Dynamic multi-objective optimization algorithm based on weighted differential prediction model

Yumeng Zhao, Xianpeng Wang, *Member, IEEE*,
Zhiming Dong, Yao Wang, Hangyu Lou, Tenghui Hu, Kai Fu

***Abstract*—In this paper, a new algorithm for solving dynamic multi-objective optimization problems(DMOPs) is proposed. Most of the traditional dynamic multi-objective optimization algorithms will make predictions based on the overall average evolutionary direction of the population, which is hardly applicable to problems where the solution set and frontier do not vary with the environmental rules. In this paper, a dynamic multi-objective optimization algorithm based on weight difference prediction model is designed to solve such problems. The algorithm contains a weighted differential prediction strategy, and a differential model is built for each individual using the weights to predict the initial population after environmental changes. With this approach, each individual in the population can be made to respond quickly to environmental changes. We used three classical comparison algorithms to conduct experiments on a series of test problems. The experimental results show that the WD-MOEA/D algorithm can significantly improve the dynamic optimization performance and is effective in solving different types of dynamic problems.**

***Index Terms*—Dynamic multi-objective optimization, evolutionary algorithms, weighted difference strategy**

I. Introduction

MULTI-objective optimization problem refers to the optimization problem of multiple mutually exclusive objectives, and the multi-objective problem that changes with the environment or time is called dynamic multi-objective optimization problem [1]. There are many such problems in the real world, such as dynamic path planning [2], shop floor scheduling [3], etc. Dynamic multi-objective optimization problems are usually divided into three categories according to their changes, objective function change, constraint change or decision variable change [4]. Due to their inconstancy over time or environment, it is very difficult to solve DMOPs [5]. In recent years, many researchers have paid extensive attention and research [6].

There are many types of dynamic multi-objective problems. This paper only studies the problem of unconstrained objective function changes, defined as follows：

$$\begin{array}{c}\min F(x,t) = (f_1(x,t), f_2(x,t), \ldots, f_M(x,t))^T \\ s.t.\;\; x \in \Omega,\;\; t \in \Omega_t\end{array} \tag{1}$$

where $x = (x_1, x_2, \ldots, x_v)$ is a v-dimensional decision variable and $\Omega \subseteq R^v$ represents the decision space. t is discrete time and $\Omega_t \subseteq R$ is the time space. $F(x,t)$ is a objective vector consisting of m objectives that change with time, where R^m is the objective space.

In recent years, in order to solve such DMOPs, researchers have added some strategies to the static evolutionary algorithm to respond to changes in the environment [7]. These strategies should not only satisfy the diversity of the population, but also ensure the convergence of the population. In designing these strategies, we divided them into four categories [8]: 1) diversity-based methods 2) multi-population-based methods 3) memory-based methods and 4) prediction-based methods. These four types of methods will be introduced in detail in chapter II of the article. Through the experimental verification of the researchers, these four methods of designing prediction strategies have achieved good results. But most approaches treat the population as a population where all individuals move in the same direction when the environment changes. This makes some problems with irregular changes in individuals not well predicted. In order to solve this problem, this paper designs a weight difference strategy to make up for the different characteristics of different individuals.

This paper proposes a dynamic multi-objective optimization algorithm based on weighted differential prediction model(WD-DMOEA), which combines weight difference strategy with static weighted static algorithm. As in the evolutionary algorithm, each individual evolves along its own matching weight vector. Therefore, we construct a difference model for each individual according to the weight vector. This difference model is simple to calculate and does not increase the computational complexity due to the large population size. The main contributions of this paper are as follows:

This research was supported by the Fund for the National Natural Science Foundation of China (62073067), and the Fundamental Research Funds for the Central Universities (N2128001). (Corresponding: Lixin Tang)

Y. Zhao is with the Key Laboratory of Data Analytics and Optimization for Smart Industry (Northeastern University), Ministry of Education, Shenyang, 110819, China (e-mail: zhaoym@stumail.neu.edu.cn).

X. Wang is with the Liaoning Engineering Laboratory of operation Analytics and Optimization for Smart Industry, Liaoning Key Laboratory of Manufacturing System and Logistics, Shenyang, 110819, China (e-mail: wangxianpeng@ise.neu.edu.cn).

Z. Dong is with the Liaoning Engineering Laboratory of Data Analytics and Optimization for Smart Industry, Shenyang, 110819, China (e-mail: dongzm@stumail.neu.edu.cn).

Y. Wang is with the Liaoning Engineering Laboratory of Data Analytics and Optimization for Smart Industry, Shenyang, 110819, China (e-mail: yaowang@stumail.neu.edu.cn).

