
A Self-Adaptive Response Strategy for Dynamic Multiobjective Evolutionary Optimization Based on Objective Space Decomposition

Ruochen Liu

ruochenliu@xidian.edu.cn

Key Lab of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, Xi'an, 710071, China

Jianxia Li

1761389691@qq.com

Key Lab of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, Xi'an, 710071, China

Yaochu Jin

yaochu.jin@surrey.ac.uk

Department of Computer Science, University of Surrey, Guildford, GU2 7XH, United Kingdom

Licheng Jiao

lcjiao@mail.xidian.edu.cn

Key Lab of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, Xi'an, 710071, China

https://doi.org/10.1162/evco_a_00289

Abstract

Dynamic multiobjective optimization deals with simultaneous optimization of multiple conflicting objectives that change over time. Several response strategies for dynamic optimization have been proposed, which do not work well for all types of environmental changes. In this article, we propose a new dynamic multiobjective evolutionary algorithm based on objective space decomposition, in which the maxi-min fitness function is adopted for selection and a self-adaptive response strategy integrating a number of different response strategies is designed to handle unknown environmental changes. The self-adaptive response strategy can adaptively select one of the strategies according to their contributions to the tracking performance in the previous environments. Experimental results indicate that the proposed algorithm is competitive and promising for solving different DMOPs in the presence of unknown environmental changes. Meanwhile, the proposed algorithm is applied to solve the parameter tuning problem of a proportional integral derivative (PID) controller of a dynamic system, obtaining better control effect.

Keywords

Dynamic multiobjective optimization, objective space decomposition, maxi-min fitness function, self-adaptive response strategy, PID control.

1 Introduction

Over the past years, a number of evolutionary algorithms have been proposed to solve multiobjective optimization problems (MOPs) that involve two or more conflicting objectives (Coello Coello, 2002; Deb et al., 2002; Fonseca and Fleming, 1993; Zhang and Li, 2007; Zhang et al., 2008; Zitzler et al., 2002). Many MOPs in the real world, however, Manuscript received: 23 December 2019; revised: 30 October 2020 and 14 January 2021; accepted: 18 January 2021.

are subject to various types of uncertainty, resulting in time-varying objective and / or constraint functions (Deb et al., 2007). These MOPs are referred to as dynamic multiobjective optimizations problems (DMOPs). In recent years, increasing attention has been paid to solving DMOPs in the real world, such as path planning (Michalewicz et al., 2007), evolutionary robotics (Tinós and Yang, 2007), and financial optimization problems (Tezuka et al., 2007).

The goal of a dynamic multiobjective optimization algorithm is to track the changing Pareto-optimal set (*PS*) (Pelosi and Selleri, 2014). In order to efficiently solve DMOPs, two basic requirements need to be considered. First, the algorithm should be able to effectively respond to environmental changes. Second, the algorithm is able to quickly find the *PS* of the current environment before the next change occurs. Recently, a large number of dynamic multiobjective evolutionary algorithms have been proposed (Jiang and Yang, 2017b; Zhou et al., 2014, 2007), which, however, usually perform well only for specific types of environmental changes. To address this issue, this article proposes a self-adaptive response strategy (SRS) that is able to adaptively select different response strategies for different environmental changes according to the contribution of each response strategy. We adopt a multiobjective evolutionary algorithm based on objective space decomposition (MOEA-OSD) as the basic optimizer. Therefore, the entire algorithm combining MOEA-OSD with SRS is called MOEA-OSD/SRS.

Note that the main new contribution of MOEA-OSD/SRS lies in SRS, which integrates several popular response strategies, including random diversity introduction (RDI) (Deb et al., 2007), mutational diversity introduction (MDI) (Deb et al., 2007), linear prediction strategy (LPS) (Zhou et al., 2007), feed-forward prediction strategy (FPS) (Hatzakis and Wallace, 2006), and population prediction strategy (PPS) (Zhou et al., 2014). By adding different labels, individuals generated by different response strategies can be traced, and their contributions in the previous environments can be recorded. Once the environment changes, the probability at which each response strategy is selected in the new environment is determined according to its contribution in the previous environments, making the proposed algorithm adaptive to unknown environmental changes.

In order to quickly find the *PS* of the current environment before the next environmental change occurs, MOEA-OSD is adopted as the basic static multiobjective evolutionary algorithm. MOEA-OSD decomposes the objective space of the MOP into a number of subobjective spaces using a set of uniformly distributed reference vectors. The reference vectors are generated uniformly in the objective space, evenly dividing the objective space into a number of subobjective spaces. The subobjective spaces are treated as an external archive, and the corresponding historical optimal solutions of each reference vector are stored in the corresponding subobjective space. Owing to the evenly divided subobjective spaces, the algorithm can naturally maintain a good diversity. In addition, the algorithm adopts the maxi-min fitness function (Balling, 2003) to calculate the fitness value of each solution, which is believed to be able to simultaneously account for both convergence and diversity of the solutions.

To examine the performance of MOEA-OSD/SRS, three sets of comprehensive empirical studies are performed, including a comparison of SRS with other six effective response strategies, a comparison of MOEA-OSD/SRS with other seven state-of-the-art dynamic multiobjective optimization algorithms, and a comparison of MOEA-OSD with other six multiobjective evolutionary algorithms. Meanwhile, MOEA-OSD/SRS is employed to tune the parameters of a proportional integral derivative (PID) controller for a dynamic system.

The rest of this article is organized as follows. Section 2 presents a brief review of response strategies considered in the current work. The proposed MOEA-OSD/SRS is presented in detail in Section 3. Section 4 introduces the benchmark functions, parameter settings, and performance metrics, followed by a detailed description of the experimental results and analysis. Section 5 applies MOEA-OSD/SRS to optimize the parameters of a PID controller of a dynamic system. Finally, Section 6 draws a conclusion of the work and suggests future research.

2 Background

This section briefly discusses the related work on solving DMOPs and provides a detailed review of the widely used response strategies.

In general, a dynamic multiobjective minimization problem can be described as follows (Farina et al., 2004):

$$\begin{cases} \min y = F(x, t) = (f_1(x, t), f_2(x, t), \dots, f_m(x, t)) \\ s.t. \quad g_i(x, t) \leq 0, \quad i = 1, 2, \dots, p \\ \quad \quad h_j(x, t) = 0, \quad j = 1, 2, \dots, q, \end{cases} \quad (1)$$

where $x = [x_1, x_2, \dots, x_n]$ is the decision vector, and $F(x, t)$ is the set of objectives to be minimized with respect to time t . Functions $g(x, t)$ and $h(x, t)$ represent inequality and equality constraints, respectively. In this work, however, we focus only on unconstrained DMOPs.

Comprehensive reviews of research on solving DMOPs can be found in Cruz et al. (2011), Goh and Tan (2009b), Helbig and Engelbrecht (2014), Jin and Branke (2005), Nguyen et al. (2012), and Raquel and Yao (2013). Evolutionary algorithms (Avdagić et al., 2009; Jiang and Yang, 2017b; Wu et al., 2015; Zeng et al., 2006), particle swarm optimization algorithms (Greeff and Engelbrecht, 2008, 2010; Helbig and Engelbrecht, 2011, 2012, 2013b; Rong et al., 2018; Salazar Lechuga, 2009), and immune algorithms (Shang et al., 2005, 2014) have been used to solve DMOPs. In addition, cooperative co-evolutionary algorithms have been designed for dynamic multiobjective optimization (Goh and Tan, 2009a; Liu, Chen et al., 2014).

While they have shown better performance in solving static MOPs (Chang et al., 2008; Ishibuchi et al., 2009; Li and Zhang, 2009; Liu and Niu, 2013; Yuen and Ramli, 2010; Zhang and Li, 2007), decomposition-based MOEAs have hardly been used to solve DMOPs. To fill the gap, this work adopts a dynamic multiobjective optimization algorithm based on objective space decomposition.

For a multiobjective optimization algorithm to be capable of solving DMOPs, four basic approaches to responding to environmental changes have been investigated. These approaches are typically used for generating the initial population in the new environment. We review the widely used DMOAs with respect to the adopted response strategy in greater detail.

2.1 Diversity Introduction Strategy

When solving a DMOP, it becomes difficult for an optimization algorithm to quickly locate the global optimum usually due to the lack of diversity in the population when an environmental change occurs. Thus, it is helpful to introduce diversity when environmental changes are detected (Cobb, 1990; Deb et al., 2007; Goh and Tan, 2009a; Greeff and Engelbrecht, 2008; Jin and Branke, 2005; Liu et al., 2017; Vavak et al., 1997).

Two typical diversity introduction strategies are proposed by Deb et al. (2007). In one of the approaches, a certain proportion of the individuals in the population are replaced by randomly generated individuals, which is referred to as DNSGA-II-A. In the other approach, a certain proportion of the individuals in the population are mutated, known as DNSGA-II-B.

The diversity introduction strategy is simple and easy to implement, but it responds to environmental changes fairly arbitrarily, which may mislead the evolutionary search. Meanwhile, the proportion of diversity introduction has a great impact on the performance of the algorithm.

2.2 Diversity Maintenance Strategy

Diversity maintenance strategies (Bui et al., 2005; Grefenstette et al., 1992; Morrison, 2002; Zeng et al., 2006) aim to maintain a certain degree of diversity of the population in the process of evolutionary optimization. For instance, Zeng et al. (2006) proposed a dynamic multiobjective evolutionary algorithm based on an orthogonal evolutionary algorithm (DOMOEA) to solve continuous DMOPs. The idea is simply to adopt the optimal solutions of the previous environment to initialize the population in the new environment. This strategy is therefore computationally very efficient.

Unfortunately, diversity maintenance strategies are mainly suited for solving continuous optimization problems. They are good for solving DMOPs with minor environmental changes, but may perform poorly when the changes are severe.

2.3 Prediction-Based Strategy

If there is a certain relationship between the optimal solutions in the new environment and those in the previous environments, it is desirable to predict the location of the new optimal solutions and include them in the initial population for the new environment. Hatzakis and Wallace (2006) proposed a forward-looking approach as the response mechanism. Once an environmental change is detected, the position of the optimal solution in the new environment is predicted by using an autoregressive model. Zhou et al. (2007) proposed a prediction-based reinitialization strategy (PRI). In PRI, two strategies are introduced when an environmental change is detected. The first strategy is to predict the new position of each individual in the population according to its previous position change. The second strategy is to perturb the current population with Gaussian noise. PRI was further extended in Zhou et al. (2014) by suggesting a prediction strategy based on the whole population (PPS). In PPS, a Pareto set is divided into two parts: a center point and a manifold. A set of center points in the previous environments are used to estimate the center in the new environment, and previous manifolds are used to predict the new manifold. Then, the initial population in the new environment can be generated by combining the predicted center and manifold. Recently, Jiang and Yang (2017b) developed a steady-state and generational evolutionary algorithm (SGEA) that can respond to changes in a steady-state manner. When an environmental change is detected, SGEA reuses multiple outdated solutions with a better distribution and relocates partial solutions to the regions near the new Pareto optimal front (*PF*) based on the information gathered from the previous and new environments. Other prediction-based strategies have been reported (Liu, 2010; Wu et al., 2015).

