

Integration of improved predictive model and adaptive differential evolution based dynamic multi-objective evolutionary optimization algorithm

Ruochen Liu · Jing Fan · Licheng Jiao

Published online: 22 January 2015
© Springer Science+Business Media New York 2015

Abstract A novel dynamic multi-objective optimization evolutionary algorithm is proposed in this paper to track the Pareto-optimal set of time-changing multi-objective optimization problems. In the proposed algorithm, to initialize the new population when a change is detected, a modified prediction model utilizing the historical optimal sets obtained in the last two times is adopted. Meantime, to improve both convergence and diversity, a self-adaptive differential evolution crossover operator is used. We conducted two experiments: the first one compares the proposed algorithm with the other three dynamic multiobjective evolutionary algorithms, and the second one investigates the performance of the two proposed operators. The statistical results indicate that the proposed algorithm has better convergence speed and diversity and it is very promising for dealing with dynamic environment.

Keywords Dynamic multiobjective optimization · Forecasting model · Self adaptive differential evolution

1 Introduction

Many real-world industrial applications and scientific research present a time-dependent feature. The dynamic multi-objective optimization problem (DMOP) is characterized in that either the objective function and constraint

or the associated parameters or both change over time with respect to static multi-objective optimization problem [1]. The difficulties of dealing with a DMOP lie in how to converge to the new true Pareto-optimal front (PF) rapidly and find a widely distributed set of the Pareto-optimal solutions (PS) when an environment change comes. A great success has been achieved since the evolutionary algorithms (EAs) were applied to the multi-objective optimization and more approximate optimal solutions widely distributed can be generated by the multi-objective optimization evolutionary algorithms (MOEAs) [2] compared to the traditional multi-objective optimization algorithm. So the dynamic multi-objective optimization evolutionary algorithm (DMOEA) would be promising for the multi-objective optimization in dynamic environments.

The Dynamic Orthogonal Multi-Objective Evolutionary Algorithm (DOMOEA) [3] applies an orthogonal design method proposed in the previous MOEA to enhance the fitness of the population during the static stages between two successive changes of environment. Goh et al. [4] proposed a new coevolutionary strategy that competitive and cooperative mechanisms are combined to solve MOPs and extended the strategy to deal with DMOPs. In this algorithm, through the iterative process of competition and cooperation, the subcomponents are optimized by different species subpopulations based on the optimization target of the particular time, which enabling the coevolutionary algorithm to handle both the static and dynamic multiobjective optimization problems. In the work of Wang et al. [5], during the static stages the static multi-objective optimization problem is changed into a two-objective optimization problem that deal with two re-defined objectives. Simultaneously, a new crossover operator and mutation operator

R. Liu (✉) · J. Fan · L. Jiao
Key Laboratory of Intelligent Perception and Image
Understanding of Ministry of Education of China,
Xidian University, Xi'an, 710071, China
e-mail: ruochenliu@xidian.edu.cn

that are suitable for the changing environment are adopted in the algorithm. Koo et al. [6] presented a new method to detect the environment changes, a new prediction strategy known as the predictive gradient, and a new memory technique suitable for solving DMOPs in a fast changing environment. The predictive strategy is easy to implement and solutions updated through the predictive gradient will remain in the vicinity of the new Pareto-optimal Set and pull the rest of the population to converge.

Artificial Immune Systems (AIS) inspired by the human immune system (HIS) have exhibited its advantages in terms of immune memory, diversity, robustness and other aspects. Since Coello Coello proposed an artificial immune system algorithm MISA [7] based on the clonal selection principle to handle MOPs, some successful multi-objective optimization immune algorithms have been proposed afterwards [8–10]. And the experiment results show that such kind of optimization techniques is one of the most promising strategies for dealing with the multi-objective optimization problem. Moreover, dynamic multi-objective optimization immune algorithms (DMOIA) have also attracted a lot of attention, but few works have been reported at present. Shang et al. [11] proposed a clone selection algorithm (CSADMO) based on nonuniform mutation strategy and distance method to solve two DMOPs. The nonuniform mutation strategy makes the algorithm search in a large range in the early evolution and implement the local search in the latter stage. Zhang [12, 13] researched immune-based optimization techniques for DMOPs in which the dimension of the variable or objective space may be time-varying. And the algorithms has been successfully applied to greenhouse control and signal simulation. A hybrid dynamic multi-objective immune algorithm using the linear prediction model (HDMIO) was proposed to deal with four DMOPs by Ma et al. [14]. Zhang et al. [15] developed an artificial immune system to solve time-varying non-linear constrained multi-objective problems with changing variable dimensions. In this algorithm, T-module, B-module, and M-module are used to create an initial population by using the history information when the environment changes, search for the desired non-dominated front of a given environment, and store temporarily the non-dominated solutions to help create initial populations for the next environment, respectively.

In this paper, a well-known static multi-objective immune algorithm with non-dominated neighbor-based selection (NNIA) [10], which has been proved to be an effective algorithm for sloving static MOPs, is extended to solve DMOPs. Due to the fact that NNIA pays more attention to the less-crowded regions in the current trade-off front in the process of crossover, the diversity of the solutions obtained by NNIA may not be ideal. To remedy this, we use an improved simple self adaptive differential evolution

crossover operator to improve the diversity of the solution in this paper. In our new dynamic MOEA (DMOEA), firstly, in the decision space, to take individuals closer to the PS at the beginning of a new environment, we propose a modified prediction model to take full advantage of search informations in the past two times. Secondly, in the objective space, to improve the convergence speed and the diversity of the population, we propose an improved adaptive differential evolution crossover operator to help the population evolution.

This paper is structured as following: Section 2 presents some related theoretical background including dynamic multiobjective optimization, forward-looking forecasting model, the original static multi-objective immune algorithm used in this paper, differential evolution crossover operator and several important terms throughout the paper. Section 3 presents the details of the proposed dynamic immune inspired multi-objective algorithm. Section 4 introduces the test instances and performance metics, and then carried out experiments to evaluate the effectiveness of our algorithm. Finally, Section 5 outlines some conclusions and proposes a few future research directions.

2 Related theoretical background

2.1 Dynamic multi-objective optimization

In the case of time changing, the objective functions and the constrained functions may be related to the time, so a static multi-objective problem extends to the following expression:

$$\begin{cases} \min_{x \in X} F(x, t) = (f_1(x, t), f_2(x, t), \dots, f_m(x, t))^T \\ s.t. g_i(x, t) \leq 0 (i = 1, 2, \dots, p), h_j(x, t) = 0 (j = 1, 2, \dots, q) \end{cases} \quad (1)$$

This is a DMOP, where t represents time. $x = (x_1, \dots, x_l) \in \Omega$ is the decision vector, Ω is the feasible region in decision space, $F(x, t)$ represents the set of m objective functions with respect to t , $g_i(x, t)$ and $h_j(x, t)$ represent the inequality and equality constraints, respectively.

When considering minimization problem, a vector x_1 is said to dominate vector x_2 (denoted as $x_1 < x_2$) if and only if x_1 is partially less than x_2 , ie,

$$\forall i \in \{i, \dots, k\}, x_1 \leq x_2 \wedge \exists i \in \{i, \dots, k\}, x_1 < x_2 \quad (2)$$

For a given DMOP $F(x, t)$, at a certain moment, the Pareto-optimal set PS_t in the decision space is defined as:

$$PS_t = \{x \in \Omega | \neg \exists x^* \in \Omega, F(x^*, t) < F(x, t)\} \quad (3)$$

That means the Pareto-optimal set in the decision space consists of the decision vectors that can not be dominated by any other decision vectors.

Accordingly, for a given DMOP $F(x, t)$ and Pareto-optimal set PS_t , at a certain moment, the Pareto-optimal front PF_t in the objective space is defined as:

$$PF_t = \{F(x, t) = (f_1(x, t), f_2(x, t), \dots, f_m(x, t))^T | x \in PS\} \tag{4}$$

According to the PS_t and the PF_t whether change over time, Farina et al. [1] divided the DMOPs into four types:

- Type I : The PS_t changes over time, the PF_t does not change.
- Type II : Both the PS_t and the PF_t change over time.
- Type III : The PS_t does not change over time, the PF_t change over time,.
- Type IV : Both the PS_t and the PF_t do not change over time.

2.2 Summary of related terms

We describe the nomenclature of immunology and the terms throughout the paper in Table 1.

2.2.1 Antibody and antibody population

To put it simply, as we use the real-valued presentation like the NNIA, an antibody $\mathbf{b}_i = (b_{i,1}, b_{i,2}, \dots, b_{i,l})$ is the coding of a decision variable x in the dynamic multi-objective optimization. So the antibody population is $B = \{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n\}$, $\mathbf{b}_i \in \Omega$, $1 \leq i \leq n$.