H. Lou is with the Liaoning Engineering Laboratory of Data Analytics and Optimization for Smart Industry, Shenyang, 110819, China (e-mail: louhangyu@stumail.neu.edu.cn).

T. Hu is with the Liaoning Engineering Laboratory of Data Analytics and Optimization for Smart Industry, Shenyang, 110819, China (e-mail: hu.tenghui@foxmail.com).

K. Fu is with the Liaoning Engineering Laboratory of Data Analytics and Optimization for Smart Industry, Shenyang, 110819, China (e-mail: 15546181953@163.com).

1) The difference-based prediction model can predict the location of individuals in the next time window more accurately.
2) Prediction according to weights separately allows each individual to respond adaptively to environmental changes.
3) In this paper, the effectiveness of the strategy is demonstrated through systematic experiments. This is very helpful in the field of dealing with DMOPs in the future.

The remainder of the paper is organized as follows: In Section II, we review the existing literature on DMOPs. Section III details WD-DMOEA. Section IV details the experimental procedure and analyzes the experimental results. Section V presents conclusions and future work.

II. Preliminaries and Related Work

A. Related Work

Compared with the static multi-objective optimization algorithm, the dynamic multi-objective optimization algorithm increases the environmental detection part. When environmental changes are detected, researchers often have the following four methods to predict the initial population in the new environment. These design methods are not completely independent and often interact with each other.

1) Diversity-based methods: Diversity-based methods focus on improving the diversity of the current convergent population [9]. This method appeared relatively early, and was mainly used to generate new solutions or increase mutation under certain specific rules. References [10] and [7] are two dynamic versions of the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) proposed by Deb and Karthik. NSGA-II-A improves diversity by randomly reinitializing some individuals, and NSGA-II-B replaces the population by randomly mutating some individuals. Diversity methods have also been refined as researchers deepen their research. Reference [11] uses a controllable mutation operator and a strategy of simulating isotropic magnetic ions to increase population diversity. Reference [12] proposed a regional co-evolutionary DMOAs.

2) Multi-population-based method: In order to take into account the diversity and convergence of the population, this method will generate multiple associated populations at the same time to maintain the characteristics of their respective populations. Reference [13] uses small groups to obtain a more comprehensive group through mutual competition-cooperation. Reference [14] designed the population of this relationship according to the competition-cooperation relationship in nature, and guided the algorithm evolution. This method is mostly used for multi-peak DMOPs [8].

3) Memory-based methods: Memory-based methods use implicit or explicit storage of the current solution and relevant information in the environment, and utilize this information in later stages [8]. In order to more accurately predict the moving position of the solution under multiple change types, multi-directional prediction is proposed in [15], an inheritance framework based on expert mixing is designed in [16], and a new information sharing strategy is proposed in [17]. Reference [18] reuses historical information to design a reference point-based forecasting strategy. Many methods of building predictive models use memory methods.

4) Prediction-based methods: When the behavior of a dynamic problem follows a certain trend, prediction-based methods utilize historical information and predict the location of the new optimal solution through a predictive model. This is the most used method in recent years. This approach typically stores information about the current solution or the current environment implicitly or explicitly. There are many ways to construct these prediction models, including models designed according to the moving trajectory of the decision space solution [18], and also using neural networks [19], machine learning [20], transfer learning [21], mathematical inverse Model [22] and other methods design a model that can predict the high-quality initial population at the next moment. After the researchers' verification, these prediction models can greatly improve the performance of the algorithm.

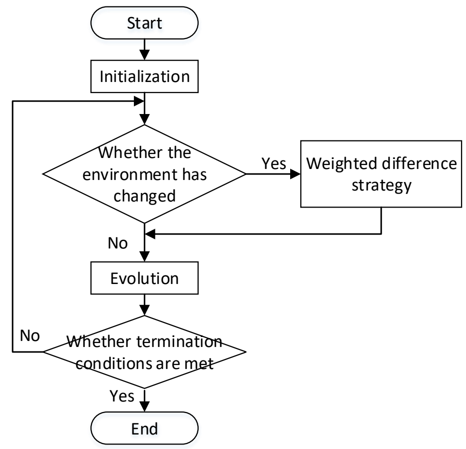

Fig. 1. Algorithm flow chart of WD-DMOEA

III. Proposed Algorithm

In this section, we propose a weighted difference model-based dynamic multi-objective optimization algorithm to address DMOPs. When the environment changes, we use the prediction strategy of the weighted difference model to predict a better initial population, thereby improving the convergence of the population. The flowchart of the proposed algorithm is shown in Figure 1.