Prediction-based strategies work effectively if the prediction is sufficiently reliable. Unfortunately, inaccurate predictions may mislead the evolutionary search. Typically, prediction-based strategies work more efficiently for DMOPs having periodic environmental changes.

2.4 Memory-Based Strategy

Last but not the least, memory-based strategies (Ng and Wong, 1995; Yang, 2006; Zhang and Qian, 2011) have also been designed to respond to environmental changes. These strategies store relevant information about each of the previous environments and reuse the stored information in the new environment. For example, Yang (2006) proposed an associative memory mechanism in which the solutions and a distribution estimation are stored as memory individuals. When an environmental change is detected, all individuals in the memory are re-evaluated. The better an individual is, the more likely its related distribution will be selected to generate the initial population in the new environment.

Similar to the prediction-based strategies, memory-based strategies are useful for solving DMOPs having periodic changes. However, storing information in and retrieving information from the memory can be computationally intensive. Note that this strategy is useful for solving periodic DMOPs only.

Almost all response strategies discussed above have both advantages and disadvantages and work well in dealing with certain types of environmental changes. For this reason, we propose a self-adaptive response strategy for dynamic multiobjective evolutionary algorithms, which is able to automatically select the most suited response strategies for given problems.

3 The Proposed MOEA-OSD/SRS

As discussed before, a DMOA should be able to achieve a set of diverse Pareto optimal solutions of the current environment before a new environmental change occurs. In addition, it should be able to timely detect an environmental change and respond to the environmental change effectively.

To meet the above requirements, the proposed algorithm adopts MOEA-OSD, a decomposition-based approach, as the base optimizer. A self-adaptive response strategy is embedded in MOEA-OSD to quickly respond to unknown environmental changes. In this section, we describe MOEA-OSD, the environmental change detection mechanism, and the self-adaptive response strategy (SRS), before the overall framework of MOEA-OSD/SRS is presented.

3.1 MOEA-OSD

This section introduces the main principle of MOEA-OSD. In MOEA-OSD, firstly, a set of uniformly distributed reference vectors (r^1, r^2, \dots, r^N) are generated to divide the objective space into a number of subobjective spaces. Then MOEA-OSD will find the closest solution of each reference vector, which will be introduced in the following section. In the evolutionary optimization process, the offspring population is produced by performing the differential crossover operator (Price et al., 2006) and Gaussian mutation operator (Higashi and Iba, 2003) on parent individuals, and then the parent and offspring populations are merged to form a combined population. The maxi-min fitness function is used to calculate the fitness value of all individuals, and the individuals with smaller fitness value are selected to generate the next population. Finally, the closest solution of each reference vector is updated. The above process repeats until a termination criterion is satisfied. The main steps of MOEA-OSD are described in Algorithm 1.

3.1.1 Finding the Closest Solution of Each Reference Vector

In order to generate a set of uniformly distributed reference vectors, we adopt the approach introduced in Cheng et al. (2016).

Algorithm 1: MOEA-OSD

- 1 **step 1. Initialization:**
 - 2 step 1.1: Initialize $iter = 0$; Initialize population POP_0 ; Initialize reference vectors (r^1, r^2, \dots, r^N) ;
 - 3 step 1.2: Find the corresponding closest solution of each reference vector in POP_0 ;
 - 4 step 1.3: Initialize reference point $Z = (Z_1, Z_2, \dots, Z_m)$,
 $Z_k = \min \{f_k(x) | x \in POP_0\}, k = 1, 2, \dots, m.$
 - 5 **step 2. Update:**
 - 6 step 2.1: Generate the offspring population Q_{iter} by performing the genetic operators on the parent population POP_{iter} ;
 - 7 step 2.2: Combine POP_{iter} and Q_{iter} , and compute the maxi-min fitness value of all the individuals of the mixed population; select the nondominated solutions to generate the next parent population POP_{iter+1} ;
 - 8 step 2.3: Update reference point, if $Z_k > \min \{f_k(x) | x \in POP_{iter+1}\}$,
 $Z_k = \min \{f_k(x) | x \in POP_{iter+1}\}, k = 1, 2, \dots, m.$
 - 9 step 2.4: Update the closest solution of each reference vector.
 - 10 **Step 3. Termination criteria:**
 - 11 If termination criterion is satisfied, output the closest solution of each reference vector. Otherwise, $iter = iter + 1$, go to **Step 2.**
-

First, a set of uniformly distributed points are generated on a unit hyperplane, as presented in Eq. (2):

$$u_i = (u_i^1, u_i^2, \dots, u_i^m)$$

$$u_i^j \in \left\{ \frac{0}{H}, \frac{1}{H}, \dots, \frac{H}{H} \right\}, \sum_{j=1}^m u_i^j = 1, \tag{2}$$

where $i = 1, 2, \dots, N$ with N being the number of uniformly distributed points, m is the number of objectives, and H is a positive integer for the simplex-lattice design.

Then, the corresponding reference vectors can be obtained by Eq. (3):

$$r^i = \frac{u_i}{\|u_i\|}. \tag{3}$$

For a set of uniformly distributed reference vectors (r^1, r^2, \dots, r^N) and a population POP , each solution finds its closest reference vector in following way:

$$V^i = \left\{ r | \Delta(F(x^i), r) = \max_{1 \leq j \leq N} \{ \Delta(F(x^i), r^j) \} \right\}$$

$$x^i \in POP, i = 1, 2, \dots, N$$

$$\Delta(F(x^i), r^j) = \frac{r^j \times (F(x^i) - Z)^T}{\|r^j\| \times \|(F(x^i) - Z)\|}, \tag{4}$$

where V^i represents the corresponding closest reference vector of the solution x^i , $Z = (Z_1, Z_2, \dots, Z_m)$ is the reference point, and $Z_k = \min \{f_k(x) | x \in POP\}$ for each $k = 1, 2, \dots, m$ and $\Delta(F(x^i), r^j)$ is the cosine of angle between the vector r^j and $F(x^i) - Z$.

By finding the closest reference vector of each solution in population POP , the solutions are grouped into different subobjective spaces. However, the number of solutions

in some subobjective spaces may be more than one or may be zero. To ensure that each subobjective space has at least one solution, we also find the closest solution for each reference vector as follows:

$$\begin{aligned}
 X^i &= \left\{ x \mid \Delta(F(x), r^i) = \max_{1 \leq j \leq N} \{ \Delta(F(x^j), r^i) \} \right\} \\
 &\quad x \in POP, i = 1, 2, \dots, N \\
 \Delta(F(x^j), r^i) &= \frac{r^i \times (F(x^j) - Z)^T}{\|r^i\| \times \|(F(x^j) - Z)\|}, \tag{5}
 \end{aligned}$$

where X^i represents the corresponding closest solution of the reference vector r^i .

3.1.2 The Maxi-Min Fitness Function

The maxi-min strategy (Luce and Raiffa, 2012; Rawls, 2009) was first introduced into multiobjective optimization in Balling (2003). Here we also adopt the maxi-min fitness function to compute the fitness of the individual, which can account for both convergence and diversity so that no additional measures for diversity is needed.

Supposing that an MOEA of a size N is used to solve an m -objective multiobjective minimization problem, f_k^i is the k -th objective function value of the i -th individual. Then, the maxi-min fitness function of i -th individual of the population is denoted as:

$$\text{fitness}^i = \max_{j \neq i, j=1, \dots, N} (\min_{k=1, \dots, m} (f_k^i - f_k^j)), i = 1, 2, \dots, N. \tag{6}$$

We can see from Eq. (6) that if the maxi-min fitness of an individual is greater than 0, this individual is regarded as a dominated individual, while if the maxi-min fitness of an individual is less than 0, this individual is regarded as a nondominated individual. If the maxi-min fitness of an individual is 0, this individual is regarded as a weakly dominated individual. The maxi-min fitness function is well suited for multiobjective optimization, because it can penalize individuals in a crowded region while rewarding those in a less crowded region (Balling, 2003).

3.1.3 Environmental Selection

Before environmental selection, parent population POP_{iter} is merged with the offspring population Q_{iter} to generate a combined population PQ_{iter} . According to Eq. (6), the fitness values of all individuals in the population PQ_{iter} are calculated. The individual with the fitness value being less than 0 is regarded a nondominated individual. If the number of nondominated individuals in PQ_{iter} is greater than N , all nondominated individuals are retained. If the number of nondominated individuals in PQ_{iter} is less than or equal to N , then the individuals in PQ_{iter} are sorted by the maxi-min fitness in an ascending order, and remaining the first N individuals. The final operation is to find the closest individual of each reference vector, which forms the new parent population with N individuals. The main purpose of this selection strategy is to allow the individuals that dominate others and contribute more to the maintenance of the population diversity to have a larger chance to survive.

3.1.4 Update Strategy

During each generation, each reference vector should find its closest solution, and if one of the following conditions is satisfied, the existing solution in the subobjective space will be replaced by the newly generated solution: (1) the new solution dominates the existing one; (2) the two solutions do not dominate each other, but the new one is closer to the reference vector of the related subobjective space. The first condition makes sure

that a nondominated solution is preserved so that the population converges gradually to the Pareto front. By contrast, the second condition aims to maintain the diversity of the population.

3.2 Environmental Change Detection

The most common way to detect environmental changes is to re-evaluate a portion of existing solutions (Carlisle and Dozier, 2000). This is realized by randomly choosing a number of individuals from the current population and re-evaluating them, then check the differences of the objective values or the constraint conditions of the chosen individuals between two generations. If there is a significant difference, then an environmental change is considered to have happened.

In this work, the following similarity detection operator presented in Farina et al. (2004) and Liu et al. (2017) is used to detect environmental changes:

$$\varepsilon(\text{iter}) = \frac{1}{n_\varepsilon} \sum_{j=1}^{n_\varepsilon} \left\| \frac{F(x^j, \text{iter}) - F(x^j, \text{iter} + 1)}{F(x^j, \text{iter})_{\max} - F(x^j, \text{iter})_{\min}} \right\|, \quad (7)$$

where n_ε is the number of the individuals chosen from the current population. If $\varepsilon(\text{iter}) > \bar{\varepsilon}$ ($\bar{\varepsilon}$ is the threshold value, 0.00001), an environmental change occurs.

3.3 Self-Adaptive Response Strategy

This section mainly introduces the ideas of SRS in detail. SRS integrates several widely used response strategies and continuously determines the probability of selecting the response strategies according to their contributions to the performance improvement in the previous environments, thereby adaptively selecting the best response strategy for different problems. This way, the self-adaptive response strategy can be used to handle unknown environmental changes.