2.2.2 Dominant population, active population, and clone population

Based on the definition of domination in the 2.1, a dominant antibody represents an antibody which cannot be dominated by any other antibody, that is to say it is a good solution. The

Table 1 The nomenclature of immunology corresponding to problems

nomenclature of immunology	the description corresponding to problems
Antibody	The coding of a decision variable x
Dominant antibody	The non-dominated solution
Active antibody	The non-dominated solution with a greater crowding-distance
Proportional Cloning	Copy the active population
Recombination	Crossover the clone population
Hypermutation	Mutate crossover solution

dominant population is denoted as D , a set of all dominant antibodies.

An active antibody represents a dominant antibody with greater crowding-distance [16] in the approximate Pareto-optimal front. Corresponding with the concept in artificial immune system, an active antibody means a competitive individual or a better solution in the dynamic multi-objective optimization. The active population is denoted as A , consisting of active antibodies. So during the evolution, the antibodies in the active population are more likely to enter the next generation.

During the evolution, copy the antibodies in the active population into the clone population, which is denoted as C , and carry out the genetic operation on C . So more active antibodies with greater crowding-distance will be evolved by the cloning operation.

2.2.3 Proportional cloning, recombination, and hypermutation

According to the interpretation of the terms above, the genetic operators corresponding to immunology will be easy to understand. The operation detail of the proportional cloning operator T^C acts on the active population $A = (a_1, a_2, \dots, a_{|A|})$ ($|A|$ is the size of A) is as following:

$$C = T^C(A) = T^C(a_1, a_2, \dots, a_{|A|}) = T^C(a_1) + T^C(a_2) + \dots + T^C(a_{|A|}) = \{a_1^1, a_1^2, \dots, a_1^{q_1}\} + \{a_2^1, a_2^2, \dots, a_2^{q_2}\} + \dots + \{a_{|A|}^1, a_{|A|}^2, \dots, a_{|A|}^{q_{|A|}}\} \tag{5}$$

where q_i is the cloning number of the active antibody a_i . q_i is a self-adaptive value depends on the crowding-distance d_i of a_i , it is calculated as following:

$$q_i = \left\lceil n_C \times \frac{d_i}{\sum_{j=1}^{|A|} d_j} \right\rceil \tag{6}$$

where n_C is an expectant value of the size of the clone population that is usually set as same as the initial population size.

And then the recombination operator T^R acts on the clone population $C = (c_1, c_2, \dots, c_{|C|})$ ($|C|$ is the size of C) is as following:

$$R = T^R(C) = T^R(c_1, c_2, \dots, c_{|C|}) = T^R(c_1) + T^R(c_2) + \dots + T^R(c_{|C|}) = crossover(c_1) + crossover(c_2) + \dots + crossover(c_{|C|}) \tag{7}$$

And the hypermutation operator T^R acts on the population $R = (r_1, r_2, \dots, r_{|R|})$ ($|R|$ is the size of R) is as following:

$$\begin{aligned}
 H &= T^H(R) = T^H(r_1, r_2, \dots, r_{|R|}) = T^H(r_1) + T^H(r_2) \\
 &\quad + \dots + T^H(r_{|R|}) \\
 &= mutate(r_1) + mutate(r_2) + \dots + mutate(r_{|R|}) \tag{8}
 \end{aligned}$$

2.3 The used static MOEA

We describe the traditional static multi-objective immune algorithm used in this paper, the NNIA. Firstly, at a certain moment of t when the DMOPs is not changing, the dominant population, active population and clone population at the g -th evolution generation are denoted as D_g, A_g, C_g , respectively. In NNIA, realize the recombination and the

hypermutation with the simulated binary crossover (SBX) operator and polynomial mutation respectively. The main loop of NNIA is shown in Table 2.

2.4 Differential evolution crossover operator

DE is a promising evolutionary optimization method proposed by Storn and Price [17]. It introduces new parameter vectors by adding a weighted difference vector between two members in the population to the third member. The principle is simple and the process can be depicted as Fig. 2.

As shown in the figure above, at the g -th generation, three different individuals are randomly selected from the population for the i -th individual $X_{i,g}$ to be crossed over. And then a new intermediate individual $V_{i,g}$ is produced. With the crossover probability CR , $X_{i,g}$ and $V_{i,g}$, the offspring $U_{i,g+1}$ is calculated as following:

$$U_{i,g+1} = \begin{cases} V_{i,g} & \text{if } (rand(0, 1) \leq CR) \text{ or } j = j_{rand} \\ X_{i,g} & \text{if } (rand(0, 1) > CR) \text{ or } j \neq j_{rand} \end{cases} \quad i = 1, 2, \dots, N, j = 1, 2, \dots, D \tag{9}$$

Owing to the good performance of DE in solving single-objective optimization problems [17], some researchers have tried to extend it to MOPs. Abbas was the first to apply DE to MOPs in the Pareto Differential Evolution algorithm [18]. Many new Multiobjective Differential Evolution Algorithm being continuously has been published since then [19, 20].

2.5 The forward-looking predictive model

In DMOEA, a DMOP is usually considered as a continuous combination of unchanged static problems in a period of time, during which a static MOEA is used to guide the evolution. The search information of the optimal solutions in the last few moments might be useful for producing initial population in the new environment. The original and simplest framework is to save and deal with optimal solutions of the last time. Deb et al. [21] proposed two re-initialization techniques to replace a portion of the optimal solutions in the last time: randomly generate new individuals (DNSGAI-A) or generate mutated solutions of existing solutions (DNSGAI-B). The hypermutation operator and randomly generate individuals strategy are combined by Zheng [22]. Wei et al. [23] proposed a hyper rectangle search to predict $m + 2$ initial solutions (m is the number of objectives) for the next environment.

The other frameworks are based on the search information of the past several times and the information is saved to help the search process in the new environment. If the

change of the optimal solutions' location that obtained in the history moment presents some kind of fixed trend in the dynamic environment, it seems important to store the optimal solutions in the past several moments to generate a forward-looking model, which is used for accelerating the convergence speed. But if the changing trend is not obvious or stable, the predictive model will go wrong either. The forward-looking predictive model based on memory storage is firstly proposed by Hatzakis and Wallace in [24] to improve the performance of DMOEA. This predictive model utilizes the optimal solution sets obtained in the past t moments, that is from 0 moment to $t - 1$ moment, to produce the initial population of the next moment, that is the t moment. This model in the two-dimensional space is shown in Fig. 3.

Since then, some researchers have also proposed new predictive models. Zhou et al. [25] proposed a linear predictive model and four re-initialization methods were proposed with this model. The initial population at the new moment obtained by the four re-initialization method is near the PS location and converge to the true PS more quickly. Koo et al. [6] proposed a new predictive gradient strategy which is suitable for solving DMOPs in a rapidly changing environment.

As Hatzakis and Wallace mentioned [24], the design of predictive model is very important and it directly affects the accuracy of the initial solutions at the new moment. But in most cases, the design depends on the feature of the dynamic multi-objective problem. The fact that the only

Table 2 The pseudo code of NNIA

Input : G_{\max} (maximum number of evolution generations), n_D (maximum size of dominant population), n_A (maximum size of active population), n_C (size of clone population)
 Output: $D_{G_{\max}+1}$ (final approximate Pareto-optimal set)

- Step1: Initialization : Random generate an initial antibody population B_0 with the population size of n_D . Create the initial $D_0 = \Phi$, $A_0 = \Phi$, and $C_0 = \Phi$. Set $g = 0$.
- Step2: Update Dominant Population: Identify the dominant antibodies in B_g . Copy all the dominant antibodies into the temporary dominant population(denoted by DT_{g+1}). If the size of DT_{g+1} is not bigger than n_D , let $D_{g+1} = DT_{g+1}$. Otherwise, calculate the crowding-distance values of all individuals in DT_{g+1} and sort the values in descending order, and put the first n_D individuals into D_{g+1} finally.
- Step3: Termination : If meet $g \geq G_{\max}$, output D_{g+1} as the solution of the algorithm, Stop. Otherwise, let $g = g + 1$, turn to Step 4.
- Step4: Nondominated Neighbor Based Selection: If the size of D_g is not bigger than n_A , let $A_g = D_g$. Otherwise, calculate the crowding-distance values of all individuals in D_g and sort the values in descending order, and put the first n_A individuals into A_g .
- Step5: Proportional Cloning: Determine the cloning number of every individual in A_g , and thus get the clone population C_g
- Step6: Recombination and Hypermutation: Apply recombination and hypermutation operation on C_g and denote the resulting population as C_g^*
- Step7: Get the antibody population B_g by combining the C_g^* and D_g turn to Step2.

available information is the optimal solutions obtained in the past moments makes it difficult to design a commonly used predictive model that can be used in different types of problems.

3 Description of the algorithm

In this section, we describe the main operators used in our dynamic immune inspired multi-objective algorithm and the whole flow of the proposed algorithm.

