

## Research Article

# Dynamic Multiobjective Optimization with Multiple Response Strategies Based on Linear Environment Detection

Qiyuan Yu, Shen Zhong, Zun Liu, Qiuzhen Lin, and Peizhi Huang 

College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China

Correspondence should be addressed to Peizhi Huang; hpz@szu.edu.cn

Received 3 March 2020; Revised 16 September 2020; Accepted 20 October 2020; Published 24 November 2020

Academic Editor: Qingling Wang

Copyright © 2020 Qiyuan Yu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Dynamic multiobjective optimization problems (DMOPs) bring more challenges for multiobjective evolutionary algorithm (MOEA) due to its time-varying characteristic. To handle this kind of DMOPs, this paper presents a dynamic MOEA with multiple response strategies based on linear environment detection, called DMOEA-LEM. In this approach, different types of environmental changes are estimated and then the corresponding response strategies are activated to generate an efficient initial population for the new environment. DMOEA-LEM not only detects whether the environmental changes but also estimates the types of linear changes so that different prediction models can be selected to initialize the population when the environmental changes. To study the performance of DMOEA-LEM, a large number of test DMOPs are adopted and the experiments validate the advantages of our algorithm when compared to three state-of-the-art dynamic MOEAs.

## 1. Introduction

In the evolutionary computation community, the studies of static multiobjective optimization problems (MOPs) have been widely conducted during the recent decades, and there are a number of effective and efficient evolutionary algorithms for tackling static MOPs. However, in some practical engineering applications, it is found that some optimization problems are very complicated and need to be solved in a dynamic or uncertain environment, as their objective functions may change with the environment, which often exist in planning and scheduling problems [1–4], parameter optimization [5, 6], resource allocation [7, 8], and control system [9–11]. This kind of MOPs is often called dynamic multiobjective optimization problems (DMOPs), which can be defined in different aspects according to the nature of dynamics [12–14]. The solving of DMOPs not only needs to quickly search the Pareto-optimal set (PS) with good convergence and diversity when the environment is stable but also has to efficiently obtain some promising solutions in new environment. Uncertainty and time-varying environment bring more challenges to DMOPs.

In recent years, evolutionary algorithms (EAs) have become the most popular and effective tool for solving DMOPs, which have attracted more and more attention [15]. In most DMOEAs [14, 16–21], they often add an environmental detection mechanism and its corresponding response mechanism to the framework of static multiobjective EAs (MOEAs). The environmental detection mechanism is used to determine whether the environment has changed, while the response mechanism aims to provide a new evolutionary direction for initial population. To visually show the solving of DMOPs, a general framework of DMOEA is plotted in Figure 1. When solving DMOPs in each new environment, evolution resource is very limited to get converged from an initial state, so an initial population close to the true Pareto-optimal front (PF) of the new environment is useful to speed up the convergence. There are two challenges that often encounter when dealing with DMOPs [22]: one is to handle the conflicts in multiple objectives, while the other is to track their dynamism caused by the time-varied objective functions and constraints. During the recent decades, different approaches have been proposed to solve DMOPs, such as the co-evolutionary approaches [23–25], the decomposition-based approaches

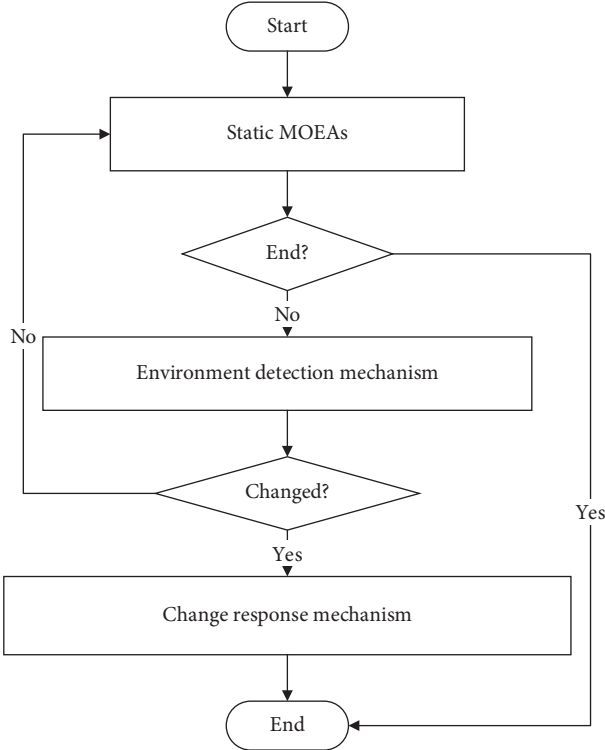


FIGURE 1: The overall framework of DMOEA.

[16], the prediction-based approaches [14, 17, 18], and some other self-learning mechanisms such as artificial neural network [26, 27].

Regarding the current DMOEAs [19, 21, 28, 29], most of change detection strategies use the method of re-evaluation to simply judge whether the environment has changes. Then, the designed response mechanism is adopted to initialize the population for new environment. If the prior knowledge is unavailable, the changing intensity and the changing rules cannot be obtained, which are likely to affect the following prediction results. For example, if the two continuous environments have a certain linear correlation, some linear response models will have a good effect. On the contrary, if they are linearly independent, the above models will be invalid. In other words, the performance of DMOEAs cannot be fully exploited when the response model is not consistent with the nature of the environmental change, which is usually not considered in these DMOEAs. In addition, many kinds of existing DMOEAs have some limitations. For example, DMOEAs that focus on diversity may cause difficulties to be converged in new environment. As pointed out in [19], proper diversity is helpful for exploring promising search regions, but too much diversity to be added may cause evolutionary stagnation. Some DMOEAs that focus on convergence perform well at the dynamic environment with high frequency change and periodic change. However, when previous environments do not appear any more, these algorithms might be inferior. Moreover, most DMOEAs with fast convergence speed often show the weak performance in maintaining diversity, which leads to falling into local optimization.

Based on the above considerations, we suggest a dynamic multiobjective evolutionary algorithm with multiple response strategies based on linear environment detection, called DMOEA-LEM. In the detection mechanism of DMOEA-LEM, if the environmental change is detected, the algorithm provides feedback about the types of the environmental change according to the flow of the population before and after the change, i.e., the different distributions of the population are exploited to estimate the changes from the old environment to the new environment as three types. This feedback reflects the intensity of environmental change to some extent. Then, in the response mechanism, there are three prediction models involving the convergence and diversity of the population, and different dynamics are considered, in which one of them is activated accordingly to generate initial population. That is to say, each type of environmental change obtained by the change detection mechanism is associated with a prediction model to initialize population in new environment. In the whole process of evolution, the optimal populations found in historical environments will be preserved to predict an initial population for the new environment. After comparing with three state-of-the-art DMOEAs on a large number of test DMOPs, the experiments show the superiority and reliability of the proposed DMOEA-LEM.

The rest of this paper is organized as follows. Section 2 describes the related works of DMOEAs and our motivations to propose DMOEA-LEM. Section 3 presents the details of DMOEA-LEM. Section 4 provides and discusses the experimental results of DMOEA-LEM with other DMOEAs. At last, our conclusions and future work are outlined in Section 5.

## 2. A Short Review of Related Work

In this section, the definition of DMOPs is first given in Section 2.1. Then, in Section 2.2, some related works on DMOEAs are summarized, and the motivations to design the proposed DMOEA-LEM are provided in Section 2.3.

*2.1. Dynamic Multiobjective Optimization Problems.* In this paper, the unconstrained DMOP is mathematically defined as follows:

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

where  $F(\mathbf{x}, t)$  is a set of  $m$  objective functions at time  $t$ , and  $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$  is a  $n$ -dimensional vector in the decision space  $\Omega_x \in R^n$ . Here,  $t$  is the time index in the time space  $\Omega_t$ , which is defined by

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

where  $\tau$  and  $\tau_t$  represent the generation counter and the frequency of change, respectively, and  $n_t$  is the change severity in a problem at time  $t$ . A smaller value of  $\tau_t$  means

more frequent changes, while a smaller value of  $n_t$  indicates a larger vibration in each change.

*Definition 1.* At time  $t$ ,  $x_1$  is said to dominate  $x_2$ , if and only if  $\forall i \in \{1, 2, \dots, m\}$ ,  $f_i(x_1, t) \leq f_i(x_2, t)$ , and  $\exists j \in \{1, 2, \dots, m\}$ ,  $f_j(x_1, t) < f_j(x_2, t)$ , where  $f_i(x, t)$  is  $i$ th objective value at time  $t$ .

*Definition 2.* At time  $t$ ,  $x^* \in \Omega_x$  is said to be Pareto optimal, if no other  $x \in \Omega_x$  can dominate  $x^*$ .

*Definition 3.* At time  $t$ , the set of Pareto-optimal solutions is termed the  $t$ th PS (PSt), and the corresponding PF is called the  $t$ th PF (PFt).

*2.2. Related Works.* At present, the research studies of DMOEAs mainly concentrate on designing the effective change detection mechanism and the corresponding response mechanism. For the change detection mechanism, the existing DMOEAs usually use two methods to judge whether the environment has changed: one is to re-evaluate the objective values of some individuals once the number of generation is increased and then determine whether the environment has changed by comparing the objective values before and after the generation counter is updated. This idea is simple and effective, which is adopted by many DMOEAs, such as [28, 29]. The other detection method is designed based on the performance and behaviors of the algorithm in the process of evolution, such as [30, 31]. The former is an easy-to-use method, and it is also used in our detection mechanism. Another emphasis on DMOEAs is to design the change response mechanism, which aims to make population track the environmental change and hopes to have a better fitness at the start of new environment. If the response mechanism can provide a good initial population that is very close to the true PF of new environment, it will significantly speed up the convergence of DMOEAs. According to the evolutionary behaviors, DMOEAs can be roughly divided into the following three categories.

- (1) Diversity-based DMOEAs: the diversity-based DMOEAs focus on maintaining the diversity of the evolutionary population. In order to quickly track the new environment, it is encouraged to mutate an already converged population in order to jump out of the current optimal state, which will increase the population diversity. This is usually implemented in two ways. The first is to respond to dynamics after a change in the environment is detected, which is called diversity introduction [20]. For example, the population is divided into two parts in [32], respectively, for running random initialization and Gaussian variation after the environmental change, and some new solutions are injected into the population to participate in the competition for survival. In [33, 34], some hypermutation operators are used to disturb some solutions with good convergence abilities in the old environment, which are expected to jump out of their original positions. The second

approach is called as diversity maintenance [20]. In this approach, solutions are only required to show a relatively scattered state in the search space so that the population can quickly respond to changes in the new environment. In [35], a random immigration mechanism and an intrinsic diversity maintenance strategy are designed to adapt the population to the dynamic changes. In [36], a threshold method is used to enhance the population diversity, and this strategy of diversity maintenance can efficiently work without any change detection mechanism. Moreover, other diversity maintenance methods are reported in [16, 37–40].

- (2) Convergence-based DMOEAs: the major purpose of this method is to speed up the convergence by using the historical information. In particular, this approach has an obvious advantage when the new PS is somewhat similar to the previous PS, or the environment changes with regular patterns. In most related algorithms, the optimal solution sets of historical environments are all maintained in the process of evolution, which will be reused in new environment. This kind of techniques is known as memory mechanism [16, 20], which is very effective and remarkable when handling DMOPs with periodic changes. For example, a memory-based DMOEA [41] is reported with three models for initialization, exploration, and archiving, respectively. In [42], a hybrid memory method is run based on the archive, which reinitializes population by re-evaluating the previous nondominated solutions or by conducting a local search for the elite solutions preserved in the archive. In addition, an immune cloning algorithm with memory strategy is extended to dynamic environment in [43] and the historical knowledge is used to build up a dynamic evolutionary model in [44]. More research studies along this direction can refer to [45–48].
- (3) Prediction strategy-based DMOEAs: this kind of approaches obtains the movement rules of the changing PS/PF by analyzing the historical records, which tries to estimate the future PSs/PFs for new environment. Prediction techniques usually generate an initial population for the optimizer, which is expected to approach or evenly cover the real PS in new environment [22]. Since historical populations are also used, the prediction-based method is usually coupled with the memory mechanism [21]. If the position of the new PS can be predicted, the tracking speed will be greatly accelerated and the tracking accuracy will be significantly improved. For example, a forward prediction strategy is proposed based on the autoregressive model [49] and other prediction models are developed by regularity property [18, 50]. They track the movement of the center of the PS manifold in different dynamic environments, thus generating a new population within the neighborhood around the expected location of the new PS. A

gradient-based prediction operator is suggested in [51] to predict the optimal moving direction of the population. More recent studies of prediction strategies for DMOEAs can be found in [17, 19, 52, 53], which have become more and more popular due to its high accuracy and flexibility.

### 3. The Proposed DMOEA-LEM Algorithm

In this section, the details of DMOEA-LEM are introduced. The framework of DMOEA-LEM is provided in Section 3.1. Then, the change detection mechanism with the function of feedback change type and the response mechanism with multiple prediction models are, respectively, introduced in Sections 3.2 and 3.3.

*3.1. The Framework of DMOEA-LEM.* The pseudocode of the proposed DMOEA-LEM is provided in Algorithm 1, which is based on the traditional framework of DMOEA, which will detect the environmental change and make a proper response during the evolutionary procedure. Different from other DMOEAs [41, 42, 49, 51] based on historical information prediction, our algorithm considers the influence of the degree of environmental change and accordingly uses a suitable environment detection mechanism. According to the type of environmental change, one specific environment response is activated. In DMOEA-LEM, there are two inputs for DMOEA-LEM: the population size  $N$  and the limited maximal generation  $maxGen$ . In line 1, the time step  $t$  is set to 1 and the set POP including the final population of each environment is initialized as an empty set. Then, the parent population  $P$  is randomly initialized with  $N$  solutions in the whole decision space, and the generation counter  $gen$  is initialized as 0 in line 2. After that, one static MOEA (MOEA/D-IRA [54] is used in this paper) is applied to obtain the population  $Q^t$  for next generation in line 3. Thereafter, DMOEA-LEM enters the main loop and iteratively runs the procedures of lines 4–15 until the termination condition is satisfied, i.e.,  $gen = maxGen$ . At each generation, it is required to clear the set  $F^Q$  in line 5, which will be used to record the objective values. The objective values of each solution in population  $Q^t$  are calculated and stored in  $F^Q$  in line 6. Then, the generation counter  $gen$  is added by 1 in line 7. Obviously, both the optimization problems and the mapping from the decision space to objective space may change with the increase of generations. Accordingly, the environment detection strategy is adopted to detect whether the environment has changed or not in line 8, and then two outputs can be obtained, i.e., the label  $flag$  which is used to indicate whether the environment is changed and the correlation coefficient  $corr$ . The details of this method are further described in Section 3.2. After that, different response strategies are activated to generate the next population according to the obtained  $flag$ , if the environment has been changed, i.e.,  $flag = true$ . Then, the population set POP is updated by adding the current population  $Q^t$  in line 10 and the time step  $t$  is added by 1 in line 11. The

initial population  $Q^t$  of the new environment is acquired by the suggested response strategy in line 12. On the contrary, as shown in line 13, the static algorithm MOEA/D-IRA [54] is used to continually evolve the population if the environment is consistent before and after the generation counter is changed. Note that the details of multiple response strategies are introduced in Section 3.3. At last, when  $gen$  reaches  $maxGen$ , the final population set POP is returned, which stores the optimal solution set in each environment in line 16.

*3.2. Environment Detection Strategy.* In this section, the environment detection strategy is introduced. For most DMOEAs [16–19, 23–27], they usually do not take into account the type and extent of environmental change, and there are only few studies [22] on which kinds of changes happen in the environment and their effects on the predicted population. In our change detection mechanism, if the environment has changed, the types of change are estimated simultaneously. Since the solutions obtained by NSGA-II [55] can embody the convergence and diversity of the current population to a certain extent, a set  $R$  including  $N/10$  solutions obtained by NSGA-II plays an important role in our environment detection strategy. The solutions in  $R$  are re-evaluated by calculating their objective values so that the environment changed or not will be determined according to the similarities and differences of the two evaluation results. Moreover, the set  $R$  is further employed to calculate the linear correlation between the two environments when the environment has changed. Then, the type of change is estimated by the linear correlation coefficient  $corr$ .

