A Dynamic Multi-objective Evolutionary Algorithm based on Polynomial Regression and Adaptive Clustering

Qiyuan Yu[1], Qiuzhen Lin[1]*, Zexuan Zhu[1]*, Ka-Chun Wong[2], Carlos A. Coello Coello[3]

[1]*College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, PR. China*

[2] *Department of Computer Science, City University of Hong Kong, Hong Kong SAR*

[3]*CINVESTAV-IPN, Department of Computer Science, Mexico, D.F., 07360, Mexico*

Abstract: In this paper, a dynamic multi-objective evolutionary algorithm is proposed based on polynomial regression and adaptive clustering, called DMOEA-PRAC. As the Pareto-optimal solutions and fronts of dynamic multi-objective optimization problems (DMOPs) may dynamically change in the optimization process, two corresponding change response strategies are presented for the decision space and objective space, respectively. In the decision space, the potentially useful information contained in all historical populations is obtained by the proposed predictor based on polynomial regression, which extracts the linear or nonlinear relationship in the historical change. This predictor can generate good initial population for the new environment. In the objective space, in order to quickly adapt to the new environment, an adaptive reference vector regulator is designed in this paper based on K-means clustering for the complex changes of Pareto-optimal fronts, in which the adjusted reference vectors can effectively guide the evolution. Finally, DMOEA-PRAC is compared with some recently proposed dynamic multi-objective evolutionary algorithms and the experimental results verify the effectiveness of DMOEA-PRAC in dealing with a variety of DMOPs.

Keywords: Dynamic multi-objective optimization; Adaptive clustering; Polynomial regression

1. Introduction

The study on dynamic multi-objective optimization problems (DMOPs) is of great practical significance in various application fields [1]. Except for the constraints and conflicts existing among the optimization objectives, DMOPs further show complex time-varying characteristics on their objectives or decision variables. This kind of problems has been widely found in a variety of practical projects, such as internet of things service problem [2], dynamic control problem [3], path planning problem [4], and railway hub dispatching problem [5]. The key challenge for solving DMOPs is how to respond to the dynamically changing environments, such as the changes in the number of objectives [6-7] and decision variables [2], and the changing constraints [8], which bring difficulties to optimization. In [9], DMOPs are classified into four types according to the changes of Pareto-optimal set (POS) and/or Pareto-optimal front (POF) with time. The solving of DMOPs should track the POS and POF in a short time. In order to solve these intricate problems, many dynamic multi-objective evolutionary algorithms (DMOEAs) have been proposed and become the mainstream for solving DMOPs. In general, the goal of solving all types of DMOPs is to continuously obtain the POS within each environment. There are various kinds of mechanisms and techniques embedded into the current DMOEAs to achieve this goal, such as diversity-based DMOEAs [10-12], memory-based DMOEAs [13-15], and prediction-based DMOEAs [16-19]. Diversity-based DMOEAs mainly focus on maintaining the diversity of populations and quickly track new environments by jumping out of the current optimal scene. The main idea of memory-based DMOEAs is to reuse the historical population according to some judgment conditions, which is a simple and effective scheme when the state of the new environment is consistent with that of a certain historical environment. Prediction-based DMOEAs use population movement or some effective models to generate the initial population of the new environment based on the assumption that there is a certain relationship between changes.

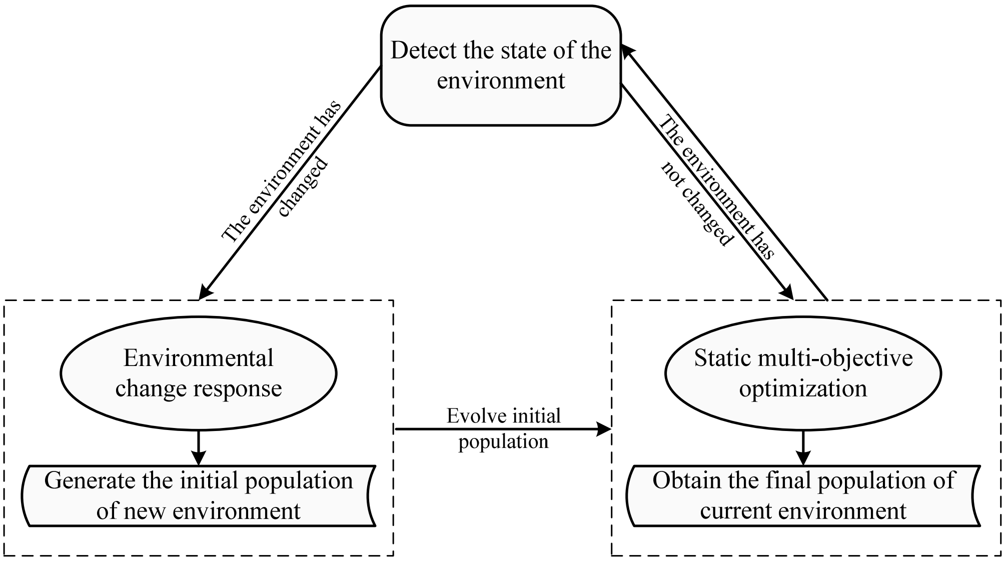

Detect the state of the environment
The environment has changed
The environment has not changed
Environmental change response
Generate the initial population of new environment
Evolve initial population
Static multi-objective optimization
Obtain the final population of current environment

Fig. 1 The general framework of DMOEAs

The general framework of DMOEAs is shown in Fig. 1. Obviously, the key for solving DMOPs includes two aspects: responding to change and promoting optimization. On the one hand, in order to alleviate the evolutionary pressure of static optimization, DMOEAs need to effectively generate the initial population in the new environment after the environment changes. On the other hand, DMOEAs need to quickly search the final population with good convergence and diversity in a limited number of evolutions for each environment.

It is a remarkable fact that the dynamic changes in DMOPs may occur either in the decision space or in the objective space, which is reflected on the changes of POS and POF, respectively. However, in order to respond to change or promote optimization, the key works in most DMOEAs mainly act on one of the spaces. Some DMOEAs only consider the change or movement of solutions in the decision space. For example, in [20], a DMOEA is proposed based on the change type detection. Although the change type detected by this algorithm is based on the definition of the problem summarized by Farina

[9], the detection method is only based on the difference in the number of non-dominated solutions before and after the change, so that this method can only detect whether the POS has changed, but cannot clearly distinguish the changes in POF. In [21], two directed search strategies are presented to obtain the initial population of the new environment: one of which reinitializes the population based on the predicted moving direction and the orthogonal direction of the movement of POS, and the other generates the desired individuals in predicted regions through the moving direction of the non-dominated solutions between two successive iterations. Beyond these, some DMOEAs only design the main strategies according to the characteristics or status of the objective space, but they ignore the acquisition of the initial population at the beginning of the change. For example, a couple of strategies are adopted to maintain the diversity and convergence of the population after environmental changes in [22], in which the simulated isotropic magnetic particles niching and the scheme of environmental selection guided by non-dominated individuals are designed based on some information in objective space, and the main role of these two strategies is to promote static optimization. However, this algorithm omits the response to changes in decision space, as it is not equipped with some measures to generate the initial population after changes, which means the lack of flexibility of the algorithm in responding to changes.

From the above analysis, it can be seen that the response strategies acting in both the decision space and the objective space jointly affect the overall performance of DMOEAs to a certain extent. For some DMOPs with little change correlation or rapid change, the DMOEA that lacks a response in a certain space may reap some disappointing final populations. In order to solve the above problem, two effective strategies which act in the decision space and the objective space are respectively proposed in this paper, and a DMOEA based on these effective components is presented, called DMOEA-PRAC. The main contributions of this paper are summarized as follows:

1). A polynomial regression based predictor (called PR) is designed for generating initial population according to the change of POS in the decision space. The potentially available information contained in all historical populations is obtained by the PR predictor based on polynomial regression model, which extracts the linear or nonlinear changing relationship of historical environments. After each environmental change, DMOEA-PRAC can generate promising initial solutions for new environment.

2). An adaptive reference vector regulator (called AC) based on K-means clustering is suggested to track the complex changes of POF in the objective space. Inspired by the evolutionary trend of the current population, the regulator obtains the suitable reference vectors for the current environment by clustering. Through the AC regulator, the adjusted vectors can guide the evolution effectively, and DMOEA-PRAC can adapt to the new environment quickly.

3). The performance of DMOEA-PRAC is studied and analyzed by numerous experimental studies for solving DMOPs with various characteristics, which involve disconnected POFs, degenerated POFs and concave-convex switching POF. Here, DMOEA-PRAC is compared with some state-of-the-art

DMOEAs, and the experimental results verify the effectiveness of DMOEA-PRAC in dealing with a variety of DMOPs.

The rest of this paper is organized as follows: the related works of current DMOEAs and the motivations of the designed DMOEA-PRAC are described in Section 2. The whole flow and details of DMOEA-PRAC are presented in Section 3. In Section 4, the discussions of the experimental results of DMOEA-PRAC with some competitive DMOEAs are provided. At last, the conclusions of this paper and the future work are outlined in Section 5.

2. Related Works and Our Motivations

2.1 Some Definitions of DMOPs

This paper mainly concerns the unconstrained DMOP with the changing objective values, which is mathematically defined as follows:

$$\begin{array}{ll} \text{minimize} & F(\mathbf{x},t)=(f_1(\mathbf{x},t),f_2(\mathbf{x},t),...,f_m(\mathbf{x},t))^T, \\ \text{subject to} & \mathbf{x}\in\Omega_x,\ t\in\Omega_t \end{array} \quad (1)$$

where $F(\mathbf{x},t)$ is a set of m objective functions at time t. In the decision space $\Omega_x \in R^n$, a solution $\mathbf{x}=(x_1,x_2,...x_n)^T$ contains n decision variables. Here, t is the time index in the time space Ω_t, which is defined by

$$t=\frac{\lfloor \tau/\tau_t \rfloor}{n_t} \quad (2)$$

where τ and τ_t represent the generation counter and the frequency of change, respectively, and n_t is the severity of change. The smaller value of τ_t means the faster dynamic change, and the smaller value of n_t indicates the larger vibration in each change.

For DMOPs, there are some basic definitions as follows:

Definition 1: For two solutions x_1 and x_2, x_1 is said to dominate x_2, if and only if: $\forall i\in\{1, 2, ..., m\}$ $f_i(x_1,t)\le f_i(x_2,t)$, and $\exists j \in\{1, 2, ..., m\}$ $f_j(x_1,t)<f_j(x_2,t)$, where $f_i(x,t)$ is ith objective value at time t.

Definition 2: For a nondominated (Pareto-optimal) solution x_t^* at time t, there is no other solution $x\in\Omega_x$ to dominate x_t^*. The Pareto-optimal set includes all nondominated solutions at time t, represented by POS$_t$.

Definition 3: At time t, the corresponding objective vectors of the POS$_t$ are called the Pareto-optimal Front, represented by POF$_t$.

According to the changing status of the POS and POF, DMOPs are divided into the following four types [9]:

Type I: The POS changes with time, while the POF remains unchanged.

Type II: Both the POS and the POF change with time.

Type III: The POS is invariable, while the POF changes with time.

Type IV: Although the objective function changes, the POS and the POF are all fixed.

2.2 A Brief Introduction of DMOEAs

At the beginning of solving DMOPs, DMOEAs usually need a change detection mechanism to judge whether the environment has changed or not. At present, the mainstream change detection mechanism is the reevaluation method [10, 17], and its main idea is to select a part of the solution from the current population to detect the environmental change by checking whether their objective values are the same. In this paper, a simple and effective reevaluation method is adopted to detect change. If a change is detected, DMOEAs need some response schemes to track the change and provide the new environment with a good initial population to approximate the true POS and POF as closely as possible. According to the change response mechanism, the existing DMOEAs can be divided into the following three categories: diversity-based DMOEAs, memory-based DMOEAs, and prediction-based DMOEAs.

1) *Diversity-based DMOEAs*. When solving DMOPs, the population may have converged to a specific region in the current environment. If the change occurs, it may be difficult for DMOEAs to track and locate the new global optimality due to the diversity loss. Diversity-based DMOEAs emphasize on improving this situation, and this kind of DMOEA first starts from DNSGA-II proposed by Deb et al. [10]. There are two response strategies in DNSGA-II, i.e., random initialization and mutation mechanism, which introduce diversity by adding some randomly generated individuals or individuals with Gaussian variation to the population. In [11], a simple response mechanism is integrated into a DMOEA with vector evaluation in the proposed DVEPSO. When the environment changes, the positions of some particles are reinitialized to respond to the change. In addition, there are some DMOEAs based on diversity maintenance that directly take the POS of the previous moment as the initial population of the new environment. For example, the orthogonal DMOEA called DOMOEA-II is proposed in [12] and the DMOEA based on immune clonal selection strategy is designed in [23], which directly regard the current POS as the initial population of the new environment. Some diversity-based DMOEAs [12, 23, 24] do not take actions to obtain an effective initial population in the new environment, and their individual selection and maintenance methods are based on the information of POF in the objective space, while the influence of POS changes in the decision space is ignored. This kind of DMOEA may be invalid for those DMOPs that change rapidly or drastically.

2) *Memory-Based DMOEAs*. The main idea of the memory-based DMOEAs is to store the POSs searched from the historical environments and reuse the historical POSs in the new environment according to some criterions. For example, a DMOEA based on artificial immune system called DCMOAIS is designed in [13]. In this approach, three modules for initialization, exploration and archiving are suggested to maintain the convergence and diversity of the population. In [14], a memory

mechanism which can store a large number of non-dominated solutions is combined with a static algorithm NSGA-II [25]. When the change occurs, the stored individuals are reused to form the initial population of the new environment. Furthermore, a DMOEA based on memory mechanism and local search is presented in [26]. This algorithm calculates the intensity of environmental change by detecting the matching degree of the current POS in the new environment, and then determines the number of individuals selected by the memory mechanism according to the intensity of change. In [15], a memory mechanism is designed based on change similarity detection. If change is detected, the central points of the POS in a historical environment where the change is similar to the current change will be reused in the initial population of the new environment. For these memory-based DMOEAs, most of them are more effective on the DMOPs with periodic changes, and a certain amount of computation is used to store and extract useful information in memory. In addition, how to design more accurate reuse criteria is also an urgent problem for the memory-based DMOEAs.

3) *Prediction-based DMOEAs.* In recent years, the prediction-based method is very popular in the research of DMOPs due to its strong robustness and flexibility. In the prediction-based DMOEAs, it is usually assumed that the change of the environment follows a predictable pattern. Through the law of change found by a prediction mechanism, the initial population of the new environment can be easily generated. According to the difference of driver kernel, the prediction-based DMOEAs can be roughly divided into experience-driven (knowledge-driven) DMOEAs and model-driven (data-driven) DMOEAs. The predictors in experience-driven (knowledge-driven) DMOEAs are often formed by the descriptive information (experience or knowledge) such as the movement of particular solutions and the types of change, which is obtained by the discriminator of algorithms or observed by researchers. For example, a multi-directional prediction strategy is employed in [27], where some representative individuals are selected from population according to the clustering results, and the representative individuals of the last two environments form a multi-directional prediction sequence. In the new environment, the displacement of each representative individual guides the evolution direction of each cluster. Based on this study, a multi-model prediction algorithm called MMP is designed in [28]. The changing types of POS are subdivided into translation, rotation and combination, which are distinguished by the discriminant strategy of MMP, and different change types correspond to different prediction schemes. Moreover, the DMOEA suggested in [29] relocates the knee points and boundary points according to the movement of the global knee points in the new environment, and the initial population of the new environment is generated near the predicted knee solutions and boundary solutions. In model-driven (data-driven) DMOEAs, the predictors tend to be inspired by some self-learning intelligent models, which can extract some hidden rules of environmental change, and the performance of models depends on the quality of data. There are many advanced techniques that have been applied to model-driven (data-driven) DMOEAs. For example, Kalman filter (KF) is introduced in [16] to solve DMOPs. When

the environment changes, a scoring mechanism is used to decide whether to use the KF-based predictor. Furthermore, a DMOEA based on linear inverse model is proposed in [30], and the inverse model guides the evolution of the population by predicting the reverse function between the decision space and objective space. Based on this study, in order to break through the limitations of linear model, an inverse Gaussian process is used to construct the relationship from the objective space to decision space [31]. In addition, the transfer learning based strategies are widely used to solve DMOPs. In [19], the transfer component analysis (TCA) is adopted to build a prediction model, which uses the acquired knowledge to explore new POS for generating the initial population of new environment. In brief, prediction-based DMOEAs are suitable for most DMOPs and have developed rapidly in recent years. However, this kind of DMOEAs still has a lot of improvement room for research, as the generalization ability of the prediction model and the computational cost of model training can be further improved.