3.1 Environment change detection and modified prediction mechanism

3.1.1 Environment change detection

The detection operator is used to detect whether the environment has changed, or whether the change is big enough to regard it as a environment change rather than a noise. In this paper, the environment change detection operator [1] is calculated by the objective function values of the several individuals at two consecutive moment. It can be described

Fig. 1 The flowchart of NNIA

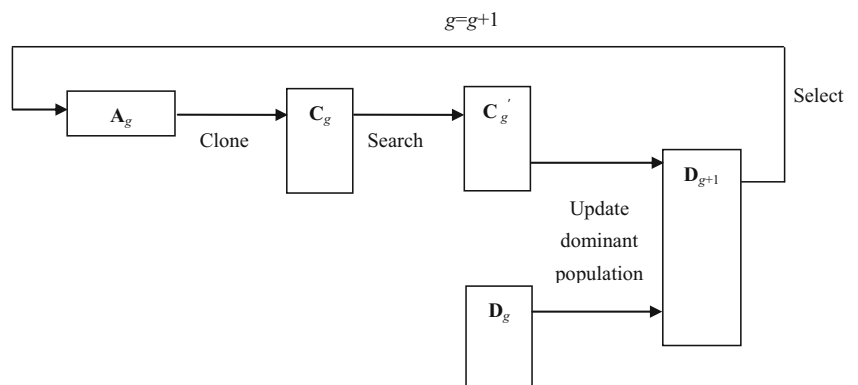
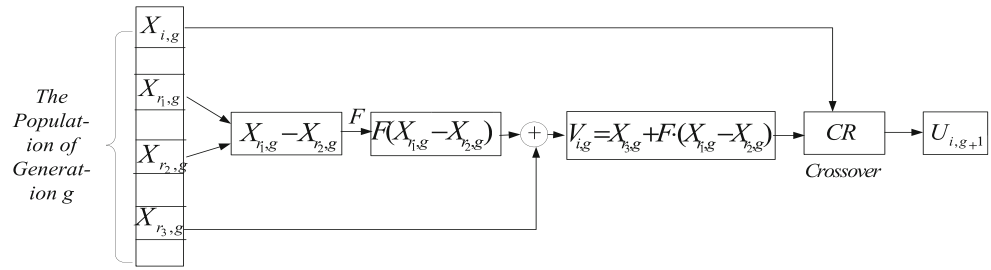


Fig. 2 Original DE Model



as following:

$$\varepsilon(t) = \frac{\sum_{j=1}^{n_\varepsilon} \left\| \frac{(f_j(X,t) - f_j(X,t-1))}{R(t) - U(t)} \right\|}{n_\varepsilon} \tag{10}$$

where $R(t) = (r_1, r_2, \dots, r_m)^T$ and $U(t) = (u_1, u_2, \dots, u_m)^T$ are the maximum vector and minimum vector of the objective function value at the t moment, respectively. m is the number of objective functions. Variable r_i is the maximum value of the i -th objective function in the population and variable u_i is the minimum value of the i -th objective function in the population. n_ε is the number of individuals chosen to detect the environment change. If the detective value $\varepsilon(t)$ is greater than a predefined value $\tilde{\varepsilon}$, it means that a significant change has taken place, and then the modified predictive model is used to predict the initial population. In this paper, n_ε and $\tilde{\varepsilon}$ are set to 5 and $2e-2$, respectively.

3.1.2 Modified prediction mechanism

When the environment changes, we use an improved predictive model to predict the initial population. The improved predictive model is based on the model proposed by Zhou et al. [25]. In [25], it is assumed that the PS at the past two moments is Q_{t-1} and Q_{t-2} , respectively, which are used to

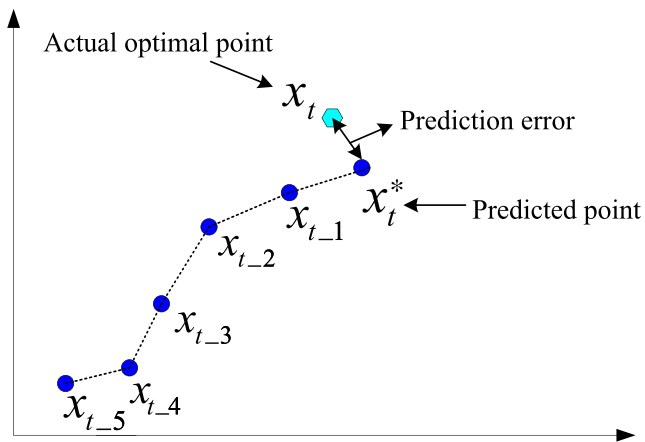


Fig. 3 The forward-looking forecasting model

predict the initial population Q_t at the t moment. The whole predictive model is inspired by the forward-looking predictive model in the Section 2.5 and it is shown in Fig. 4 (left).

The predictive model in [25] is described as following:

$$x_t = x_{t-1} + (x_{t-1} - x_{t-2}) + \sigma \tag{11}$$

where x_{t-1} is an individual in Q_{t-1} , and x_{t-2} is the individual in Q_{t-2} and it is the nearest individual to x_{t-1} according to the Euclidean distance:

$$x_{t-2} = \arg \min_{y \in Q_{t-2}} \|y - x_{t-1}\|_2 \tag{12}$$

That means the new initial individuals at the new moment depend on the shortest distance between the two points at the previous two moments and the predictive model is linear. Then we add a Gaussian noise $\sigma \sim N(0, I\delta)$ to the predicted locations to increase the possibility of the new initial individuals falling on the true PS_t at the new moment. Where I is an identity matrix and δ is the standard deviation, which is calculated by the Euclidean distance of x_{t-1} and x_{t-2} as follows:

$$\delta^2 = \frac{1}{4n} \|x_{t-1} - x_{t-2}\|_2^2 \tag{13}$$

where n is the dimensions of the decision variables.

Although the predictive model proposed above can be commonly used and is particularly applicable for the problems with PS_t linear changing over time. As demonstrated in [25], when the PS_t is nonlinear changing over time, its performance is poor. Therefore, to make the predictive model more generic in more types of DMOPs, we propose a modified predictive model and it is shown in Fig. 4 (right).

In the modified predictive model, half of the individuals are randomly selected to use in the linear formula (11), the remaining individuals are used in the new model. The new model expression is as following:

$$x_t = \begin{cases} x_{t-1} + (x_{t-1} - x_{t-2}) + \sigma & \text{if } rand() < 0.5 \\ x_{t-1} + rand() * (x_{t-1} - x_{t-2}^{r1}) + rand() * (x_{t-1} - x_{t-2}^{r2}) & \text{otherwise} \end{cases} \tag{14}$$

where x_{t-2}^{r1} and x_{t-2}^{r2} are two individuals randomly selected from Q_{t-2} , $rand()$ returns an uniform random number in $[0, 1]$. Selecting two individuals to guide the new predictive

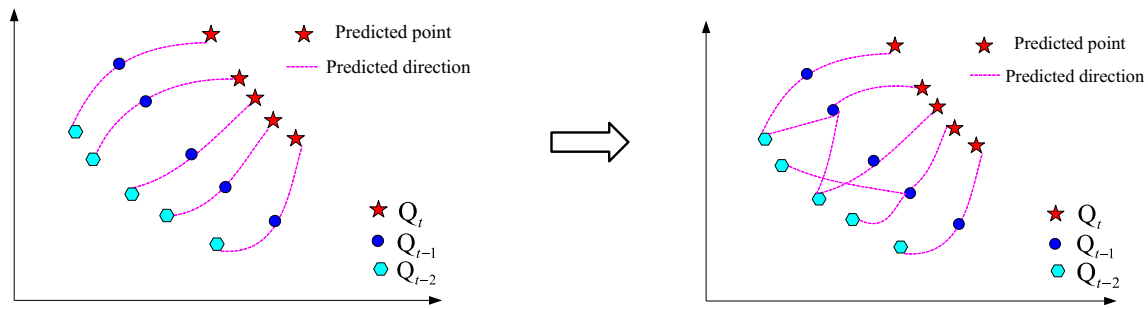


Fig. 4 Original Predictive Model and Modified Predictive Model

individual can prevent the new individual moving toward the wrong direction when it is only guided by one individual, just like a PSO based crossover operator. Meanwhile, in the new model, only half of the individuals are calculated by distance, which means the computational complexity of the new model will reduce. It can be observed from the statistical experimental results in Section 4 that the modified predictive performs better than the original predictive model and it can obtain better results.