In our empirical studies, SRS integrates five commonly used response strategies, namely RDI (Deb et al., 2007), MDI (Deb et al., 2007), LPS (Zhou et al., 2007), FPS (Hatzakis and Wallace, 2006), and PPS (Zhou et al., 2014). The description of the each of these is shown in Section S1 of the Supplementary material, available at https://doi.org/10.1162/evco_a_00289. In the following, we will present the details of SRS.

3.3.1 Proposed SRS

It should be pointed out that according to the original reference, the prediction strategy in PPS and FPS comes into play only after 23 environmental changes. Thus, SRS adopts RDI as the response strategy in the first 23 environmental changes so that the five strategies can fairly compete with each other, because the original LPS, FPS, and PPS all use the RDI strategy when the prediction model is not yet ready to work in the early environmental changes.

When the 24th environmental change is detected, SRS selects the five different response strategies for initializing the new population at an equal probability of 0.2. That is to say, the five different response strategies all generate a new population, then SRS selects 20% of its individuals from each population to form a new population. By assigning a specific label to the individuals generated by each of the five response strategies, SRS is able to track them during the evolution. Suppose RDI is labeled as 1, MDI, LPS, FPS, and PPS are labeled as 2, 3, 4, and 5, respectively. 1, 2, 3, 4, and 5 are considered as the meta labels (*MLs*). Consequently, the individuals generated by RDI, MDI, LPS, FPS, and PPS are assigned labels as (1,1,1), (2,2,2), (3,3,3), (4,4,4), and (5,5,5), respectively.

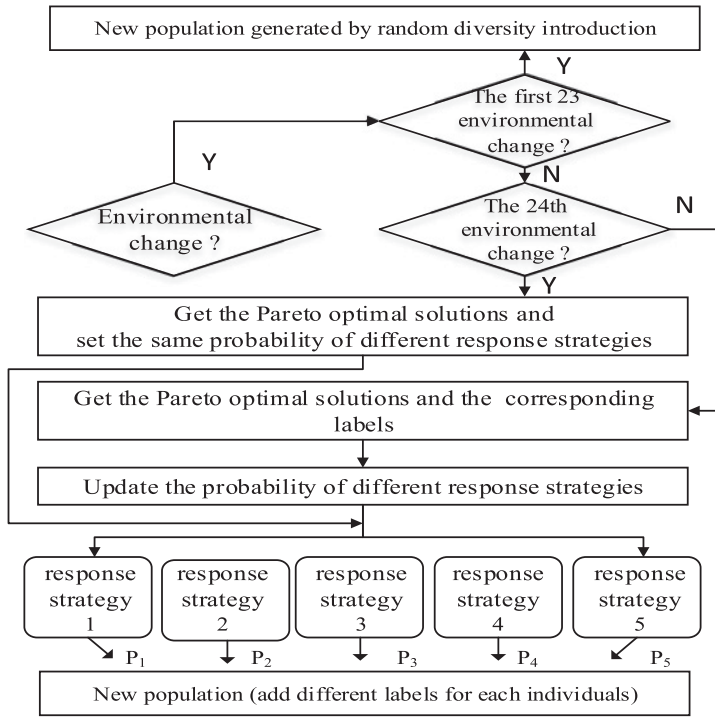


Figure 1: Self-adaptive Response Strategy (SRS).

We will explain at the end of this section why we assign label (1,1,1) (i.e., the label of an individual consists of three *MLs*) instead of (1) to an individual. In the entire evolutionary process, each individual will always carry a triplet label (I, J, K) to trace the origin of the individual, where $I, J, K \in \{1, 2, 3, 4, 5\}$.

When a new (except for the first 24) environmental change is detected, SRS can obtain a set of nondominated solutions, each having a label in the format of (I, J, K) . Then it examines how much performance improvement each individual in the previous environment has contributed to the performance improvement by counting the total number of *MLs* corresponding to each response strategy. For example, the number of the *MLs* in one individual's label (1,1,1) is 3. Then we calculate the contribution ratio of each response strategy. Assume we have 100 individuals in total, the number of all the *MLs* is 300, and the number of RDI's label (1) is 45. Thus the contribution ratio of strategy RDI is 0.15. A strategy having a higher contribution ratio is assigned a greater probability for generating new individuals to respond to new environmental changes. Once an environmental change occurs, the five response strategies all generate a new population, then SRS selects different ratios of individuals from each population to form a new population and passes them to the new environment. This self-adaptive response strategy is illustrated in Figure 1.

When an environmental change is detected, SRS will respond to the change and generate a new initial population for the new environment. In the new environment, we use MOEA-OSD to search for nondominated solutions starting from the new initial population. Therefore, it is important to assign labels in the format of (I, J, K) to these individuals generated by the crossover operator and mutation operator, because the

Table 1: Determining the label of an offspring after the crossover in Situation 1.

	RDI	MDI	LPS	FPS	PPS
<i>ML</i>	1	2	3	4	5
<i>NL</i>	2	3	1	2	1
<i>p</i>	2/9	3/9	1/9	2/9	1/9
<i>Ap (Rp)</i>	2/3 (1)	1 (1)	1/3 (0)	2/3 (1)	1/3 (0)
<i>Dp</i>	-1/3	0	1/3	-1/3	1/3

label assignment process determines whether the contribution of each response strategy is correctly calculated.

3.3.2 Determining the Label of an Offspring Individual

When a mutation operator is applied, the label of the offspring individual is the same as the label of its parent individual. For example, if the label of a parent individual is (1,3,2), the label of the offspring individual is (1,3,2), too. When a differential crossover operator is applied to generate offspring, which uses three parent individuals to generate an offspring individual, the label of the offspring individual is difficult to determine.

Algorithm 2 describes the main components of determining the label of an offspring individual generated by the differential crossover operator. We provide the following three instances to elaborate how Algorithm 2 works.

a) **Situation 1** (Lines 6–8 in Algorithm 2). The following gives a simple example: suppose the labels of the three parent individuals are (1,2,3), (4,2,4), and (1,2,5), respectively. Now we count the number of different *ML*s in these three individuals' labels, which is called as the number of labels (*NL*). Thus, the number of RDI's label (1) is 2, resulting in a probability of 2/9 for the response strategy RDI, this probability is denoted as *p*. Then, we compute the probability of each *ML* appearing in the label of the offspring individual generated by the crossover operator, which is denoted as appearing probability (*Ap*). *Ap* after being rounded is expressed as a rounded probability (*Rp*). The difference between *Ap* and *Rp* is denoted as difference probability ($Dp = Ap - Rp$), which could also be considered as a redundancy probability. For RDI, its label (1) will appear in the label of offspring individual with a probability of $3 \times 2/9 = 2/3$. This will be rounded to 1; that is, $Rp_1 = 1$. Thus, based on *Rp*, we can determine the final label of the offspring individual. In this example, the label of the offspring individual will be (1,2,4). The detailed steps for computing the labels of the offspring individual are listed in Table 1.

b) **Situation 2** (Lines 10–18 in Algorithm 2). Here is a simple example: suppose the labels of three parent individuals are (1,1,2), (2,3,4), and (3,5,4), respectively. As shown in Table 2, according to *Rp*, there are four "1"s. Thus the label of the offspring individual will be (1,2,3,4), which is no longer a triplet (*I, J, K*). Then, according to *Dp*, the response strategies corresponding to labels 1, 2, 3, 4 will have a negative redundancy (-1/3). Therefore, we will randomly select three labels from 1, 2, 3, and 4 to obtain the label of the offspring individual; that is, the offspring individual in this situation will be labeled as (1,2,3) or (1,2,4) or (1,3,4) or (2,3,4).

c) **Situation 3** (Lines 20–23 in Algorithm 2). The following is a simple instance: suppose the labels of the three parent individuals are (1,1,2), (1,2,3), and (2,2,5). As shown in Table 3, according to *Rp*, we can see that only part of the label of the offspring individual can be determined, which is (1,2), since there are only two "1"s. According to

Algorithm 2: Determining the label of an offspring individual generated by differential crossover operator

```

1 step 1: Suppose that RDI, MDI, LPS, FPS, PPS are labeled as 1, 2, 3, 4 and 5 (MLs),
   respectively, and three parent individuals' labels are  $(I_1, J_1, K_1)$ ,  $(I_2, J_2, K_2)$ ,
    $(I_3, J_3, K_3)$ , count the number of different MLs in these three parent individuals'
   labels, then compute the probability (p) of each ML,  $p_1, p_2, p_3, p_4, p_5$ ;
2 Step 2: Compute the appearing probability (Ap) that each ML in the label of an
   offspring individual generated by the crossover operator,  $Ap_1, Ap_2, Ap_3, Ap_4, Ap_5$ ;
3 Step 3: Round  $Ap_1, Ap_2, Ap_3, Ap_4, Ap_5$ , and obtain the rounding probability (Rp) (the
   frequency of different MLs appearing in the label of the offspring individual),
    $Rp_1, Rp_2, Rp_3, Rp_4, Rp_5$ ;
4 Step 4: Compute the difference (redundancy) probability (Dp),
    $Dp_1, Dp_2, Dp_3, Dp_4, Dp_5$ , and  $Dp_k = Ap_k - Rp_k, k = 1, 2, 3, 4, 5$ ;
5 Step 5: Determine the label of offspring individual generated by differential crossover;
6 if  $Rp_1 + Rp_2 + Rp_3 + Rp_4 + Rp_5 = 3$  then
7   /*(situation 1)*/
8   The label of offspring individual can be directly determined as the combination of
   the labels of the response strategies where  $Rp_k > 0, k = 1, 2, 3, 4, 5$ ;
9 else
10  if  $Rp_1 + Rp_2 + Rp_3 + Rp_4 + Rp_5 > 3$  then
11    /*(situation 2)*/
12    /* For this situation, nine MLs of three parent individuals' labels correspond
       five response strategies, so it must be
        $(Rp_1 + Rp_2 + Rp_3 + Rp_4 + Rp_5) = 4 > 3.$ */
13    /* Selecting three labels from four labels as the label of offspring individual.*/
14    if there exists  $Dp_k = 0 \& \& Rp_k > 0, k = 1, 2, 3, 4, 5$  then
15      The label of the offspring individual can be determined directly by
       combining the label of the response strategy where
        $Dp_k = 0 \& \& Rp_k > 0, k = 1, 2, 3, 4, 5$  and the remaining two labels
       randomly selecting from the other three response strategies' labels;
16    else
17      The label of the offspring individual can be determined as the three labels
       randomly selecting from the labels of four response strategies with
        $Rp_k > 0, k = 1, 2, 3, 4, 5$ ;
18    end
19  end
20  else
21    /*(situation 3)*/
22    /* For this situation, nine MLs of three parent individuals' labels correspond
       five response strategies, so it must be
        $(Rp_1 + Rp_2 + Rp_3 + Rp_4 + Rp_5) = 2 < 3.$ */
23    The label of an offspring individual consists two parts: two labels of the
       response strategies where  $Rp_k > 0, k = 1, 2, 3, 4, 5$  and one label randomly
       selecting from the other response strategies' labels;
24  end
25 end

```

Table 2: Determining the label of an offspring after crossover in Situation 2.

