Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 814210, 18 pages
http://dx.doi.org/10.1155/2015/814210

Research Article

The Evolutionary Algorithm to Find Robust Pareto-Optimal Solutions over Time

Meirong Chen,[1,2] Yinan Guo,[1] Haiyuan Liu,[1,2] and Chun Wang[1]

[1] *School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou 221116, China*
[2] *Science College, China University of Mining and Technology, Xuzhou 221116, China*

Correspondence should be addressed to Yinan Guo; guoyinan@cumt.edu.cn

Received 7 March 2014; Revised 28 May 2014; Accepted 17 August 2014

Academic Editor: Yun Li

In dynamic multiobjective optimization problems, the environmental parameters change over time, which makes the true pareto fronts shifted. So far, most works of research on dynamic multiobjective optimization methods have concentrated on detecting the changed environment and triggering the population based optimization methods so as to track the moving pareto fronts over time. Yet, in many real-world applications, it is not necessary to find the optimal nondominant solutions in each dynamic environment. To solve this weakness, a novel method called robust pareto-optimal solution over time is proposed. It is in fact to replace the optimal pareto front at each time-varying moment with the series of robust pareto-optimal solutions. This means that each robust solution can fit for more than one time-varying moment. Two metrics, including the average survival time and average robust generational distance, are present to measure the robustness of the robust pareto solution set. Another contribution is to construct the algorithm framework searching for robust pareto-optimal solutions over time based on the survival time. Experimental results indicate that this definition is a more practical and time-saving method of addressing dynamic multiobjective optimization problems changing over time.

1. Introduction

In many practical fields, such as engineering design, scientific computing, social economy, and network communication, there exist a large number of complex optimization problems. Particularly, many optimization problems contain multiple objective functions and dynamic parameters that make the objective functions change over time. Moreover, the number of objective functions and constraints may also vary from time to time. We call multiobjective optimization problems with above uncertain factors as dynamic multiobjective optimization problems (DMOPs). In this paper, we focus on the DMOPs with continuously changed dynamic parameters. Suppose $F(\vec{x},\vec{\alpha}_t) = \{f_1(\vec{x},\vec{\alpha}_t), f_2(\vec{x},\vec{\alpha}_t), \ldots, f_M(\vec{x},\vec{\alpha}_t)\}$ is the objective vector in DMOPs and $\vec{\alpha}_t$ is the dynamic parameters depending on t. The aim of DMOPs is to find the pareto front (PF^t) approximating to the true pareto front of $F(\vec{x},\vec{\alpha}_t)$ as soon as possible for all of the dynamic environment.

Aiming at tracking the moving true pareto fronts over time, dynamic multiobjective evolutionary optimization algorithms (DMOEA) were proposed [1–12]. The universal framework of DMOEA is presented in Algorithm 1. First of all, we need to accurately judge whether the environment has changed. It is the basic premise of using evolutionary optimization methods to respond to the environmental changes. The most common change-detection approach is to reevaluate the detectors. The detectors can be the current best solutions, a memory-based subpopulation, or a feasible subpopulation [5].

Once the various environment parameters happened, the new evolutionary optimization process was triggered. Many evolutionary algorithms with good performances on the static multiobjective optimization problems have been introduced into the DMOPs one after another, such as genetic algorithm [6], particle swarm algorithm [7], differential evolution algorithm [8], quantum immune clonal coevolutionary algorithm [9], and memetic algorithm [10]. As we know, the goal of the static optimization problems is to make the population gradually converge to optimal nondominant solution. The diversity of the population becomes weakened

```
Set $t = 0$;
Initialize a population $P_t$;
**Repeat**
    Detect the change of the environment;
    **if** the environment varies **then**
        Set $t = 0$;
        Reinitialize the population $P_t$;
    **end**
    Excute the evolutionary operations;
    $t = t + 1$;
    **Until** termination criteria met
```

Algorithm 1: DMOEA (the universal framework).

during the optimization methods. How to improve and maintain the evolutionary algorithm's ability to adapt to the various environment is a major challenge faced in the dynamic evolutionary computation. In recent years, various methods have been used to improve the diversity of the population. Chen et al. [11] extended additional objectives to deal with the DMOPs. Individual diversity is used as an additional objective to provide the historical information. Zhou et al. [12] proposed a population prediction strategy to improve the DMOEA's performance when the new environment is detected. The reinitial population was formed by a center point and a manifold. The track of center points was preserved to train autoregressive model so as to predict the center point in the new environment. A new coevolutionary paradigm [2] combining competition with cooperation was proposed to track the true pareto front in dynamic environment.

The conventional methods are mostly to trigger the multiobjective optimization process after detecting the change and then finish the evolution process as the following new environment occurs. Yet, this is impractical in many real-world optimization problems due to the following reasons. These methods are not suitable for rapidly changing environments, in which the environmental parameters vary quickly or frequently. Moreover, it is difficult to find the satisfied pareto fronts before detecting the change of fitness landscape because the methods are extremely time-consuming.

To address the above concern, several methods to find robust pareto front of multiobjective optimization problems with noise were presented by [13–17]. The emphasis is to seek an insensitive robust pareto front instead of the global optimal pareto front. The detailed definition about the robustness is illustrated in Section 2. For dynamic scalar optimization problems, robust optimization over time (ROOT) has been defined clearly by Yu et al. [18]. The task for ROOT is to find a sequence of robust solutions over time intervals. They have acceptable qualities and are relatively insensitive to the dynamic environment. By ROOT, the uncertainties in the parameter space and their cumulative effect on objective space are considered simultaneously. Furthermore, Jin et al. [19] gave a framework of ROOT which consists of a population-based optimization algorithm, the database, a fitness approximator, and a fitness predictor. A solution's robustness over time is estimated by both its past and its future performances. Subsequently, Fu et al. [20] provided a feasible algorithm in order to find robust optimal solutions over time and gave a detailed definition of the survival time and the average fitness. The robust solutions are expected to have longer survival time or larger average fitness. Though ROOT is easy to be realized and computed, it only fits for dynamic scalar optimization problems.

For DMOPs, the detailed robust definition over time is still an open issue. There were seldom relevant literatures about this area. We will introduce this idea about ROOT into DMOPs. The main contribution in this paper is to propose a novel concept on robust pareto-optima over time (RPOOT) to DMOPs that search for robust pareto-optimal solution set for all dynamic environments. Subsequently, the population-based multiobjective evolutionary algorithm is introduced to find RPOOT in terms of the nondomination solutions' robust performance. The robustness is measured by the survival time derived from the robust index given by Deb and Gupta [14]. We believe it is a more practical way of addressing continuously changed DMOPs.

The remainder of the paper is structured as follows. Section 2 presents a brief introduction to research on existing robust optimization methods and analyzes the existing problems in detail. Section 3 presents a class of DMOPs and describes formally the definition of RPOOT. In Section 4, the robustness and the performance metrics are defined for RPOOT. Furthermore, a population-based evolutionary algorithm to find robust pareto-optimal solutions over time is presented. Section 5 provides a brief overview of the existing benchmark functions for DMOPs and the experimental result for RPOOT. Conclusions and future work are presented in Section 6.

2. Related Works

In order to solve multiobjective optimization problems with uncontrollable variations, Li et al. [13] presented a robust multiobjective genetic algorithm by considering two objective functions: the fitness value f_v, which measures a solution's performance by a combined objective, and the robustness index η. They investigated the trade-off between the convergence and robustness of the nondominant solutions. Furthermore, Li proposed the Outer-Inner optimization structure. The outer subproblem was to simultaneously minimize the fitness value and to maximize the robustness index. The inner subproblem calculates the radius R, which represents a solution's robustness.

Deb and Gupta [14] extended an existing approach that finds robust solutions for single-objective optimization problems to MOPs with dynamic parameters. They defined the mean effective objective functions instead of the original objective functions.

Consider a multiobjective optimization problem as follows:

$$\begin{aligned} \min \quad & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})) \\ \text{subject to} \quad & \mathbf{x} \in S. \end{aligned} \tag{1}$$

In order to avoid obtaining the global optimal solutions which are quite sensitive to such variable perturbation in their vicinity, the following two approaches are defined for robust optimization by Deb and Gupta [14].

Definition 1 (multiobjective robust solution of type I (MORS1)). A solution $\mathbf{x}^*$ is called a multiobjective robust solution of type I if it is the global feasible pareto-optimal solution to the following multiobjective minimization problem (suppose a δ-neighborhood of a solution $\mathbf{x}$ is $B_\delta(\mathbf{x})$):

$$\begin{aligned} &\text{Minimize} \quad \mathbf{F}^{\text{eff}}(\mathbf{x}) = \left(f_1^{\text{eff}}(\mathbf{x}), f_2^{\text{eff}}(\mathbf{x}), \ldots, f_M^{\text{eff}}(\mathbf{x})\right) \\ &\text{subject to} \quad \mathbf{x} \in S, \end{aligned} \tag{2}$$

where $f_j^{\text{eff}}(\mathbf{x})$ is defined as follows:

$$f_j^{\text{eff}}(\mathbf{x}) = \frac{1}{\left|B_\delta(\vec{x})\right|} \int_{\mathbf{y}\in B_\delta(\mathbf{x})} f_j(\mathbf{y})\, d\mathbf{y}. \tag{3}$$

Definition 2 (multiobjective robust solution of type II (MORS2)). A solution $\mathbf{x}^*$ is called a multiobjective robust solution of type II if it is the global feasible pareto-optimal solution to the following multiobjective minimization problem:

$$\begin{aligned} &\text{Minimize} \quad \mathbf{F}(\mathbf{x}) = \left(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})\right), \\ &\text{subject to} \quad \frac{\left\|\mathbf{F}^{\text{eff}}(\mathbf{x}) - \mathbf{F}(\mathbf{x})\right\|}{\|\mathbf{F}(\mathbf{x})\|} \le \eta, \\ &\qquad\qquad\quad \mathbf{x} \in S. \end{aligned} \tag{4}$$

MORS1 replaces the original objective function $\mathbf{F}(\mathbf{x})$ with the effective objective function $\mathbf{F}^{\text{eff}}(\mathbf{x})$. Where $\mathbf{f}_i^{\text{eff}}(\mathbf{x})$ is the mean of ith objective function values in the vicinity of $\mathbf{x}$. In Definition 2, the original objective functions need to be optimized. At the same time, the feasible solutions must satisfy the constraint; that is, the objective function values among neighboring solutions are limited to a user-defined threshold η. Subsequently, the solution's robustness is judged. Two kinds of performances can not be analyzed at the same time. Both definitions for robustness took the variable perturbation into account. However, this is not DMOPs in fact.

