2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

A Population Diversity Maintaining Strategy Based on Dynamic Environment Evolutionary Model for Dynamic Multiobjective Optimization

Zhou Peng, Jinhua Zheng, Juan Zou
College of Information Engineering
Xiangtan University
Xiangtan, China
Email: zpeng@xtu.edu.cn, jhzheng@xtu.edu.cn(corresponding author)
zoujuan@xtu.edu.cn

***Abstract*—Maintaining population diversity is a crucial issue for the performance of dynamic multiobjective optimization algorithms. However traditional dynamic multiobjective evolutionary algorithms usually imitate the biological evolution of their own, maintain population diversity through different strategies and make the population be able to track the Pareto optimal solution set after the change efficiently. Nevertheless, these algorithms neglect the role of dynamic environment in evolution, lead to the lacking of active and instructional search. In this paper, a population diversity maintaining strategy based on dynamic environment evolutionary model is proposed (DEE-PDMS). This strategy builds a dynamic environment evolutionary model when a change is detected, which makes use of the dynamic environment to record the different knowledge and information generated by population before and after environmental change, and in turn the knowledge and information guide the search in new environment. The model enhances population diversity by guided fashion, makes the simultaneous evolution of the environment and population. A comparison study with other two state-of-the-art strategies on five test problems with linear or nonlinear correlation between design variables has shown the effectiveness of the DEE-PDMS for dealing with dynamic environments.**

I. INTRODUCTION

Many real-world problems are dynamic multiobjective optimization problems (DMOPs), with not only the conflict among multiple objectives but also the objective, constraint and related parameters may change over time [1]. How to track the Pareto optimal solution set after the change is an important and challenging issue. On these issues, the researched objective is changing intricately. The goal of traditional evolutionary algorithm is to make the population gradually converge to get a satisfactory solution set ultimately, 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 changes, which are the challenges of traditional evolutionary algorithm in a dynamic environment [2]. In order to track the optimal solution in a timely manner after changes, researchers need to make some adjustments on the traditional static multiobjective algorithm [3], [4], so that it can quickly respond to environmental changes.

In recent years, researchers have designed many new ways on the basis of static algorithms to solve DMOPs [5]-[18], such as random initialization, hyper mutation, dynamic migration, memory and prediction et al. Those strategies have been proved by several researchers to be some effective methods to solve DMOPs. However, there are many defects in these methods along with the development of DMOPs, which mainly reflected in the following respects. Firstly, random initialization, hyper mutation, dynamic migration et al. strategies are all a blind way to enhance population diversity without a right guidance, the performance of convergence are unsatisfactory when dealing with more complex DMOPs. Secondly 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 cycle of changing environment, population is still in the process of blind evolution, and algorithm is difficult to obtain a good convergence. Lastly, methods that based on prediction generate a new optimal solution set by the prediction model for the evolution of population, and help algorithm to respond quickly to new changes. So far, the accuracy of prediction is the main difficulty, how to design a more accurate prediction model is still the focus of the present research.

To solve these problems, on the premise of less history information and utilizing the characteristics of the evolutionary environment itself, this paper proposes a novel population diversity maintaining strategy based on dynamic environment evolutionary model, referred to DEE-PDMS, which enhances population diversity by guided fashion, makes the population can respond quickly to the different degrees of environmental changes. Current dynamic multiobjective optimization algorithms did not consider the role of dynamic environment for the evolutionary population. Actually, the affect of environment on the evolutionary individuals is very important and individuals must survive and evolve in a specific environment. It is because of the wonderful interaction between the nature environment and the biology that makes biomass have such a present perfect structure. Therefore, how to research from the perspective of the dynamic environment, use the dynamic environmental knowledge to guide the evolution of population in the new environment and accelerate convergence of population is the research focus in this paper.

The rest of the paper is organized as follows. Section II provides definitions that are required for the rest of the paper. Section III describes the dynamic environment evolutionary model. Section IV describes the implementation of the evolutionary model. Section V gives the detailed description of DEE-PDMS. Section VI introduces the test problems and evaluation metric. Section VII gives experimental results and analysis. Section VIII outlines the conclusions and future work.

II. BACKGROUND

A minimization problem is considered here without loss of generality. The dynamic multiobjective optimization problem [1] can be described as:

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

where t is the time variable, $x = (x_1, x_2, ..., x_n)$ is the n-dimensional decision variables bounded by the decision space Ω, $F = (f_1, f_2, ..., f_m)$ presents the set of m objectives to be minimized, the functions of $g_i \le 0\ \ i = 1,2,...,p$ and $h_j = 0\ \ j = 1,2,...,q$ 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* $f_i(p) \le f_i(q)\ \ \forall i = \{1,2,...,m\}$ and $f_j(p) < f_j(q)\ \exists j \in \{1,2,...,m\}$.