As the predictive model is based on the history search information, the model can not work at the early evolution with deficient information. We use a simple Gaussian interference to generate the initial population. So the predictive strategy is showed as following:

$$Q_t = \begin{cases} \text{Perturbing 20 \% of } Q_{t-1} \text{ with Gauss noise,} & \text{if } 0 < t < 3 \\ \text{Using the modified prediction model} & \text{otherwise} \end{cases} \quad (15)$$

3.2 Modified self-adaptive differential evolution crossover operator

The parameter F is important in the DE crossover operator. In the most literatures, there is no discussion about the influence of parameter F on the performance of multi-objective differential evolution algorithm, but take the value between 0.5 and 0.9 according to the experience. Qian et al. [26] proposed an adaptive method to make up for the defect above, the value of F is depend on the crowding distance in different Pareto levels, the sizes of the dominant population and the size of the whole population. This adaptive method makes the algorithm focus on the global search to find the feasible region in the early stage of evolution, and focus on the local search to speed up the coverage speed in the late stage of evolution. The F is calculated as following:

$$F = \max\left(\frac{\sum_{j=1}^k \sum_{i=1}^{m_j} |d_{ij} - \bar{d}_j| + df}{\sum |Q| \cdot \bar{d} + df}, 1 - \frac{2|P|}{|Q|}, l_{\min}\right) \quad (16)$$

where d_{ij} is the crowding distance of the i -th solution in the j -th Pareto level; \bar{d}_j is the average value of crowding

distances of the all solutions in the j -th Pareto level; \bar{d} is the average value of crowding distance of the all solutions in every iteration; $|P|$ is the number of the dominant solutions; $|Q|$ is the number of the population set; the parameter df represents the Euclidean distance between two boundary solutions in Q ; l_{\min} is the minimum value for F . The results in [26] showed that the convergence and the diversity of the algorithm with this differential evolution crossover operator are significantly improved when dealing with different static multi-objective problems.

In our algorithm, because the DE crossover operator is implemented in the non-dominated population consisting of individuals with greater crowding-distance, we can ignore the influence of boundary solutions on F and propose a more simple and effective model of F . The new F is calculated as following:

$$F = \frac{\sum_{i=1}^k |d_i - \bar{d}|}{\sum k \cdot \bar{d}} \quad (17)$$

where d_i is the crowding distance of the i -th antibody in the population, \bar{d} is the average value of crowding distances of the all antibodies in the population, k is the size of the population. It can be observed from the comparative trial results in Section 4 that the modified self-adaptive differential evolution crossover operator can achieve better results compared with the original self-adaptive differential evolution crossover operator.

3.3 The whole flow

In this paper, unlike the common strategy for dealing with the DMOP, at the beginning of a new moment, an environment change detection operator is used to detect whether the problem has really changed or not. Then the algorithm uses the modified forecasting model to get the new initial population. And finally, NNIA based on the modified self-adaptive differential evolution crossover is used to solve the static DMOPs.

The whole flow of the proposed algorithm is shown in Table 3:

Table 3 The pseudo code of PDNNIA

Algorithm: PDNNIA
Input: $\tau_T, n_D, n_A, n_C, T_{max}$
Output : the PSs obtained at every moment, that is, $Q_1, \dots, Q_{T_{max}}$
Initialization: Randomly generate an initial antibody population $B_g(0)$ with size n_D , and get the dominant population $D_g(0)$, select the first n_A individuals with greater crowding-distance to form the active population $A_g(0)$, Set $t = 0$;
while $t < T_{max}$ do
Set $g = 0$;
if $t > 0$, do
Apply Environment Change Detection and new re-initialization strategy to get $B_g(t)$, then apply Non-dominated Sorting and Crowding-Distance Sorting on $B_g(t)$ to get $D_g(t)$ and $A_g(t)$;
end if
while $g < \tau_T$, do
Apply Proportional Cloning to $A_g(t)$ to get the clone population $C_g(t)$;
Apply new DE strategy and polynomial mutation on $C_g(t)$ to get $C'_g(t)$;
Combine $C'_g(t)$ and $D_g(t)$ to get the antibody population $B_{g+1}(t)$ of the next generation;
Get the dominant population $D_{g+1}(t)$ and the active population $A_{g+1}(t)$ of the next generation from $B_{g+1}(t)$;
$g = g + 1$;
end while
$Q_t = D_g(t)$;
$t = t + 1$;
end while

Where g is the generation counter, τ_T is the total number of generations at t moment. T_{max} is the maximum number of moments, n_D is maximum size of dominant Population. $A_g(t)$ is Active Population with the maximum size of n_A . $C_g(t)$ is Clone Population with the maximum size of n_C . In PDNNIA, as the general method does, we assume that a dynamic problem remains unchanged within the t moment. That is, during the entire period of t moment, there is no change in the environment and thus a DMOP can be treated as a static MOP which can be solved by the modified NNIA which is improved with the modified self-adaptive differential evolution crossover operator.

The flowchart is shown in Fig. 5.

At the beginning of time t , Environment Change Detection and Modified Prediction Mechanism are applied to detect environment change and produce the initial population. After obtaining the dominant population $D_g(t)$ and the active population $A_g(t)$, evolution operators including proportional cloning, Modified Self-adaptive DE operator and polynomial mutation are used to guide the population to evolve until $metg < \tau_T$. In the process of forming the non-dominated population $D_g(t)$, if the number of non-dominated antibodies is greater than the maximum limitation n_D , the algorithm uses the crowding-distance sorting strategy [16] to select n_D antibodies with greater crowding-distance. Likewise, in the process of forming the active population $A_g(t)$, if the size of non-dominated population

n_D is greater than the limitation size of active population n_A , the algorithm applies the crowding-distance sorting strategy as above.

4 Experimental results

In this section, we introduce six DMOPs and three metrics to test this algorithm and evaluate its performance. As a comparison, we test three other DMOEAs. And finally, we conducted two sets of comparative experiments to test the performance of the two mechanism, that is, the modified predictive model and adaptive differential evolution operator.

4.1 Test problems

In the introduction of the DMOP in Section 2.1, we do not introduce the time t in detail. In fact, the time t is controlled by the formula as following:

$$t = \frac{1}{n_T} \left\lfloor \frac{\tau}{\tau_T} \right\rfloor \tag{18}$$

where n_T and τ_T are the severity and frequency of the environment change, respectively. τ represents the current iterative counter or the current function evaluation counter and is added from zero to τ_T . So τ_T also corresponds to the total number of iteration or function evaluation when the environment remains unchanged. As this formula describes, a large value of n_T will cause a small severity of change. Likewise, a large value of τ_T will make the frequency of change slow.

Six different DMOPs are tested in this paper. The first four problems are DMOP1(FDA1), DMOP2(FDA3) [1], DMOP3(dMOP1), DMOP4(dMOP2) [4] and they are two objective functions. The last two problems are DMOP5(FDA4), DMOP6(FDA5) [1] and they are three objective functions. The details of all the six test problems are shown in Table 4.

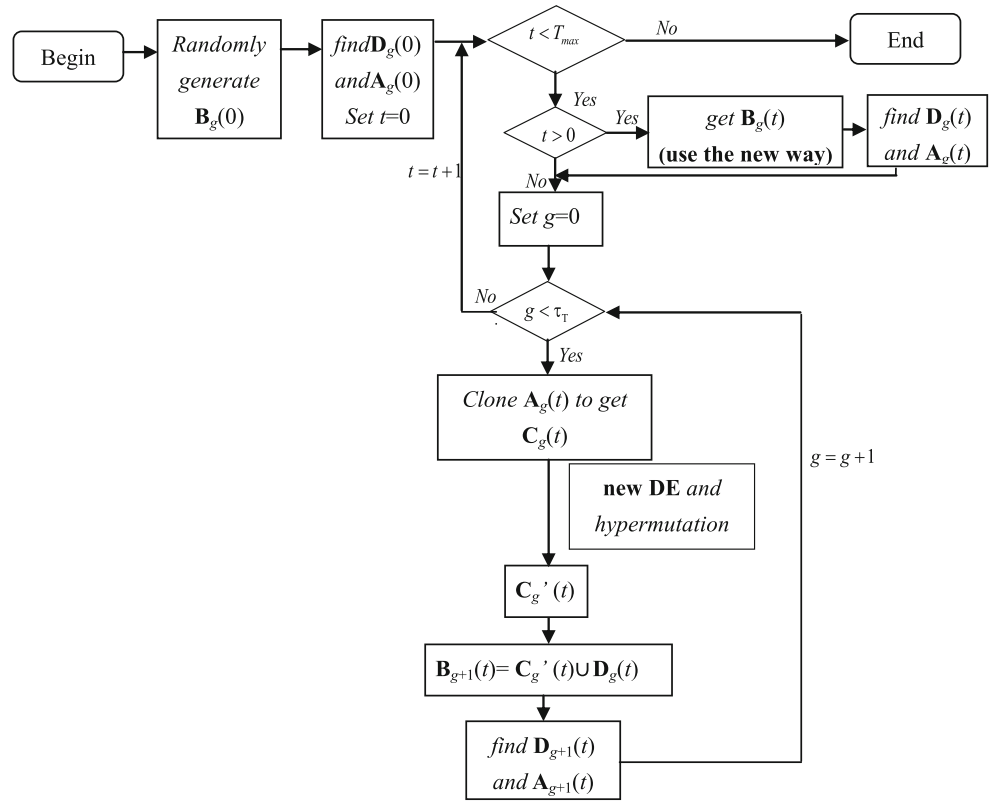
4.2 Performance metrics

The Generational Distance (GD) [27] and the Spacing (SP)[28] are used for measuring the convergence and the distribution of solutions obtained by four algorithms, respectively, and the lower value of GD or SP represents the better performance. The two metrics are summarized as follows.

