Soft Comput (2015) 19:2633–2653
DOI 10.1007/s00500-014-1433-3

METHODOLOGIES AND APPLICATION

Novel prediction and memory strategies for dynamic multiobjective optimization

Zhou Peng · Jinhua Zheng · Juan Zou · Min Liu

Published online: 28 August 2014

Abstract Dynamic multiobjective optimization problems (DMOPs) exist widely in real life, which requires the optimization algorithms to be able to track the Pareto optimal solution set after the change efficiently. In this paper, novel prediction and memory strategies (PMS) are proposed to solve DMOPs. Regarding prediction, the prediction strategy contains two parts, i.e., exploration and exploitation. Exploration can enhance the ability to search the entire solution space, making it adapt to the environmental change with a great extent. Exploitation can improve the accuracy of local search, making the algorithm to have a faster response to environmental change particularly in the solution set having relevance in the environment. In terms of memory, an optimal solution set preservation mechanism is employed, by reusing the previously found elite solutions, which improves the performance of the algorithm in solving periodic problems. Compared with two representative prediction strategies and a hybrid strategy combining prediction and memory both on seven traditional benchmark problems and on five newly appeared ones, PMS has been shown to have faster response to the environmental changes than the peer algorithms, performing well in terms of convergence and diversity.

Keywords Dynamic multiobjective optimization · Evolutionary algorithms · Prediction · Memory

Communicated by V. Loia.

Z. Peng · J. Zheng (✉) · J. Zou
Key Laboratory of Intelligent Computing and Information Processing, Ministry of Education, Information Engineering College of Xiangtan University, Xiangtan, Hunan Province, China
e-mail: jhzheng@xtu.edu.cn

M. Liu
Hunan University of Science and Technology, Xiangtan, Hunan Province, China

1 Introduction

Many real-world problems are dynamic multiobjective optimization problems (DMOPs), with not only the conflict among multiple objectives but also the objective function changing over time (Farina et al. 2004). Dealing with a complex dynamic system, static optimization method has obvious limitations. On these issues, the research objective is changing intricately. Traditional evolutionary algorithm's goal is to make the population gradually converge to ultimately get a satisfactory solution set, but this would make the population lose diversity; especially in the later stages of the evolution, population will gradually lose ability to adapt to environmental change, which is the challenge of traditional evolutionary algorithm in a dynamic environment (Branke 2002; Jin and Branke 2005; Goh and Tan 2009; Nguyen et al. 2012). To track the Pareto optimal solution in a timely manner after change, researchers need to make adjustments to traditional static multiobjective algorithm (Coello Coello 2006, 2007; Yang et al. 2013; Li et al. 2013), so that it can quickly respond to environmental changes.

In recent years, researchers have proposed some dynamic multiobjective optimization algorithms to adapt DMOPs in artificial immune algorithm, co-evolutionary algorithm, particle swarm optimization algorithm, and memetic computing, which are based on the natural computation methods (Avdagi'c et al. 2009; Shang et al. 2013; Greeff and Engelbrecht 2008; Huang et al. 2011; Isaacs et al. 2008; Liu et al. 2011; Wei and Wang 2012; Azevedo and Araujo 2011; Manriquez et al. 2010; Cámara et al. 2010; Zheng 2007). These methods are mostly concentrated on maintaining population diversity and adopting the strategy of multi-population. However, these methods are slightly blind to solve DMOPs, because the population still relies on the ability to evolve independently to find the optimal solution after the diversity of the population is maintained, and the rate of convergence is the main problem. As to the way to post for objectives after change which provides a guiding direction for population evolution, it will play a vital role to improve the rate of convergence and convergence of the algorithm.

Therefore, some scholars have proposed the strategies of introducing memory (Deb et al. 2007; Zhang 2008; Wang and Li 2009; Isaacs et al. 2008; Guan et al. 2005; Goh and Tan 2009 and prediction (Hatzakis and Wallace 2006a, b; Zhou et al. 2007, 2013; Ma et al. 2011 to solve this problem. Memory strategy reuses the optimal solutions which are previously searched by the memory to rapidly response to changes in the new environment. This strategy can achieve good results for periodic problems, but for non-periodic problems or in the first period of changing environment, population is still in the process of blind evolution, and algorithm is difficult to obtain good convergence. For example, Deb et al. (2007) proposed a dynamic NSGA-II algorithm, Zhang (2008) proposed a dynamic multiobjective immune algorithm, and Goh and Tan (2009) proposed a dynamic competition–cooperation coevolutionary algorithm. These algorithms are simply introducing memory strategy to solve DMOPs, but the effect is not very good in the first period and changes in a particular environment. After each environmental change, methods based on prediction provide guidance direction for the population evolution by the prediction mechanism, and help algorithm to respond quickly to new changes. And forecast accuracy is the main difficulty. So far prediction method has few in-depth studies in DMOPs. Hatzakis and Wallace (2006a) proposed a feed-forward prediction strategy, referred to as FPS, in 2013, Zhou and Jin et al. (2013) proposed a population prediction strategy, referred to as PPS. Prediction strategies are used in these algorithms to solve the DMOPs, and have achieved good results, but there are some defects. For example, prediction is not accurate enough in FPS by simply using autoregressive model. And PPS has poor convergence due to the lack of historical information accumulation in the beginning stage of environmental change.

The combination of memory and prediction is a new trend to solve DMOPs (Yang et al. 2012). Prediction strategy can guarantee the algorithm to quickly respond to environmental changes, and in a timely manner to search for a new optimal solution set. Meanwhile, the memory strategy can improve the ability of dealing periodic problems and has reduced the error rate because of every prediction that may generate. In recent years, a few researches have used this hybrid strategy to solve DMOPs (Wang and Li 2009; Helbig and Engelbrecht 2011; Koo et al. 2009), for example, in 2010, Wang and Li proposed a multi-strategy ensemble evolutionary algorithm, referred to as dMS-MOEA (Wang and Li 2009), in the same year, Koo et al. (2009) proposed a predictive gradient strategy, referred to as dMO-EGS. Both these two algorithms use hybrid strategies which combine memory and prediction to solve DMOPs, but they have their own defects. In the aspect of prediction, dMS-MOEA uses an adaptive genetic and difference operator. In the early stage of running the algorithm, genetic operator can guarantee rapid convergence. And in the later stages of running, the difference operator can ensure a better distribution for the solution set having a better convergence. However, this prediction does not build on the basis of available information, but only increases the diversity of population, so this is a blind prediction. dMO-EGS predicts the evolutionary gradient of current population by recording the final two previous positions of nondominated population center which have been searched and the last predictive evolutionary gradients. This gradient prediction applies only to the same or a similar degree of change in the environment; when there is a large degree of the environmental change, the accuracy of prediction is difficult to be guaranteed. In the aspect of memory, dMS-MOEA produces new individuals on the basis of the searched final nondominated populations by the difference operator and Gaussian mutation operation, and they are stored to the archive. The archive uses Gaussian local search operator to get a new population. dMO-EGS stores past population centers and variance vectors of nondominated solution set, and then generate new populations according to the centers and the variance vectors by normal distribution sampling. Since the sampling process prone to distortion, so the two strategies are incomplete memory strategies.

To solve these problems, the paper proposes novel prediction and memory strategies, referred to as PMS. The strategy consists of three parts: (1) exploration operator based on population evolutionary direction; (2) exploitation operator based on the direction of nondominated solutions linkage; (3) memory strategy based on the optimal solution set.

The rest of the paper is organized as follows. Section 2 presents background information. Section 3 describes the novel PMS in detail. Section 4 introduces the test problems and evaluation metric. Section 5 gives experimental results and analysis. Section 6 outlines the conclusions and future work.

2 Background

2.1 Dynamic multiobjective optimization

A minimization problem is considered here without loss of generality. The DMOP can be described as (Farina et al. 2004):

$$\begin{cases} \min\limits_{x\in\Omega} F(x,t) = (f_1(x,t), f_2(x,t), \ldots, f_m(x,t))^{\mathrm{T}} \\ s.t.\;\; g_i(x,t) \le 0 \; i = 1, 2, \ldots, p;\;\; h_j(x,t) = 0 \\ \; j = 1, 2, \ldots, q \end{cases}$$

where t is the time variable, $x = (x_1, x_2, \dots, x_n)$ is the n-dimensional decision variables bounded by the decision space Ω, $F = (f_1, f_2, \dots, f_m)$ presents the set of m objectives to be minimized the functions of $g_i \leq 0\ i = 1, 2, \dots, p$ and $h_j = 0\ j = 1, 2, \dots, q$ present the set of inequality and equality constraints.

Definition 1 *Pareto Dominance*: p and q are any two individuals in the population, p is said to dominate q, denoted by $f(p) \prec f(q)$ iff $f_i(p) \leq f_i(q) \forall i = \{1, 2, \dots, m\}$ and $f_j(p) < f_j(q) \exists j \in \{1, 2, \dots, m\}$.

Definition 2 *Pareto Optimal Set (PS).* x is the decision variable, Ω is the decision space, F is the objective function, thus the PS (Coello Coello 2007) is the set of all nondominated solutions and is defined mathematically as:

$$\mathrm{PS} := \left\{x \in \Omega\ \nexists | x^* \in \Omega, F(x^*) \prec F(x)\right\}$$

Definition 3 *Pareto Optimal Front (PF).* x is the decision variable, F is the objective function, thus the PF (Coello Coello 2007) is the set of nondominated solutions with respect to the objective space and is defined mathematically as:

$$\mathrm{PF} := \{y = F(x) | x \in \mathrm{PS}\}$$

To distinguish between different types of DMOPs, Farina et al. (2004) identified four different types of DMOPs:

- *Type I* PS changes with time, but PF remains fixed.
- *Type II* Both PF and PS change with time.
- *Type III* PF changes with time, but PS remains fixed.
- *Type IV* Both PF and PS remain fixed.

In real life, the above four types of changes may occur at the same time when the problem is changed. However, we mainly research the first three types of changes.

2.2 General steps of dynamic multiobjective evolutionary algorithm

Dynamic multiobjective evolutionary algorithm is generally divided into the following steps:

Step 1. Initialize a population, set the related parameters;

Step 2. Optimization algorithm: this paper chooses RM-MEDA which will be introduced in Sect. 2.2.1 as the MOEA optimizer.

Step 3. Change detection: if the environmental change has been detected, the mechanism of change reaction will be called to respond to the changes; otherwise, continue to perform optimization algorithm. A good change detection operator can exclude external factors related to disturbance, and determine environmental changes correctly.

Step 4. The mechanism of change reaction: the mechanism can respond quickly and correctly to environmental change, so that the population can quickly converge to the new optimal solution set. General mechanism has the following kinds: re-initialize population, parameter adjustment, dynamic migration, memory and prediction.

Step 5. Judgment of terminal condition.

Figure 1 illustrates the flowchart of dynamic multiobjective evolutionary algorithm:

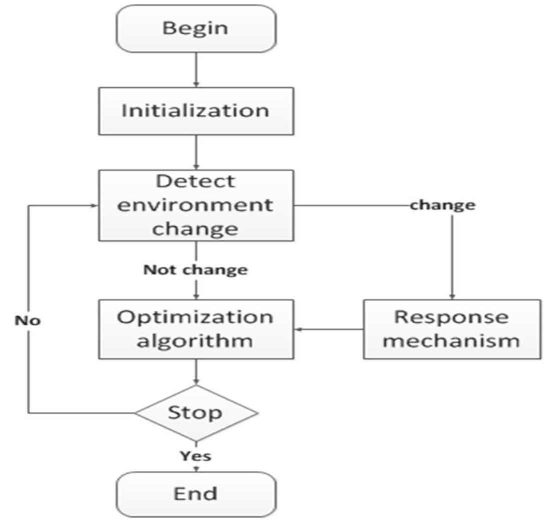


Fig. 1 Flowchart of DMOEA

2.2.1 RM-MEDA

Estimation of distribution algorithm, referred to as EDA Larrañaga and Lozano (2001), is a class of random optimization algorithm which newly emerged on the field of evolutionary computation. Different from genetic algorithm, EDA is built on the basis of the estimation of distribution of solutions. EDA describes the distribution of solutions in the solution space by establishing a probability model, and then generates new population by random sampling for the probability model.