Definition 2: Pareto Optimal Set (PS). x is the decision variable, Ω is the decision space, F is the objective function, thus the PS [3] is the set of all nondominated solutions and is defined mathematically as:

$$\text{PS} := \{x \in \Omega \mid \nexists x^* \in \Omega, F(x^*) \prec F(x)\}$$

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

$$\text{PF} := \{y = F(x) \mid x \in \text{PS}\}$$

III. DYNAMIC ENVIRONMENT EVOLUTIONARY MODEL

Environment is relative to the case of something, refers to the all external matters which have an affect on a certain matter. In ecology, environment refers to external matters such as the surrounding ecosystem which have affect on biological communities. In our dynamic environment evolutionary model, the environment refers to a group of entities which can guide and promote the evolution of population. Especially after environmental changes, it can guide the evolution and convergence of population in the new environment.

Evolutionary population must survive and evolve in a specific environment. Environment plays constraint, facilitating and guiding roles for the evolution of population, and these three environmental roles are completely different. Constraint is mainly used to ensure the legitimacy of individuals, facilitating is mainly used to enhance the efficiency of the evolution, guiding is mainly used to improve the distribution of evolutionary population. At the same time, the evolutionary population is counteractive to the evolutionary environment, which is mainly shown in the impact on the attributes of evolutionary environment, such as the changes of the current evolutionary state, the update of environmental knowledge, etc.

In dynamic environment, how to maintain the population diversity after environmental changes is the key to solve the DMOPs, and when a change is detected, environmental information and environmental knowledge will also make a difference. How to make full use of the information to help population adapt to the new environment and enhance population diversity by guided fashion plays an important role for solving DMOPs.

Figure 1 shows a general framework of dynamic environment evolutionary model. The evolutionary model consists of two different kinds of environment before and after environmental changes and the evolutionary population in dynamic environment. Environmental elements include environmental knowledge, environmental evaluation and environmental regulation in new environment. Among them, the environmental knowledge can be divided into static knowledge and dynamic knowledge. Static knowledge is the preset environmental attributes which maintain constant values in the process of environmental change, such as environmental capacity, dimensions, etc. Dynamic knowledge is the environmental attributes which affected by population in the process of environmental change, such as the size of environment domain, direction of environmental change, new generated individuals for guiding evolution.

Environmental evaluation mechanism evaluates the living conditions of population or individuals according to environmental knowledge, such as individual location in the environment, the entire population distribution.

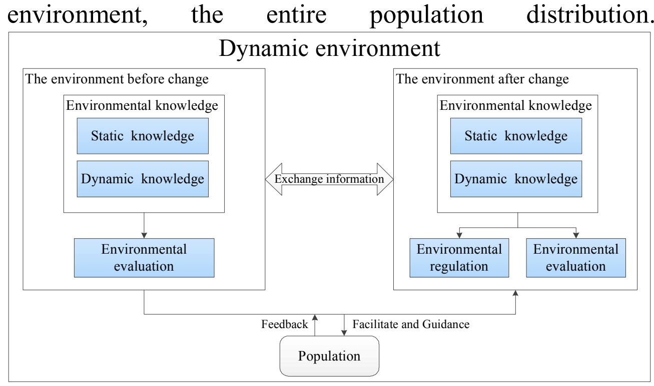


Fig. 1. A general framework of dynamic environment evolutionary model

Environmental evaluation is to prepare for guiding evolution. Environmental regulation in new environment means that individuals need to make the corresponding change in order to adapt to the new environment. Dynamic environment exchanges information between two different kinds of environment before and after change to facilitate and guide the evolution of population. In turn, population will send the feedback information which generated in process of evolution to environment, update the environmental knowledge and achieve co-evolution.

IV. Implementation of the Evolutionary Model

Each individual in dynamic environment has a living space. Here we use a mechanism which is similar to the grid—environment domain to store the individuals in dynamic environment. Environment domain consists of many same grids which called the unit domain. The dimension of environment domain and unit domain is the same as objective dimension. The position of individual in the environment domain will also be changed accordingly when the environment changed. Therefore, in dynamic environment, the range of environment domain and the size of unit domain are codetermined by the different distribution of the population before and after environmental change. Bottom and top boundaries of each dimension in environment domain are calculated as follows:

$$\begin{aligned} lb_i &= \min(P_i) - (\max(P_i) - \min(P_i)) / (2 \times num) \\ ub_i &= \max(P_i) + (\max(P_i) - \min(P_i)) / (2 \times num) \end{aligned} \quad (1)$$

where *num* is the number of unit domain on each dimension in the objective space, the higher the objective dimension, the smaller the value of *num*. For example, num can be set 40 for two objectives and can be set 10 for three objectives. $\min(P_i)$ and $\max(P_i)$ denote the *i*-dimensional objective minimum and maximum of population *P* in the two different kinds of environments before and after change, namely $\min(P.oldF_i,\ P.newF_i)$ and $\max(P.oldF_i,\ P.newF_i)$. As shown in figure 2, the size of unit domain on the *i*-dimensional objective is $area\_size_i = (ub_i - lb_i) / num$.