The pseudocode of our environment detection strategy is provided in Algorithm 2, which includes three inputs: the objective vector set  $F^Q$  of population  $Q^t$  before the generation counter is increased, the current population  $Q^t$ , and the population size  $N$ . In line 1, the label  $flag$  is initialized to **false**, which indicates that the environment has not changed, and  $F^R$  is set to an empty set. Note that  $F^R$  records the objective values of  $Q^t$  after the generation counter is increased. To effectively detect the degree of correlation before and after the environmental change, our detection mechanism adopts the fast nondominated sorting and the crowding distance [55] to select  $N/10$  solutions in line 2, and these solutions constitute a solution set  $R$ . After that, the obtained solution set  $R$  is used to detect whether the environment has changed in lines 3–10. Here, for each solution  $R_i \in R$ , where  $i = 1, 2, \dots, N/10$ , its objective values for the current environment are calculated and the objective vector is recorded as  $F_i^R$  in line 4. Then, find the original objective vector  $F_i^Q$  of the corresponding solution from  $F^Q$  in line 5 and then compare  $F_i^R$  and  $F_i^Q$  in line 6. If  $F_i^R$  is different from  $F_i^Q$ , the  $flag$  is marked as **true** in line 7, which means that the environment has changed with the increase of generation. Particularly, if the environment changes, equation (3) is applied for estimating the degree of linear correlation coefficient  $corr$  of the two adjacent environments in line 12 as follows:

$$\text{corr}(t, t+1) = \left| \frac{\text{cov}(\mathbf{p}^t, \mathbf{p}^{t+1})}{\text{std}(\mathbf{p}^t) \times \text{std}(\mathbf{p}^{t+1})} \right|, \quad (3)$$

$$\text{cov}(\mathbf{p}^t, \mathbf{p}^{t+1}) = E((\mathbf{p}^t - E(\mathbf{p}^t)) \cdot (\mathbf{p}^{t+1} - E(\mathbf{p}^{t+1}))), \quad (4)$$

where equation (4) is used to calculate the covariance of  $\mathbf{p}^t$  and  $\mathbf{p}^{t+1}$ , and  $\mathbf{p}$  is an one-dimensional array. Equation (5) and equation (6), respectively, denote the average and the standard deviation of  $\mathbf{p}$ :

$$E(\mathbf{p}) = \frac{1}{|\mathbf{p}|} \sum_{i=1}^{|\mathbf{p}|} \mathbf{p}_i, \quad (5)$$

$$\text{std}(\mathbf{p}) = \left( \frac{1}{|\mathbf{p}|} \sum_{i=1}^{|\mathbf{p}|} (\mathbf{p}_i - E(\mathbf{p}))^2 \right)^{(1/2)}. \quad (6)$$

In this paper, the interval in which the *corr* is located determines the type of the change and three types of environmental changes, i.e., linear independence, strong linear dependence, and weak linear dependence, are used in this paper, using the following principles:

Type I: the change is linearly independent if  $\text{corr} \in [0.0, 0.2)$

Type II: the change is strong linearly dependent if  $\text{corr} \in [0.7, 1.0]$

Type III: the change is weak linearly dependent if  $\text{corr} \in [0.2, 0.7)$

It is observed that the higher *corr* value indicates the stronger linear correlation, and this indicator is used for controlling the use of multiple response strategies. At last, the label *flag* and the correlation coefficient *corr* are outputted in line 14.

In order to facilitate the understanding of the process of calculating the linear correlation coefficient *corr*, a simple example is given in Figure 2. Here, there are eight solutions in  $R$ , and the environments before and after the change are recorded at time  $t$  and time  $t+1$ , respectively. As shown in Figure 2(a), at time  $t$ , the eight white points indicate the objective values of solutions in 2D objective space at current moment. For each objective, the solutions with the minimum objective values are selected as the boundary points, i.e., the red points  $x_1$  and  $x_2$ , and then a line  $H$  is constructed to go through these two points, which is formalized as  $k_1 f_1 + k_2 f_2 + p = 0$ . After that, this line is translated (i.e., fix the two coefficients  $k_1$  and  $k_2$ ) so that it passes through each solution in  $R$ . For each solution,  $x_i \in R$  corresponds to a line  $H_i^t$  parallel to  $H$  and has an intercept  $p_i^t \in \mathbf{p}^t$ , where  $i = 1, 2, \dots, N/10$  and  $\mathbf{p}^t$  is the set of intercepts in time  $t$ . In Figure 2(a),  $H$  and  $H_a^t$  are represented by black solid lines and blue dashed lines, respectively, where  $a$  is a point at time  $t$ . Suppose that the environment has changed at time  $t+1$ , and the objective values of solutions may be different from that at time  $t$ . Thus, each solution corresponds to a new point in the objective space at time  $t+1$ , i.e., the black points in Figure 2. Just like at time  $t$ , the same line  $H$  is translated and goes through all the black points, and then each solution

$x_i \in R$  corresponds to a new line  $H_i^{t+1}$  and an intercept  $p_i^{t+1} \in \mathbf{p}^{t+1}$  at time  $t+1$ . In Figure 2(a), for the point  $b$  at time  $t+1$ , the corresponding  $H_b^{t+1}$  is marked by the red dashed line. Note that the relationships between the intercepts  $p/p_i^t/p_i^{t+1}$  and the coefficients  $k_1/k_2$  are reflected in the intersection of the corresponding line and each objective axis, such as the point  $(0, -p/k_2)$  and  $(-p/k_1, 0)$ . Similarly, for the 3D objective space plotted in Figure 2(b), the calculation of *corr* is similar to that in the 2D objective space, in which  $H$  is a hyperplane.

**3.3. Multiple Response Strategies.** In general, DMOEAs reinitialize the population when the environment has changed. A good initial population can effectively promote MOEAs to find some superior solutions in the new environment. However, how to provide a good initial population after the change of environment is a challenging problem for solving DMOPs. Thus, multiple response strategies are suggested according to the types of changes which are detected by the above environmental change detection in Section 3.2, which belongs to the category of prediction-based DMOEAs. In this paper, each type of environmental change will activate a distinct initialization scheme, in which three models are built for generating new solutions under different situations.

After the environmental change, the position of each individual in the objective space usually changes. For each solution of current population, the moving vector  $\mathbf{v}$  of a solution is formed by its former and latter two positions in the objective space, which shows the direction and step length of a solution in the objective space. In our multiple response strategies, the vector  $\mathbf{v}$  is a key element that not only implicitly guides the model to be used but also roughly determines the position of new population.

**3.3.1. Model A.** If the estimated type of the change is linear independent (i.e., Type I), the initial population  $Q^t$  for new environment is generated by initializing the nondominated individuals randomly as follows:

$$Q^t = Q_{\text{nd}}^T \cup X', \quad (7)$$

where  $Q_{\text{nd}}^T$  is the nondominated solution set of the last population  $POP^T$  in  $POP$  at time  $t$  ( $T$  is also the size of  $POP$ ) and  $X'$  is the new solution set generated by random initialization as follows:

$$X' = \text{randomInitialize}(N - |Q_{\text{nd}}^T|) + \varepsilon, \quad (8)$$

where the function `randomInitialize()` is used to randomly obtain  $N - |Q_{\text{nd}}^T|$  solutions, as shown in line 2 of Algorithm 1,  $N$  is the population size, and  $\varepsilon$  is the Gaussian noise.

In this scheme, model A is suitable for the situations where the environment changes dramatically and irregularly. The lower value of *corr* indicates that there is basically no linear dependent between the two environments before and after the change. Under this linearly independent change, the moving step lengths and directions of

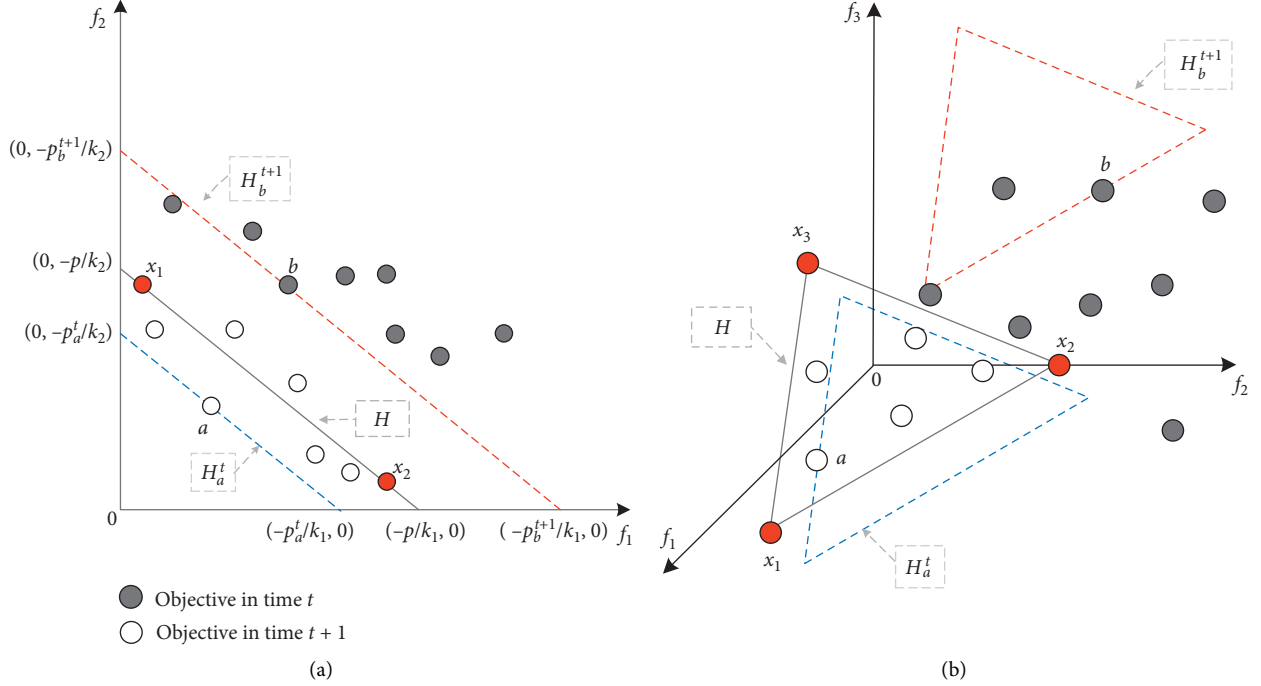


FIGURE 2: The process of calculating the linear correlation coefficient  $corr$ .

individuals in the objective space are quite different, and the obtained intercepts of new environment are scattered. Therefore, some solutions for new environment are obtained by random initialization, which is common to respond to environmental changes. Considering the good convergence of nondominated solutions, the set  $Q_{nd}^T$  is preserved as another part of initial population for new environment. In summary, model A considers both diversity and convergence after a disordered environmental change.

**3.3.2. Model B.** If the estimated type of the change is strong linearly dependent (i.e., Type II), the initial population  $Q^t$  is generated by this model B, which is a scheme based on the centroid movement of nondominated solutions as inspired by SEGA [19]. In model B, considering that the moving vectors of most solutions in current population are highly consistent, it is expected that the rules contained in historical information can effectively help to predict the new population. Here, the final populations  $POP^T$  and  $POP^{T-1}$  in the last two historical environments stored in  $POP$  are used to extract the moving vector  $\mathbf{v}^t$ . Then, the moving vector  $\mathbf{v}^t$  from  $POP^{T-1}$  to  $POP^T$  is calculated by

$$\mathbf{v}^t = \mathbf{x}_{nd}^T - \mathbf{x}_{nd}^{T-1}, \quad (9)$$

where  $\mathbf{x}_{nd}^T$  and  $\mathbf{x}_{nd}^{T-1}$ , respectively, indicate the centroids of  $POP^T$  and  $POP^{T-1}$ . Unlike the moving vector of solutions mentioned above,  $\mathbf{v}^t$  represents the moving vector of population. In this paper, the centroid  $\mathbf{x}_{nd}^T$  of population  $POP^T$  is defined as the centroid of nondominated solutions, which is given by

$$\mathbf{x}_{nd}^T = \frac{1}{|Q_{nd}^T|} \sum_{i=1}^{|Q_{nd}^T|} \mathbf{x}_i^T, \quad (10)$$

where  $\mathbf{x}_i^T \in Q_{nd}^T$  and  $Q_{nd}^T$  is the nondominated solution set of  $POP^T$ . As shown in Figure 3, in the 2D decision space, the solutions of  $POP^{T-1}$  and  $POP^T$  are expressed by the white and black points, respectively, and the two red points are their corresponding centroids. Then, for each solution in the initial population  $Q^t$ , it is derived from the movement of the solution in  $POP^T$  along the moving vector  $\mathbf{v}^t$ , and the  $i$ th solution  $\mathbf{x}_i^t$  of the population  $Q^t$  is calculated by

$$\mathbf{x}_i^t = \mathbf{x}_i^T + \mathbf{v}^t + \varepsilon, \quad (11)$$

where  $\mathbf{x}_i^T$  is the  $i$ th solution in  $POP^T$  ( $i=1, 2, \dots, N$ ). In Figure 3, the yellow points are the solutions in  $Q^t$ . In conclusion, for model B, it can be abstracted as the following mathematical expression:

$$\begin{cases} Q^t = \mathbf{x}_1^t \cup \mathbf{x}_2^t \cup \dots \cup \mathbf{x}_N^t, \\ \mathbf{x}_i^t = \mathbf{x}_i^T + \mathbf{v}^t + \varepsilon. \end{cases} \quad (12)$$

where  $i=1, 2, \dots, N$ .

This response model uses the final populations of the two adjacent environments to predict the initial population  $Q^t$  of new environment. In the case of relatively small environmental changes, there is a great deal of consistency in the moving vectors of most individuals, and the higher value of  $corr$  implicitly expresses this characteristic. Under this type of change, we regard the environments before and after the change as a strong linear dependent type. According to the principle based on the prediction method, the movement trend of historical populations can provide useful

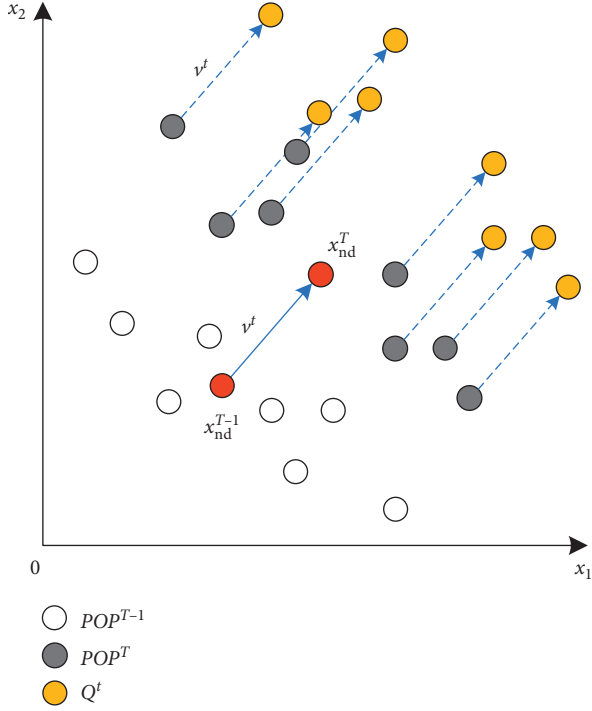


FIGURE 3: The process of generating an initial population using model (B).

information for prediction, and a simple strategy of centroid movement based on linearity is proposed. In addition, since the nondominated solutions are more consistent with the true PF at time  $t$ , the centroid is calculated by using the nondominated solution set instead of the whole population, so as to reduce a certain error. In summary, model B is used for the situation that the environmental changes are approximately linear, which has very low complexity on time and space.

**3.3.3. Model C.** If the estimated type of the change is weak linearly dependent (i.e., Type III), the population  $Q^t$  is obtained by this model C. Its main idea is to make the algorithm search randomly in a certain range, which is a compromise scheme between model A and model B. When the environmental change is detected as Type III, the  $i$ th solution of initial population  $Q^t$  is obtained by

$$x_i^t = x_i^T + r \times \mathbf{v}^t + \varepsilon, \quad (13)$$

where  $r$  is a random value in the range  $(-1, 1)$  and  $\mathbf{v}^t$  is calculated by equation (9). From equation (13), it can be observed that each solution in  $POP^T$  moves randomly within a certain interval, which is determined by the moving vector of the historical centroids. Then, model C can be summarized as follows:

$$\begin{cases} Q^t = x_1^t \cup x_2^t \cup \dots \cup x_N^t, \\ x_i^t = x_i^T + r \times \mathbf{v}^t + \varepsilon, \end{cases} \quad (14)$$

where  $i = 1, 2, \dots, N$ . Figure 4 gives an illustration about the search range of model C, where the white points are the

solutions in  $POP^T$  and the red point indicates the centroid of  $POP^T$ . In the range of  $\pm \mathbf{v}^t$ , all solutions in  $POP^T$  move randomly along the direction of  $\mathbf{v}^t / -\mathbf{v}^t$  and then the yellow points are the obtained solutions of  $Q^t$ .