Under certain smoothness assumptions, it can be induced from the Karush–Kuhn–Tucker condition that the PS of a continuous MOP defines a piecewise continuous $(m-1)$-dimensional manifold in the decision space. Therefore, on the basis of EDA, combined with the distribution features of solution set of continuous multiobjective problem, Zhang et al. (2008) proposed a regularity model-based multiobjective estimation of distribution algorithm in 2008, referred to as RM-MEDA.

RM-MEDA can make full use of the global information provided by current population to generate the new individual, without the need for local information directly used from population, which has a strong ability of heuristic optimization. Since RM-MEDA utilized the regularity property of continuous MOPs, it has been verified in Zhou et al. (2013)

to be an excellent optimization algorithm for DMOPs, and performs better than NSGA-II-DE in the DMOEA framework.

2.3 Performance analysis for three kinds of DMOEA

The method based on prediction is one of the effective ways to solve DMOPs, which is to generate the next new population under environmental change by prediction. It can accelerate the convergence of algorithm to the new optimal solution. Meanwhile, the way combined with memory which reuses the past information under the same or similar environment is another new way to solve DMOPs. This paper mainly discusses the comparison of two representative prediction strategies and a hybrid strategy combined prediction and memory and, on this basis, we propose a new PMS to solve DMOPs.

(1) Feed-forward prediction strategy (FPS)
FPS (Hatzakis and Wallace 2006a) records information of boundary point on the adjacent PF of objective space, and predicts the location of PS after environmental change by autoregressive model (AR). The initial population in FPS is composed of three parts: the nondominated solution set, the dominated solution set and the solution set by prediction. Nondominated solutions make the algorithm more conducive to solve the DMOPs whose PS does not change with time, and dominated solutions maintain the diversity of population, using the algorithm to search for the new optimal solution.

(2) Population prediction strategy (PPS)
In PPS Zhou et al. (2013), the optimal solution set is divided into two parts: the population center and shape. PPS uses autoregressive model (AR) to predict a new population center at the next time by storing population center on a continuous time series. Meanwhile, it predicts a new population distribution of the next time by recording the shape of population in the last two moments. These two parts make up the initial population after change.
Comparing these two strategies, FPS produces only a small part of the new population by prediction, the majority of its population is inherited from the final population before environmental change; PPS generates a whole new population by prediction, rather than just a few initial individuals. With the algorithm running, the accumulated history information increases in PPS, and the accuracy of prediction will be improved. Therefore, at the early stages, PPS is slightly worse than the FPS in performance due to insufficient historical information. At the latter stages, the accumulation of experience make PPS performance becomes stable, which is better than FPS (Zhou et al. 2013).

(3) Predictive gradient strategy (dMO-EGS)
dMO-EGS (Koo et al. 2009) consists of three parts: mutation, prediction and memory.
To increase the diversity of population, firstly, dMO-EGS performs mutation operation for part of the nondominated solutions in current population. Mutation uses the sampling of normal distribution with expectation of zero and standard deviation of 0.01. In the aspect of prediction, dMO-EGS predicts the evolutionary gradient of current population by recording the final two previous positions of nondominated population center which have been searched and the last predictive evolutionary gradients, and updates some individuals of the population with the predicted gradient and the product of a scaling factor. Meanwhile, we select an individual randomly from the original population and compare with a new individual. If the new individual is dominated, the scaling factor will be negated. However, the predictive gradient is reversed permanently at the end of the updating only if the scaling factor is negative for at least half of the time to allow the possibility of random fluctuations. In the aspect of memory, dMO-EGS designs a memory method based on center and variance. Firstly, calculate the center vector $C\tau$ and variance vector $\widetilde{C}\tau$ of the nondominated solution set after environmental change, then $C\tau$ and $\widetilde{C}\tau$ will be combined into a memory item to be stored. For the each item of the memory, generate new solutions according to the normal distribution with the expectation of $C\tau$ and variance of $\widetilde{C}\tau$.

The three operators in dMO-EGS are effective in increasing the diversity of population, which can better adapt to the DMOPs whose degree of change is smaller. However, this gradient prediction of simple correlation based on the solution set applies only to the same or similar degree of change in the environment; when there is a large degree of the environmental change, the accuracy of prediction is difficult to be guaranteed. Moreover, the last searched information is not intact in its memory. This will be prone to distortion in the course of normal sampling, and there is greater influence on the convergence of algorithm.

3 Novel PMS for dynamic multiobjective optimization

3.1 Prediction strategy

To solve the problem which cannot respond quickly to changes in DMOPs, in a timely manner to search for the optimal solution set of new problem, this paper proposes a prediction strategy of both exploration and exploitation. Exploration uses exploration operator based on population evolutionary direction, and generates a bunch of guide individuals in population evolutionary direction to ensure the algorithm's ability to search for the optimal solution space. Exploitation uses exploitation operator based on the direction of nondominated solutions linkage to improve the searching accuracy of algorithm under similar environmental change.

3.1.1 Exploration operator

Whenever the environmental changes, the population will spontaneously evolve towards the direction of the new Pareto optimal solutions. So taking advantage of this characteristic, let the population evolve independently for a short time in the initial stage of environmental change, then we judge the evolutionary direction of the population in the current environment by recording the different center positions of current population before and after the change.

Figure 2 illustrates the principle of exploration operator, for the final searched population (Population_{t-1}) in the last time, and the new population ($\text{Population}_{t-1} + \Delta t$) after evolving spontaneously for time Δt, it is the connecting direction between these two different population center positions at different time that represents the predicted population evolutionary direction. On this basis, firstly, evenly select several initial individuals from current population, and then a certain number of individuals are evenly distributed from the initial individuals to the boundary of the decision space according to the predicted direction, which corresponds to the guide individual in Fig. 2. After nondominated sorting these individuals, the relatively better individuals are selected, namely, the individuals which are closest to PS_t and connected by dotted lines.

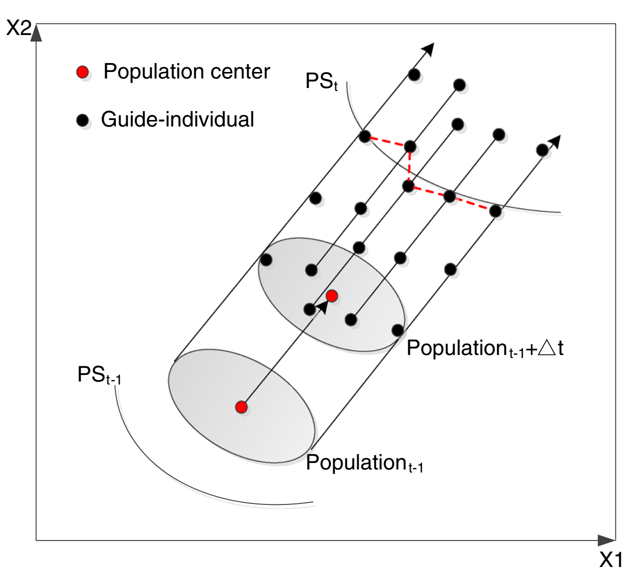


Fig. 2 Principle of exploration operator

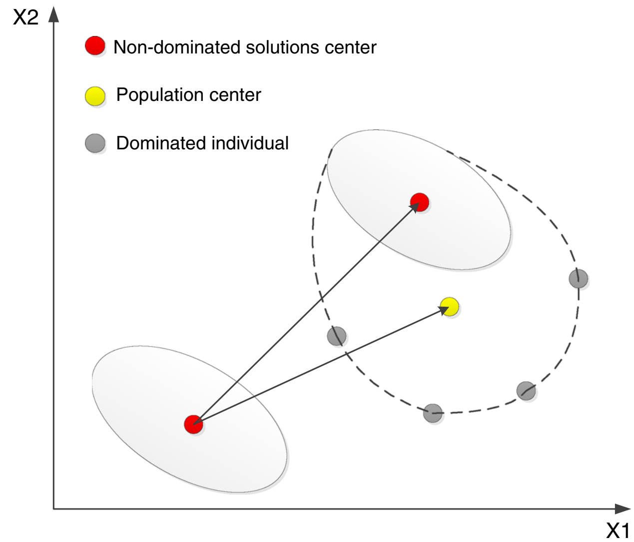


Fig. 3 The impact of judging the evolutionary direction when population is not fully convergent

Determining operator for the predicted population evolutionary direction is defined as:

$$\partial = \{\partial_1, \partial_2, \ldots, \partial_i, \ldots, \partial_n\} = \left\{C_1^{t+\Delta t} - C_1^t, C_2^{t+\Delta t} - C_2^t, \ldots, C_i^{t+\Delta t} - C_i^t, \ldots, C_n^{t+\Delta t} - C_n^t\right\}, \quad (1)$$

where ∂ represents evolutionary direction vector of population in decision space, $i = 1, 2, \ldots, n$. C^t is the population center at time t, $C^{t+\Delta t}$ is the population center after population evolve spontaneously for time Δt.

As the center of nondominated solutions represents the position of population center best, so this can avoid the situation illustrated in Fig. 3. When the population is not completely converged, the number of nondominated solutions is smaller than the size of population, so there may be a small number of poor individuals affecting the center of the whole population. Therefore, to define the position of population center in formula (1) more accurately, the center of nondominated solutions will be indicated by the position of population center. Let $P_{\text{non-do}}^t = \{x^t\}$ represents nondominated solutions at time t, and then the position of population center is defined as:

$$C_i^t = \frac{1}{\left|P_{\text{non-do}}^t\right|} \sum_{x_i^t \in P_{\text{non-do}}^t} x_i^t \quad (2)$$

where $\left|P_{\text{non-do}}^t\right|$ is the size of nondominated solutions, x_i^t is the individual at time t, $i = 1, 2, \ldots, n$. Determine the population evolutionary direction by calculating the position of population center each dimension in the decision space before and after the change. The results are denoted by a set of vectors that forward is positive and backward is negative.

The judgment about population evolutionary direction provides a basis for predicting the location of Pareto optimal solutions more accurately, for which the exploration operator is needed to obtain a bunch of guide individuals on the direction. Firstly, the method of calculating the crowding distance (Deb et al. 2002) of current population will be used in the decision space, so that it can get some initial individuals more evenly. From the initial individuals to the boundary of the decision space, a bunch of detecting individuals are distributed equal distance intervals according to the direction predicted in formula (1). Finally, we get ObSize guide individuals by a nondominated sorting. Exploration algorithm based on population evolutionary direction is described in Algorithm 1.

3.1.2 Exploitation operator

Changes in the environment present regularity in many cases; the current environment may have the same or similar degree of change with the previous environment. Therefore, taking advantage of this feature is conducive to improve the searching accuracy of algorithm under similar environmental changes. As shown in Fig. 4, the connection vectors between centers of the final searched populations under each environment change represent the degree of change in the environment, and the connection vector on the adjacent time series may have similar direction and distance. In the current environment, based on the connection vector of a recent time

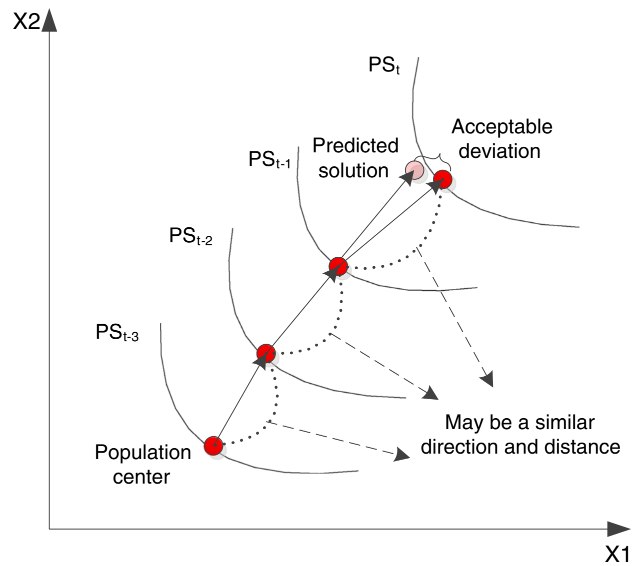


Fig. 4 Principle of exploitation operator

Algorithm 1 Exploration algorithm ($P_{duplicate}$)

Input: $P_{duplicate}$, duplicate of population for exploration.