In dynamic environment, each individual maybe belongs to two different unit domains before and after environmental change. Therefore, we denote *indiv.old_area* as the unit domain which individual *indiv* belongs to before environmental change, *indiv.new_area* as the unit domain which individual *indiv* belongs to after environmental change. According to the boundary of environment domain and the size of unit domain, the two different unit domain positions (domain coordinates) of each individual can be determined. The domain coordinates of *indiv.old_area* and *indiv.new_area* on the *i*-dimensional objective can be calculated as formula (2).

$$\begin{aligned} indiv.old\_area_i &= \lfloor (indiv.oldF_i - lb_i) / area\_size_i \rfloor \\ indiv.new\_area_i &= \lfloor (indiv.newF_i - lb_i) / area\_size_i \rfloor \end{aligned} \quad (2)$$

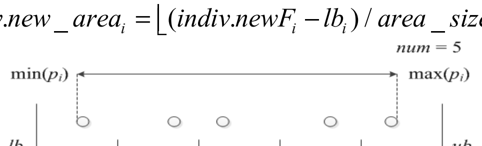


Fig. 2. The environment domain set on *i*-dimensional objective

where $indiv.oldF_i$ and $indiv.newF_i$ are respectively the *i*-dimensional objective values before and after environmental change.

The environment domain and unit domain have been set, and then the various elements of composing dynamic environment and its implementation are introduced.

A. Environmental Knowledge

Environmental knowledge is an important part of the environment, which records the information in the current dynamic environment. In our approach, environmental knowledge is divided into two types: the environment domain knowledge and unit domain knowledge. Among them, the environment domain knowledge is divided into static and dynamic environment domain knowledge. Static environment domain knowledge includes environmental capacity, the number of unit domains on each dimension and other preset environmental attributes. Dynamic environment domain knowledge includes the bottom boundary and top boundary of environment domain on each dimension, the size of unit domain, the direction of environmental change, the new generated individuals for guiding the evolution, and other environmental attributes which are affected by population. Here, the new individuals are a series of reinitialized individuals to help population adapt to the new environment after environmental change and accelerate the convergence of population and individuals, which will be described in detail in Section B.

Unit domain knowledge is dynamic knowledge, which includes the number of individuals in each unit domain, representative individual and so on. The representative individual is the optimal individual in a unit domain. Here we set the individual with nearest Euclidean distance to the origin of unit domain as representative individual. The origin of unit domain is the minimum on each dimension.

B. Environmental Evaluation

In dynamic environment evolution model, the evaluation mechanism not only needs to evaluate the fitness of the population, but also needs to evaluate the living conditions of population and individuals according to environmental knowledge, and to prepare for guiding evolution. Evaluation mechanism is divided into two types: evaluation for individual and evaluation for population.

Evaluation for individual refers to calculate two different unit domain coordinates for each individual after environmental change according to the formula (2), and give the environment feedback to construct a new dynamic environment.

Evaluation for population refers to evaluate the distribution of population in the new environment according to the environmental knowledge, and then generate a new series of guide-individuals to prepare for guiding evolution. The new guide-individuals are defined as:

$$\begin{aligned} init_k^t &= x_k^t + \left|(C_k^t - C_k^{t-1})\,\text{Gaussian}\right|, \ if\ C_k^t - C_k^{t-1} > 0 \\ init_k^t &= x_k^t - \left|(C_k^t - C_k^{t-1})\,\text{Gaussian}\right|, \ if\ C_k^t - C_k^{t-1} < 0 \end{aligned} \quad (3)$$

where x_k^t is the individual at time *t*, $k = 1,2,\ldots,n$, *n* is the dimensions of the decision space. Gaussian is a random number generated from a standard normal distribution (mean = 0, variance = 1)，which has been testified in [19] to be a good strategy to enhance the ability of elaborate search. C_k^t is the center of nondominated solutions obtained at time *t*, which can be defined as formula (4):

$$C_k^t = \frac{1}{\left|P_{N-\text{dominance}}^t\right|} \sum_{x_k^t \in P_{N-\text{dominance}}^t} x_k^t \qquad (4)$$

where $\left|P_{N-\text{dominance}}^t\right|$ is the size of nondominated solutions.

Similarly, the domain coordinates of new guide-individuals are also calculated.

In this way, we use the possible correlation between environmental changes to produce a series of guide-individuals. These individuals will be served as the alternative individuals in the process of environmental facilitating and guiding, try to help the population to adapt to the new environment, accelerate convergence of population to the new PF.

C. Environmental Regulation

In dynamic environment, different problems have different regulations. The location and distribution of population in the new environment domain may not be suitable for its evolution and convergence. Therefore, population needs to make the corresponding change in order to adapt to the new environment.