Model C is used in the case of weak linear dependent type, i.e., the correlation between the two environments before and after change is not very strong but not absent. Under this type of change, the moving vectors of individuals are not as discrete as that in Type I and are not as consistent as that in Type II. Thus, the random initialization strategy and the centroid-based movement strategy are combined to randomly initialize each point in  $POP^T$  in a certain area.

The pseudocode of our multiple response strategies is provided in Algorithm 3. After getting the value of  $corr$  by the environment detection strategy in Algorithm 2, DMOEA-LEM generates an initial population  $Q^t$  for the new environment once a change is detected. In line 1, population  $Q^t$  is initialized as an empty set and  $T$  is set to the size of  $POP$ , where  $POP$  stores all historical populations. If there is only one population in  $POP$ , this means that the environment has changed only once so far. Then, model A is adopted to obtain population  $Q^t$  at current time  $t$  in line 3, and  $POP^T$  is the newest population in  $POP$ . In other words, it is the final population in the last environment. Moreover, if the estimated type of environmental change is linearly independent (i.e., line 5), model A is still used to generate the initial population  $Q^t$  in line 6. Additionally, if the estimated type of environmental change is strong linearly dependent or weak linearly dependent,  $Q^t$  is acquired by model B or model C (i.e., line 8 and line 9), where  $POP^{T-1}$  is the final population of the second last environment in  $POP$ . At last, an initial population  $Q^t$  for new environment is exported in line 12.

## 4. The Experimental Studies

### 4.1. Related Experimental Information

**4.1.1. Compared MOEAs.** In order to validate the performance of our proposed method, three competitive DMOEAs are used to testify the performance of our proposed DMOEA-LEM, which are briefly introduced below.

- (1) PPS (see [14]): considering continuous and periodic DMOPs, Zhou et al. proposed an environmental response strategy named PPS based on prediction of historical information. In this algorithm, the Pareto solution set is divided into two parts: (1) the centroid of the population and (2) manifold. In a certain sense, the centroid determines the position of the population, and the manifold indicates the contour of the population. During the evolutionary process, the sequence of centroid points of each environment is preserved, and then the centroid of next environment is predicted using an autoregression (AR) model over a number of consecutive time steps. Analogously, the similarity and variance between the historical manifolds are calculated to estimate the new manifold. As the environment changes, the initial population of the new environment will be

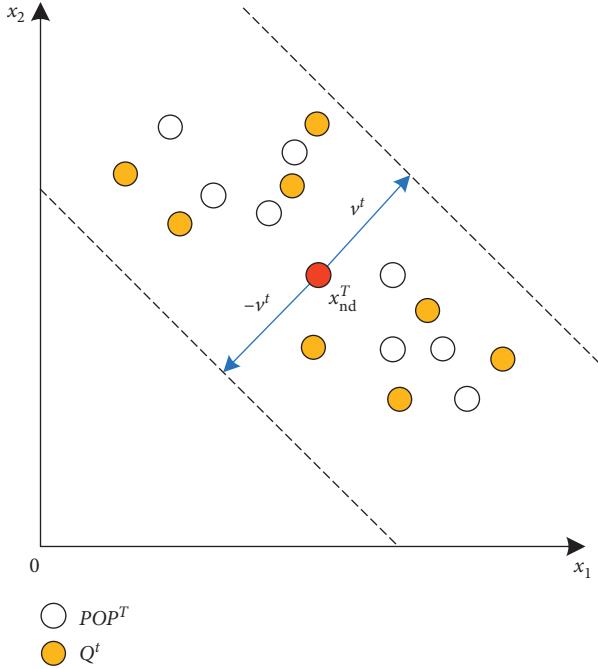


FIGURE 4: The process of generating an initial population using model (C).

created by the predicted centroid and manifold. In general, PPS is a dynamic multiobjective distribution estimation algorithm based on the regularity model, which has good performance when dealing with DMOPs with continuous changes.

- (2) MOEA/D-KF (see [20]): in MOEA/D-KF, Arrchana et al. embed the Kalman Filter model [56] into MOEA/D as the dynamic response mechanism. In the response process, half of the initial population for the new environment is randomly generated, and the other half is predicted by the Kalman Filter by tracking individual trajectories. In this algorithm, there are two filter models, i.e., 2-D KF (2by2KF) and 3-D KF (3by3KF), in which 2D KF corresponds to a second-order observation matrix, which records the displacement and velocity of the individual, while 3D KF corresponds to a third-order observation matrix, which also records the acceleration of the individual. When the change is detected, the learning trajectory is used to predict the location of the Pareto solution set. This algorithm has more advantages in solving some problems with simple linear changes on account of the setting of the Kalman Filter. However, if the environment changes sharply, the solution movements are haphazard, so the trajectories tracked by the Kalman Filter are relatively chaotic which will cause large errors.
- (3) SGEA (see [19]): this algorithm combines the fast and steadily tracking ability of steady-state algorithms and good preservation on diversity. After successfully detecting the environmental change, it answers the environmental change in a steady-state

manner. The initial population of the new environment consists of two kinds of solutions, half of which are old solutions and the others are new solutions. For the old solutions, they are derived from some well-distributed solutions in historical environments, and some basic understanding of the landscape of the new environment is provided to this algorithm by re-evaluating these old solutions. The new solutions are predicted based on the interactions with the external archive to improve the convergence speed. In general, this algorithm can quickly adapt to the changing environment and is expected to provide good tracking ability.

**4.1.2. Benchmark Problems.** In this study, four test suites containing 27 dynamic benchmark problems are used to examine the performance of the proposed approach, i.e., FDA1-FDA5 [12], dMOP1-dMOP2 [16], ZJZ1-ZJZ6 [14], and DF1-DF14 [57]. In the research of DMOEAs, both FDA and dMOP are classic and typical test suites, which are created in 2004 and 2009, respectively. In [12], M. Farina et al. presented the FDA dynamic test suite based on ZDT and DTLZ test functions, and they classified DMOPs into four types according to the changers of PS and/or PF with time. In the FDA test suite, there are two problems of type I (i.e., FDA1 and FDA4), two problems of type II (i.e., FDA3 and FDA5), and one problem of type III (i.e., FDA3). Specially, the density of solutions along the PF has time-varying property in FDA3 and FDA5, which increases the challenge of problems. For the dMOP test suite, we choose type III problem (dMOP1) and type II problem (dMOP2), and the changing PSs of dMOP1 and dMOP2 follow a sinusoidal pattern. In six ZJZ problems (F5-F10) [14], all of them have nonlinear correlation between decision variables. Besides that, the PSs of ZJZ1-ZJZ3 are 1D curves and the PSs of ZJZ4 are 2D surfaces. Most of them change the environment smoothly for each environment, and the geometric shapes of two consecutive PFs are similar to a certain degree [14, 20]. Then, in order to further verify the performance of all the compared algorithms on more difficult problems, two more complicated test instances (ZJZ5 and ZJZ6) are designed. Recently, S. Jiang et al. propose a novel DF test suite, which includes nine biobjective problems (i.e., DF1-DF9) and five three-objective problems (i.e., DF10-DF14). In particular, the DF test suite includes some functions from other test suites, e.g., FDA, dMOP, ZJZ, and JY [58]. Compared with the previous classic DMOPs, DF has more complex and difficult properties, such as disconnection, irregular PF shapes, time-dependent geometries, and variable linkages [57] so that this test suite is more consistent with the dynamic optimization problems of the real world. Most of the above test instances have periodical changes, and their definitions can be found in the corresponding original literatures.

**4.1.3. Performance Metrics.** In this paper, two metrics are adopted to investigate the performance of convergence and diversity, i.e., mean inverted generational distance (MIGD)



[59] and mean hypervolume (MHV) [13]. In the study of static MOEAs, the inverted generational distance (IGD) [60] and the hypervolume (HV) [61] are widely applied to evaluate the optimization performance, which are slightly modified for evaluating DMOEAs.

- (1) Mean inverted generational distance (MIGD): IGD is a main performance indicator which provides a quantitative measurement for convergence and diversity. The IGD is calculated by

$$\text{IGD}(P_t^*, P_t) = \frac{1}{|P_t^*|} \sum_{x \in P_t^*} \text{dis}(x, P_t), \quad (15)$$

where  $P_t^*$  is a set of points which are uniformly sampled on the true PF at time  $t$ ,  $P_t$  is the population obtained by MOEAs, and  $\text{dis}(x, P_t)$  is the minimum Euclidean distance between a point  $x \in P_t^*$  and the points in  $P_t$ . Note that the calculation of IGD requires the prior knowledge of true PF. In this paper, we use the method suggested in [62] to sample 10,000 and 22,500 uniformly distributed points on the true PF at each corresponding time for biobjective and three-objective problems, respectively, so as to form  $P_t^*$ . Then, the MIGD metric is defined as the average of all the IGD values in a time step  $T$  over a run, which is given by

$$\text{MIGD} = \frac{1}{T} \sum_{t \in T} \text{IGD}(P_t^*, P_t). \quad (16)$$

In this paper,  $T$  is set to (0, 120) and the smaller MIGD value indicates the better convergence and diversity.

- (2) Mean hypervolume (MHV): in the static MOEAs, HV is also a popular performance metric. For the computation of HV, the reference point  $z^t = (z_1^t + 0.5, z_2^t + 0.5, \dots, z_m^t + 0.5)$  is required, where  $z_m^t$  is the maximum value of the  $j$ th objective in the objective space at time  $t$ , and  $z^t$  is dominated by all Pareto-optimal solutions at the corresponding time. The HV metric is defined as

$$\text{HV}(P^t) = \text{VOL}\left(\bigcup_{x \in P^t} [f_1(x), z_1^t] \times \dots \times [f_m(x), z_m^t]\right), \quad (17)$$

where VOL is the Lebesgue measure. In particular, the solutions dominated by  $z^t$  are discarded for HV calculation. Then, the MHV metric is calculated by

$$\text{MHV} = \frac{1}{T} \sum_{t \in T} \text{HV}(P^t). \quad (18)$$

**4.1.4. Experimental Parameter Settings.** In the experiments, all the considered DMOEAs are independently run 20 times on each test problem and their parameters are set as sug-

gested from their original studies. Some key parameters in these algorithms are set as follows:

- (1) Population size and number of variables: the population size  $N$  is set to 100 and 300 for biobjective and three-objective problems, respectively; the number of variables  $d$  is set to 20 for FDA, dMOP, and ZJZ problems, while  $d$  is set to 10 for the CEC test problem.
- (2) Reproduction operators: the crossover probability is  $\text{CR} = 1.0$  and the scaling factor is  $F = 0.5$  in the DE operator [63]. The distribution index is  $\eta_m = 20$  and the mutation probability is  $\text{pm} = 1/d$  in the polynomial mutation (PM) operator [64], where  $d$  is the number of variables. The crossover probability is  $\text{pc} = 1.0$  and its distributions index is  $\eta_c = 20$  in simulation binary crossover (SBX) [65].
- (3) Stopping criterion, the change frequency and severity: all the algorithms stop when there are 3600 iterations, i.e.,  $\max\_t = 3600$ . The change frequency is  $\tau_T = 30$  and the change severity is  $n_T = 10$ ; therefore, there are 120 environmental changes in the evolutionary process.
- (4) Other parameters: for the PPS, the AR model order is  $p = 3$ , and the length of history mean point series is  $M = 23$ . In SGEA, the archive size is set the same as the population size  $N$ .

**4.2. Comparison with Several Competitive DMOEAs.** In this paper, the MIGD and MHV metrics are used as the indicators to evaluate the performance of considered DMOEAs, and the comparison results of DMOEA-LEM with other three algorithms using MIGD and MHV are summarized in Tables 1 and 2, respectively. Here, the MIGD (or MHV) and the mean of standard deviations (in parenthesis) of 20 runs are provided, and for each test problem, the best results, i.e., the smallest MIGD or the largest MHV among these comparison algorithms, are highlighted in bold with a gray background. Furthermore, the Wilcoxon rank-sum test [66] with 0.05 significant level is carried out to verify the statistically significant differences between the results of DMOEA-LEM and other three compared DMOEAs. The symbols “+”, “ $\approx$ ” and “-” are, respectively, used to indicate that the metric values of corresponding algorithm are significantly better than, similar to and worse than that of DMOEA-LEM.

As shown in Table 1, the MIGD values of four algorithms are listed. It can be seen that DMOEA-LEM performs better than its three competitors on 27 test problems. As observed from the second last row of Table 1, DMOEA-LEM performs best in 10 out of 27 problems, while PPS, MOEA/D-KF, and SGEA are best in 5, 6, and 6 problems, respectively. In the last row of Table 1, DMOEA-LEM outperforms PPS, Kalman, and SGEA in 18, 15, and 17 problems, respectively. Considering the FDA test problems, for the 2-objective FDA1 and FDA3 with regular convex PFs, the performance of DMOEA-LEM is worse than that of MOEA/D-KF on

```

Input:  $N, maxGen$ 
Output: the stored solution sets in each environment:  $POP = \{Q^1, Q^2, \dots\}$ 
(1)  $t = 1, POP = \emptyset$ 
(2) initialize a population  $P$  and set  $gen = 0$ 
(3)  $Q^t = \text{MOEA/D-IRA}(P)$ 
(4) while not terminate do
(5)    $F^Q = \emptyset$ 
(6)   calculate the objective values for each solution in  $Q^t$  (stored in  $F^Q$ )
(7)    $gen = gen + 1$ 
(8)    $[flag, corr] = \text{ChangeDetection}(F^Q, Q^t, N)$  //Algorithm 2
(9)   if  $flag == \text{true}$  //is Changed
(10)      $POP = POP \cup \{Q^t\}$ 
(11)      $t = t + 1$ 
(12)      $Q^t = \text{Response}(corr, POP)$  //Algorithm 3
(13)   else  $Q^t = \text{MOEA/D-IRA}(Q^t)$ 
(14)   end if
(15) end while
(16) return  $POP$ 

```

ALGORITHM 1: General framework of DMOEA-LEM

```

(1)  $flag = \text{false}, F^R = \emptyset$ 
(2) select  $N/10$  solutions from  $Q^t$  by fast nondominated sorting and crowding distance (stored in  $R$ )
(3) for  $i = 1$  to  $N/10$ 
(4)   calculate the objective values of  $i$ th solution in  $R$  (stored as  $F_i^R$ )
(5)   find the original objective values of  $i$ th solution in  $R$  (stored as  $F_i^Q$ )
(6)   if  $F_i^R \neq F_i^Q$ 
(7)      $flag = \text{true}$ 
(8)     break
(9)   end if
(10) end for
(11) if  $flag == \text{true}$ 
(12)   calculate the  $corr$  of  $F_i^R$  and  $F_i^Q$  using equation (3)
(13) end if
(14) return  $[flag, corr]$ 

```

ALGORITHM 2: ChangeDetection ( $F^Q, Q^t, N$ ).

```

(1)  $Q^t = \emptyset, T = |POP|$ 
(2) if  $T == 1$ 
(3)   the initial population  $Q^t$  is generated by model A consisting of  $POP^T$  //equation (7)
(4) else
(5)   if  $corr < 0.2$ 
(6)     the initial population  $Q^t$  is generated by model A consisting of  $POP^T$ 
(7)   else if  $corr \geq 0.7$ 
(8)      $Q^t$  is generated by model B consisting of  $POP^T$  and  $POP^{T-1}$  //equation (12)
(9)   else  $Q^t$  is generated by model C consisting of  $POP^T$  and  $POP^{T-1}$  //equation (14)
(10)  end if
(11) end if
(12) return  $Q^t$ 

```

ALGORITHM 3: Response( $corr, POP$ ).

them. On 2-objective FDA2, DMOEA-LEM has the best MIGD value. For 3-objective FDA4 with a fixed and non-convex PF, the performance of DMOEA-LEM is also

promising. Regarding the 3-objective FDA5 with variable and nonconvex PFs, DMOEA-LEM performs slightly worse than SGEA. Specifically, the PF changes of FDA5 tend to the

TABLE 1: Comparison results of MSDA-LEM and three competitive DMOEAs on 27 test problems using MIGD.