Output: $P_{exploration}$, set of exploration population obtained.

Step 1 Calculate the crowding-distance of each individual of $P_{duplicate}$ in the decision space, select OP individuals from $P_{duplicate}$ whose crowding-distance is the largest, denoted as $Ob^j, j = 1, 2, ..., OP$.

Step 2 According to the direction vector obtained in formula (1), if it is positive, then the decision space from the maximum boundary of the i-th dimension to the initial individuals is equally divided into OB parts; if it is negative, then the decision space from the minimum boundary of the i-th dimension to the initial individuals is equally divided into OB parts. The OB*(OP-1) detecting individuals obtained are represented as:

$$Ob_i^{k*OP+j} = \begin{cases} Ob_i^j + \dfrac{Up_i - Ob_i^j}{OB} * k, & if\ \partial_i > 0 \\ Ob_i^j - \dfrac{Ob_i^j - Low_i}{OB} * k, & if\ \partial_i < 0 \end{cases} \tag{3}$$

$$j = 1, 2, \ldots, OP \quad k = 1, 2, \ldots, OB - 1$$

Where Up_i is the maximum boundary of the i-th dimension, Low_i is the minimum boundary of the i-th dimension.

Step 3 Nondominated sort these OB*OP individuals, select ObSize optimal detecting individuals as the guide-individuals in current environment, and store them in $P_{exploration}$.

Step 4 Return $P_{exploration}$.

Algorithm 2 Exploitation algorithm ($P_{duplicate}$)

Input: $P_{duplicate}$, duplicate of population for exploitation; t, number of environmental change.

Output: $P_{exploitation}$, set of exploitation population obtained.

Step 1 Calculate the crowding-distance of each individual of $P_{duplicate}$ in the decision space, and select OP individuals from $P_{duplicate}$ whose crowding-distance is the largest, denoted as $Ob^j, j = 1, 2, ..., OP$.

Step 2 If t<3, calculate C according to the formula (2), then store C in Archive C, and return to step 1; else, turn to step 3.

Step 3 Update Ob^j to $P_{exploitation}$ according to the formula (4), C(t-1) and C(t) corresponding to the first and second terms of Archive C.

Step 4 Calculate C at the current moment, then update Archive C according to the principle of FIFO (first in first out), namely store C and remove the first item.

Step 5 Return $P_{exploitation}$.

series, some predicted solutions can be produced; there will be some errors between these solutions and the true Pareto optimal solutions, but these errors are within the acceptable range.

This paper adopts an exploitation operator based on the direction of nondominated solutions linkage. Exploitation operator is defined as:

$$x(t+1) = x(t) + C(t) - C(t-1), \qquad (3)$$

where $C(t)$ is the center of nondominated solutions obtained according to the formula (2) at time t, $C(t-1)$ is the center of nondominated solutions at time $t-1$. We predict the new optimal solutions, according to the center of nondominated solutions before and after the change of two recent different environments. Exploitation algorithm based on the direction of nondominated solutions linkage is described in Algorithm 2.

3.2 Memory strategy

For periodic DMOPs, the optimal solution under the new environment may return to the past-searched position. Therefore, making full use of the past-searched optimal solutions can speed up the convergence rate which is equivalent to iterate on the original searching process. This paper proposes a memory strategy based on optimal solution set preservation mechanism. When the environment changes, we nondominating sort these stored individuals in memory pool, and select optimal individuals which adapt best to the new environment. After that, we store the nondominated individuals of current population to memory pool.

As shown in Fig. 5, the individuals stored in memory pool are optimal with best convergence and diversity in past environment. These individuals can cover a period of environmental changes and apply to different environmental changes which have been detected. When a new change is detected, the optimal individuals in the last same environment will be

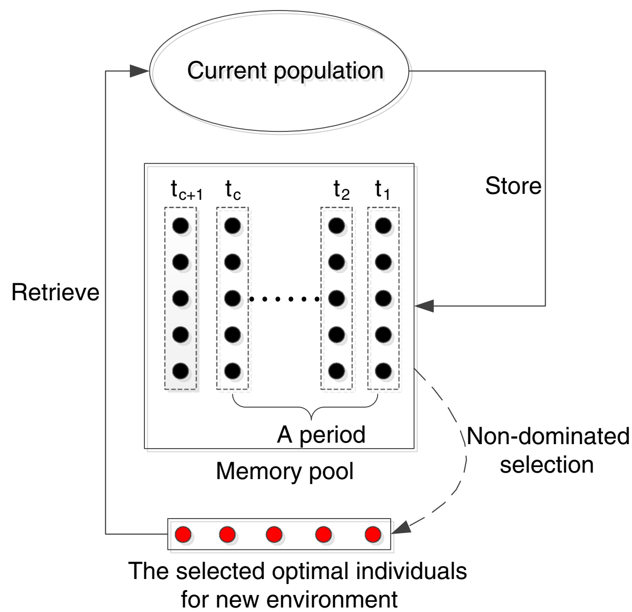


Fig. 5 Principle of memory strategy

selected by nondominated selection, and continue to evolve in new environment after retrieval. So the effect of distortion can be avoided due to normal distribution sampling by reusing directly the optimal solutions which are previously searched. This method improves searching accuracy and convergence speed. When the size of memory pool exceeds the value of set, we use the principle of FIFO to update memory pool, which ensures that the algorithm does not consume too much extra storage space and evaluation. Memory algorithm based on optimal solution set is described in Algorithm 3.

3.3 Detailed description of PMS

PMS iterates under the general framework of DMOEA, the purpose is to obtain new initial population after each envi-

Algorithm 3 Memory algorithm ($P_{duplicate}$, Memory_pool)

Input: $P_{duplicate}$, duplicate of population for memory; Memory_pool, memory pool.

Output: P_{memory}, set of memory population obtained.

Step 1 Copy the Memory_pool, denoted as M_copy.

Step 2 If M_copy is empty, turn to Step 3; else, evaluate individuals of M_copy, select Msize optimal individuals by nondominating sorting, and stored them in P_{memory}.

Step 3 Select Msize optimal individuals from $P_{duplicate}$, stored in Memory_pool.

Step 4 If the size of Memory_pool is over twice of the population size, remove the first stored Msize individuals of Memory.

Step 5 Return P_{memory}.

Algorithm 4 PMS

Input: Pop, current population; gmax, total number of generation; τ_t, frequency of change. n_t, severity of changes; DMOPs.

Output: P_t, updated population.

Initialization set t := 0; initialize a population P_0; set iteration counter gt := 0;

Step 1 Detect changes in the environment, if environment has not changed, turn to Step 7; else copy current population Pop and denoted it as P_copy, calculate the center C_i^t of each dimension in current population according to the formula (2).

Step 2 Pop evolves independently for a short time Δt, calculate the center $C_i^{t+\Delta t}$ of each dimension after evolution, set gt := gt+ Δt;

Step 3 For Pop, obtain exploration population $P_{exploration}$ according to Algorithm 1(Pop).

Step 4 For P_copy, obtain exploitation population $P_{exploitation}$ according to Algorithm 2(P_copy).

Step 5 For P_copy, obtain memory population P_{memory} according to Algorithm 3(P_copy, Memory_pool).

Step 6 Set current archive $P_{archive} = P_{exploration} \cup P_{exploitation} \cup P_{memory}$, and perform boundary detection for

$P_{archive}$: if $P_{archive}(x_i) > Up_i$, $P_{archive}(x_i) = Up_i - random(0, \Delta r)$;

if $P_{archive}(x_i) < Low_i$, $P_{archive}(x_i) = Low_i + random(0, \Delta r)$.

Where Δr is a small radius. Use the post-test $P_{archive}$ to replace the worst individuals of Pop.

Step 7 Optimize population with RM-MEDA.

Step 8 If gt > gmax, output P_t and stop; else, set gt := gt+1, return to step 1.

ronmental change, so that the new population can quickly respond to changes and track to the new optimal solutions. The PMS is described in detail in Algorithm 4.

4 Test instances and performance metrics

4.1 Test instances

This section presents the DMOPs test problems used in the experiment. FDA test suite has been proposed in Farina et al. (2004) and is often used to examine the performance of DMOEA. DMOP test suite has been proposed in Goh and Tan (2009) and is an extension to the FDA. The two test suites are linear correlation between the decision variables. F5–F9 proposed in Zhou et al. (2013) are the new DMOPS test problems. The characteristic of F5–F9 is nonlinear correlation between the decision variables. Especially for F9, the environment changes smoothly in most cases and, occasionally, the Pareto set jumps from one area to another area, and the difficulty of convergence is larger.

(1) FDA and DMOP test suite

FDA test suite is proposed by Farina and Deb et al. for assessing the performance of DMOEA. Four test problems

Table 1 Test problems of FDA and DMOP

Problems	Search space	Objectives, PF, PS	Remarks
FDA1	$[0,1]\times[-1,1]^{n-1}$	$f_1(x)=x_1,\ f_2(x)=g\cdot h$ $g(x)=1+\sum_{i=2}^{n}(x_i-G(t))^2,\ h(x)=1-\sqrt{f_1/g}$ $G(t)=\sin(0.5\pi t),\ t=\lfloor\tau/\tau_T\rfloor/n_T$ $\mathrm{PS}(t): 0\le x_1\le 1,\ x_i=G(t),\ i=2,\dots,n$ $\mathrm{PF}(t): 0\le f_1\le 1,\ f_2=1-\sqrt{f_1}$	Two objectives PF is fixed PS changes
FDA2	$[0,1]\times[-1,1]^{n-1}$	$f_1(x)=x_1,\ f_2(x)=g\cdot h$ $g(x)=1+\sum_{i=2}^{n/2}x_i^2,\ h(x)=1-\left(\frac{f_1}{g}\right)^{(H(t)+\sum_{i=n/2+1}^{n}(x_i-H(t))^2)^{-1}}$ $H(t)=0.75+0.7\sin(0.5\pi t),\ t=\lfloor\tau/\tau_T\rfloor/n_T$ $\mathrm{PS}(t): 0\le x_1\le 1,\ x_i=0,\ i=2,\dots,n/2,\ x_j=-1,\ j=n/2+1,\dots,n$ $\mathrm{PF}(t): 0\le f_1\le 1,\ f_2=1-f_1^{H(t)^{-1}}$	Two objectives PF changes PS is fixed
FDA3	$[0,1]\times[-1,1]^{n-1}$	$f_1(x)=x_1^{F(t)},\ f_2(x)=g\cdot h$ $g(x)=1+G(t)+\sum_{i=2}^{n}(x_i-G(t))^2,\ h(x)=1-\sqrt{f_1/g}$ $G(t)=\lvert\sin(0.5\pi t)\rvert,\ F(t)=10^{2\sin(0.5\pi t)},\ t=\lfloor\tau/\tau_T\rfloor/n_T$ $\mathrm{PS}(t): 0\le x_1\le 1,\ x_i=G(t),\ i=2,\dots,n$ $\mathrm{PF}(t): 0\le f_1=x_1^{F(t)}\le 1,\ f_2=(1+G(t))(1-\sqrt{\frac{f_1}{1+G(t)}})$	Two objectives PF changes PS changes
FDA4	$[0,1]^n$	$f_1(x)=(1+g)\cos(0.5\pi x_2)\cos(0.5\pi x_1)$ $f_2(x)=(1+g)\cos(0.5\pi x_2)\sin(0.5\pi x_1)$ $f_3(x)=(1+g)\sin(0.5\pi x_2)$ $g(x)=\left\lvert\sum_{i=3}^{n}(x_i-G)^2\right\rvert$ $G(t)=\sin(0.5\pi t),\ t=\lfloor\tau/\tau_T\rfloor/n_T$ $\mathrm{PS}(t): 0\le x_1,x_2\le 1,\ x_i=G(t),\ i=3,\dots,n$ $\mathrm{PF}(t): f_1=\cos(u)\cos(v),\ f_2=\cos(u)\sin(v)$ $f_3=\sin(u),\ 0\le u,v\le\pi/2$	Three objectives PF is fixed PS changes
DMOP1	$[0,1]\times[-1,1]^{n-1}$	$f_1(x)=x_1,\ f_2(x)=g\cdot h$ $g(x)=1+9\sum_{i=2}^{n}x_i^2,\ h(x)=1-\left(\frac{f_1}{g}\right)^{H(t)}$ $H(t)=1.25+0.75\sin(0.5\pi t),\ t=\lfloor\tau/\tau_T\rfloor/n_T$ $\mathrm{PS}(t): 0\le x_1\le 1,\ x_i=0,\ i=2,\dots,n$ $\mathrm{PF}(t): 0\le f_1\le 1,\ f_2=1-f_1^{H(t)}$	Two objectives PF changes PS is fixed
DMOP2	$[0,1]\times[-1,1]^{n-1}$	$f_1(x)=x_1,\ f_2(x)=g\cdot h$ $g(x)=1+\sum_{i=2}^{n}(x_i-G(t))^2,\ h(x)=1-\left(\frac{f_1}{g}\right)^{H(t)}$ $G(t)=\sin(0.5\pi t),\ H(t)=1.25+0.75\sin(0.5\pi t),\ t=\lfloor\tau/\tau_T\rfloor/n_T$ $\mathrm{PS}(t): 0\le x_1\le 1,\ x_i=G(t),\ i=2,\dots,n$ $\mathrm{PF}(t): 0\le f_1\le 1,\ f_2=1-f_1^{H(t)}$	Two objectives PF changes PS changes
DMOP3	$[0,1]\times[-1,1]^{n-1}$	$f_1(x_r)=x_r,\ f_2(x\backslash x_r)=g\cdot h$ $g(x)=1+\sum_{i=1}^{x\backslash x_r}(x_i-G(t))^2,\ h(x)=1-\sqrt{\frac{f_1}{g}}$ $G(t)=\sin(0.5\pi t),\ r=\cup(1,2,\dots,n),\ t=\lfloor\tau/\tau_T\rfloor/n_T$ $\mathrm{PS}(t): 0\le x_1\le 1,\ x_i=G(t),\ i=2,\dots,n$ $\mathrm{PF}(t): 0\le f_1\le 1,\ f_2=1-\sqrt{f_1}$	Two objectives PF is fixed PS changes