Furthermore, Barrico and Antunes [15] defined the degree of robustness based on the solutions' behavior in their neighborhood in the decision space. The degree of robustness was also used to evaluate the solutions' behavior in the neighborhood of the reference scenario in the space of the objective functions' coefficients [16]. The weakness of above definition about the degree of robust is that the pareto-optimal set has to be known in advance. To solve this problem, Cromvik et al. [17] put forward a new definition for robustness index and introduced the utility function to convert multiobjective optimization problem into an approximation for a single decision maker. In this method, the robust index cannot be used as an objective during the optimization.

In a word, the above methods can solve the multiobjective optimization problems with perturbation in the decision space or the space of the objective function coefficients. However, only a robust pareto front meeting all multiobjective optimization problems with the disturbance is found. In this paper, we will discuss the robust solution set to a class of DMOPs with changing parameters.

3. The Definition of RPOOT

In this paper, we focus on the DMOPs, in which the environmental parameters continuously change over time and keep stationary between two time-varying moments. Obviously, the true pareto fronts shift from time to time. In other words, the objective functions depending on t are deterministic during each changing stage. Hence, this kind of DMOPs with continuous parameters can be discretized into a series of multiobjective optimization problems (MOPs) at each time-varying moment. Namely, the pareto fronts at each time-varying moment are regarded as the basis of optimization.

Without loss of generality, dynamic multiobjective optimization problem is defined as follows:

$$\begin{aligned} \min \quad & F(\vec{x}, \vec{\alpha}_t) = \{f_1(\vec{x}, \vec{\alpha}_t), \\ & \qquad f_2(\vec{x}, \vec{\alpha}_t), \ldots, f_M(\vec{x}, \vec{\alpha}_t)\} \\ \text{subject to} \quad & \vec{x} \in S, \end{aligned} \tag{5}$$

where $\vec{x}$ stands for the decision variable vector, $S \subset R^N$ is the decision space, and R^M is the objective space. $F : (S, t) \rightarrow R^M$ consists of M objective functions $f_i(\vec{x}, t)$ $(i = 1, 2, \ldots, M)$. $\vec{\alpha}_t$ is the time-depending parameter vector. $\bar{t} \in [0, t_{\text{end}}]$ represents the time. Suppose $1/T$ is the frequency for dynamic environments. There may be $N = \lfloor t_{\text{end}}/T \rfloor$ different time steps. DMOPs are divided into N MOPs, denoted by $\langle F(\vec{x}, \vec{\alpha}_1), F(\vec{x}, \vec{\alpha}_2), \ldots, F(\vec{x}, \vec{\alpha}_N)\rangle$.

Based on the definition of robust solutions' index given by Deb and Gupta [14] and the description of ROOT presented by Yu et al. [18], we propose a novel concept, called robust pareto-optimal over time (RPOOT) for DMOPs. It consists of a sequence of robust pareto-optimal sets, denoted by $\langle S_P^1, S_P^2, \ldots, S_P^l\rangle$ $(1 \le l \le N)$. Each robust pareto-optimal solution S_P^i, $i = 1, 2, \ldots, l$ may not approximate to the true pareto front at each period between changes but is relatively closer to all true pareto fronts during at least two consecutive time intervals.

Two hypotheses for RPOOT are that the environmental parameters change over time with stationary periods T between changes and that the robust solution fits for the consecutive changes during $\bar{t} \in [t_l, t_u] \subseteq [0, t_{\text{end}}]$. With $l_\alpha = \lceil t_l/T \rceil$ and $u_\alpha = \lceil t_u/T \rceil$, we obtain the problem

$$\left\langle F\left(\vec{x}, \vec{\alpha}_{l_\alpha}\right), F\left(\vec{x}, \vec{\alpha}_{l_\alpha+1}\right), \ldots, F\left(\vec{x}, \vec{\alpha}_{u_\alpha}\right)\right\rangle, \tag{6}$$

where $\vec{\alpha}_i = \vec{\alpha}_{i-1} + \Delta\vec{\alpha}$. The dynamics parameter $\Delta\vec{\alpha}$ is a random variable obeying a certain distribution such as a Gaussian distribution or a Uniform distribution. Let $P(\Delta\vec{\alpha})$ be the probability density function of $\Delta\vec{\alpha}$. The nondominant solutions' performances are measured by the following two definitions.

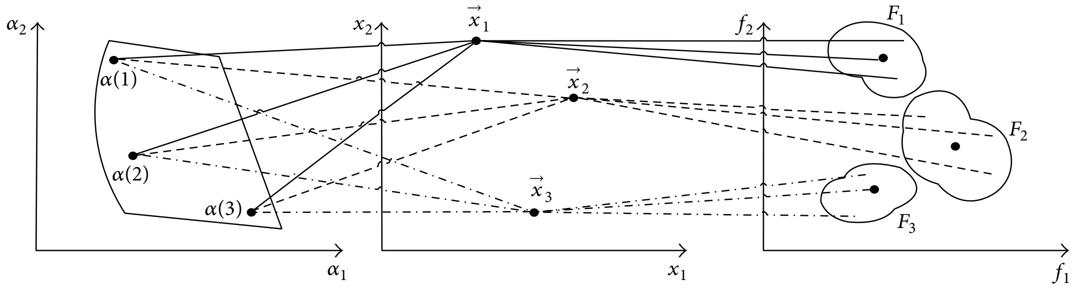


Figure 1: The relationship of neighborhoods among robust solutions in objective space (for 2-dimension spaces and all functions to be minimized).

Definition 3 (the average fitness). Consider

$$F\left(\vec{x};l_\alpha,u_\alpha\right)=\left(f_1\left(\vec{x};l_\alpha,u_\alpha\right),\right.$$
$$\left.f_2\left(\vec{x};l_\alpha,u_\alpha\right),\ldots,f_M^{\text{ave}}\left(\vec{x};l_\alpha,u_\alpha\right)\right),$$
$$f_i\left(\vec{x};l_\alpha,u_\alpha\right)$$
$$=\frac{1}{u_\alpha-l_\alpha+1}\sum_{j=1}^{u_\alpha-l_\alpha+1}\int f_i\left(\vec{x},\vec{\alpha}_{l_\alpha}+\left(j-1\right)\Delta\vec{\alpha}\right)p\left(\Delta\vec{\alpha}\right)d\Delta\vec{\alpha}. \tag{7}$$

Definition 4 (the variance of fitness). Consider

$$D\left(\vec{x};l_\alpha,u_\alpha\right)=\left(\delta_1\left(\vec{x};l_\alpha,u_\alpha\right),\right.$$
$$\left.\delta_2\left(\vec{x};l_\alpha,u_\alpha\right),\ldots,\delta_M\left(\vec{x};l_\alpha,u_\alpha\right)\right),$$
$$\delta_i\left(\vec{x};l_\alpha,u_\alpha\right)=\frac{1}{u_\alpha-l_\alpha+1}$$
$$\cdot\sum_{j=1}^{u_\alpha-l_\alpha+1}\int\left(f_i\left(\vec{x};\vec{\alpha}_{l_\alpha}+\left(j-1\right)\Delta\vec{\alpha}\right)\right.$$
$$\left.-f_i\left(\vec{x};l_\alpha,u_\alpha\right)\right)^2\cdot p\left(\Delta\vec{\alpha}\right)d\Delta\vec{\alpha}. \tag{8}$$

In essence, F measures the average performance of each objective function within the time interval $[t_l,t_u]$. D is the degree of the performance influenced by changing the time-varying environment.

Definition 5 (robustness). A solution $\vec{x}_i(t)$ is called a robust pareto solution if it is nondominant individuals satisfied to the following DMOPs:

$$\begin{aligned}\text{Minimize}\quad & F\left(\vec{x};l_\alpha,u_\alpha\right)=\left(f_1\left(\vec{x};l_\alpha,u_\alpha\right),f_2\left(\vec{x};l_\alpha,u_\alpha\right),\ldots,\right.\\ & \left.f_M\left(\vec{x};l_\alpha,u_\alpha\right)\right)\\ \text{subject to}\quad & \max_{1\le j\le M}\delta_j\left(\vec{x}_i;l_\alpha,u_\alpha\right)<\eta.\end{aligned} \tag{9}$$

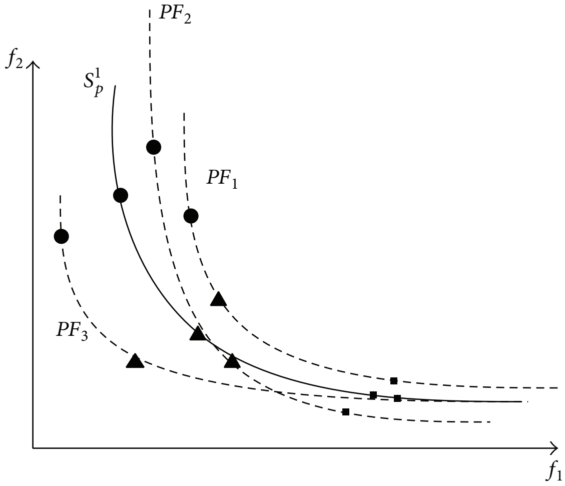


Figure 2: The definition of robust pareto-optimal set in objective space.