Problems	T	PPS	MOEA/D-KF	SGEA	DMOEA-LEM
FDA1	0-120	1.2889E-02(1.15E-02)-	<b>8.4088E-03(7.58E-04)+</b>	9.6888E-03(1.77E-03)+	1.0115E-02(2.52E-03)
FDA2	0-120	9.3733E-03(2.05E-03)-	1.0048E-02(5.23E-03)-	1.5964E-01(3.20E-03)-	<b>6.5859E-03(4.54E-04)</b>
FDA3	0-120	1.5654E-01(6.79E-02)-	<b>2.5006E-02(4.96E-03)+</b>	7.6209E-02(6.92E-03)-	4.3860E-02(7.38E-03)
FDA4	0-120	7.0570E-02(5.22E-03)-	4.5440E-02(1.83E-03)-	4.0543E-02(1.64E-03)≈	<b>4.0311E-02(1.71E-03)</b>
FDA5	0-120	1.1568E-01(1.63E-02)-	8.8994E-02(9.92E-03)-	<b>6.3885E-02(2.35E-03)≈</b>	6.6916E-02(3.42E-03)
dMop1	0-120	3.5702E-02(3.77E-02)-	8.8277E-03(1.24E-03)+	<b>7.4419E-03(2.09E-03)+</b>	1.1232E-02(1.74E-03)
dMop2	0-120	2.4418E-02(2.20E-02)-	<b>8.8243E-03(1.31E-03)+</b>	1.1928E-02(3.03E-03)-	9.8082E-03(2.72E-03)
ZJZ1	0-120	4.9316E-02(4.64E-02)+	<b>2.9442E-02(8.88E-03)+</b>	6.4112E-01(7.49E-02)-	5.8137E-02(3.68E-02)
ZJZ2	0-120	7.6121E-02(2.63E-02)-	6.8823E-02(3.12E-02)-	3.1354E-01(6.04E-02)-	<b>5.9327E-02(3.50E-02)</b>
ZJZ3	0-120	<b>3.4444E-02(2.31E-02)+</b>	3.5631E-02(8.99E-03)+	1.6510E-01(4.35E-02)-	3.9169E-02(1.73E-02)
ZJZ4	0-120	4.0125E-01(5.43E-02)-	5.0992E-02(2.43E-03)-	2.5178E-01(3.35E-02)-	<b>3.8294E-02(1.07E-03)</b>
ZJZ5	0-120	<b>1.1754E-01(1.25E-01)+</b>	1.3097E-01(1.06E-01)+	1.2305E+00(2.43E-01)-	3.7891E-01(3.25E-01)
ZJZ6	0-120	1.2698E-01(6.20E-02)+	<b>4.3115E-02(2.24E-02)+</b>	4.5433E-01(8.64E-02)-	1.5437E-01(1.36E-01)
DF1	0-120	5.5305E-03(3.59E-04)-	5.2318E-03(4.13E-04)-	4.9660E-03(3.32E-04)-	<b>4.8700E-03(3.28E-04)</b>
DF2	0-120	8.2571E-02(4.99E-04)-	8.1519E-02(3.46E-04)-	8.3628E-02(2.86E-03)-	<b>6.5758E-03(4.96E-04)</b>
DF3	0-120	5.9692E-03(1.46E-03)-	5.8328E-03(4.74E-04)-	2.9943E-01(3.70E-02)-	<b>5.6119E-03(3.86E-04)</b>
DF4	0-120	<b>5.7102E-02(3.47E-03)+</b>	7.8713E-01(1.49E-03)-	8.3908E-02(1.25E-02)-	6.7776E-02(2.27E-03)
DF5	0-120	4.9816E-03(3.02E-04)+	<b>4.9689E-03(1.83E-04)+</b>	5.3249E-03(3.39E-04)-	5.2347E-03(5.21E-04)
DF6	0-120	3.8414E+00(1.43E+00)-	5.6860E-01(3.70E-01)+	<b>2.6958E-01(1.81E-01)+</b>	6.8176E-01(7.13E-01)
DF7	0-120	<b>1.7029E-02(4.55E-04)+</b>	3.1020E-01(1.85E-02)+	2.1328E-01(1.39E-01)+	3.1811E-01(2.15E-02)
DF8	0-120	<b>8.0924E-03(2.01E-03)+</b>	1.7642E-02(2.58E-03)+	1.4459E-02(2.67E-03)+	1.8091E-02(2.49E-03)
DF9	0-120	2.8554E-02(9.45E-03)-	1.5822E-02(5.12E-03)-	7.6924E-02(3.38E-02)-	<b>1.2041E-02(2.96E-03)</b>
DF10	0-120	1.3923E-01(1.27E-02)-	1.2312E-01(1.31E-02)-	<b>9.1282E-02(1.04E-02)+</b>	1.0290E-01(2.48E-02)
DF11	0-120	6.6103E-01(2.78E-03)-	6.5075E-01(1.27E-03)-	6.5611E-01(1.70E-03)-	<b>6.4929E-01(1.16E-03)</b>
DF12	0-120	1.9669E-01(1.89E-02)-	1.3376E-01(8.16E-03)-	1.5959E-01(9.42E-03)-	<b>9.9877E-02(2.77E-03)</b>
DF13	0-120	1.4545E-01(8.88E-03)+	2.4749E+00(4.47E-02)-	<b>8.1594E-02(2.64E-03)+</b>	1.6193E-01(5.24E-03)
DF14	0-120	5.5027E-02(3.01E-03)-	1.0376E+00(1.13E-02)-	<b>3.7214E-02(1.57E-03)+</b>	4.7877E-02(1.46E-03)
best/all	5/27	6/27	6/27	10/27	
+/~/-	9/0/18	12/0/15	8/2/17	—	

translation of the front, so SGEA and the proposed DMOEA-LEM make a better response to environmental change with strong linear dependence. For the dMOP test problems, the PF changes of dMOP1 and dMOP2 are similar to that of FDA2. It seems that our DMOEA-LEM is worse than SGEA and MOEA/D-KF in dMOP1 and dMOP2, respectively. Because the model *A* in the proposed response mechanism has significant randomness, our algorithm performs poorly for such problems with drastic changes in the environment. In ZJZ test suite, except 3-objective ZJZ4 with a fixed and regular concave PF, other test problems are two-dimensional and have separable PF changes. From Table 1, there is no significant difference in the performance of PPS, MOEA/D-KF, and DMOEA-LEM for solving ZJZ test suite, since each of them performs best on two test problems, respectively, while the results of SGEA on this test suite are relatively poor.

Apart from the test problems mentioned above, the results of four algorithms on DF test suite are also provided. It is obvious that DMOEA-LEM is better than other algorithms as it obtains the best results on 6 problems, while PPS, MOEA/D-KF, and SGEA are, respectively, best on 3, 1, and 4 problems. For the 2-objective problems, DMOEA-LEM obtains the best results on DF1-DF3 and DF9. For DF1 and DF2, they have a simple and identical dynamic on the PS, while the PFs of DF1 change from convex to concave over time, and the PFs of DF2 are fixed. Note that the switch of the position-related variables on DF2 is a challenging

problem [57], which causes some variables to lose their correlations in time series. The changes of PFs on DF3 are similar to DF1, but its changes on PSs are more complicated. For DF9, its PFs have a time-varying number of disconnected segments, and there is no doubt that this is also a difficulty in responding the changes. PPS performs better on DF4, DF7, and DF9. The characteristics of these three problems are special and more complicated. Therefore, our DMOEA-LEM is difficult to obtain satisfactory results within the given evaluation budget. In addition, MOEA/D-KF and SGEA are only good at DF5 and DF6, respectively. Particularly, the PFs of DF5 are sometimes linear and sometimes contain several locally concave/convex segments. Also, the challenging property of DF6 is its varying PFs with knee regions/points and long tails [57]. Although MOEA/D-KF and SGEA can solve these two problems better, they perform poorly on other 2-objective DF test problems, which reflect that they have some limitations in solving general DMOPs.

To further investigate and compare the performance differences between DMOEA-LEM and three competitors, the statistical results of MHV metric are provided in Table 2. Overall, DMOEA-LEM shows a better performance when considering all 27 test problems, as it performs best on 11 cases while PPS, Kalman, and SGEA are, respectively, best in 4, 7, and 5 cases. From the one-by-one comparison results which are listed in the second last row, DMOEA-LEM performs better than its three competitors in 19, 15, and 22

TABLE 2: Comparison of results of MSDA-LEM and three competitive DMOEAs on 27 test problems using MHV.

Problems	T	PPS	MOEA/D-KF	SGEA	DMOEA-LEM
FDA1	0-120	7.0738E-01(1.05E-02)+	<b>7.1226E-01(1.32E-03)+</b>	7.0948E-01(2.29E-03)-	7.0999E-01(4.11E-03)
FDA2	0-120	5.7399E-01(2.54E-03)-	<b>5.8237E-01(7.35E-03)+</b>	5.5317E-01(2.99E-03)-	5.7854E-01(7.62E-04)
FDA3	0-120	5.0135E-01(3.12E-02)-	<b>5.6870E-01(5.80E-03)+</b>	5.2743E-01(6.30E-03)-	5.5305E-01(8.31E-03)
FDA4	0-120	4.9395E-01(7.99E-03)-	5.4267E-01(4.11E-03)-	5.5231E-01(3.02E-03)-	<b>5.6285E-01(1.76E-03)</b>
FDA5	0-120	5.1468E-01(9.79E-03)-	5.4703E-01(4.67E-03)-	5.5553E-01(2.82E-03)-	<b>5.6406E-01(1.96E-03)</b>
dMop1	0-120	5.3517E-01(2.22E-02)-	5.5131E-01(1.35E-03)+	<b>5.5531E-01(1.68E-03)+</b>	5.4669E-01(1.93E-03)
dMop2	0-120	5.4088E-01(8.85E-03)-	<b>5.4871E-01(1.92E-03)+</b>	5.4484E-01(3.49E-03)-	5.4763E-01(4.29E-03)
ZJZ1	0-120	5.8610E-01(3.19E-02)+	<b>5.9753E-01(1.21E-02)+</b>	1.6313E-01(2.49E-02)-	5.6136E-01(4.29E-02)
ZJZ2	0-120	<b>5.8181E-01(1.68E-02)+</b>	5.7625E-01(1.88E-02)+	3.7886E-01(3.28E-02)-	5.6657E-01(3.60E-02)
ZJZ3	0-120	<b>6.0776E-01(1.94E-02)+</b>	5.9760E-01(9.99E-03)+	4.9035E-01(2.67E-02)-	5.8742E-01(2.11E-02)
ZJZ4	0-120	1.5196E-01(2.34E-02)-	5.2608E-01(3.64E-03)-	3.7274E-01(2.33E-02)-	<b>5.6039E-01(1.28E-03)</b>
ZJZ5	0-120	5.1720E-01(7.31E-02)+	<b>5.4510E-01(4.57E-02)+</b>	2.1556E-01(3.96E-02)-	3.8459E-01(1.32E-01)
ZJZ6	0-120	4.7096E-01(5.38E-02)+	<b>5.7869E-01(2.93E-02)+</b>	2.8024E-01(3.80E-02)-	4.5322E-01(1.18E-01)
DF1	0-120	5.5454E-01(6.51E-04)-	5.5523E-01(7.66E-04)-	5.5551E-01(6.25E-04)-	<b>5.5620E-01(6.49E-04)</b>
DF2	0-120	6.1380E-01(6.35E-04)-	6.1554E-01(4.88E-04)-	6.1417E-01(1.52E-03)-	<b>7.1626E-01(9.45E-04)</b>
DF3	0-120	5.2502E-01(1.99E-03)-	5.2486E-01(8.53E-04)-	3.1600E-01(2.17E-02)-	<b>5.2561E-01(7.91E-04)</b>
DF4	0-120	6.6303E-01(2.46E-03)-	3.7539E-01(2.84E-04)-	6.4561E-01(4.02E-03)-	<b>6.6341E-01(6.91E-04)</b>
DF5	0-120	<b>5.7900E-01(5.44E-04)+</b>	5.7828E-01(4.76E-04)+	5.7796E-01(6.00E-04)-	5.7825E-01(8.84E-04)
DF6	0-120	9.1614E-03(2.25E-02)-	4.2665E-01(1.88E-01)-	<b>6.8886E-01(3.29E-02)+</b>	5.4100E-01(1.80E-01)
DF7	0-120	<b>4.6136E-01(2.93E-04)+</b>	4.3710E-01(2.08E-03)+	4.0741E-01(2.16E-02)-	4.3547E-01(2.90E-03)
DF8	0-120	6.1724E-01(1.76E-03)-	6.1792E-01(8.19E-04)-	6.1726E-01(2.07E-03)-	<b>6.1802E-01(3.89E-04)</b>
DF9	0-120	5.1398E-01(1.38E-02)-	5.3831E-01(6.91E-03)-	4.5920E-01(3.59E-02)-	<b>5.4413E-01(4.80E-03)</b>
DF10	0-120	6.0496E-01(1.35E-02)-	7.0691E-01(4.74E-03)-	<b>7.1994E-01(4.58E-03)+</b>	7.1552E-01(1.19E-02)
DF11	0-120	5.9278E-02(2.06E-03)-	6.4815E-02(6.75E-04)-	<b>6.9978E-02(6.57E-04)+</b>	6.7767E-02(6.10E-04)
DF12	0-120	7.8594E-01(1.28E-02)-	8.3956E-01(3.29E-03)-	8.2200E-01(1.04E-02)-	<b>8.6360E-01(8.77E-04)</b>
DF13	0-120	6.2312E-01(9.98E-03)-	3.3035E-01(7.06E-03)-	<b>6.8362E-01(2.94E-03)+</b>	6.4882E-01(8.85E-03)
DF14	0-120	5.7718E-01(6.15E-03)-	3.0956E-01(1.60E-03)-	6.1092E-01(2.28E-03)-	<b>6.1341E-01(1.75E-03)</b>
best/all	4/27	7/27	5/27	11/27	
+/-/-	8/0/19	12/0/15	5/0/22	--	

out of 27 cases. It is obvious that the results obtained by four algorithms on MHV metric and MIGD metric are quite divergent. Specifically, although DMOEA-LEM is worse than MOEA/D-LEM on the classical DMOPs such as FDA, dMOP, and ZJZ test suites, there are 8 best MHV values obtained by DMOEA-LEM on the novel 14 DF test problems, while MOEA/D-KF is completely inferior to DMOEA-LEM on the DF problems. In addition, for all test instances, both PPS and SGEA fail to show encouraging performance in terms of the MHV metric. In summary, according to Tables 1 and 2, the proposed DMOEA-LEM shows some appealing results in a variety of DMOPs, which benefits from the effective detection and response mechanism for environmental changes. However, when considering that PPS and MOEA/D-KF need to train a large number of mathematical models, their prediction strategies may fail to detect periodic drastic environmental changes. For SGEA, in the early stage of evolution, the quantity and quality of non-dominated solutions may not be sufficient to support the algorithm to obtain effective population movement information. Furthermore, SGEA also has certain limitations in solving the problems with strong variable linkages or changes causing a significant diversity loss.