FDA1–FDA4 will be used in this paper. DMOP test suite has been proposed by Goh and Tan et al. and is an extension of the FDA. Table 1 lists all the seven test problems and their PF and PS in detail. FDA4 is a three-objective problem among them.

(2) F5–F9 test suite

F5–F9 are newly proposed by Zhou and Jin et al., which have nonlinear correlation between decision variables. Among them, F5-F7 are two-objective problems, F8 is a three-objective problem, and F9 is a complicated problem which is more difficult to converge than other test problems. Table 2 lists all the five test problems and their PF and PS in detail.

Table 2 Test problems of F5–F9

Problems	Search space	Objectives, PF, PS	Remarks
F5	$[0,5]^n$	$f_1(x)=\lvert x_1-a\rvert^{H(t)}+\sum_{i\in I_1} y_i^2$ $f_2(x)=\lvert x_1-a-1\rvert^{H(t)}+\sum_{i\in I_2} y_i^2$ $y_i=x_i-b-1+\lvert x_1-a\rvert^{H(t)+\frac{i}{n}}$, $H(t)=1.25+0.75\sin(\pi t)$ $a=2\cos(\pi t)+2$, $b=2\sin(2\pi t)+2$, $t=\lfloor\tau/\tau_T\rfloor/n_T$ $I_1=\{i\lvert 1\le i\le n, i \text{ odd}\}$, $I_2=\{i\lvert 1\le i\le n, i \text{ is even}\}$ $\text{PS}(t): a\le x_1\le a+1, x_i=b+1-\lvert x_1-a\rvert^{H(t)+\frac{i}{n}}, i=2,\ldots,n$ $\text{PF}(t): f_1=s^{H(t)}, f_2=(1-s)^{H(t)}, 0\le s\le 1$	Two objectives PF changes PS changes
F6	$[0,5]^n$	$f_1(x)=\lvert x_1-a\rvert^{H(t)}+\sum_{i\in I_1} y_i^2$ $f_2(x)=\lvert x_1-a-1\rvert^{H(t)}+\sum_{i\in I_2} y_i^2$ $y_i=x_i-b-1+\lvert x_1-a\rvert^{H(t)+\frac{i}{n}}$, $H(t)=1.25+0.75\sin(\pi t)$ $a=2\cos(1.5\pi t)\sin(0.5\pi t)+2$, $b=2\cos(1.5\pi t)\cos(0.5\pi t)+2$ $t=\lfloor\tau/\tau_T\rfloor/n_T$ $I_1=\{i\lvert 1\le i\le n, i \text{ is odd}, \}$ $I_2=\{i\lvert 1\le i\le n, i \text{ is even}\}$ $\text{PS}(t): a\le x_1\le a+1, x_i=b+1-\lvert x_1-a\rvert^{H(t)+\frac{i}{n}}, i=2,\ldots,n$ $\text{PF}(t): f_1=s^{H(t)}, f_2=(1-s)^{H(t)}, 0\le s\le 1$	Two objectives PF changes PS changes
F7	$[0,5]^n$	$f_1(x)=\lvert x_1-a\rvert^{H(t)}+\sum_{i\in I_1} y_i^2$ $f_2(x)=\lvert x_1-a-1\rvert^{H(t)}+\sum_{i\in I_2} y_i^2$ $y_i=x_i-b-1+\lvert x_1-a\rvert^{H(t)+\frac{i}{n}}$, $H(t)=1.25+0.75\sin(\pi t)$ $a=1.7(1-\sin(\pi t))\sin(\pi t)+3.4$, $b=1.4(1-\sin(\pi t))\cos(\pi t)+2.1$ $t=\lfloor\tau/\tau_T\rfloor/n_T$ $I_1=\{i\lvert 1\le i\le n, i \text{ is odd}\}$, $I_2=\{i\lvert 1\le i\le n, i \text{ is even}\}$ $\text{PS}(t): a\le x_1\le a+1, x_i=b+1-\lvert x_1-a\rvert^{H(t)+\frac{i}{n}}, i=2,\ldots,n$ $\text{PF}(t): f_1=s^{H(t)}, f_2=(1-s)^{H(t)}, 0\le s\le 1$	Two objectives PF changes PS changes
F8	$[0,1]^2\times[-1,2]^{n-2}$	$f_1(x)=(1+g)\cos(0.5\pi x_2)\cos(0.5\pi x_1)$ $f_2(x)=(1+g)\cos(0.5\pi x_2)\sin(0.5\pi x_1)$ $f_3(x)=(1+g)\sin(0.5\pi x_2)$ $g(x)=\sum_{i=3}^{n}(x_i-(\frac{x_1+x_2}{2})^{H(t)}-G(t))^2$ $G(t)=\sin(0.5\pi t)$, $H(t)=1.25+0.75\sin(\pi t)$ $t=\lfloor\tau/\tau_T\rfloor/n_T$ $\text{PS}(t): 0\le x_1,x_2\le 1, x_i=(\frac{x_1+x_2}{2})^{H(t)}+G(t), i=3,\ldots,n$ $\text{PF}(t): f_1=\cos(u)\cos(v), f_2=\cos(u)\sin(v), f_3=\sin(u)$ $0\le u,v\le\pi/2$	Three objectives PF changes PS changes
F9	$[0,5]^n$	$f_1(x)=\lvert x_1-a\rvert^{H(t)}+\sum_{i\in I_1} y_i^2$ $f_2(x)=\lvert x_1-a-1\rvert^{H(t)}+\sum_{i\in I_2} y_i^2$ $y_i=x_i-b-1+\lvert x_1-a\rvert^{H(t)+\frac{i}{n}}$, $H(t)=1.25+0.75\sin(\pi t)$ $a=2\cos((\frac{t}{n_T}-\lfloor\frac{t}{n_T}\rfloor)\pi)+2$, $b=2\sin(2(\frac{t}{n_T}-\lfloor\frac{t}{n_T}\rfloor)\pi)+2$ $t=\lfloor\tau/\tau_T\rfloor/n_T$ $I_1=\{i\lvert 1\le i\le n, i \text{ is odd}\}$, $I_2=\{i\lvert 1\le i\le n, i \text{ is even}\}$ $\text{PS}(t): a\le x_1\le a+1, x_i=b+1-\lvert x_1-a\rvert^{H(t)+\frac{i}{n}}, i=2,\ldots,n$ $\text{PF}(t): f_1=s^{H(t)}, f_2=(1-s)^{H(t)}, 0\le s\le 1$	Two objectives PF changes PS changes

4.2 Performance metrics

Some metrics have been designed for dynamic optimization (Cámara et al. 2009; Tantar et al. 2011; Helbig and Engelbrecht 2013). In this paper, we firstly introduce the dynamic generational distance (DGD) (Goh and Tan 2009) and inverted generational distance (DIGD) (Zhou et al. 2013) metric for DMOPs. The DGD and DIGD metrics are defined as follows:

$$\begin{aligned}
\text{DGD} &= \frac{1}{|T|}\sum_{t\in T}\text{GD}(\text{PF}^t, P^t),\\
\text{GD}(\text{PF}^t, P^t) &= \frac{\sum_{v\in P^t} d(\text{PF}^t, v)}{|P^t|}\\
\text{DIGD} &= \frac{1}{|T|}\sum_{t\in T}\text{IGD}(\text{PF}^t, P^t),\\
\text{IGD}(\text{PF}^t, P^t) &= \frac{\sum_{v\in \text{PF}^t} d(v, P^t)}{|\text{PF}^t|}
\end{aligned} \tag{4}$$

where PF^t is a set of uniformly distributed Pareto optimal points in the PF at time t, and P^t is the solution obtained at time t. $d(\mathrm{PF}^t, v) = \min_{u \in \mathrm{PF}^t} \sqrt{\sum_{j=1}^{m}\left(f_j^{(u)} - f_j^{(v)}\right)^2}$ is the distance between v and PF^t, $d(v, P^t) = \min_{u \in P^t} \sqrt{\sum_{j=1}^{m}\left(f_j^{(v)} - f_j^{(u)}\right)^2}$ is the distance between v and P^t, T is a set of discrete time points in a run and $|T|$ is the cardinality of T. DGD evaluates convergence of the algorithm, DGD value is lower, and the solution obtained set has better convergence. DIGD is a comprehensive metric to evaluate the convergence and distribution; lower DIGD value means that solution set obtained has a better convergence and distribution.

5 Experiments

5.1 Compared strategies and parameter settings

The novel strategy which proposed in this paper is denoted as PMS, and in this section, PMS will be compared to other three strategies: forward-looking prediction strategy (FPS) (Hatzakis and Wallace 2006a), PPS (Zhou et al. 2013), and predictive gradient strategy(dMO-EGS) (Koo et al. 2009) (Since only we compared the impact between different strategies for solving DMOPs, so in this section, mutation, PMS in dMO-EGS are denoted as MPMS, and integrated into an unified optimization algorithm), we choose RM-MEDA mentioned in Sect. 2.2.1 as the MOEA optimizer. In PMS, the number of guide individual in exploration operator is Obsize = 10, the number of initial individual is OP = 10, the number of aliquots is OB = 9, and the time of evolving independently is $\Delta t = 2$. The number of selected optimal individuals from memory pool is Msize = 5 (Three objectives: 10). Other parameter settings of three strategies use the given setting in Zhou et al. (2013) and Koo et al. (2009). The detailed settings of other parameters are shown in Table 3.

Since the proposed PMS in this paper and MPMS in dMO-EGS need to consume a certain number of evaluations in the prediction and memory, to be fair, the algorithm iterations require removing the number of evaluations consumed at every environmental change and reduce the corresponding number of iterations. Therefore, the frequency of change is set to be $\tau_T = 22$ in PMS and $\tau_T = 23$ in MPMS. We run each algorithm 30 times for each test instance independently. Each simulation runs for 2,500 generations (PMS: 2,200 generations, MPMS: 2,300 generations) and each strategy tracks to 100 times of environmental changes. As the dynamic test problems introduced in Sect. 4.1 are all periods, according to the parameter setting of n_T, the environment will change periodically with unequal frequency ranging from 2 to 40 in FDA and DMOP test suites, and change with the frequency of 40 in F5–F9 test suite. So to discuss the performances of different strategies in each period, the result of the experiment is divided into three stages except for the first environmental change. Each stage tracks to 33 times of environmental changes and its average is taken as the result.