In fact, all solutions composed of robust pareto-optimal solution sets must be compromise. Subsequently, the non-dominant relationship is defined based on the expected fitness vector over time. Define $\vec{x}_i(t)\succ\vec{x}_j(t)$ if $\forall l$, $f_l(\vec{x}_i;l_\alpha,u_\alpha)\le f_l(\vec{x}_j;l_\alpha,u_\alpha)$ and $\exists k$, $f_k(\vec{x}_i;l_\alpha,u_\alpha)<f_k(\vec{x}_j;l_\alpha,u_\alpha)$ is satisfied. $\vec{x}_i(t)$ and $\vec{x}_j(t)$ are compromise, if $\exists l$, $f_l(\vec{x}_i;l_\alpha,u_\alpha)>f_l(\vec{x}_j;l_\alpha,u_\alpha)$ and $\forall k\ne l$, $f_k(\vec{x}_i;l_\alpha,u_\alpha)\le f_k(\vec{x}_j;l_\alpha,u_\alpha)$. Consequently, we define robust pareto-optimal solutions as $S_P^t=\{\vec{x}^*(t)\mid\neg\exists\vec{x}_i(t)\succ\vec{x}^*(t))$ and $\max_{1\le j\le M}\delta_j(\vec{x}_i;l_\alpha,u_\alpha)<\eta\}$. Obviously, RPOOT takes both the approximation and the robustness into account.

In Figure 1, the parameters vary in different periods, which map the uncertain area in the objective space. Obviously, $\vec{x}_3(t)\succ\vec{x}_2(t)$ and $\vec{x}_3(t)\succ\vec{x}_1(t)$. $\vec{x}_1(t)$ and $\vec{x}_2(t)$ are nondominated. However, $\vec{x}_2(t)$ is not robust due to its large variance. As shown in Figure 2, the true pareto-optimal sets for three consecutive time-varying periods are PF_1, PF_2, PF_3 plotted by the dash lines. S_p^1 plotted by real line in Figure 2 is the robust pareto-optimal solution. Obviously, S_p^1 is not the best pareto-optimal set during this period but the satisfied pareto front over time.

4. The Robust Solutions' Performance

In this paper, a robust pareto-optimal solution set needs not only to approximate to the true pareto front during each stage to the largest extent but also to satisfy the requirements of robustness. We would like to make a clear distinction between the definition of robustness for solutions and the approximation of each solution.

4.1. The Robustness. For DMOPs, the task of the traditional optimization methods is to find the pareto-optimal solutions after detecting the new environment. This is time-consuming. If the pareto-optimal front can meet the requirements of more than one kind of environment in a certain accuracy, the cost for search will be less. Consequently, the robustness has two means. One is the insensitivity to the fluctuation parameters. Deb and Gupta [14] has presented a general definition based on δ-neighborhood perturbation. The other is the survival time, which reflects how many consecutive changed environments this solution can fit for. Based on the above two aspects, corresponding metric called the survival time is proposed to measure the robustness in DMOPs.

Suppose $\vec{x}(t)$ is a nondomination solution at time t. The robustness of $\vec{x}(t)$ is defined by maximum survival time Lstarting from time t when all fitness values of $\vec{x}(t)$ from t to $t+L$ belong to η-neighborhood of $F(\vec{x}(t), \alpha_t)$.

Define the survival time of solution's robustness as L;

$$L\left(S_P^t, \eta\right) = \min_{\vec{x}(t)\in S_P^t} L^s\left(\vec{x}(t), \eta\right), \tag{10}$$

$$\begin{aligned} &L^s\left(\vec{x}(t), \eta\right) \\ &= \max\left\{ l \,\middle|\, \frac{\left\|\widehat{F}\left(\vec{x}(t), \vec{\alpha}_{t+i}\right) - F\left(\vec{x}(t), \vec{\alpha}_t\right)\right\|}{\left\|F\left(\vec{x}(t), \vec{\alpha}_t\right)\right\|} \leq \eta, \forall i, 1 \leq i \leq l \right\}. \end{aligned} \tag{11}$$

$L^s(\vec{x}(t), \eta)$ stands for the survival time of any optimal solution $\vec{x}(t)$ in robust pareto-optimal set S_P^i. η is the user-defined threshold, which is the key parameter having a direct impact on L. The larger η means the tolerance to the varieties of the fitness is better. It makes L larger. $\|\cdot\|$ operator is used to measure the distance between the current fitness values and the future predictive fitness values. Here, the Euclidean norm is used. But any other suitable norm can also be adopted. $\widehat{F}(\vec{x}, \alpha_{t+i})$ is the approximated fitness instead of the real fitness value by a predictor [18]. Because the time-depending parameters vary randomly, each nondominated solution's real fitness values for each time-varying environment cannot be all evaluated accurately. So we need a prediction method to approximate the fitness in the future dynamic environments. As shown in Figure 3, if the robust pareto-optimal solutions at time t are satisfied during the consecutive time-varying moments from t to $t+L$, the fitnesses $F(\vec{x}, \vec{\alpha}_{t+1}), F(\vec{x}, \vec{\alpha}_{t+2}), \ldots, F(\vec{x}, \vec{\alpha}_{t+L})$ are all constricted to the η-neighborhood of $F(\vec{x}, \vec{\alpha}_t)$.

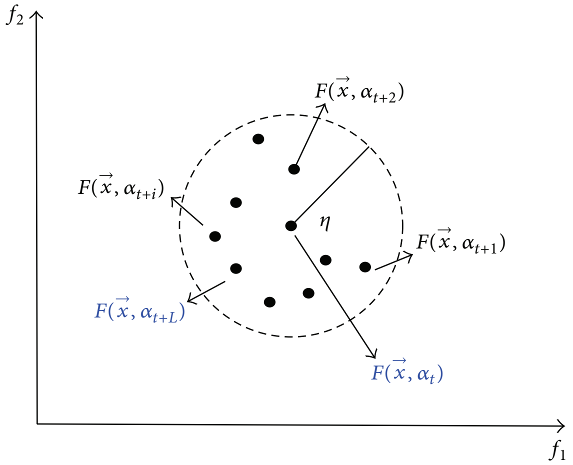

f_2
$F(\vec{x}, \alpha_{t+2})$
$F(\vec{x}, \alpha_{t+i})$
η
$F(\vec{x}, \alpha_{t+1})$
$F(\vec{x}, \alpha_{t+L})$
$F(\vec{x}, \alpha_t)$
f_1

Figure 3: The definition of robust survival time in the neighborhood.

4.2. The Average Fitness Function. Generally, the integral part of (7) is not easy to be calculated since there is little knowledge to get the accurate probability density function. So we take the following average performance as the robustness of solution over the considered time interval.

We define S_P^t as a robust pareto-optimal solution over time if it is the global feasible pareto-optimal solution to the following multiobjective optimization problem:

$$\begin{aligned} \min\ F^{\text{ave}}\left(\vec{x}, \vec{\alpha}_t\right) &= \left\{ f_1^{\text{ave}}\left(\vec{x}, \vec{\alpha}_t\right), \right. \\ &\qquad \left. f_2^{\text{ave}}\left(\vec{x}, \vec{\alpha}_t\right), \ldots, f_M^{\text{ave}}\left(\vec{x}, \vec{\alpha}_t\right) \right\} \\ f_i^{\text{ave}}\left(\vec{x}, \vec{\alpha}_t\right) &= \frac{1}{L+1} \sum_{j=0}^{L} \widehat{f}_i\left(\vec{x}, \vec{\alpha}_{(t+j)}\right), \end{aligned} \tag{12}$$

where $f_i^{\text{ave}}(\vec{x}, \vec{\alpha}_t)$ stands for the average fitness value of ith objective during the consecutive time-varying moments from t to $t+L$. Subsequently, $F^{\text{ave}}(\vec{x}, \vec{\alpha}_t)$ measures the average performance of S_P^t in each objective during the time interval L. The smaller average fitness value means that S_P^t is more approximate to the true pareto fronts of corresponding time-varying environments.

In a word, the average survival time reflects the robustness of S_P^t on the time space; that is, how many time-varying environments it fits for. The average fitness value measures how the pareto-optimal fronts approximate to the true pareto fronts in the objective space during these dynamic stages. Consequently, a robust pareto-optimal solution must be the one with the minimum average fitness values.

4.3. The Population-Based Optimization Method. Taking above metrics as the objectives, a novel framework to solve RPOOT problems is constructed for DMOPs. In the time-varying moments, many population-based evolutionary algorithms may be adopted to find a set of robust pareto-optimal fronts over time, denoted by $S_P^1, S_P^2, \ldots, S_P^l$.

It is worth noting that we must take into account the future performance of all objectives during the calculation of

```
Set $t = 1$;
Set evolutionary generation $g = 0$;
Initialize a population $P_t$;
Evaluate the fitness value and survival time $L$ of every individual in $P_t$;
Caculate the average fitness value $F^{\text{ave}}$ by formula (12);
**Repeat**
    $t_0 = t$;
    Detect change in the environment landspaces;
    **if** the environment changes **then**
        Set $t = t + 1$;
    **end**
    **if** $t = t_0 + L$
        Reinitialize the population $P_t$;
        Evaluate the fitness value and survival time $L$ of every individual in $P_t$;
        Caculate the average fitness value $F^{\text{ave}}$;
        Excute evolutionary operations;
    **else**
        Optimize the $t_0$th MOP by MOEA/D;
    **End**
        $g = g + 1$;
**Until** termination criteria met
```

Algorithm 2: EA for RPOOT (a framework finding RPOOT).

robust survival time and robustness performance. We assume that future performance can be estimated by a database and a predictor. The database is used to store historical data, and the task of the predictor is to estimate a solution's future performances [19]. A framework of population-based optimization algorithm for RPOOT is presented as following Algorithm 2. By this method, the robust pareto solution could not only approximate to the true pareto front as close as possible but also fit for more than one dynamic environment.

4.4. The Measurement of Algorithm Performance. It is important to measure the performance of the RPOOT algorithm. We should not only consider the robustness of the robust pareto-optimal solutions but also consider the accuracy of the solutions to the true pareto front. On the one hand, the average survival time measures the solutions' robustness on the time scale. On the other hand, the average robust metric measures the approximation to the true pareto front in the objective space.

4.4.1. The Average Survival Time. In DMOPs, the robustness of the robust optimal pareto solutions is measured by the average survival time. Moreover, the algorithms will be compared across the whole time period $t = 1, 2, \ldots, N$. Based on the robustness defined in (11), the robustness of the algorithm performance is defined as follows:

$$\overline{L} = \frac{1}{N}\sum_{t=1}^{N} L\left(S_p^t, \eta\right). \tag{13}$$

$L(S_p^t, \eta)$ is the robustness survival time of the nondomination solutions obtained from the algorithm during the timeline. Obviously, the longer the average survival time is, the better the robustness of solutions is. The robustness of optimal parato front on the time scale can be reflected by the average survival time. Moreover, L depends on the threshold η. Therefore, a more exhaustive analysis is necessary for the robust optimal pareto solutions under different η.