A. First-order difference model

In the algorithm proposed in this paper, a first-order difference model is used to predict the position of individuals in the population. The model uses the difference between the previous two moments of the individual as the difference operator, and assumes that the same individual in each new environment has the same moving direction and step change. When a change in the environment is detected, the new location of the individual in the next time window is predicted

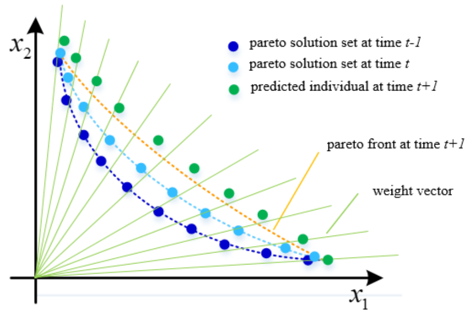

Fig. 2. Weighted Difference Model Prediction Schematic

based on the location of the individual in the previous time window. The individual predicted by the model and the individual in the previous time window, whichever is better, is the initial individual in the new environment.

We define x_t as the individuals in the Pareto solution set(POS) POS_t at the end of time window t. The difference operator is represented by $\overrightarrow{x_t - x_{t-1}}$. The individuals in the next time window predicted by the first-order difference model are:

$$x_{t+1} = x_t + \overrightarrow{x_t - x_{t-1}} \tag{2}$$

Since the dynamic problem cannot generate a difference operator when the time window 0 changes to the time window 1, the prediction model is used for prediction from the time window 3 onwards.

B. Prediction strategy based on weight difference

When the environment changes, POF and POS do not necessarily change in the same direction with the same step size. Therefore, we perform a first-order difference model prediction for each individual in the population to ensure that the prediction method responds to many changes in the population. In order to record the information of each individual changing with the environment, we choose MOEA/D as the baseline algorithm of the algorithm proposed in this paper. All individuals in the MOEA/D algorithm are arranged in order according to the weight vector. Each weight vector produces a sequence that records information about a particular individual in the population as a function of the environment. Through this sequence prediction, the individual position of the next time window is obtained as:

$$x^n_{t+1} = \mathrm{D}\left(x^n_t, x^n_{t-1}\right) \tag{3}$$

where x^n_{t+1} is the individual associated with the nth weight vector in the time window $t+1$. The function $\mathrm{D}(\cdot)$ is the constructed first-order difference model. The schematic diagram of the prediction principle is shown in Figure 2.

The algorithm of the weighted difference prediction strategy is shown in Algorithm 1. Firstly, according to the weight vector, the position of the individual is predicted through the first-order difference model, and the boundary judgment is performed on the calculation result to make it meet the condition of the decision variable range. Because in practical situations, most of the consecutive positions of individuals are similar [23]. Individual positions in the former environment may be closer to the optimal solution than the positions predicted by the difference model. Therefore, we keep the individual positions before the change, and select two positions in the fifth line of Algorithm 1. The individual position closer to the optimal solution is reserved as the final individual position solved by the weighted difference prediction strategy.

Algorithm 1: Weight difference strategy(WD)

Input: The solutions at time $t-1$, POS_{t-1}; the ith entry of x is denoted by $x^i_{t-1} \in POS_{t-1}$; the solutions at time $t-2$, POS_{t-2}; the population size, N;
Output: The solutions at time t, POS_t;
1 **for** $i = 1$ *to* N **do**
2 Calculate the value of x^i_{t*} using formula 2;
3 Boundary check x^i_{t*};
4 $x^i_t = Choose(x^i_{t*}, x^i_{t-1})$;
5 Return $POS_t = \{x^1_t, x^2_t, \ldots, x^N_t\}$;

C. Framework of the proposed algorithm

Algorithm 2 describes the overall framework of the algorithm proposed in this paper. First, the population is randomly initialized in the search space. Then use the static algorithm MOEA/D to find the optimal solution of the initial environment. When the environment changes, we use a weighted difference prediction strategy to predict the initial population of the new environment. Continue optimization with MOEA/D in the new environment. When the termination criterion is satisfied, the algorithm terminates.