	RDI	MDI	LPS	FPS	PPS
<i>ML</i>	1	2	3	4	5
<i>NL</i>	2	2	2	2	1
<i>p</i>	2/9	2/9	2/9	2/9	1/9
<i>Ap (Rp)</i>	2/3 (1)	2/3 (1)	2/3 (1)	2/3 (1)	1/3 (0)
<i>Dp</i>	-1/3	-1/3	-1/3	-1/3	1/3

Table 3: Determining the label of an offspring after crossover in Situation 3.

	RDI	MDI	LPS	FPS	PPS
<i>ML</i>	1	2	3	4	5
<i>NL</i>	3	4	1	0	1
<i>p</i>	3/9	4/9	1/9	0	1/9
<i>Ap (Rp)</i>	1 (1)	4/3 (1)	1/3 (0)	0 (0)	1/3 (0)
<i>Dp</i>	0	1/3	1/3	0	1/3

Dp, the response strategies corresponding to the labels 2, 3, and 5 will have a positive redundancy 1/3, while response strategies of the label 1 and 4 have no redundancy. Thus we can randomly select a label from 2, 3, and 5 to assign the entire label of the offspring individual. As a result, the offspring individual may be labelled as (1,2,2), (1,2,3), or (1,2,5).

In summary, in order to use SRS to determine the probability of selecting different response strategies and to generate the new initial population, it is important to determine the individuals' labels in triplet during the evolution process of MOEA-OSD.

The reason why we assign a triplet label instead of a single number to an individual is as follows. Assume the individuals generated by RDI, MDI, LPS, FPS, and PPS are assigned a label of 1, 2, 3, 4, and 5, respectively. Now let us assume that the labels of the three parent individuals are 1, 4, and 5, and then we need to randomly select a label from 1, 4, and 5 to assign a label to the offspring individual, that is, (1) or (4) or (5). If the label of the offspring individual is randomly selected as 1, only RDI will be selected, FPS and PPS may be ignored outright, while RDI, FPS, and PPS are equal in this situation. Therefore, we assign a label (1,1,1) to the individual instead of (1) to increase the accuracy of selecting different response strategies.

3.4 Overall MOEA-OSD/SRS

A flowchart of the overall MOEA-OSD/SRS is given in Figure 2. Firstly, an initial population of size *N* is randomly generated (*POP*₀), and a set of *N* reference vectors *r*¹, *r*², ..., *r*^{*N*} evenly distributed in the whole objective space are generated. Then, the algorithm identifies the closest solution of each reference vector and initializes the reference point *Z*.

At the beginning of each generation, the similarity detection operator is applied to detect environmental changes. If a change is detected, SRS is triggered to select the best response strategy. Otherwise, MOEA-OSD continues to evolve using the given strategy. In the process, the assignment of the individual's labels is very important, as presented in Section 3.3.

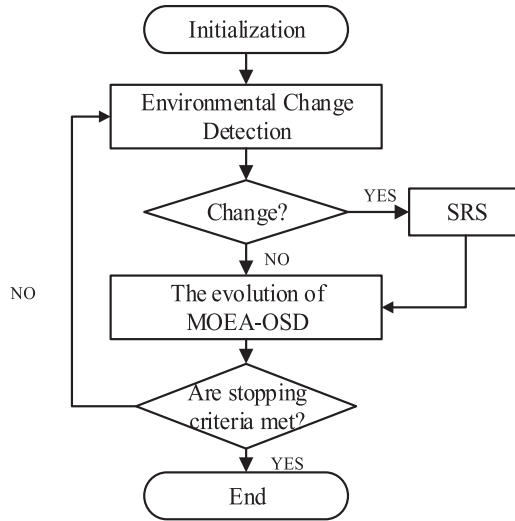


Figure 2: The flowchart of the proposed MOEA-OSD/SRS.

The above process is repeated until the termination criterion is met. When an environmental change is detected, the *PF* of the previous environment is outputted.

3.5 Computational Complexity

For MOEA-OSD/SRS, in each generation, computational resources are mainly consumed by MOEA-OSD.

As described in Algorithm 1, MOEA-OSD consists of the following components at each generation: finding the closest solution of each reference vector, performing genetic operations, evaluating the maxi-min fitness value, performing environmental selection and the update strategy. The time complexity for finding the closest solution of each reference vector is $O(m \times N^2)$, where m is the number of objectives and N is the population size. The computational complexity for the genetic operations is $O(m \times N)$. The calculation of the maxi-min fitness value holds a computational complexity of $O(m \times N^2)$. In addition, the computational complexity for environmental selection and the update strategy are $O(m \times N^2)$ and $O(m \times N)$, respectively.

To summarize, the total computational complexity of MOEA-OSD/SRS for one generation is $O(m \times N^2)$.

4 Empirical Studies

This section verifies the performance of MOEA-OSD/SRS by conducting comprehensive empirical studies, including a comparison of SRS with other six effective response strategies, a comparison of MOEA-OSD/SRS with other seven state-of-the-art dynamic multiobjective optimization algorithms, and a comparison of MOEA-OSD with other six state-of-the-art multiobjective optimization algorithms.

4.1 Benchmark Functions

Eleven benchmark functions F1-F11 are used to test the performance of MOEA-OSD/SRS. F1, F2, and F3 are *DMOP1*, *DMOP2*, and *DMOP3* of the *DMOP* suite (Goh and Tan, 2009a), F4, F5, F6, F7, and F8 are *FDA1*, *FDA2*, *FDA3*, *FDA4*, and *FDA5* of

the *FDA* suite (Farina et al., 2004), *F9* and *F10* are *JY2* and *JY5* of the *JY* suite (Jiang and Yang, 2017a), and *F11* (Zhou et al., 2014) is a function with noncyclic changes. The definitions and characteristics of those functions are given in Section S2 of the Supplementary material.

The changing dynamics of these benchmark functions is implemented by Eq. (8):

$$t = \frac{1}{n_T} \left\lfloor \frac{\tau}{\tau_T} \right\rfloor, \quad (8)$$

where n_T and τ_T is the severity and frequency of the environmental changes, respectively.

To indicate the performance of DMOAs in different types of environmental changes, we define the following settings of (τ_T, n_T) , including (10,10), (15,10), and (20,10). For each benchmark function, the environment changes 100 times in each run and each algorithm is performed 20 independent runs on each benchmark function.

4.2 Performance Indicators

In this work, we adopt the following performance indicators to assess the convergence and diversity performance of the solution sets obtained by the algorithms under comparison, which are Inverted generational distance (IGD) (Liu et al., 2017; Zhou et al., 2007), Spacing (Schott, 1995), and Hypervolume (Zitzler and Thiele, 1999). The description of the three performance indicators is presented in Section S3 of the Supplementary material.

4.3 Comparison between SRS and Other Six Effective Response Strategies

To assess the effectiveness of the proposed SRS, here we compared the performance of SRS with six most widely used response strategies on 11 benchmark functions.

4.3.1 Comparison Algorithms and Parameter Settings

This section verifies the better performance of SRS by conducting a comparison with six well-known response strategies, that are RDI, MDI, LPS, FPS, and PPS, and self-adaptive diversity introduction (SADI; Liu, Zheng et al., 2014). For comparisons, each of these response strategies is embedded into MOEA-OSD, which is denoted as MOEA-OSD/SRS, MOEA-OSD/RDI, MOEA-OSD/MDI, MOEA-OSD/LPS, MOEA-OSD/FPS, MOEA-OSD/PPS, and MOEA-OSD/SADI, respectively. Since SRS integrates five response strategies, that is, RDI, MDI, LPS, FPS, and PPS, we therefore compare SRS with the five strategies. We also compare SRS with SADI, as SADI is a self-adaptive response strategy, too. SADI adaptively determines the diversity introduction ratio according to the severity of environmental changes, which is described in Section S1 of the Supplementary material in detail. Most parameter settings of each strategy are based on the recommendations in the original references and the details are presented in Section S4.1 of the Supplementary material.

4.3.2 Experimental Results

Table 4 presents the statistical results of \overline{IGD} of the solution sets obtained by the seven algorithms averaged over 20 runs; \bar{S} and \overline{HV} metric values can be found in the Section S4.2 of Supplementary material. In Table 4, the best results of the first five algorithms using a different single response strategy are highlighted. These results indicate the best single response strategy among RDI, MDI, LPS, FPS, and PPS for each benchmark

Table 4: The statistical results of \overline{IGD} for different algorithms. “+”, “\$”, and “≈” denote the performance of MOEA-OSD/SRS is better than, worse than, or comparable with that of the best of the first five algorithms.