As shown in Figure 3, like people's psychological reactions in real life, some individuals want to return to the past environment and continue to survive and evolve, consider that the environment before change is more conducive for evolution. While some individuals do not want to return to the past environment, at the same time they are also confused about where they should go. There is also a part of individuals did not want to make any change, they consider that the current environment is an ideal evolutionary environment.

Therefore, we need to divide the current population into three subpopulations according to the different behavioral characteristics of each individual when the environment changed. Meanwhile, in order to maintain the distribution of subpopulations and avoid crowding of the solution set, the three subpopulations need to be more evenly divided. The division strategies of subpopulations are as follows (illustrated by the example of two objectives):

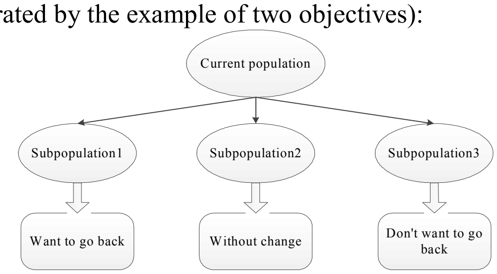


Fig. 3. The division of population

The sizes of three subpopulations are respectively set to 30, 40 and 30 (three objectives: 60, 80 and 60).

For the subpopulation2 which is without any change: we gather directly 40 nondominated individuals whose crowding-distance [24] is the largest from the original population to subpopulation2.

For the subpulation1 which want to go back and the subpopulation3 which do not want to go back: Algorithm 1 gives a detailed procedure of this strategy.

Algorithm 1 *SubpopulationDivision*

```
Require: ND (population without division), q (picked individual)
for all q ∈ ND do
    p := q->next
    for p != null do
        flag := false
        if p is domain-adjacent with q then
            for all k ∈ ND\p do
                if k.new_area = p.new_area then
                    swap (p, q->next)
                    flag := true
                    break
                end if
            end for
            if flag = false then
                p := p->next
            end if
        else
            swap (p, q->next)
            break
        end if
    end for
end for
Select the top 30 individuals in ND as subpopulation1, the rest of 30 individuals as subpopulation3.
```

where the domain-adjacent is defined as follows:

Definition 4. *U* and *V* are any two individuals in the environment domain, *U* is domain-adjacent with *V*, *iff* $|U.new\_area_i - V.new\_area_i| <= min\_diff(i)$. *min_diff*(*i*) is the minimum difference on each dimension between any two unit domain coordinates. The unit domain here refers to the unit domain where there is individual existence.

Figure 4 is an example about division of the subpopulations. Select the first individual A, and then select the second individual to compare with A, the second individual is assumed to be B. Since B is domain-adjacent with A, and its unit domain does not include multiple individuals, therefore, B is discarded. Next select the individual C, C is not domain-adjacent with A, so the C will be divided into the same subpopulation with A and serves as the next compared individual. Similarly, E is not domain-adjacent with C and is divided into the same subpopulation. Despite F is domain-adjacent with E, its unit domain includes another individual G, so F will be divided into the same subpopulation with A, C and E. Thus division end, A, C, E, F are divided into the same subpopulation, B, D, G, H are divided into another subpopulation.

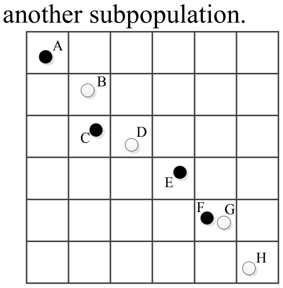


Fig. 4. An example about division of the subpopulations

It is worth noting that the environmental regulation in this paper is clearly different with the random division of subpopulations such as charged PSO [20]. The environmental regulation considers the use of different subpopulations to adapt to different environmental changes, at the same time, takes into account the distribution of the solution set, digs and utilizes the environmental knowledge to guide the evolution.

D. *Environmental Facilitating and Guiding*

Dynamic environmental facilitating and guiding refer to guide the different subpopulations to evolve toward their desired environments according to the new environmental knowledge and regulation, so that to enhance population diversity by guided fashion. In environmental facilitating and guiding, we design the recombination operator as follows:

Let $U=(u_1,u_2,...,u_n)$ and $V=(v_1,v_2,...,v_n)$ represent the parent individuals for recombination, n is the dimension of decision space, then the offspring is defined as $W=(w_1,w_2,...,w_n)$, $w_i=a(u_i-v_i)+v_i$. Where a is a random number between 0.8 and 1. It is not hard to find that, w_i located between u_i and v_i, and closer to u_i. Because most of the multiobjective optimization problems meet the connectivity [21], that is to say, the solutions that are distributed like neighborhood in the decision will be also distributed like neighborhood when mapped to the objective space. Therefore, the new generated individual is more likely located in the area that between U and V and more close to U.