Algorithm 2: Framework of WD-DMOEA

Input: The dynamic optimization, $F(x,t)$; the population size, N; the number of decision variables, V
Output: The solutions at time t, POS_t; the front at time t, POF_t;
Randomly initialize a population POP_0, $t = 0$;
$(POS_0, POS_0) =$ $\mathrm{MOEA}/D\left(POP_0,\ F(x,0),\ \ N,V\right)$;
while *the termination criterion is not satisfied* **do**
  if *change detected* **then**
    $t = t+1$;
    $POS_t = \mathrm{WD}\left(POS_{t-1}, POS_{t-2}, N\right)$;
  $(POF_t, POS_t) =$ $\mathrm{MOEA}/D\left(POS_t,\ F(x,t),\ \ N,V\right)$;
Return POS_t, POF_t;

IV. Experimental study

A. Benchmark problems and Performance matrixs

The algorithm proposed in this paper is tested on 14 benchmark problems. This set of test questions is the CEC 2018 DMO test question [24] that covers several classic dynamic multi-objective test questions [14], [25]–[27]. The CEC 2018 DMO test question is a relatively new dynamic test question. Among the 14 standard test functions, D1-D9 are dual-objective questions, and D10-D14 are three-objective questions. It covers a variety of problem features such as irregular, discontinuous Pareto front (POF), POS with time-dependent geometry, etc. The decision variables in the set of test questions we use in this paper are uniformly set to 10.

In this experiment, there are many indicators to evaluate the performance of the algorithm. According to the experimental needs, we finally choose MIGD [23] to measure the performance of the algorithm.

Mean Inverted Generational Distance (MIGD) is a commonly used metric to measure population performance in dynamic problems. The convergence and diversity of the solution are measured by calculating the difference between the true POF and the POF estimated by the algorithm. The smaller the value of MIGD, the better the performance of the algorithm. MIGD is calculated as follows:

$$MIGD\left(POF_t^*, POF_t\right) = \frac{\sum_{p \in POF_t^*} d\left(p, POF_t\right)}{|T| * |POF_t^*|} \tag{4}$$

where $POF_t^*$is a series of sampling points of the real POF at time window t, and POF_t is the value estimated by the algorithm at time window t. $d\left(p, POF_t\right)$ represents the minimum Euclidean distance from the point p in the real POF to the algorithm estimated value POF_t. $|POF_t^*|$ is the base of the true POF, and $|T|$ is the base of the environmental change.

B. Compared algorithms and Parameter settings

In this experiment, we use three algorithms including typical models for comparison, namely IGP-DMOEA [22] , ISVM-DMOEA [28] and PPS [23]. Some key parameters of the WD-DMOEA algorithm are as follows:

1) The population size is set to be 100 for bi-objective problems and 300 for tri-objective problems. The number of decision variables is set as 10 for all test problems.

2) The Techebycheff method is used as the decomposition method in MOEA/D, and the neighbor scale T is set to 20. Crossover probability, mutation probability and neighborhood selection probability are 0.5, 0.5 and 0.8, respectively.

3)We set up three groups of comparative experiments according to the severity and frequency of environmental changes. Environmental changes were made after 50 generations of operation in the initial environment. Each algorithm runs 20 times independently, and the average value is the most experimental result.

C. Experimental Study

The statistical results of the mean and standard deviation of MIGD obtained by WD-MOEA/D and the three comparison algorithms are given in Table I. We separate the mean and standard deviation of 20 runs of each algorithm into a table. For visual comparison, the best value for each instance is shown in bold. Second, we performed the Wilcoxon rank-sum test at the 0.05 significance level to indicate the significance between the different results. When the results of the compared algorithms were statistically significantly better than, inferior to, or substantially equivalent to the algorithms proposed in this paper, the results were marked with '+', '-', and '=', and were finalized at the bottom of the table, respectively. registration. Figure 3 depicts the evolution of the IGD values for DF1-DF14 with $n_t = 10$ and $\tau_t = 10$, with better independent run results.

It is evident from Table I that the algorithm proposed in this paper achieves the best MIGD on most of the problems. Poor performance on DF1, DF2, DF9 and DF13, but not much difference from the best results. Although the advantages of some problems are not obvious, they can still meet the diversity of populations in changing environments. The test of various problems shows that WD-DMOEA can have various problems and can quickly respond to various environmental changes.

DF3, DF4, DF6, DF7, DF11 and DF12 have obvious advantages, while DF5, DF9 and DF12 are greatly affected by the frequency of environmental changes. The PS of DF3 and DF4 is greatly affected by the differential factor, which just coincides with the WD strategy. The POS of DF5, DF6, DF7 and DF8 is greatly affected by trigonometric functions, and the WD strategy is still very applicable. The WD strategy has a better effect on problems with large differences in decision variables.

Figure 3 shows the relationship between the IGD value and the environment of different algorithms for all test problems when $n_t = 10, \tau_t = 10$. From the figure, we can observe that the WD-DMOEA algorithm is more stable than other algorithms, and is not easily disturbed by more historical information or the environment. The performance of all algorithms is related to the population convergence in a static process. When the IGD value of the initial environment is low, the overall IGD value of the algorithm will decrease accordingly. Therefore, the prediction effect of the algorithm will be better.