functions	(τ_T, n_T)	Statistic	MOEA-OSD /RDI	MOEA-OSD /MDI	MOEA-OSD /LPS	MOEA-OSD /FPS	MOEA-OSD /PPS	MOEA-OSD /SADI	MOEA-OSD /SRS	p-value	+/\$/≈
F1	(20,10)	Mean	0.0044	0.0043	0.0060	0.0043	0.0053	0.0149	0.0042	0.0947	≈
		Std	5.71E-04	2.17E-04	9.62E-05	4.83E-04	0.0012	0.0035	3.49E-05		
	(15,10)	Mean	0.0054	0.0049	0.0073	0.0051	0.0061	0.0255	0.0047	0.6761	≈
		Std	8.45E-04	4.05E-04	2.49E-04	6.58E-04	3.88E-04	0.0056	5.51E-05		
	(10,10)	Mean	0.0089	0.0080	0.0156	0.0079	0.0111	0.0507	0.0072	0.6791	≈
		Std	0.0016	0.0020	0.0023	0.0014	0.0024	0.0147	1.30E-04		
F2	(20,10)	Mean	0.0082	0.0077	0.0056	0.0071	0.0061	0.0088	0.0056	0.5309	≈
		Std	1.08E-04	7.07E-04	4.78E-05	1.23E-04	1.01E-04	1.40E-04	3.27E-05		
	(15,10)	Mean	0.0118	0.0107	0.0067	0.0096	0.0079	0.0139	0.0067	0.6761	≈
		Std	1.54E-04	1.51E-04	7.85E-05	2.31E-04	1.57E-04	3.60E-04	5.58E-05		
	(10,10)	Mean	0.0228	0.0192	0.0099	0.0169	0.0133	0.0297	0.0098	0.2446	≈
		Std	3.79E-04	7.51E-04	2.26E-04	7.60E-04	4.40E-04	0.0016	1.35E-04		
F3	(20,10)	Mean	0.0064	0.0071	0.0050	0.0060	0.0058	0.0064	0.0050	0.5006	≈
		Std	6.62E-05	6.81E-05	3.81E-05	7.46E-05	5.80E-05	1.23E-04	2.99E-05		
	(15,10)	Mean	0.0086	0.0085	0.0059	0.0079	0.0073	0.0085	0.0059	0.1984	≈
		Std	1.75E-04	1.43E-04	7.21E-05	1.47E-04	1.44E-04	1.00E-04	5.49E-05		
	(10,10)	Mean	0.0148	0.0150	0.0082	0.0127	0.0126	0.0142	0.0081	0.1258	≈
		Std	4.22E-04	6.64E-04	1.09E-04	5.16E-04	5.69E-04	4.64E-04	1.71E-04		
F4	(20,10)	Mean	0.0073	0.0069	0.0054	0.0065	0.0057	0.0088	0.0054	0.2446	≈
		Std	8.01E-05	7.48E-05	3.35E-05	9.65E-05	8.06E-05	1.40E-04	3.98E-05		
	(15,10)	Mean	0.0099	0.0091	0.0063	0.0084	0.0070	0.0113	0.0062	0.0235	+
		Std	1.18E-04	1.05E-04	5.12E-05	1.84E-04	1.68E-04	2.22E-03	1.44E-04		
	(10,10)	Mean	0.0180	0.0151	0.0087	0.0140	0.0108	0.0226	0.0086	0.0982	≈
		Std	4.18E-04	3.57E-04	1.59E-04	4.71E-04	3.39E-04	0.0011	2.09E-04		
F5	(20,10)	Mean	0.0060	0.0058	0.0073	0.0059	0.0065	0.0073	0.0057	0.0122	+
		Std	6.86E-05	1.40E-04	1.65E-04	9.86E-05	1.40E-04	6.93E-04	7.39E-05		
	(15,10)	Mean	0.0072	0.0066	0.0088	0.0068	0.0079	0.0095	0.0066	0.0367	+
		Std	1.88E-04	1.01E-04	1.81E-04	1.65E-04	3.76E-04	7.43E-04	1.23E-04		
	(10,10)	Mean	0.0100	0.0087	0.0130	0.0088	0.0121	0.0198	0.0085	0.5815	≈
		Std	2.03E-04	2.05E-04	2.98E-04	1.51E-04	0.0015	0.0036	1.54E-04		
F6	(20,10)	Mean	0.0131	0.0156	0.0084	0.0128	0.0109	0.0117	0.0083	0.5815	≈
		Std	8.68E-04	0.0016	4.69E-04	0.0016	9.33E-04	0.0011	7.04E-04		
	(15,10)	Mean	0.0186	0.0206	0.0108	0.0169	0.0160	0.0213	0.0109	0.2979	≈
		Std	0.0012	0.0014	8.34E-04	0.0020	0.0017	0.0029	0.0011		
	(10,10)	Mean	0.0273	0.0320	0.0159	0.0242	0.0271	0.0301	0.0160	0.2979	≈
		Std	9.16E-04	0.0025	8.79E-04	0.0018	0.0034	0.0017	0.0010		
F7	(20,10)	Mean	0.0577	0.0581	0.0541	0.0569	0.0575	0.0630	0.0468	0.0122	+
		Std	2.32E-04	2.27E-04	2.28E-04	4.09E-04	4.00E-04	3.15E-04	0.0011		
	(15,10)	Mean	0.0625	0.0633	0.0571	0.0612	0.0628	0.0676	0.0508	0.0027	+
		Std	2.93E-04	4.18E-04	2.45E-04	3.86E-04	8.25E-04	3.82E-04	0.0020		
	(10,10)	Mean	0.0732	0.0763	0.0630	0.0715	0.0804	0.0730	0.0572	0.0235	+
		Std	7.28E-04	0.0011	5.21E-04	0.0010	0.0033	0.0018	0.0040		
F8	(20,10)	Mean	0.0435	0.0446	0.0386	0.0422	0.0441	0.0838	0.0372	0.0963	≈
		Std	0.0011	0.0011	3.66E-04	0.0012	0.0012	0.0012	0.0025		
	(15,10)	Mean	0.0518	0.0539	0.0448	0.0491	0.0544	0.0863	0.0440	0.1542	≈
		Std	9.43E-04	0.0015	8.88E-04	0.0010	0.0018	0.0014	0.0026		
	(10,10)	Mean	0.0678	0.0781	0.0548	0.0654	0.0874	0.1001	0.0538	0.0583	≈
		Std	0.0024	0.0017	0.0019	0.0024	0.0085	0.0030	0.0054		
F9	(20,10)	Mean	0.0094	0.0086	0.0062	0.0075	0.0064	0.0109	0.0061	0.0216	+
		Std	1.28E-04	7.63E-04	4.58E-05	1.70E-04	1.08E-04	4.79E-04	7.86E-05		
	(15,10)	Mean	0.0133	0.0177	0.0074	0.0102	0.0079	0.0166	0.0071	0.0122	+
		Std	2.21E-04	1.40E-04	5.86E-05	3.49E-04	1.83E-04	5.95E-04	5.76E-05		
	(10,10)	Mean	0.0245	0.0198	0.0103	0.0170	0.0123	0.0324	0.0097	0.0216	+
		Std	8.78E-04	3.66E-04	1.85E-04	5.04E-04	3.27E-04	0.0027	3.33E-04		

Downloaded from http://direct.mit.edu/evco/article-pdf/29/4/491/1974878/evco_a_00289.pdf by guest on 08 September 2023

Table 4: Continued.

functions	(τ_T, n_T)	Statistic	MOEA	MOEA	MOEA	MOEA	MOEA	MOEA	MOEA	p-value	+/ §/ ≈
			-OSD /RDI	-OSD /MDI	-OSD /LPS	-OSD /FPS	-OSD /PPS	-OSD /SADI	-OSD /SRS		
F10	(20,10)	Mean	0.0062	0.0060	0.0068	0.0060	0.0062	0.0103	0.0060	0.8345	≈
		Std	2.79E-05	1.73E-05	2.18E-05	1.73E-05	4.93E-05	4.48E-04	3.24E-05		
	(15,10)	Mean	0.0065	0.0062	0.0076	0.0062	0.0064	0.0154	0.0062	0.0601	≈
		Std	3.05E-05	2.79E-05	6.39E-05	2.52E-05	7.65E-05	0.0010	3.80E-05		
	(10,10)	Mean	0.0073	0.0067	0.0094	0.0066	0.0072	0.0286	0.0067	0.1437	≈
		Std	9.04E-05	5.76E-05	1.31E-04	9.42E-05	2.10E-04	0.0030	3.01E-04		
F11	(20,10)	Mean	0.2911	0.4092	0.1174	0.1654	0.1393	0.0555	0.1156	0.0122	†
		Std	0.0217	0.0661	5.41E-04	0.0147	0.0435	0.0354	0.0033		
	(15,10)	Mean	0.3552	0.7125	0.1666	0.2651	0.3119	0.4015	0.1632	0.2453	≈
		Std	0.0227	0.0088	0.0015	0.0368	0.1604	0.0133	0.0038		
	(10,10)	Mean	0.5452	0.9708	0.2780	0.5454	0.6613	0.6536	0.2785	0.6985	≈
		Std	0.0178	0.1062	0.0249	0.0057	0.3908	0.0180	0.0116		

function in the presence of different environmental changes (columns 4 to 8). The experimental results (\overline{IGD}) of the proposed algorithm and the best of the first five algorithms are also analyzed using the Wilcoxon rank-sum test (Derrac et al., 2011). In Table 4, p-value represents the probability of two set of \overline{IGD} obtained by two different algorithms come from the same distribution, “†”, “§”, and “≈” denote the performance of the proposed algorithm is better than, worse than, or comparable to that of the best of the first five algorithms, respectively.

In Table 4, column 10 provides the results of MOEA-OSD/SRS in terms of \overline{IGD} for different settings, which are all as good as or better than the best results obtained by the five algorithms. From the statistical results of Wilcoxon rank-sum test, we can see that on these 11 benchmark functions with three environmental settings (33 instances in total), the proposed SRS significantly outperforms the best of the first five compared response strategies on 11 instances and performs equally well on the rest 22 instances.

All the above results confirm that SRS can indeed select the most suited response strategy for different benchmark functions, thereby being capable of solving DMOPs subject to unknown environmental changes.

In addition, the 9th column of Table 4 also presents the results of \overline{IGD} obtained by MOEA-OSD/SADI. Based on these results, we can conclude that MOEA-OSD/SRS has a better performance than MOEA-OSD/SADI on all instances.

Figure 3 presents the change of the probability of selecting different response strategies on F1 and F2 when (τ_T, n_T) is set to (10,10). We can see that the final selected response strategy matches the experimental results in Table 4; that is, MDI is the best response strategy for F1 because the PS of F1 is fixed, and LPS is the best for F2 in that the PS of F2 changes with time.

In addition, the average running time of 20 independent runs of all algorithms on all benchmark functions when (τ_T, n_T) is set to (10,10), is listed in Table 5 of the Supplementary material. On average, MOEA-OSD/SRS is a little bit time-consuming. This is reasonable since SRS needs to implement all five response strategies. When an environmental change occurs, the five response strategies all generate a new population, then SRS selects different ratios of individuals from each population to form a new population to respond to the environmental change. It should be pointed out that the increase in computational complexity might be negligible compared to the very time-consuming fitness evaluations in many expensive real-world optimization problems (Jin and Sendhoff, 2002; Jin et al., 2018).