2.3 Our Motivations

Most of the existing works to solve DMOPs intend to consider only the change of one space, that is, the change of POS in the decision space or the change of POF in the objective space. Some DMOEAs that only consider tracking POF in the objective space place emphasize on promoting static optimization, but rarely perform new search in the decision space. If the environmental change is significant, the population of the previous environment is obviously no longer applicable. Therefore, a predictor that acts on the decision space is designed in this paper, which explores the changing trends of POS from the historical populations. Assuming that the change of POS can be expressed by a function, the motion such as translation and rotation can be simply represented by some linear functions, i.e., the vector addition of some individuals with displacement vector or angle vector. However, it is difficult to find the change equation for the inenarrable nonlinear motions. Given that most POS changes can be abstracted by polynomial equation, the designed predictor uses the polynomial model to fit the relationship of POSs. From the perspective of obtaining the implicit information of change, this model-driven predictor is self-learning, which reduces the burden of designing additional judgment conditions to obtain explicit information. In short, this predictor explored in the decision space acquires the initial population of the new environment by learning the linear or nonlinear change relationship between historical POSs, which helps the algorithm respond to change rapidly.

If the response strategy in a DMOEA focuses on searching the decision domain, it may be difficult for the algorithm to deal with the DMOPs whose POF has complex changing characteristics, such as expansion or scaling, disconnection, concave-convex switching and knee region transformation. In Fig. 2, the POFs of DF9, DF11 and DF12 [32] have different truncations and sizes in different environments, while the DMOEAs that only consider the decision space cannot obtain the information about POF changes. In addition, the number of static evolution in each environment is limited, which also makes it difficult for such DMOEAs to track the true POF in the new environment. Therefore, a strategy that works in the objective space is designed in this paper to respond to the changes of POF. Considering that the fixed reference vectors in the decomposition-based algorithm may fail in some regions, an adaptive reference vector regulator is developed in our algorithm. In this regulator, the K-means clustering method is adopted to collect the information about population evolution, and the new reference vectors suitable for the current environment are obtained based on the clustering results. According to the adjusted reference vectors, the proposed algorithm can adapt to the change of the objective space more quickly and track the true POF in the new environment more accurately.

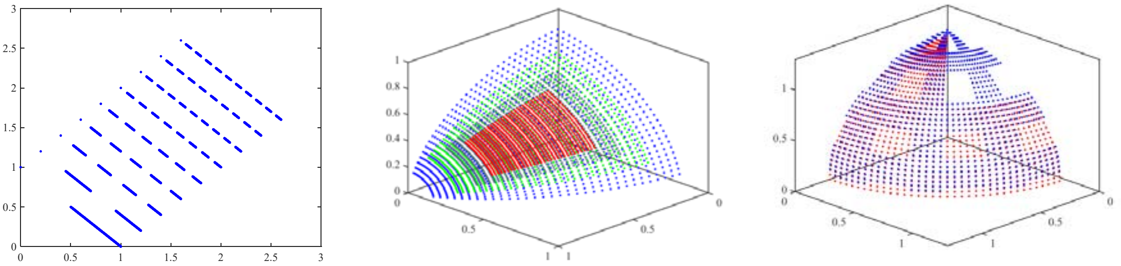


Fig. 2 The changes of POF in DF9, DF11 and DF12

3. The Proposed Algorithm DMOEA-PRAC

This section introduces the proposed DMOEA-PRAC, and the overall framework of DMOEA-PRAC is given in Section 3.1. When the change occurs, DMOEA-PRAC responds to the change in the decision space and the objective space, respectively. For the decision space, a predictor based on polynomial regression is adopted to generate the initial population of the new environment. For the objective space, a reference vector regulator based on adaptive K-means clustering is designed to quickly tackle to the complex changes of POF. The two strategies are described in detail in Section 3.2 and Section 3.3, respectively. In addition, in order to obtain the final population with good convergence and diversity for each environment, a simple and effective environmental selection strategy is suggested based on the adjusted reference vectors, which is provided in Section 3.4.

3.1 The Framework of DMOEA-PRAC

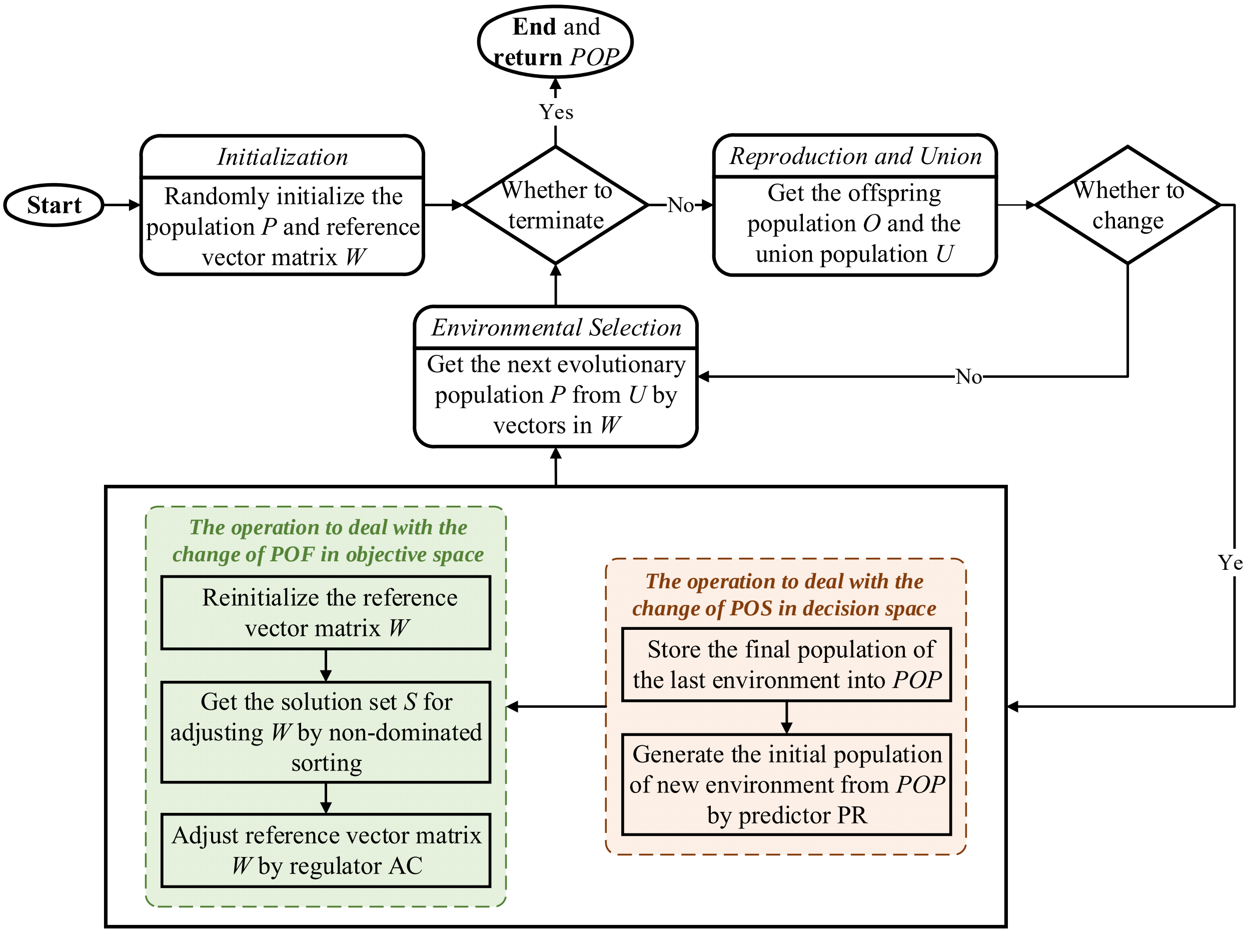

End and return POP
Yes
Start
Initialization
Randomly initialize the population P and reference vector matrix W
Whether to terminate
No
Reproduction and Union
Get the offspring population O and the union population U
Whether to change
No
Environmental Selection
Get the next evolutionary population P from U by vectors in W
Yes
The operation to deal with the change of POF in objective space
Reinitialize the reference vector matrix W
Get the solution set S for adjusting W by non-dominated sorting
Adjust reference vector matrix W by regulator AC
The operation to deal with the change of POS in decision space
Store the final population of the last environment into POP
Generate the initial population of new environment from POP by predictor PR

Fig. 3 The framework of DMOEA-PRAC

The basic framework of the proposed DMOEA-PRAC is shown in Fig. 3, and the pseudo code of the overall process is shown in Algorithm 1. There are five inputs in Algorithm 1, which are a DMOP, the termination condition, the population size N, the number of objective m and the number of variables d. In line 1, the set POP used to preserve the final population in all environments is set to an empty set, and then the population P and the reference vector matrix W are randomly initialized in line 2, where the number of solutions in population P and the number of vectors in W are N. Then, DMOEA-PRAC enters the main loop in lines 4-18 until the termination condition is met. Firstly, the offspring population O is generated by the differential evolution (DE) operator [33], and the union population U is obtained by merging the parent population P and the offspring population O in line 4. If the change is detected in line 5, the final population of the previous environment is preserved in the population set POP in line 6. According to the historical populations retained in POP, the designed PR predictor introduced in Section 3.2 is used in DMOEA-PRAC to predict the initial population P of the new environment in line 7.

Algorithm 1 General Framework of DMOEA-PRAC

Input: a DMOP, the termination criterion, the population size N, the number of objective M, the number of variable d

Output: The stored final population sets in each environment: $POP = \{P_1, P_2, ..., P_t\}$

```
 1: Set $POP = \varnothing$
 2: Initialize and evaluate a population $P$ and generate a reference vector matrix $W$
 3: while the termination criterion is not met do
 4:     Generate the offspring $O$ from $P$ and get the union population $U$ by merging $O$ and $P$
 5:     if change is detected
 6:         Store the final population of the last environment into population set $POP$
 7:         $P = PR(POP, N, d)$          //Algorithm 2
 8:         Reinitialize the vector matrix $W$ randomly
 9:         if the reference vector adjustment condition is met
10:             Divide $U$ into multiple frontiers ($F_1, F_2, ..., F_L$) by the non-dominated sorting
11:             Start from $F_1$, $i = 1$
12:             while $|S| < N$ do
13:                 $S = S + F_i$, $i = i+1$
14:             end while
15:             $W = AC(S, N, M)$          //Algorithm 3
16:         end if
17:     end if
18:     $P = EnvironmentalSelection(U, W, N)$          //Algorithm 4
19: end while
20: return $POP$
```

Subsequently, the algorithm prepares for the adjustment of the reference vectors in the objective space. In order to eliminate the negative influence in the process of vector adjustment in the past, and avoid the adaptive vectors entering a limited or cramped area in the later stage of change, DMOEA-PRAC needs to firstly reinitialize the reference vectors randomly in line 8. Next, if the condition for reference vector adjustment is met, the algorithm will adaptively adjust the vectors based on the proposed AC regulator in lines 10-15. In line 10, the algorithm divides the union population U into multiple Pareto frontiers ($F_1, F_2, ..., F_L$) by fast non-dominated sorting [25]. In order to adjust reference vector effectively, the algorithm obtains a solution set S from different dominance levels in lines 11-14, which starts from non-dominated frontier F_1 in line 11, and then F_i ($i = 1$~L) is orderly added to the solution set S until the number of solutions in S is not less than N in line 13. Subsequently, the reference vector will be updated by the AC regulator suggested in Section 3.3 in line 15. In the K-means clustering based regulator, the generation of new reference vectors is based on the centroids of clusters. Finally, in order to obtain the population with good convergence and diversity, the elite solutions are selected from the union population *U* according to a simple and effective environmental selection strategy in line 18, and the corresponding pseudo code is shown in Algorithm 4. At the end of Algorithm 1, the set *POP* containing the final population of each environment is outputted.

3.2 The Predictor based on Polynomial Regression

3.2.1 The Polynomial Regression

The PR predictor based on polynomial regression is presented to excavate the potential relationship between the historical environments and the new environment. Through the regression analysis, the motion equation of each decision variable is obtained by the polynomial model, which is used to predict the initial solutions in the new environment. In this paper, the polynomial equation is taken as the general form of the movement equation as it can express most linear or nonlinear processes. The unary polynomials are defined as:

$$p(x) = \varpi_0 + \varpi_1 x + \varpi_2 x^2 + \ldots + \varpi_n x^n \tag{3}$$

where ϖ_n is the coefficient of each item, and n is the degree of the polynomial equation. If n is set to 0, it is the linear equation, which can be used to describe the linear relationship of historical changes. If the variable about x in each item is regarded as a feature, and set $x_1 = x$, $x_2 = x^2$ and $x_n = x^n$, Eq. (3) can be transformed into:

$$p(x) = \varpi_0 + \varpi_1 x_1 + \varpi_2 x_2 + \ldots + \varpi_n x_n \tag{4}$$

Obviously, regardless of the form of the feature x_n, Eq. (4) can be regarded as a linear equation about coefficient ϖ_n. For a set of training data $\{(\mathbf{x}, \mathbf{y})\}=\{(\overset{u}{x}_1, y_1), (\overset{uuu}{x}_2, y_2), \ldots, (\overset{uuu}{x}_m, y_m)\}$ containing m training samples, where $\mathbf{x}$ and $\mathbf{y}$ represent the training input matrix and the training output matrix, respectively, and $(\overset{u}{x}_i, y_i)$ is a training sample (i = 1~m). The length of the input data of each training sample is $|\overset{u}{x}_i|$, and the length of the output data is 1. According to the regression process, DMOEA-PRAC learns a polynomial mapping between the input data and the output data, which can reflect the changing trend of the solution in the decision space. Each obtained polynomial mapping serves as a prediction model to provide a variable of a solution for the new environment. In particular, the key of obtaining such a prediction model is to get the appropriate coefficient ϖ_n of Eq. (3) by training. Supposing that $\boldsymbol{\varpi}$ and $\mathbf{X}$ represent the coefficient matrix and the feature matrix, respectively, Eq. (4) can be expressed as:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\varpi} + \boldsymbol{\varepsilon} \tag{5}$$

where $\boldsymbol{\varepsilon}$ is the matrix of constant terms, and the feature matrix $\mathbf{X}$ consists of the input matrix $\mathbf{x}$. For the unary polynomial, $\mathbf{X}$ can be defined as:

$$\mathbf{X}=\begin{bmatrix}1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ & & \dots & & \\ 1 & x_m & x_m^2 & \dots & x_m^n\end{bmatrix} \tag{6}$$

where x_i^j is the *j*th power feature of the input data of the *i*th training sample.

Regarding Eq. (5), although this abstract model is a function about feature terms formed by several variables, it can also be regarded as a linear function about coefficient terms in the process of regression. Therefore, the appropriate model coefficients can be easily obtained through the linear regression. In this paper, the least square method is used to solve the regression coefficients. The main idea is to minimize the sum of squares of deviations between all predicted outputs and true outputs, which is expressed as follows:

$$\min J(\varpi)=\frac{1}{2}\sum_{i=1}^{m}(h_{\varpi}(X_i)-y_i)^2 \tag{7}$$

where $h_{\varpi}(X_i)$ and y_i are the predicted output value and the true output value corresponding to the *i*th input sample, respectively, and $J(\varpi)$ is the regression loss function. In the process of minimizing $J(\varpi)$, the algorithm takes the partial derivative of the loss function, and the matrix derivation is used to obtain the optimal ϖ, which can be calculated as follows:

$$\varpi=(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y} \tag{8}$$

where $\mathbf{X}^{\mathrm{T}}$ is the transpose matrix of **X**. After the optimal ϖ is obtained, the corresponding prediction model can be obtained by bringing the coefficients into Eq. (3).