4.4.2. The Average Robust Generational Distance. This metric reflects the quality of the robust nondominant solution sets. The general distance (GD) [21] indicates how close the obtained PFs are to the true pareto front in multiobjective optimization problems. Furthermore, the inverted generational distance (IGD) [21] is used to assess the approximation performance of the algorithms. IGD measures both the diversity and the convergence of the population. In our experimental studies, robust generational distance (RGD) and robust inverted generational distance (RIGD) indicate the average distance between each robust optimal pareto front and the true pareto fronts within its survival time. Suppose l is the size of robust nondominant pareto solution set during the whole time interval. The RGD and RIGD metrics are defined as follows:

$$\begin{gathered} \mathrm{RGD}^{\mathrm{RPOOT}} = \frac{1}{l}\sum_{i=1}^{l} \mathrm{MGD}(i), \\ \mathrm{MGD}(i) = \max_{k=t_i,\ldots,t_i+L_{t_i}} \mathrm{GD}(k), \end{gathered} \tag{14}$$

$$\begin{gathered} \mathrm{RIGD}^{\mathrm{RPOOT}} = \frac{1}{l}\sum_{i=1}^{l} \mathrm{MIGD}(i), \\ \mathrm{MIGD}(i) = \max_{k=t_i,\ldots,t_i+L_{t_i}} \mathrm{IGD}(k), \end{gathered} \tag{15}$$

where t_i stands for the time step of robust nondominant solution set. L_{t_i} is their survival time. $\mathrm{GD}(k) = (1/|P^k|)\sum_{\nu\in P^k} d(PF^k,\nu))$ and $\mathrm{IGD}(k) = (1/|PF^k|)\sum_{\nu\in PF^k} d(\nu, P^k)$. PF^k is a set of uniformly distributed optimal solutions in the true PF at t; P^t is the solutions obtained at t. $d(PF^k,\nu) = \min_{u\in PF^k}\sqrt{\sum_{j=1}^{m}(f_j^{(u)} - f_j^{(\nu)})^2}$ is the distance between ν and PF^k. $d(\nu, P^k) = \min_{u\in P^k}\sqrt{\sum_{j=1}^{m}(f_j^{(\nu)} - f_j^{(u)})^2}$ is the distance between ν and P^k. $|P^k|$ and $|PF^k|$ are the cardinalities of P^k and PF^k. In our experiments, we select 100 evenly distributed solutions in PFs.

5. Analysis of the Experimental Results

In this section, eight dynamic multiobjective benchmark functions are adopted in the experiments. Simulation results and further analysis on solutions' performance are conducted in Section 5.2.

5.1. Benchmark Functions. Eight dynamic multiobjective benchmark functions are adopted to test whether or not the algorithm can find robust pareto-optimal solutions set. The first five functions are FDA1-FDA5 presented by Farina et al. [3]. The other three functions are DMOP1, DMOP2, and DMOP3 [22]. FDA4 and FDA5 contain three objectives, and the others include two objectives. τ is the generation counter. τ_t is the number of iterations under the time window t. n_t is the number of distinct steps under t, which controls the distance between two consecutive PSs.

The first type of benchmark are FDA1, FDA4, and DMOP3. For Type I problem, only the pareto sets (PSs) in the decision space dynamically change over time. However, the corresponding pareto fronts (PFs) in the objective space do not change with time. At any moment, the optimal pareto fronts are respectively $f_2 = 1 - \sqrt{f_1}$, $f_1^2 + f_2^2 + f_3^2 = 1$, and $f_2 = 1 - \sqrt{f_1}$. FDA2 and DMOP1 belong to Type III problems, in which only the PFs in the objective space change while the PSs in the decision space remain the same. The optimal pareto front of FDA2 is $f_2 = 1 - f_1^{(0.75+0.7\sin(0.5\pi t))^{-1}}$ that changes from a convex to a nonconvex shape. DMOP1 has a convex optimal pareto front $f_2 = 1 - f_1^{(1.25+0.75\sin(0.5\pi t))}$. FDA3, FDA5, and DMOP2 belonging to Type II problems have changing PSs and PFs: $f_2 = 1 - f_1^{(0.75+0.7\sin(0.5\pi t))^{-1}}$, $f_1^2 + f_2^2 + f_3^2 = (1 + G(t))^2$, $f_2 = 1 - f_1^{(1.25+0.75\sin(0.5\pi t))}$. The definitions of these dynamic multiobjective benchmark functions are summarized in Table 1. Their true PFs when $t = 5, 10, 20, 23, 26, 34$ are shown in Figure 4.

In this paper, we adopt a multiobjective evolutionary algorithm based on decomposition (MOEA/D) [21] to track the moving optimal pareto front over time. In MOEA/D, the penalty-based boundary intersection (PBI) approach is used as the surrogate model. For the benchmark functions, the population size is 100. In all experiments, time-varying moment t alters with the evolutionary generation and is associated with the parameter τ_t. The larger τ_t means that the environmental parameters change more infrequently and multiobjective optimization algorithms can spend more sufficient time tracking the new pareto front. Otherwise, the less τ_t makes multiobjective optimization algorithm triggered after detecting the new environment, hardly finding the satisfied pareto solutions closed to the true pareto front during the limited iterations. Now we discuss the algorithm performances under different frequencies. As shown in Figure 5, when the environment changes every 20 generations, the algorithm can track dynamic pareto-optimal fronts better. But within 5 generations, the satisfied PFs for the new environment are difficultly obtained. As a result, for all experiments, $\tau_t = 20$. The corresponding optimal pareto fronts of each benchmark function gotten by MOEA/D are shown below in Figure 6.

5.2. Simulation Results and Analysis. In this section, two groups of experiments have been done. In the first group, the optimal nondomination solutions starting from the time t_0 are obtained. If the minimum survival time of this moment is L, robust pareto solution fits for the environment from t_0 to $t_0 + L$. The process is repeated until the last environment occurs. The purpose of the second group of experiments is to obtain each robust optimal pareto front, respectively, for RPOOT at 100 time-varying moments.

5.2.1. The Effect of Neighborhood Size η. The neighborhood size η directly influences the evaluation criterion for the robust optimal pareto fronts. In the first group of experiments, we compare and analyze the performances of the robust optimal pareto fronts under different thresholds η. The numbers of robust pareto fronts (NRPFs) on 100 time-varying moments are listed in Table 2. The statistical average results of RGD and RIGD on eight benchmarks over 15 runs also can be found in Table 3 under different neighborhood sizes η.

As shown in Tables 2 and 3, with the increasing of the neighborhood size η, less robust pareto fronts are contained in the whole period. It means that the robustness of RPOOT is better. At the same time, their average inverted generational distances become larger, which means that the convergence of the RPOOT is worse. For each benchmark function, the standard errors of NRPF, RGD, and RIGD over 15 runs are given in Tables 2 and 3. The means and standard errors shown in these tables indicate that the stability of RPOOTs is good enough. We choose $\eta = 0.4$ in the following experiments. The robust optimal pareto fronts obtained from MOEA/D are shown in Figure 7.

From Figures 7(a)–7(h) and Table 4, we find that the GD and IGD of tracking multiobjective (TMO) algorithm are less than RGD and RIGD of RPOOT algorithm. But the number of robust pareto solution sets of RPOOT is far less than 100.

5.2.2. The Average Survival Time. The second group of experiments record each robust pareto front in RPOOT at each time-varying moment. Independent 15-time run is done for each benchmark. The results of the second group of experiments are plotted in Figure 8. It can be seen from Figure 8 that the average survival time of the robust pareto

Table 1: The dynamic benchmark functions.

Benchmark function	Definition	Type
FDA1	$f_1(X_{\mathrm{I}}) = x_1$ $f_2 = g \cdot h$ $g(X_{\mathrm{II}}) = 1 + \sum_{x_i \in X_{\mathrm{II}}} (x_i - G(t))^2$ $h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}}$ $G(t) = \sin(0.5\pi t), \quad t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_t} \right\rfloor$ where: $m = 10, X_{\mathrm{I}} = (x_1) \in [0,1]; \quad X_{\mathrm{I}} = (x_2, \ldots, x_m) \in [-1,1]$	Type I
FDA2	$f_1(X_{\mathrm{I}}) = x_1$ $f_2 = g \cdot h$ $g(X_{\mathrm{II}}) = 1 + \sum_{x_i \in X_{\mathrm{II}}} x_i^2$ $h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^{H(t)^{-1}}$ $H(t) = 0.75 + 0.7\sin(0.5\pi t), \quad t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_t} \right\rfloor$ where: $X_{\mathrm{I}} = (x_1) \in [0,1]; \quad X_{\mathrm{II}} \in [-1,1], \lvert X_{\mathrm{II}} \rvert = 15$	Type III
FDA3	$f_1(X_{\mathrm{I}}) = \sum_{x_i \in X_{\mathrm{I}}}^{n} x_i^{F(t)}$ $f_2 = g \cdot h$ $g(X_{\mathrm{II}}) = 1 + G(t) + \sum_{x_i \in X_{\mathrm{II}}} (x_i - G(t))^2$ $h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}}$ $G(t) = \lvert \sin(0.5\pi t) \rvert$ $F(t) = 10^{2\sin(0.5\pi t)}, \quad t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_t} \right\rfloor$ where: $\lvert X_{\mathrm{I}} \rvert = 1, \lvert X_{\mathrm{II}} \rvert = 9, X_{\mathrm{I}} \in [0,1]; X_{\mathrm{II}} \in [-1,1]$	Type II
FDA4	$f_1(x) = (1+g)\cos(0.5\pi x_1)\cos(0.5\pi x_2)$ $f_2(x) = (1+g)\cos(0.5\pi x_1)\sin(0.5\pi x_2)$ $f_3(x) = (1+g)\sin(0.5\pi x_1)$ $g(x) = \sum_{i=3}^{n} (x_i - G(t))^2$ $G(t) = \lvert \sin(0.5\pi t) \rvert, \quad t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_t} \right\rfloor$ where: $x_i \in [0,1], n = 12$	Type I
FDA5	$f_1(x) = (1+g)\cos(0.5\pi y_1)\cos(0.5\pi y_2)$ $f_2(x) = (1+g)\cos(0.5\pi y_1)\sin(0.5\pi y_2)$ $f_3(x) = (1+g)\sin(0.5\pi y_1)$ $g(x) = G(t) + \sum_{i=3}^{n} (x_i - G(t))^2, \quad y_i = x_i^{F(t)}$ $G(t) = \lvert \sin(0.5\pi t) \rvert, \quad F(t) = 1 + 100\sin^4(0.5\pi t), \quad t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_t} \right\rfloor$ where: $x_i \in [0,1], n = 12$	Type II
DMOP1	$f_1(x_1) = x_1$ $f_2 = g \cdot h$ $g(x_2, \ldots, x_m, t) = 1 + 9\sum_{i=2}^{m} x_i^2$ $h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^{H(t)}$ $H(t) = 0.75\sin(0.5\pi t) + 1.25, \quad t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_t} \right\rfloor$ where: $m = 10, \ x_i \in [0,1], \forall i = 1, 2, \ldots, m$	Type III

Table 1: Continued.