Convergence Metric: The Generational Distance (GD) represents the convergence of the solutions. GD_t is calculated at the last generation of each t moment and is defined as following:

$$GD_t(A_t, P F_t) = \frac{\sum_{v \in A_t} d(v, P F_t)}{|A_t|} \tag{19}$$

Fig. 5 The flowchart of PDNNIA



So the mean convergence indicators \overline{GD} is expressed as follows:

$$\overline{GD} = \frac{\sum_{t=1}^{T_{max}} GD_t}{T_{max}} \tag{20}$$

Let PF_t be a set of uniformly distributed points on the true Pareto-optimal front at the t moment, and A_t be the approximate Pareto-optimal set of dominant antibodies obtained by DMOEA at the t moment. Where $d(v, PF_t)$ is the minimum Euclidean distance among all the distances between each vector v in A_t and every point in PF_t , T_{max} is the maximum number of environmental changes in a run.

Distribution Metric: The Spacing (SP) represents the distribution of the solutions. SP_t is calculated at the last generation of each t moment and is defined as following:

$$SP_t = \sqrt{\frac{1}{|A_t| - 1} \sum_{i=1}^{|A_t|} (\bar{d} - d_i)^2} \tag{21}$$

where

$$d_i = \min_j \left\{ \sum_{m=1}^k |f_m(a_i) - f_m(a_j)| \right\} \tag{22}$$

($a_i, a_j \in A_t; i, j \in \{1, 2, \dots, |A_t|\}$), $\bar{d} = \frac{1}{|A_t|} \sum_{i=1}^{|A_t|} d_i$)

So the mean Spacing \overline{SP} is expressed as follows:

$$\overline{SP} = \frac{\sum_{t=1}^{T_{max}} SP_t}{T_{max}} \tag{23}$$

Let PF_t be a set of uniformly distributed points on the true Pareto-optimal front at the t moment, and A_t be the approximate Pareto-optimal set of dominant antibodies obtained by DMOEA at the t moment. Where $d(v, PF_t)$ is the minimum Euclidean distance among all the distances between each vector v in A_t and every point in PF_t , \bar{d} is the average value of all d_i , and k is the number of the objective functions.

4.3 Comparison of different DMOEAs

The other three algorithms that are used to be compared with PDNNIA are DNSGAI-A [3], CSADMO [9], HDMIO [10]. In all the four algorithms, the population uses real number coding and the population size is set to be $N = 100$. In DNSGAI-A, set the crossover probability p_c and the mutation probability p_m to be 1 and $1/n$, respectively, where n is the dimensions of decision variables. In CSADMO, the clone proportion is set to be 3. In HDMIO the parameters are set as following: $n_D = n_C = 100, n_A = 20$, in the DE, $CR = 0.1, F = 0.5$. In PDNNIA, the parameters are set as following: $n_D = n_C = 100, n_A = 20$, in the DE, $CR = 0.1$. For each DMOP, we measure the performance of the algorithm in different

Table 4 The description of test problems

Test Function	Definition
DMOP1 (FDA1)	$f_1(x_1) = x_1$ $f_2(x_2, \dots, x_n) = g \cdot h$ $g(x_2, \dots, x_n) = 1 + \sum_{i=2}^n (x_i - G(t))^2$ $h(f_1, g) = 1 - \sqrt{(f_1/g)}$ $G(t) = \sin(0.5\pi \cdot t)$ where $x_i \in [0, 1], -1 \leq x_i \leq 1, \forall i = 2, \dots, 10$
DMOP2 (FDA3)	$f_1(x_1) = \sum_{x_i \in X_I} x_i^{F(t)}$ $f_2(x) = g \cdot h$ $g(x_{II}) = 1 + G(t) + \sum_{x_i \in X_{II}} (x_i - G(t))^2$ $h(f_1, g) = 1 - \sqrt{(f_1/g)}$ $G(t) = \sin(0.5\pi \cdot t) $ $F(t) = 10^{2 \sin(0.5\pi \cdot t)}$ where $X_I = (x_1, \dots, x_5) \in [0, 1], X_{II} = (x_6, \dots, x_{10}) \in [-1, 1]$
DMOP3 (dMOP1)	$f_1(x_1) = x_1$ $f_2(x_2, \dots, x_n) = g \cdot h$ $g(x_2, \dots, x_n) = 1 + 9 \cdot \sum_{i=2}^n x_i^2$ $h(f_1, g) = 1 - (f_1/g)^{H(t)}$ $H(x) = 0.75 \cdot \sin(0.5\pi \cdot t) + 1.25$ where $n = 10, x_i \in [0, 1]$
DMOP4 (dMOP2)	$f_1(x_1) = x_1$ $f_2(x_2, \dots, x_n) = g \cdot h$ $g(x_2, \dots, x_n) = 1 + \sum_{i=2}^n (x_i - G(t))^2$ $h(f_1, g) = 1 - (f_1/g)^{H(t)}$ $G(t) = \sin(0.5\pi \cdot t)$ $H(x) = 0.75 \cdot \sin(0.5\pi \cdot t) + 1.25$ where $n = 10, x_i \in [0, 1]$
DMOP5 (FDA4)	$f_1(x_1, x_2) = (1 + g) \cdot \cos(0.5\pi \cdot x_2) \cdot \cos(0.5\pi \cdot x_1)$ $f_2(x_1, x_2) = (1 + g) \cdot \cos(0.5\pi \cdot x_2) \cdot \sin(0.5\pi \cdot x_1)$ $f_3(x_2) = (1 + g) \cdot \sin(0.5\pi \cdot x_2)$ $g(x_3, \dots, x_n) = \sum_{i=3}^n (x_i - G(t))^2$ $G(t) = \sin(0.5\pi \cdot t) $ where $n = 12, x_i \in [0, 1]$
DMOP6 (FDA6)	$f_1(x, t) = (1 + g) \cdot \cos(0.5\pi y_1) \cdot \cos(0.5\pi y_2)$ $f_3(x, t) = (1 + g) \cdot \sin(0.5\pi y_1)$ $g = G(t) + \sum_{j=3}^n (x_j - G(t))^2, G(t) = \sin(0.5\pi t) $ $y_1 = x_1^{F(t)}, y_2 = x_2^{F(t)}, F(t) = 1 + 100\sin^4(0.5\pi t),$ $t = \frac{1}{n_T} \left\lfloor \frac{\tau}{\tau_T} \right\rfloor$ where $n = 12, x_i \in [0, 1]$

environmental change combinations (τ_T, n_T) , that is the τ_T can take a value of 15 or 25 and n_T can take a value of 5 or 10. We run each algorithm 20 times for each test instance in each environment change combination independently. The algorithms stop when $t > 50$, i.e., there are 50 environmental

changes in each run. Table 5 shows the means and variances of GD and SP obtained by the four algorithms, in which M represents the mean and V represents the variance.

As can be seen from Table 5, for all the six test problems and in all the environmental change combinations (τ_T, n_T) ,

