



Research article

A dynamic multi-objective evolutionary algorithm using center and multi-direction prediction strategies

Hongtao Gao¹, Hecheng Li^{2,*} and Yu Shen¹

¹ School of Computer Science and Technology, Qinghai Normal University, Xining 810008, China

² School of Mathematics and Statistics, Qinghai Normal University, Xining 810008, China

***Correspondence:** Email: lihecheng@qhnu.edu.cn.

Abstract: Dynamic multi-objective optimization problems have been popular because of its extensive application. The difficulty of solving the problem focuses on the moving PS as well as PF dynamically. A large number of efficient strategies have been put forward to deal with such problems by speeding up convergence and keeping diversity. Prediction strategy is a common method which is widely used in dynamic optimization environment. However, how to increase the efficiency of prediction is always a key but difficult issue. In this paper, a new prediction model is designed by using the rank sums of individuals, and the position difference of individuals in the previous two adjacent environments is defined to identify the present change type. The proposed prediction strategy depends on environment change types. In order to show the effectiveness of the proposed algorithm, the comparison is carried out with five state-of-the-art approaches on 20 benchmark instances of dynamic multi-objective problems. The experimental results indicate the proposed algorithm can get good convergence and distribution in dynamic environments.

Keywords: dynamic multi-objective optimization; evolutionary algorithm; multi-direction; prediction; Pareto optimal solutions

1. Introduction

A large number of multi-objective optimizations which change over time can be found in scientific and engineering fields, which are always known as dynamic multi-objective optimization problems (DMOPs) [1]. When the state changes over time, some multi-objective optimization models

in path planning [2], resource distribution [3], energy scheduling [4,5], network layout [6], and electrode-magnetic micro-mirrors become dynamic ones [7]. Therefore, it is necessary to study the optimal solutions to DMOPs at each moment as well as convergence. More and more researchers devote themselves to solving such problems in recent years. In any current environment of dynamic optimization, the Pareto optimal solutions (PS) and fronts (PF) need to be found in time before the next environment is detected, which brings great challenges for optimization approaches to deal with this kind of problem.

The types of DMOPs can be found in [8,9]. A general mode of DMOPs can be formulated as follows:

$$\begin{aligned} \min f(x, t) &= (f_1(x, t), f_2(x, t), \dots, f_m(x, t)) \\ \text{subject to } x &\in \Omega \end{aligned} \quad (1)$$

where m is the scale of objectives. The objective $f(x, t)$ is a function on the decision-making vector, which can change over time or environment t taken from the set $\{0, 1, 2, \dots, T\}$. T is the total number of environment changes, and the objective space is denoted by R^m . Decision-making vector $x = (x_1(t), x_2(t), \dots, x_n(t))$ is n -dimensional in space Ω . Both the decision variable $x_i(t)$, $i = 1, 2, \dots, n$, and objective function $f_k(x, t)$, $k = 1, 2, \dots, m$, depend on time variable t and dynamically change, once the new environment is detected.

DMOPs consist of static multi-objective optimization problems (MOPs) and dynamic reaction mechanisms and are composed of a series of multi-objective optimization procedures located in different environments. The objectives in a MOP are conflicting with each other, so it is essential to explore the trade-off solutions set of all objectives. Different from MOPs, in a DMOP, the objective functions as well as variables maybe change as the time varies from one environment to another. It follows that solving DMOPs is to deal with a series of MOPs generated at different times and obtain the Pareto optimal solutions for each MOP. One optimization procedure is always followed by another and is expected to provide a high-quality Pareto optimal set for each of these static MOPs before the time t changes. As a result, it has become a popular method to divide DMOPs into a sequence of MOPs.

In a DMOP, it is always assumed that the optimization problems at adjacent moments may have certain dependence. Hence, the optimal solutions in the previous moments are always reused to generate the initial population at new environment. In order to make full use of the efficient information of better individuals at past moments, a number of prediction models have been developed. These prediction methods increase the exploitation capability of the population for new solutions in some degree, and by which some classical multi-objective evolutionary algorithms are adopted to deal with DMOPs.

Sahmoud and Topcuoglu [10] presented a method for judging change types by calculating the difference of non-dominated solutions in two adjacent environments, and some other similar approaches have also been proposed in the past years [11]. These schemes are feasible and efficient when the change of solution sets is regular and stable. However, when the states change sharply, the optimization solutions may deviate significantly from the prediction area. Most of the existing approaches have limitations to deal with this kind DMOPs. This paper intends to develop a strategy combining center and multi-direction that can capture the complex environmental situations.

In our study, we consider the change type of environment with a new method. When the new environment happens, according to the characteristic of the two nearest historical moments, the change

type and corresponding different predicting tactics are proposed. The main contributions of this paper are listed as follows:

1) A new method of evaluating individuals is adopted. Rank values of each individual for all objective functions are recorded. Then, solutions are distinguished by their corresponding rank sums, which are used to guide the processes of optimization and dynamic response of the algorithm.

2) On the basis of statistical information of rank sums for all individuals in two adjacent environments, translational change and non-translational change are determined. Different prediction strategies relying on different environment change types are adaptively adopted to respond the new environments.

3) Experiments have been conducted to analyze the performance of our proposed algorithm MOEA/D/CMDS compared with others under different environmental change frequencies and severities.

Next, the related work and preliminaries about DMOPs are introduced. The proposed algorithm is described in Section 4. The test instances, compared algorithms, performance metrics, and parameter settings are presented in Section 5. A series of experiment results and analysis are provided in Section 6. The conclusion and further research are expressed in Section 7.

2. Related research

In recent years, a large number of researchers proposed dozens of effective methods to solve DMOPs. According to their properties and characteristics, these approaches can be categorized by some representative schemes adopted in algorithm development, such as diversity, memory, prediction, decision variable classification and others. For classical DMOPs, most of these methods are competitive in keeping convergence, or remain superior in decreasing the loss of diversity.

2.1. Methods based on diversity

In DMOPs, once a new environment occurs, the populations have been convergent. As a result, the optimization algorithm always lacks the exploration ability, and then hardly jumps out of the local region. In order to solve this issue, some techniques, such as random initialization [12], hyper mutation [13], and immigration strategy [14], were used to overcome diversity loss. Deb et al. [15] proposed DNSGA-II-A and DNSGA-II-B to increase the population diversity. In DNSGA-II-A, a part of the initial individuals of the population were regenerated to replace the original ones. In DNSGA-II-B, a certain proportion of individuals in the current population was updated by the mutation operation. Woldesenbet and Yen presented a triggered hyper mutation method [16]. A new mutation strategy [17] was combined with the prediction model to increase the diversity of individuals. In [18], Camara and Ortega selected some non-dominated individuals by crowding distance mechanisms, and these individuals were taken as candidates for mutation. Furthermore, a diversity preservation strategy was stated by Ruan et al. [19]. Most of these methods are good at keeping diversity, but they seem ineffective for complicated and unpredictable environments.

2.2. Methods based on convergence

With the purpose of obtaining high quality initial population to speed up convergence, the strategies based on memory and prediction are widely used in DMOPs. Memory-based approaches are

available for periodic or cycle environment changes. In this framework, the known historical optimal information of individuals is always utilized to respond to the new environment. Liang et al. [20] proposed a hybrid of memory (HMPS) and prediction strategies to improve the performance of the algorithm on the basis of whether the changes were similar to the previous ones. A new prediction and memory strategy (PMS) was applied by Peng et al. [21] as an effective response method. In [22], Wang and Li considered two memory schemes. The first one selected individuals from the archive and reused them. The second scheme selected individuals from the archive, and then used Gaussian perturbation on these individuals to produce new individuals which survived in the new environment. In [23], Koo et al. came up with a novel memory technique by sieving individuals with lesser crowding distance and generating new ones to act as elements for initial population. These schemes, as memory strategies, have been used for solving DMOPs and showed better performance than earlier approaches. Nevertheless, how to take full advantage of existing data is still a key but hard issue. To adopt redundant history information may be computation-expensive and even useless, whereas too little historical information could not provide useful value for exploring new environment.

To fully take advantage of previous experience to guide new search direction, some techniques based on prediction have been utilized generally in DMOPs. A large number of experiment results show that effective prediction mechanism is helpful to accelerate the convergence of the algorithm. Strategies using prediction can learn the change patterns from the previous environment and predict the new locations of the optimal solutions, which can improve the efficiency of the algorithm. In [24], PS was conceived to be constructed by a center as well as manifolds (PPS). The known centers were serving to predict new centers, and the subsequent manifolds were estimated by using the previous manifolds. New centers and corresponding manifolds were taken as new individuals for the next generation of the evolution. Zou et al. [25] proposed a prediction strategy (CKPS) in which individuals were classified into three categories: non-dominated points, knee points and adaptive random points. The non-dominated ones with existing center point were going to participate in predicting. The evolutionary direction was controlled by knee points. The number of random points was determined by the severity of environmental changes. Kalman filter (KF) strategy was employed to look for PS and PF in the new environment by Muruganatham et al. [26]. A multi-directional prediction strategy (MDP) was designed by predicting the chosen representative individuals to guide the evolution direction [27]. Cao et al. [28] used a differential model to predict the locations of new individuals, in which three adjacent centers were adopted from different environments. In [29], Zheng et al. tested the effect of different prediction methods with different decision variables on the convergence and distribution of individuals, respectively. Li et al. [30] put forward a dual prediction strategy with an inverse model which predicted individuals in objective space as well as decision space, and good information of objective space was used to provide feedback for decision space. Rambabu et al. [31] built a mixture-of-experts-based ensemble framework which combined diversified types of predictors with gating network to describe prediction performance at different environments. In a word, these prediction methods try to get close to the new PS and PF by generating superior individuals to enhance the exploration capability of algorithms. However, the prediction error may be true if the historical information is not enough or predicting model is not inappropriate. Hence, it is necessary to establish a reasonable prediction model to improve the search performance of the algorithm.