Table 5: The statistical results of \overline{IGD} for different algorithms. “+”, “S”, and “≈” denote the performance of MOEA-OSD/SRS is better than, worse than, and comparable with that of the corresponding comparison algorithm.

functions	(τ_r, n_r)	Statistic	DNSGA		FPS	PPS	SGEA	MOEA/D	Immune	MOEA
			-II-A	-II-B	-RM -MEDA	-RM -MEDA			GDE3	-OSD /SRS
F1	(20,10)	Mean	0.0158 †	0.0124 †	0.009 5 †	0.0095 †	0.0087 †	0.0101 †	0.0094 †	0.0042
		Std	0.0075	0.0064	0.0025	0.0025	0.0022	0.0040	0.0020	3.49E-05
	(15,10)	Mean	0.0238 †	0.0194 †	0.0128 †	0.0127 †	0.0112 †	0.0145 †	0.0120 †	0.0047
		Std	0.0088	0.0078	0.0020	0.0025	0.0028	0.0028	0.0088	5.51E-05
	(10,10)	Mean	0.0360 †	0.0307 †	0.0244 †	0.0253 †	0.0156 †	0.0264 †	0.0220 †	0.0072
		Std	0.0125	0.0097	0.0064	0.0125	0.0039	7.78E-04	0.0125	1.30E-04
F2	(20,10)	Mean	0.0113 †	0.0111 †	0.0109 †	0.0093 †	0.0074 †	0.0098 †	0.0089 †	0.0056
		Std	5.83E-04	2.76E-04	3.00E-05	6.03E-04	5.57E-04	1.31E-04	6.42E-04	3.27E-05
	(15,10)	Mean	0.0168 †	0.0169 †	0.0197 †	0.0171 †	0.0096 †	0.0115 †	0.0101 †	0.0067
		Std	5.69E-04	5.78E-04	9.71E-04	0.0020	8.70E-04	7.43E-04	5.69E-04	5.58E-05
	(10,10)	Mean	0.0368 †	0.0381 †	0.0430 †	0.0973 †	0.0144 †	0.0236 †	0.0153 †	0.0098
		Std	0.0024	0.0035	0.0021	0.0444	9.15E-04	0.0074	0.0014	1.35E-04
F3	(20,10)	Mean	0.0099 †	0.0096 †	0.0084 †	0.0119 †	0.0718 †	0.0088 †	0.0085 †	0.0050
		Std	7.23E-04	5.75E-05	2.92E-04	9.47E-04	0.0088	1.63E-04	6.16E-04	2.99E-05
	(15,10)	Mean	0.0158 †	0.0163 †	0.0116 †	0.0229 †	0.0869 †	0.0121 †	0.0118 †	0.0059
		Std	0.0010	0.0011	4.06E-04	0.0015	0.0103	5.09E-04	0.0020	5.49E-05
	(10,10)	Mean	0.0508 †	0.0542 †	0.0210 †	0.0655 †	0.1303 †	0.0256 †	0.0189 †	0.0081
		Std	0.0031	0.0064	0.0010	0.0114	0.0105	0.0049	0.0018	1.71E-04
F4	(20,10)	Mean	0.0099 †	0.0096 †	0.0084 †	0.0119 †	0.0718 †	0.0088 †	0.0085 †	0.0050
		Std	3.05E-04	2.54E-05	1.60E-04	4.02E-04	7.63E-04	5.42E-04	1.41E-04	3.98E-05
	(15,10)	Mean	0.0137 †	0.0141 †	0.0129 †	0.0130 †	0.0091 †	0.0121 †	0.0108 †	0.0062
		Std	5.41E-04	9.29E-04	4.97E-04	0.0014	0.0012	3.05E-04	5.86E-04	1.44E-04
	(10,10)	Mean	0.0299 †	0.0297 †	0.0266 †	0.0626 †	0.0135 †	0.0263 †	0.0228 †	0.0086
		Std	0.0026	0.0019	0.0012	0.0233	0.0012	0.0016	0.0026	2.09E-04
F5	(20,10)	Mean	0.1547 †	0.1548 †	0.0071 †	0.0107 †	0.0067 †	0.0087 †	0.0079 †	0.0057
		Std	8.89E-05	2.90E-04	1.54E-04	0.0012	0.0011	8.47E-05	7.75E-04	7.39E-05
	(15,10)	Mean	0.1553 †	0.1552 †	0.0079 †	0.0153 †	0.0084 †	0.0094 †	0.0088 †	0.0066
		Std	2.65E-04	1.77E-04	2.20E-04	0.0014	0.0028	1.93E-04	8.95E-04	1.23E-04
	(10,10)	Mean	0.1565 †	0.1564 †	0.0111 †	0.0385 †	0.0099 †	0.0138 †	0.0118 †	0.0085
		Std	6.42E-04	7.24E-04	3.88E-04	0.0084	0.0014	0.0021	5.58E-04	1.54E-04
F6	(20,10)	Mean	0.1436 †	0.1435 †	0.0110 †	0.0173 †	0.0108 †	0.0153 †	0.0144 †	0.0083
		Std	0.0043	0.0041	5.57E-04	0.0020	0.0042	2.84E-04	0.0010	7.04E-04
	(15,10)	Mean	0.1516 †	0.1461 †	0.0114 †	0.0362 †	0.0134 †	0.0176 †	0.0153 †	0.0109
		Std	0.0090	0.0039	4.53E-04	0.0075	0.0033	0.0012	0.0040	0.0011
	(10,10)	Mean	0.1713 †	0.1872 †	0.0199 †	0.1972 †	0.0175 †	0.0213 †	0.0195 †	0.0160
		Std	0.0594	0.0707	0.0011	0.0582	0.0051	0.0046	0.0018	0.0010
F7	(20,10)	Mean	0.1372 †	0.1585 †	0.0994 †	0.0988 †	0.0764 †	0.0910 †	0.0795 †	0.0468
		Std	0.0122	0.0091	0.0012	0.0016	0.0012	0.0076	0.0012	0.0011
	(15,10)	Mean	0.2255 †	0.2424 †	0.1033 †	0.1076 †	0.0907 †	0.1053 †	0.0948 †	0.0508
		Std	0.0227	0.0218	0.0017	0.0020	0.0018	0.0021	0.0020	0.0020
	(10,10)	Mean	0.4007 †	0.4180 †	0.1138 †	0.1303 †	0.1307 †	0.1206 †	0.1181 †	0.0572
		Std	0.0348	0.0324	0.0016	0.0067	0.0058	0.0034	0.0040	0.0040
F8	(20,10)	Mean	0.2609 †	0.2595 †	0.0998 †	0.0988 †	0.0622 †	0.0713 †	0.0680 †	0.0372
		Std	0.0074	0.0078	0.0043	0.0016	0.0044	0.0018	0.0017	0.0025
	(15,10)	Mean	0.2797 †	0.2833 †	0.0948 †	0.1059 †	0.0746 †	0.0833 †	0.0785 †	0.0440
		Std	0.0073	0.0080	0.0017	0.0073	0.0045	0.0042	0.0043	0.0026
	(10,10)	Mean	0.3504 †	0.3566 †	0.0840 †	0.1204 †	0.0980 †	0.1072 †	0.1001 †	0.0538
		Std	0.0143	0.0145	0.0016	0.0093	0.0079	0.0044	0.0041	0.0054
F9	(20,10)	Mean	0.0153 †	0.0156 †	0.0468 †	0.0541 †	0.0077 †	0.0098 †	0.0095 †	0.0061
		Std	6.25E-04	7.24E-04	1.28E-04	0.0155	1.76E-04	1.77E-04	4.55E-04	7.86E-05
	(15,10)	Mean	0.0257 †	0.0255 †	0.0471 †	0.0477 †	0.0103 †	0.0114 †	0.0110 †	0.0071
		Std	0.0012	0.0013	2.01E-04	3.91E-04	3.05E-04	0.0013	0.0010	5.76E-05
	(10,10)	Mean	0.0607 †	0.0625 †	0.0514 †	0.0667 †	0.0158 †	0.0159 †	0.0168 †	0.0097
		Std	0.0048	0.0054	6.13E-04	0.0141	4.33E-04	0.0026	0.0030	3.33E-04

Downloaded from http://direct.mit.edu/evco/article-pdf/29/4/491/1974878/evco_a_00289.pdf by guest on 08 September 2023

Table 5: Continued.

functions	(τ_T, n_T)	Statistic	DNSGA		FPS	PPS	SGEA	MOEA/D	Immune	MOEA
			-II-A	-II-B	-RM -MEDA	-RM -MEDA			GDE3	-OSD /SRS
F10	(20,10)	Mean	0.0067 ≈	0.0066 ≈	0.0058 ≈	0.0058 ≈	0.0042 §	0.0063 ≈	0.0060 ≈	0.0060
		Std	5.62E-04	4.45E-04	3.14E-05	5.51E-05	1.25E-05	9.70E-05	3.85E-04	3.24E-05
	(15,10)	Mean	0.0060 ≈	0.0062 ≈	0.0060 ≈	0.0062 ≈	0.0043 §	0.0068 ≈	0.0065 ≈	0.0062
		Std	3.17E-04	3.82E-04	4.25E-05	8.53E-05	2.35E-05	2.89E-04	4.28E-04	3.80E-05
	(10,10)	Mean	0.0059 §	0.0060 ≈	0.0067 ≈	0.0073 †	0.0044 ≈	0.0087 †	0.0070 ≈	0.0067
		Std	3.60E-04	2.91E-04	1.10E-04	4.03E-04	5.45E-05	6.01E-04	5.55E-04	3.01E-04
F11	(20,10)	Mean	0.4826 †	0.5418 †	0.2260 †	0.2560 †	0.1241 †	0.8585 †	0.3746 †	0.1156
		Std	0.0127	0.0242	0.0035	0.0053	0.0019	0.0160	0.0252	0.0033
	(15,10)	Mean	0.5971 †	0.5576 †	0.3376 †	0.3988 †	0.1889 †	0.8673 †	0.2150 †	0.1632
		Std	0.0269	0.0331	0.0046	0.0101	0.0154	0.1085	0.0146	0.0038
	(10,10)	Mean	0.9034 †	0.8868 †	0.7904 †	0.8852 †	0.4915 †	0.8484 †	0.5976 †	0.2785
		Std	0.1505	0.0053	0.0215	0.0232	0.0718	0.0975	0.0127	0.0116

4.4 Comparison between MOEA-OSD/SRS and the Other Seven DMOAs

To evaluate the performances of the entire MOEA-OSD/SRS for dynamic multiobjective optimization, we compare it with seven popular DMOAs.

4.4.1 Compared Algorithms and Parameter Settings

To further demonstrate the effectiveness of MOEA-OSD/SRS, we compare it with the other seven state-of-the-art DMOAs, which are DNSGA-II-A (Deb et al., 2007), DNSGA-II-B (Deb et al., 2007), RM-MEDA based on FPS (denoted as FPS-RM-MEDA) (Zhou et al., 2014), RM-MEDA based on PPS (denoted as PPS-RM-MEDA) (Zhou et al., 2014), SGEA (Jiang and Yang, 2017b), and multiobjective evolutionary algorithm based on decomposition (MOEA/D) (Zhang and Li, 2007), and Immune Generalized Differential Evolution 3 (Immune-GDE3) (Martínez-Peñalosa and Mezura-Montes, 2018). The detailed parameter settings can be found in Section S5.1 of the Supplementary material.

4.4.2 Experimental Results of \overline{IGD} , \overline{S} , and \overline{HV}

Tables 5–7 list the statistical results of \overline{IGD} , \overline{S} , and \overline{HV} obtained by MOEA-OSD/SRS and seven compared algorithms over 20 runs when (τ_T, n_T) is set as (10,10), (15,10), and (20,10). In those tables, the results of the best performing algorithm, that is, the one obtaining the smallest mean values of \overline{IGD} , the smallest mean values of \overline{S} , and the largest values of \overline{HV} , are highlighted. The Wilcoxon rank-sum test (Derrac et al., 2011) is conducted to compare the significance of difference between MOEA-OSD/SRS and the seven compared algorithms. Since it is necessary to compare the performance difference between MOEA-OSD/SRS and each of comparison algorithm, in order to better present the experimental results, we just use symbols “†”, “§”, and “≈” to express the statistical analysis results. “†”, “§”, and “≈” indicate that the performance of MOEA-OSD/SRS is better than, worse than, or comparable to that of the algorithm under comparison, respectively.