Table 5 The mean and variance of GD and SP obtained by the four algorithms

DMOPs	(τ_T, n_T)		GD				SP			
			DNSGA II-A	CSA DMO	HD MIO	PDNN IA	DNSGA II-A	CSA DMO	HD MIO	PDNN IA
DMOP1 (FDA1)	(15,10)	M	4.96E-02	6.90E-01	1.52E-02	1.12E-02	1.49E-02	1.84E-01	1.21E-02	1.01E-02
		V	6.13E-03	5.08E-02	8.19E-04	7.31E-04	9.72E-04	1.87E-02	3.35E-04	4.33E-04
	(25,10)	M	1.71E-02	1.89E-01	6.50E-03	4.12E-03	7.68E-03	7.26E-02	7.10E-03	5.80E-03
		V	1.00E-03	1.37E-02	2.76E-04	3.19E-04	2.80E-04	7.29E-03	1.94E-04	1.77E-04
	(25,5)	M	6.45E-02	8.21E-01	8.91E-03	7.28E-03	1.51E-02	1.93E-01	8.75E-03	7.22E-03
		V	1.01E-02	5.73E-02	3.64E-04	4.41E-04	9.51E-04	3.19E-02	2.99E-04	3.08E-04
DMOP2 (FDA3)	(15,10)	M	1.36E+00	3.11E+00	5.88E-01	4.34E-01	2.16E-01	2.78E-01	1.88E-01	1.79E-01
		V	2.23E-01	3.47E-01	6.73E-02	5.22E-02	5.18E-02	9.21E-02	1.24E-02	1.51E-02
	(25,10)	M	8.26E-02	2.36E-01	2.61E-02	2.07E-02	3.82E-02	2.40E-02	6.66E-02	5.70E-02
		V	1.18E-02	1.39E-02	8.71E-04	1.02E-03	1.15E-02	2.47E-03	1.20E-03	1.59E-03
	(25,5)	M	1.11E+00	2.24E+00	7.40E-01	4.91E-01	1.57E-01	1.65E-01	1.65E-01	1.45E-01
		V	1.49E-01	1.47E-01	9.47E-02	5.13E-02	2.91E-02	4.67E-02	8.74E-03	8.57E-03
DMOP3 (dMOP1)	(15,10)	M	8.85E-03	2.93E-01	1.43E-02	1.02E-02	2.38E-02	9.71E-02	7.78E-03	7.73E-03
		V	8.65E-03	1.25E-01	1.11E-02	5.95E-03	1.87E-02	5.07E-02	2.68E-03	1.74E-03
	(25,10)	M	4.02E-04	8.10E-02	6.13E-04	5.90E-04	9.12E-03	2.57E-02	3.37E-03	3.51E-03
		V	2.53E-05	4.96E-02	4.74E-05	7.19E-05	2.80E-04	2.38E-02	1.05E-04	1.02E-04
	(25,5)	M	4.16E-04	6.81E-02	6.44E-04	6.10E-04	9.43E-03	1.46E-02	3.65E-03	3.81E-03
		V	6.05E-05	4.80E-02	5.80E-05	6.46E-05	1.21E-03	9.40E-03	1.21E-04	1.02E-04
DMOP4 (dMOP2)	(15,10)	M	1.46E-02	6.03E-02	4.96E-03	3.35E-03	1.01E-02	4.10E-02	6.96E-03	5.75E-03
		V	6.20E-04	4.68E-03	1.80E-04	1.70E-04	3.70E-04	4.55E-03	1.68E-04	1.18E-04
	(25,10)	M	6.28E-03	1.63E-02	2.29E-03	1.46E-03	6.61E-03	1.30E-02	4.60E-03	3.99E-03
		V	2.66E-04	1.78E-03	5.87E-05	4.50E-05	1.55E-04	1.47E-03	9.05E-05	6.88E-05
	(25,5)	M	4.29E-02	2.53E-01	6.07E-03	4.74E-03	1.97E-02	1.14E-01	8.25E-03	6.87E-03
		V	4.17E-03	2.89E-02	2.08E-04	2.17E-04	8.59E-04	1.41E-02	2.48E-04	2.08E-04
DMOP5 (FDA4)	(15,10)	M	4.08E-01	6.05E-01	1.13E-01	9.17E-02	1.24E-01	1.17E-01	9.24E-02	8.63E-02
		V	3.17E-02	1.48E-02	5.55E-03	6.61E-03	5.87E-03	2.89E-03	2.06E-03	2.38E-03
	(25,10)	M	6.10E-02	5.94E-02	2.06E-02	1.81E-02	6.13E-02	6.31E-02	5.63E-02	5.54E-02
		V	4.20E-03	3.19E-03	3.48E-04	2.65E-04	1.18E-03	8.78E-04	7.06E-04	5.39E-04
	(25,5)	M	1.71E-01	3.52E-01	4.12E-02	3.67E-02	7.52E-02	9.27E-02	6.25E-02	6.15E-02
		V	8.79E-03	1.12E-02	2.29E-03	1.80E-03	3.05E-03	1.69E-03	1.02E-03	9.57E-04
DMOP6 (FDA5)	(15,10)	M	2.12E-01	2.88E-01	6.36E-02	5.45E-02	1.19E-01	1.21E-01	9.92E-02	9.70E-02
		V	1.09E-02	8.04E-03	1.62E-03	1.66E-03	4.45E-03	2.78E-03	1.40E-03	1.08E-03
	(25,10)	M	7.66E-02	6.68E-02	2.99E-02	2.67E-02	9.82E-02	9.64E-02	8.94E-02	8.87E-02
		V	3.45E-03	3.67E-03	3.27E-04	3.36E-04	1.21E-03	1.09E-03	1.12E-03	9.79E-04
	(25,5)	M	1.44E-01	2.35E-01	4.05E-02	3.72E-02	1.08E-01	1.14E-01	9.26E-02	9.15E-02
		V	6.55E-03	8.51E-03	1.21E-03	1.77E-03	2.52E-03	2.56E-03	1.05E-03	1.32E-03

PDNNIA can get the best means except for DMOP3 and one environmental change combination of DMOP2, and in most cases, the smallest variance values are also obtained by PDNNIA. This means that PDNNIA performs the best among the four algorithms and has stable performance. The results obtained by PDNNIA have the best the convergence and distribution.

The inverted generational distance (IGD_t) [29] reflects both the convergence and diversity. IGD_t is used to

evaluate the performance of the algorithms in all 50 iterations ($0 \sim 49$). IGD_t is calculated in the last generation of every time t and is defined as follows:

$$IGD_t(A_t, PF_t) = \frac{\sum_{v \in PF_t} d(v, A_t)}{|PF_t|} \quad (24)$$

Let PF_t be a set of uniformly distributed points on the true Pareto-optimal front at the t moment, and A_t be the approximate Pareto-optimal set of dominant antibodies obtained by

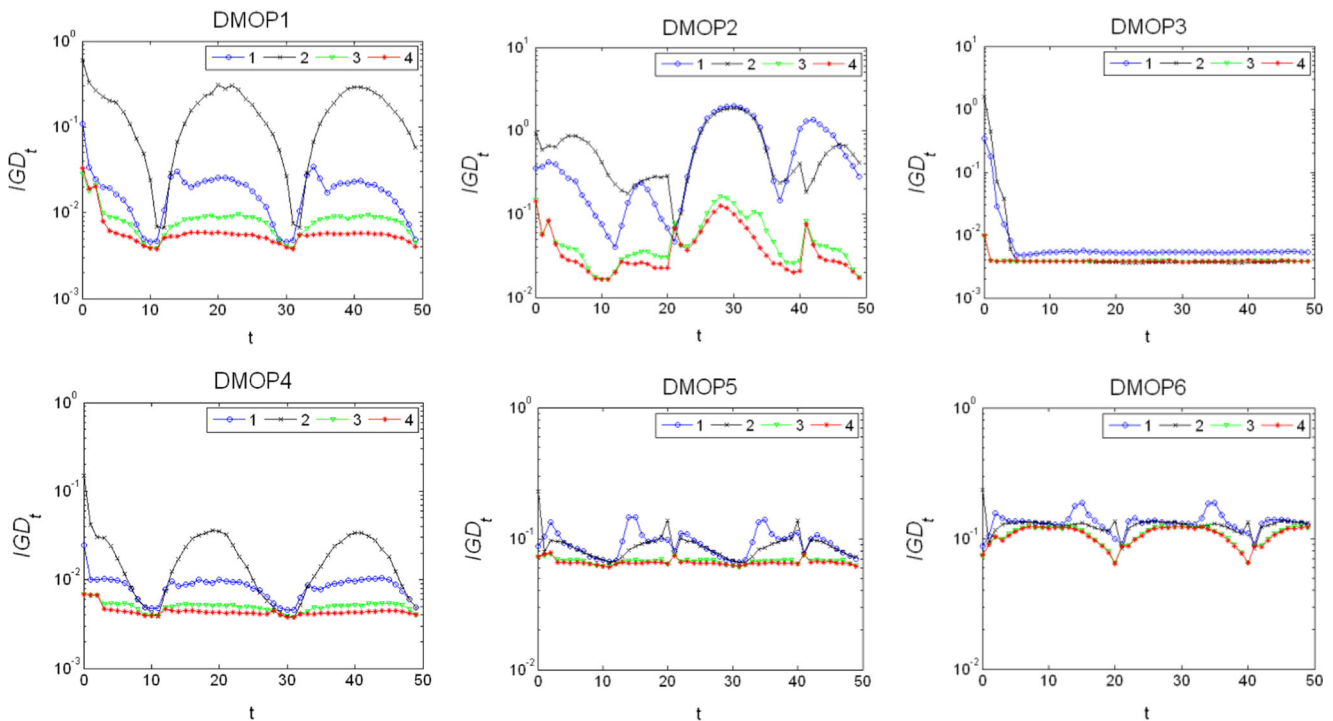


Fig. 6 IGD_t obtained by the four algorithms

DMOEA at the t moment. Where $d(v, PF_t)$ is the minimum Euclidean distance among all the distances between each vector v in PF_t and every point in A_t , T_{max} is the maximum number of environmental change in a run.

For each test problem, we compare the IGD_t obtained by each algorithm in the whole process of the evolution with the second environmental change combination, i.e. $(\tau_T, n_T) = (25, 10)$, and the results are shown in Fig. 6. In the figure, for convenience, algorithm 1, algorithm 2, algorithm 3 and algorithm 4 represent DNSGAI-A, CSADMO, HDMIO and PDNNIA respectively.