To have a graphical view of these algorithms' tracking ability, the PF approximations of 2-objective FDA2, dMOP1, DF8, and DF4 are plotted in Figures 5 and 6. As shown in

Figure 5, for FDA2, its true PFs change from convex to concave. When the continuous environment changes significantly, the distributions of the corresponding true PFs are relatively sparse (i.e., the middle region of the FDA2's true PFs). It can be observed that the solutions obtained by PPS are roughly uniformly distributed on the corresponding sparse PFs, but it is difficult for PPS to achieve convergence when the changes of PFs are not obvious (i.e., the extremely convex and extremely concave regions). For SGEA, its obtained solutions cannot match the corresponding PFs evenly, which mainly concentrate on the middle region of the true PFs. The performance of MOEA/D-KF is not satisfactory, as it could not explore the solutions in the densely distributed concave PFs. By contrast, the proposed DMOEA-LEM outperforms the competitors in both convergence and diversity, and the obtained solutions are distributed evenly and widely to cover all the PFs. For dMOP1, it has the same characteristics as FDA2, but the change degree between the adjacent PFs is reduced. Hence, the true PFs of dMOP1 are denser. There is no significant difference in the performance of PPS, SGEA and DMOEA-LEM, as their obtained solutions can cover each PF uniformly and completely. However, for MOEA/D-KF, although its obtained solutions also have an excellent distribution, there are some solutions outside the boundary of PFs, which are useless to the evolution of population. For

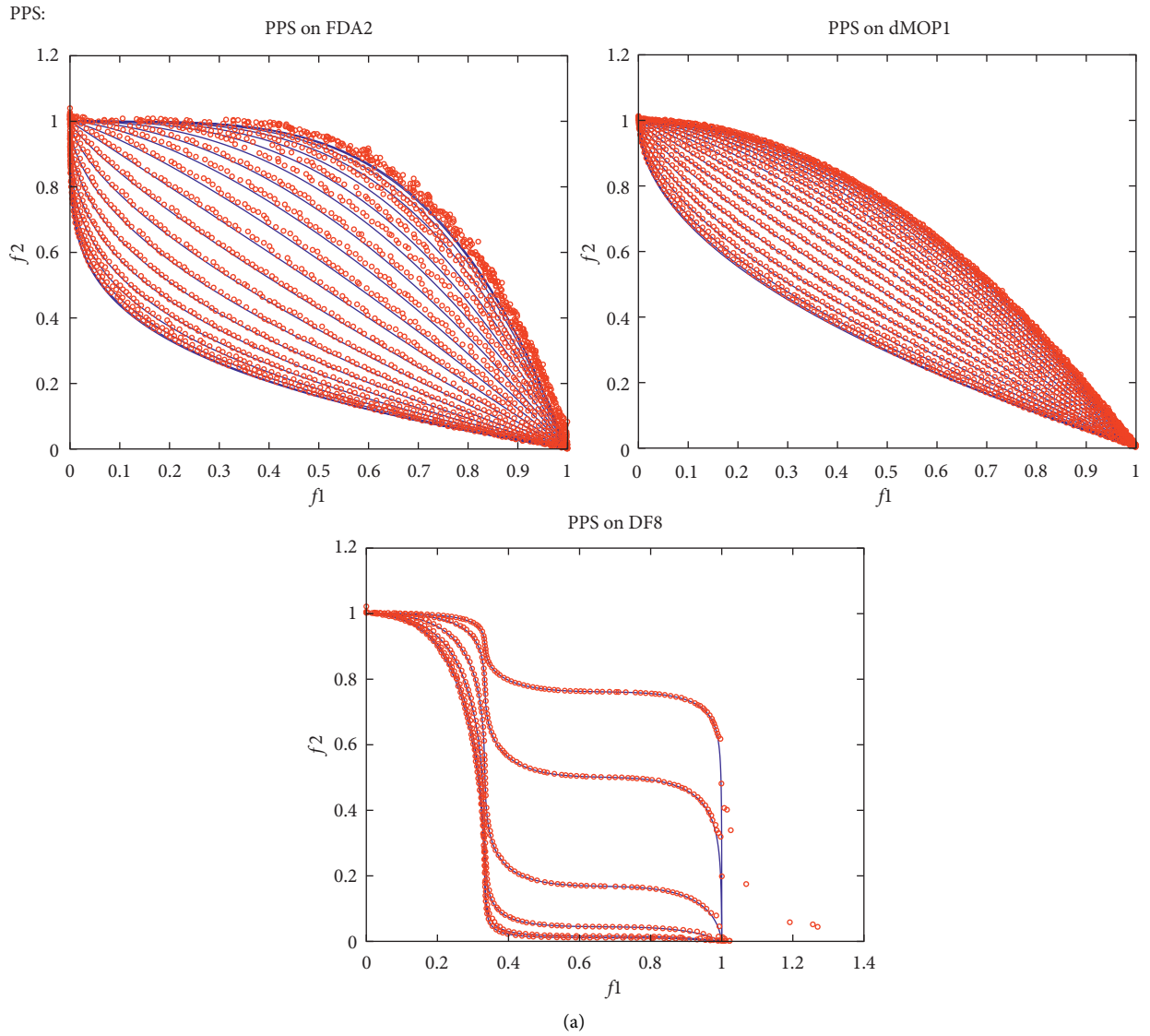


FIGURE 5: Continued.

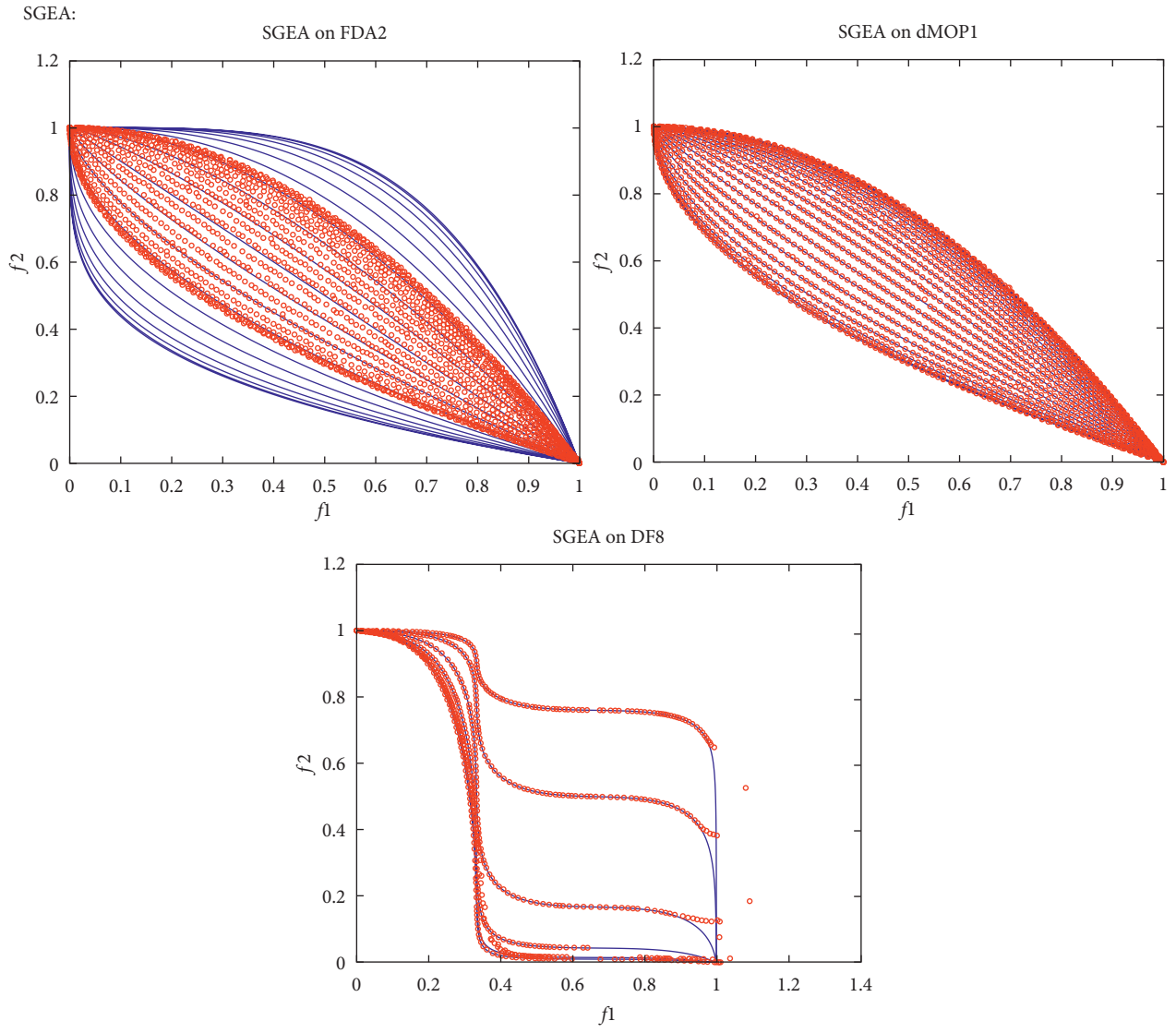
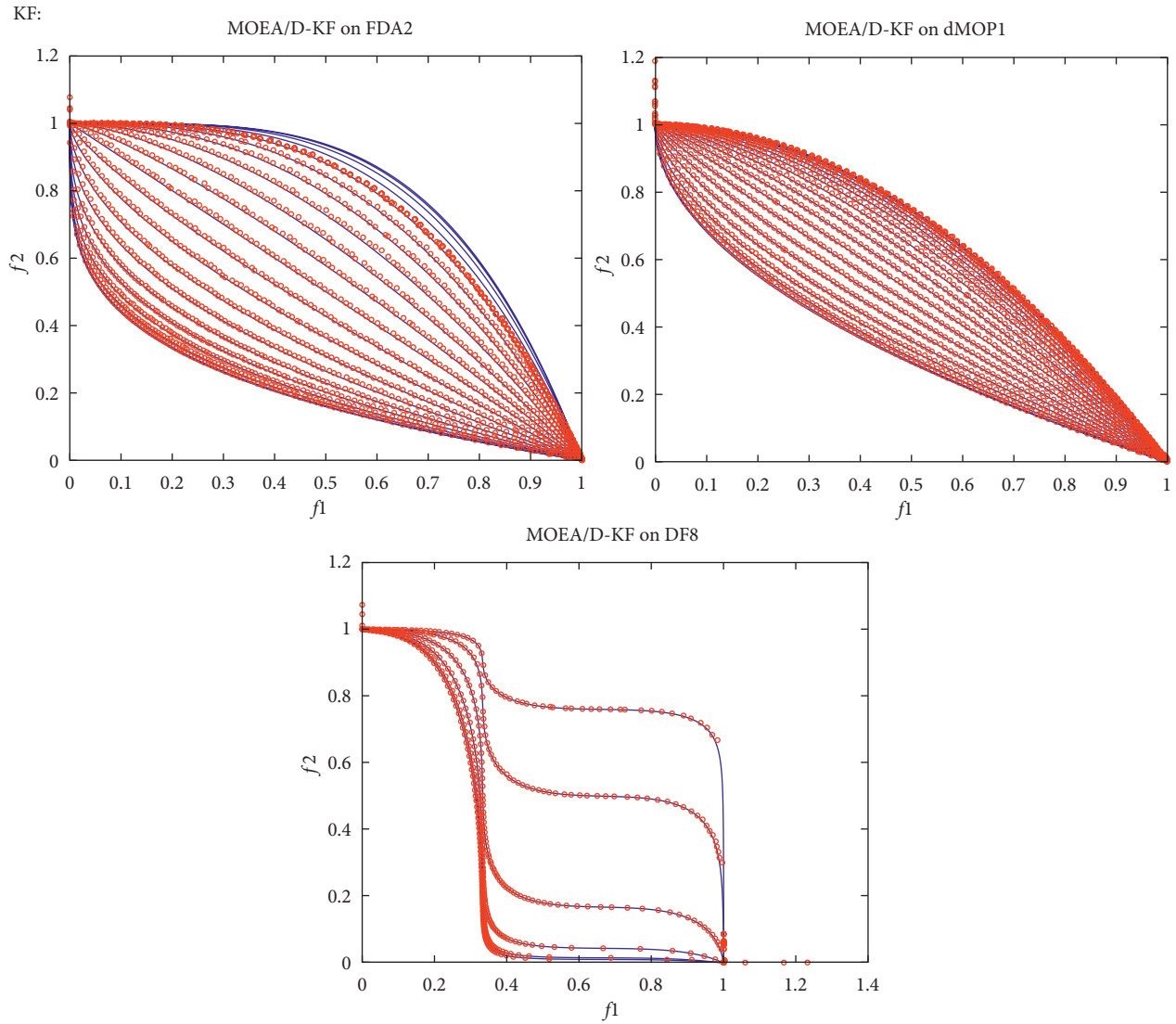


FIGURE 5: Continued.



(c)

FIGURE 5: Continued.

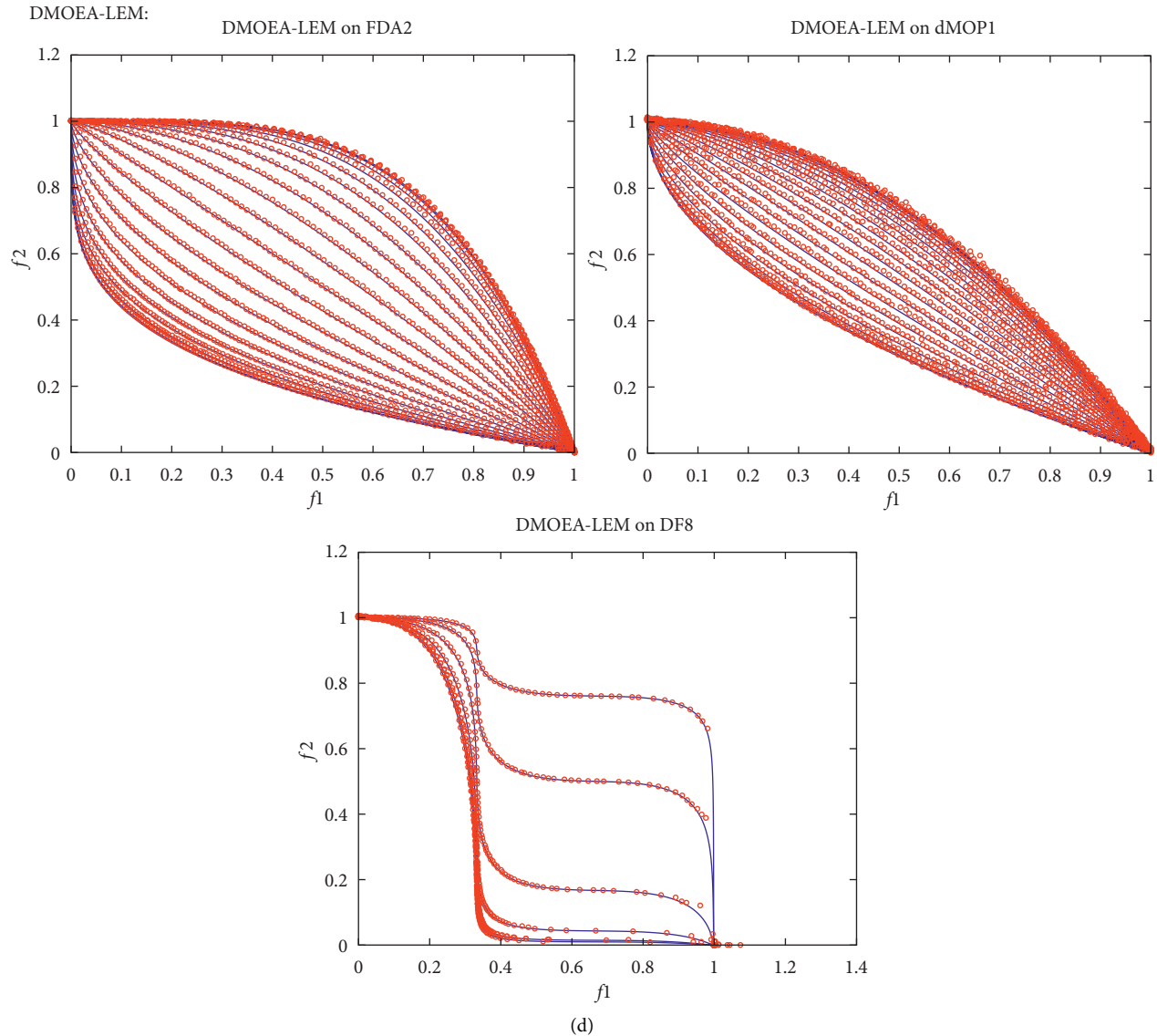
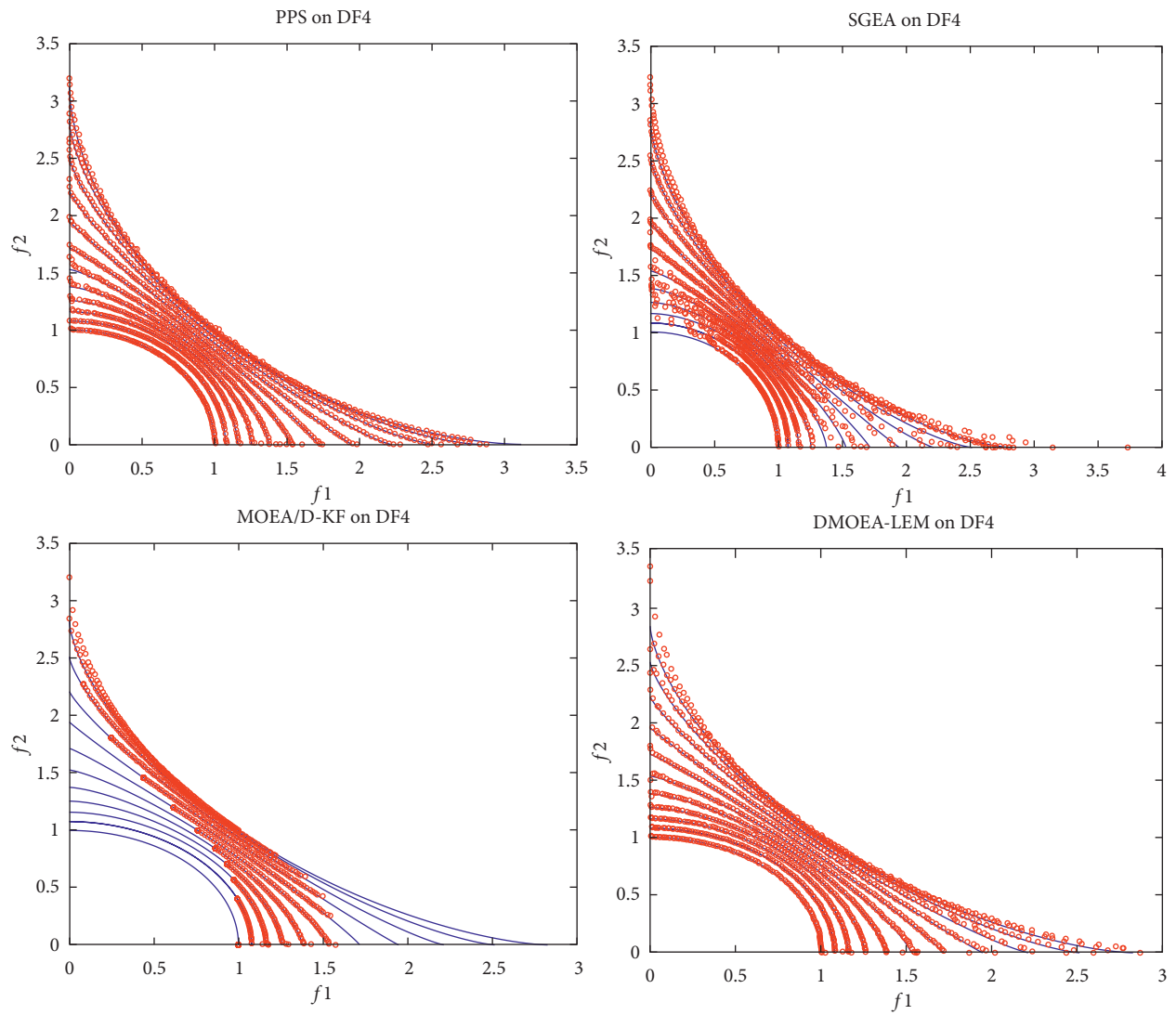