2.3. Methods based on other schemes

Except the methods based on diversity and convergence, there are other strategies to tackle DMOPs. For example, Xie et al. [32] adopted decision-variables-classification based cooperative evolution to improve the performance of the algorithm. In DMOPs, the probability distributions of the decision variables may change in different environments. On this basis, Liang et al. [33] used the Spearman rank correlation coefficient (SRCC) to measure the correlations between decision variables and objectives functions, which could efficiently produce potential high-quality points. In [34], Chen et al. proposed a domain adaptation learning strategy by introducing a mapping matrix between the search spaces of past and current environments. Zhang and Yang [35], sampled individuals by classification-based sampling to enhance population diversity and used the method of probability-based space shrinkage to generate high quality solutions. Zhang et al. [36] made use of the precision controllable mutation and the simulated isotropic magnetic particles niching to make individuals approximate the entire Pareto front automatically. In addition, other effective strategies have also been proposed in recent years. These methods above can enhance the search performance of algorithms to some extent.

3. Preliminaries

3.1. Basic definitions

Definition 1. Time parameter

The time t , as a discrete parameter, is defined by the following mathematical formula [37]:

$$t = \frac{1}{n_t} \cdot \left\lfloor \frac{\tau}{\tau_t} \right\rfloor \quad (2)$$

where τ is the number of generations, n_t means change severity, τ_t is used to describe the change frequency of environment, and parameters n_t and τ_t control the degree of environmental change.

Definition 2. Dynamic Pareto dominance [37]

Suppose $u(t)$ and $v(t)$ are arbitrary individuals from decision space Ω at time t . $u(t)$ is said to dominate $v(t)$ (denoted by $u(t) < v(t)$) if and only if the relational expressions $f_i(u, t) \leq f_i(v, t)$, $\forall i \in \{1, 2, 3, \dots, m\}$, and $f_j(u(t)) < f_j(v(t))$, $\exists j \in \{1, 2, 3, \dots, m\}$, are true. $u(t)$ is called a non-dominated solution if there is not a solution which can dominate $u(t)$.

Definition 3. Pareto optimal solution set

Pareto optimal solutions set at time t is composed of selected non-dominated individuals and denoted by PS_t , which can be expressed by this form:

$$PS_t = \{u(t) \mid \nexists v(t) \in \Omega, v(t) < u(t)\} \quad (3)$$

Definition 4. Pareto optimal front

The Pareto optimal front at time t , known as PF_t , can be expressed by the set as below:

$$PF_t = \{f(u(t)) \mid u(t) \in PS_t\} \quad (4)$$

Definition 5. The types of DMOPs

According to the dynamic change characteristic of the PS and PF, Farina et al. [38] classified DMOPs into four types.

Type I: The PS changes over time but the PF is fixed.

Type II: Both the PS and PF change over time.

Type III: The PS remains fixed, while the PF changes over time.

Type IV: Both the PS and PF remain fixed.

Although the fourth type appears on occasion, the optimal solutions keep constant. As a result, we mainly focus on studying the first three types.

Definition 6. Prediction based on center

Given the stability of the geometrical center, the prediction schemes based on centers are widely used in dynamic optimization algorithms. This is because the moving direction of individuals are consistent with centers in some cases. The center of the PS can be expressed as follows:

$$C_t = \frac{1}{|PS_t|} \sum_{x \in PS_t} x \quad (5)$$

where the number of solutions in PS_t is denoted by $|PS_t|$, x is an element pertaining to the set PS_t . This type of prediction model works well for a class of DMOPs in which the change of PS can be described by the center of the PS.

3.2. Change detection

Dynamic multi-objective optimization always involves change detection, change response and other related procedures. There are two common methods which are used in DMOPs to probe the change of environment. The first one is a fixed detectors approach, which always evaluates a number of detector vectors to observe whether the objective values of them undergo changes. This approach needs to re-evaluate the objective function values at every generation and is not suitable for noisy environments. The second method is a behavior-based approach [39,40], which involves the discrepancy and the statistical information of solutions in objective space. The behavior-based detection approach is popular because there are no additional fitness evaluations. As a result, we prefer the behavior-based scheme to detect whether the new environment appears. Once environment changes are detected, most of the strategies described above based on diversities and convergence belong to common change response methods.

4. Proposed algorithm

In the section, a new method of dividing change type is proposed. The change types of environments are described in Figure 1. They are divided into translational changes and non-translational changes. Predicting models mainly depend on different change types. Therefore, for the translational change, a predicting strategy with center is adopted. Conversely, the multi-direction strategy is used to generate new individuals adaptively. The predicting model is named CMDS. In our studies, MOEA/D is used to execute evolution and the Tchebycheff approach acts as a decomposition method in static optimization stage [41]. Figure 2 shows the framework of the proposed algorithm (MOEA/D/CMDS). Next, we describe the detailed process of the algorithm. The whole structure of MOEA/D/CMDS is shown in Algorithm 1.

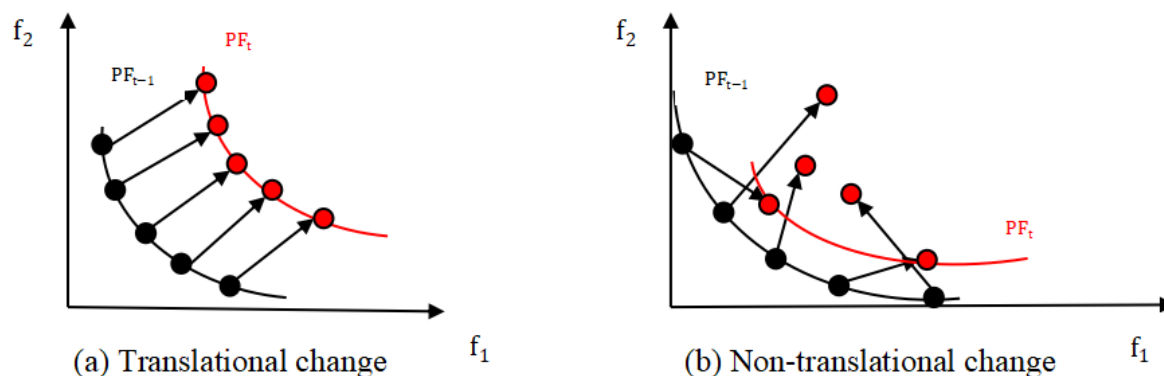


Figure 1. The change type of environment.

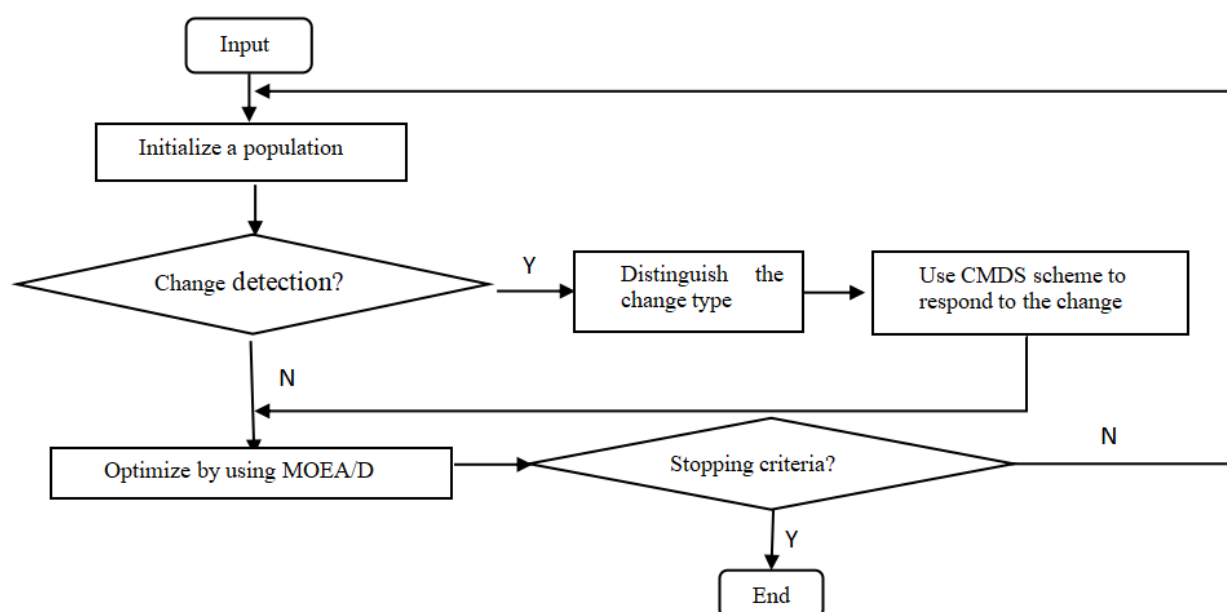


Figure 2. The framework of MOEA/D/CMDS.

Algorithm1: The whole framework of MOEA/D/CMDS

- 1: Input: N (population size), time step $t = 0$, initial population P_0 ;
 - 2: Output: Approximated PS_t and PF_t at different moments;
 - 3: while stopping criteria is not satisfied do
 - 4: Detect an environment for change
 - 5: if change appears, then
 - 6: Use Algorithm 2 to decide the change type and apply CMDS scheme to generate new solutions;
 - 7: $t = t + 1$;
 - 8: end if
 - 9: Optimize the static MOPs with MOEA/D;
 - 10: end while
-