From Fig. 6, we can see that IGD_t obtained by PDNNIA always maintains the minimum value in the whole process of evolution for each DMOP. For DMOP1, DMOP2, DMOP4 and DMOP5, the proposed algorithm PDNNIA significantly outperforms the other three algorithms. For DMOP3 and DMOP6, the performance of PDNNIA as good as HDMIO and they get the best results together.

4.4 Experiential study on the modified predictive model and the modified adaptive DE

In the following experiments, we conducted two sets of comparative experiments to test the performance of the two modified operators, i.e., the modified predictive model and adaptive differential evolution operator.

4.4.1 Experiential study on the modified predictive model

The predictive model comes into action at the beginning generation of a new environment that used to predict the initial population. In this experiment, the modified predictive model will compare with the other two predictive model, which are shown in Table 6. In Table 6, the predictive strategy of [25] is the original distance model. The predictive strategy proposed in this paper is described in Section 3.1.2. We combine the three predictive models with the same static

Table 6 The description of the three predictive models

Model	Description of the predictive model
predictive model 1	The initial population is produced by the predictive strategy of [25].
predictive model 2	The half of the initial population is produced by the prediction strategy of [25], the other half is randomly generated.
predictive model 3	The predictive strategy proposed in this paper.

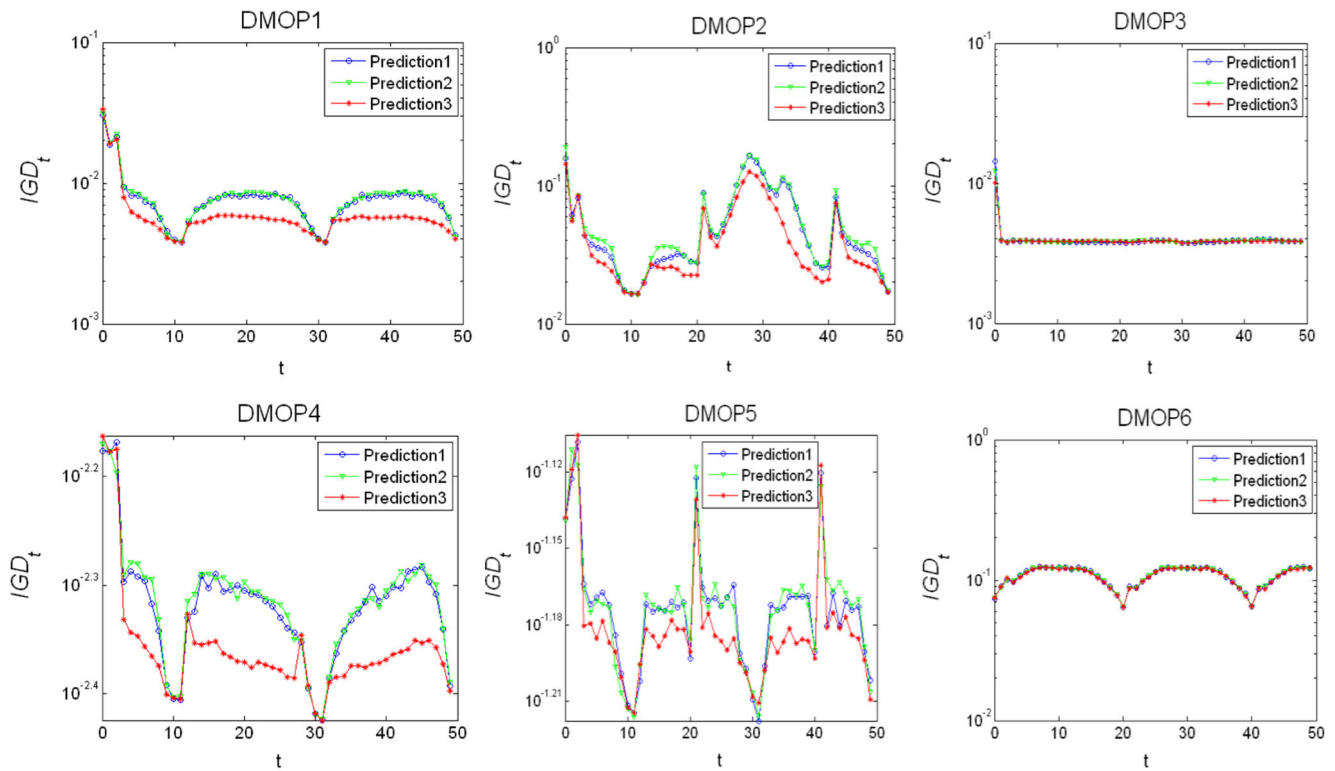


Fig. 7 IGD_t obtained by the three algorithms

MOP algorithm, that is the NNIA which based on the modified self-adaptive differential evolution crossover, and get three DMOP algorithm, which are denoted as Prediction 1, Prediction 2 and Prediction 3 respectively. Expect the predictive model, the other operators and parameters of all the three algorithms are set as same as those in Section 4.1.

With the three DMOP algorithm testing the six DMOPs respectively, We use the metric IGD_t to evaluate the performance of the three predictive model. We run each algorithm 20 times for each test instance in the second environmental change combination, i.e., $(\tau_T, n_T) = (25, 10)$, independently. There are 50 environmental changes in each run, i.e., $T_{max} = 50$. The results are shown in Fig. 7.

Because the predictive model is based on the historical information, we only analyze the performance of the three algorithm when $t \geq 3$. When $0 < t < 3$, the predictive can not come into action, so the three algorithms use the same strategy to initialize the polulation.

We can see from Fig. 7 that, for DMOP1, DMOP2, DMOP4 and DMOP5, the Prediction 3 algorithm significantly outperforms the other two algorithms, this suggests that the model proposed in this paper is significantly better than the other two predictive models. For DMOP3 and DMOP6, the performances of the three predictive models are almost the same. So we can draw a conclusion that the predictive model proposed in this paper has advantages in predicting and tracking the environmental change.

4.4.2 Experimental study on the modified adaptive DE

In this experiment, the modified adaptive DE will compare with the other two DE operator, which are shown in Table 7. DE1 is the original differential evolution operator and the parameter F is set to be 0.5. The DE2 is the adaptive differential evolution operator in [26]. DE3 is the

Table 7 The Description of the three algorithms

Algorithm	Description of the algorithm
DE1	The original differential evolution operator(F=0.5).
DE2	The adaptive differential evolution operator in [26].
DE3	The modified simpler adaptive differential evolution operator proposed in this paper.

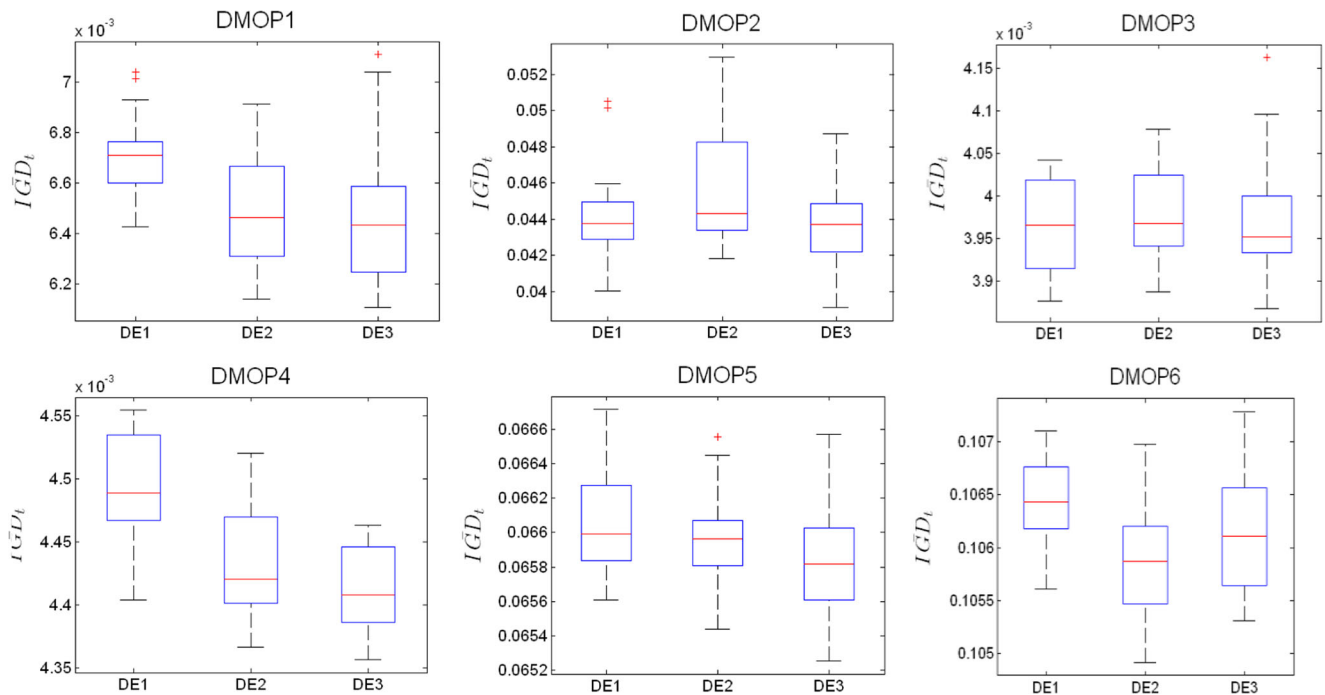


Fig. 8 \overline{IGD}_t obtained by the three algorithms

modified adaptive differential evolution operator proposed in this paper. Introduce the three DE operator to the same DMOP algorithm respectively and get three DMOP algorithm, which are denoted as Algorithm 1, Algorithm 2 and Algorithm 3 respectively. Except the DE operator, the other operators and parameters of all the three algorithms are set as same as those in Section 4.1.