Meanwhile, it can be seen from Tables 5–7, among the three different types of environmental changes, when (τ_T, n_T) is set as (20,10), the performance of different algorithms is better than the situations that (τ_T, n_T) is set as (10,10) and (15,10), and the performance of the algorithm is getting better and better with the increasing of τ_T .

a) From Table 5 (\overline{IGD}): we observe that MOEA-OSD/SRS outperforms other algorithms on most benchmark functions, except on F10, while SGEA performs significantly better than all compared algorithms on F10. For F10, as seen from Table 4,

Table 6: The statistical results of \bar{S} for different algorithms. “+”, “§”, and “≈” denote the performance of MOEA-OSD/SRS is better than, worse than, and comparable with that of the corresponding comparison algorithm.

functions	(τ_T, n_T)	Statistic	DNSGA		FPS	PPS	SGEA	MOEA/D	Immune	MOEA
			-II-A	-II-B	-RM -MEDA	-RM -MEDA				
F1	(20,10)	Mean	0.0089 †	0.0089 †	0.0109 †	0.0119 †	0.0059 †	0.0080 †	0.0075 †	0.0058
		Std	0.0017	0.0017	0.0151	0.0039	1.70E-04	1.59E-04	2.60E-04	9.94E-05
	(15,10)	Mean	0.0145 †	0.0123 †	0.0145 †	0.0196 †	0.0077 †	0.0087 †	0.0082 †	0.0064
		Std	0.0045	0.0044	0.0371	0.0103	1.15E-04	8.46E-04	0.0088	1.14E-04
	(10,10)	Mean	0.0210 †	0.0206 †	0.0195 †	0.0171 †	0.0085 †	0.0099 †	0.0095 †	0.0075
		Std	0.0053	0.0099	0.0385	0.0144	5.62E-04	1.85E-04	0.0125	1.20E-04
F2	(20,10)	Mean	0.0073 †	0.0073 †	0.0222 †	0.0077 †	0.0073 †	0.0085 †	0.0080 †	0.0069
		Std	1.28E-04	1.15E-04	0.0089	1.97E-04	1.79E-04	1.06E-04	5.83E-04	5.88E-05
	(15,10)	Mean	0.0098 †	0.0098 †	0.0437 †	0.0123 †	0.0086 †	0.0101 †	0.0092 †	0.0076
		Std	3.21E-04	2.23E-04	0.0133	0.0011	2.49E-04	1.21E-04	5.69E-04	4.92E-05
	(10,10)	Mean	0.0168 †	0.0166 †	0.0919 †	0.0243 †	0.0093 †	0.0115 †	0.0100 †	0.0090
		Std	9.60E-04	7.80E-04	0.0161	0.0043	1.91E-04	5.23E-04	4.58E-04	7.43E-05
F3	(20,10)	Mean	0.0068 §	0.0067 §	0.0155 †	0.0079 §	0.0060 §	0.0115 †	0.0099 §	0.0108
		Std	1.66E-04	1.07E-05	0.0016	0.0010	4.30E-04	8.81E-05	7.23E-04	7.99E-05
	(15,10)	Mean	0.008 §	0.0088 §	0.0155 †	0.0111 ≈	0.0062 §	0.0158 †	0.0088 §	0.0112
		Std	1.35E-04	1.69E-04	0.0050	0.0022	5.28E-04	1.61E-04	0.0010	5.01E-05
	(10,10)	Mean	0.0164 †	0.0166 †	0.0234 †	0.0153 †	0.0075 §	0.0145 †	0.0080 §	0.0116
		Std	6.98E-04	6.11E-04	0.0051	0.0011	4.79E-04	3.60E-04	0.0031	4.72E-05
F4	(20,10)	Mean	0.0067 §	0.0067 §	0.0118 †	0.0074 §	0.0050 §	0.0118 †	0.0065 §	0.0109
		Std	1.30E-04	1.03E-04	0.0018	0.0012	8.94E-05	8.83E-05	3.05E-04	6.80E-05
	(15,10)	Mean	0.0083 §	0.0083 §	0.0387 †	0.0092 §	0.0062 §	0.0135 †	0.0085 §	0.0112
		Std	2.87E-04	2.19E-04	0.0128	8.80E-04	2.48E-04	4.52E-04	5.41E-04	1.50E-04
	(10,10)	Mean	0.0137 †	0.0137 †	0.0650 †	0.0236 †	0.0089 §	0.0154 †	0.0120 †	0.0117
		Std	7.39E-04	8.04E-04	0.0211	0.0085	3.86E-04	1.92E-03	0.0026	8.76E-05
F5	(20,10)	Mean	0.0066 §	0.0066 §	0.0086 §	0.0064 §	0.0053 §	0.0123 †	0.0084 §	0.0113
		Std	1.17E-04	7.90E-05	0.0017	2.92E-04	2.26E-04	2.04E-04	8.89E-04	7.24E-05
	(15,10)	Mean	0.0066 §	0.0066 §	0.0089 §	0.0068 §	0.0058 §	0.0138 †	0.0088 §	0.0115
		Std	1.76E-04	2.02E-04	0.0016	1.89E-04	1.98E-04	2.28E-04	2.65E-04	3.70E-05
	(10,10)	Mean	0.0069 §	0.0068 §	0.0137 †	0.0100 §	0.0071 §	0.0155 †	0.0089 §	0.0118
		Std	3.54E-04	2.81E-04	0.0034	5.88E-04	2.44E-04	1.91E-04	6.42E-04	5.33E-05
F6	(20,10)	Mean	0.0296 †	0.0305 †	0.0184 †	0.0167 †	0.0456 †	0.0432 †	0.0285 †	0.0128
		Std	0.0079	0.0081	0.0028	0.0027	0.0011	0.0075	0.0043	2.03E-04
	(15,10)	Mean	0.0229 †	0.0319 †	0.0300 †	0.0288 †	0.0472 †	0.0577 †	0.0299 †	0.0131
		Std	0.0133	0.0071	0.0081	0.0087	7.51E-04	0.0019	0.0090	7.09E-05
	(10,10)	Mean	0.0383 †	0.0350 †	0.0647 †	0.0626 †	0.0542 †	0.0690 †	0.0305 †	0.0142
		Std	0.0117	0.0136	0.0154	0.0144	0.0013	0.0019	0.0059	3.70E-04
F7	(20,10)	Mean	0.0667 †	0.0686 †	0.1118 †	0.0705 †	0.0332 §	0.0925 †	0.0825 †	0.0553
		Std	0.0014	0.0018	0.0124	0.0016	6.52E-04	0.0012	0.0012	0.0016
	(15,10)	Mean	0.0801 †	0.0794 †	0.1239 †	0.0837 †	0.0398 §	0.0927 †	0.0766 †	0.0586
		Std	0.0015	0.0029	0.0095	0.0079	0.0020	1.99E-04	0.0022	0.0020
	(10,10)	Mean	0.0989 †	0.0982 †	0.1408 †	0.0975 †	0.0529 §	0.0930 †	0.0959 †	0.0617
		Std	0.0040	0.0029	0.0069	0.0075	0.0016	6.19E-04	0.0034	7.90E-04
F8	(20,10)	Mean	0.1008 †	0.1002 †	0.1537 †	0.0804 §	0.0436 §	0.1451 †	0.0899 §	0.0999
		Std	0.0013	0.0012	0.0066	0.0044	6.05E-04	0.0012	0.0074	0.0024
	(15,10)	Mean	0.1056 †	0.1076 †	0.1239 †	0.1403 †	0.0470 §	0.1488 †	0.0988 †	0.1035
		Std	0.0020	0.0014	0.0095	0.0052	6.20E-04	0.0015	0.0073	0.0022
	(10,10)	Mean	0.1188 †	0.1181 †	0.1869 †	0.1539 †	0.0550 §	0.1551 †	0.1095 †	0.1083
		Std	0.0021	0.0014	0.0071	0.0068	0.0015	0.0016	0.0143	0.0027
F9	(20,10)	Mean	0.0114 †	0.0116 †	0.0177 †	0.0086 ≈	0.0064 §	0.0188 †	0.0099 †	0.0085
		Std	6.02E-04	5.46E-04	0.0058	7.97E-04	1.44E-04	1.74E-04	6.25E-04	5.75E-05
	(15,10)	Mean	0.0157 †	0.0153 †	0.0315 †	0.0131 †	0.0084 §	0.0320 †	0.0115 †	0.0091
		Std	0.0012	7.15E-04	0.0058	0.0012	2.70E-04	2.05E-04	0.0012	4.07E-05
	(10,10)	Mean	0.0271 †	0.0273 †	0.0640 †	0.0268 †	0.0124 †	0.0642 †	0.0138 †	0.0099
		Std	0.0014	0.0012	0.0077	0.0056	3.47E-04	2.38E-04	0.0048	4.95E-05

Downloaded from http://direct.mli.edu/co/article-pdf/29/4/491/1974878/evco_a_00289.pdf by guest on 08 September 2023

Table 6: Continued.

functions	(τ_T, n_T)	Statistic	DNSGA		FPS	PPS	SGEA	MOEA/D	Immune	MOEA
			-II-A	-II-B	-RM -MEDA	-RM -MEDA			GDE3	-OSD /SRS
F10	(20,10)	Mean	0.0113 ≈	0.0117 †	0.0090 §	0.0065 §	0.0034 §	0.0128 †	0.0098 §	0.0108
		Std	0.0026	0.0040	0.0015	1.48E-04	6.68E-05	1.27E-04	5.62E-04	6.55E-05
	(15,10)	Mean	0.0142 †	0.0157 †	0.0130 †	0.0069 §	0.0037 §	0.0158 †	0.0106 §	0.0111
		Std	0.0025	0.0031	0.0034	4.19E-04	5.36E-04	3.01E-05	3.17E-04	9.43E-05
	(10,10)	Mean	0.0213 †	0.0207 †	0.0221 †	0.0082 §	0.0039 §	0.0209 †	0.0125 ≈	0.0118
		Std	0.0049	0.0042	0.0052	8.03E-04	5.53E-4	2.23E-04	3.60E-04	1.03E-04
F11	(20,10)	Mean	0.0828 †	0.0801 †	0.6646 †	0.4047 †	0.0649 †	0.0885 †	0.0778 †	0.0633
		Std	0.0088	0.0092	0.0572	0.0326	0.0207	0.0060	0.0019	0.0032
	(15,10)	Mean	0.1054 †	0.1566 †	0.7423 †	0.5610 †	0.0907 †	0.1083 †	0.1022 †	0.0807
		Std	0.0132	0.0225	0.0656	0.0329	0.0134	0.0043	0.0102	0.0113
	(10,10)	Mean	0.2448 †	0.2996 †	0.7673 †	0.5560 †	0.1623 †	0.2480 †	0.2470 †	0.1323
		Std	0.0168	0.0261	0.0427	0.0440	0.0238	0.0279	0.0062	0.0155

MOEA-OSD/SRS is able to select the most effective response strategy for F10, so the poor performance of MOEA-OSD/SRS on F10 may be attributed to the poor performance of the static algorithm MOEA-OSD on F10.

b) From Table 6 (\bar{S}): it is noted that MOEA-OSD/SRS obtains the best results on F1, F2, F6, and F11, while SGEA performs the best on other seven benchmark functions. It implies that MOEA-OSD/SRS only maintains better distribution on few benchmark functions, and MOEA-OSD/SRS is inferior to SGEA. From these experimental results, we can conclude that MOEA-OSD/SRS should incorporate some strategy to improve the diversity when dealing with DMOPs.

c) From Table 7 (\overline{HV}): we can see that the \overline{HV} values obtained by the compared algorithms are largely consistent with the \overline{IGD} values presented in Table 7. Clearly, MOEA-OSD/SRS is more promising than the compared algorithms for solving DMOPs, although it is outperformed by SGEA on F10. The possible reason for this has been discussed previously in the analysis of \overline{IGD} .