V. Conclusion

In this paper, a WD-DMOEA algorithm for solving DMOPs is proposed. The algorithm uses the weighted difference prediction strategy, and relies on the weights to construct a difference model for different individuals in turn to predict the initial population after environmental changes. Compared with the traditional prediction method, this prediction method is more suitable for the problem of irregular POS changes. We conduct experiments on 14 test problems with three classical contrast algorithms. Experimental results show that the algorithm proposed in this paper is suitable for many types of problems. The speed of response to environmental changes in two-dimensional problems and three-dimensional problems shows the effectiveness of the algorithm. However, it does not perform well in some overly complex problems and the change

TABLE I
MEAN AND STANDARD DEVIATION VALUES OF MIGD OBTAINED BY WD-DMOEA AND COMPARED ALGORITHMS

Prob.	(τ_t, n_t)	IGP-DMOEA	ISVM-DMOEA	PPS	WD-DMOEA
DF1	(5,10)	**2.7377e-02(8.78e-04)-**	6.8655e-02(1.21e-02)-	3.0385e-02(3.17e-02)-	1.0465e-01(2.72e-03)
	(10,10)	**9.1181e-03(1.17e-03)-**	2.5843e-02(8.99e-03)+	1.2824e-02(1.77e-02)-	2.0726e-02(6.48e-03)
	(20,10)	**4.0701e-03(3.64e-04)-**	6.1841e-03(7.49e-04)=	1.1514e-02(2.38e-03)+	6.0183e-03(4.06e-04)
DF2	(5,10)	**4.1148e-02(1.89e-03)-**	9.8484e-02(1.16e-02)+	1.4115e-01(4.32e-02)+	6.0891e-02(9.46e-03)
	(10,10)	**1.6645e-02(2.35e-03)-**	2.1821e-02(4.88e-03)=	2.8375e-02(1.82e-02)+	2.1281e-02(5.02e-03)
	(20,10)	**4.4246e-03(6.17e-04)=**	7.1084e-03(7.69e-04)+	6.8702e-03(3.35e-03)+	6.2443e-03(1.20e-03)
DF3	(5,10)	4.5083e-02(6.65e-03)+	2.2755e-01(4.14e-02)+	1.7625e-01(1.43e-02)+	**3.4866e-02(2.11e-03)**
	(10,10)	2.8327e-02(1.94e-03)+	1.4258e-01(2.20e-02)+	6.7399e-02(1.09e-02)+	**1.3240e-02(6.44e-03)**
	(20,10)	1.6410e-02(7.28e-04)+	1.8376e-02(6.23e-03)+	2.2409e-02(4.32e-03)+	**7.9130e-03(5.11e-04)**
DF4	(5,10)	9.5713e-02(5.22e-03)+	3.4184e-01(3.86e-02)+	8.5203e-02(3.58e-02)+	**7.2909e-02(3.14e-03)**
	(10,10)	6.4577e-02(1.63e-03)+	1.1326e-01(1.58e-02)+	6.7966e-02(3.30e-02)+	**5.8844e-02(2.44e-03)**
	(20,10)	5.2659e-02(1.44e-03)+	6.6060e-02(7.47e-03)+	5.6166e-02(1.27e-02)+	**4.9677e-02(7.95e-04)**
DF5	(5,10)	**2.4565e-02(1.89e-03)-**	8.5704e-02(4.17e-02)+	2.4448e-01(5.33e-02)+	4.0947e-02(1.20e-03)
	(10,10)	1.4406e-02(8.89e-04)+	2.6950e-02(1.59e-02)+	3.1123e-02(2.65e-02)+	**1.3126e-02(5.43e-03)**
	(20,10)	1.3294e-02(3.57e-04)+	1.4175e-02(6.23e-03)+	9.6019e-03(9.77e-03)+	**5.7684e-03(2.11e-04)**
DF6	(5,10)	2.3145e-00(1.19e-00)+	2.3642e-00(3.27e-01)-	2.9317e-00(6.93e-01)+	**2.2743e-00(4.43e-01)**
	(10,10)	2.0329e-00(8.75e-01)+	1.5228e-00(3.32e-01)-	2.7781e-00(4.62e-01)+	**1.8465e-00(3.07e-01)**