FIGURE 5: The approximate PFs of the compared algorithms for different 2-objective problems with  $\tau_T = 30$  and  $n_T = 10$ .

the mixed and complex DF8, its overall PF geometry switches between concavity and convexity, which contains knee regions [57]. Moreover, although its PSs change over time, the centroid of PSs is stationary, which indicates that the difficulties for optimization are increased. Among the four compared algorithms, the solutions obtained by PPS are basically distributed around the true PFs. However, in the extremely concave area of the top three true PFs, the hillside is not covered by sufficient solutions. This phenomenon appears in all compared algorithms. For SGEA, there are some faults in the distribution of its solutions on several PFs at the bottom. For MOEA/D-KF and DMOEA-LEM, the diversity is also not promising, as their obtained solutions are mainly distributed in the corner near the 2nd objective axis and the convex knee regions. In addition, all compared algorithms have shown the case that solutions are beyond the boundary of true PFs, which implicitly confirms the challenge of solving DF8.

In Figure 6, the approximate PFs of DMOEA-LEM and its competitors on DF4 are provided. Since some different PFs intersect each other in the objective space during evolution, the introduction of PF changes in a cycle are divided into two parts in detail. Figures 6(a) and 6(b) show the distribution of solutions obtained by four algorithms in the first and second parts of a cycle, respectively. In Figure 6(a), it can be seen that the solutions obtained by PPS and DMOEA-LEM can fully and uniformly cover all the true PFs. In contrast, the distribution of solutions obtained by DMOEA-LEM on the convex PF at the top is slightly worse than that of PPS. For MOEA/D-KF, each of its populations has a partial absence, which cannot cover the true PFs entirely. The performance of SGEA is better than that of MOEA/D-KF, but its diversity on the true PFs near the coordinate axes is poor. The performance differences of four algorithms are reflected more incisively and vividly in Figure 6(b), and the solutions obtained by PPS and





(a)

FIGURE 6: Continued.

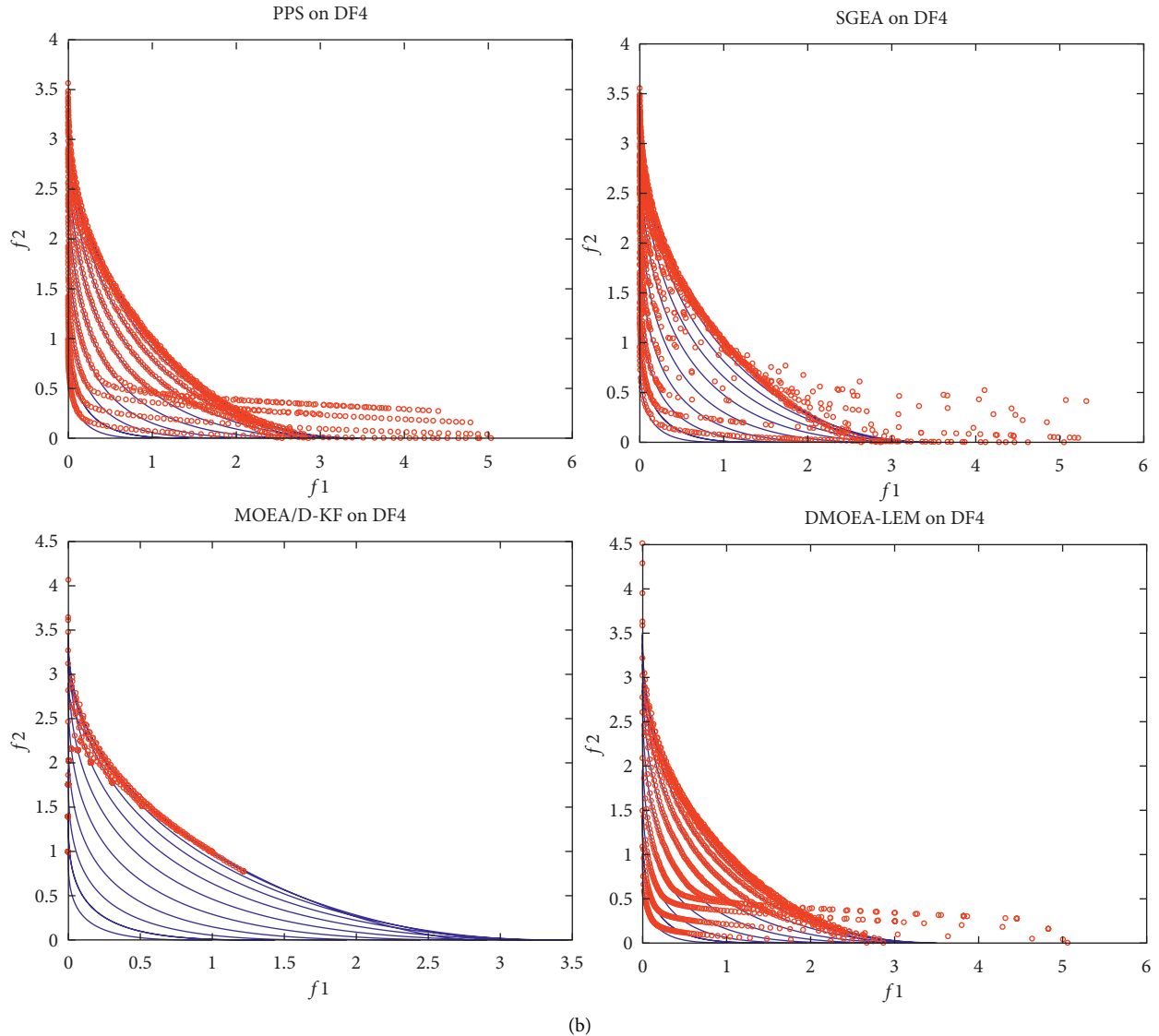


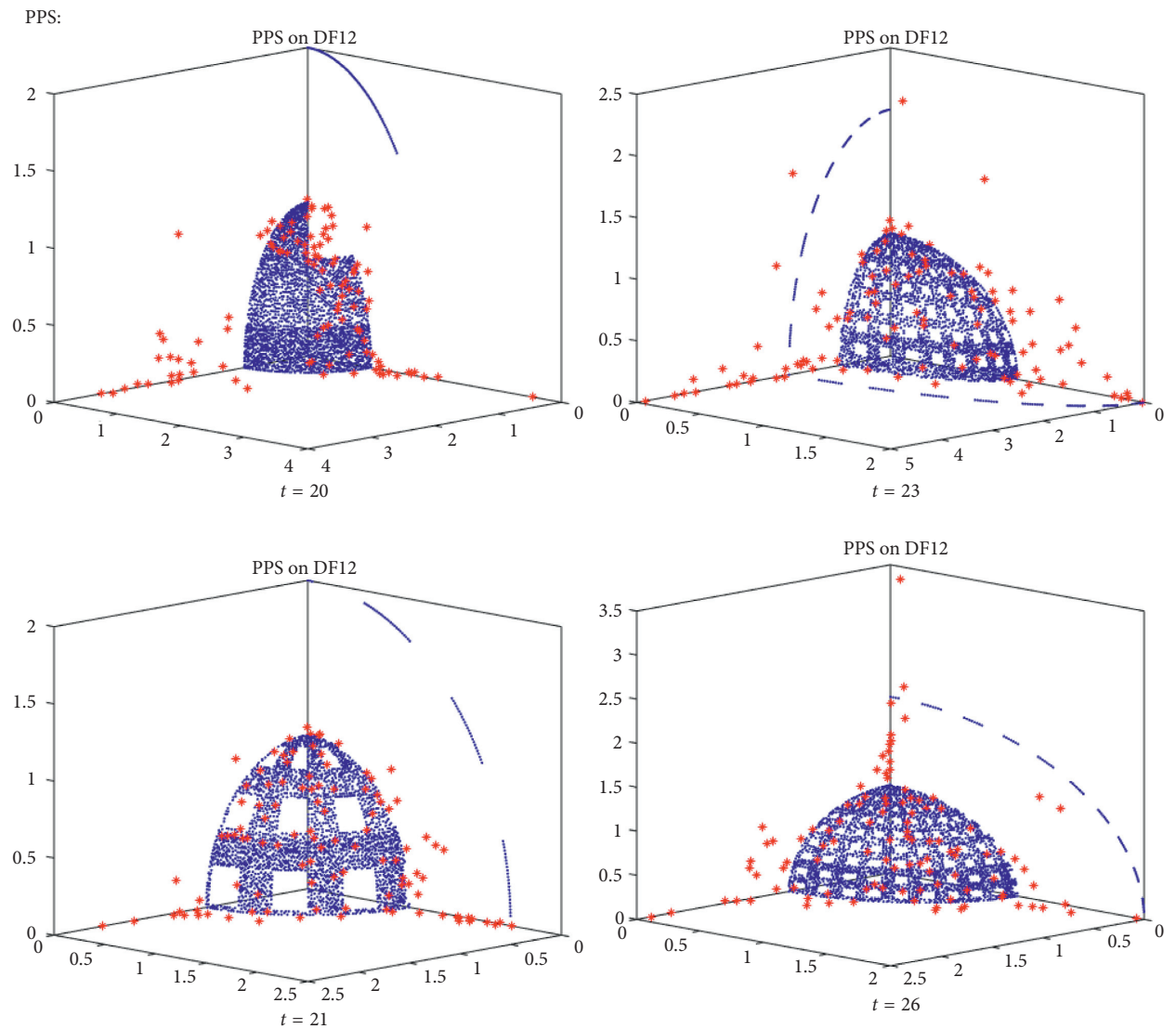
FIGURE 6: The approximate PFs of the compared algorithms on 2-objective DF4 problems with  $\tau_T = 30$  and  $n_T = 10$ .

DMOEA-LEM generally detect the profile of the corresponding PFs, although there are some challenges in population convergence on the true PFs near the 1st axis. For SGEA, the convergence of population on the middle part of the true PFs is not encouraging. However, the performance of MOEA/D-KF is even worse, as its solutions are all concentrated on one curve and it seems that the number of solutions is less than the population size.

To further investigate the performance of DMOEA-LEM in tackling 3-objective DMOPs, the solutions obtained by all the considered algorithms at four moments are, respectively, given in Figure 7 for DF12. For this test problem, its PS changes over time and it has a time-varying number of PF holes. In these figures, the blue region represents the true PF and the red points still indicate the solutions obtained by DMOEAs. In a cycle, we show the simulations of four times (i.e.,  $t = 20, 21, 23, 26$ ) with significant environmental changes. For PPS, its obtained solutions in each environment are pretty discrete, and they cannot reach a good

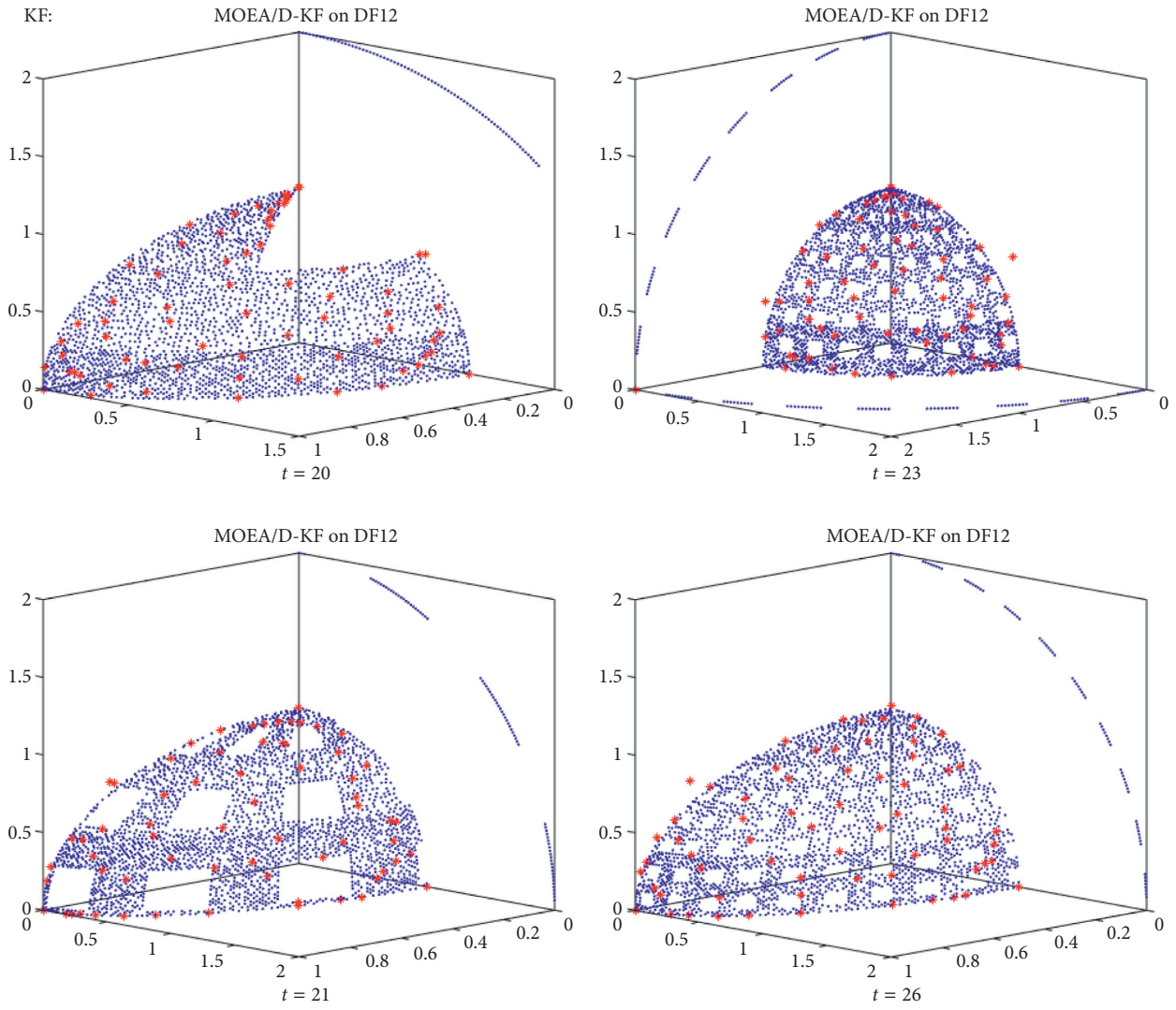
convergence. Considering MOEA/D-KF, its solutions can be uniformly distributed on each PF, but the diversity of solutions is not enough, which leads to the sparse distribution of solutions. For SGEA, its solutions can evenly cover the PFs at  $t = 20$  and  $t = 21$ , while in the latter two environments, the solutions are mainly concentrated in the upper part of the true PF. Our DMOEA-LEM performs relatively well, and its obtained solutions can approximate the true PFs with even distribution.