Table 3 Other parameters setting for experiment

Parameter	Value
Population size (N)	Two objectives: N=100; three objectives: N=200
Frequency of change (τ_T)	25
Severity of change (n_T)	10
Dimensions of the decision space (n)	20
In FPS and PPS number of cluster	5
In FPS and PPS AR model order	3
In FPS and PPS length of history mean point series	23
In FPS probability in prediction model	0.9
In MPMS number of outdated solutions to retain	40
In MPMS number of individuals selected in prediction	30
In MPMS size of memory archive	100
In MPMS number of evaluations allocated for retrieval	10
In MPMS number of memory items retrieve	10

5.2 The statistical results of evaluation

5.2.1 Results on FDA and DMOP

The statistical results of DGD and DIGD on FDA and DMOP over 30 runs can be found in Table 4. It is clear from Table 4 that:

(1) Besides FDA2 and DMOP1 whose PS is fixed, the mean DGD and DIGD of PMS are less than the other three strategies, and the metric values are relatively average in three stages, and become more and more stable. Especially in the first stage, the metric values are greatly better than the other three strategies.

Table 4 Statistical result of DGD and DIGD metrics for four strategies on FDA and DMOP

Problems	Statistic	DGD				DIGD			
		FPS	PPS	MPMS	PMS	FPS	PPS	MPMS	PMS
FDA1	Mean	0.0459	0.0491	0.0446	**0.0117**	0.0302	0.0320	0.0289	**0.0118**
	1st stage	0.1072	0.1377	0.1073	**0.0201**	0.0609	0.0836	0.0618	**0.0186**
	2nd stage	0.0145	**0.0048**	0.0139	0.0079	0.0141	**0.0063**	0.0132	0.0087
	3rd stage	0.0159	**0.0048**	0.0127	0.0072	0.0157	**0.0062**	0.0118	0.0082
FDA2	Mean	0.0165	0.0155	**0.0148**	0.0152	0.0083	0.0086	**0.0074**	0.0077
	1st stage	0.0254	0.0225	**0.0214**	0.0215	0.0137	0.0139	**0.0112**	0.0121
	2nd stage	0.0123	0.0119	**0.0110**	0.0120	**0.0056**	0.0059	**0.0056**	**0.0056**
	3rd stage	**0.0119**	0.0121	**0.0119**	0.0121	**0.0055**	0.0060	**0.0055**	**0.0055**
FDA3	Mean	0.1386	0.1107	0.0942	**0.0755**	0.0436	0.0564	0.0282	**0.0120**
	1st stage	0.1417	0.1707	0.0978	**0.0479**	0.0673	0.1121	0.0611	**0.0177**
	2nd stage	0.1282	0.0911	0.0934	**0.0858**	0.0215	0.0257	0.0095	**0.0081**
	3rd stage	0.1460	**0.0702**	0.0915	0.0927	0.0421	0.0315	0.0141	**0.0102**
FDA4	Mean	0.1424	0.1115	0.1129	**0.1087**	0.0984	0.0904	0.0831	**0.0816**
	1st stage	0.1662	0.1333	0.1358	**0.1321**	0.1097	0.0988	0.0911	**0.0885**
	2nd stage	0.1197	0.0968	0.0973	**0.0961**	0.0914	0.0855	0.0787	**0.0783**
	3rd stage	0.1414	0.1044	0.1057	**0.0978**	0.0941	0.0869	0.0794	**0.0779**
DMOP1	Mean	**0.0027**	0.0051	0.0052	0.0039	**0.0052**	0.0076	0.0086	0.0056
	1st stage	**0.0042**	0.0075	0.0109	0.0074	**0.0072**	0.0116	0.0165	0.0078
	2nd stage	**0.0021**	0.0039	0.0023	0.0022	**0.0044**	0.0057	0.0046	0.0046
	3rd stage	**0.0018**	0.0039	0.0023	0.0021	**0.0042**	0.0056	0.0046	0.0044
DMOP2	Mean	0.0685	0.0730	0.0497	**0.0310**	0.0524	0.0572	0.0330	**0.0148**
	1st stage	0.1671	0.2099	0.1066	**0.0758**	0.1225	0.1594	0.0588	**0.0277**
	2nd stage	0.0159	**0.0045**	0.0227	0.0083	0.0157	**0.0062**	0.0211	0.0082
	3rd stage	0.0225	**0.0045**	0.0198	0.0090	0.0191	**0.0059**	0.0190	0.0085
DMOP3	Mean	0.0435	0.0456	0.0352	**0.0089**	0.0321	0.0339	0.0286	**0.0092**
	1st stage	0.0972	0.1271	0.0732	**0.0127**	0.0664	0.0892	0.0544	**0.0122**
	2nd stage	0.0163	**0.0050**	0.0139	0.0072	0.0151	**0.0064**	0.0142	0.0081
	3rd stage	0.0170	**0.0046**	0.0185	0.0067	0.0147	**0.0061**	0.0173	0.0072

(2) In the latter two stages, the DGD and DIGD of PPS are relatively average and slightly better than the PMS. But overall, the metric values are similar to PMS. On FDA2, FDA3 and DMOP1 these three test problems, PMS shows better performance than PPS.

(3) On FDA4 which is a three-objective problem, the DGD and DIGD of PMS are better than the other three strategies in each stage.

It is not hard to explain the results, mainly because the exploration operator of PMS does not need the accumulation of experience, combined with the introduction of exploitation operator, which cannot only improve the depth of search, but also improve the accuracy of search. Therefore, it is able to respond more quickly to changes in the environment. Meanwhile, with the combination of two kinds of operators, the algorithm can adapt not only to a greater degree of environmental change, but also to the environment whose degree of change is smaller or similar. Compared with FPS which uses AR model and MPMS that uses predictive gradient, the prediction of PMS is more accurate, so PMS is greatly better than the other three strategies in the first stage. For problem whose PS is fixed, the diversity of the population will hinder the rapid convergence of population. So on FDA2 and DMOP1 in these two test problems whose PS is fixed, the stored portion of the previous population will improve the convergence of the algorithm.

In addition, since the environmental periodic changes and the accumulation of experience, the convergence and diversity of PPS, MPMS and PMS will stabilize in the latter two stages due to the introduction of memory strategy. But the memory of MPMS is an incomplete memory strategy, so the performance is not as good as PPS and PMS. PPS predicts on the basis of memory, which is slightly better than the PMS on simple linear problem, but overall the difference is small. And on FDA3, the accumulated experience of PPS is not enough

Table 5 Statistical result of DGD and DIGD metrics for four strategies on F5–F8

Problems	Statistic	DGD				DIGD			
		FPS	PPS	MPMS	PMS	FPS	PPS	MPMS	PMS
F5	Mean	0.2823	0.1860	0.0770	**0.0131**	0.0916	0.0765	0.0471	**0.0167**
	1st stage	0.3855	0.5196	0.1148	**0.0176**	0.1222	0.1933	0.0455	**0.0188**
	2nd stage	0.2182	0.0231	0.0612	**0.0125**	0.0716	0.0204	0.0498	**0.0157**
	3rd stage	0.2433	0.0153	0.0551	**0.0093**	0.0811	0.0157	0.0459	**0.0155**
F6	Mean	0.0563	0.0890	0.0845	**0.0223**	0.0382	0.0395	0.0328	**0.0151**
	1st stage	0.0661	0.2363	0.2055	**0.0453**	0.0545	0.0879	0.0544	**0.0219**
	2nd stage	0.0575	0.0182	0.0189	**0.0115**	0.0312	0.0175	0.0182	**0.0123**
	3rd stage	0.0452	0.0126	0.0291	**0.0102**	0.0288	0.0131	0.0259	**0.0112**
F7	Mean	0.0589	0.1686	0.1485	**0.0198**	0.0413	0.0543	0.0741	**0.0203**
	1st stage	0.0712	0.4795	0.1314	**0.0313**	0.0460	0.1361	0.0933	**0.0289**
	2nd stage	0.0490	**0.0142**	0.2813	0.0146	0.0351	**0.0143**	0.0941	0.0172
	3rd stage	0.0565	**0.0120**	0.0327	0.0136	0.0427	**0.0125**	0.0350	0.0147
F8	Mean	0.6016	1.2140	0.2253	**0.1566**	0.2123	0.3825	0.1067	**0.0965**
	1st stage	1.1380	1.5452	0.4218	**0.2583**	0.3597	0.4645	0.1371	**0.1183**
	2nd stage	0.2944	0.9951	0.1130	**0.0934**	0.1322	0.3220	0.0849	**0.0818**
	3rd stage	0.3723	1.1017	0.1412	**0.1181**	0.1451	0.3611	0.0980	**0.0895**

Table 6 Statistical result of DGD and DIGD metrics for four strategies on F9

Problems	Statistic	DGD				DIGD			
		FPS	PPS	MPMS	PMS	FPS	PPS	MPMS	PMS
F9	Mean	0.2887	0.4251	0.1423	**0.0705**	0.1421	0.2386	0.1104	**0.0420**
	1st stage	0.4495	1.0362	0.1897	**0.1607**	0.2284	0.5660	0.1473	**0.0801**
	2nd stage	0.1354	0.1207	0.0951	**0.0292**	0.0780	0.0711	0.0877	**0.0248**
	3rd stage	0.2811	0.1183	0.1422	**0.0215**	0.1198	0.0788	0.0963	**0.0210**

to make accurate predictions, which need more times iteration of algorithm, so the average performance metrics of PMS are better than PPS on FDA3.

On FDA4 which is a three-objective problem, performance of four strategies is not very different, and PMS performs better than the other three strategies at each stage, which indicates that PMS is more suitable for high-dimensional DMOPs.

5.2.2 Results on F5–F8

The statistical results of DGD and DIGD on F5–F8 over 30 runs can be found in Table 5. F5–F8 are harder than FDA and DMOP because of the nonlinear correlation between decision variables. Since the same overhead is assigned to all the algorithms on all the test problems, the statistical results on F5–F8 are worse than those on FDA and DMOP. However, according to Table 5, it is clear that the advantage of PMS is more obvious in solving such problems. Except for the latter two stages of F7, PPS is slightly better than PMS, the other performance metrics of PMS at each stage are better than the other three strategies.

5.2.3 Results on F9

The statistical results of DGD and DIGD on F9 over 30 runs can be found in Table 6. The experimental results are similar to those in the previous section. PMS also performs relatively stable convergence and diversity on complicated problems.

5.3 Comparison of distribution of final populations obtained

To visually analyze the performance of each strategy, we choose these five typical test problems, FDA1, DMOP2, F5, F6, and F9, and draw the distribution of final populations obtained for four strategies for solving different problems at different time, shown in Figs. 6, 7, 8, 9, 10.