3.2.2 The PR Predictor

In this work, each prediction model obtained by regression is individual-based, that is, every decision variable of a solution corresponds to a prediction model. Therefore, the designed PR predictor contains *N* × *d* polynomial prediction models, where *N* is the population size and *d* is the number of the decision variables. The structure of the proposed PR can be abstracted as follows:

$$\mathrm{PR}=\begin{cases} x_1 &=[p(x_1^1),\ p(x_1^2), \dots,\ p(x_1^d)] \\ x_2 &=[p(x_2^1),\ p(x_2^2), \dots,\ p(x_2^d)] \\ & \dots \\ x_N &=[p(x_N^1),\ p(x_N^2), \dots,\ p(x_N^d)] \end{cases} \tag{9}$$

where $p(x_i^j)$ is the prediction model of the *j*th decision variable of the *i*th solution in the population (*i* = 1~*N*, *j* = 1~*d*). The data used to predict the new solutions come from the most recent set of input sequences in the historical populations, which are shaped like the input sample of training data. The initial population of the new environment can be generated by inputting the polynomial feature matrices of the prediction data into the PR predictor. The pseudo code of PR predictor is shown in Algorithm 2.

```
Algorithm 2 PR (POP, N, d)
Input: The historical population POP, the population size N, the number of decision variable d
Output: The initial population of new environment: P
 1: Set $P = \varnothing$
 2: Use time window to divide training data and prediction data from POP
 3: for $i$ = 1~$N$
 4:     for $j$ = 1~$d$
 5:         Construct polynomial feature matrix X based on Eq. (10)
 6:         Train the polynomial prediction model in Eq. (3) by linear regression
 7:         The $j$-D variable $x_i^j$ of the $i$th solution is predicted by the fitting model
 8:     end for
 9:     Adjust the boundary of $i$th solution $x_i$
10:     Add $x_i$ to the initial population $P$
11: end for
12: return P
```

In this algorithm, all final populations in the historical environments are fully utilized to predict the new solutions in the decision space. Firstly, the training samples and prediction data are divided from the historical population set *POP* by the time window method in line 2. The algorithm starts from the population of the first environment in *POP*, and the training samples are collected by the time-series movement of the window. Each window corresponds to a sample until the window reaches the last population of *POP*. Each training sample includes two parts: input data **x** and output data **y**, and a time window contains an input data window and an output data window. It should be noted that the length of the output data in the training sample is 1. Suppose that the size of the time window is *win*, and the length of the input data is *win*-1. If the window is not filled up, the algorithm will use random initialization to simply maintain the diversity of the population. If there are *t* historical populations in *POP*, *t*+1-*win* training samples can be collected from *POP*, and the number of training samples increases with the number of environmental changes. For the collection of prediction data, the time window eliminates the output data window and continues to move, and the length of the prediction data is *win*-1.

The PR predictor is individual-based, so each decision variable of a solution corresponds to a prediction model. In lines 3-10, the algorithm will learn each prediction model and use the model to predict the decision variables of new solutions. First of all, the input data composed of decision variables are used to construct the polynomial feature matrix **X** in line 5. In this paper, the length of the training input data is 2, so the corresponding model is a binary polynomial equation, and the polynomial feature matrix **X** is expressed as:

$$\mathbf{X}=\begin{bmatrix}1, x_{1a}, x_{1b}, \ldots, x_{1a}^{i}, x_{1b}^{j}, \ldots, x_{1a}^{i}x_{1b}^{j}, \ldots \\ 1, x_{2a}, x_{2b}, \ldots, x_{2a}^{i}, x_{2b}^{j}, \ldots, x_{2a}^{i}x_{2b}^{j}, \ldots \\ \ldots \\ 1, x_{ma}, x_{mb}, \ldots, x_{ma}^{i}, x_{mb}^{j}, \ldots, x_{ma}^{i}x_{mb}^{j}, \ldots\end{bmatrix}(i, j=1\sim n) \qquad (10)$$

where x_{ma} and x_{mb} are the binary input data of the *m*th sample, x_{ma}^{i} and x_{mb}^{j} represent the higher power of the binary input data, and $x_{ma}^{i}x_{mb}^{j}$ represents the product of different higher powers of the binary input data. After obtaining the polynomial feature matrix **X**, the algorithm trains the polynomial prediction model in line 6. In this process, the optimal coefficient matrix $\boldsymbol{\varpi}$ of the prediction model is obtained through Eq. (8), and the corresponding prediction model can be obtained by substituting the coefficients in $\boldsymbol{\varpi}$ into Eq. (3). Finally, the prediction model is used to calculate the *j*th decision variable of the *i*th solution (i = 1~N and j = 1~d) in line 7. For each predicted solution x_i, its boundary must be checked and adjusted in line 8 to ensure that the solution is within the range of the decision variable. In line 9, the solution x_i is added into the initial population P. In short, when a change occurs, DMOEA-PRAC can quickly search effective initial solutions of the new environment through the PR predictor.

For the PR predictor, the matrix operation of the normal equation method in the model training process costs the major time, and its time complexity is $O(mn^3)$, where m is the number of training samples and n is the dimension of the features. The dimension of features depends on the number of training input data *win*-1 and the degree of polynomial. This paper comprehensively considers the acquisition of nonlinear models and the avoidance of over-fitting, and the degree is set to 2, so the dimension of features is $2^1C_{win-1}^1+2^2C_{win-1}^2$, i.e., $2C_{win-1}^1+4C_{win-1}^2$. When obtaining the initial population of the t+1 environment, there are t+1-*win* samples for model training, and the training complexity of a model is $O((t+1-win)\times(2C_{win-1}^1+4C_{win-1}^2)^3)$. Since the PR predictor contains N×d polynomial prediction models, the total training complexity is $O(Nd(t+1-win)(2C_{win-1}^1+4C_{win-1}^2)^3)$.

3.3 The Adaptive Reference Vector Regulator Based on K-means Clustering

In order to make the algorithm quickly adapt to the change in objective space and track the true POF of new environment, an adaptive AC regulator is designed based on K-means clustering to guide the evolution of the population. Some appropriate reference vectors are obtained from the current population by the AC regulator, which can eliminate invalid reference vectors and generate reference vectors in the effective region after the environmental change. In the process of adjusting the reference vector by the K-means clustering, the main idea is to cluster the solutions according to similarity, and take the centroids of different clusters as reference points to form the reference vectors. In this paper, the acute angle is used as the similarity measure of clustering, and the number of clusters is equal to the population size N. The pseudo code for the AC regulator is shown in Algorithm 3.

In Algorithm 3, the AC regulator starts from the initialization of the centroids of the clusters in line 1. For the population S, the principle of centroid search is to distribute the centroid as evenly as possible, so as to minimize the similarity between clusters. In this algorithm, the initial centroid set $\boldsymbol{c}$ consists of two types of centroids, which are the boundary centroid $\boldsymbol{c}_1$ and the conventional centroid $\boldsymbol{c}_2$, and they indicate the boundary cluster set C_1 and conventional cluster set C_2, respectively. Specially, C_1 is used to maintain the boundary information of the population. The initialization of the two types of centroids adopts the following Strategy 1 and Strategy 2, respectively.

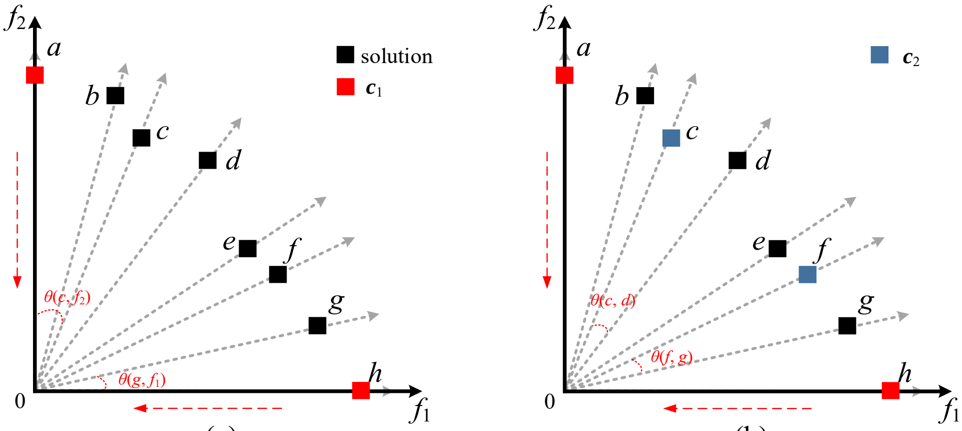
Strategy 1 selects M boundary centroids from the population S, where M is the number of objectives. The principle of selection is to pick the solution with the smallest angle to the objective axis from S. The ith centroid $x^{c_1(i)}$ in $\boldsymbol{c}_1$ is selected as follows:

$$x^{c_1(i)} = \arg\min_{x \in S} \theta(x, e^i) \tag{11}$$

where $\theta(x, e^i)$ is the acute angle value between solution x and the ith axis e^i. The remaining N-M conventional centroids are selected by Strategy 2. In Strategy 2, the separation distance $\delta(x)$ of each solution x is calculated except the boundary centroids, and it is calculated by Eq. (12), which represents the minimum angle between the solution x and other solutions in population S.

$$\delta(x) = \min_{x, y \in S} \theta(x, y) \tag{12}$$

After obtaining the separation distance, the solutions in S are sorted according to the descending order of $\delta(x)$, and the former N-M solutions are taken as the initial conventional centroids in $\boldsymbol{c}_2$. The selection of the two types of the initial centroids is shown in Fig. 4. In the bi-objective space, 4 initial centroids need to be selected from the 8 solutions. First, two boundary centroids are selected according to Eq. (11), i.e., the red points a and h in Fig. 4 (a), which have the minimum angle with the axes f_2 and f_1, respectively. Then, two conventional centroids are selected according to Strategy 2, i.e., the blue points c and f in Fig. 4 (b), which have the smallest angle with other solutions except the boundary centroids. The four selected initial centroids are used for the K-means clustering.



(a) (b)

Fig. 4 The initialized two types of centroids

After getting the initial centroids, the population *S* is iteratively clustered based on K-means, and the final centroids of each cluster are preserved. In lines 2-4, the initial cluster set *C* is got by angle-based similarity. In this process, each solution x_j in *S* is classified into a cluster in *C* according to Eq. (13) in line 3, and the corresponding cluster ID cl_j of the solution x_j is retained, where $cl_j = k$. Eq. (13) is defined as follows:

$$C^k = \{k : \arg\min_{k,j=1\sim N} q\ (x_j, c_k)\} \tag{13}$$

where $q\ (x_j, c_k)$ represents the angle between the *i*th solution x_j and the *k*th centroid c_k.

Algorithm 3 *AC*(*S*, *N*, *M*)

```
1: Find $N$ initial centroids $\boldsymbol{c} = \boldsymbol{c}_1 \cup \boldsymbol{c}_2$ of clusters in $C = C_1 \cup C_2$
2: for each solution $x_j \in$ S
3:     $x_j$ is classified into a cluster of $C$ by Eq. (13) and get the corresponding cluster ID $cl_j \in \boldsymbol{cl}$
4: end for
5: Set $flag$ = false, $times$ = 0
6: while $flag$ = false || $times \le 4M$ do
7:     Obtain the new centroids $\boldsymbol{c}_2$ of clusters in $C_2$ based on Eq. (14)
8:     Update the clusters in $C_2$ and cluster ID $c_j$ of each solution $x_j$ in $S$ like lines 2-4
9:     if not $\exists\ cl_j \in \boldsymbol{cl}$ changes
10:        $flag$ = true
11:    end if
12:    $times$++
13: end while
14: Get new vector matrix $W$ by projecting all centroids in $\boldsymbol{c}$ to the unit hypersphere
15: return $W$
```

Next, the AC regulator enters the adaptive clustering process. This process is stopped when the mark *flag* is **true**. In addition, the number of clustering iterations is set to 0 in line 5, and the maximum *times* is 4*M* (*M* is the number of objectives), which is used to avoid redundant clustering and unnecessary resource consumption. In the iterative process in lines 6-13, the algorithm first updates the centroid of each cluster in the conventional cluster set C_2 according to Eq. (14) in line 7, and the centroid update formula is as follows:

$$f_j(c_k) = \frac{\sum_{x \in S^k} f_j(x)}{|S^k|} \tag{14}$$

where $f_j(c_k)$ is the jth objective value of the centroid c_k and S^k is the set of all solutions in the kth cluster. In line 8, each cluster in C_2 and the cluster ID cl_j of each solution are updated by using the same method as lines 2-4. In lines 9-11, if it is detected that there are no solutions with changed cluster ID in S, the stop mark *flag* is set to **true**. In the clustering process, the number of iterations *times* is added by 1 after each clustering in line 12. The whole process of adaptive clustering is the alternating update of centroids and clusters, and it should be noted that in order to effectively preserve the boundary information, the boundary centroids in $\boldsymbol{c}_1$ are fixed after they are first obtained in line 1. When the clustering process is completed, all centroids are projected onto the unit hypersphere in line 14, and new reference vectors in W are obtained.

For the AC regulator, the acquisition of N initial centroids requires a time complexity of $O(M|S|^2)$, where M is the number of objectives, and $N \leq |S| \leq 2N$. For the associating procedure of lines 2-4 in Algorithm 3, the time complexity is also $O(M|S|^2)$. To finish the vectors adjustment of centroids, $O(M|S|^2\lambda)$ is required, where $\lambda = \min\{4M, \sigma\}$, and $\sigma \geq 1$ indicates the number of times that the centroids need to be adjusted in order to reach a stable state (i.e., *flag* = **false**).

3.4 Environmental Selection

In the prediction-based method, the effectiveness of the response strategy is related to the quality of the historical population, so it is important to acquire the final population with good convergence and diversity in each environment. Here, a simple and effective environmental selection scheme based on reference vector is designed in this paper. Regarding the diversity, the algorithm selects individuals by associating solutions to reference vectors in W. Considering the convergence, the solution with the best convergence index is added to the population one by one. In line 1 of Algorithm 4, the population P is set to empty. In lines 2-4, all the solutions x_i in the union population U are associated to one reference vector w_j in W ($i = 1$~$|U|$, $j = 1$~N). The binding method is similar to the process of classifying solutions to clusters. In line 6, for each vector w_j in W, the set B^j of all solutions associated to w_j is obtained as follows:

$$B^j = \{x_i : \arg \min_{x_i \in U, w_j \in W} q\ (x_i, w_j)\} \tag{15}$$

where $q\ (x_i, w_j)$ is the angle between the solution x_i and the vector w_j.

Algorithm 4 *EnvironmentalSelection*(U, W, N)

```
 1: Set $P = \varnothing$
 2: for each solution $x_i \in U$
 3:     Associate $x_i$ to a reference vector $w_j$ in $W$ based on Eq. (15)
 4: end for
 5: for each reference vector $w_j \in W$
 6:     Obtain the set $B^j$ of solutions associated to $w_j$
 7:     if $|B^j|$ != 0
 8:         Calculate the Chebyshev distance $D_{ch}$ of each solution $x^j_i$ in $B^j$
 9:     end if
10: end for
11: while $|P|<N$ do
12:     Add the solution with the minimum $D_{ch}$ in $B^j \in B$ to population $P$, and remove it from $B^j$
13: end while
14: return $P$
```

Next, in line 7 of Algorithm 4, if the number of solutions in the binding set B^j is not 0, the Chebyshev distance [34] of each solution in B^j is calculated in line 8, which is defined by:

$$D_{ch}(x_i) = \max_{l=1\sim M} w_j^l \left| f_l(x_i) - z_l \right| \tag{16}$$

where $f_l(x_i)$ is the lth objective value and z_l is the lth dimension of the ideal point. In the case of the same diversity, the solution with smaller Chebyshev distance is often preferred due to its better convergence. In lines 11-13, if the number of solutions in the population P is less than N, the solution with the minimum Chebyshev distance will be selected one by one and circularly from the binding set B^j and added into the population P in line 12. Then, the selected solution is removed from the set B^j. Here, the one-by-one selection means that only one solution is selected from a binding set at a time, and the number of solutions in P is updated immediately after a solution is added into P. The cyclic selection refers to the selection of solutions from B in turn ($B = \{j = 1 \sim N \mid B^j\}$), i.e., if the number of solutions in P is still less than N after a round of selection, a new round of selection will be started from B until the size of population P reaches N. The purpose of cyclic selection is to avoid invalid selection when the binding set is empty. Finally, the evolutionary population P containing N solutions is outputted in line 14. The environmental selection strategy designed according to the reference vector is a decomposition-based method, which considers the diversity and convergence of the population comprehensively, and some good solutions can be preserved for each environment.

4. The Experimental Studies

4.1 Related Experimental Settings

4.1.1 Benchmark Problems and Compared DMOEAs

In our experimental studies, the benchmark suite DF [32] with various changes are chosen for the performance evaluation of the proposed algorithm, which basically comes from some static optimization problems or some complex variants of other benchmarks, such as FDA [9], dMOP [35], ZJZ [17] and JY [36]. The DF test suite consists of 9 bi-objective problems DF1~DF9 and 5 tri-objective problems DF10~DF14. Specially, DF2 is part of the type I DMOP in which the POS changes but the POF is invariable, while DF1, DF3~DF9 and DF10~DF14 belong to the type II DMOP in which both the POS and the POF are constantly changing. In these test problems, the mapping relationship of functions and the shape of the corresponding POFs involve many complex dynamic properties, such as variable links, irregular shapes and time-dependent geometry, and they are widely used to test the performance of DMOEAs.

In order to validate the performance of our proposed method, five competitive DMOEAs are used as the comparison algorithms, which are briefly introduced below.