	(20,10)	6.4548e-01(3.41e-01)+	1.4523e-00(2.55e-01)-	2.5576e-00(3.13e-01)+	**2.0239e-01(2.25e-01)**
DF7	(5,10)	1.9686e-02(8.33e-03)+	3.8254e-02(2.80e-02)+	2.2526e-02(2.70e-02)+	**9.5049e-03(3.30e-03)**
	(10,10)	1.4509e-02(7.59e-03)+	1.7864e-02(2.83e-02)+	9.1089e-03(9.39e-03)+	**7.8015e-03(4.89e-04)**
	(20,10)	8.3253e-03(8.06e-03)+	6.6504e-03(2.55e-02)=	5.7535e-03(6.55e-03)+	**6.1422e-03(5.00e-03)**
DF8	(5,10)	2.3287e-02(1.81e-03)=	4.2413e-02(2.69e-02)+	2.7532e-03(7.80e-03)+	**2.1212e-03(7.03e-04)**
	(10,10)	1.7391e-03(9.83e-04)=	1.7508e-03(2.20e-02)=	1.7331e-03(1.17e-02)=	**1.5947e-03(4.34e-04)**
	(20,10)	1.3736e-03(5.03e-04)=	1.6855e-03(2.62e-02)+	1.5417e-03(1.35e-03)+	**1.3561e-03(4.37e-04)**
DF9	(5,10)	2.5820e-01(8.18e-03)+	3.3758e-01(6.84e-02)+	3.4367e-01(9.64e-02)+	**2.2456e-01(4.85e-02)**
	(10,10)	**1.3728e-01(6.33e-03)=**	1.6499e-01(6.57e-02)+	1.9020e-01(4.27e-02)+	1.4613e-01(3.84e-02)
	(20,10)	**9.0215e-02(6.82e-03)-**	1.2344e-01(4.05e-02)+	1.0595e-01(4.24e-02)=	1.0890e-01(3.82e-02)
DF10	(5,10)	**2.5108e-02(1.65e-02)-**	9.3973e-02(6.80e-02)+	4.6393e-02(1.33e-02)=	4.2506e-02(1.08e-02)
	(10,10)	**2.1166e-02(1.57e-02)-**	6.4788e-02(7.42e-02)+	5.1691e-02(6.21e-03)+	4.3188e-02(4.14e-03)
	(20,10)	**2.1843e-02(1.21e-02)-**	4.7325e-02(7.94e-02)+	4.4697e-02(7.12e-03)=	4.5008e-02(4.65e-03)
DF11	(5,10)	2.2156e-02(3.36e-03)+	4.1899e-02(2.98e-02)+	2.7945e-02(4.03e-03)+	**1.8585e-02(3.39e-03)**
	(10,10)	1.8715e-02(2.64e-03)+	2.2843e-02(3.45e-02)+	1.6683e-02(2.43e-03)+	**1.5291e-02(4.73e-03)**
	(20,10)	1.6241e-02(2.08e-03)+	1.7910e-02(1.60e-02)+	1.4075e-02(2.58e-03)=	**1.3608e-02(2.43e-03)**
DF12	(5,10)	1.2775e-02(3.38e-03)+	3.8971e-01(8.00e-03)+	2.3995e-02(7.16e-03)+	**1.1101e-02(3.66e-03)**
	(10,10)	1.1711e-02(4.19e-03)+	2.4517e-01(4.79e-03)+	1.4303e-02(6.39e-03)+	**1.0615e-02(4.19e-03)**
	(20,10)	1.0908e-02(4.42e-03)+	2.2268e-01(5.69e-03)+	1.3436e-02(2.87e-03)+	**9.7077e-03(2.48e-03)**
DF13	(5,10)	**4.0351e-02(1.35e-03)-**	7.7061e-02(3.37e-03)=	7.9339e-02(8.73e-03)+	7.6630e-02(9.84e-04)
	(10,10)	**2.0147e-02(1.05e-03)-**	3.7825e-02(1.05e-03)-	4.7796e-02(3.04e-03)+	4.2125e-02(5.07e-03)
	(20,10)	**1.3106e-02(9.14e-04)-**	1.5935e-02(9.87e-04)-	3.6576e-02(5.48e-03)+	3.2448e-02(9.64e-04)
DF14	(5,10)	9.6582e-02(1.39e-03)=	1.7065e-01(1.43e-02)+	2.3048e-01(1.45e-02)+	**9.5672e-02(1.21e-03)**
	(10,10)	8.9389e-02(1.86e-03)+	1.5720e-01(6.68e-03)+	1.9631e-01(1.93e-03)+	**7.6864e-02(2.86e-03)**
	(20,10)	4.0538e-02(1.93e-03)+	1.4647e-01(1.38e-03)+	9.5876e-02(1.14e-03)+	**3.7766e-02(8.94e-04)**
best/all		15/42	0/42	0/42	27/42
+/=/-		23/6/13	30/5/6	35/4/2	-/-/-

of decision quantity is not applicable to the WD strategy. This will be our future research direction.