By comparison, it is not hard to come with the same conclusion as in Sect. 5.2, the convergence and diversity of PMS are far better than FPS and PPS at the beginning stages of

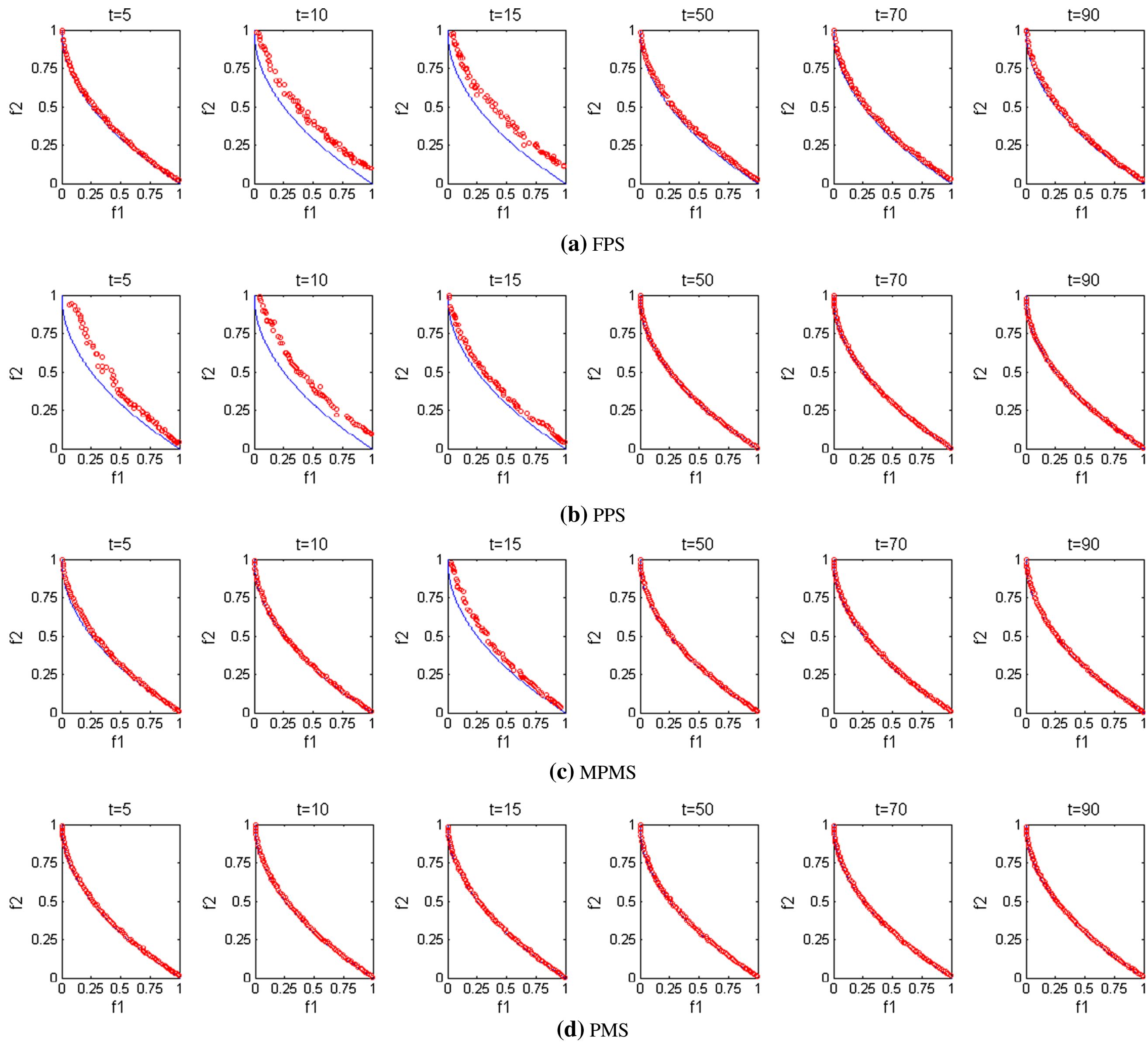


Fig. 6 Solution sets founded by four strategies at six different time steps on FDA1

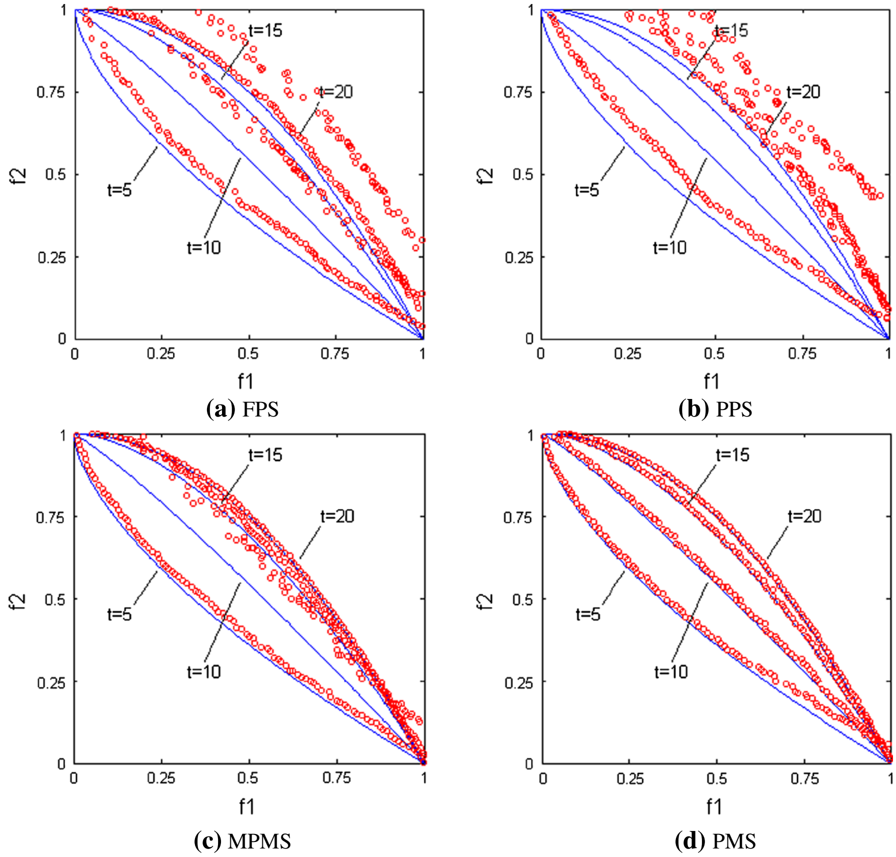
environmental change, which indicates that PMS is able to respond to environmental changes more quickly and accurately. There will be several times when the convergence and diversity of MPMS are poor, indicating that when a greater degree of environmental changes occurs, MPMS cannot make accurate prediction. To the later stage of running, PMS is the same as PPS, which has a better convergence and distribution, and slightly better than PPS on the nonlinear problems. As to the ability to solve complicated problem F9, the advantage of PMS is more obvious; the other three strategies cannot achieve better convergence and distribution, while the PMS can more accurately track to a new optimal solutions and obtain Pareto optimal solution set with better convergence and distribution. It indicates that PMS is more suitable for solving complicated nonlinear problems than the other three strategies.

5.4 Influence of different strategies

To consider the influence and necessity of each strategy, we choose FDA1, DMOP2, F6 and F7 to test. The results of average DIGD values versus the time with different strategies are shown in Fig. 11.

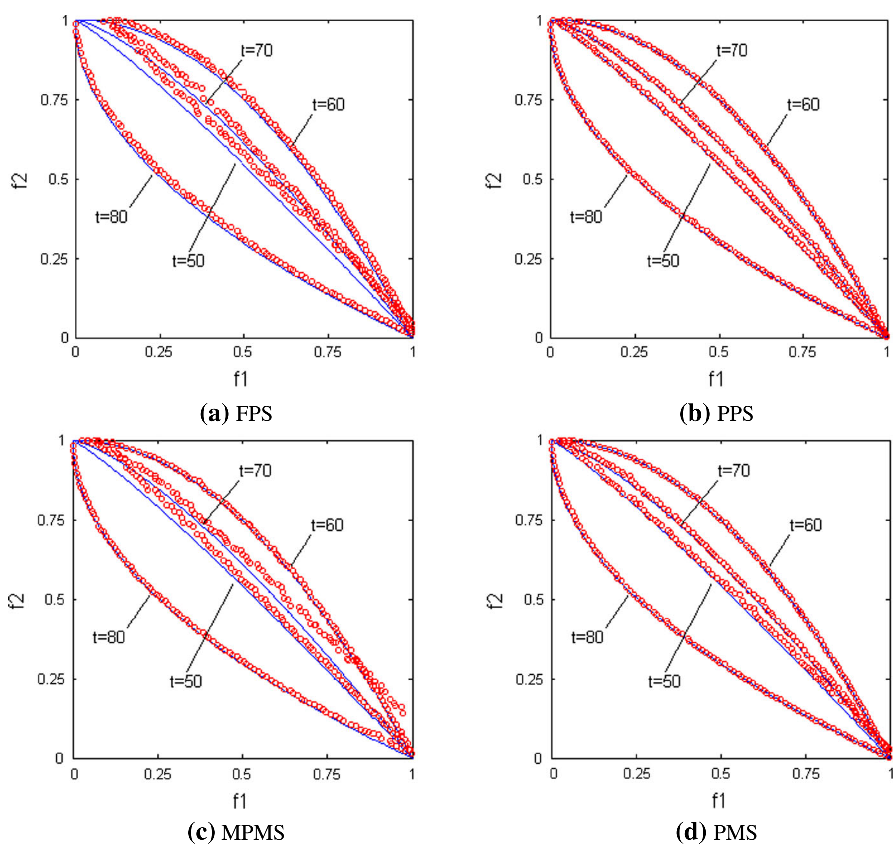
From Fig. 11, we can see that the algorithm which only employs prediction strategy performs well in terms of convergence and diversity at the early stages. Especially for solving the FDA1 and DMOP2 with linear correlation between the decision variables, the average DIGD of obtained solu-

Fig. 7 Solution sets founded by four strategies at eight different time steps on DMOP2



(a) FPS **(b)** PPS

(c) MPMS **(d)** PMS



(a) FPS **(b)** PPS

(c) MPMS **(d)** PMS

Fig. 8 Solution sets founded by four strategies at eight different time steps on F5

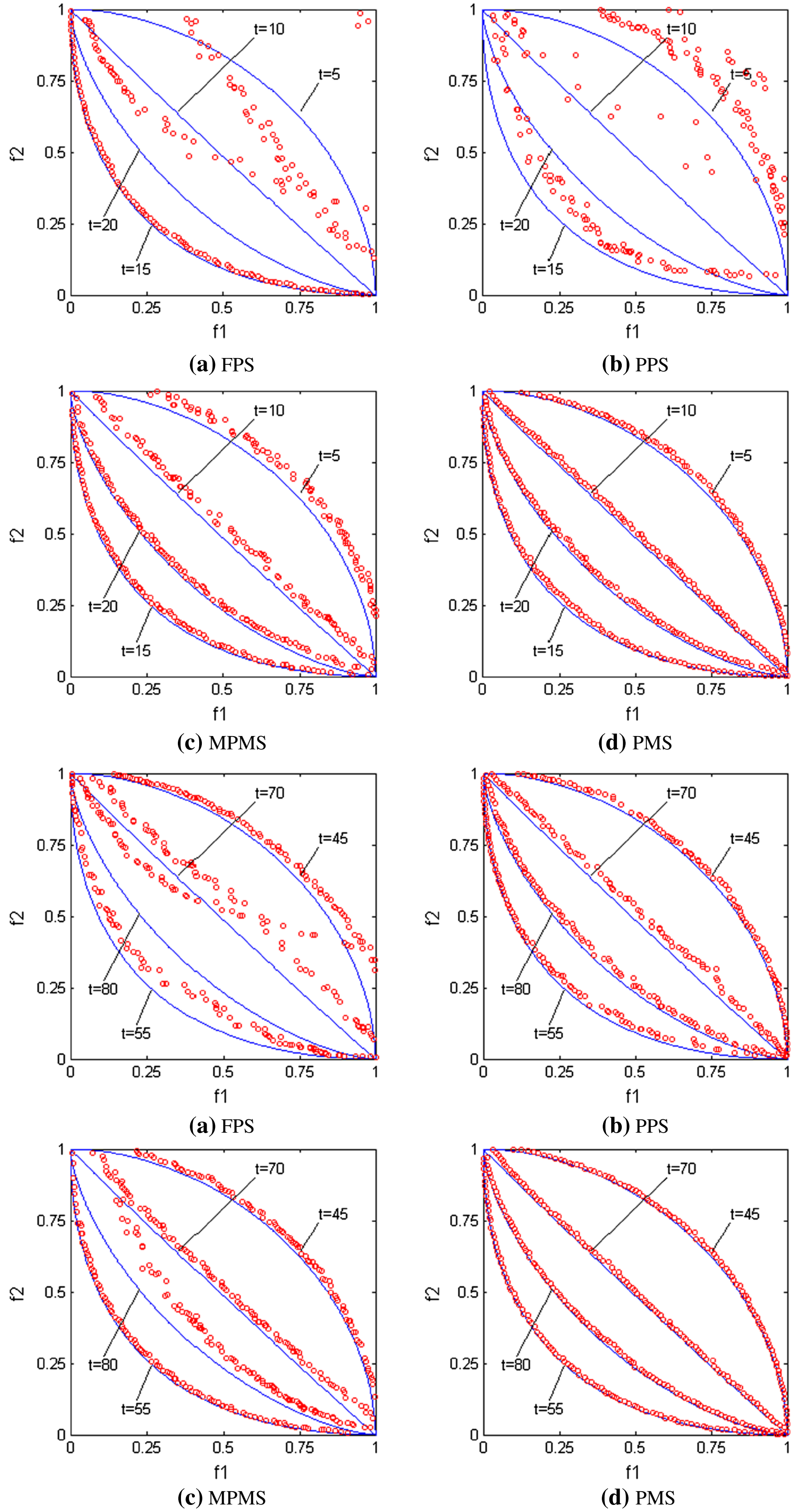
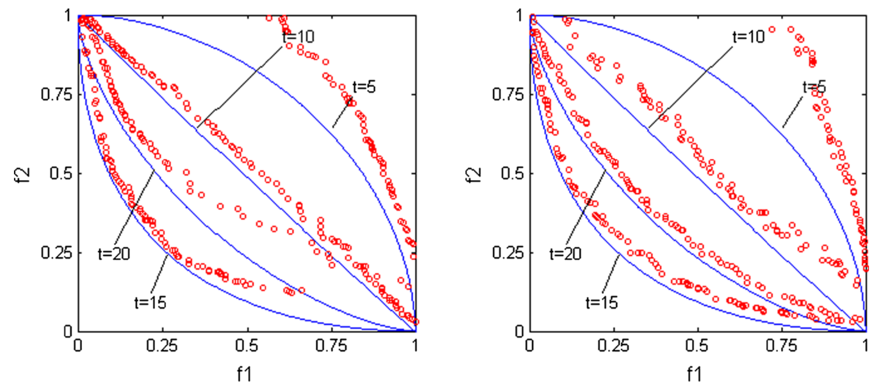
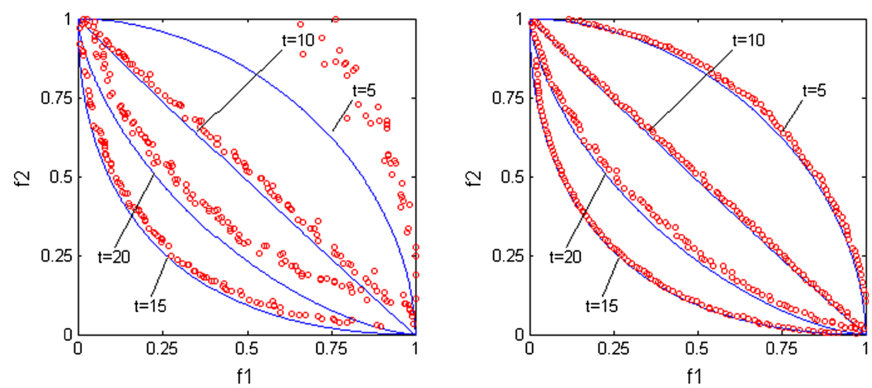