With the three algorithms testing the six DMOPs respectively, we run each algorithm 20 times for each test instance in the second environmental change combination, i.e., $(\tau_T, n_T) = (25, 10)$, independently. There are 50 environmental changes in each run, i.e., $T_{max} = 50$. We keep a record of the IGD_t of each time in one run and evaluate the average value of the 50 IGD_t , which denoted by \overline{IGD}_t . We figure out the boxplot of \overline{IGD}_t in 20 independent runs in Fig. 8.

We can see from Fig. 8 that our modified simpler adaptive differential evolution operator can obtain a relatively better values of \overline{IGD}_t with a better stability for the most problems except for DMOP6. In DMOP6, the algorithm 3 with the DE3 operator obtain the second best result. This shows that our modified simpler adaptive differential DE is effective though it is simple.

5 Conclusion

A novel DMOEA based on the improved predictive model and the modified adaptive differential evolution crossover operator is proposed in this paper. Two sets of comparative

experiments are conducted to test the performances of the proposed algorithm and the effectiveness of the two modified operators. Experimental results show that the proposed algorithm is effective and generic for all types of DMOPs, and two modified operators can significantly obtain better performances with respect to the original versions. For the two-objective problems with unchanging PS_t and changing PF_t or the three-objective problems with changing PS_t and changing PF_t , the superiority of the proposed algorithm is not very obvious. In the next work, we will analyze which types of problems our algorithm is particularly applicable for and modify the algorithm further to improve the performance of the algorithm on these two types of DMOPs mentioned above.

References

1. Farina M, Amato P, Deb K (2004) Dynamic multi-objective optimization problems: Test cases, approximations and applications. *IEEE Trans Evol Comput* 8:425–442
2. Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol Comput* 8: 173–195
3. Zeng SY, Chen G, Zheng L, Shi H, Garis HD, Ding LX, Kang LS (2006) A dynamic multi-objective evolutionary algorithm based on an orthogonal design. In: *IEEE congress on evolutionary computation*, pp 573–580
4. Goh CK, Tan KC (2009) A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization. *IEEE Trans Evol Comput* 13:103–127

5. Wang YP, Dang CY (2008) An evolutionary algorithm for dynamic multi-objective optimization. *Appl Math Comput* 205: 6–18
6. Koo WT, Goh CK, Tan KC (2010) A predictive gradient strategy for multiobjective evolutionary algorithms in a fast changing environment. *Memetic Computing* 2:87–110
7. Coello Coello CA, Cortes NC (2002) An approach to solve multiobjective optimization problems based on an artificial immune system. *Proceedings of the First International Conference on Artificial Immune Systems*, pp 212–221
8. Freschi F, Repetto M (2005) Multiobjective optimization by a modified artificial immune system algorithm. *Proceedings of the Fourth International Conference on Artificial Immune Systems, ICARIS 2005*, vol 3627 of *Lecture Notes in Computer Science*, pp 248–261
9. Cutello V, Narzisi G, Nicosia G (2005) A class of Pareto archived evolution strategy algorithms using immune inspired operators for ab-initio protein structure prediction. *Third European Workshop on Evolutionary Computation and Bioinformatics, EvoWorkshops 2005–EvoBio 2005*, vol. 3449 of *Lecture Notes in Computer Science*, pp 54–63
10. Gong MG, Jiao LC, Du HF, Bo LF (2008) Multi-objective immune algorithm with nondominated neighbor-based selection. *Evol Comput* 16:225–255
11. Shang RH, Jiao LC, Gong MG et al. (2005) Clonal Selection Algorithm for Dynamic Multiobjective Optimization. In: Hao Y. (ed) *CIS 2005, Part I, LNCS(LNAI)*, vol 3801. Springer, Heidelberg, pp 846–851
12. Zhang ZH (2008) Multiobjective optimization immune algorithm in dynamic environments and its application to greenhouse control. *Appl Soft Comput* 8:959–971
13. Zhang ZH, Qian SQ (2009) Multi-objective immune optimization in dynamic environments and its application to signal simulation. In: *2009 International conference on measuring technology and mechatronics automation*, vol 3. Hunan, China, pp 246–250
14. Ma YJ, Liu RC, Shang RH (2011) A Hybrid Dynamic Multi-objective Immune Optimization Algorithm Using Prediction Strategy and Improved Differential Evolution Crossover Operator. In: Lu B. L., Zhang L., Kwok J. (eds) *ICONIP 2011, Part II, LNCS*, vol 7063. Springer, Heidelberg, pp 435–444
15. Zhang ZH, Qian SQ (2011) Artificial immune system in dynamic environments solving time-varying non-linear constrained multi-objective problems. *Soft Comput* 15:1333–1349
16. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6:182–197
17. Storn R, Price K (1997) Differential Evolution—A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *J Glob Optim* 11:341–359
18. Abbass HA, Sarker R., Newton C. (2001) PDE, A Pareto-frontier differential evolution approach for multi-objective optimization problems. *Proceedings of the Congress on Evolutionary Computation 2001 (CEC2001)*, vol 2. IEEE Service Center, Piscataway, pp 971–978
19. Xue F, Sanderson AC, Graves RJ Pareto-based multi-objective differential evolution. *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, vol 2003. IEEE Press, Canberra, Australia, pp 862–869
20. Rolic T, Filipic B (2005) DEMO: Differential Evolution for Multiobjective Optimization. In: Coello CA et al. (eds) *EMO 2005, LNCS 3410*, pp 520–533
21. Deb K, Bhaskara UN, Karthik S (2007) Dynamic Multi-Objective Optimization and Decision-Making Using Modified NSGA-II: A Case Study on Hydro-Thermal Power Scheduling. In: Obayashi S et al. (eds) *Proceedings of EMO 2007, LNCS*, vol 4403. Springer, Verlag, pp 803–817
22. Zheng BJ (2007) A New Dynamic Multi-objective Optimization Evolutionary Algorithm. In: *Third International Conference on Natural Computation (ICNC 2007)*, pp 565–570
23. Wei JX, Wang YP (2012) Hyper rectangle search based particle swarm algorithm for dynamic constrained multi-objective optimization problems. In: *IEEE World Congress on Computational Intelligence (WCCI 2012)*, pp 1–8
24. Hatzakis I, Wallace D (2006) Dynamic multi-objective optimization with evolutionary algorithms: A forward-looking approach. In: *Proceedings of Genetic and Evolutionary Computation Conference (GECCO 2006)*. Seattle, Washington, USA, pp 1201–1208
25. Zhou AM, Jin YC, Zhang QF, Sendhoff B, Tsang E (2007) Prediction-Based Population Re-Initialization for Evolutionary Dynamic Multi-Objective Optimization. In: Obayashi S et al. (eds) *EMO 2007, LNCS*, vol 4403. Springer, Heidelberg, pp 832–846
26. Qian WY, Li AJ (2008) Adaptive differential evolution algorithm for multiobjective optimization problems. *Appl Math Comput* 201:431–440
27. Van V (1999) A. D, Multi-Objective evolutionary algorithms: Classification, analyzes, and new innovations. Ph.D. Thesis. Wright-Patterson AFB. Air Force Institute of Technology
28. Schott JR (1995) Fault tolerant design using single and multicriteria genetic algorithm optimization. Master thesis. Massachusetts Institute of Technology
29. Zhang QF, Li H (2007) MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans Evol Comput* 11:712–731



Ruochen Liu is currently an associate professor with the Intelligent Information Processing Innovative Research Team of the Ministry of Education of China at Xidian University, Xian, China. She received her PhD from Xidian University, Xian, China, in 2005. Her research interests are broadly in the area of computational intelligence. Her areas of special interest include artificial immune systems, evolutionary computation, and data mining.



Jing Fan received her B.S. from Xidian University, Xian, China. She is currently working toward her M.S. in Xidian University. Her current research focuses on evolutionary computation.



Licheng Jiao received his PhD from Xian Jiaotong University, Xian, China, in 1990. He is currently professor and dean of the Electronic Engineering School at Xidian University, China. His current research focuses on intelligent information processing.