REFERENCES

[1] Shouyong, Jiang, Shengxiang, and Yang, "A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 1, pp. 65–82, 2017.

[2] M. Mavrovouniotis, F. M. Muller, and S. Yang, "Ant colony optimization with local search for dynamic traveling salesman problems," *IEEE Transactions on Cybernetics*, vol. 47, no. 7, pp. 1743–1756, 2017.

[3] S. Nguyen, M. Zhang, M. Johnston, and K. C. Tan, "Automatic programming via iterated local search for dynamic job shop scheduling," *IEEE Transactions on Cybernetics*, vol. 45, no. 1, pp. 1–14, 2015.

[4] S. B. Gee, K. C. Tan, and H. A. Abbass, "A benchmark test suite for dynamic evolutionary multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 47, no. 2, pp. 461–472, 2017.

[5] C. Cruz, J. R. Gonz á lez, and D. A. Pelta, "Optimization in dynamic environments: a survey on problems, methods and measures," *Soft Computing*, vol. 15, no. 7, pp. 1427–1448, 2011.

[6] K. Deb, U. N, and K. Sindhya, "Dynamic multi-objective optimization and decision-making using modified nsga-ii: A case study on hydro-thermal power scheduling," 05 2007, pp. 803–817.

[7] H. Wang, D. Wang, and S. Yang, "A memetic algorithm with adaptive hill climbing strategy for dynamic optimization problems," *Soft Computing*, vol. 13, no. 8-9, pp. 763–780, 2009.

[8] J. Li, T. Sun, Q. Lin, M. Jiang, and K. C. Tan, "Reducing negative transfer learning via clustering for dynamic multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2022.

[9] W. T. Koo, C. K. Goh, and K. C. Tan, "A predictive gradient strategy for multiobjective evolutionary algorithms in a fast changing environment," *Memetic Computing*, vol. 2, no. 2, pp. 87–110, 2010.

[10] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[11] K. Zhang, C. Shen, X. Liu, and G. G. Yen, "Multiobjective evolution strategy for dynamic multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 5, pp. 974–988, 2020.

[12] X. Ma, J. Yang, H. Sun, Z. Hu, and L. Wei, "Multiregional co-evolutionary algorithm for dynamic multiobjective optimization," *Information Sciences*, vol. 545, pp. 1–24, 2021.