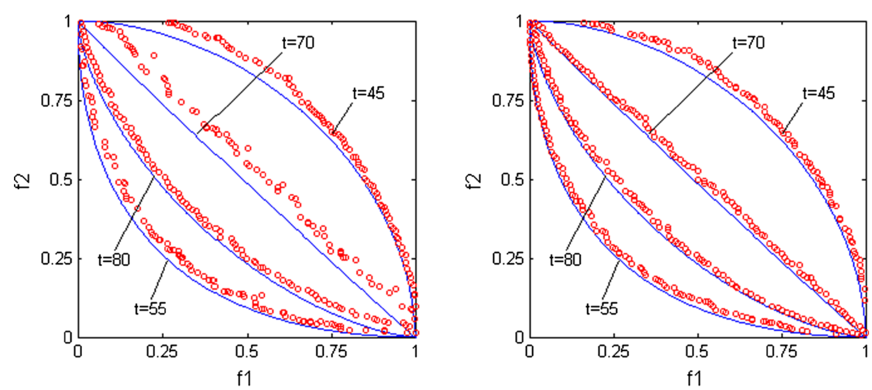
Fig. 9 Solution sets founded by four strategies at eight different time steps on F6



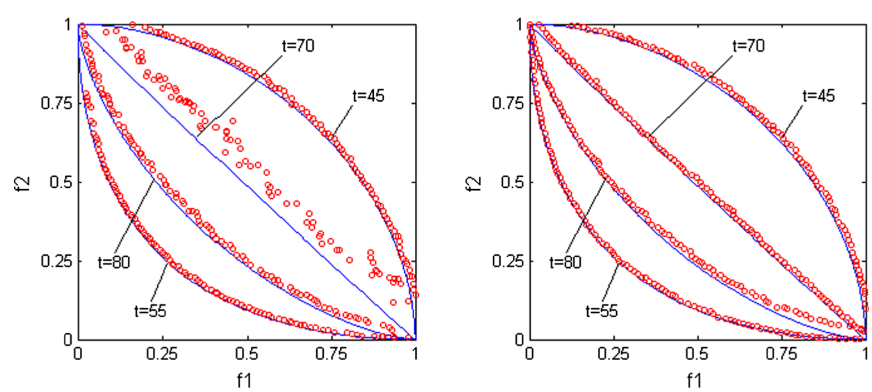
(a) FPS **(b)** PPS



(c) MPMS **(d)** PMS



(a) FPS **(b)** PPS



(c) MPMS **(d)** PMS

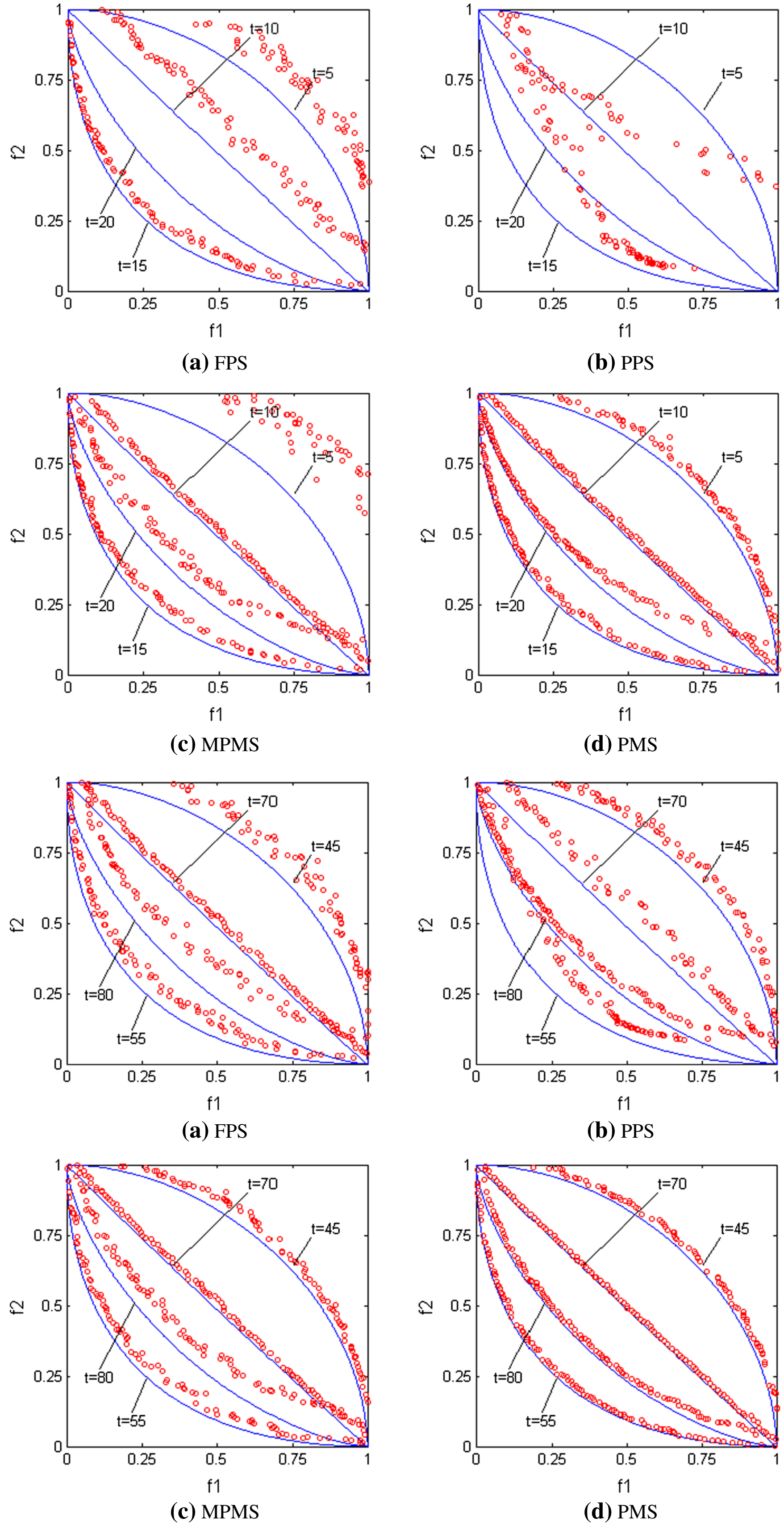


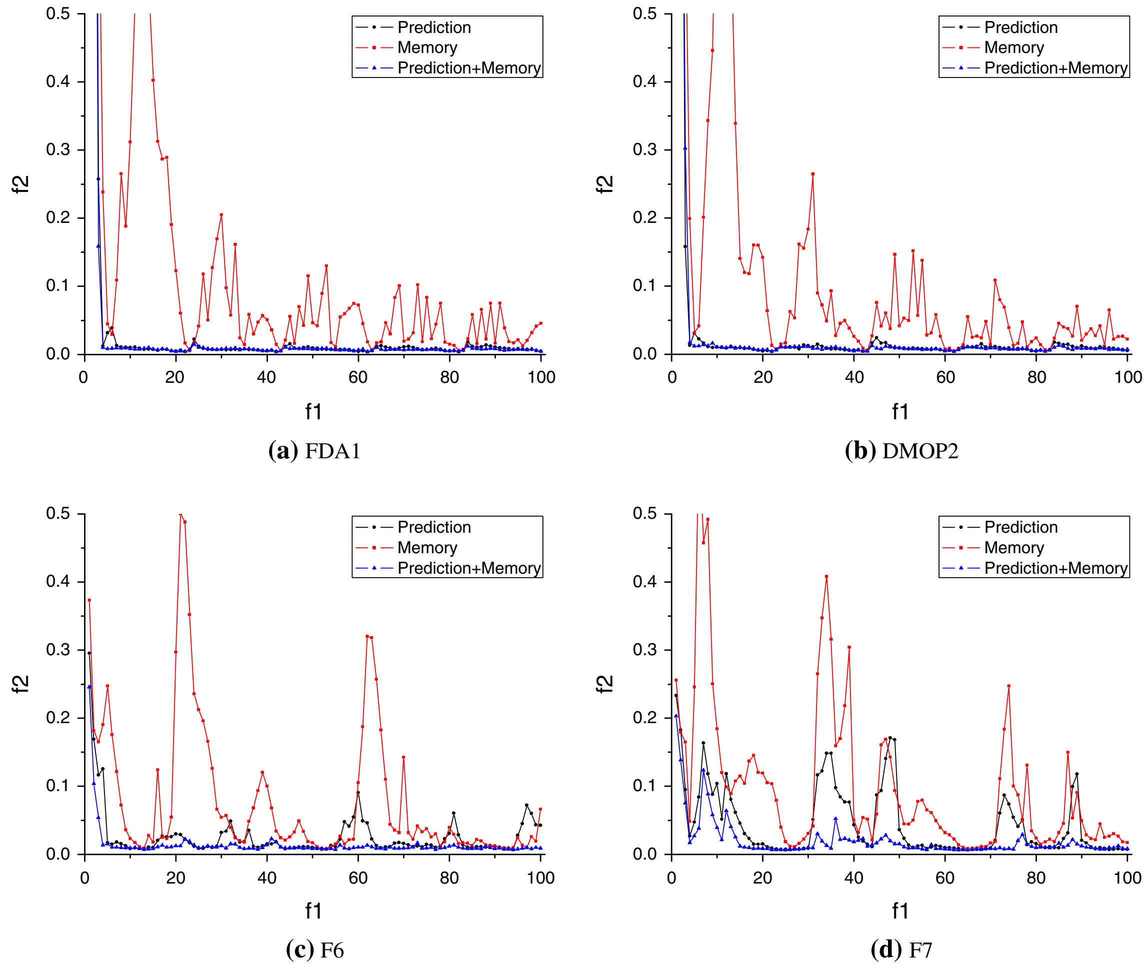
Fig. 10 Solution sets founded by four strategies at eight different time steps on F9

(a) FPS **(b)** PPS

(c) MPMS **(d)** PMS

(a) FPS **(b)** PPS

(c) MPMS **(d)** PMS



(a) FDA1 (b) DMOP2 (c) F6 (d) F7

Fig. 11 The average DIGD over 30 runs versus time on **a** FDA1, **b** DMOP2, **c** F6 and **d** F7

tions is well. But for F6 and F7 with nonlinear correlation between the decision variables, some deviations may be generated in the prediction, leading to the poor performance of the obtained solutions.