4.1. Types of changes

For each individual x_i^t , $i \in \{1, 2, \dots, N\}$ at time t , its corresponding objective function value is $f_k(x_i^t)$, $k \in \{1, 2, \dots, m\}$. All elements in set $\{f_k(x_1^t), f_k(x_2^t), \dots, f_k(x_N^t)\}$ are sorted from the smallest value to the largest one. We record the sequential number of x_i^t with R_i^{kt} . In other words, the rank value of x_i^t is R_i^{kt} for the k -th objective at moment t . The smaller sequential number means smaller function value for each solution on the corresponding objective. The sum of R_i^{kt} for individual i in each objective function is denoted by R_i^t , and its expression is as follows:

$$R_i^t = \sum_{k=1}^m R_i^{kt} \quad (6)$$

where the maximum value of R_i^t is \max^t and the minimum value of it is \min^t for the varying i . The sequence $R^t = (R_1^t, R_2^t, \dots, R_N^t)$ represents the rank values of all solutions at moment t . As it should be, the sequence $R^{t-1} = (R_1^{t-1}, R_2^{t-1}, \dots, R_N^{t-1})$ displays the rank values at moment $t - 1$. The intensity variation of the rank values corresponding to individual i is represented by the difference between R_i^t and R_i^{t-1} . That is to say, the location variation degree for the i -th individual at t moment and $t - 1$ moment is measured by the value $|R_i^t - R_i^{t-1}|$. When the absolute value is small enough, we can assume that the change of location is not obvious for corresponding solution, or else the change of location is sharp. In order to distinguish the change type, the average value of $|R_i^t - R_i^{t-1}|$ ($i = 1, 2, \dots, N$) is calculated and denoted by $V(R_t)$. It is expressed below:

$$V(R_t) = \frac{\sum_{i=1}^N |R_i^t - R_i^{t-1}|}{N} \quad (7)$$

we construct threshold value α_t which is below at time t :

$$\alpha_t = \frac{\max^t - \min^t + 1}{e^m} \quad (8)$$

where α_t varies with environment. At the same time, as the dimensionality m of the objective function increases, \max^t and \min^t grow intensely. Hence, the difference of rank sums between the best solutions and the worst solutions is used to depict a change degree for extreme individuals. Smaller value of it means better stability of population relative to the increment of objective function number m . If the position movement of all individuals between moments $t - 1$ and t is not drastic, $V(R_t)$ must be smaller. Conversely, the larger $V(R_t)$ is true. According to the above idea, the threshold value α_t can be set. When $V(R_t)$ is less than α_t , the change type of environment is translational; Otherwise, it is considered as non-translational. For example, in the Figure 1, the black dots represent individuals at $t - 1$ moment, and the red dots show solutions at t moment. The solutions are noted as x_1, x_2, x_3, x_4, x_5 from left to right at moment. Arrows indicate the direction of individual evolution. In part (a) of Figure 1, rank sums of each solution do not vary. Hence, $V(R_t)$ is smaller than α_t and the change type of environment is translational. However, in part (b) of Figure 1, individual x_2 has the maximum rank sum 8 and individual x_1 has the minimum rank sum 3 at the current moment t . It is obvious that $V(R_t)$ is larger than α_t , so the change type of environment is determined. The frame of identifying the change type of new environment is stated in Algorithm 2.

Algorithm 2: A method of distinguishing the change type

- 1: Calculate the rank values R_i^t and R_i^{t-1} at two adjacent moments;
 - 2: Compute the variation degree $|R_i^t - R_i^{t-1}|$ of individual i at time t ;
 - 3: Estimate the average variation degree $V(R_t)$ of the whole population according to Eq (7) in the present environment;
 - 4: Create the threshold value α_t by Eq (8);
 - 4: if $V(R_t) < \alpha_t$ holds
 - 5: The change type of environment is translational;
 - 5: else
 - 6: The change type of environment is non-translational;
 - 7: end if
 - 8: Return the type of environment change;
-

4.2. Prediction strategies

The intention of prediction is to make the individuals quickly converge to the new PF before the next environment change is detected. Effective prediction strategies can reduce computation costs and promote algorithm convergence. Cao et al. proposed a second-order difference model to predict the moving trends of centroid of the solutions obtained [28], but the distinctions of individuals were neglected. The centers adopted by them were determined with all ones of the population. In our paper, solutions are selected, making up for the above shortcomings, and we employ a new prediction strategy which responds to the environment in terms of environment change type.

4.3. Response to translational change

Due to the high efficiency of using center predicting, when the change is translational, there are no extreme individuals whose rank sums alter intensely. Thus, we can conclude that the present environment could have great similarity with previous one. The change of environment is regular and not sharp. That is to say, the change is relatively stable. Hence, we can assume that the movement direction of the individual is consistent with that of the center of PS.

In this paper, a new estimation method of PS is put forward. For the sequence R^t , if the value of R_i^t is relatively small, it follows that individual i has good characteristic. All individuals with the same properties as individual i constitute the set B_t , which approximates PS_t at the moment t . The individuals which do not belong to set B_t are deemed to be in the set A_t . It is apparent that present population is constituted by the elements of the set A_t and B_t . How to determine elements in the set B_t ? The smaller rank value of individual implies the better proximity to PS. However, if there are too few elements in set B_t , the characteristic of the population could not be reflected fully. For the same reason, too many elements do not exactly reveal the superiority of individuals with lower rank values for B_t . Taking these factors into consideration, the threshold value β_t is calculated in the following way:

$$\beta_t = \min^t + \frac{\max^t - \min^t}{m} \quad (9)$$

It is evident that β_t varies over time t , and the average allocation of the difference between max^t and min^t for each objective function is used to balance discrepancies of whole objectives. If the value of R_i^t less than β_t is true, corresponding individual i is put into set B_t . By the same principle, the sets B_{t-1} and B_{t-2} can be determined by the corresponding β_{t-1} and β_{t-2} which are threshold values corresponding to moments $t-1$ and $t-2$. Apparently, individuals with higher quality are gathered in sets B_t , B_{t-1} and B_{t-2} . Then, the centers of these sets are separately denoted by $C(B_t)$, $C(B_{t-1})$, and $C(B_{t-2})$. Without loss of generality, in order to save computation cost, the previous three consecutive centers are used to construct new movement direction. In the process of predicting, maybe prediction error exists. To overcome this shortcoming, Gaussian perturbation with mean value 0 and standard deviation $\delta(t)$ is introduced. The prediction formula is designed as follows:

$$x_i^{t+1} = x_i^t + r_1 * (C(B_t) - C(B_{t-1})) + (1 - r_1) * (C(B_{t-1}) - C(B_{t-2})) + Gaussian(0, \delta(t)) \quad (10)$$

where r_1 is a parameter randomly taken in (0,1) and $\delta(t)$ is expressed in the following form:

$$\delta(t) = \frac{1}{max^t - min^{t+1}} \quad (11)$$

It is obvious that smaller value $max^t - min^t$ means more clustered individuals, so the disturbance amplitude should be larger. This symbolizes that larger value of $\delta(t)$ is established. Otherwise, the distribution of individuals is relatively scattered. The population has relatively better diversities, then only smaller disturbance amplitude is required. In a word, Gaussian perturbation is introduced to improve the search and exploration capability of population. In this response mechanism, the present population is divided into two subpopulations. Solutions of the first subpopulation are used to execute prediction. The individuals in the second subpopulation serve as the initial ones in the new environment directly. It is worth noting that there is no stored historical information to meet with the prediction formula of Eq (10) when the algorithm runs at the earliest three moments. In this case, all of the individuals are randomly initialized to respond the new environment change.

4.4. Response to non-translational change

When the non-translational change is detected, the population is divided into three parts. For different parts, the corresponding individual generating scheme is adopted. The set B_t is as described above at present moment. It is easy to find that subpopulations B_t and A_t are distinguished by the value of β_t . For a given individual x_u^t in set B_t , considering each individual x_v^{t-1} , $v = 1, 2, 3, \dots, |B_{t-1}|$ in set B_{t-1} , the Euclidean distance $dist(x_u^t, x_v^{t-1})$ between x_u^t and x_v^{t-1} is calculated. The individual x_v^{t-1} with minimum $dist(x_u^t, x_v^{t-1})$ is recorded as $x_{u^*}^{t-1}$. It follows that $x_{u^*}^{t-1}$ is the nearest point to x_u^t for all individuals in set B_{t-1} . Using the same way, the individual $x_{u^*}^{t-2}$ could be found in the set B_{t-2} . The Euclidean distance of x_u^t and $x_{u^*}^{t-2}$ is the minimum for all solutions in the set B_{t-2} . Next, a new prediction model which is called multi-direction strategy is proposed for each individual x_u^t in set B_t .

$$x_u^{t+1} = x_u^t + r_2 * (x_u^t - x_{u^*}^{t-1}) + (1 - r_2) * (x_u^t - x_{u^*}^{t-2}) \quad (12)$$

where r_2 is a random number from (0,1). We utilize pointwise prediction model for each element in the set B_t . Each individual in B_t has its direction. Obviously, the individuals which do not take part in this prediction belong to the set A_t . We consider the individuals whose rank sum is min^t at present

time t . In many cases, more than one individual has such feature, so these ones are known as the best solutions, which constitute the set G_t . We assume that the center of G_t is $C(G_t)$. Then, we categorize the elements in the set A_t according to random probability to increase the diversity of the population. For individuals in set A_t , the first part solutions which are regarded as belonging to set A_{t1} are selected by a random probability to accept the below prediction strategy:

$$x_r^{t+1} = x_r^t + \gamma * (C(G_t) - C_{tt}) + (1 - \gamma) * (C(G_{t-1}) - C_{tt}) \quad (13)$$

where C_{tt} is the center of the whole population and γ is a random number from (0,1). The second part solutions which constitute the set A_{t2} are the rest ones in the set A_t . The way of generating new individuals for it is random initialization. The method is showed as below:

$$x_{ri}^{t+1} = Lb_i + (Ub_i - Lb_i) * \theta \quad (14)$$

where Lb_i and Ub_i which are given in the experiment are the upper bound and lower bound of each dimension variable, $i = 1, 2, \dots, n$. i represents dimension of individuals x_{ri}^t and x_{ri}^{t+1} . θ is a random number relying on $[0, 1]$. The detailed description about response strategies for two change types of environments of CMDS is described in Algorithm 3.

Algorithm 3: CMDS scheme

- 1: Input: The population P^t at t moment;
 - 2: Output: Initial population P^{t+1} at $t + 1$ moment;
 - 3: if the translational change appears
 - 4: for $i = 1$ to N do
 - 5: if $i \bmod 2 = 0$
 - 6: Apply Eq (10) to generate new solutions
 - 7: else
 - 8: The i -th individual is directly put into the initial population at next moment;
 - 9: end if
 - 10: end for
 - 11: else
 - if individuals belong to B_t
 - Generate new solutions by Eq (12);
 - 12: else if $rand(0,1) < 0.5$
 - Put individuals into the set A_{t1} and use Eq (13) to produce new ones;
 - 13: else
 - Adopt Eq (14) to initialize individuals;
 - 14: end if
 - 15: Enter into static optimization;
-

In general, a method relying on rank sums of individual is designed in this paper. Different prediction strategies are designed for individuals of different categories. Next, the results of the experiment are showed in the following section.

5. Experiment design

5.1. Test instances

The proposed algorithm MOEAD/D/CMDS is tested on 20 benchmark instances, including FDA1–FDA5 test suites [38], dMOP1–dMOP3 [42] test suites, ZJZ test suites (F5–F8) and JY1–JY8 test suites [24]. By extending FDA problems, the dMOP test suites are acquired. A linear relationship exists among the decision variables of FDA and dMOP. However, there are nonlinear correlations in decision variables of F5–F8. For JY test suites, there are mixed PF and intricate relationships among decision variables which are all challenging features for dynamic multi-objectives optimization.