**4.3. Further Discussion.** In order to further demonstrate the superiority of our algorithm in responding to changes, we test the performance of the prediction model on DF1 and DF7 problems, respectively. Because the number of evolutions of each environment is limited, for new optimization problems, an initial population close to the true PF will greatly reinforce the convergence pressure. Figures 8 and 9 show the distribution of the initial population in the

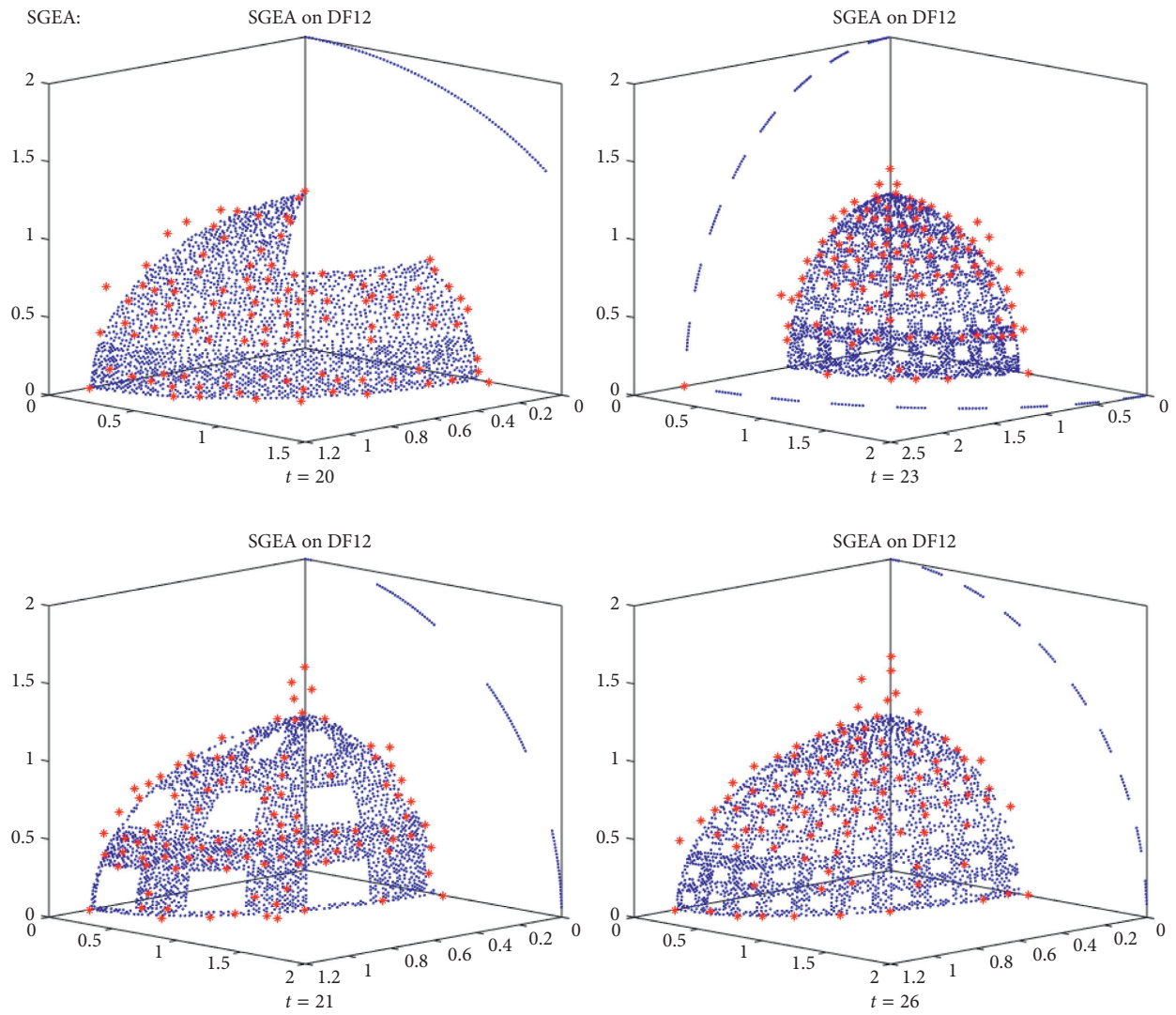


(a)

FIGURE 7: Continued.



(b)  
FIGURE 7: Continued.



(c)  
FIGURE 7: Continued.

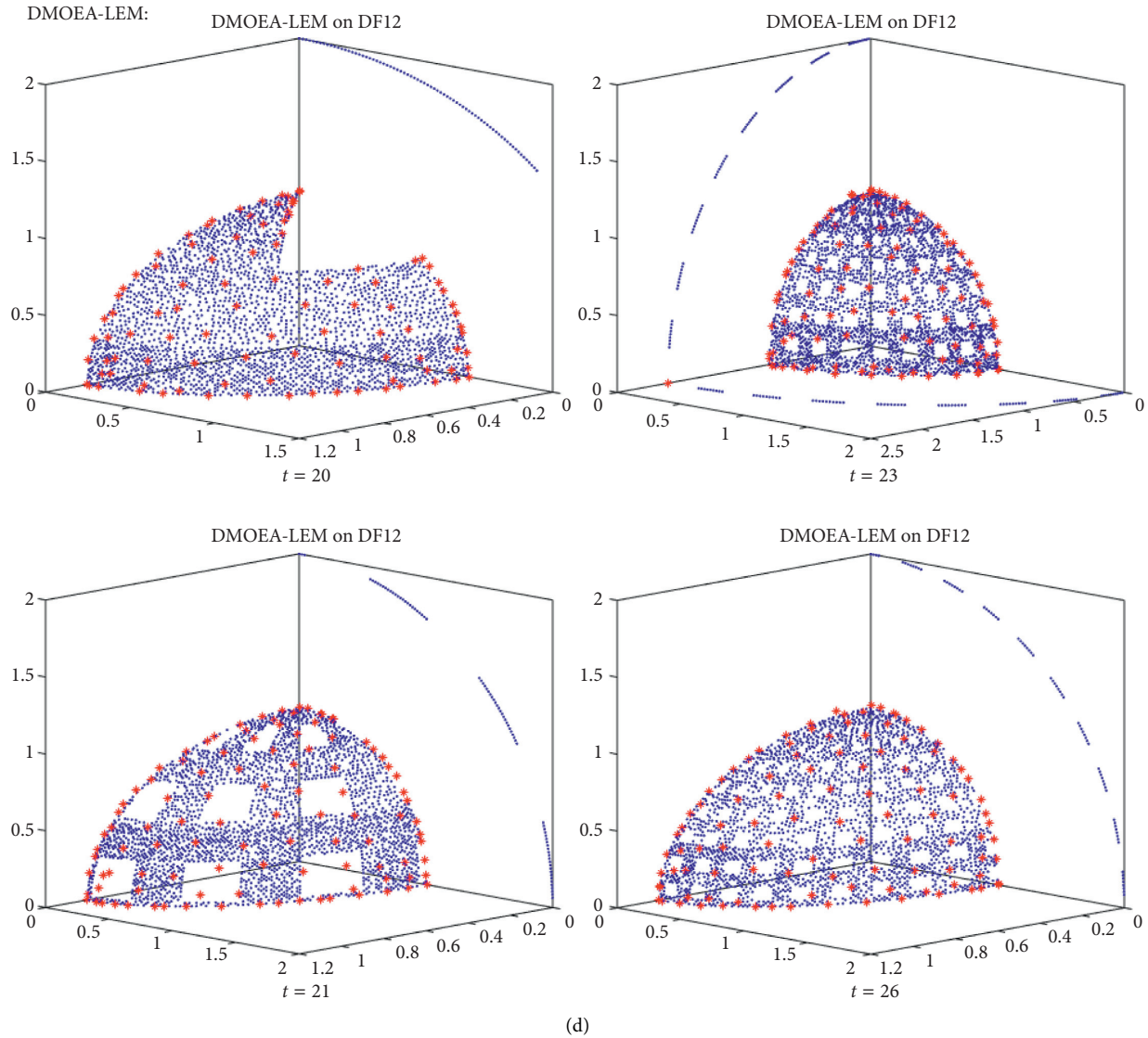


FIGURE 7: The approximate PFs of the compared algorithms on 3-objective DF12 problems with  $\tau_T = 30$  and  $n_T = 10$ .

objective space after the change of the four algorithms. As can be seen from Figure 8, when the change occurs, the initial populations generated by PPS are difficult to cover the true PFs, and it is obvious that some initial populations are far away from the true PF. For MOEA/D-KF and SGEA, the diversity of the initial populations is poor, and the efficient solutions close to the true PF cannot be detected in some

changes. Similar shortcomings are also shown in Figure 9. In contrast, the proposed DMOEA-LEM performs well in response to changes. The initial populations generated by our response mechanism basically cover the true PFs, but the deficiency is that there are still some solutions that are far away from the range of true PFs. From Figure 9, except for the last two changes, each initial population generated by

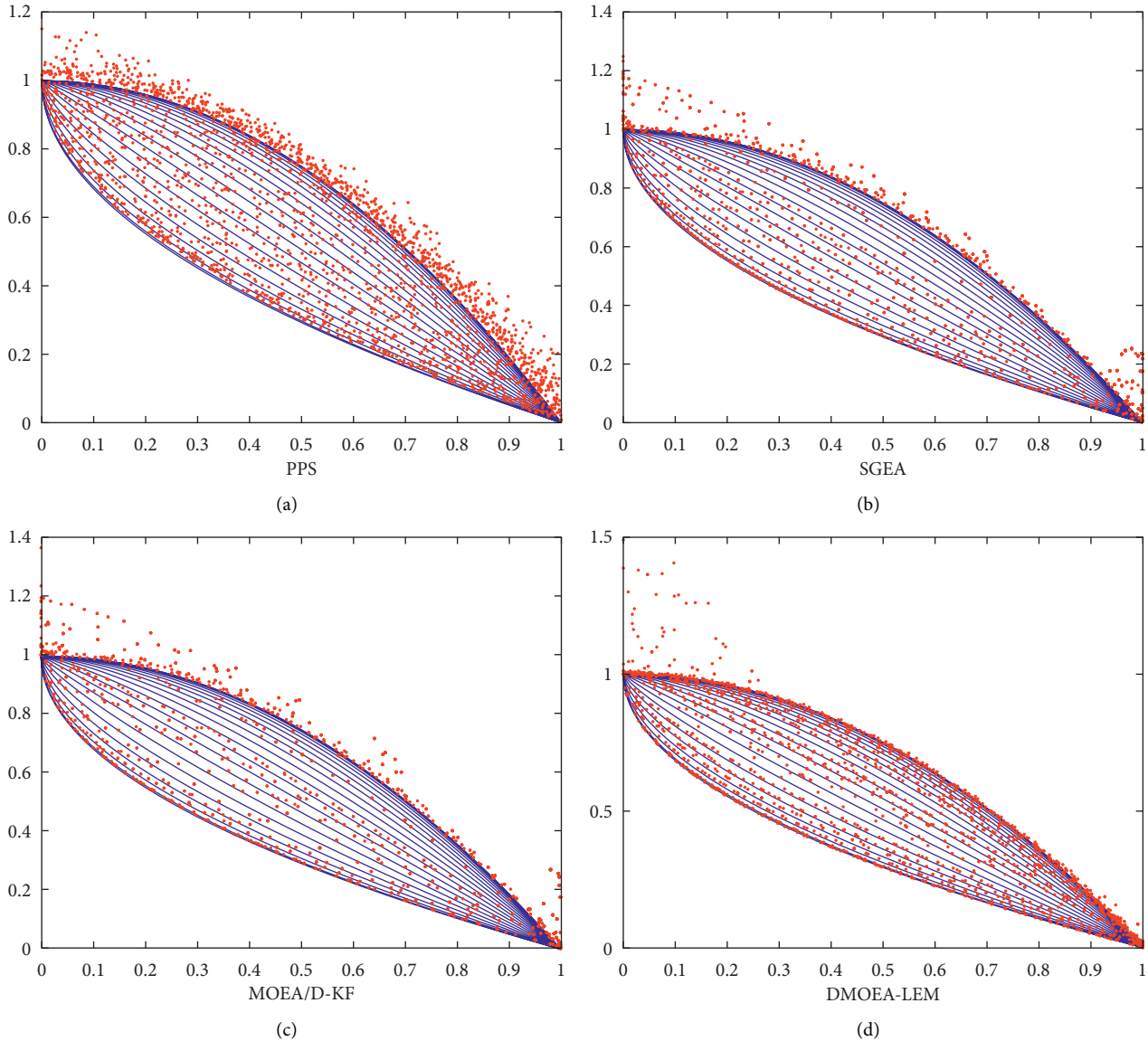


FIGURE 8: The distribution of initial populations generated by four DMOEAs on DF1.

DMOEA-LEM can generally match the corresponding true PF, while the initial population generated by PPS is relatively divergent.

## 5. Conclusions and Future Work

In this paper, we propose a novel DMOEAs with multiple response strategies based on linear environment detection to solve various DMOPs. This algorithm includes two main components, i.e., the environment detection mechanism based on linear changes and the change response mechanism based on change types. When the environmental change is successfully detected, the linear correlation coefficient of two continuous environments is calculated by using the movement of population in the objective space before and after the change, which indicates three types of

environmental change. Then, the change response mechanism will activate a prediction model corresponding to each type, which is used to initialize population. In this approach, we use random initialization of nondominated solutions for linear independent changes, a centroid translation strategy for strong linear correlation, and a compromise strategy for weak linear correlation. The effectiveness of our algorithm is validated by numerous experimental studies on solving different test DMOPs (i.e., FDA1-FDA5, dMOP1-dMOP2, ZJZ1-ZJZ6, and DF1-DF14). When compared to three competitive DMOEAs (PPS, MOEA/D-KF, and SGEA), the experiments confirm our superior performance especially on the novel test suite DF.

In the future work, we will try to solve the problems with more than three objectives in dynamic environment and develop more effective population prediction methods. Moreover, the application of DMOEAs on some

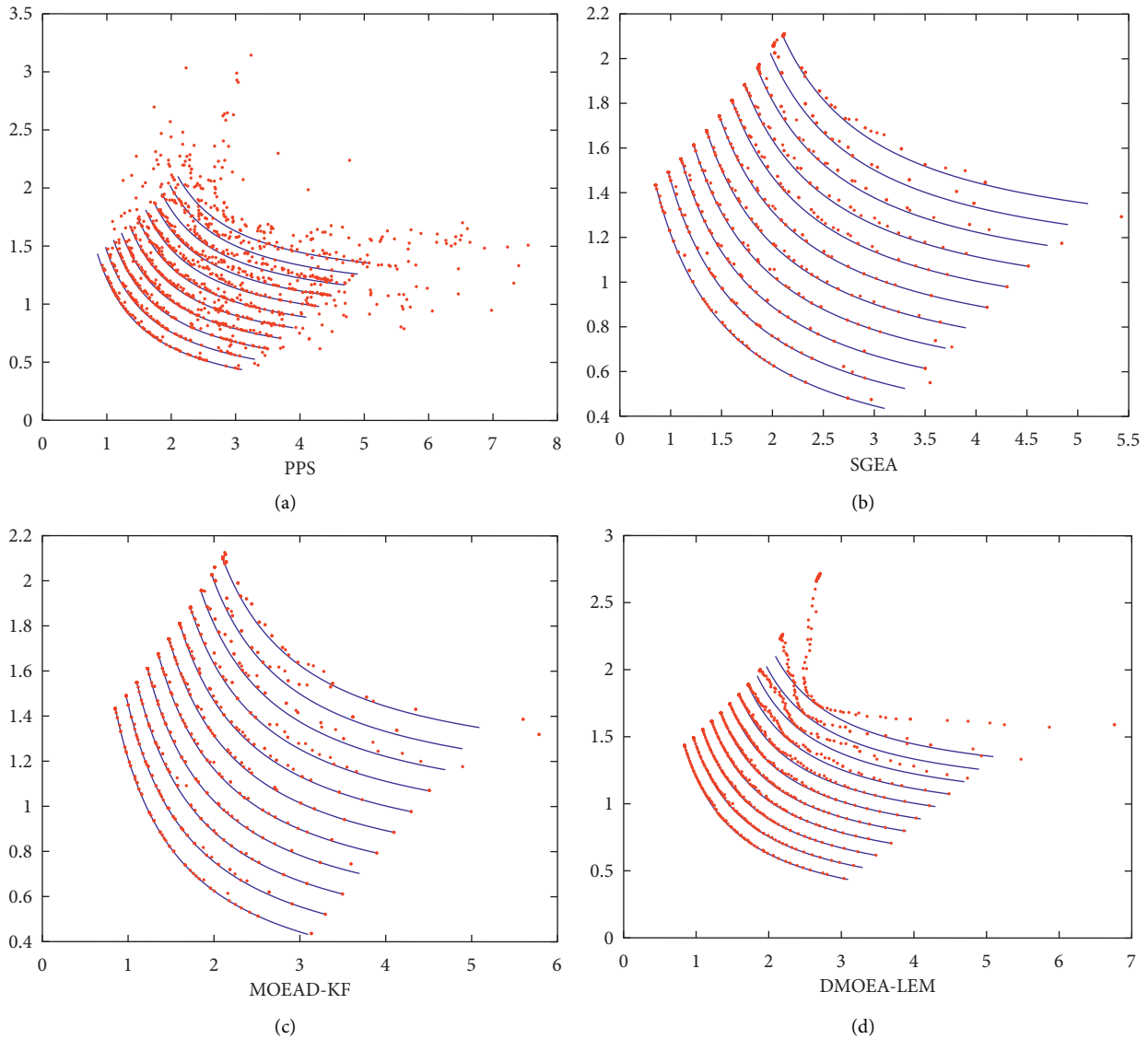


FIGURE 9: The distribution of initial populations generated by four DMOEAs on DF7.

real-world problems will also be the focus of our work in the future.