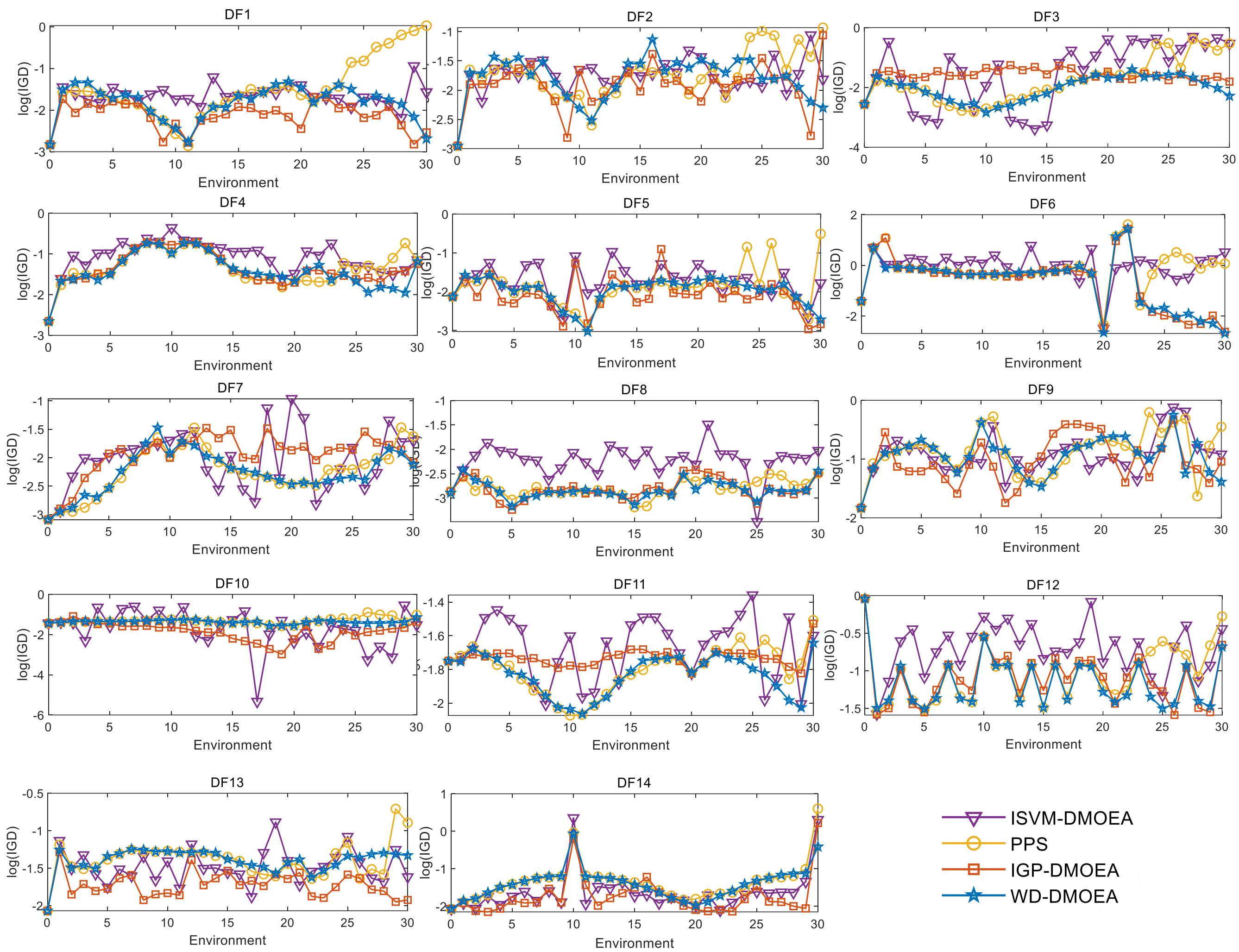

Fig. 3. The relationship between IGD value and environmental change

[13] H. Xie, J. Zou, S. Yang, J. Zheng, J. Ou, and Y. Hu, "A decision variable classification-based cooperative coevolutionary algorithm for dynamic multiobjective optimization," *Information Sciences*, vol. 560, pp. 307–330, 2021.

[14] 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.

[15] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on evolutionary computation*, vol. 11, no. 6, pp. 712–731, 2007.

[16] R. Rambabu, P. Vadakkepat, K. C. Tan, and M. Jiang, "A mixture-of-experts prediction framework for evolutionary dynamic multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 50, no. 12, pp. 5099–5112, 2020.

[17] A. Ahrari, S. Elsayed, R. Sarker, D. Essam, and C. A. C. Coello, "Weighted pointwise prediction method for dynamic multiobjective optimization," *Information Sciences*, vol. 546, pp. 349–367, 2021.

[18] X. F. Liu, Y. R. Zhou, and X. Yu, "Cooperative particle swarm optimization with reference-point-based prediction strategy for dynamic multiobjective optimization," *Applied Soft Computing*, vol. 87, p. 105988, 2020.

[19] R. Zhang and J. Tao, "Data-driven modeling using improved multi-objective optimization based neural network for coke furnace system," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 4, pp. 3147–3155, 2017.

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

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

[22] H. Zhang, J. Ding, M. Jiang, K. C. Tan, and T. Chai, "Inverse gaussian process modeling for evolutionary dynamic multiobjective optimization," *IEEE Transactions on Cybernetics*, pp. 1–14, 2021.

[23] A. Zhou, Y. Jin, and Q. Zhang, "A population prediction strategy for evolutionary dynamic multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 44, no. 1, pp. 40–53, 2014.

[24] S. Jiang, S. Yang, X. Yao, K. Tan, M. Kaiser, and N. Krasnogor, "Benchmark problems for cec2018 competition on dynamic multiobjective optimisation: Tech. rep." 2017.

[25] M. Farina, K. Deb, and P. Amato, "Dynamic multiobjective optimization problems: test cases, approximations, and applications," *IEEE Transactions on evolutionary computation*, vol. 8, no. 5, pp. 425–442, 2004.

[26] S. Jiang and S. Yang, "Evolutionary dynamic multiobjective optimization: Benchmarks and algorithm comparisons," *IEEE Transactions on Cybernetics*, vol. 47, no. 1, pp. 198–211, 2017.

[27] S. Jiang, M. Kaiser, S. Yang, S. Kollias, and N. Krasnogor, "A scalable test suite for continuous dynamic multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 50, no. 6, pp. 2814–2826, 2020.

[28] D. Xu, M. Jiang, W. Hu, S. Li, R. Pan, and G. G. Yen, "An online prediction approach based on incremental support vector machine for dynamic multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2021.