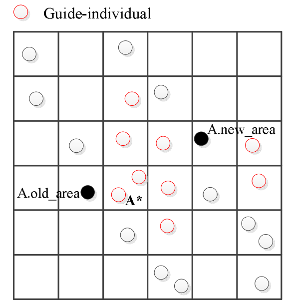
In addition, for the different subpopulations, the strategy to select the parent individuals to be recombined would be different:

For the individual sub1_indiv that want to go back in subpopulation1, firstly, we need to calculate which unit domain coordinates of guide-individuals are located between sub1_indiv.old_area$_i$ and sub1_indiv.new_area$_i$, and then select the individual that leaves closest to sub1_indiv.old_area$_i$. If multiple individuals are in the same unit domain, select the representative individual in the unit domain.

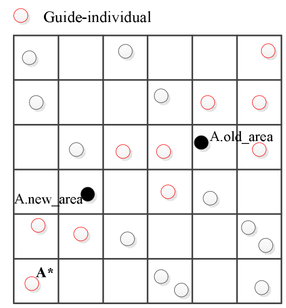
For the individual sub2_indiv that did not want to make any change in subpopulation2, need not any recombination operation.

For the individual sub3_indiv that don't want to go back in subpopulation3: if sub3_indiv.old_area$_i$ > sub3_indiv.new_area$_i$, we need to calculate which unit domain coordinates of guide-individuals are greater than sub3_indiv.old_area$_i$, and then select the individual that leaves farthest to sub3_indiv.old_area$_i$. if sub3_indiv.old_area$_i$ < sub3_indiv.new_area$_i$, we need to calculate which unit domain coordinates of guide-individuals are less than sub3_indiv.old_area$_i$, and then select the individual that leaves farthest to sub3_indiv.old_area$_i$. Similarly, if multiple individuals are in the same unit domain, select the representative individual in the unit domain.

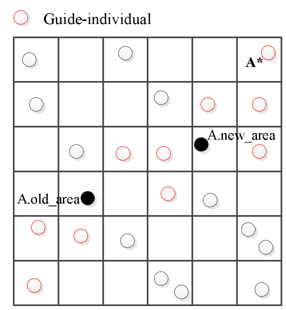
Figure 5 is an example about recombination strategy. For individual A, the selected another parent individual for recombination is A*.



(a) Subpopulation1



(b) Subpopulation3 and A.old_area$_i$ > A.new_area$_i$



(c) Subpopulation3 and A.old_area$_i$ < A.new_area$_i$

Fig. 5. Example about different recombination

V. Detailed Description of DEE-PDMS

DEE-PDMS iterates under the general framework of DMOEA, the purpose is to obtain new initial population after each environmental change, so that the new population can quickly respond to changes in the environment. The DEE-PDMS is described in detail as Algorithm 2.

Algorithm 2 DEE-PDMS

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

Output: P_t, updated population.

Initialization set t := 0; initialize a population p_0; set iteration counter gt := 0.

1: Detect changes in the environment, if environment has not changed, turn to step 6; else construct the dynamic environment according to formula (2).

2: Environmental evaluation, generate guide-individuals.

3: *SubpopulationDivision* for current population.

4: Recombine.

5: Set $P_t=P_{sub1}\cup P_{sub2}\cup P_{sub3}$.

6: Optimize population with optimize algorithm, this paper chooses RM-MEDA [23] as the MOEA optimizer.

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

VI. TEST INSTANCES AND PERFORMANCE METRICS

A. Test Instances

In this paper, five benchmark functions were selected of various DMOOP types [1] to compare the performance. These functions include DMOP1-DMOP3 [12] three functions with linear correlation between the decision variables and F6, F7 [15] two functions with nonlinear correlation between the decision variables. Table 1 lists all the five benchmark functions and their PF and PS in details.