### Data Availability

The data used to support the findings of the study are available from the corresponding author upon request.

### Conflicts of Interest

The authors declare that they have no conflicts of interest.

### Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 61876110, 61836005, and 61672358, Joint Funds of the National Natural Science Foundation of China under Key Program Grant U1713212,

Natural Science Foundation of Guangdong Province under Grant 2017A030313338, and Shenzhen Technology Plan under Grants JCYJ20170817102218122 and JCYJ190808164211203.

### References

- [1] P. P.-Y. Wu, D. Campbell, and T. Merz, "Multi-objective four-dimensional vehicle motion planning in large dynamic environments," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 3, pp. 621–634, 2011.
- [2] Y. Le, D. K. He, and B. Li, "A selection hyper-heuristic algorithm for multiobjective dynamic economic and environmental load dispatch," *Complexity*, vol. 2020, Article ID 4939268, 18 pages, 2020.
- [3] T. Cheng, M. Chen, and Y. Wang, "Adaptive robust method for dynamic economic emission dispatch incorporating



- renewable energy and energy storage,” *Complexity*, vol. 2018, Article ID 2517987, 13 pages, 2018.
- [4] X. Li, J. G. Lai, and R. L. Tang, “A hybrid constraints handling strategy for multiconstrained multiobjective optimization problem of microgrid economical/environmental dispatch,” *Complexity*, vol. 2017, Article ID 6249432, 12 pages, 2017.
  - [5] M. H. Du, Z. Cheng, Y. F. Zhang et al., “Multiobjective optimization of tool geometric parameters using genetic algorithm,” *Complexity*, vol. 2018, Article ID 9692764, 14 pages, 2018.
  - [6] J. Zhao, X. Lei, and F. X. Wu, “Predicting protein complexes in weighted dynamic PPI networks based on ICSC,” *Complexity*, vol. 2017, 11 pages, 2017.
  - [7] S. Palaniappan, S. Zein-Sabatto, and A. Sekmen, “Dynamic multiobjective optimization of war resource allocation using adaptive genetic algorithms,” in *Proceedings of the IEEE SoutheastCon*, pp. 160–165, Clemson, SC, USA, April 2001.
  - [8] X. Wu, Z. Ma, and Y. Wang, “Joint user grouping and resource allocation for multi-user dual layer beamforming in LTE-a,” *IEEE Communications Letters*, vol. 19, no. 10, pp. 1822–1825, 2015.
  - [9] Q. Wang, “Integral reinforcement learning control for a class of high-order multivariable nonlinear dynamics with unknown control coefficients,” *IEEE Access*, vol. 8, pp. 86223–86229, 2020.
  - [10] Q. Wang and C. Sun, “Adaptive consensus of multiagent systems with unknown high-frequency gain signs under directed graphs,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 6, pp. 2181–2186, 2020.
  - [11] Q. Wang, H. E. Psillakis, and C. Sun, “Cooperative control of multiple agents with unknown high-frequency gain signs under unbalanced and switching topologies,” *IEEE Transactions on Automatic Control*, vol. 64, no. 6, pp. 2495–2501, 2019.
  - [12] M. Farina, K. Deb, and P. Amato, “Dynamic multiobjective optimization problems: test cases, approximations, and applications,” *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 5, pp. 425–442, 2004.
  - [13] E. Tantar, A.-A. Tantar, and P. Bouvry, “On dynamic multi-objective optimization, classification and performance measures,” in *Proceedings of the IEEE Congress of Evolutionary Computation (CEC)*, pp. 2759–2766, New Orleans, LA, USA, June 2011.
  - [14] A. Zhou, Y. Jin, and Q. Zhang, “A population prediction strategy for evolutionary dynamic multiobjective optimization,” *IEEE Transactions on Cybernetics*, vol. 44, no. 1, pp. 40–53, 2014.
  - [15] G. Ruan, G. Yu, J. Zheng, J. Zou, and S. Yang, “The effect of diversity maintenance on prediction in dynamic multi-objective optimization,” *Applied Soft Computing*, vol. 58, pp. 631–647, 2017.
  - [16] C.-K. Goh and K. C. Tan, “A competitive-cooperative co-evolutionary paradigm for dynamic multi-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 1, pp. 103–127, 2009.
  - [17] Y. Wu, Y. Jin, and X. Liu, “A directed search strategy for evolutionary dynamic multiobjective optimization,” *Soft Computing*, vol. 19, no. 11, pp. 3221–3235, 2015.
  - [18] Z. Peng, J. Zheng, J. Zou, and M. Liu, “Novel prediction and memory strategies for dynamic multiobjective optimization,” *Soft Computing*, vol. 19, no. 9, pp. 2633–2653, 2015.
  - [19] S. Jiang and S. Yang, “A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 1, pp. 65–82, 2016.
  - [20] M. Arrchana, K. C. Tan, V. Prahlaad et al., “Evolutionary dynamic multiobjective optimization via kalman filter prediction,” *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 2862–2873, 2016.
  - [21] R. Chen, K. Li, and X. Yao, “Dynamic multiobjectives optimization with a changing number of objectives,” *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 157–171, 2017.
  - [22] M. Rong, D. Gong, W. Pedrycz et al., “A multi-model prediction method for dynamic multi-objective evolutionary optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, 2019.
  - [23] R. C. Liu, Y. Y. Chen, W. P. Ma et al., “A novel cooperative coevolutionary dynamic multi-objective optimization algorithm using a new predictive model,” *Soft Computing*, vol. 18, no. 10, pp. 1913–1929, 2014.
  - [24] S. Nguyen, M. Zhang, M. Johnston, and K. C. Tan, “Automatic design of scheduling policies for dynamic multi-objective job shop scheduling via cooperative coevolution genetic programming,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 193–208, 2014.
  - [25] G. Yao, Y. Ding, Y. Jin, and K. Hao, “Endocrine-based co-evolutionary multi-swarm for multi-objective workflow scheduling in a cloud system,” *Soft Computing*, vol. 21, no. 15, pp. 4309–4322, 2017.
  - [26] G. H. Nguyen, J.-H. Shin, and W.-H. Kim, “Autotuning controller for motion control system based on intelligent neural network and relay feedback approach,” *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 3, pp. 1138–1148, 2015.
  - [27] Q. Wei, D. Liu, and X. Yang, “Infinite horizon self-learning optimal control of nonaffine discrete-time nonlinear systems/finite horizon self-learning optimal control of nonaffine discrete-time nonlinear systems,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 4, pp. 866–879, 2015.
  - [28] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff et al., “Predictionbased population re-initialization for evolutionary dynamic multi-objective optimization,” in *LNCS*, vol. 4403, pp. 832–846, Springer, Berlin, Germany, 2007.
  - [29] M. Greff and A. P. Engelbrecht, “Solving dynamic multi-objective problems with vector evaluated particle swarm optimisation,” in *Proceedings of the IEEE CEC*, pp. 2922–2929, Hongkong, China, July 2008.
  - [30] R. Liu, J. Fan, and L. Jiao, “Integration of improved predictive model and adaptive differential evolution based dynamic multi-objective evolutionary optimization algorithm,” *Applied Intelligence*, vol. 43, no. 1, pp. 192–207, 2015.
  - [31] H. Richter, “Detecting change in dynamic fitness landscapes,” in *Proceedings of the IEEE (CEC)*, pp. 1613–1620, Trondheim, Norway, February 2009.
  - [32] K. Deb and S. Karthik, “Dynamic multi-objective optimization and decision-making using modified NSGA-II: a case study on hydro-thermal power scheduling,” *International Conference on Evolutionary Multi-Criterion Optimization*, Springer, Berlin, Heidelberg, pp. 803–817, 2007.
  - [33] B. Zheng, “A new dynamic multi-objective optimization evolutionary algorithm,” vol. 5, pp. 565–570, in *Proceedings of the Third International Conference on Natural Computation*, vol. 5, IEEE, Haikou, China, August 2007.
  - [34] R. Liu, W. Zhang, L. Jiao et al., “A sphere-dominance based preference immune-inspired algorithm for dynamic multi-objective optimization,” in *Proceedings of the Genetic*

- Evolutionary Computations Conference*, pp. 423–430, Portland, OR, USA, September 2010.
- [35] J. J. Grefenstette, “Genetic algorithms for changing environments,” *Parallel Problem Solving from Nature2*, pp. 137–144, Elsevier, Amsterdam, Netherlands, 1992.
- [36] C. Li and S. Yang, “A general framework of multipopulation methods with clustering in undetectable dynamic environments,” *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 4, pp. 556–577, 2012.
- [37] F. Vavak, K. Jukes, and T. C. Fogarty, “Learning the local search range for genetic optimisation in nonstationary environments,” in *Proceedings of the IEEE International Conference on Evolutionary Computation ICEC’97*, pp. 355–360, IEEE Publishing, Indianapolis, IN, USA, April 1997.
- [38] R. W. Morrison, *Designing Evolutionary Algorithms for Dynamic Environments*, Springer-Verlag, Berlin, Germany, 2004.
- [39] H. G. Cobb and J. J. Grefenstette, “Genetic algorithms for tracking changing environments,” in *Proceedings of the 5th International Conference on Genetic Algorithms*, pp. 523–530, Morgan Kaufmann Publishers Inc., Burlington, MA, USA, 1993.
- [40] R.-I. Chang, H.-M. Hsu, S.-Y. Lin, C.-C. Chang, and J.-M. Ho, “Query-based learning for dynamic particle swarm optimization,” *IEEE Access*, vol. 5, pp. 7648–7658, Apr 2017.
- [41] Z. Zhang and S. Qian, “Artificial immune system in dynamic environments solving time-varying non-linear constrained multi-objective problems,” *Soft Computing*, vol. 15, no. 7, pp. 1333–1349, 2011.
- [42] Y. Wang and B. Li, “Investigation of memory-based multi-objective optimization evolutionary algorithm in dynamic environment,” in *Proceedings of the 2009 IEEE Congress on Evolutionary Computation*, pp. 630–637, IEEE, Trondheim, Norway, May 2009.
- [43] R. Shang, L. Jiao, Y. Ren, L. Li, and L. Wang, “Quantum immune clonal coevolutionary algorithm for dynamic multiobjective optimization,” *Soft Computing*, vol. 18, no. 4, pp. 743–756, 2014.
- [44] Z. Peng, J. Zheng, and J. Zou, “A population diversity maintaining strategy based on dynamic environment evolutionary model for dynamic multiobjective optimization,” in *Proceedings of the IEEE Congress Evolutionary Computation (CEC)*, pp. 274–281, Beijing, China, July 2014.
- [45] S. Yang, “Non-stationary problem optimization using the primal-dual genetic algorithm,” in *Proceedings of the 2003 Congress on Evolutionary Computation*, Canberra, Australia, December 2003.
- [46] Y. Jin and J. Branke, “Evolutionary optimization in uncertain environments—a survey,” *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 3, pp. 303–317, 2005.
- [47] J. Branke, “Memory enhanced evolutionary algorithms for changing optimization problems,” in *Proceedings of the 1999 Congress on Evolutionary Computation*, Washington, DC, USA, July 1999.
- [48] N. Mori, H. Kita, and Y. Nishikawa, “Adaptation to a changing environment by means of the feedback thermodynamical genetic algorithm,” in *Parallel Problem Solving from Nature—PPSN V* Springer, Berlin, Germany, 1998.
- [49] I. Hatzakis and D. Wallace, “Dynamic multi-objective optimization with evolutionary algorithms: a forward-looking approach,” in *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, pp. 1201–1208, ACM, New York, NY, USA, 2006.
- [50] A. Zhou, Y. Jin, Q. Zhang et al., “Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization,” *International Conference on Evolutionary Multi-Criterion Optimization*, pp. 832–846, Springer, Berlin, Germany, 2007.
- [51] 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, no. 2, pp. 87–110, 2010.
- [52] J. Zou, Q. Li, S. Yang, H. Bai, and J. Zheng, “A prediction strategy based on center points and knee points for evolutionary dynamic multi-objective optimization,” *Applied Soft Computing*, vol. 61, pp. 806–818, 2017.
- [53] A. Muruganantham, Y. Zhao, S. B. Gee et al., “Dynamic multiobjective optimization using evolutionary algorithm with kalman filter,” *Procedia Computer Science*, vol. 24, no. C, pp. 66–75, 2013.
- [54] Q. Lin, G. Jin, Y. Ma et al., “A diversity-enhanced resource allocation strategy for decomposition-based multiobjective evolutionary algorithm,” *IEEE Transactions on Cybernetics*, vol. 48, no. 8, pp. 2388–2401, 2018.
- [55] 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.
- [56] G. Welch and G. Bishop, “An introduction to the Kalman filter,” Tech. Rep. 95-041, University of North Carolina, Chapel Hill, NC, USA, 1995.
- [57] S. Jiang, S. Yang, X. Yao et al., *Benchmark Problems for IEEE CEC 2018 Competition on Dynamic Multiobjective Optimization*, IEEE, Piscataway, NJ, USA, 2018.
- [58] S. Jiang and S. Yang, “Evolutionary dynamic multiobjective optimization: benchmarks and algorithm comparisons,” *IEEE Transactions on Cybernetics*, vol. 47, no. 1, pp. 198–211, 2017.
- [59] M. C’amara, J. Ortega, and F. de Toro, “Performance measures for dynamic multiobjective optimization,” in *Bio-Inspired Systems: Computational and Ambient Intelligence*, vol. 5517, pp. 760–767, Springer, Berlin, Germany, 2009.
- [60] P. A. N. Bosman and D. Thierens, “The balance between proximity and diversity in multiobjective evolutionary algorithms,” *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 174–188, 2003.
- [61] E. Zitzler, *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*, Shaker Verlag, Herzogenrath, Germany, 1999.
- [62] K. Li, K. Deb, Q. Zhang, and S. Kwong, “An evolutionary many-objective optimization algorithm based on dominance and decomposition,” *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 694–716, 2015.
- [63] H. Li and Q. Zhang, “Comparison between NSGA-II and MOEA/D on a set of multiobjective optimization problems with complicated Pareto sets,” *IEEE Trans. Evol. Comput.* vol. 13, no. 2, pp. 284–302, 2009.
- [64] K. Deb and M. Goyal, “A combined genetic adaptive search (GeneAS) for engineering design,” *Computer Science and Informatics*, vol. 26, no. 4, pp. 30–45, 1996.
- [65] K. Deb and R. B. Agrawal, “Simulated binary crossover for continuous search space,” *Complex Systems*, vol. 9, no. 4, pp. 115–148, 1995.
- [66] F. Wilcoxon, “Individual comparisons by ranking methods,” *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.