5.2. Compared algorithms

The proposed algorithm is compared with MOEA/D/DM [28], MOEA/D [41], MOEA/D/KF [26], DNSGA-II-A [15] and DSS [43]. MOEA/D/DM used a second-order center difference model to predict the moving trends of part individuals. Classical multi-objective optimization algorithm MOEA/D was employed for DMOPs after being slightly modified, wherein a part of individuals was randomly initialized. As is known to all, the same idea was applied to DNSGA-II-A. In the algorithm MOEA/D/KF, Kalman filter model was used to search the new direction of solutions. However, different from the above DMOPs algorithms, a directed local search method with linear prediction model is taken into DSS.

5.3. Performance metrics

Performance metrics can evaluate convergence, distribution, and diversity of the obtained population. The generation distance (GD) has been widely used in evaluating algorithm and was introduced in [42]. The expression form of GD is given as follows:

$$GD = \frac{\sum_{v \in P_t} d(PF_t, v)}{|P_t|} \quad (15)$$

where the PF_t is consisted by evenly distributed points of the true PF at t moment and P_t is a set which is approximate to PF_t . The definition of $d(PF_t, v)$ is as follows:

$$d(PF_t, v) = \min_{u \in PF_t} \sqrt{\sum_{j=1}^m (f_j(v) - f_j(u))^2} \quad (16)$$

where $d(PF_t, v)$ is the smallest Euclidean distance between individual v and the set PF_t . $|\cdot|$ represents cardinality of the corresponding set. The smaller GD of the algorithm symbolizes the better convergence. The IGD [19,44] is a modified version of GD . The expression of it is as below:

$$IGD = \frac{\sum_{u \in PF_t} d(u, P_t)}{|PF_t|} \quad (17)$$

and $d(u, P_t)$ can be expressed as:

$$d(u, P_t) = \min_{v \in P_t} \sqrt{\sum_{j=1}^m (f_j(v) - f_j(u))^2} \quad (18)$$

$d(u, P_t)$ means the minimum Euclidean distance between point u on the PF_t and the set P_t . As is known to all, IGD is a useful index to measure the convergence and diversity of the obtained solutions.

The average value of IGD is defined as $MIGD$ metric for all time windows in a single run [45].

$$MIGD = \frac{1}{|T|} \sum_{t \in T} IGD(PF_t, P_t) \quad (19)$$

where T is a set and represents all moments. Its cardinality is $|T|$. The smaller value of $MIGD$ means the better performance of the algorithm. In this paper, we employ $MIGD$ value to measure the performance of the algorithm MOEA/D/CMDS.

5.4. Parameter setting

Considering the change frequency and change severity affect the performance of algorithm. The experiment is tested on $n_t = 5, 10, \tau_t = 5, 10$. Hence, there are 4 types for each problem. In order to maintain fairness, the parameters in the comparison algorithms are taken from original references. The size of the population is 100 and 300 for bi-objective problems and tri-objective problems. The values of parameters r_1 and r_2 are equal to 0.6. At the same time, the value of γ is set to 0.8. Relatively larger values of r_1 , r_2 , and γ reflect that present environment data is dominant for generating new solutions. Lb_1 and Ub_1 are taken as 0 and 1, separately. Meanwhile, Lb_i is assigned as -1 and Ub_i is set as 1 when i changes from 2 to n .

In DE operator, CR is fixed as 0.5 and F is set to 0.5. In the polynomial mutation operation, the value of parameter η is 20. $p_m = 1/n$ is true, in which n represents the dimensions of the decision space. The size of the neighborhood is 20. The coefficient δ is 0.8. The size of n_r is the same as the cardinality of mating pool E, where parent individuals are selected to generate new ones. The total number of iterations is set to $40\tau_t$, which assure 40 environmental changes to be carried out in each run. The proposed algorithm is executed independently 30 times for each example, and the $MIGD$ value is taken as a measure metric, the smaller $MIGD$, the better performance of MOEA/D/CMDS.

6. Experiment result analysis

The Tables 1–4 show the experiment results of each type of problem on $MIGD$ values and their standard deviations. The best results are marked in bold. The detailed analysis about the test result is as follows. The curves of tracking the IGD trend over 40 environment changes for 20 runs are showed in the Figures 3–5.

6.1. Results on FDA and dMOP problems

This section shows the experiment results about test functions FDA and dMOP. The PS of FDA1 changes as a sine curve and the PF keeps fixed. MOEA/D/CMDS performs better than or close to MOEA/D/DE except the case $n_t = 5, \tau_t = 10$. For FDA2 problem, the shape of PF is convex and non-convex alternation in the objective space. At this case, MOEA/D/CMDS is slightly superior to or equivalent to other compared algorithms for all cases, which indicates that prediction strategy can improve the search ability of the algorithm. In FDA3 and FDA5, different density distributions exist

in distinct segments of the PF. These two types of changes don't follow any assumed models, which cause certain challenges for all algorithms. However, the individuals with smaller rank sums play a positive guidance in the former adjacent environment for high severity change, which show the effectiveness of MOEA/D/CMDS. The test function FDA4 is a tri-objective problem whose PF and PS change with time. In this case, the moving direction provided by MOEA/D/CMDS is a disadvantage because the change of position sometimes inaccurately describes the movement direction of individuals. The results of FDA4 for all parameters are inferior to MOEA/D/DM, but they are obviously better than other compared algorithms. Specific experimental results are shown by Table 1.

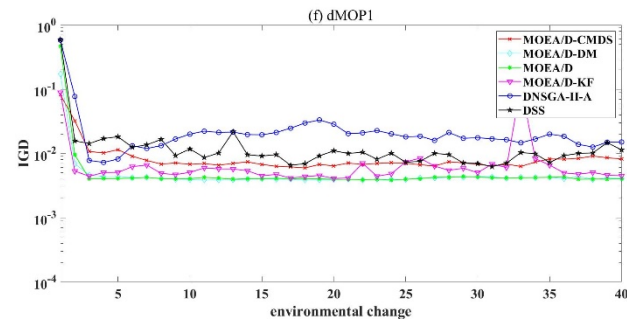
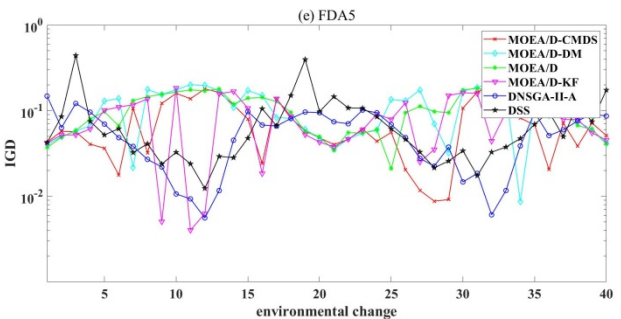
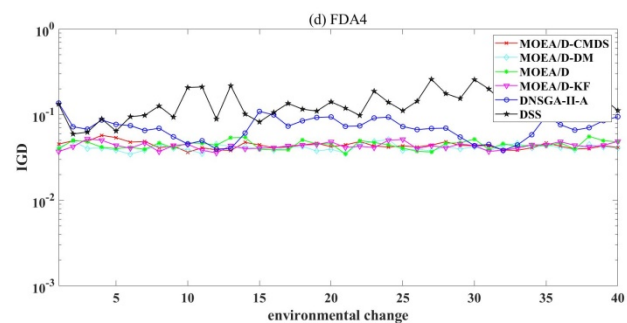
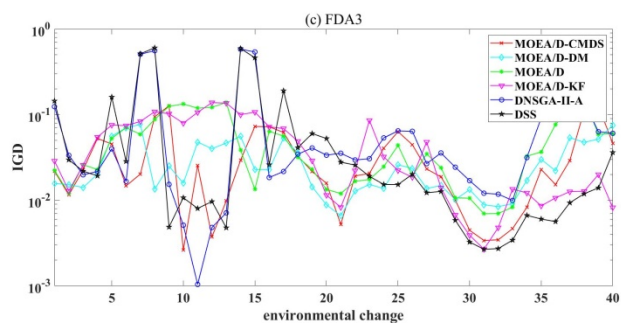
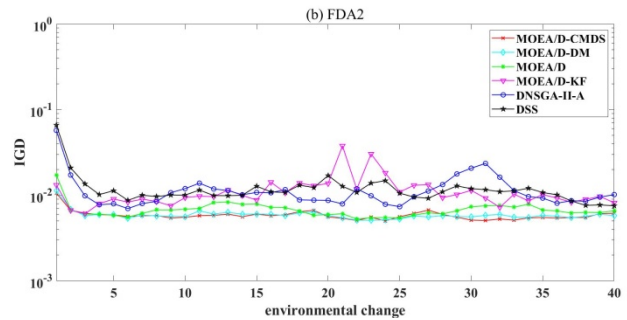
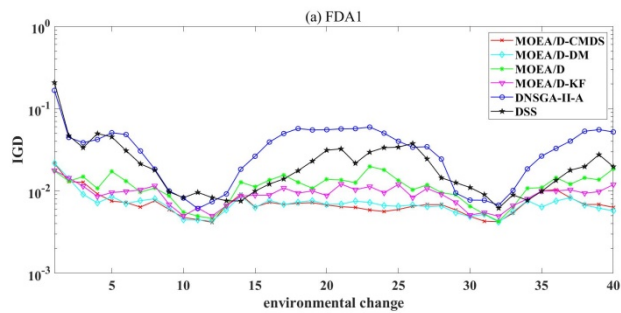
Table 1. Mean and standard deviations of *MIGD* values on FDA test set.