The algorithm which only employs memory strategy has the worst performance. The reason might be that the environment does not form a periodic change at the early stages. The stored history optimal solutions cannot help the population to evolve in current environment. But with the environmental periodic change, the memory strategy makes the performance of algorithm gradually improved by reusing the previously searched elite solutions. To further confirm the conclusion, we choose DMOP2 and draw the distribution of final obtained populations of the algorithm which only employs memory strategy in the same environment of different periods, shown in Fig. 12. We can see the change of performance of obtained solutions in the same environment of different time, which shows a gradual improved trend.

The strategy of combining prediction and memory has the best performance. The prediction can guarantee the algorithm to quickly respond to environmental changes, and in a timely manner to search for a new optimal solution set. Meanwhile, the memory strategy can improve the ability of dealing periodic problems and has reduced the error rate because of every prediction that may generate. The results indicate that PMS is very promising for dealing with dynamic environments.

6 Conclusions and future work

In this paper, we have proposed novel PMS to enhance the performance of multiobjective optimization evolutionary algorithms in dealing with dynamic environments. In the proposed prediction strategy, the combination of exploration and exploitation provides the basis for the algorithm to accurately predict the new optimal solution set, which accelerates the speed of response to environmental change; thus the optimal solution set can be obtained with good convergence and diversity at the initial stages. Meanwhile, the memory strategy allows the algorithm to better adapt to periodic

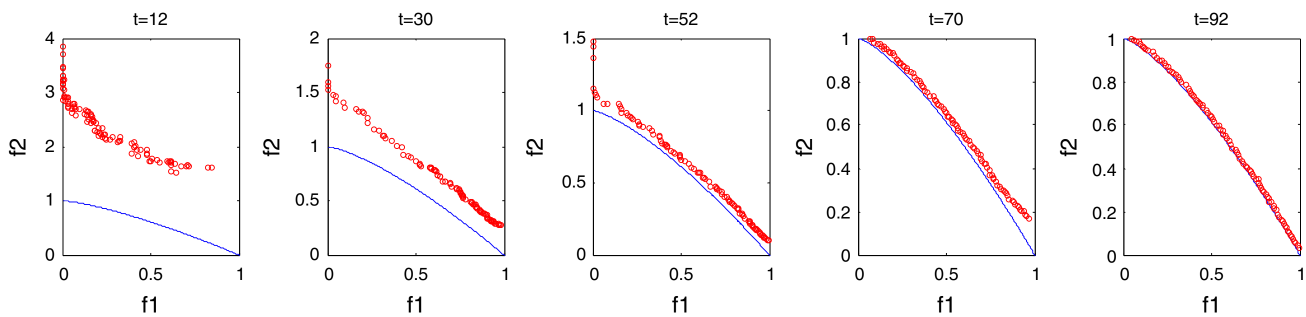


Fig. 12 Solution sets founded by memory strategy at five different time steps on DMOP2

problems, so that the algorithm can obtain a stable optimal solution set in the late stages of running. Compared with other three strategies both on seven traditional benchmark problems and on five newly appeared ones, PMS has shown faster response to the environmental changes than peer strategies in solving whether linear or nonlinear problems, with its solution set having better convergence and diversity. It has highlighted superiority for solving complicated nonlinear problems. Finally, please note that, when the population is not completely converged, the judgment errors may occur about evolutionary direction of the population in the current environment by recording the different center positions of nondominated solutions before and after the change. The prediction is not accurate and convergence is not good under some environmental changes. Therefore, our future work will be designing a more accurate prediction model. Furthermore, our focus in the future will also be the influence of different optimization algorithms for solving different problems.

Acknowledgments The authors wish to thank the support of the National Natural Science Foundation of China (No. 61379062, 61372049), the Science and Technology Project of Hunan Province (NO. 2014GK3027), the Science Research Project of the Education Office of Hunan Province (Grant No. 12A135, 12C0378), the Hunan Province Natural Science Foundation (Grant No. 14JJ2072, 13JJ8006), the Hunan Provincial Innovation Foundation For Postgraduate (Grant No. CX2013A011), and the Construct Program of the Key Discipline in Hunan Province.

References

Avdagi'c Z, Konjicija S, Omanovi'c S (2009) Foundations of computational intelligence. In: Abraham A, Hassanien A-E, Siarry P, Engelbrecht A (eds) Evolutionary approach to solving non-stationary dynamic multiobjective problems., Vol 3, Series studies in computational intelligenceSpringer, Berlin, pp 267–289

Azevedo CRB, Araujo AFR (2011) Generalized immigration schemes for dynamic evolutionary multiobjective optimization. In: IEEE Congress on evolutionary computation (CEC 2011), pp 2033–2040

Branke J (2002) Evolutionary optimization in dynamic environments. Kluwer Academic Publishers, Dordrecht

Cámara M, Ortega J, de Toro F (2009) Performance measures for dynamic multiobjective optimization. In: Bio-inspired systems: computational and ambient intelligence. Springer, New York, vol 5517, pp 760–767

Cámara M, Ortega J, de Toro F (2010) Approaching dynamic multiobjective optimization problems by using parallel evolutionary algorithms. Advances in multiobjective nature inspired computing, vol 272, pp 63–86

Coello Coello CA (2006) 20 years of evolutionary multiobjective optimization: what has been done and what remains to be done. In: Computational intelligence: principles and practice. IEEE Computational Intelligence Society, pp 73–88

Coello Coello CA, van Veldhuizen DA, Lamont GB (2007) Evolutionary algorithms for solving multiobjective problems. Springer, New York

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197

Deb K, Rao UV, Karthik S (2007) Dynamic multiobjective optimization and decision-making using modified NSGA-II—a case study on hydro-thermal power scheduling. In: Evolutionary multi-criterion optimization (EMO, 2007), LNCS, vol 4403, pp 803–817

Farina M, Deb K, Amato P (2004) Dynamic multiobjective optimization problems: test cases, approximations, and applications. IEEE Trans Evol Comput 8(5):425–442

Goh CK, Tan KC (2009) A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization. IEEE Trans Evol Comput 13:103–127

Goh CK, Tan KC (2009) Evolutionary multiobjective optimization in uncertain environments: issues and algorithms. Springer, Berlin

Greeff M, Engelbrecht AP (2008) Solving dynamic multiobjective problems with vector evaluated particle swarm optimization. In: IEEE Congress on Evolutionary Computation (CEC 2008), pp 2922–2929

Guan S, Chen Q, Mo W (2005) Evolving dynamic multiobjective optimization problems with objective replacement. Artif Intell Rev 23:267–293

Hatzakis I, Wallace D (2006) Dynamic multiobjective optimization with evolutionary algorithms: a forward-looking approach. In: Genetic and evolutionary computation conference (GECCO 2006), pp 1201–1208

Hatzakis I, Wallace D (2006) Topology of anticipatory populations for evolutionary dynamic multiobjective optimization. In: 11th AIAA/ISSMO multidisciplinary analysis and optimization conference, Portsmouth, Virginia, USA

Helbig M, Engelbrecht A (2013) Issues with performance measures for dynamic multiobjective optimization. IEEE Symposium Series on Computational Intelligence, Singapore, pp 17–24

Helbig M, Engelbrecht AP (2011) Archive management for dynamic multiobjective optimisation problems using vector evaluated particle swarm optimization. In: IEEE congress on evolutionary computation (CEC 2011), pp 2047–2054

Huang L, Suh H, Abraham A (2011) Dynamic multiobjective optimization based on membrane computing for control of time-varying unstable plants. Inf Sci 181:2370–2391

Isaacs A, Puttige V, Ray T, Smith W, Anavatti S (2008) Development of a memetic algorithm for dynamic multiobjective optimization and its applications for online neural network modeling of UAVs. In: International joint conference on neural networks (IJCNN 2008), pp 548–554

Isaacs A, Puttige V, Ray T, Smith W, Anavatti S (2008) Development of a memetic algorithm for dynamic multiobjective optimization and its applications for online neural network modeling of UAVs. In: International joint conference on neural networks (IJCNN 2008), pp 548–554

Jin Y, Branke J (2005) Evolutionary optimization in uncertain environments—a survey. IEEE Trans Evol Comput 9(3):303–317

Koo WT, Goh CK, Tan KC (2009) A predictive gradient strategy for multiobjective evolutionary algorithms in a fast changing environment. Memet Comput 2:87–110

Larrañaga P, Lozano JA (2001) Estimation of distribution algorithms: a new tool for evolutionary computation Norwell. Kluwer, Massachusetts

Li MQ, Yang SX, Liu XH (2013) Shift-based density estimation for pareto-based algorithms in many-objective optimization. IEEE Trans Evol Comput 18(3):348–365

Liu R, Zhang W, Jiao L, Liu F, Ma J (2010) A sphere-dominance based preference immune-inspired algorithm for dynamic multiobjective optimization. In: Genetic and evolutionary computation conference (GECCO 2010), pp 423-430

Ma Y, Liu R, Shang R (2011) A hybrid dynamic multiobjective immune optimization algorithm using prediction strategy and improved differential evolution crossover operator. Neural Information Processing, LNCS, vol 7063, pp 435–444

Manriquez AD, Pulido GT, Torres JGR (2010) Handling dynamic multiobjective problems with particle swarm optimization. In: 2nd International conference on agents and artificial intelligence (ICAART2010), pp 337–342

Nguyen TT, Yang SX, Branke J (2012) Evolutionary dynamic optimization: a survey of the state of the art. Swarm Evol Comput 6:1–24

Shang R, Jiao L, Ren Y, Li L, Wang L (2013) Quantum immune clonal coevolutionary algorithm for dynamic multiobjective optimization. Soft Comput 18(4):743–756

Tantar E, Tantar AA, Bouvry P (2011) On dynamic multiobjective optimization classification and performance measures. In: IEEE Congress on Evolutionary Computation (CEC 2011), pp 2759–2766

Wang Y, Li B (2009) Multi-strategy ensemble evolutionary algorithm for dynamic multiobjective optimization. Memet Comput 2:3–24

Wang Y, Li B (2009) Investigation of memory-based multiobjective optimization evolutionary algorithm in dynamic environment. In: IEEE congress on evolutionary computation (CEC 2009), pp 630–637

Wei J, Wang Y (2012) Hyper rectangle search based particle swarm algorithm for dynamic constrained multiobjective optimization problems. In: IEEE congress on evolutionary computation (CEC 2012), pp 259–266

Yang SX, Jiang Y, Nguyen TT (2012) Metaheuristics for dynamic combinatorial optimization problems. IMA J Manag Math 24:451–480

Yang SX, Li MQ, Liu X, Zheng JH (2013) A grid-based evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 17:721–736

Zhang Z (2008) Multiobjective optimization immune algorithm in dynamic environments and its application to greenhouse control. Appl Soft Comput 8(2):959–971

Zhang Q, Zhou A, Jin Y (2008) RM-MEDA: a regularity model based multiobjective estimation of distribution algorithm. IEEE Trans Evol Comput 12(1):41–63

Zheng B (2007) A new dynamic multiobjective optimization evolutionary algorithm. In: Third international conference on natural computation (ICNC 2007), pp 565–570

Zhou A, Jin Y, Zhang Q (2013) A population prediction strategy for evolutionary dynamic multiobjective optimization. IEEE Trans Cybern 44(1):40–53

Zhou A, Jin Y, Zhang Q, Sendhoff B, Tsang E (2007) Prediction-based population re-initialization for evolutionary dynamic multiobjective optimization. In: Evolutionary Multi-Criterion Optimization (EMO 2007), pp 832–846