1). *PPS* [17]: This algorithm is a representative algorithm that uses the technique of linear regression to deal with DMOPs. In the process of evolution, the optimal solutions are divided into two parts: population center and manifold, in which center is used to determine the location of the population, and manifold implies the outline information of the population. When environment changes, PPS uses the autoregressive model (AR) and the preserved historical center sequences to predict the new population center. Similarly, the algorithm estimates the new manifold by calculating the similarity and variance between the historical manifolds, and the initial population of the new environment is generated based on the new center and manifold.

2). *SGEA* [18]: The main idea of SGEA is to respond to environmental changes in a steady-state manner. When the environmental change is detected, the initial population of the new environment generated by SGEA consists of two kinds of solutions, half of which are derived from the well-distributed solutions in the old environment, and the other half are predicted based on the movement direction and step size of individuals collected from the historical environments. In the process of optimization, the evolutionary population interacts with an external archive to improve the convergence speed and help the algorithm to adapt to changes quickly, so as to provide good tracking capability.

3). *MOEAD-SVR* [37]: MOEA/D-SVR uses the predictor of support vector regression (SVR) to generate the initial population in the new environment. As a representative prediction model of nonlinear regression, SVR plays a great role in exploring the excellent solutions of the new environment. In addition, MOEA/D-SVR combines with the decomposition-based multi-objective evolutionary algorithm MOEA/D, and each sub-problem will train a SVR predictor. A large number of experiments show the advancement of MOEA/D-SVR. However, due to the characteristics of SVR model, MOEA/D-SVR depends on the quality of historical populations, so there is still room for improvement in the performance of MOEA/D-SVR in the early stage of evolution.

4). *KT-DMOEA* [38]: KT-DMOEA uses transfer learning to cope with changes. This algorithm extracts representative solutions to reduce the occurrence of negative transfer, which are the knee points of the population. In order to make full use of this information, KT-DMOEA establishes a trend prediction model for knee points firstly, which estimates the distribution of the knee points in the new environment according to the knee points sets obtained from the most recent two environments in history. Then, KT-DMOEA uses the unbalanced TrAdaBoost algorithm to improve the accuracy of the estimated knee points. The initial population of the new environment will be composed of these new knee points and their Gaussian noise points.

5). *IT-RM-MEDA* [39]: IT-RM-MEDA is also an algorithm based on transfer learning, which uses a pre-search strategy to screen out some high-quality individuals with good diversity, and the guided population composed of them is used as the target domain of transfer. At the same time, IT-RM-MEDA trains a strong classifier through the TrAdaBoost algorithm, which identifies the outstanding solutions from a large number of sampled solutions to form the initial population of the new environment. IT-RM-MEDA combines many strategies, which not only maintains the advantages of transfer learning, but also avoids the occurrence of negative transfer, so as to greatly improve the quality of solutions and the speed of convergence.

4.1.2 Performance metrics and Experimental Settings

When assessing the performance of solutions obtained by different DMOEAs, the following two performance metrics are adopted in this work, which can evaluate the performance of compared algorithms in terms of convergence and diversity.

1). *Mean Inverted Generational Distance* (*MIGD*) [40]: The MIGD is a widely used performance indicator which can reflect the convergence and diversity of the population simultaneously. At time *t*, supposing that POF_t^* is a set of uniformly distributed points in the true POF, and POF_t is the corresponding approximation POF obtained from a DMOEA, IGD can be calculated as:

$$\mathrm{IGD}(POF_t^*, POF_t) = \frac{\sum_{x \in POF_t^*} \min_{y \in POF_t} d(x, y)}{|POF_t^*|} \tag{17}$$

where $d\,(x, y)$ refers to the Euclidian distance between x and y, and $|POF_t^*|$ is the number of points in POF_t^*. Then the MIGD is defined as follows：

$$\mathrm{MIGD} = \frac{\sum_{t \in T} \mathrm{IGD}(POF_t^*, POF_t)}{|T|} \tag{18}$$

where T is a set of discrete time instances and $|\,T\,|$ is the total number of changes in a run. In general, the smaller MIGD value indicates the better performance.

2). *Mean Schott's spacing metric* (*MSP*) [41]: The SP metric [42] measures the uniformity of population. A smaller SP value reflects a more uniform distribution of the solution obtained by the algorithm. This metric is calculated as:

$$\mathrm{SP}(POF_t)=\sqrt{\frac{1}{|POF_t|-1}(\sum_{i=1}^{|POF_t|}(D_i-\overline{D}))} \tag{19}$$

where D_i is the Euclidean distance between the ith point in POF_t and its nearest point, and $\overline{D}$ represents the average value of all D_i. The MSP can be denoted by:

$$\mathrm{MSP}=\frac{\sum_{t\in T}\mathrm{SP}(POF_t)}{|T|} \tag{20}$$

In this paper, all the considered algorithms are independently run 20 times on each test problem, and other general experimental parameters are configured according to the original references, which are summarized as follows.

1). *Dynamic Test Settings*: For the selected benchmark problems, the frequency of change τ_{T} and the severity of change n_{T} are considered to be set to 5 and 10, and four sets of experiments are set up by (τ_{T}, n_{T}) as (5, 5), (5, 10), (10, 5) and (10, 10). In all problems, the number of decision variables is set to 10.

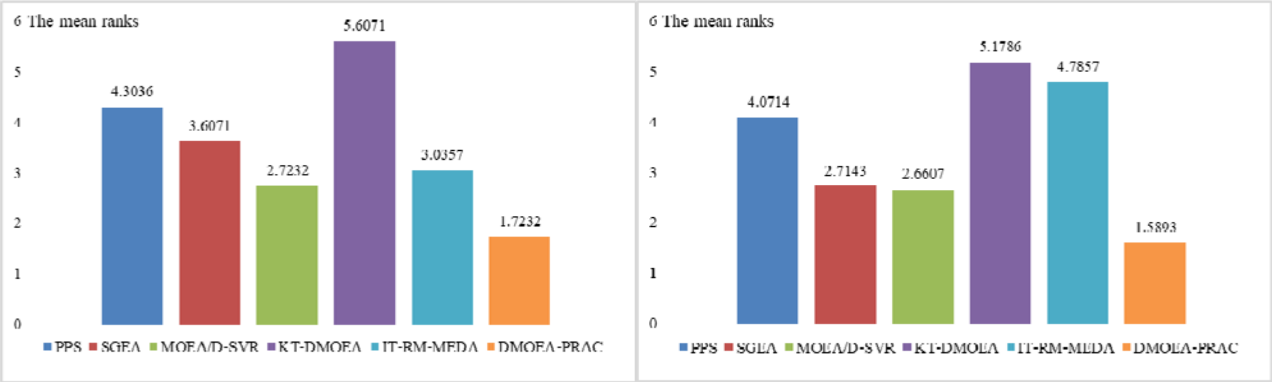
2). *Population size and termination condition*: The population size N is set to 100 and 300 respectively when the objective number is set to two and three. Each run consists of 100 environmental changes, and the termination condition of all algorithms is that the maximum number of evolutions G is no more than $\tau_{\mathrm{T}} \times 100$.

3). Settings for reproduction operators: In the SBX operator and PM operator [43], the probability of crossover p_c and mutation p_m are respectively set to 1.0 and $1/d$, where d is the number of decision variables. The distribution indexes for SBX and PM are all set to 20. In DE operator [33], the crossover probability CR and the scaling F are set to 0.5 and 1.0, respectively.

4). *Other Parameters settings*: In DMOEA-PRAC, the condition of reference vector adjustment is that G **mod** τ_{T} equals to 3 and 5 when τ_{T} = 5 and 10, which means that there is one adjustment in the middle of each environment. In addition, the size of time window *win* used to obtain the training data in DMOEA-PRAC is set to 3.

4.2 Comparison with Several Competitive DMOEAs

The statistical MIGD and MSP results of the six DMOEAs on all DF problems are respectively summarized in Table 1 and Table 2, in which both the mean value and standard deviation (in parenthesis) are provided. For each test problem, the best results, i.e., the smallest MIGD and MSC values among these six DMOEAs, are highlighted in bold with a gray background. Here, the multi-comparison procedure based on Friedman test [44] is used to test the statistical significance of the results, For the MIGD metric and MSP metric, the calculated p values of the test are 1.3230e-29 and 6.0322e-32, respectively, both of which are less than 0.05, which means that the performance of the six DMOEAs is considered to be significantly different at the 5% significance level. In addition, Fig. 5 (a) and Fig. 5 (b) show the global mean ranks of the six DMOEAs under the MIGD metric and the MSP metric, respectively. It is obvious that the proposed DMOEA-PRAC achieves the best ranks in both performance metrics among the six DMOEAs. Moreover, the pair-wise comparisons based on Wilcoxon rank-sum test [45] at the 5% significance level is applied to verify the statistically significant differences between the results of DMOEA-PRAC and its five competitors. In the following tables, the symbols "+", "=" and "-" represent that the metric values of the corresponding algorithm are significantly better than, similar to and worse than that of DMOEA-PRAC, respectively. In addition, the population distributions of the six algorithms corresponding to each environment in some typical DMOPs are given, and the characteristics and performance of these simulations are discussed and analyzed in detail.



(a) (b)

Fig. 5 The mean ranks of the six DMOEAs under the MIGD metric (a) and the MSP metric (b)

4.2.1 The MIGD metric values and The MSP metric values

From the MIGD values shown in Table 1, it can be seen that the performance of the DMOEA-PRAC proposed in this paper on 14 DF test problems is obviously better than that of the five comparison DMOEAs. Among the 36 test cases of 9 bi-objective problems DF1~DF9, DMOEA-PRAC, MOEA/D-SVR and IT-RM-MEDA perform best in 25, 3 and 8 cases, respectively, while the performances of PPS, SGEA and KT-DMOEA on all bi-objective problems are worse than the first three DMOEAs. For the 5 tri-objective problems (DF10~DF14), DMOEA-PRAC performs best in 9 out of 20 test cases. SGEA, MOEA/D-SVR and IT-RM-MEDA perform best in 4, 3 and 4 cases respectively, while PPS and KT-DMOEA still do not obtain the best MIGD results in all tri-objective cases. In general, the proposed DMOEA-PRAC performs best in 34 out of 56 test cases, while PPS, SGEA, MOEA/D-SVR, KT-DMOEA and IT-RM-MEDA gain the best MIGD values in 0, 4, 6, 0 and 12 cases. From the one-by-one comparisons in the last row of Table 1, DMOEA-PRAC performs better than PPS, SGEA, MOEA/D-SVR, KT-DMOEA and IT-RM-MEDA in 48, 46, 40, 56 and 43 out of 56 cases, respectively, while it is only outperformed by these five competitors in 5, 8, 11, 0 and 12 cases, respectively. Obviously, DMOEA-PRAC shows the best performance for DF problems.

Regarding DF1 and DF3, they have similar POF shapes in one period, i.e., the shape of POF continually changes among convex, linear and concave POFs in the optimization process. In the decision space, the change of POS in DF1 is translation, and the step of each translation conforms to the property of trigonometric function, while the changes of POS in DF3 include translation and rotation. For these two similar problems, DMOEA-PRAC gets the best MIGD results in 7 cases. When both values of τ_T and n_T are 10, the performance of DMOEA-PRAC on DF1 is slightly worse than that of MOEA/D-SVR, and its performance is similar to PPS and SGEA. Obviously, DMOEA-PRCA has good performance in solving the problem with constantly changing concave and convex PF. Especially when the frequency of change is fast (τ_T= 5), DMOEA-PRAC can quickly respond to the change.

Table 1

Mean and standard deviation values of MIGD obtained by DMOEA-PRAC and five competitors