Problem	(n_r, τ_t)	MOEA/D/CMDS	MOEA/D/DM	MOEA/D	MOEA/D/KF	DNSGA-II-A	DSS
FDA1	(10,5)	0.0120(0.0006)	0.0128(0.0007)+	0.0297(0.0007)+	0.0249(0.0024)+	0.1135(0.0044)+	0.0596(0.0041)+
	(5,5)	0.0169(0.0007)	0.0201(0.0019)+	0.0807(0.0122)+	0.0392(0.0161)+	0.2005(0.0053)+	0.0757(0.0036)+
	(5,10)	0.0086(0.0004)	0.0082(0.0002)≈	0.0169(0.0007)+	0.0101(0.0002)+	0.0792(0.0031)+	0.0392(0.0021)+
	(10,10)	0.0069(0.0006)	0.0071(0.0002)≈	0.0116(0.0002)+	0.0099(0.0001)+	0.0381(0.0020)+	0.0236(0.0011)+
FDA2	(10,5)	0.0083(0.0002)	0.0089(0.0006)≈	0.0107(0.0005)+	0.0587(0.0179)+	0.0221(0.0016)+	0.0312(0.0041)+
	(5,5)	0.0099(0.0003)	0.0125(0.0019)+	0.0157(0.0011)+	0.0573(0.0204)+	0.0342(0.0014)+	0.0541(0.0039)+
	(5,10)	0.0067(0.0004)	0.0070(0.0007)≈	0.0084(0.0005)≈	0.0110(0.0004)+	0.0154(0.0006)+	0.0190(0.0012)+
	(10,10)	0.0059(0.0001)	0.0060(0.0002)≈	0.0070(0.0004)+	0.0107(0.0006)+	0.0119(0.0007)+	0.0126(0.0004)+
FDA3	(10,5)	0.0510(0.0023)	0.0552(0.0074)+	0.0736(0.0088)+	0.0732(0.0099)+	0.1256(0.0061)+	0.1007(0.0126)+
	(5,5)	0.0680(0.0015)	0.0726(0.0069)+	0.1089(0.0070)+	0.0795(0.0097)+	0.1668(0.0072)+	0.1031(0.0109)+
	(5,10)	0.0461(0.0059)	0.0471(0.0067)≈	0.0671(0.0060)≈	0.0533(0.0006)+	0.1016(0.0054)+	0.0882(0.0071)+
	(10,10)	0.0345(0.0001)	0.0281(0.0027)-	0.0511(0.0084)+	0.0466(0.0073)+	0.0960(0.0055)+	0.0852(0.0029)+
FDA4	(10,5)	0.0563(0.0006)	0.0469(0.0005)-	0.0572(0.0007)+	0.0544(0.0013)-	0.1257(0.0023)+	0.1376(0.0080)+
	(5,5)	0.0542(0.0015)	0.0488(0.0017)-	0.0576(0.0011)+	0.0545(0.0004)-	0.1816(0.0047)+	0.1527(0.0027)+
	(5,10)	0.0427(0.0003)	0.0411(0.0006)-	0.0439(0.0001)+	0.0429(0.0001)+	0.0949(0.0010)+	0.1501(0.0088)+
	(10,10)	0.0424(0.0001)	0.0401(0.0027)-	0.0454(0.0002)+	0.0430(0.0004)+	0.0715(0.0005)+	0.1269(0.0095)+
FDA5	(10,5)	0.0658(0.0039)	0.0986(0.0073)+	0.0854(0.0153)+	0.0544(0.0013)+	0.1049(0.0010)+	0.0813(0.0017)+
	(5,5)	0.0674(0.0035)	0.1063(0.0012)+	0.0992(0.0016)+	0.0873(0.0012)+	0.1387(0.0013)+	0.0841(0.0085)+
	(5,10)	0.0784(0.0053)	0.0818(0.0032)+	0.0656(0.0013)-	0.0636(0.0073)-	0.0767(0.0067)-	0.0819(0.0072)+
	(10,10)	0.0746(0.0024)	0.0787(0.0040)+	0.0705(0.0069)-	0.0602(0.0009)-	0.0608(0.0070)-	0.0725(0.0015)+

The statistical results about dMOP are listed in Table 2. The change type of dMOP1 belongs to Type III which keeps static PS, whereas the PF and PS of dMOP2 are all time-varying. For these two instances, when the change frequency parameter τ_t is equal to 5, MOEA/D/CMDS performs well because of powerful guidance of better history solutions. At the same time, if the value of τ_t is equal to 10, the degree of change of individual position is very small. Hence, the loss of diversity is obvious, which causes the degradation of algorithm performance. In dMOP3, only PS changes over time, and the variables which control the spread of the PF vary randomly. This mechanism of change brings certain difficulties for prediction schemes. Good performance of MOEA/D/CMDS shows its superiority in dMOP3. Figure 3 presents the trajectories of IGD on these two test problems.

Table 2. Mean and standard deviations of *MIGD* values on dMOP test set.

Problem	(n_v, τ_v)	MOEA/D-CMDS	MOEA/D-DM	MOEA/D	MOEA/D-KF	DNSGA-II-A	DSS
dMOP1	(10,5)	0.0248(0.0085)	0.0261(0.0108)≈	0.0339(0.0184)-	0.0303(0.0102)-	0.1129(0.0460)+	0.0832(0.0023)+
	(5,5)	0.0250(0.0080)	0.0298(0.0100)+	0.0572(0.0319)+	0.0594(0.0297)+	0.1172(0.0337)+	0.0882(0.0060)+
	(5,10)	0.0133(0.0012)	0.0093(0.0039)-	0.0106(0.0036)≈	0.0104(0.0070)≈	0.0448(0.0066)+	0.0380(0.0194)+
	(10,10)	0.0116(0.0015)	0.0081(0.0030)-	0.0107(0.0061)≈	0.0125(0.0025)+	0.0396(0.0175)+	0.0350(0.0116)+
dMOP2	(10,5)	0.0180(0.0018)	0.0261(0.0087)+	0.0449(0.0014)+	0.0319(0.0105)+	0.0982(0.0026)+	0.0538(0.0061)+
	(5,5)	0.0293(0.0024)	0.0404(0.0022)+	0.0914(0.0075)-	0.0416(0.0062)+	0.1637(0.0068)+	0.0638(0.0126)+
	(5,10)	0.0109 (0.0007)	0.0129(0.0024)+	0.0203(0.0006)+	0.0135(0.0005)+	0.0656(0.0015)+	0.0431(0.0036)+
	(10,10)	0.0084 (0.0022)	0.0077(0.0009)-	0.0141(0.0026)+	0.0098(0.0004)+	0.0407(0.0005)+	0.0256(0.0081)+
dMOP3	(10,5)	0.0391(0.0055)	0.0748(0.0048)+	0.0535(0.0074)-	0.0644(0.0146)+	0.0746(0.0019)+	0.0809(0.0055)+
	(5,5)	0.0511(0.0062)	0.1138(0.0091)+	0.0831(0.0106)+	0.0845(0.0131)+	0.1193(0.0035)+	0.0913(0.0107)+
	(5,10)	0.0150(0.0037)	0.0205(0.0026)+	0.0197(0.0018)+	0.0156(0.0091)≈	0.0478(0.0016)+	0.0639(0.0037)+
	(10,10)	0.0132(0.0010)	0.0184 (0.0006)+	0.0159(0.0020)+	0.0141(0.0009)+	0.0277(0.0011)+	0.0443(0.0030)+



Continued on next page

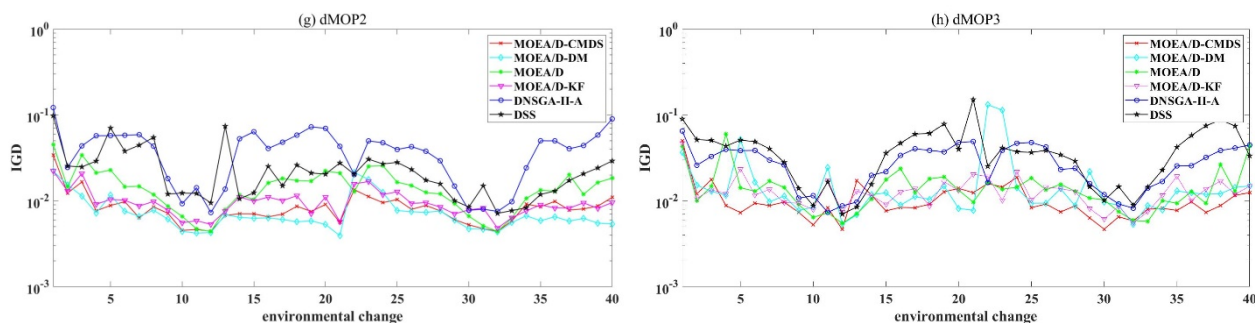


Figure 3. Tracking curves obtained by the six algorithms on FDA and dMOP test instances with $n_t = 10, \tau_t = 10$ over 20 runs.

6.2. Results on ZJZ problems

The ZJZ test instances include F5–F8. The nonlinear relation is obvious between decision variables. Besides, according to the value of $V(R_t)$, the changes type of PF is non-translational. The change of the position for individuals is sharp in two adjacent environments. Hence, it is difficult to fully utilize the information of the previous time windows. Therefore, ZJZ problems cause more challenges for DMOPs, especially when they face higher frequency and more severity changes. From Table 3, the algorithm proposed in this paper is very advantageous compared with other rivals except F8 which is a test function of three objectives. It is obvious that the MIGD values of MOEA/D/CMDS are far less than competitors. But for F8, the individuals which have smaller rank sums in the previous environment can't guide new search direction well, so MOEA/D/CMDS is disadvantage. In a word, MOEA/D/CMDS performs well on most of the test instances for ZJZ problems, and the value of MIGD has superiority. The IGD curves of tracking characteristic are showed in Figure 4, the results of experiment are provided by Table 3.

Table 3. Mean and standard deviations of *MIGD* values on ZJZ test set.