TABLE I. TEST PROBLEMS OF DMOP, F6 AND F7

Problems	**Search Space**	**Objectives, PF, PS**	**Remarks**
DMOP1	$[0,1]\times[-1,1]^{n-1}$	$f_1(x)=x_1, f_2(x)=g\cdot h$ $g(x)=1+9\sum_{i=2}^{n}x_i^2, h(x)=1-\left(\frac{f_1}{g}\right)^{H(t)}$ $H(t)=1.25+0.75\sin(0.5\pi t), t=\lfloor \tau/\tau_T \rfloor/n_T$ $PS(t): 0\le x_1\le 1, x_i=0, i=2,\dots,n$ $PF(t): 0\le f_1\le 1, f_2=1-f_1^{H(t)}$	two objectives PF changes PS is fixd
DMOP2	$[0,1]\times[-1,1]^{n-1}$	$f_1(x)=x_1, f_2(x)=g\cdot h$ $g(x)=1+\sum_{i=2}^{n}(x_i-G(t))^2, h(x)=1-\left(\frac{f_1}{g}\right)^{H(t)}$ $G(t)=\sin(0.5\pi t), H(t)=1.25+0.75\sin(0.5\pi t), t=\lfloor \tau/\tau_T \rfloor/n_T$ $PS(t): 0\le x_1\le 1, x_i=G(t), i=2,\dots,n$ $PF(t): 0\le f_1\le 1, f_2=1-f_1^{H(t)}$	two objectives PF changes PS changes
DMOP3	$[0,1]\times[-1,1]^{n-1}$	$f_1(x_r)=x_r, f_2(x\setminus x_r)=g\cdot h$ $g(x)=1+\sum_{i=1}^{x\setminus x_r}(x_i-G(t))^2, h(x)=1-\sqrt{\frac{f_1}{g}}$ $G(t)=\sin(0.5\pi t), r=\bigcup(1,2,\dots,n), t=\lfloor \tau/\tau_T \rfloor/n_T$ $PS(t): 0\le x_1\le 1, x_i=G(t), i=2,\dots,n$ $PF(t): 0\le f_1\le 1, f_2=1-\sqrt{f_1}$	two objectives PF is fixed PS changes
F6	$[0,5]^n$	$f_1(x)=\lvert x_1-a\rvert^{H(t)}+\sum_{i\in I_1}y_i^2$ $f_2(x)=\lvert x_1-a-1\rvert^{H(t)}+\sum_{i\in I_2}y_i^2$ $y_i=x_i-b-1+\lvert x_1-a\rvert^{H(t)+\frac{i}{n}}, H(t)=1.25+0.75\sin(\pi t)$ $a=2\cos(1.5\pi t)\sin(0.5\pi t)+2, b=2\cos(1.5\pi t)\cos(0.5\pi t)+2$ $t=\lfloor \tau/\tau_T \rfloor/n_T$ $I_1=\{i \mid 1\le i\le n, i\ \ is\ \ odd\}, I_2=\{i \mid 1\le i\le n, i\ \ is\ \ even\}$ $PS(t): a\le x_1\le a+1, x_i=b+1-\lvert x_1-a\rvert^{H(t)+\frac{i}{n}}, i=2,\dots,n$ $PF(t): f_1=s^{H(t)}, f_2=(1-s)^{H(t)}, 0\le s\le 1$	two objectives PF changes PS changes
F7	$[0,5]^n$	$f_1(x)=\lvert x_1-a\rvert^{H(t)}+\sum_{i\in I_1}y_i^2$ $f_2(x)=\lvert x_1-a-1\rvert^{H(t)}+\sum_{i\in I_2}y_i^2$ $y_i=x_i-b-1+\lvert x_1-a\rvert^{H(t)+\frac{i}{n}}, H(t)=1.25+0.75\sin(\pi t)$ $a=1.7(1-\sin(\pi t))\sin(\pi t)+3.4, b=1.4(1-\sin(\pi t))\cos(\pi t)+2.1$ $t=\lfloor \tau/\tau_T \rfloor/n_T$ $I_1=\{i \mid 1\le i\le n, i\ \ is\ \ odd\}, I_2=\{i \mid 1\le i\le n, i\ \ is\ \ even\}$ $PS(t): a\le x_1\le a+1, x_i=b+1-\lvert x_1-a\rvert^{H(t)+\frac{i}{n}}, i=2,\dots,n$ $PF(t): f_1=s^{H(t)}, f_2=(1-s)^{H(t)}, 0\le s\le 1$	two objectives PF changes PS changes

B. Performance Metrics

Some metrics have been designed for dynamic optimization [22]. In this paper, we firstly introduce the inverted generational distance (DIGD) [15] metric for DMOPs. The DIGD metric is defined as follow:

$$\mathrm{DIGD}=\frac{1}{|T|}\sum_{t\in T}\mathrm{IGD}(P^{t*},P^{t})$$
$$\mathrm{IGD}(P^{t*},P^{t})=\frac{\sum_{v\in P^{t*}}d(v,P^{t})}{\left|P^{t*}\right|} \quad (5)$$

where $d(v, P^t) = \min_{u \in P^t} \|F(v) - F(u)\|$ is the distance between v and P^t, P^{t*} is a set of uniformly distributed Pareto optimal points in the PF at time t, P^t is the solutions obtained at time t. 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.

VII. EXPERIMENTS

In this section, DEE-PDMS will be compared to other two strategies: forward-looking prediction strategy (FPS) proposed in [13], population prediction strategy (PPS) proposed in [15]. We choose RM-MEDA [23] as the MOEA optimizer. In DEE-PDMS, the number of unit domain on each dimension is 40, the number of guide-individual is 100. Population size N = 100, frequency of change τ_T =25, severity of change n_T=10. Other parameter settings of two strategies use the given setting in [15].

Since the DEE-PDMS in this paper need to consume evaluations in the generating guide-individuals, 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 τ_T = 24 in DEE-PDMS. We run each algorithm 20 times for each test instance independently. Each simulation runs for 2500 generations (DEE-PDMS: 2400 generations) and each strategy tracks to 100 times of environmental changes. The statistical results of DIGD over 20 runs can be found in Table 2.