Benchmark function	Definition	Type
DMOP2	$f_1(x_1) = x_1$ $f_2 = g \cdot h$ $g(x_2, \ldots, x_m, t) = 1 + \sum_{i=2}^{m} (x_i - G(t))^2$ $h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^{H(t)}$ $H(t) = 0.75 \sin(0.5\pi t) + 1.25, \quad t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_t} \right\rfloor$ where: $m = 10, \ x_i \in [0,1], \forall i = 1, 2, \ldots, m$	Type II
DMOP3	$f_1(X) = x_1$ $f_2 = g \cdot h$ $g(X, t) = 1 + 9 \sum_{i=2}^{n} (x_i - G(t))^2$ $h(X, t) = 1 - \sqrt{\frac{f_1}{g}}$ $G(t) = \sin(0.5\pi t), \quad t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_t} \right\rfloor$ where: $X \in [0,1] \times [-1,2]^{n-1}, m = 10$	Type II

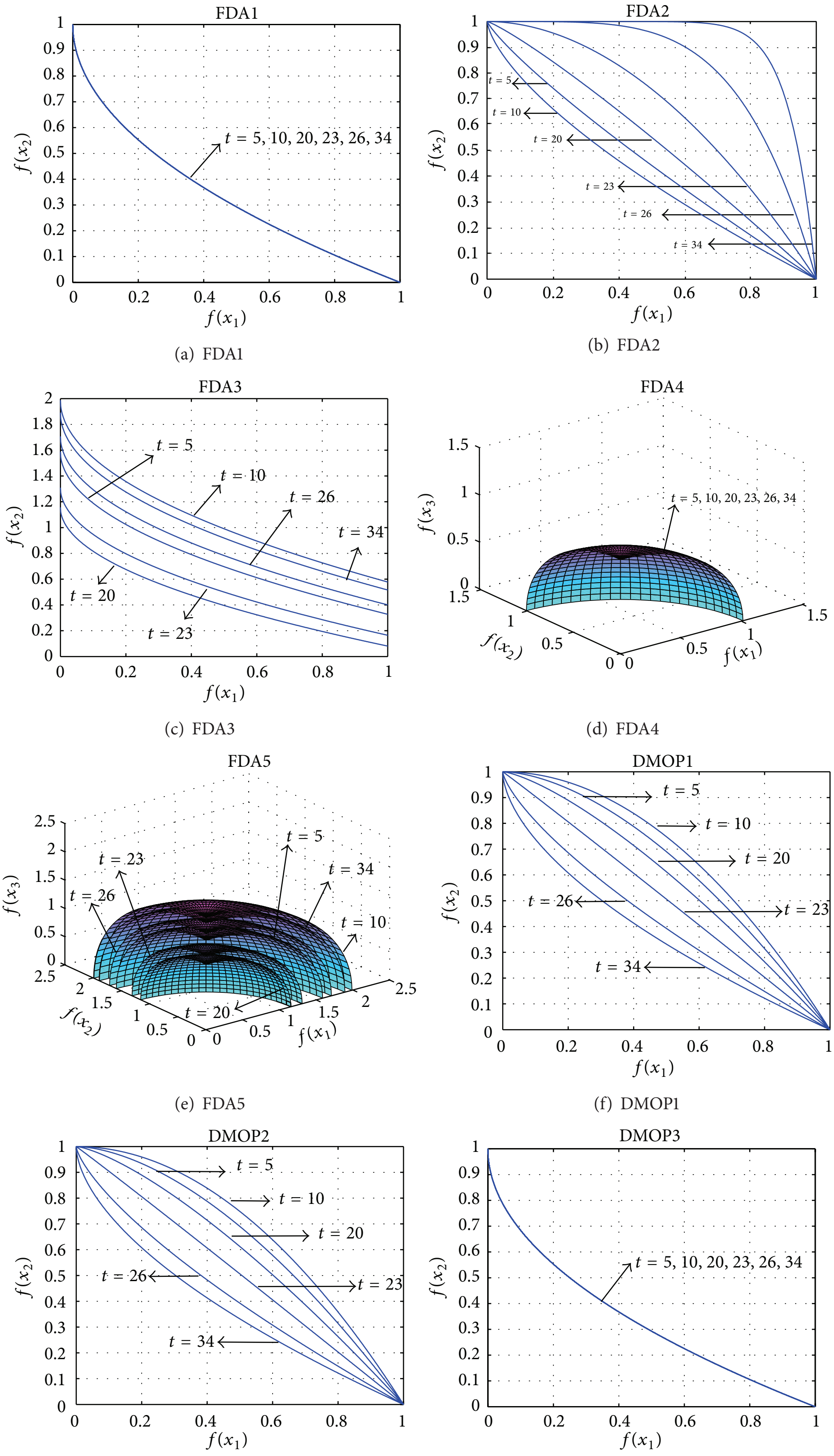
Table 2: Comparison of the number of robust pareto fronts (NRPFs) on 100 time-varying moments under different η.

Functions	Measures		η							
			$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.3$	$\eta = 0.4$	$\eta = 0.5$	$\eta = 0.6$	$\eta = 0.7$	$\eta = 0.8$
FDA1	NRPFs	μ	70.9	61	50	37.4	35.9	30	33.1	30
		σ	0.3162	0	0	0.5164	0.3162	0	0.7379	0
FDA2	NRPFs	μ	40	19	9	9.4	1	1.8	2	1.6
		σ	0	0	0	1.2649	0	0.4216	0	0.5164
FDA3	NRPFs	μ	83.4	69	64	51.2	45.5	35.5	34.7	30.8
		σ	0.5164	0	0	0.6325	1.5811	0.9718	0.4830	0.6325
FDA4	NRPFs	μ	79	69.5	56	42.4	36.1	33.4	28.9	29.9
		σ	0	1.5811	0	1.2649	1.4491	1.3499	0.3162	0.3162
FDA5	NRPFs	μ	90	90	70	58.5	50	61.7	45	35
		σ	0	0	0	1.0801	0	0.4830	0	0
DMOP1	NRPFs	μ	24.8	5.9	6	1	1	1	1	1
		σ	0.6325	1.4491	0	0	0	0	0	0
DMOP2	NRPFs	μ	76.4	68	54.5	39.3	32.3	32.8	29.2	28
		σ	1.2649	0	0.7071	1.1595	0.9487	0.6325	0.7888	0
DMOP3	NRPFs	μ	90	89.6	84.6	78.8	80	80	80	79.8
		σ	0	0.5164	0.6992	1.0328	0	0	0	0.6325

front at each time-varying moment is far larger than 1. The results of the average survival time listed in Table 5 are all far longer than changing time 100.

6. Conclusions

Dynamic multiobjective optimization problems with changing parameters widely exist in real life. The aim of the traditional optimization algorithms is to track the optimal pareto solution set after detecting the environment change efficiently. These algorithms may not obtain the satisfied nondominant solutions between two time-varying moments. In this paper, we proposed a new perspective for solving DMOPs with consecutive time-varying periods. Its goal is to find the robust pareto solution set over time. Three contributions are contained in RPOOT. At first, the detailed concept