Problem	(n_t, τ_t)	MOEA/D/CMDS	MOEA/D/DM	MOEA/D	MOEA/D/KF	DNSGA-II-A	DSS
F5	(10,5)	0.2973(0.0323)	0.5565(0.0514)+	0.7451(0.0695)+	0.6644(0.2153)+	2.0327(0.0764)+	0.7313(0.0285)+
	(5,5)	0.1492(0.0275)	1.0183(0.2008)+	1.3587(0.0694)+	0.7054(0.7055)+	3.3076(0.0957)+	1.5468(0.2493)+
	(5,10)	0.1294(0.0150)	0.7266(0.0596)+	0.8078(0.0864)+	0.3495(0.0596)+	1.4451(0.0763)+	0.7022(0.0788)+
	(10,10)	0.0769(0.0159)	0.1691(0.0611)+	0.6148(0.0597)+	0.2738(0.0690)+	0.8145(0.1088)+	0.3515(0.0213)+
F6	(10,5)	0.0917(0.0228)	0.3688(0.0718)+	0.6259(0.1432)+	0.4923(0.1305)+	1.4238(0.0249)+	1.0297(0.1411)+
	(5,5)	0.0976(0.0181)	0.4232(0.0618)+	0.8785(0.1421)+	0.7259(0.0347)+	2.1295(0.0314)+	1.8156(0.3615)+
	(5,10)	0.0822(0.0141)	0.4518(0.0171)+	0.6953(0.0969)+	0.5842(0.0691)+	1.1189(0.0015)+	0.8147(0.0718)+
	(10,10)	0.0759(0.0233)	0.2304(0.0550)+	0.2858(0.0217)+	0.2474(0.0770)+	0.7294(0.0596)+	0.3385(0.0075)+
F7	(10,5)	0.1364(0.0753)	0.4115(0.0338)+	0.4735(0.0529)+	0.5940(0.2637)+	1.5068(0.0382)+	1.0893(0.0843)+
	(5,5)	0.1192(0.0167)	0.5551(0.0563)+	0.6335(0.0559)+	0.5889(0.0968)+	2.0911(0.0531)+	1.9872(0.0827)+
	(5,10)	0.0931(0.0181)	0.4418(0.0196)+	0.5227(0.0571)+	0.4971(0.0249)+	1.1363(0.0689)+	0.8015(0.1077)+
	(10,10)	0.0949(0.0357)	0.1767(0.0275)+	0.3590(0.0352)+	0.1993(0.0207)+	0.7488(0.0615)+	0.3621(0.0335)+
F8	(10,5)	0.1081(0.0022)	0.0869(0.0056)-	0.1212(0.0032)+	0.1432(0.0009)+	0.2648(0.0031)+	0.2847(0.0008)+
	(5,5)	0.1306(0.0064)	0.1083(0.0002)-	0.1423(0.0019)+	0.1478(0.0005)+	0.4124(0.0016)+	0.3066(0.0199)+
	(5,10)	0.1021(0.0030)	0.0868(0.0028)-	0.1061(0.0038) ≈	0.1029(0.0041)+	0.2231(0.0025)+	0.2267(0.0006)+
	(10,10)	0.0828(0.0035)	0.0742(0.0019)-	0.0966(0.0052)+	0.1081(0.0045)+	0.1568(0.0037)+	0.1820(0.0050)+

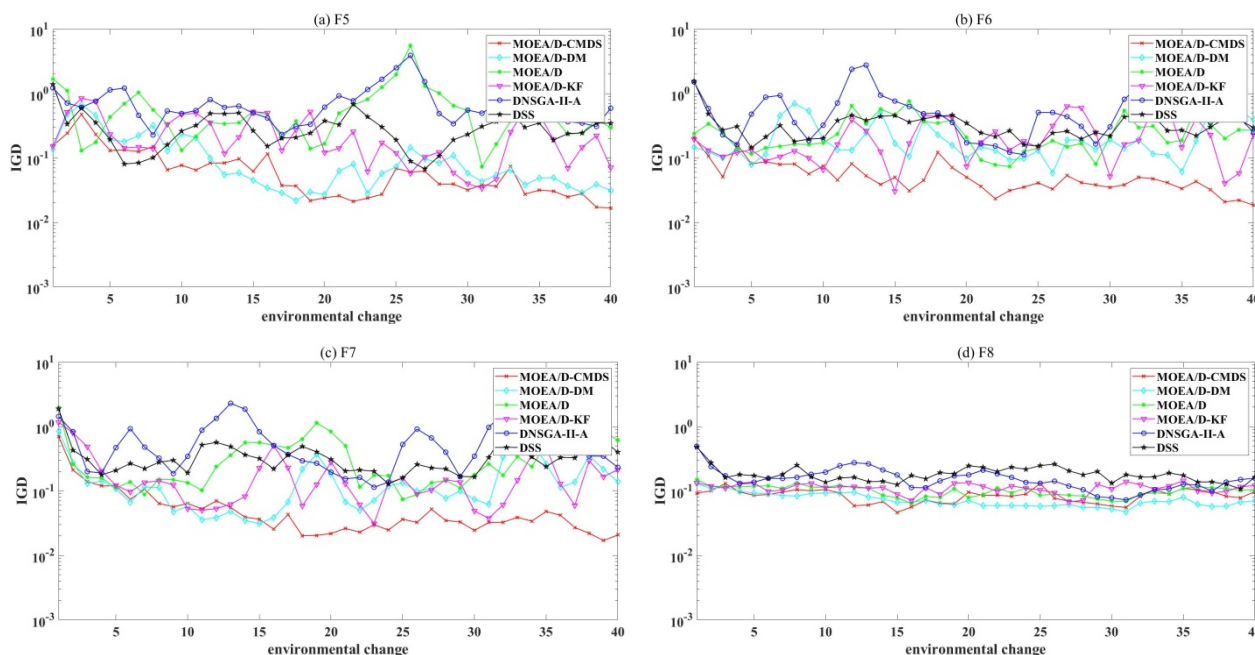


Figure 4. Tracking curves obtained by the four algorithms on ZJZ test instances with $n_t = 10, \tau_t = 10$ over 20 runs.

6.3. Results on JY problems

The results of the JY test are shown in Table 4. MOEA/D/CMDS has the better superiority on four problems JY3, JY4, JY6, and JY7, but it performs a little worse than the rest test problems. The decision variables of JY1 and JY2 are irrelevant and the center difference model of the whole population in MOEA/D/DM could improve the performance of the algorithm. For the JY3 problem, any two decision variables have time-varying non-monotonic dependencies. As time goes on, the PS becomes more and more complicated. At this case, MOEA/D/CMDS considers the nearest ones in $B_{(t-1)}$ and $B_{(t-2)}$ for each individual in B_t in order to make full use of history information. Of course, individuals with poor quality are discarded at the same time. Therefore, solutions from previous moments with smaller rank sums could improve the exploration ability of algorithm. The MIGD values of JY3 are far less than other compared algorithms for all cases. The PF of JY4 is discontinuous and its disconnected parts change with time. It is difficult to cover all the PF components. Although this type of problem is difficult, the method adopted in this paper is competitive when the environment changes severely. In other cases, the results are similar to MOEA/D/DM because the PF of JY5 keeps stationary. Thus, MOEA/D/CMDS performs poorly on this kind of problem. For the test functions JY6 and JY7, they are multimodal problems and the PF shapes of them appear concave and convex alternately. The number of local optima changes for JY6, but the opposite situation is true for JY7. Local optimum increases more evaluations to acquire the PS when the environment does not change. As it should be, algorithms may fall into local optimum, which affects their performance. In short, dealing with these two types of problems is very difficult. The PS of JY8 is static, and yet the segments of PF change with time. When the change of environment is fast and severe, the experiment results are approximate to the results of MOEA/D/DM. In other cases, MOEA/D/CMDS performs worse than MOEA/D/DM and MOEA/D, but it is obviously superior to most of the comparison

algorithms. Features of JY instances are reflected by Figure 5. The experiment results about JY series are shown by Table 4.

We mark the computational results with “+”, “-“ or “≈”, which indicate MOEA/D/CMDS is better than, worse than or equal to the compared algorithms. In a word, on FDA test set, the proposed algorithm shows its superiority on 10 among 15 test problems. For dMOP and ZJZ test problems, the proposed algorithm defeats other compared algorithms on at least three fourths of the problems. For the test functions JY, half test results of MOEA/D/CMDS are evidently superior to other compared approaches.

Table 4. Mean and standard deviations of *MIGD* values on JY test set.