Problems	τ_t, n_t	PPS	SGEA	MOEA/D-SVR	KT-DMOEA	IT-RM-MEDA	DMOEA-PRAC
DF1	5, 5	0.2230±6.52E-03(-)	0.1222±2.05E-02(-)	0.0435±7.40E-03(-)	0.1423±1.23E-02(-)	0.0299±1.35E-02(-)	**0.0062±6.82E-05**
	5, 10	0.0505±6.32E-03(-)	0.0300±1.28E-03(-)	0.0278±2.36E-03(-)	0.1120±9.90E-03(-)	0.0319±1.03E-02(-)	**0.0062±6.40E-05**
	10, 5	0.0549±1.46E-03(-)	0.0348±1.58E-03(-)	0.0119±2.94E-03(=)	0.1423±7.26E-03(-)	0.0358±1.70E-02(-)	**0.0107±4.51E-04**
	10, 10	0.0180±2.47E-03(=)	0.0166±2.30E-04(=)	**0.0089±4.74E-04(+)**	0.1120±1.04E-02(-)	0.0338±8.63E-03(-)	0.0112±5.16E-04
DF2	5, 5	0.2031±8.62E-03(-)	0.1794±1.06E-02(-)	0.0394±3.04E-03(-)	0.1906±7.96E-03(-)	0.0232±8.34E-03(-)	**0.0070±2.13E-04**
	5, 10	0.1054±2.92E-03(-)	0.1169±4.73E-03(-)	0.0360±2.84E-03(-)	0.1671±6.09E-03(-)	0.0194±7.73E-03(-)	**0.0070±1.86E-04**
	10, 5	0.0979±1.78E-03(-)	0.1032±4.71E-03(-)	0.0105±8.51E-04(-)	0.1151±8.23E-03(-)	0.0213±4.93E-03(-)	**0.0054±7.72E-05**
	10, 10	0.0846±1.07E-03(-)	0.0949±3.64E-03(-)	0.0107±9.77E-04(-)	0.1003±5.87E-03(-)	0.0220±8.50E-03(-)	**0.0054±5.17E-05**
DF3	5, 5	0.2915±2.50E-02(-)	0.3557±1.04E-02(-)	0.3509±2.26E-02(-)	0.4357±1.91E-02(-)	0.0414±6.86E-03(-)	**0.0063±4.70E-05**
	5, 10	0.2370±2.00E-02(-)	0.3167±1.20E-02(-)	0.3581±2.85E-02(-)	0.4555±1.57E-02(-)	0.0448±9.75E-03(-)	**0.0063±5.89E-05**
	10, 5	0.0691±2.84E-02(-)	0.3183±1.47E-02(-)	0.2641±4.28E-02(-)	0.4015±2.83E-02(-)	0.0487±2.44E-02(-)	**0.0052±3.26E-05**
	10, 10	0.0568±2.82E-02(-)	0.2887±1.89E-02(-)	0.2866±1.92E-02(-)	0.4228±3.02E-02(-)	0.0451±6.35E-03(-)	**0.0052±6.01E-05**
DF4	5, 5	0.5460±4.50E-02(-)	0.2709±1.14E-02(-)	0.0702±1.27E-03(-)	0.9880±2.78E-02(-)	0.2180±9.89E-02(-)	**0.0666±4.98E-04**
	5, 10	0.4418±6.09E-02(-)	0.2088±1.62E-02(-)	0.0807±9.28E-04(-)	1.0908±2.98E-02(-)	0.2203±5.61E-02(-)	**0.0774±9.76E-04**
	10, 5	0.1954±2.64E-02(-)	0.2129±4.09E-03(-)	**0.0665±3.86E-04(=)**	0.9598±2.45E-02(-)	0.2019±9.58E-02(-)	0.0669±3.91E-04
	10, 10	0.1231±1.92E-02(-)	0.1665±8.86E-03(-)	**0.0779±3.54E-04(+)**	1.0441±4.36E-02(-)	0.1876±8.13E-02(-)	0.0785±6.19E-04
DF5	5, 5	0.1650±3.03E-02(-)	0.0571±2.05E-03(-)	0.0500±1.87E-02(-)	1.7121±3.76E-02(-)	0.0176±7.05E-03(-)	**0.0064±6.90E-05**
	5, 10	0.0759±2.95E-02(-)	0.0319±6.62E-04(-)	0.0197±8.91E-04(-)	1.3587±3.38E-02(-)	0.0166±7.23E-03(-)	**0.0064±9.69E-05**
	10, 5	0.0295±9.42E-03(-)	0.0275±3.85E-04(-)	0.0080±3.51E-04(-)	1.6302±1.81E-02(-)	0.0244±1.05E-02(-)	**0.0053±3.95E-05**
	10, 10	0.0463±3.24E-02(-)	0.0221±2.29E-04(-)	0.0080±1.89E-04(-)	1.3093±1.76E-02(-)	0.0200±1.22E-02(-)	**0.0053±2.64E-05**
DF6	5, 5	12.3826±3.79E-01(-)	3.1475±2.38E-01(-)	2.1467±6.06E-01(-)	4.0323±3.02E-01(-)	26.2397±4.61E+00(-)	**0.0179±2.58E-03**
	5, 10	11.5324±2.00E-01(-)	1.1384±2.22E-01(-)	0.6055±7.25E-01(-)	4.7204±4.10E-01(-)	22.8553±6.51E+00(-)	**0.0160±2.09E-03**
	10, 5	10.0170±2.47E-01(-)	0.8712±1.93E-01(-)	1.9187±5.06E-01(-)	2.7908±3.34E-01(-)	23.4216±6.76E+00(-)	**0.0154±1.56E-03**
	10, 10	9.4320±3.09E-01(-)	0.5660±1.32E-01(-)	1.6981±5.83E-01(-)	3.5657±2.85E-01(-)	24.9518±7.95E+00(-)	**0.0127±1.67E-03**
DF7	5, 5	0.2643±8.89E-02(+)	2.4494±3.90E-02(+)	3.4112±9.36E-05(+)	8.6136±3.68E-01(-)	**0.0323±1.13E-02(+)**	3.9514±4.14E-02
	5, 10	0.1166±8.49E-03(+)	1.0887±2.07E-01(+)	0.8985±1.58E-03(+)	4.5473±3.01E-01(-)	**0.0316±5.01E-03(+)**	1.2797±3.80E-02
	10, 5	0.0864±6.91E-03(+)	2.3730±3.20E-02(+)	3.4114±4.53E-05(+)	8.2772±4.05E-01(-)	**0.0345±8.34E-03(+)**	3.8663±5.71E-02
	10, 10	0.0403±5.50E-03(+)	0.8504±2.50E-01(+)	0.8981±1.24E-04(+)	4.2866±1.96E-01(-)	**0.0300±9.33E-03(+)**	1.2197±1.74E-02
DF8	5, 5	0.0411±5.05E-03(-)	0.0997±3.03E-03(-)	0.0193±6.92E-04(+)	0.1966±7.40E-03(-)	**0.0065±5.81E-04(+)**	0.0212±9.66E-04
	5, 10	0.0430±5.84E-03(-)	0.0816±2.29E-03(-)	0.0196±6.24E-04(=)	0.2038±9.07E-03(-)	**0.0067±7.79E-04(+)**	0.0193±1.22E-03
	10, 5	0.0137±9.93E-04(+)	0.0934±1.01E-03(-)	0.0169±5.35E-04(+)	0.1733±7.20E-03(-)	**0.0069±6.79E-03(+)**	0.0204±8.86E-04
	10, 10	0.0134±7.40E-04(=)	0.0758±8.81E-04(-)	0.0168±6.49E-04(+)	0.1755±9.28E-03(-)	**0.0068±6.93E-04(+)**	0.0186±7.30E-04
DF9	5, 5	0.4400±1.73E-02(-)	1.0709±6.26E-02(-)	0.1022±4.79E-05(-)	1.6574±5.72E-02(-)	0.1894±5.75E-02(-)	**0.0051±9.75E-05**
	5, 10	0.3917±2.67E-02(-)	0.5109±3.69E-02(-)	0.0988±5.37E-05(-)	1.4471±3.22E-02(-)	0.1782±6.64E-02(-)	**0.0052±1.34E-04**
	10, 5	0.2377±1.05E-02(-)	0.5891±3.75E-02(-)	0.1019±1.70E-04(-)	1.4834±1.82E-02(-)	0.1662±3.42E-02(-)	**0.0042±6.14E-05**
	10, 10	0.1628±1.10E-02(-)	0.2479±1.12E-02(-)	0.0986±1.71E-04(-)	1.3754±2.52E-02(-)	0.2176±6.27E-02(-)	**0.0042±6.54E-05**
DF10	5, 5	0.3169±5.16E-03(-)	0.1077±1.15E-03(-)	0.0655±1.23E-03(-)	0.2095±1.36E-02(-)	**0.0213±2.58E-03(+)**	0.0378±5.27E-04
	5, 10	0.3097±5.16E-03(-)	0.0837±1.52E-03(-)	0.0660±8.83E-04(-)	0.2142±1.09E-02(-)	**0.0233±3.08E-03(+)**	0.0384±5.70E-04
	10, 5	0.2171±4.90E-03(-)	0.1048±9.31E-04(-)	0.0654±1.15E-03(-)	0.1865±8.52E-03(-)	**0.0229±4.10E-03(+)**	0.0383±6.06E-04
	10, 10	0.2106±4.76E-03(-)	0.0805±3.19E-03(-)	0.0649±1.91E-03(-)	0.1790±1.24E-02(-)	**0.0228±5.29E-03(+)**	0.0386±3.53E-04
DF11	5, 5	0.6968±2.75E-03(-)	0.6797±1.15E-03(-)	0.1328±1.34E-02(-)	0.6855±2.17E-03(-)	0.0737±2.03E-02(-)	**0.0264±1.32E-04**
	5, 10	0.7002±3.47E-03(-)	0.6711±4.36E-04(-)	**0.0261±8.40E-05(=)**	0.6866±1.39E-03(-)	0.1007±5.61E-02(-)	0.0266±8.89E-05
	10, 5	0.6745±1.65E-03(-)	0.6700±4.77E-04(-)	**0.0253±7.63E-05(+)**	0.6721±2.09E-03(-)	0.0841±2.39E-02(-)	0.0268±8.58E-05
	10, 10	0.6755±1.88E-03(-)	0.6616±3.86E-04(-)	**0.0252±8.77E-05(+)**	0.6710±2.22E-03(-)	0.0858±5.06E-02(-)	0.0269±8.05E-05
DF12	5, 5	0.6192±2.37E-02(-)	0.2091±7.49E-03(-)	0.3709±1.08E-02(-)	0.8799±2.15E-02(-)	0.9998±4.74E-04(-)	**0.0374±2.08E-04**
	5, 10	0.5482±1.22E-02(-)	0.2861±8.09E-03(-)	0.3415±1.45E-04(-)	0.8723±1.54E-02(-)	0.9995±7.60E-04(-)	**0.0373±1.80E-04**

	10, 5	0.4383±6.42E-03(-)	0.1990±6.33E-03(-)	0.3207±7.50E-03(-)	0.9021±1.41E-02(-)	0.9998±2.24E-04(-)	**0.0361±1.22E-04**
	10, 10	0.3822±1.00E-02(-)	0.3071±1.19E-01(-)	0.3071±1.05E-02(-)	0.8917±2.07E-02(-)	0.9995±4.63E-04(-)	**0.0359±1.47E-04**
DF13	5, 5	0.6392±6.27E-02(-)	**0.1708±1.42E-03(+)**	0.2600±2.55E-03(-)	1.5839±2.21E-02(-)	0.2202±1.25E-02(-)	0.1904±1.74E-03
	5, 10	0.4740±3.58E-02(-)	**0.1538±1.09E-03(+)**	0.2569±1.92E-03(-)	1.3185±1.40E-02(-)	0.2287±1.66E-02(-)	0.1892±2.73E-03
	10, 5	0.2250±1.13E-02(-)	**0.1511±7.06E-04(+)**	0.2728±1.82E-03(-)	1.5388±9.16E-03(-)	0.2171±1.49E-02(=)	0.2155±2.49E-03
	10, 10	0.2141±1.63E-02(=)	**0.1419±3.89E-04(+)**	0.2710±2.82E-03(-)	1.2768±4.60E-03(-)	0.2261±1.60E-02(-)	0.2155±4.71E-03
DF14	5, 5	0.2946±3.18E-02(-)	0.0591±9.25E-04(-)	0.0366±2.56E-04(-)	1.0280±8.71E-03(-)	0.0372±9.89E-03(-)	**0.0298±5.21E-04**
	5, 10	0.1442±1.76E-02(-)	0.0341±3.45E-04(-)	0.0383±2.50E-04(-)	0.8446±6.46E-03(-)	0.0421±6.83E-03(-)	**0.0309±2.96E-04**
	10, 5	0.0869±5.43E-03(-)	0.0476±5.64E-04(-)	0.0319±7.60E-05(=)	1.0125±3.08E-03(-)	0.0413±1.02E-02(-)	**0.0316±1.11E-03**
	10, 10	0.0744±5.94E-03(-)	0.0321±1.98E-04(=)	0.0336±7.50E-05(-)	0.8190±1.73E-03(-)	0.0365±4.77E-03(-)	**0.0321±3.15E-04**
best/*all*		0/56	4/56	6/56	0/56	12/56	34/56
+/-/=		5/48/3	8/46/2	11/40/5	0/56/0	12/43/1	--

The change of POS in DF2 is similar to that in DF1, but its POF is consistent. The switching of position-related variables in DF2 often brings difficulties to optimization. Nevertheless, in all test cases, DMOEA-PRAC outperforms all comparison algorithms. For DF4, the distribution of its POS in the decision space presents a radial shape, and the changes involve translation, rotation, expansion and scaling. In the 4 test cases of DF4, when τ_T= 10 and n_T= 5, the MIGD value of MOEA/D-SVR are the best, while DMOEA-PRAC and MOEA/D-SVR perform similarly. When τ_T= 10 and n_T= 10, DMOEA-PRAC is slightly worse than MOEA/D-SVR. From DF1 and DF4, it can be seen that the SVR prediction model is more dependent on the quality of the historical population, so MOEA/D-SVR can occasionally outperform DMOEA-PRAC when the number of static evolutions is large (τ_T=10). However, when the project faces the challenge of rapid environmental changes, DMOEA-PRAC can solve the problem more effectively.

Considering DF5 and DF6, the changes of their POS are also similar to that of DF1. What's more, their POF exists different numbers of knee points in different environments. The POF of DF5 is always distributed around the unit hyperplane in the objective space, while the POF of DF6 spans the entire objective space. On these two problems, DMOEA-PRAC gets the best MIGD results in all test cases. The changes of POS in DF7 and DF8 respectively depend on exponential function and trigonometric function, so the rotation characteristics are displayed in the decision space, i.e., their POSs may be symmetrical about certain points. IT-RM-MEDA achieves the best MIGD results on all instances of DF7 and DF8, which may be benefited from the effectiveness of the guided population provided by its pre-search mechanism in individual transfer. For the proposed DMOEA-PRAC, to a certain extent, the PR predictor used in decision space is more difficult to properly learn the characteristics of the exponential function, so the effect of DMOEA-PRAC is relatively poor. The changing POS of DF9 is grid-distributed in the decision space, and the POF of each environment is composed of broken line segments. On this discontinuous problem, the AC regulator designed in this paper can help the algorithm to explore the evolution direction quickly, so that the population can adapt to the new environment in time, and DMOEA-PRAC obtains the best MIGD results in all test cases.

For the 5 tri-objective problems, the change of the POS in DF10 is similar to that in the bi-objective problem DF7, and the shape of POF gradually changes from extreme convexity to extreme concavity. In the four test cases, DMOEA-PRAC is slightly worse than IT-RM-MEDA. For DF11 and DF12 with similar changes, their POSs have a common rotation point, and the POFs all change on the unit sphere. Among the 8 test cases of DF11 and DF12, DMOEA-PRAC performs slightly worse than MOEA/D-SVR on DF11 when τ_T= 10, but DMOEA-PRAC performs well in many cases as a whole. DF13 generates broken POFs in the process of change, and the number of truncated areas changes over time. Among all DMOEAs, SGEA obtains the best MIGD results in all cases of DF13. For DF14 with complex changes in POF, it may degenerate in the process of change, so the POF may be a curve or a continuous surface. When there is no degeneration, the size of the POF region and the distribution of the knee region will continue to change. The proposed DMOEA-PRAC gets the best MIGD results in all cases of DF14, which shows the superior performance in dealing with complex dynamic problems of three objectives.

Table 2

Mean and standard deviation values of MSP obtained by DMOEA-PRAC and five competitors

Problems	τ_t, n_t	PPS	SGEA	MOEA/D-SVR	KT-DMOEA	IT-RM-MEDA	DMOEA-PRAC
DF1	5, 5	0.0324±1.75E-03(-)	0.0243±2.31E-03(-)	**0.0070±3.72E-04(+)**	0.1936±3.78E-02(-)	0.1133±6.31E-02(-)	0.0096±1.73E-04
	5, 10	0.0140±1.49E-03(-)	0.0133±8.83E-04(-)	**0.0072±1.29E-04(+)**	0.1477±3.94E-02(-)	0.1606±5.86E-02(-)	0.0095±1.47E-04
	10, 5	0.0151±1.08E-03(+)	0.0112±8.01E-04(+)	**0.0056±1.00E-04(+)**	0.0633±1.73E-02(-)	0.1530±4.93E-02(-)	0.0199±2.32E-03
	10, 10	**0.0055±7.13E-04(+)**	0.0072±4.43E-04(+)	0.0057±1.16E-04(+)	0.0494±1.93E-02(-)	0.0954±2.09E-02(-)	0.0186±4.31E-03
DF2	5, 5	0.0295±1.77E-03(-)	0.0250±1.51E-03(-)	**0.0091±2.39E-04(+)**	0.2777±4.12E-02(-)	0.0475±2.27E-03(-)	0.0130±5.06E-04
	5, 10	0.0154±1.13E-03(-)	0.0132±5.79E-04(=)	**0.0091±2.72E-04(+)**	0.2088±3.52E-02(-)	0.0343±1.81E-03(-)	0.0128±2.40E-04
	10, 5	0.0167±1.22E-03(-)	0.0113±5.60E-04(-)	**0.0072±1.63E-04(+)**	0.1078±1.65E-02(-)	0.0222±1.68E-03(-)	0.0103±1.57E-04
	10, 10	0.0087±5.01E-04(+)	**0.0071±3.56E-04(+)**	0.0076±1.62E-04(+)	0.0743±2.10E-02(-)	0.0204±1.10E-03(-)	0.0103±1.64E-04
DF3	5, 5	0.1085±1.14E-02(-)	0.0376±6.82E-03(-)	0.0240±4.18E-03(-)	0.3061±5.25E-02(-)	0.0605±6.41E-03(-)	**0.0099±1.53E-04**
	5, 10	0.1068±1.28E-02(-)	0.0218±2.84E-03(-)	0.0134±4.38E-03(-)	0.2786±4.44E-02(-)	0.0563±7.44E-03(-)	**0.0099±2.13E-04**
	10, 5	0.0346±5.54E-03(-)	0.0169±1.74E-03(-)	0.0097±1.84E-03(-)	0.2076±5.02E-02(-)	0.1489±1.08E-01(-)	**0.0078±1.24E-04**
	10, 10	0.0282±4.42E-03(-)	0.0121±1.30E-03(-)	**0.0054±6.59E-04(+)**	0.1513±2.92E-02(-)	0.0940±4.57E-02(-)	0.0078±1.45E-04
DF4	5, 5	0.3285±1.78E-02(-)	0.1587±2.59E-02(-)	**0.0210±1.83E-03(+)**	0.3836±5.52E-02(-)	0.4795±5.40E-02(-)	0.0555±2.13E-02
	5, 10	0.2670±4.08E-02(-)	0.1411±1.90E-02(-)	**0.0221±1.96E-03(+)**	0.4094±5.72E-02(-)	0.3041±9.62E-02(-)	0.0969±3.98E-02
	10, 5	0.1359±2.31E-02(-)	0.1083±8.16E-03(-)	**0.0183±1.12E-03(=)**	0.3081±4.07E-02(-)	0.1146±1.51E-02(-)	0.0196±4.24E-03
	10, 10	0.0936±1.13E-02(-)	0.1049±1.09E-02(-)	**0.0199±1.32E-03(+)**	0.3365±3.33E-02(-)	0.0783±2.21E-02(-)	0.0210±7.44E-03
DF5	5, 5	0.0469±7.11E-03(-)	0.0345±2.39E-03(-)	0.1062±1.81E-02(-)	0.2109±2.69E-02(-)	0.1620±3.35E-02(-)	**0.0090±1.48E-04**
	5, 10	0.0320±9.61E-03(-)	0.0181±1.64E-03(-)	0.1089±1.83E-02(-)	0.1745±3.60E-02(-)	0.1507±5.79E-02(-)	**0.0089±1.75E-04**
	10, 5	0.0118±3.27E-03(-)	0.0147±1.68E-03(-)	0.0599±1.36E-02(-)	0.0757±9.88E-03(-)	0.3109±4.01E-02(-)	**0.0072±1.10E-04**
	10, 10	0.0137±6.24E-03(-)	0.0091±4.12E-04(-)	0.0940±2.17E-02(-)	0.0689±2.14E-02(-)	0.2418±3.44E-02(-)	**0.0072±8.50E-05**
DF6	5, 5	2.3765±7.79E-02(-)	3.4259±1.67E-01(-)	0.3123±1.30E-01(-)	10.4315±6.09E-01(-)	4.5064±2.11E-01(-)	**0.1507±5.21E-04**
	5, 10	2.3692±8.06E-02(-)	1.1496±1.94E-01(-)	0.2015±1.29E-01(-)	9.8499±7.97E-01(-)	4.2518±1.96E-01(-)	**0.1554±3.41E-04**
	10, 5	2.0398±6.68E-02(-)	0.5270±1.77E-01(-)	0.2068±3.63E-02(-)	6.9209±6.80E-01(-)	4.2179±2.14E-01(-)	**0.1339±1.82E-04**
	10, 10	2.0300±1.04E-01(-)	0.2399±2.97E-02(-)	**0.0967±4.23E-02(+)**	5.8996±5.52E-01(-)	3.8044±1.81E-01(-)	0.1381±2.74E-04
DF7	5, 5	0.1634±2.62E-02(-)	0.0254±8.19E-03(-)	0.0526±2.66E-04(-)	MAX(-)	MAX(-)	**0.0108±1.30E-03**
	5, 10	0.0605±4.96E-03(-)	0.0688±4.65E-02(-)	0.0522±8.55E-04(-)	MAX(-)	MAX(-)	**0.0105±1.50E-03**
	10, 5	0.0833±8.68E-03(-)	0.0285±1.44E-02(-)	0.0529±1.56E-04(-)	MAX(-)	MAX(-)	**0.0058±8.27E-04**
	10, 10	0.0309±2.70E-03(-)	0.0973±6.15E-02(-)	0.0528±1.33E-04(-)	MAX(-)	MAX(-)	**0.0056±7.66E-04**
DF8	5, 5	0.0457±2.71E-03(-)	0.0285±5.15E-03(-)	0.0139±7.92E-04(=)	0.2519±2.89E-02(-)	0.1681±1.60E-02(-)	**0.0131±3.21E-04**
	5, 10	0.0450±4.84E-03(-)	0.0310±1.90E-03(-)	**0.0132±4.55E-04(+)**	0.3116±4.95E-02(-)	0.1722±1.79E-02(-)	0.0150±6.39E-04
	10, 5	0.0172±1.62E-03(-)	0.0226±1.69E-03(-)	0.0131±2.37E-04(-)	0.1299±1.80E-02(-)	0.1308±1.55E-02(-)	**0.0124±2.87E-04**
	10, 10	0.0214±3.04E-03(-)	0.0257±3.22E-03(-)	**0.0131±3.02E-04(+)**	0.1391±2.39E-02(-)	0.1683±2.00E-02(-)	0.0141±3.79E-04
DF9	5, 5	0.1666±9.29E-03(-)	0.3835±3.51E-02(-)	**0.0058±4.20E-05(+)**	0.8699±1.01E-01(-)	0.4161±3.28E-02(-)	0.0180±7.95E-04
	5, 10	0.1515±9.33E-03(-)	0.2180±2.30E-02(-)	**0.0058±5.98E-05(+)**	0.9203±1.60E-01(-)	0.2790±2.08E-02(-)	0.0189±5.08E-04
	10, 5	0.1067±1.02E-02(-)	0.2101±1.77E-02(-)	**0.0061±1.57E-05(+)**	0.6341±9.46E-02(-)	0.2498±4.02E-02(-)	0.0124±5.42E-04
	10, 10	0.0790±5.58E-03(-)	0.1178±1.27E-02(-)	**0.0061±2.86E-05(+)**	0.6838±5.47E-02(-)	0.1251±2.00E-02(-)	0.0128±8.47E-04
DF10	5, 5	0.3605±1.39E-02(-)	0.0340±3.52E-03(-)	0.0248±1.07E-03(-)	0.2173±2.11E-02(-)	0.1405±2.55E-02(-)	**0.0214±1.64E-04**
	5, 10	0.3482±1.50E-02(-)	0.0365±4.00E-03(-)	0.0241±4.32E-04(-)	0.2011±2.06E-02(-)	0.1503±1.39E-02(-)	**0.0213±2.74E-04**
	10, 5	0.2303±1.39E-02(-)	0.0317±3.23E-03(-)	0.0240±7.15E-04(-)	0.1150±7.78E-03(-)	0.1565±3.16E-02(-)	**0.0204±2.45E-04**
	10, 10	0.2208±1.19E-02(-)	0.0334±3.68E-03(-)	0.0243±1.07E-03(-)	0.1089±1.04E-02(-)	0.1518±1.88E-02(-)	**0.0204±1.97E-04**
DF11	5, 5	0.0371±6.93E-04(-)	0.0180±8.46E-04(-)	0.0203±1.08E-04(-)	0.0385±1.11E-03(-)	0.0741±5.80E-03(-)	**0.0148±1.02E-04**
	5, 10	0.0370±7.68E-04(-)	0.0162±5.14E-04(-)	0.0202±8.16E-05(-)	0.0390±7.92E-04(-)	0.0775±5.99E-03(-)	**0.0148±1.05E-04**
	10, 5	0.0337±4.61E-04(-)	0.0152±4.99E-04(-)	0.0200±9.17E-05(-)	0.0333±6.32E-04(-)	0.0706±5.22E-03(-)	**0.0142±5.57E-05**
	10, 10	0.0338±5.78E-04(-)	**0.0120±3.22E-04(+)**	0.0199±9.15E-05(-)	0.0331±3.19E-04(-)	0.0757±5.35E-03(-)	0.0143±6.80E-05
DF12	5, 5	0.6109±2.00E-02(-)	0.0609±5.47E-03(-)	0.2538±6.62E-02(-)	0.3640±6.09E-02(-)	**0.0000±2.13E-18(+)**	0.0252±1.34E-04

