026
Accepted: February 2026

Contact addresses:
Sanyi Li
School of Electrical and Information Engineering
Zhengzhou University of Light Industry
Zhengzhou
China
e-mail: wslisanyi@163.com

Shuang Liu
College of Mechanical and Electrical Engineering
Linzhou College of Architectural Technology
Anyang Henan
China
e-mail: 582787358@qq.com

Wenjie Hou
School of Electrical and Information Engineering
Zhengzhou University of Light Industry
Zhengzhou
China
e-mail: hwenjie2018@163.com

Yuanyuan Zhang
College of Mechanical and Marine Engineering
Beibu Gulf University
Qinzhou Guangxi
China
e-mail: zhangyuan8706@163.com

Sanyi Li received his PhD, from the Beijing University of Technology in 2018 and is currently working as an Associate Professor at the Zhengzhou University of Light Industry. His main research interests include multi-objective optimization, process automation in process industry, neural network modeling and fault diagnosis of power system equipment.

Shuang Liu is obtained his Master's degree in 2024 from the Zhengzhou University of Light Industry and is currently working as a Lecturer at the Linzhou College of Architectural Technology. His main research focus is on dynamic multi-objective optimization.

Wenjie Hou is currently a Master's degree student at the Zhengzhou University of Light Industry. His main research interest is multi-objective optimization.

Yuanyuan Zhang received her M.S. degree from Guangxi University, Nanning, China, in 2013, where she is presently working towards her PhD degree in the School of Electrical Engineering. She is currently working as an Associate Professor at the Beibu Gulf University. Her current research interests include inverter control and modeling, and intelligent detection and control.