TABLE II. STATISTICAL RESULT OF DIGD FOR THREE STRATEGIES

Problems	Statistic	FPS	PPS	DEE-PDMS
DMOP1	Mean	5.24E-3	7.72E-3	**5.01E-3**
	Variance	8.24E-8	6.74E-7	**4.65E-8**
DMOP2	Mean	5.23E-2	5.68E-2	**3.13E-2**
	Variance	8.47E-6	5.72E-5	**6.32E-6**
DMOP3	Mean	3.17E-2	3.40E-2	**1.70E-2**
	Variance	7.36E-6	6.59E-5	**4.15E-6**
F6	Mean	3.79E-2	3.95E-2	**3.48E-2**
	Variance	**2.82E-5**	7.91E-5	3.82E-5
F7	Mean	4.08E-2	5.42E-2	**2.73E-2**
	Variance	5.24E-5	1.83E-4	**3.54E-5**

To show the run time performance, Figure 6 plots the average DIGD versus the time. It is not hard to come with the conclusion that the mean DIGD and variance of DEE-PDMS are less than the other two strategies, especially when $0<t<20$, the metric values of DEE-PDMS are greatly better than the other two strategies which indicates DEE-PDMS is able to respond to environmental changes more quickly and accurately. Despite since the environmental cycle changes and the accumulation of experience, the convergence and diversity of PPS will stabilize in the latter stages due to the introduction of memory strategy and will be slightly better than DEE-PDMS, overall, the difference is small, and DEE-PDMS is better than FPS. For problem whose PS is fixed, the diversity of the population will hinder the rapid convergence of population. So on DMOP1 whose PS is fixed, the stored portion of the current population in DEE-PDMS will improve the convergence of the algorithm. As to the ability to solve nonlinear problem F7, the advantage of DEE-PDMS is more obvious. It indicates that DEE-PDMS is suitable for solving complicated nonlinear problems.

There might be some reasons to explain the results, mainly because the environmental regulation of DEE-PDMS partitions the current population into three subpopulations according to the different behavioral characteristics of each individual when a change is detected. The three subpopulations will be used to deal with possible different degrees of environmental changes. Meanwhile, with the environmental facilitating and guiding, the three subpopulations will be guided to evolve toward their desired environments according to the new environmental knowledge and regulation. Therefore, DEE-PDMS is able to respond more quickly to environmental changes.

VIII. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a population diversity maintaining strategy based on dynamic environment evolutionary model (DEE-PDMS) to enhance the performance of multiobjective optimization evolutionary algorithms in dealing with dynamic environments. In the proposed strategy, we build a dynamic environment evolutionary model, which enhances population diversity by guided fashion, thus the optimal solution set can be obtained with good convergence and diversity at the initial stages. Compared with other two strategies, DEE-PDMS 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. Our future work will be the adaptive adjustment of subpopulation size. Furthermore, our focus in the future will also be designing a more accurate dynamic environment evolutionary model.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China (No. 61379062), the Graduate Innovation Foundation of Hunan Province of China (No. CX2013A011) and the Science and Technology Project of Hunan Province (NO. 2014GK3027).

REFERENCES

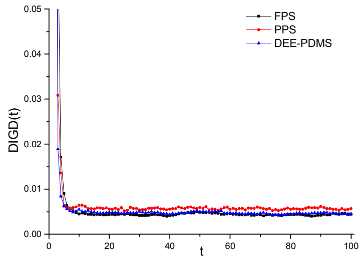
[1] M. Farina, K. Deb, and P. Amato, “Dynamic multiobjective optimization problems: test cases, approximations, and applications,” IEEE Transactions on Evolutionary Computation, 2004.8(5):425-442.

[2] T.T. Nguyen, S.X. Yang and J. Branke, “Evolutionary dynamic optimization: A survey of the state of the art,” Swarm and Evolutionary Computation, 2012.6:1-24.

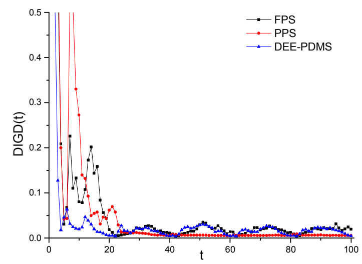
[3] C.A. Coello Coello, D.A. van Veldhuizen, and G.B. Lamont, “Evolutionary algorithms for solving multiobjective problems,” New York: Springer-Verlag, 2007.

[4] S.X. Yang, M.Q. Li, X. Liu, and J.H. Zheng, “A grid-based evolutionary algorithm for many-objective optimization,” IEEE Transactions on Evolutionary Computation, 2013.17:721–736.