	5, 10	0.4951±2.13E-02(-)	0.0597±8.05E-03(-)	0.0763±1.26E-03(-)	0.3453±7.80E-02(-)	**0.0057±1.72E-02(+)**	0.0253±7.34E-05
	10, 5	0.4757±3.35E-02(-)	0.0523±5.76E-03(-)	0.2206±9.22E-02(-)	0.2818±7.86E-02(-)	0.2078±1.83E-01(-)	**0.0245±1.27E-04**
	10, 10	0.3778±1.65E-02(-)	0.0517±7.49E-03(-)	0.2486±1.68E-01(-)	0.2246±5.68E-02(-)	0.1132±1.40E-01(-)	**0.0244±1.10E-04**
DF13	5, 5	0.5929±2.67E-02(-)	0.0942±1.74E-02(-)	0.3610±4.51E-02(-)	0.3209±3.01E-02(-)	0.2132±1.19E-02(-)	**0.0477±3.46E-04**
	5, 10	0.5653±4.33E-02(-)	0.0617±8.09E-03(-)	0.3787±5.46E-02(-)	0.3062±2.06E-02(-)	0.1582±1.70E-02(-)	**0.0480±7.25E-04**
	10, 5	0.2577±3.21E-02(-)	0.0554±9.53E-03(-)	0.3579±1.31E-02(-)	0.1830±1.10E-02(-)	0.1625±2.20E-02(-)	**0.0430±1.13E-03**
	10, 10	0.2302±3.14E-02(-)	**0.0414±8.71E-03(+)**	0.3651±5.31E-02(-)	0.1564±1.48E-02(-)	0.1427±3.06E-02(-)	0.0429±8.70E-04
DF14	5, 5	0.2081±1.15E-02(-)	0.0174±6.36E-04(-)	0.3588±2.00E-02(-)	0.1342±9.55E-03(-)	0.2983±3.76E-02(-)	**0.0161±8.20E-05**
	5, 10	0.1111±1.75E-02(-)	**0.0139±3.65E-04(+)**	0.3539±1.97E-02(-)	0.1279±1.97E-02(-)	0.2423±3.90E-02(-)	0.0164±1.14E-04
	10, 5	0.0630±6.47E-03(-)	**0.0106±1.52E-04(+)**	0.3931±3.41E-02(-)	0.0790±8.35E-03(-)	0.3226±1.82E-02(-)	0.0154±8.13E-05
	10, 10	0.0472±8.78E-03(-)	**0.0090±2.42E-04(+)**	0.3676±4.64E-02(-)	0.0658±5.87E-03(-)	0.2848±2.66E-02(-)	0.0156±9.40E-05
best/all		1/56	6/56	18/56	0/56	2/56	29/56
+/-/=		3/53/0	8/47/1	19/35/2	0/56/0	2/54/0	--

Table 2 provides the MSP comparison results of all the considered DMOEAs on DF problems. Observing these results, DMOEA-PRAC shows the better distribution, as it gains the best results in 29 cases, while the other five algorithms are respectively best in 1, 6, 18, 0 and 2 cases. As shown in the last row of Table 2, DMOEA-PRAC performs better than its five competitors respectively in 53, 47, 35, 56 and 54 out of 56 instances, while it is only outperformed by them in 3, 8, 19, 0 and 2 instances, respectively.

In the 36 cases of 9 bi-objective problems, the distribution of MOEA/D-SVR is better than that of the proposed DMOEA-PRAC, as it gets 18 best MSP values, while DMOEA-PRAC gets 16 best values. This may be due to the uncertainty of the distribution of reference vectors adjusted by AC regulator in the early stage of evolution in a low-dimensional environment. The adjusted reference vectors may be dense in some areas, resulting in the aggregation of partial solutions. Among the six algorithms, PPS, SGEA, KT-DMOEA and IT-RM-MEDA obtain poor population distributions, as they get the best MSP values in 1, 1, 0 and 0 out of 36 cases, respectively. In addition, for KT-DMOEA and IT-RM-MEDA, their MSP on DF7 is very large and exceeds the order of magnitude of 1000, which is represented by MAX in this article.

In the tri-objective problems, the population distribution of DMOEA-PRAC is obviously better, as it obtains 13 of the best MSP values in 20 cases, while PPS, SGEA, MOEA/D-SVR, KT-DMOEA and IT-RM-MEDA get 0, 5, 0, 0 and 2 best MSP values, respectively. Generally speaking, the performance of the proposed algorithm is relatively good in terms of distribution.

4.2.2 Further Discussion and Analysis

In order to further study the performance of DMOEA-PRAC and comparison algorithms, several sets of simulation figures of six algorithms on DF2, DF3, DF5 and DF9 when τ_T= 5 and n_T = 10 are provided in Fig. 6, Fig. 7, Fig. 8 and Fig. 9, which show the final population distribution of each algorithm in some continuous environments. In these figures, the lines or surfaces composed of gray points represent the POFs of a problem in different environments, and the blue points represent the solutions of a final population obtained by a DMOEA in each environment.

For the type I dynamic problem DF2, among the six algorithms, PPS and SGEA have similar performance, as the populations obtained by them can only cover part of the POFs and cannot get converged in general, but the diversity of them is slightly better than that of KT-DMOEA and IT-RM-MEDA. The convergence and diversity of the populations generated by KT-DMOEA and IT-RM-MEDA are obviously poor, and they only obtain the convergent populations in a few changes. By comparison, MOEA/D-SVR and DMOEA-PRAC have better performance. However, in some environments, there is still a certain distance between the population obtained from MOEA/D-SVR and the corresponding POF, so the convergence of MOEA/D-SVR is slightly worse than that of DMOEA-PRAC. In addition, the population obtained by MOEA/D-SVR is difficult to detect solutions at both ends of some POFs, while the populations obtained by DMOEA-PRAC have some missing solutions near the extreme points, so both of them have some deficiencies in diversity.

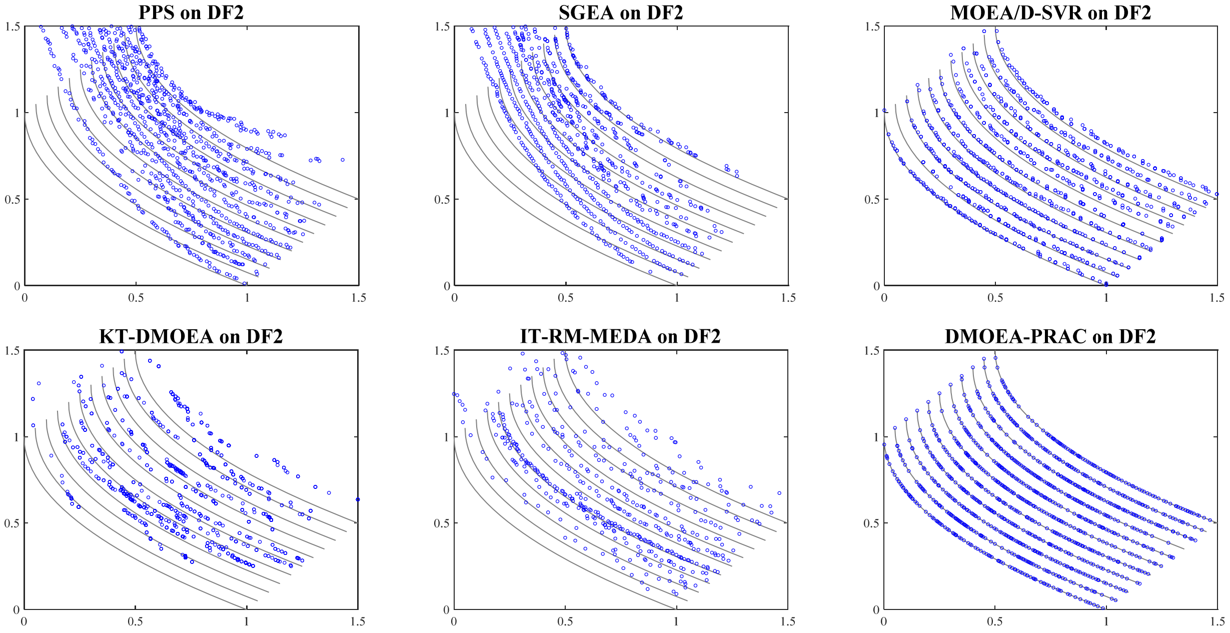
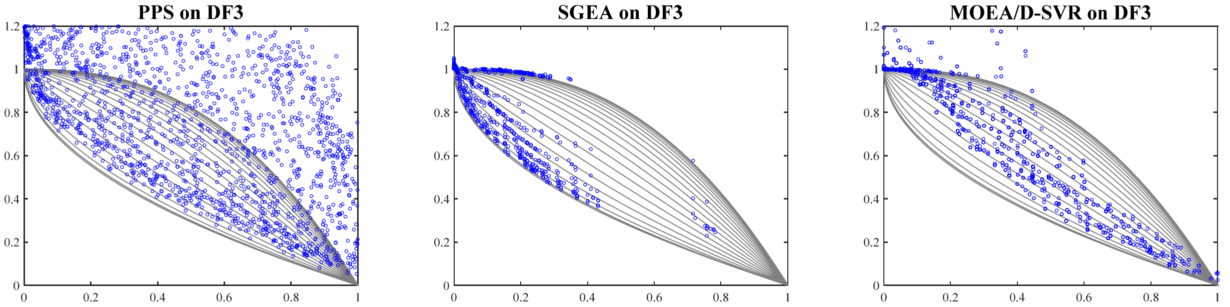


Fig. 6 Final solutions sets obtained by six DMOEAs on the DF2 problem



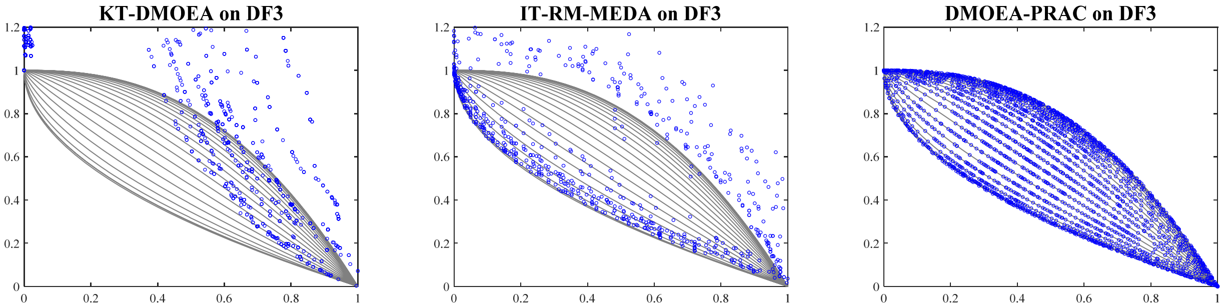


Fig. 7 Final solutions sets obtained by six DMOEAs on the DF3 problem

When solving the DF3 problem, it can be observed from Fig. 7 that the populations obtained by DMOEA-PRAC have good convergence and diversity as the solutions can be uniformly and neatly distributed on the POF corresponding to each environment. By contrast, in the changing environments, the populations obtained by PPS do not converge obviously, while the diversity of populations generated by SGEA is extremely poor, and the corresponding solutions are mainly distributed near an objective axis. For MOEA/D-SVR, KT-DMOEA and IT-RM-MEDA, the populations obtained from them have poor convergence and diversity, in which IT-RM-MEDA only converges in a few environments where the shapes of POFs are convex, but it cannot explore some effective solutions in other environments. According to the population distribution of each algorithm on DF3, it can be seen that for the DMOPs in which the shapes of POFs are constantly switching between convexity, linearity and concavity, the proposed DMOEA-PRAC can adapt to various environments, and can explore and exploit outstanding populations under many kinds of changes.