(a) FDA1

(b) FDA2

(c) FDA3

(d) FDA4

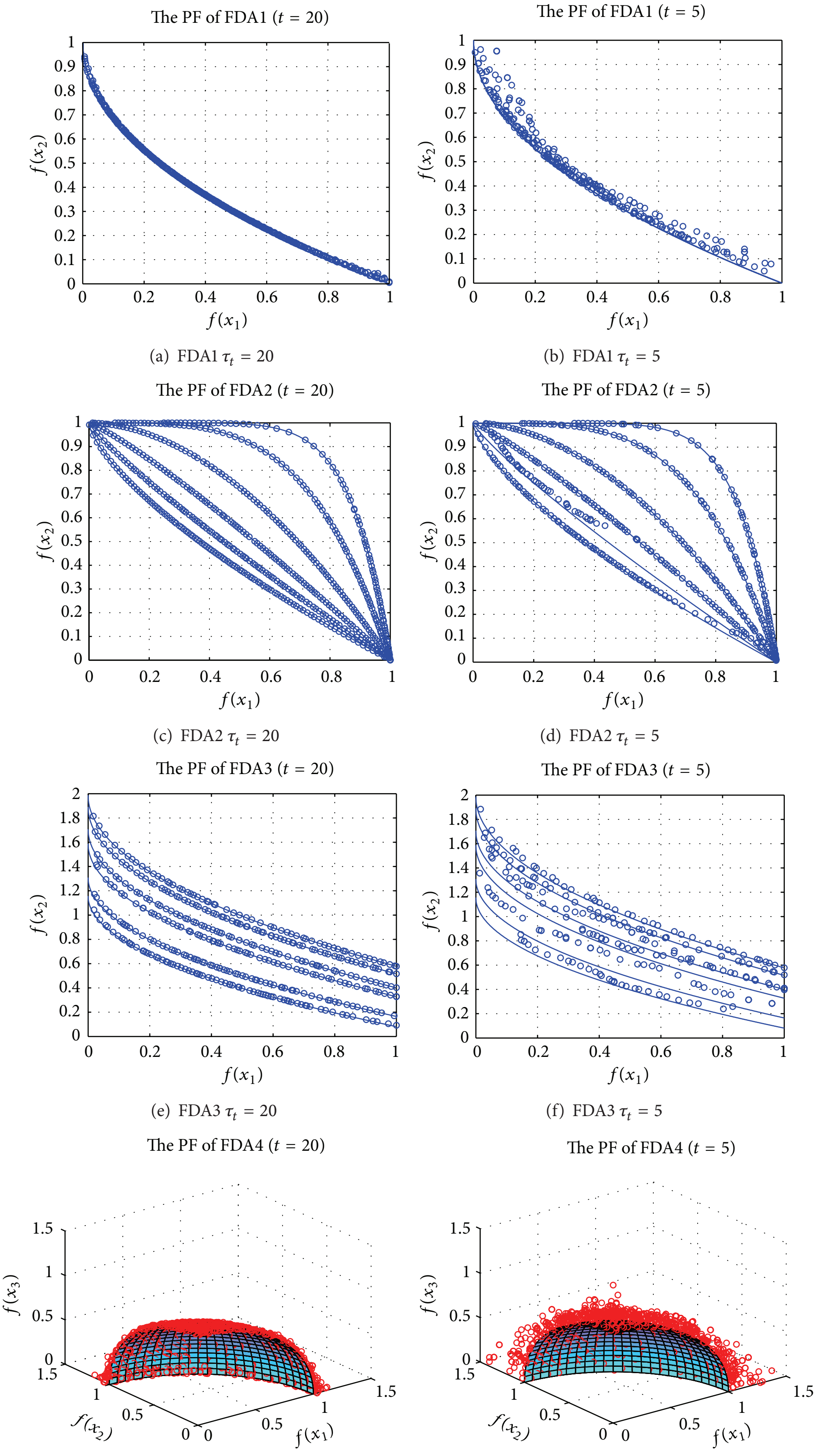
(e) FDA5

(f) DMOP1

(g) DMOP2

(h) DMOP3

Figure 4: The true pareto fronts of benchmark functions.



(a) FDA1 $\tau_t = 20$

(b) FDA1 $\tau_t = 5$

(c) FDA2 $\tau_t = 20$

(d) FDA2 $\tau_t = 5$

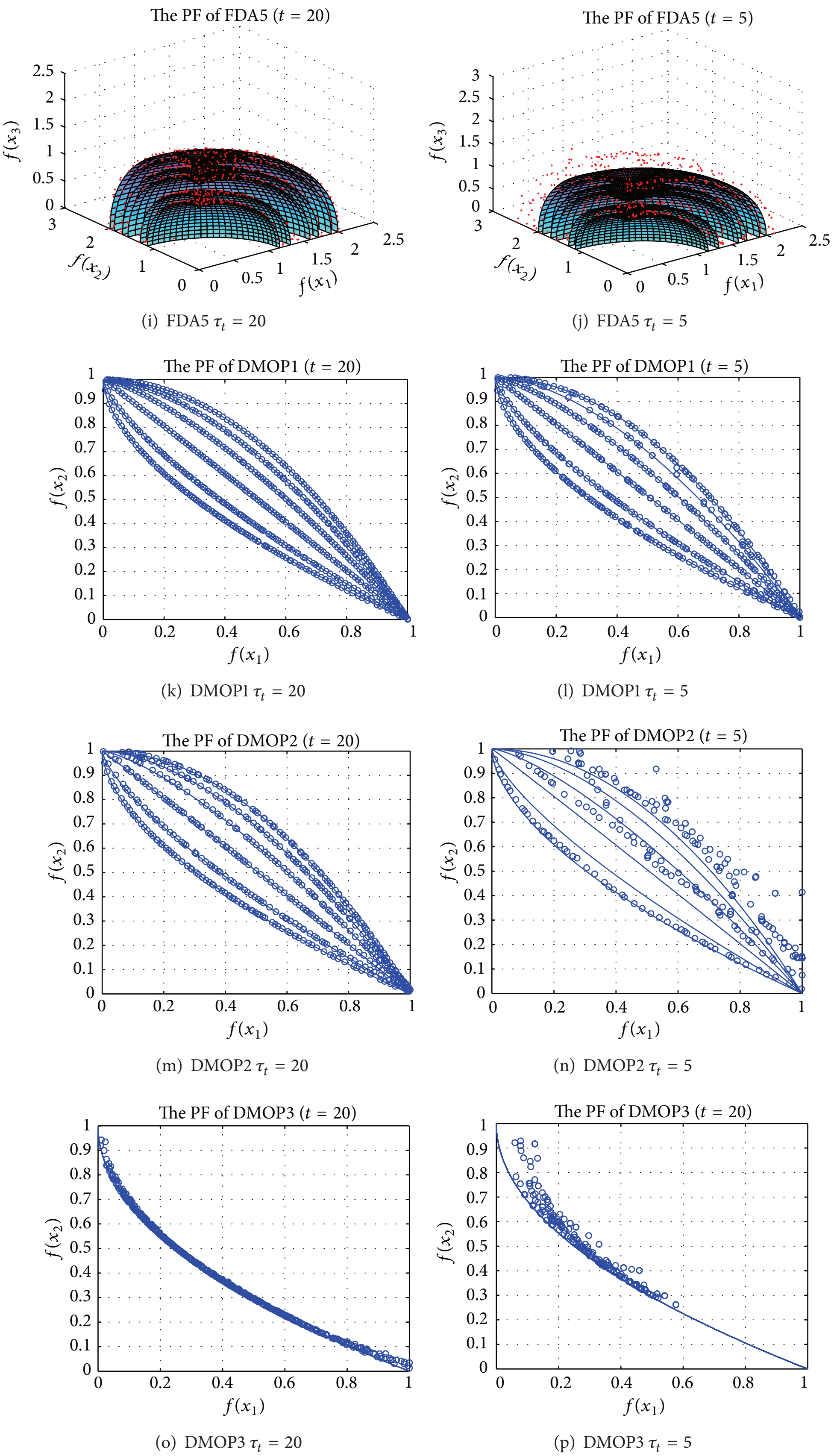
(e) FDA3 $\tau_t = 20$

(f) FDA3 $\tau_t = 5$

(g) FDA4 $\tau_t = 20$

(h) FDA4 $\tau_t = 5$

Figure 5: Continued.



(i) FDA5 τ_t = 20

(j) FDA5 τ_t = 5

(k) DMOP1 τ_t = 20

(l) DMOP1 τ_t = 5

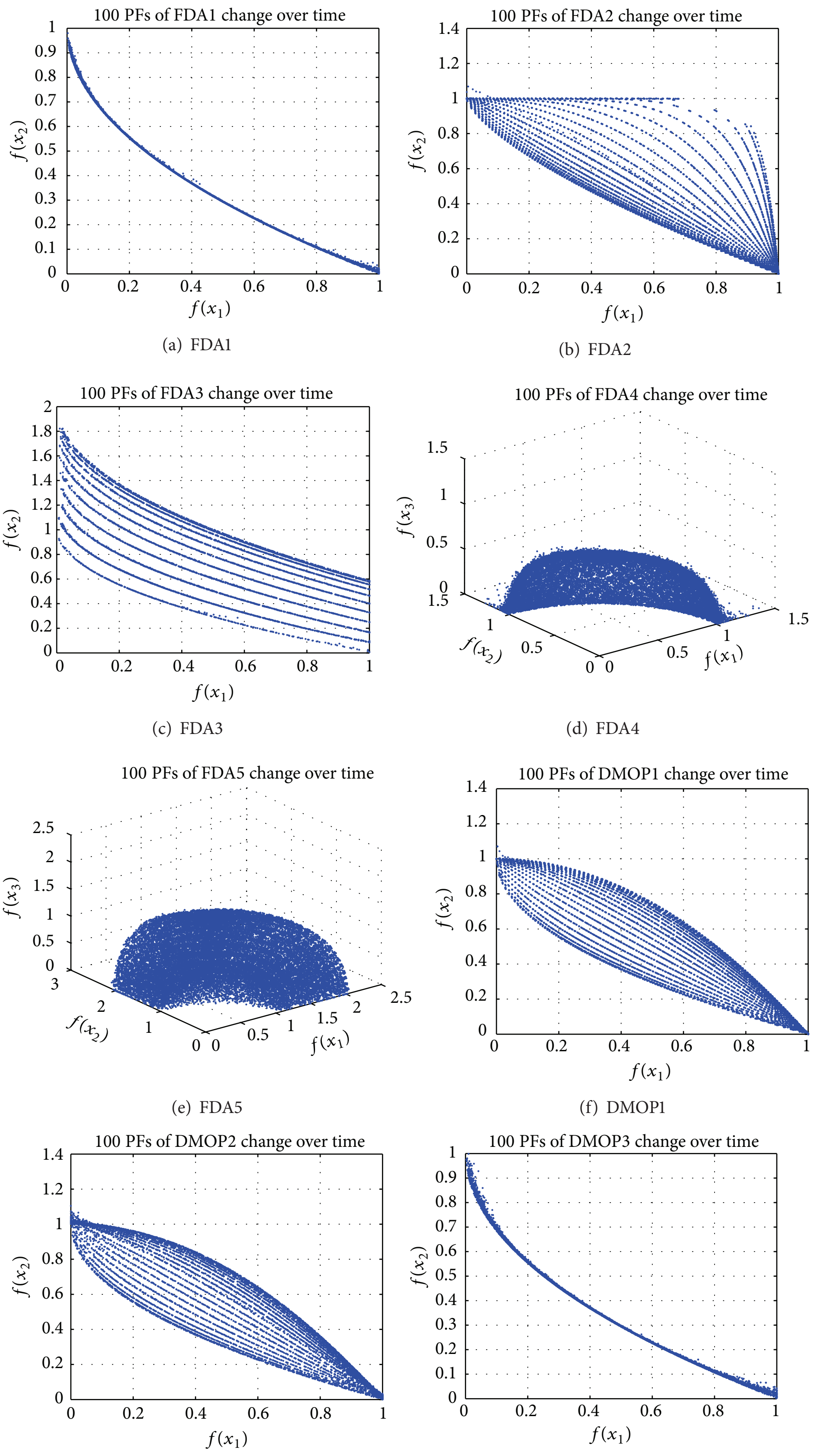
(m) DMOP2 τ_t = 20

(n) DMOP2 τ_t = 5

(o) DMOP3 τ_t = 20

(p) DMOP3 τ_t = 5

FIGURE 5: The pareto fronts of benchmarks under different frequencies when $t = 5, 10, 20, 23, 26, 34$.