Problem	(n_r, τ_r)	MOEA/D/CMDS	MOEA/D/DM	MOEA/D	MOEA/D/KF	DNSGA-II-A	DSS
JY1	(10,5)	0.0140(0.0004)	0.0119(0.0005)-	0.0221(0.0013)+	0.0165(0.0007)+	2.2917(0.1051)+	0.1353(0.0211)+
	(5,5)	0.0165(0.0006)	0.0140(0.0006)-	0.0331(0.0024)+	0.0168(0.0008)+	2.3698(0.1423)+	0.1417(0.0152)-
	(5,10)	0.0096(0.0001)	0.0088(0.0001)-	0.0124(0.0002)+	0.0094(0.0002)≈	2.3243(0.0606)+	0.0975(0.0055)+
	(10,10)	0.0091(0.0001)	0.0081(0.0001)-	0.0112(0.0004)+	0.0095(0.0001)+	2.2579(0.2837)+	0.0788(0.0091)+
JY2	(10,5)	0.0137(0.0003)	0.0119(0.0004)+	0.0211(0.0001)+	0.0153(0.0179)+	0.2006(0.0077)+	0.0934(0.0085)+
	(5,5)	0.0185(0.0007)	0.0139(0.0001)-	0.0321(0.0019)+	0.0165(0.0006)-	0.0314(0.0028)+	0.1197(0.0123)+
	(5,10)	0.0093(0.0001)	0.0075(0.0001)-	0.0114(0.0004)≈	0.0082(0.0001)-	0.0809(0.0119)-	0.0761(0.0080)-
	(10,10)	0.0083(0.0001)	0.0068(0.0001)-	0.0101(0.0002)+	0.0082(0.0002)≈	0.1114(0.0104)+	0.0514(0.0019)+
JY3	(10,5)	0.0248(0.0071)	0.1014(0.0002)+	0.1026(0.0009)+	0.1658(0.0210)+	0.1371(0.0050)+	0.1522(0.0326)+
	(5,5)	0.0223(0.0114)	0.1065(0.0050)+	0.1021(0.0001)+	0.1669(0.0050)+	0.1398(0.0072)+	0.1593(0.0228)+
	(5,10)	0.0161(0.0044)	0.1001(0.0002)+	0.1008(0.0016)+	0.0784(0.0245)+	0.1111(0.0012)+	0.1188(0.0107)+
	(10,10)	0.0135(0.0034)	0.0991(0.0007)+	0.0987(0.0003)+	0.0768(0.0367)+	0.1070(0.0023)+	0.1287(0.0272)+
JY4	(10,5)	0.0631(0.0004)	0.0703(0.0001)+	0.0693(0.0001)+	0.0690(0.0003)+	0.3046(0.0350)+	0.1143(0.0022)+
	(5,5)	0.0617(0.0005)	0.0697(0.0002)+	0.0689(0.0001)+	0.0690(0.0001)+	0.4135(0.0148)+	0.1322(0.0059)+
	(5,10)	0.0638(0.0004)	0.0677(0.0001)+	0.0648(0.0001)≈	0.0665(0.0005)≈	0.2791(0.0182)+	0.1164(0.0126)+
	(10,10)	0.0650(0.0002)	0.0688(0.0002)≈	0.0662(0.0002)≈	0.0665(0.0002)≈	0.2134(0.0078)+	0.0834(0.0037)+
JY5	(10,5)	0.0088(0.0001)	0.0082(0.0002)-	0.0086(0.0153)≈	0.0185(0.0003)+	0.2129(0.0163)+	0.0325(0.0030)+
	(5,5)	0.0095(0.0003)	0.0087(0.0001)-	0.0088(0.0001)-	0.0189(0.0004)+	0.1037(0.0265)+	0.0355(0.0051)+
	(5,10)	0.0074(0.0001)	0.0072(0.0001)≈	0.0073(0.0001)≈	0.0101(0.0002)+	0.0866(0.0073)+	0.0199(0.0015)+
	(10,10)	0.0072(0.0008)	0.0070(0.0002)≈	0.0072(0.0001)≈	0.0099(0.0002)+	0.2098(0.0408)-	0.0183(0.0021)+
JY6	(10,5)	0.4981(0.0555)	0.7683(0.2305)+	1.2888(0.1813)+	2.0337(0.2023)+	2.9309(0.1072)+	2.0209(0.0813)+
	(5,5)	1.2010(0.1772)	1.7921(0.0977)+	2.0366(0.0671)+	2.5656(0.1670)+	3.6841(0.0481)+	2.6077(0.1800)+
	(5,10)	0.6598(0.0869)	1.0961(0.0252)+	1.1478(0.1417)+	1.2035(0.0854)+	2.4085(0.0481)+	2.1725(0.0874)+
	(10,10)	0.2849(0.0523)	0.0709(0.0251)-	0.5860(0.0408)-	0.4830(0.1970)+	1.8941(0.0671)+	1.5976(0.0462)+
JY7	(10,5)	0.9299(0.1895)	2.0947(0.2360)+	2.1380(0.2570)+	3.1130(0.9544)+	8.3470(0.2555)+	5.3513(0.3697)+
	(5,5)	0.6616(0.2297)	2.0939(0.3763)+	2.2270(0.2962)+	2.6165(0.3025)+	6.2693(0.4685)+	5.6526(0.6497)+
	(5,10)	0.8156(0.3413)	1.9093(0.2225)+	1.2515(0.1999)+	2.0159(0.6514)+	6.9805(0.3010)+	3.7794(0.3253)+
	(10,10)	1.3306(0.6101)	1.5058(0.3002)+	2.0049(0.5105)+	3.8106(0.6338)+	5.0254(0.4324)+	3.8062(0.4774)+
JY8	(10,5)	0.0235(0.0007)	0.0226(0.0012)≈	0.0236(0.0011)≈	0.0330(0.0027)+	0.0330(0.0039)+	0.0706(0.0016)+
	(5,5)	0.0254(0.0010)	0.0256(0.0012)≈	0.0258(0.0008)≈	0.0343(0.0008)+	0.0370(0.0036)+	0.1734(0.0158)+
	(5,10)	0.0233(0.0014)	0.0209(0.0006)≈	0.0220(0.0009)≈	0.0249(0.0017)+	0.0167(0.0015)+	0.0508(0.0066)+
	(10,10)	0.0206(0.0007)	0.0189(0.0007)-	0.0204(0.0012)≈	0.0219(0.0008)+	0.0153(0.0012)-	0.0257(0.0030)+

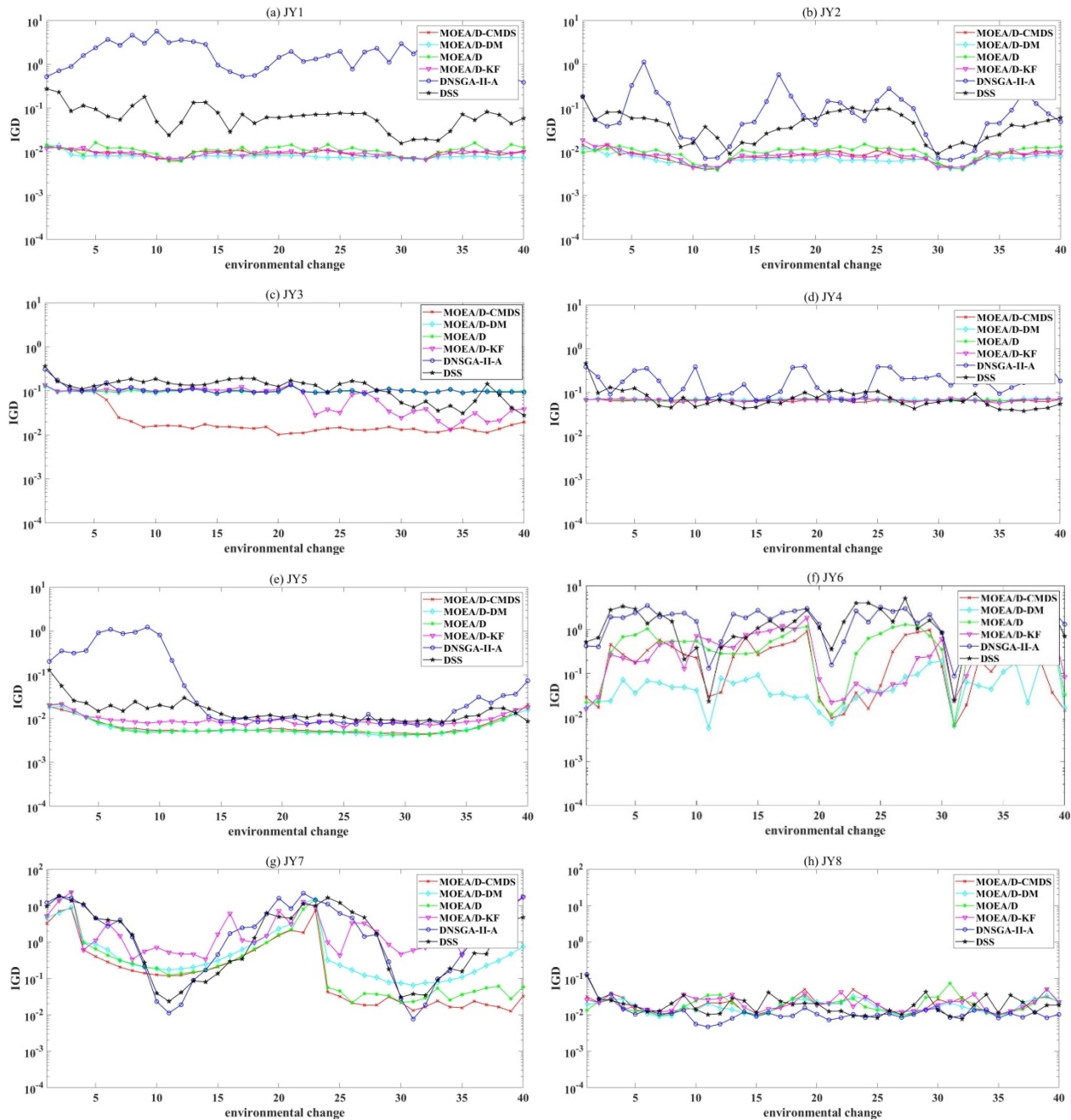


Figure 5. Tracking curves obtained by the six algorithms on JY test instances with $n_t = 10, \tau_t = 10$ over 20 runs.

7. Conclusions and future work

In this paper, a new prediction scheme named CMDS is first presented to adaptively select prediction strategy to respond new environment. When environment change happens, CMDS method firstly judges the type of change, which is translational or no-translational. Next, the way of generating individual is determined in order to provide initial individuals at next moment. As a result, CMDS method is integrated into the frame of MOEA/D. From the experiment results, MOEA/D/CMDS can show its superiority in tracking the changing PS or PF.

In the future research, we will be devoted to studying more efficient prediction mechanisms to handle more complex dynamic optimization problems, and further extend to large-scale practical problems.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant (No.61966030) and the Qinghai Provincial Key Laboratory of Internet of Things (No.2022-ZJ-Y21).

Conflict of interest

The authors declare that there are no conflicting interests.

References

1. Z. H. Zhang, Multi-objective optimization immune algorithm in dynamic environments and its application to green house controlling, *Appl. Soft Comput.*, **8** (2008), 959–971 <http://doi.org/10.1016/j.saoc.2007.07.005>
2. Z. Wang, G. F. Li, J. Ren, Dynamic path planning for unmanned surface vehicle in complex offshore areas based on hybrid algorithm, *Comput. Commun.*, **166** (2021), 49–56. <http://doi.org/10.1016/j.comcom.2020.11.012>
3. X. H. Wang, K. Z. Wang, S. Wu, S. Di, H. Jin, K. Yang, et al., Dynamic resource scheduling in mobile edge cloud with cloud radio access network, *IEEE Trans. Parallel Distrib. Syst.*, **29** (2018), 2429–2445. <http://doi.org/10.1109/TPDS.2018.2832124>
4. H. Lotfi, Multi-objective energy management approach in distribution grid integrated with energy storage units considering the demand response program, *Energy Res.*, **44** (2020), 10662–10681. <https://doi.org/10.1002/er.5709>
5. Y. Kuang, J. Sun, X. J. Gan, D. W. Gong, Z. P. Liu, M. M. Zha, Dynamic multi-objective cooperative coevolutionary scheduling for mobile underwater wireless sensor networks, *Comput. Ind. Eng.*, **156** (2021), 1–10. <https://doi.org/10.1016/j.cie.2021>
6. N. Shiono, H. Suzuki, Y. Saruwatari, A dynamic programming approach for the pipe network layout problem, *Eur. J. Oper. Res.*, **277** (2019), 52–61. <https://doi.org/10.1016/j.ejor.2019.02.036>
7. G. R. Feng, Y. Lan, X. P. Zhang, Z. X. Qian, Dynamic adjustment of hidden node parameters for extreme learning machine, *IEEE Trans. Cybern.*, **45** (2015), 279–288. <https://doi.org/10.1109/TCYB.2014.2325594>
8. S. B. Gee, K. C. Tan, H. A. Abbass, A benchmark test suite for dynamic evolutionary multi-objective optimization, *IEEE Trans. Cybern.*, **47** (2017), 461–472. <http://doi.org/10.1109/TCYB.2016.2519450>
9. S. Y. Jiang, S. X. Yang, Evolutionary dynamic multi-objective optimization: benchmarks and algorithm comparisons, *IEEE Trans. Cybern.*, **47** (2016), 198–211. <http://doi.org/10.1109/TCYB.20152510698>