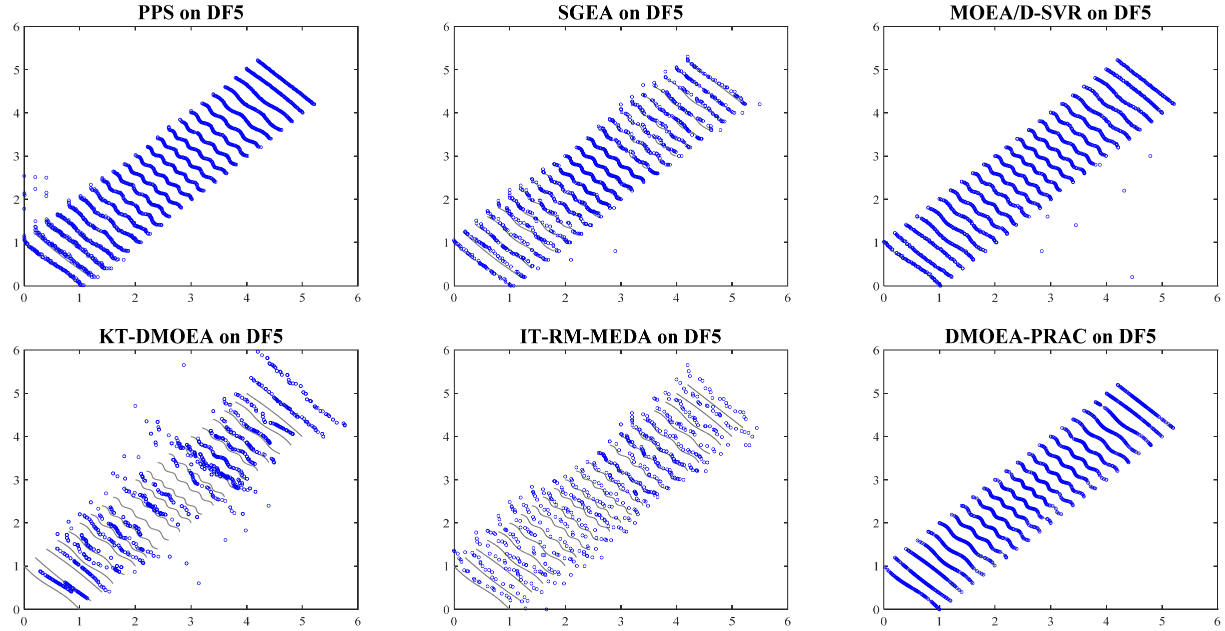


Fig. 8 Final solutions sets obtained by six DMOEAs on the DF5 problem

From the DF5 problem in Fig. 8, there are some constantly changing knee areas in its POFs during the change process. PPS, SGEA, MOEA/D-SVR and DMOEA-PRAC can all obtain populations with good convergence and diversity in each environment of the DF5 problem, but the populations obtained from the first three have some invalid solutions beyond the boundaries of POFs, and the performance of KT-DMOEA and IT-RM-MEDA is obviously inferior. For DF9, the changing POFs are discontinuous, and the number of line segments is continually changing. It can be observed from Fig. 9 that MOEA/D-SVR and DMOEA-PRAC can easily obtain populations with good convergence and diversity in each environment, while the corresponding populations of PPS, SGEA, KT-DMOEA and IT-RM-MEDA are far less than satisfactory. From the simulation of DF5 and DF9, apparently the suggested DMOEA-PRAC also has good performance in solving DMOPs with complex changes.

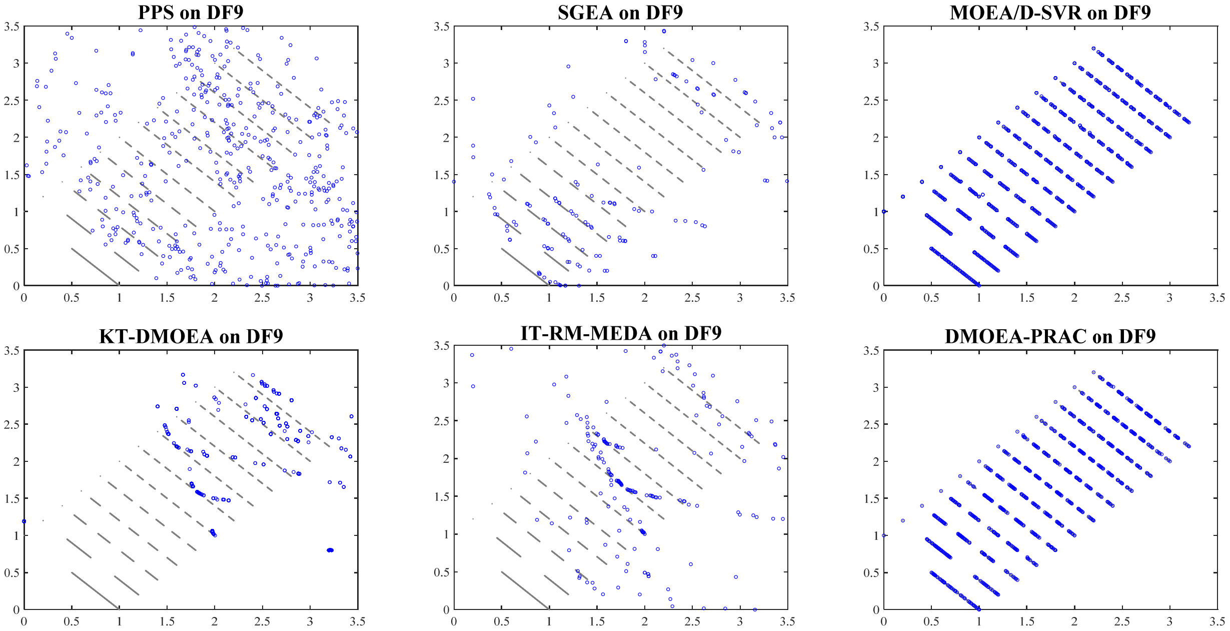


Fig. 9 Final solutions sets obtained by six DMOEAs on the DF9 problem

In order to decipher the influence of PR predictor and AC regulator in DMOEA-PRAC, some ablation experiments are carried out to verify their role in the algorithm. The statistical MIGD values of DMOEA-PR and DMOEA-AC on 7 representative problems (5 bi-objective problems and 2 tri-objective problems) are shown in Table 3, where DMOEA-PR and DMOEA-AC are two versions of DMOEA-PRAC that remove regulator and predictor, respectively. In DMOEA-PR, the algorithm only uses PR predictor to search the decision space after the change occurs, while DMOEA-AC only responds to the change by adjusting the reference vectors in the objective space by the AC regulator. In Table 3, the better MIGD value obtained by DMOEA-PR or DMOEA-AC in each instance is bold, while the best MIGD value obtained by the three algorithms is marked with a gray background. Among the 14 test cases, DMOEA-PRAC performs significantly better than the other two algorithms that use a single responder on 9 cases. DMOEA-AC which only uses the AC regulator performs better on the type I problem DF2, which may be due to the historical experiences of reference vector adjustment is beneficial to solving the problem in which the POF is fixed. For DF12 problem, DMOEA-AC also obtains the best results, but it is obvious that the difference between the three algorithms is small. Considering the algorithm using a single responder, DMOEA-AC using only the AC regulator is better in 9 cases, while DMOEA-PR using only the PR predictor is better in 5 test cases, which means that the AC regulator is more effective in improving the performance of the algorithm to a certain extent. Generally speaking, DMOEA-PRAC with two responders can solve dynamic problems better, as it takes into account the dynamic responses of the two spaces, and the PR predictor and the AC regulator jointly help the algorithm respond to changes and track the optimal solutions.

Table 3

Mean and standard deviation values of MIGD obtained by ablation experiments

Problems	τ_T, n_T	DMOEA-PR	DMOEA-AC	DMOEA-PRAC
DF1	5, 5	**0.0387±1.19E-04**	0.0918±4.46E-05	0.0062±6.82E-05
	5, 10	**0.0417±1.35E-04**	0.0799±3.05E-05	0.0062±6.40E-05
DF2	5, 5	0.0051±3.30E-04	**0.0049±8.73E-05**	0.0070±2.13E-04
	5, 10	0.0050±1.95E-04	**0.0049±2.27E-04**	0.0070±1.86E-04
DF4	5, 5	**0.3510±1.64E-03**	0.3956±1.48E-03	0.0666±4.98E-04
	5, 10	**0.3117±2.04E-03**	0.3922±1.67E-03	0.0774±9.76E-04
DF8	5, 5	0.1094±9.85E-03	**0.1037±1.64E-03**	0.0212±9.66E-04
	5, 10	0.1155±8.67E-03	**0.1145±1.84E-03**	0.0193±1.22E-03
DF9	5, 5	0.0474±3.68E-04	**0.0219±3.72E-04**	0.0051±9.75E-05
	5, 10	0.0599±6.28E-04	**0.0220±1.89E-04**	0.0052±1.34E-04
DF12	5, 5	0.0369±4.52E-03	**0.0352±6.89E-06**	0.0374±2.08E-04
	5, 10	0.0361±2.53E-03	**0.0323±4.34E-04**	0.0373±1.80E-04
DF14	5, 5	0.0610±2.30E-04	**0.0284±5.05E-05**	0.0298±5.21E-04
	5, 10	**0.0575±3.09E-04**	0.0593±5.78E-04	0.0309±2.96E-04

To evaluate the actual runtime of DMOEA-PRAC and 5 comparison algorithms, the average running times (in seconds: *s*) from 20 runs are plotted in Fig. 10, for DF1-DF14 problems with $\tau_T = 5$ and $n_T = 5$. It can be seen that SGEA takes the shortest time on 9 bi-objective problems, while PPS has the fastest running speed on 5 tri-objective problems. The running time of PPS is relatively stable on all problems, while the running times of the other five DMOEAs on tri-objective problems increase significantly, especially MOEA/D-SVR, KT-DMOEA and DMOEA-PRAC. For the four DMOEAs using machine learning-based responders, i.e., MOEA/D-SVR, KT-DMOEA, IT-RM-MEDA and DMOEA-PRAC, they are generally slower than the two classical algorithms PPS and SGEA, which is usually greatly affected by model training.

Considering the convergence of each algorithm in each environment, it depends on two factors: dynamic response and static optimization, whose complexities reflect the computational speed of each algorithm in each environment. For the proposed DMOEA-PRAC, the PR predictor and the AC regulator determine the worst time complexity; Thus, the overall complexity of DMOEA-PRAC is the larger one of $O(Nd(t+1-win)(2C_{win-1}^{1}+4C_{win-1}^{2})^{3})$ for running the PR predictor and $O(M|S|^2\lambda)$ for running the AC regulator, where N is the population size, d is the number of decision variables, t is the time instant, win is the size of the time window, M is the number of objectives, S is the number of solutions used for clustering and λ is the times of centroid adjustments. For PPS, the AR model determines the worst time complexity with $O(Ndk(3^1 C_k^1 + 3^2 C_k^2 + 3^3 C_k^3)^3)$, where 3 is the model degree and k is the length of history mean point series. For SGEA, its predictor is simple and the response strategy is based on quantifiable change rules; So its worst time complexity is the larger one of $O(MN^2)$ and $O(N^2\log N)$. For MOEA/D-SVR, the SVR-based model determines the worst time complexity between $O(Ndq(t\text{-}q)^2)$ and $O(Ndq(t\text{-}q)^3)$, where q is the parameter of sliding window. For KT-DMOEA, its worst time complexity is $O(N^2d)$. For IT-RM-MEDA, its worst time complexity is the larger value of $O(HNI_p)$ and $O(I_t dN^2)$, where H is the number of reference vectors, I_p is the number of iterations in presearch stage and I_t is the number of iterations for individuals transfer.

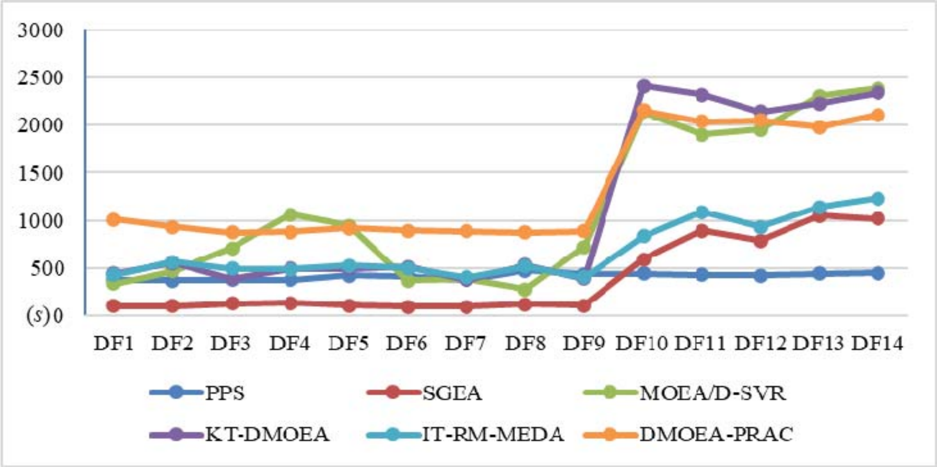


Fig. 10 The average running times of six DMOEAs on DF1-DF14 problems ($\tau_T = 5$, $n_T = 5$)

4.3 Application in controller design for dynamic system

In this section, the performance of the proposed DMOEA-PRAC in solving a parameter optimization problem based on the proportional-integral-derivative (PID) controller on unstable systems is discussed. PID controller is a common feedback loop component in industrial control applications, which consists of the proportional unit P, the integral unit I and the differential unit D, and its control quantity is a linear combination of error signal $e(t)$ on proportional, integral and differential. The output signal $u(t)$ of PID is calculated as follows:

$$u(t) = Kp[e(t) + \frac{1}{Ti}\int_0^t e(t)dt + Td\frac{de(t)}{dt}] \tag{21}$$

where Ti and Td are integral time constant and differential time constant, respectively. After Laplace transformation, Eq. (21) can be expressed as:

$$C(s) = Kp(1 + \frac{1}{Ti} \times \frac{1}{s} + Td \times s) \tag{22}$$

where *Kp*, *Ki* (*Kp*/*Ti*), *Kd* (*Kp*×*Td*) are the coefficients of proportional control, integral control and differential control, respectively. For any system, the algorithm needs to find these three corresponding parameters that make the PID controller work well.

In general, the performance of PID controller is usually evaluated from two aspects: the rise time *R* of the system and the maximum overshoot *O*, in which the rise time refers to the time from the initial time to the first time that the response output reaches the steady state value, and the maximum overshoot indicates the maximum degree to which the response output deviates from the steady state value during the adjustment process. In this paper, these two evaluation values of the PID controller are the two optimization objectives to be considered, and the faster rise time and the smaller overshoot are usually expected by a good PID controller. Here, the optimization problem is expressed as:

$$\min_{Kp, Ki \in (0.5, 5.0) \times (0.1, 1.0)} \{R(Kp, Ki, Kd), O(Kp, Ki, Kd)\} \tag{23}$$

For an unstable negative feedback control system, the parameters of the open-loop transfer function will change, so the parameters of each unit of the PID controller need to be constantly adjusted to achieve a satisfactory control effect for a specific system. In this paper, the parameter optimization of a garbage combustion controller [9] is discussed, and the transfer functions of the system *G*(*s*) and the controller *C*(*s*) are set as follows:

$$\begin{cases} G(s) = \dfrac{1.5}{50s^3 + a_2(t)s^2 + a_1(t)s + 1} \\ C(s) = Kp + Ki \times \dfrac{1}{s} + Kd \times s \end{cases} \tag{24}$$

where $a_2(t) = 43 + 30\sin(\pi t/18)$, $a_1(t) = 3 + 30\sin(\pi t/18)$, and *Kd* is set to 8.3317 according to the experience in [9].

The simulation of PID controller in this paper is based on the Control System Designer toolbox in MATLAB. The two decision variables *Kp* and *Ki* are encoded by real numbers, and the ranges of their values are (0.5, 5.0) and (0.1, 1.0), respectively. The value ranges of the two objectives *R* and *O* are set to (0, 100), and the values of the steady state are set to 1.0. It is worth noting that the parameters of PID controller are highly sensitive, and the law of its parameter adjustment is more complex, as the difference of system transfer function under different time steps increases with the transformation of higher-order terms, so this problem is an unpredictable DMOP to some extent.