(a) FDA1

(b) FDA2

(c) FDA3

(d) FDA4

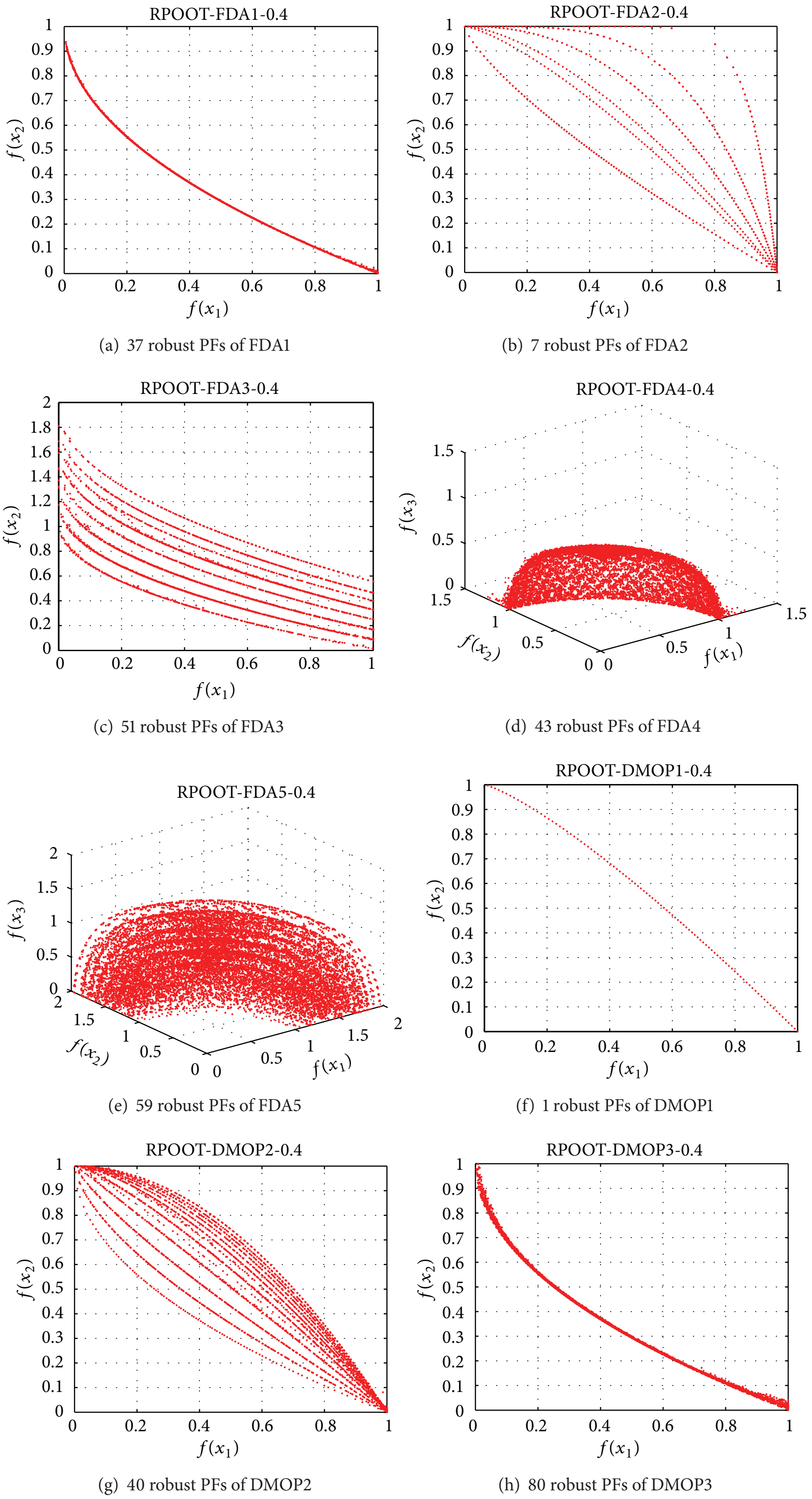
(e) FDA5

(f) DMOP1

(g) DMOP2

(h) DMOP3

FIGURE 6: The 100 PFs of benchmark functions changing over time from $t = 1$ to 100.



(a) 37 robust PFs of FDA1

(b) 7 robust PFs of FDA2

(c) 51 robust PFs of FDA3

(d) 43 robust PFs of FDA4

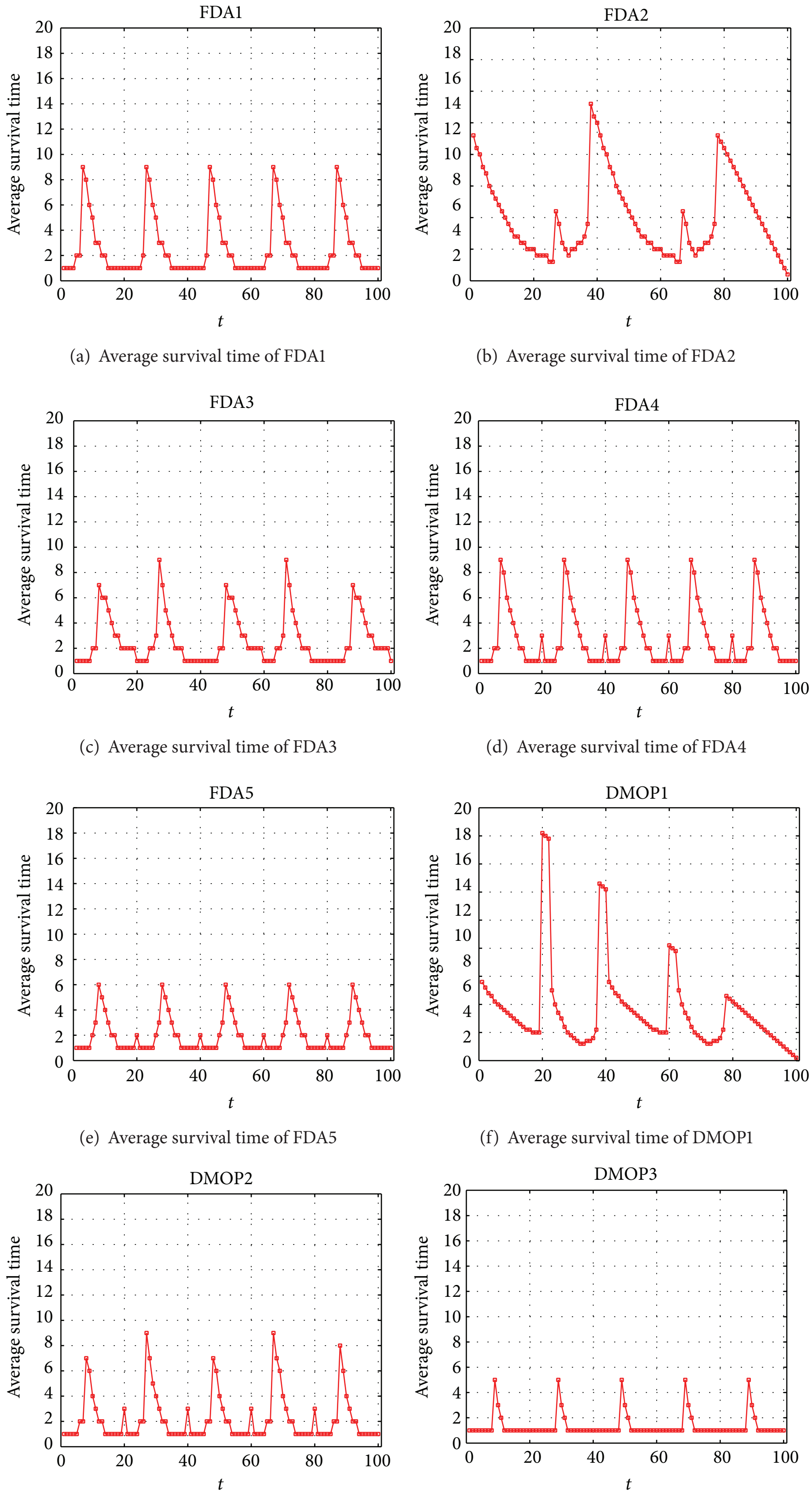
(e) 59 robust PFs of FDA5

(f) 1 robust PFs of DMOP1

(g) 40 robust PFs of DMOP2

(h) 80 robust PFs of DMOP3

Figure 7: The robust optimal pareto fronts of benchmark functions.



(a) Average survival time of FDA1

(b) Average survival time of FDA2

(c) Average survival time of FDA3

(d) Average survival time of FDA4

(e) Average survival time of FDA5

(f) Average survival time of DMOP1

(g) Average survival time of DMOP2

(h) Average survival time of DMOP3

Figure 8: Average survival time of benchmark functions.

Table 3: Comparison of the robust generation distances of robust optimal pareto fronts under different η.