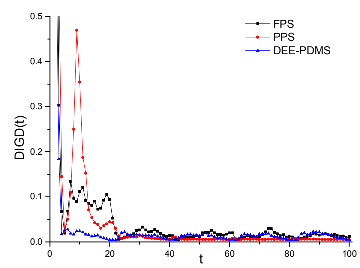
[5] Z. Zhang, “Multiobjective optimization immune algorithm in dynamic environments and its application to greenhouse control,” Applied Soft Computing, vol. 8, no. 2, pp. 959–971, 2008.



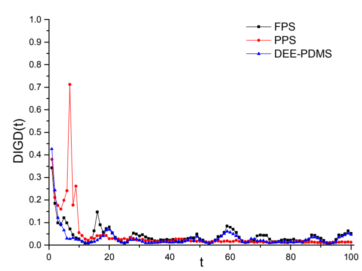
(a) DMOP1



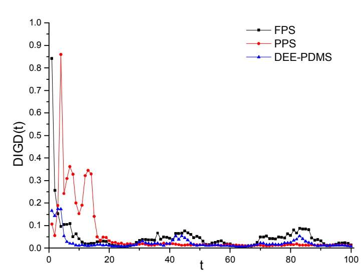
(b) DMOP2



(c) DMOP3



(d) F6



(e) F7

Fig. 6. The average DIGD over 20 runs versus time for FPS, PPS and DEE-PDMS on (a) DMOP1, (b) DMOP2, (c) DMOP3, (d) F6 and (e) F7.

[6] K. Deb, U. V. Rao, and S. Karthik, “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), 2007, vol. LNCS 4403, pp. 803–817.

[7] C.R.B. Azevedo and A.F.R. Araujo, “Generalized immigration schemes for dynamic evolutionary multiobjective optimization,” in IEEE Congress on Evolutionary Computation (CEC 2011).2011. 2033–2040.

[8] J. Wei and Y. Wang, “Hyper rectangle search based particle swarm algorithm for dynamic constrained multiobjective optimization problems,” in IEEE Congress on Evolutionary Computation (CEC 2012).2012.259-266.

[9] R. Liu, W. Zhang, L. Jiao, F. Liu and J. Ma, “A sphere-dominance based preference immune-inspired algorithm for dynamic multiobjective optimization,” in Genetic and Evolutionary Computation Conference (GECCO 2010).2010.423-430.

[10] M. Greeff and A. P. Engelbrecht, “Solving dynamic multiobjective problems with vector evaluated particle swarm optimization,” in IEEE Congress on Evolutionary Computation (CEC 2008).2008.2922-2929.

[11] Y. Wang and B. Li, “Investigation of memory-based multiobjective optimization evolutionary algorithm in dynamic environment,” in IEEE Congress on Evolutionary Computation (CEC 2009).2009. 630–637.

[12] C.K. Goh and K.C. Tan, “A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization,” IEEE Transactions on Evolutionary Computation, 2009.13:103–127.

[13] I. Hatzakis, and D. Wallace, “Dynamic multiobjective optimization with evolutionary algorithms: A forward-looking approach,” in Genetic and Evolutionary Computation Conference (GECCO 2006).2006.1201-1208.

[14] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff and E. Tsang, “Prediction-based population re-initialization for evolutionary dynamic multiobjective optimization,” in Evolutionary Multi-Criterion Optimization (EMO 2007).2007.832-846.

[15] A. Zhou, Y. Jin, Q. Zhang, “A population prediction strategy for evolutionary dynamic multiobjective optimization,” IEEE Transactions on Cybernetics, 2013.

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

[17] M. Helbig and A.P. Engelbrecht, “Archive management for dynamic multiobjective optimisation problems using vector evaluated particle swarm optimization,” in IEEE Congress on Evolutionary Computation (CEC 2011), 2011, pp. 2047–2054.

[18] W.T. Koo, C.K. Goh and K.C. Tan, “A predictive gradient strategy for multiobjective evolutionary algorithms in a fast changing environment,” Memetic Computing, vol. 2, pp. 87–110, 2009.

[19] X. Yao, Y. Liu, G. Lin, “Evolutionary programming made faster,” IEEE Transactions on Evolutionary Computation, 3(2):82–102, 1999.

[20] T.M. Blackwell and P.J. Bentley, “Dynamic search with charged swarms,” in Genetic and Evolutionary Computation Conference (GECCO 2002).2002.19-26.

[21] P. Borges, and M. Hansen, “A basis for future successes in multiobjective combinatorial optimization,” Technical Report IMM-REP-1998-8, Institute of Mathematical Modelling, Technical University of Denmark, March 1998.

[22] E. Tantar, A.A. Tantar and P. Bouvry, “On dynamic multiobjective optimization classification and performance measures,” in IEEE Congress on Evolutionary Computation (CEC 2011). 2011.2759–2766.

[23] Q. Zhang, A. Zhou and Y. Jin, “RM-MEDA: A regularity model based multiobjective estimation of distribution algorithm,” IEEE Transactions on Evolutionary Computation, 2008.12(1):41–63.

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