In this paper, the performance of the proposed DMOEA-PRAC and two comparison algorithms MOEA/D-SVR and IT-RM-MEDA, which perform better in the benchmark problems, are tested on the parameter optimization problem of the dynamic PID controller. Fig. 11 shows the distribution of the non-dominated solutions obtained by the three algorithms in the objective space at *t* = 10, 12 and 16, respectively. The horizontal axis and the vertical axis represent the rise time and the maximum overshoot (unit: %) of the controller, respectively. In three environments, the solutions obtained by MOEA/D-SVR can converge uniformly, and the number of non-dominated solutions in the population is more than that of the other two algorithms, but its deficiency is that the search for boundary solutions is limited, and there are solutions that are extremely effective on one objective and poor on the other. In comparison, the diversity of the population obtained by IT-RM-MEDA is poor, especially at *t* =16. However, when *t* is 12, the optimization effect of IT-RM-MEDA on the rise time is better than that of the other two DMOEAs. For DMOEA-PRAC, although the diversity of the population from it is slightly worse than that of MOEA/D-SVR, its overall convergence performance is relatively stable. Generally speaking, the three algorithms can provide appropriate parameters for the dynamic PID controller, and in real life, the specific modeling design, the control of constraints and the selection of solutions may require some additional considerations.

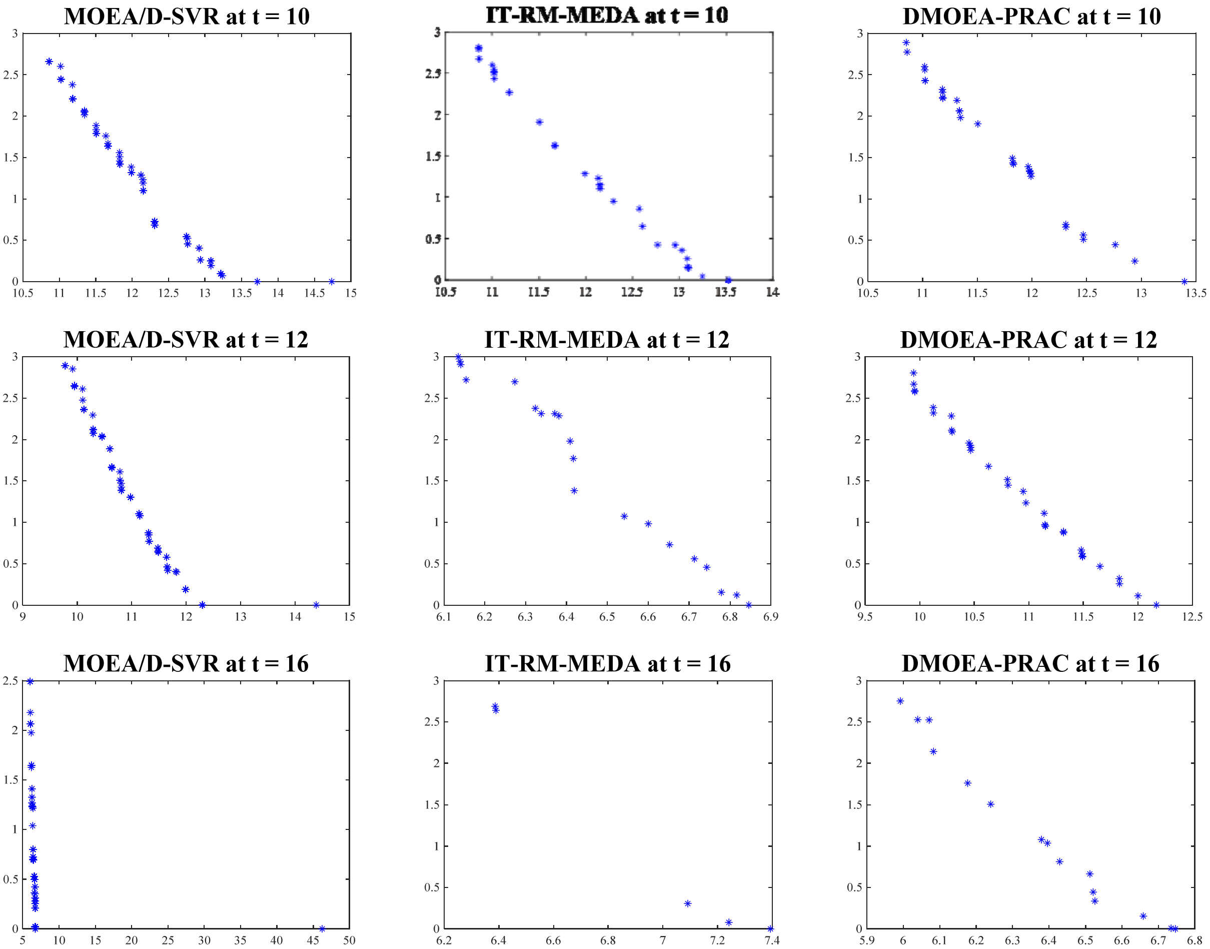


Fig. 11 The Distribution of the non-dominated solutions of the parameters of the PID Controller obtained by MOEA/D-SVR, IT-RM-MEDA and DMOEA-PRAC in the objective Space at t = 10, 12, 16.

5. Conclusions and Future Work

In this paper, a dynamic multi-objective optimization algorithm DMOEA-PRAC based on polynomial regression and adaptive clustering is proposed. When change occurs, the algorithm responds to the change in the decision space and objective space, and the corresponding PR predictor and AC regulator are introduced in the proposed DMOEA-PRAC, which can quickly and effectively track the true POS and POF corresponding to the new environment. The PR predictor is used to generate the initial population for the new environment, while the AC regulator is adopted to adjust the reference vectors. These two important components improve the performance of DMOEA-PRAC together. Finally, some recently proposed DMOEAs are selected to compare with DMOEA-PRAC, and the experimental results verify the effectiveness of DMOEA-PRAC in dealing with a variety of DMOPs.

In the future work, we will improve the efficiency and complexity of our algorithm and enhance its performance to solve more complex DMOPs with the changes in the number of objectives, decision variables, and constraints. Moreover, the application of DMOEAs on more real-world DMOPs will also be the focus of our future work.

Acknowledge

This work is the National Natural Science Foundation of China (NSFC) under Grant 61876110, the Joint Funds of the NSFC under Key Program Grant U1713212, the Shenzhen Scientific Research and Development Funding Program under Grant JCYJ20190808164211203, the Guangdong “Pearl River Talent Recruitment Program” under Grant 2019ZT08X603, and Shenzhen Science and Technology Innovation Commission (R2020A045). Prof. Coello Coello acknowledges support from the CONACyT project no. 1920, a 2018 SEP-Cinvestav Grant (application no. 4), and the Basque Government through the BERC 2022-2025 program by the Spanish Ministry of Science.

References

[1] P. P.-Y. Wu, D. Campbell, and T. Merz, “Multiobjective four-dimensional vehicle motion planning in large dynamic environments,” IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 41, no. 3, pp. 621–634, Jun. 2011.

[2] Z. Yang, Y. Jin and K. Hao, “A Bio-Inspired Self-Learning Coevolutionary Dynamic Multiobjective Optimization Algorithm for Internet of Things Services,” IEEE Transactions on Evolutionary Computation, vol. 23, no. 4, pp. 675-688, Aug. 2019.

[3] C. Yang and J. Ding, “Constrained dynamic multi-objective evolutionary optimization for operational indices of beneficiation process,” J. Intell. Manuf., vol. 30, (7), pp. 2701-2713, 2019.

[4] Y. N. Guo, J. Cheng, S. Luo, D. Gong and Y. Xue, “Robust Dynamic Multi-Objective Vehicle Routing Optimization Method,” IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 15,

no. 6, pp. 1891-1903, 1 Nov.-Dec. 2018.

[5] J. Eaton, S. Yang and M. Gongora, “Ant Colony Optimization for Simulated Dynamic Multi-Objective Railway Junction Rescheduling,” IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 11, pp. 2980-2992, Nov. 2017.

[6] R. Chen, K. Li and X. Yao, “Dynamic Multiobjectives Optimization With a Changing Number of Objectives,” IEEE Transactions on Evolutionary Computation, vol. 22, no. 1, pp. 157-171, Feb. 2018.

[7] S. U. Guan, Q. Chen, and W. Mo, “Evolving dynamic multi-objective optimization problems with objective replacement,” Artificial Intelligence Review, vol. 23, no. 3, pp. 267–293, 2005.

[8] Q. Chen, J. Ding, S. Yang and T. Chai, “A Novel Evolutionary Algorithm for Dynamic Constrained Multiobjective Optimization Problems,” IEEE Transactions on Evolutionary Computation, vol. 24, no. 4, pp. 792-806, Aug. 2020.

[9] M. Farina, K. Deb and P. Amato, “Dynamic multiobjective optimization problems: test cases, approximations, and applications,” IEEE Transactions on evolutionary computation, vol. 4, no. 5, pp. 425-442, 2004.

[10] K. Deb, N. U. B. Rao and S. Karthik, “Dynamic Multi-objective Optimization and Decision-Making Using Modified NSGA-II: A Case Study on Hydro-thermal Power Scheduling,” Proceedings of the 4th international conference on Evolutionary multi-criterion optimization (EMO'07). Springer-Verlag, Berlin, Heidelberg, 803–817, 2007.

[11] M. Greeff and A. P. Engelbrecht, “Solving dynamic multi-objective problems with vector evaluated particle swarm optimisation,” 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), pp. 2917-2924, 2008.

[12] S. Zeng, G. Chen, L. Zheng, H. Shi, H. D. Garis, L. X. Ding and L. S. Kang, “A Dynamic Multi-Objective Evolutionary Algorithm Based on an Orthogonal Design,” 2006 IEEE International Conference on Evolutionary Computation, pp. 573-580, 2006.

[13] Z. H. Zhang and S. Q. Qian, “Artificial immune system in dynamic environments solving time-varying non-linear constrained multi-objective problems,” Soft Computing, vol. 15, no. 7, pp. 1333–1349, July 2011.

[14] S. Sahmoud and H. R. Topcuoglu, “A Memory-Based NSGA-II Algorithm for Dynamic Multi-objective Optimization Problems,” Proceedings of the European Conference on the Applications of Evolutionary Computation, Porto, Portugal, vol. 9598, pp. 296-310, 2016.

[15] Z. P. Liang, S. X. Zheng, Z. X. Zhu and S. X. Yang, “Hybrid of memory and prediction strategies for dynamic multiobjective optimization,” Information Sciences, vol. 485, pp. 200-218, June 2019.

[16] M. Arrchana, K.C.Tan and V. Prahlad, “Evolutionary Dynamic Multiobjective Optimization Via Kalman Filter Prediction,” IEEE Transactions on Cybernetics, vol. 46, no. 12, pp. 2862-2873, 2016.

[17] A. Zhou, Y. Jin, and Q. Zhang, “A population prediction strategy for evolutionary dynamic multiobjective optimization,” IEEE Transactions on Cybernetics, vol. 44, no. 1, pp. 40-53, Jan. 2014.

[18] S. Jiang and S.Yang, “A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization,” IEEE Transactions on Evolutionary Computation, vol. 21, no. 1, pp. 65-82, 2016.

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

2017.

[20] S. Sahmoud and H. R. Topcuoglu, “Sensor-based change detection schemes for dynamic multi-objective optimization problems,” 2016 IEEE Symposium Series on Computational Intelligence (SSCI), 2016, pp. 1-8.

[21] Y. Wu, Y. C. Jin and X. X. Liu, “A directed search strategy for evolutionary dynamic multiobjective optimization,” Soft Computing, vol. 19, no. 11, pp. 3221 – 3235, Nov. 2015.

[22] K. Zhang, C. Shen, X. Liu and G. G. Yen, “Multiobjective Evolution Strategy for Dynamic Multiobjective Optimization,” IEEE Transactions on Evolutionary Computation, vol. 24, no. 5, pp. 974-988, Oct. 2020.

[23] R. H. Shang, L. C. Jiao, Y. J. Ren, L. Li and L. P. Wang, “Quantum immune clonal coevolutionary algorithm for dynamic multiobjective optimization,” Soft Computing, vol. 18, no. 4, pp. 743 – 756, Apr. 2014.

[24] C. Li and S. Yang, “A general framework of multipopulation methods with clustering in undetectable dynamic environments,” IEEE Transactions on Evolutionary Computation, vol. 16, no. 4, pp. 556–577, Aug 2012.

[25] K. Deb, A. Pratap and S. Agarwal et al., “A fast and elitist multiobjective genetic algorithm: NSGA-II,” IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182 – 197, Apr. 2002.

[26] R. Azzouz, S. Bechikh and L. B. Said, “A dynamic multi-objective evolutionary algorithm using a change severity-based adaptive population management strategy,” Soft Computing, vol. 21, no. 4, pp. 885 – 906, Feb. 2017.

[27] M. Rong, D. Gong, Y. Zhang, Y. Jin and W. Pedrycz, “Multidirectional Prediction Approach for Dynamic Multiobjective Optimization Problems,” IEEE Transactions on Cybernetics, vol. 49, no. 9, pp. 3362-3374, Sept. 2019.

[28] M. Rong, D. Gong, W. Pedrycz and L. Wang, “A Multimodel Prediction Method for Dynamic Multiobjective Evolutionary Optimization,” IEEE Transactions on Evolutionary Computation, vol. 24, no. 2, pp. 290-304, April 2020.

[29] F. Zou, G. Yen and L. Tang, “A knee-guided prediction approach for dynamic multi-objective optimization,” Information Sciences, vol. 509, pp. 193-209, Jan. 2020.

[30] S. B. Gee, K. C. Tan and C. Alippi, “Solving Multiobjective Optimization Problems in Unknown Dynamic Environments: An Inverse Modeling Approach,” IEEE Transactions on Cybernetics, vol. 47, no. 12, pp. 4223-4234, Dec. 2017.

[31] H. Zhang, J. Ding, M. Jiang, K. C. Tan and T. Chai, “Inverse Gaussian Process Modeling for Evolutionary Dynamic Multiobjective Optimization,” IEEE Transactions on Cybernetics (early access), 2021. doi：10.1109/TCYB.2021.3070434.

[32] S. Jiang, S. Yang, and X. Yao et al., “Benchmark problems for IEEE CEC 2018 Competition on Dynamic Multiobjective Optimization,” IEEE Congress on Evolutionary Computation 2018, July 2018.

[33] H. Li and Q. Zhang, “Comparison between NSGA-II and MOEA/D on a set of multiobjective optimization problems with complicated Pareto sets,” IEEE Transactions on Evolutionary Computation, vol. 13, no. 2, pp. 284–302, Apr. 2009.

[34] L. Hui, Q. Zhang. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. IEEE Transactions on Evolutionary Computation, 2009, 13(2):284-302.

[35] C. K. Goh, K. C. Tan. A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization. IEEE Transactions on Evolutionary Computation, 2009, 13(1): 103 – 127.

[36] S. Jiang and S. Yang. Evolutionary dynamic multi-objective optimization: Benchmarks and algorithm comparisons. IEEE Transactions on Cybernetics, 2017, 47(1): 198-211.

[37] L. Cao, L. Xu, E. D. Goodman, C. Bao, S. Zhu. Evolutionary Dynamic Multiobjective Optimization Assisted by a Support Vector Regression Predictor. IEEE Transactions on Evolutionary Computation, 2020, 24(2): 305-319.

[38] M. Jiang, Z. Wang, H. Hong, K. C. Tan. Knee Point Based Imbalanced Transfer Learning for Dynamic Multi-objective Optimization. IEEE Transactions on Evolutionary Computation, 2021, 25(1): 117-129.

[39] M. Jiang, Z. Wang, S. Guo, X. Gao, K. C. Tan. Individual-Based Transfer Learning for Dynamic Multiobjective Optimization. IEEE Transactions on Cybernetics (early access), 2021.

[40] M. C´amara, J. Ortega, F. de Toro. Performance measures for dynamic multiobjective optimization. International Work-conference on Artificial Neural Networks, 2009: 760–767.

[41] Zhang, Q. Yang, S. Jiang, S. Wang, R. & Li, X. (2019). Novel prediction strategies for dynamic multi-objective optimization. IEEE Transactions on Evolutionary Computation, PP. (99), 1-1.

[42] R. C. Liu, Y. Y. Chen, W. P. Ma, C. H. Mu and L. C. Jiao, "A competitive cooperative dynamic multiobjective optimization algorithm using a new predictive model," Soft. Comput., vol. 18, pp. 1913–1929, 2014.

[43] K. Deb, R. B. Agrawal. Simulated binary crossover for continuous search space, Complex Systems, 1995, 9(4): 115–148.

[44] J. Carrasco, S. García, M.M. Rueda, S. Das, F. Herrera, Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review, Swarm and Evolutionary Computation, 54:100665, 2020.

[45] E. Zitzler. Evolutionary algorithms for multiobjective optimization: Methods and applications. Ph.d. thesis Swiss Federal Institute of Technology, 1999.