10. S. Sahmoud, H. R. Topcuoglu, Sensor-based change detection schemes for dynamic multi-objective optimization problems, in *Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, (2016), 1–8. <http://doi.org/10.1109/SSCI.2016.7849963>
11. M. Rong, D. W. Gong, W. Pedrycz, L. Wang, A multi-model prediction method for dynamic multi-objective evolutionary optimization, *IEEE Trans. Evol. Comput.*, **99** (2019), 1–15. <http://doi.org/10.1109/TEVC.2019.2925358>
12. M. Greeff, A. P. Engelbrecht, Solving dynamic multi-objective problems with vector evaluated particle swarm optimization, *Evol. Comput.*, **29** (2008), 17–24. <http://doi.org/10.1109/CEC.20084631190>
13. R. W. Morrison, K. A. D. Jong, Triggered hypermutation revisited, in *Proceedings of the 2000 IEEE Congress on Evolutionary Computation CEC00 (Cat. No.00TH8512)*, **2** (2003), 1025–1032. <http://doi.org/10.1109/CEC.2000.870759>
14. C. R. B. Azevedo, A. F. R. Araújo, Generalized immigration schemes for dynamic evolutionary multi-objective optimization, in *Proceedings of the 2011 IEEE Congress on Evolutionary Computation (CEC)*, (2011), 2033–2040.
15. K. Deb, U. B. Rao, S. Karthik, Dynamic multi-objective optimization and decision making using modified NSGA-II: a case study on hydro-thermal power scheduling, in *Proceedings of International Conference on Evolutionary Multi-Criterion Optimization*, (2007), 803–817. http://doi.org/10.1007/978-3-540-70928-2_60
16. Y. G. Woldesenbet, G. G. Yen, Dynamic evolutionary algorithm with variable relocation, *IEEE Trans. Evol. Comput.*, **13** (2009), 500–513. <http://doi.org/10.1109/TEVC.2008.2009031>
17. Y. Chen, J. Zou, Y. Liu, S. X. Yang, J. H. Zheng, W. X. Huang, Combining a hybrid prediction strategy and a mutation strategy for dynamic multi-objective optimization, *Swarm Evol. Comput.*, **70** (2022), 1–16. <http://doi.org/10.1016/j.swevo.2022.101041>
18. M. Cámara, J. Ortega, F. D. Toro, A single front genetic algorithm for parallel multi-objective optimization in dynamic environments, *Neurocomputing*, **72** (2009), 3570–3579. <http://doi.org/10.1016/j.neucom.2008.12.041>
19. G. Ruan, G. Yu, J. H. Zheng, J. Zou, S. X. Yang, The effect of diversity maintenance on prediction in dynamic multi-objective optimization, *Appl. Soft Comput.*, **58** (2017), 631–647. <http://doi.org/10.1016/j.asoc.2017.05.008>
20. Z. P. Liang, S. X. Zheng, Z. X. Zhu, S. X. Yang, Hybrid of memory and prediction strategies for dynamic multi-objective optimization, *Inf. Sci.*, **485** (2019), 200–218. <http://doi.org/10.1016/j.ins.2019.01.066>
21. Z. Peng, J. H. Zheng, J. Zou, M. Liu, Novel prediction and memory strategies for dynamic multi-objective optimization, *Soft Comput.*, **19** (2015), 2633–2653. <http://doi.org/10.1007/s00500-014-1433-3>
22. Y. Wang, B. Li, Investigation of memory-based multi-objective optimization evolutionary algorithm in dynamic environment, in *Proceedings of the 2009 IEEE Congress on Evolutionary Computation (CEC)*, (2009), 630–637. <http://doi.org/10.1109/CEC.2009.4983004>
23. W. T. Koo, C. K. Goh, K. C. Tan, A predictive gradient strategy for multi-objective evolutionary algorithms in a fast changing environment, *Memet. Comput.*, **2** (2010), 87–110. <https://doi.org/10.1007/s12293-009-0026-7>

24. A. M. Zhou, Y. C. Jin, Q. F. Zhang, A population prediction strategy for evolutionary dynamic multi-objective optimization, *IEEE Trans. Cybern.*, **44** (2013), 40–53. <http://doi.org/10.1109/TCYB.2013.2245892>
25. J. Zou, Q. Y. Li, S. X. Yang, H. Bai, J. H. Zheng, A prediction strategy based on center points and knee points for evolutionary dynamic multi-objective optimization, *Appl. Soft Comput.*, **61** (2017), 806–818. <http://doi.org/10.1016/j.asoc.2017.08.004>
26. A. Muruganantham, Y. Zhao, S. B. Gee, X. Qiu, K. C. Tan, Dynamic multi-objective optimization using evolutionary algorithm with kalman filter, *Procedia Comput. Sci.*, **24** (2013), 66–75. <https://doi.org/10.1016/j.procs.2013.10.028>
27. M. Rong, D. W. Gong, Y. Zhang, Y. C. Jin, W. Pedrycz, Multidirectional prediction approach for dynamic multi-objective optimization problems, *IEEE Trans. Cybern.*, **49** (2019), 3362–3374. <https://doi.org/10.1109/tcyb.2018.2842158>
28. L. L. Cao, L. H. Xu, E. D. Goodman, H. Li, Decomposition-based evolutionary dynamic multi-objective optimization using a difference model, *Appl. Soft Comput.*, **76** (2019), 473–490. <https://doi.org/10.1016/j.asoc.2018.12.031>
29. J. H. Zheng, Y. B. Zhou, J. Zou, S. X. Yang, J. W. Ou, Y. R. Hu, A prediction strategy based on decision variable analysis for dynamic multi-objective optimization, *Swarm Evol. Comput.*, **60** (2021), 100786. <https://doi.org/10.1016/j.swevo.2020.100786>
30. X. X. Li, J. M. Yang, H. Sun, Z. Y. Hu, A. R. Cao, A dual prediction strategy with inverse model for evolutionary dynamic multi-objective optimization, *ISA Trans.*, **117** (2021), 196–209. <https://doi.org/10.1016/j.isatra.2021.01.053>
31. R. Rambabu, P. Vadakkepat, K. C. Tan, M. Jiang, A mixture-of-experts prediction framework for evolutionary dynamic multi-objective optimization, *IEEE Trans. Cybern.*, **50** (2020), 5099–5112. <https://doi.org/10.1109/TCYB.2019.2909806>
32. H. P. Xie, J. Zou, S. X. Yang, J. H. Zheng, J. W. Ou, Y. R. Hu, A decision variable classification-based cooperative coevolutionary algorithm for dynamic multi-objective optimization, *Inf. Sci.*, **560** (2021), 307–330. <https://doi.org/10.1016/j.ins.2021.01.021>
33. Z. P. Liang, T. C. Wu, X. L. Ma, Z. X. Zhu, S. X. Yang, A dynamic multi-objective evolutionary algorithm based on decision variable classification, *IEEE Trans. Cybern.*, **52** (2020), 1602–1615. <https://doi.org/10.1109/TCYB.2020.2986600>
34. G. Y. Chen, Y. N. Guo, M. Y. Huang, D. W. Gong, Z. K. Yu, A domain adaptation learning strategy for dynamic multi-objective optimization, *Inf. Sci.*, **606** (2022), 328–349. <https://doi.org/10.1016/j.ins.2022.05.050>
35. Q. Y. Zhang, S. X. Yang, Novel prediction strategies for dynamic multi-objective optimization, *IEEE Trans. Evol. Comput.*, **24** (2020), 260–274. <https://doi.org/10.1109/tevc.2019.2922834>
36. K. Zhang, C. N. Sheng, X. M. Liu, Multi-objective evolution strategy for dynamic multi-objective optimization, *IEEE Trans. Evol. Comput.*, **24** (2020), 974–988. <https://doi.org/10.1109/TEVC.2020.2985323>
37. K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II, *IEEE Trans. Evol. Comput.*, **6** (2002), 182–197. <https://doi.org/10.1109/4235.996017>
38. M. Farina, K. Deb, P. Amato, Dynamic multi-objective optimization problems: test cases, approximations, and applications, *IEEE Trans. Evol. Comput.*, **8** (2004), 425–442. https://doi.org/10.1007/3-540-36970-8_22

39. H. Richter, Detecting change in dynamic fitness landscapes, in *Proceedings of 2009 IEEE Congress on Evolutionary Computation (CEC)*, (2009), 1613–1620. <https://doi.org/10.1109/CEC.2009.4983135>
40. T. T. Nguyen, S. Yang, J. Branke, Evolutionary dynamic optimization: A survey of the state of the art, *Swarm Evol. Comput.*, **6** (2012), 1–24. <https://doi.org/10.1016/j.swevo.2012.05.001>
41. H. Li, Q. F. Zhang, Multi-objective optimization problems with complicated pareto sets, MOEA/D and NSGA-II, *IEEE Trans. Evol. Comput.*, **13** (2009), 284–302. <https://doi.org/10.1109/TEVC.2008.925798>
42. C. K. Goh, K. C. Tan, A competitive-cooperative co-evolutionary paradigm for dynamic multi-objective optimization, *IEEE Trans. Cybern.*, **13** (2009), 103–127. <https://doi.org/10.1109/TEVC.2008.920671>
43. Y. Wu, Y. C. Jin, X. X. Liu, A directed search strategy for evolutionary dynamic multi-objective optimization, *Soft Comput.*, **19** (2015), 3221–3235. <https://doi.org/10.1007/s00500-014-1477-4>
44. C. F. Wang, G. G. Yen, M. Jiang, A grey prediction-based evolutionary algorithm for dynamic multi-objective optimization, *Swarm Evol. Comput.*, **56** (2020), 100695. <https://doi.org/10.1016/j.swevo.2020.100695>
45. C. Rossi, M. Abderrahim, J. C. Diaz, Tracking moving optima using kalman-based predictions, *Evol. Comput.*, **16** (2008), 1–30. <https://doi.org/10.1162/evco.2008.16.1.1>



AIMS Press

©2024 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)