Functions	Measures		η							
			$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.3$	$\eta = 0.4$	$\eta = 0.5$	$\eta = 0.6$	$\eta = 0.7$	$\eta = 0.8$
FDA1	RGD	μ	0.0064	0.0067	0.0066	0.0054	0.0050	0.0046	0.0050	0.0043
		σ	0.0014	0.0010	0.0013	$6.2714e-4$	$4.1899e-4$	$7.0798e-4$	$2.2421e-4$	$1.2005e-4$
	RIGD	μ	0.1258	0.1028	0.1079	0.0095	0.0096	0.0100	0.0094	0.0077
		σ	0.1012	0.0996	0.0623	0.0021	0.0025	0.0050	0.0022	0.0016
FDA2	RGD	μ	0.0493	0.0802	0.2016	0.2044	0.2147	0.2359	0.2510	0.2397
		σ	0.0082	0.0055	0.0256	0.0186	$3.3963e-6$	0.0128	0.0148	0.0236
	RIGD	μ	0.0486	0.0822	0.2416	0.2438	0.2613	0.2675	0.2772	0.2762
		σ	0.0113	0.0049	0.0388	0.0316	$8.0722e-7$	0.0037	0.0148	0.0140
FDA3	RGD	μ	0.0438	0.0611	0.0689	0.1205	0.3034	0.3363	0.2115	0.3202
		σ	0.0183	0.0039	0.0032	0.0084	0.1128	0.1459	0.0240	0.0984
	RIGD	μ	0.1346	0.1342	0.1308	0.1544	0.3095	0.3452	0.2337	0.3341
		σ	0.0714	0.0608	0.0742	0.0579	0.1003	0.1544	0.0299	0.1021
FDA4	RGD	μ	0.3935	0.3898	0.3886	0.3879	0.3874	0.3877	0.3877	0.3869
		σ	0.0045	0.0030	$7.0914e-4$	$4.6005e-4$	0.0017	$8.8703e-4$	$4.4630e-4$	$5.2637e-4$
	RIGD	μ	0.0699	0.0573	0.0520	0.0543	0.0497	0.0476	0.0496	0.0424
		σ	0.0140	0.0131	0.0021	0.0043	0.0049	0.0033	0.0051	0.0023
FDA5	RGD	μ	0.7700	0.7703	0.7698	0.7761	0.7640	0.7650	0.7700	0.7637
		σ	0.0055	0.0061	0.0059	0.0172	0.0066	0.0060	0.0073	0.0074
	RIGD	μ	0.0897	0.0879	0.2593	0.3619	0.1659	0.1737	0.1984	0.1973
		σ	0.0063	0.0044	0.0506	0.1683	0.0020	0.0048	0.0019	0.0018
DMOP1	RGD	μ	0.0564	0.0913	0.1195	0.1523	0.1523	0.1523	0.1523	0.1523
		σ	0.0232	0.0080	0.0093	$2.2854e-6$	$3.8348e-6$	$3.1345e-6$	$3.8348e-6$	$3.4498e-6$
	RIGD	μ	0.0580	0.0845	0.1240	0.1564	0.1564	0.1564	0.1564	0.1564
		σ	0.0235	0.0083	0.0084	$1.0404e-7$	$1.3379e-7$	$1.2821e-7$	$1.3379e-7$	$1.7569e-7$
DMOP2	RGD	μ	0.0606	0.0611	0.0375	0.1255	0.0509	0.0634	0.0871	0.1198
		σ	0.0269	0.0193	0.0027	0.0671	0.0033	0.0134	0.0178	0.0594
	RIGD	μ	0.1779	0.1211	0.0789	0.1335	0.0518	0.0674	0.0866	0.1270
		σ	0.1300	0.0775	0.0553	0.0743	0.0030	0.0114	0.0218	0.0667
DMOP3	RGD	μ	0.0093	0.0098	0.0104	0.0101	0.0129	0.0107	0.0099	0.0107
		σ	$7.9206e-004$	0.0011	0.0022	0.0025	0.0063	0.0012	0.0013	0.0022
	RIGD	μ	0.1525	0.1847	0.1824	0.1465	0.1866	0.1517	0.2238	0.1993
		σ	0.0721	0.0407	0.0711	0.0508	$1.1782e-6$	0.0642	0.1010	0.1035

of the robust pareto-optimal over time is pointed out by the robustness definition of nondomination solution in the time scale. Secondly, we developed the new definition survival time which means how many time-varying environments it fits for. Thirdly, a framework for finding robust pareto fronts is proposed, and a MOEA/D is employed as an optimizer. Lastly, eight dynamic multiobjective benchmark functions are adopted to indicate the feasibility of the algorithm. From the simulation results, we receive that the robustness of RPOOT depends on the parameter η. Less robust pareto fronts are contained in the whole period with the increasing of the neighborhood size η. It means that the robustness is better. At the same time, the convergence of the RPOOT is worse. Moreover, the survival time of the robust pareto front at each time-varying moment is far larger than 1. That is, each robust pareto front of the robust pareto-optimal solution set can fit for more than 1 consecutive changed environments.

Due to the fact that the dynamic system is performed online, we do not obtain the future landscape on the current environment. The future fitness values of the solutions considered in RPOOT should be predicted through the past fitness values. So, the estimation and prediction task are inevitable in the future work.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Table 4: Comparison of the performance between TMO and RPOOT.

Functions		TMO		RPOOT ($\eta = 0.4$)	
		GD	IGD	RGD	RIGD
FDA1	μ	0.0016	0.0019	0.0054	0.0095
	σ	$6.3501e - 007$	$5.9935e - 008$	$6.2714e - 4$	0.0021
FDA2	μ	0.0016	0.0019	0.2044	0.2438
	σ	$6.3501e - 007$	$5.9935e - 008$	0.0186	0.0316
FDA3	μ	0.0029	0.0030	0.1205	0.1544
	σ	$5.4816e - 006$	$6.6723e - 005$	0.0084	0.0579
FDA4	μ	0.0070	0.0067	0.3879	0.0543
	σ	$1.6289e - 005$	$1.3751e - 004$	$4.6005e - 4$	0.0043
FDA5	μ	0.0119	0.0115	0.7761	0.3619
	σ	$1.1204e - 005$	$1.3446e - 004$	0.0172	0.1683
DMOP1	μ	0.0042	$2.6464e - 005$	0.1523	0.1564
	σ	$1.2932e - 006$	$1.3513e - 006$	$2.2854e - 6$	$1.0404e - 7$
DMOP2	μ	$4.5087e - 005$	$5.6999e - 005$	0.1255	0.1335
	σ	$1.5932e - 005$	$1.6302e - 005$	0.0671	0.0743
DMOP3	μ	$3.8105e - 005$	$6.7778e - 005$	0.1763	0.1866
	σ	$6.4825e - 006$	$8.7792e - 006$	$1.4261e - 5$	$1.4064e - 6$

Table 5: The overall survival time of robust PFs.

Function	FDA1	FDA2	FDA3	FDA4	FDA5	DMOP1	DMOP2	DMOP3
Overall survival time	258	1009	262	273	199	1794	224	135

Acknowledgment

This research is jointly supported by the Fundamental Research Funds for the Central Universities (no. 2013XK09) and by National Basic Research Program of China (2014CB046300).

References

[1] M. Helbig and A. P. Engelbrecht, "Archive management for dynamic multi-objective optimisation problems using vector evaluated particle swarm optimisation," in *Proceedings of the IEEE Congress of Evolutionary Computation (CEC '11)*, pp. 2047–2054, New Orleans, La, USA, June 2011.

[2] C.-K. Goh and K. C. Tan, "A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 1, pp. 103–127, 2009.

[3] 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.

[4] Y. Jin and B. Sendhof, "Constructing dynamic optimization test problems using the multi-objective optimization concept," in *Applications of Evolutionary Computing*, vol. 3005 of *Lecture Notes in Computer Science*, pp. 525–536, Springer, New York, NY, USA, 2004.

[5] T. T. Nguyen, S. Yang, and J. Branke, "Evolutionary dynamic optimization: a survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 6, pp. 1–24, 2012.

[6] M. Cámara, J. Ortega, and F. de Toro, "A single front genetic algorithm for parallel multi-objective optimization in dynamic environments," *Neurocomputing*, vol. 72, no. 16–18, pp. 3570–3579, 2009.

[7] C. Hu, H. Yao, and X. Yan, "Multiple particle swarms coevolutionary algorithm for dynamic multi-objective optimization problems and its application," *Journal of Computer Research and Development*, vol. 50, no. 6, pp. 1313–1323, 2013.

[8] L. Wu, *Research on Multi-Objective Dynamic Differential Evolution and Its Applications*, Hunan University, Changsha, China, 2011.

[9] 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.

[10] T. Ray, A. Isaacs, and W. Smith, "A memetic algorithm for dynamic multiobjective optimization," *Multi-Objective Memetic Algorithms*, vol. 171, pp. 353–367, 2009.

[11] H. Chen, M. Li, and X. Chen, "Using diversity as an additional-objective in dynamic multi-objective optimization algorithms," in *Proceedings of the 2nd International Symposium on Electronic Commerce and Security (ISECS '09)*, pp. 484–487, Nanchang, China, May 2009.

[12] 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.

[13] M. Li, S. Azarm, and V. Aute, "A multi-objective genetic algorithm for robust design optimization," in *Proceedings of Genetic and Evolutionary Computation Conference*, pp. 771–778, June 2005.

[14] K. Deb and H. Gupta, "Introducing robustness in multiple-objective optimization," KanGAL Report 2004016, Kanpur

Genetic Algorithms Laboratory, Indian Institute of Technology, Kanpur, India, 2004.

[15] C. Barrico and C. H. Antunes, “A new approach to robustness analysis in multi-objective optimization,” in *Proceedings of the 7th International Conference on Multi-Objective Programming and Goal Programming*, Loire Valley (Tours), France, 2006.

[16] C. Barrico and C. H. Antunes, “Robustness analysis in multi-objective optimization using a degree of robustness concept,” in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC ’06)*, pp. 1887–1892, Vancouver, Canada, July 2006.

[17] C. Cromvik, P. Lindroth, M. Patriksson, and A.-B. Strömberg, *A New Robustness Index for Multi-Objective Optimization Based on a User Perspective*, 2011.

[18] X. Yu, Y. Jin, K. Tang, and X. Yao, “Robust optimization over time—a new perspective on dynamic optimization problems,” in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC ’10)*, pp. 3998–4003, July 2010.

[19] Y. Jin, K. Tang, X. Yu, B. Sendhoff, and X. Yao, “A framework for finding robust optimal solutions over time,” *Memetic Computing*, vol. 5, no. 1, pp. 3–18, 2013.

[20] H. Fu, B. Sendhoff, K. Tang, and X. Yao, “Finding robust solutions to dynamic optimization problems,” in *Applications of Evolutionary Computation*, vol. 7835 of *Lecture Notes in Computer Science*, pp. 616–625, Springer, Berlin, Germany, 2013.

[21] Q. Zhang and H. Li, “MOEA/D: a multiobjective evolutionary algorithm based on decomposition,” *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[22] C.-K. Goh and K.-C. Tan, *Evolutionary Multi-Objective Optimisation in Uncertain Environments: Issues and Algorithms*, Springer, Berlin, Germany, 2009.