Last but not least, recall that \overline{IGD} and \overline{HV} measure the performance of a solution set in terms of both convergence and distribution. As a result, if an algorithm achieves solutions with a poor performance in diversity, it is possible that it exhibits a good \overline{IGD} and \overline{HV} value on account of a superior performance in convergence.

Meanwhile, MOEA-OSD/SRS is relatively better than the seven compared algorithms on F11, which is a nonperiodic function, although all the algorithms are not able to obtain better performance in terms of \overline{IGD} , \bar{S} , and \overline{HV} .

4.4.3 Comparison of the Tracking Ability of the Eight Algorithms

It is necessary to investigate how an algorithm performs after changes and in different environments after a change (Helbig and Engelbrecht, 2013a). Figure 4 provides the IGD over time averaged over 20 runs when (τ_T, n_T) is set as (10,10). It can be clearly seen that MOEA-OSD/SRS responds to the changes more stably and recovers faster on most of the benchmark functions than the compared algorithms, thereby obtaining better convergence performance. The only exception occurs on F10, where SGEA performs the best, the \overline{IGD} value obtained by MOEA-OSD/SRS fluctuates strongly, probably due to poor convergence and diversity when a change occurs.

In addition to the above experimental results, the comparison results of the obtained PF of the eight compared algorithms are also presented in Section S5.2 of the Supplementary material. Meanwhile, the average runtime of 20 independent runs of the

Table 7: The statistical results of \overline{HV} for different algorithms. “+”, “S”, and “≈” denote the performance of MOEA-OSD/SRS is better than, worse than, and comparable with that of the corresponding comparison algorithm.

functions	(τ_T, n_T)	Statistic	DNSGA	DNSGA	FPS	PPS	SGEA	MOEA/D	Immune	MOEA
			-II-A	-II-B	-RM -MEDA	-RM -MEDA			GDE3	-OSD /SRS
F1	(20,10)	Mean	0.6436 †	0.6405 †	0.6456 †	0.6448 †	0.6488 †	0.6434 †	0.6425 †	0.6550
		Std	0.0047	0.0017	8.76E-04	0.0015	2.35E-04	5.69E-04	6.75E-04	6.69E-05
	(15,10)	Mean	0.6411 †	0.6403 †	0.6429 †	0.6439 †	0.6446 †	0.6391 †	0.6399 †	0.6538
		Std	0.0023	0.0029	6.85E-04	1.59E-04	6.11E-04	0.0030	0.0040	1.59E-04
	(10,10)	Mean	0.6368 †	0.6330 †	0.6356 †	0.6321 †	0.6435 †	0.6356 †	0.6365 †	0.6500
		Std	0.0039	0.0024	0.0015	0.0018	0.0029	1.04E-04	0.0054	3.63E-04
F2	(20,10)	Mean	0.6404 †	0.6403 †	0.6398 †	0.6424 †	0.6468 †	0.6417 †	0.6406 †	0.6511
		Std	3.19E-04	7.82E-04	0.0010	2.97E-04	0.0010	1.62E-04	5.83E-04	6.37E-05
	(15,10)	Mean	0.6303 †	0.6299 †	0.6231 †	0.6254 †	0.6426 †	0.6250 †	0.6250 †	0.6488
		Std	7.78E-04	6.27E-04	0.0013	0.0032	0.0016	0.0011	5.69E-04	1.43E-04
	(10,10)	Mean	0.5962 †	0.5888 †	0.5800 †	0.5894 †	0.6340 †	0.5846 †	0.5885 †	0.6429
		Std	0.0041	0.0121	0.0011	0.0270	8.67E-04	0.0079	0.0024	2.93E-04
F3	(20,10)	Mean	0.8597 †	0.8586 †	0.8611 †	0.8542 †	0.7645 †	0.7694 †	0.8588 †	0.8682
		Std	9.54E-04	4.75E-05	3.82E-04	0.0012	0.0106	1.02E-04	7.23E-04	7.26E-05
	(15,10)	Mean	0.8491 †	0.8491 †	0.8557 †	0.8383 †	0.7312 †	0.7606 †	0.8221 †	0.8665
		Std	9.74E-04	0.0025	1.61E-04	0.0031	0.0045	8.70E-04	0.0010	1.12E-04
	(10,10)	Mean	0.7949 †	0.7904 †	0.8392 †	0.7627 †	0.6759 †	0.7502 †	0.7895 †	0.8619
		Std	0.0042	0.0045	0.0012	0.0020	0.0034	1.56E-04	0.0031	1.89E-04
F4	(20,10)	Mean	0.8607 †	0.8605 †	0.8617 †	0.8632 †	0.8646 †	0.8602 †	0.8622 †	0.8674
		Std	2.11E-04	5.38E-05	1.94E-04	1.64E-04	1.23E-04	2.04E-04	3.05E-04	3.86E-05
	(15,10)	Mean	0.8516 †	0.8526 †	0.8531 †	0.8492 †	0.8611 †	0.8529 †	0.8541 †	0.8658
		Std	0.0011	2.45E-04	9.69E-04	0.0020	1.63E-04	1.41E-04	5.41E-04	3.37E-04
	(10,10)	Mean	0.8216 †	0.8250 †	0.8313 †	0.8295 †	0.8539 †	0.8301 †	0.8342 †	0.8612
		Std	0.0054	0.0053	0.0021	0.0011	8.65E-04	0.0016	0.0026	1.73E-04
F5	(20,10)	Mean	0.7030 †	0.7029 †	0.6598 †	0.6516 †	0.7034 †	0.6256 †	0.6556 †	0.7043
		Std	1.32E-04	1.39E-05	2.55E-04	0.0011	8.67E-04	1.76E-05	8.89E-04	1.00E-04
	(15,10)	Mean	0.7005 †	0.7011 †	0.6579 †	0.6459 †	0.7006 †	0.6239 †	0.6531 †	0.7029
		Std	2.63E-04	1.02E-04	6.92E-04	7.59E-04	0.0028	6.34E-04	2.65E-04	1.22E-04
	(10,10)	Mean	0.6966 †	0.6964 †	0.6529 †	0.6037 †	0.6986 †	0.6175 †	0.6428 †	0.6997
		Std	1.08E-04	1.77E-04	2.91E-04	0.0020	9.37E-04	0.0024	6.42E-04	1.28E-04
F6	(20,10)	Mean	1.0147 †	1.0154 †	1.0170 †	0.9985 †	1.0169 †	0.9788 †	0.9988 †	1.0214
		Std	0.0038	0.0018	0.0017	0.0055	0.0079	9.14E-04	0.0043	0.0056
	(15,10)	Mean	1.0010 †	1.0060 †	1.0040 †	0.9923 †	1.0105 †	0.9773 †	0.9956 †	1.0115
		Std	0.0126	0.0116	0.0020	0.0107	0.0034	0.0049	0.0090	0.0026
	(10,10)	Mean	0.9988 †	0.9985 †	0.9934 †	0.9857 †	1.0015 †	0.9758 †	0.9915 †	1.0056
		Std	0.0083	0.0038	0.0117	0.0518	0.0079	0.0048	0.0059	0.0010
F7	(20,10)	Mean	0.5294 †	0.5201 †	0.5886 †	0.5821 †	0.6724 †	0.5977 †	0.6022 †	0.7351
		Std	0.0214	0.0102	0.0021	0.0026	9.74E-04	3.31E-04	0.0021	1.50E-04
	(15,10)	Mean	0.4084 †	0.4119 †	0.5776 †	0.5642 †	0.6524 †	0.5887 †	0.5999 †	0.7220
		Std	0.0026	0.0228	3.53E-04	0.0018	0.0027	2.22E-04	0.0022	0.0076
	(10,10)	Mean	0.2902 †	0.3036 †	0.5571 †	0.5007 †	0.6416 †	0.5690 †	0.5901 †	0.7166
		Std	0.0057	0.0033	0.0065	0.0070	0.0010	5.84E-04	0.0034	2.40E-04
F8	(20,10)	Mean	3.0936 †	3.0475 †	2.8990 †	2.7988 †	3.4270 †	3.2993 †	3.3345 †	3.5342
		Std	0.0642	0.0467	0.0174	0.0165	0.0044	0.0013	0.0074	0.0019
	(15,10)	Mean	2.8396 †	2.8047 †	2.7445 †	2.4886 †	3.3664 †	3.2760 †	3.2985 †	3.4784
		Std	0.0019	0.0532	0.0517	0.1279	0.0023	0.0011	0.0073	0.0023
	(10,10)	Mean	2.4662 †	2.4445 †	2.3414 †	2.0190 †	3.2432 †	3.2101 †	3.2564 †	3.4415
		Std	0.0089	0.0874	0.0213	0.0126	0.0084	0.0053	0.0143	0.0475
F9	(20,10)	Mean	0.6891 †	0.6878 †	0.6937 †	0.6979 †	0.6999 †	0.6874 †	0.6957 †	0.7022
		Std	9.54E-04	1.36E-04	2.02E-04	2.72E-04	1.09E-04	8.49E-04	6.25E-04	2.17E-05
	(15,10)	Mean	0.6730 †	0.6689 †	0.6819 †	0.6904 †	0.6964 †	0.6246 †	0.6942 †	0.7011
		Std	0.0014	0.0054	7.05E-04	3.47E-04	1.75E-04	0.0013	0.0012	2.72E-04
	(10,10)	Mean	0.6117 †	0.6055 †	0.6508 †	0.6307 †	0.6893 †	0.6173 †	0.6921 †	0.7001
		Std	0.0022	0.0081	0.0035	0.0082	2.69E-04	0.0041	0.0048	2.17E-04

Downloaded from http://direct.milieu.edu.co/article-pdf/29/4/491/1974878/evco_a_00289.pdf by guest on 08 September 2023

Table 7: Continued.

functions	(τ_T, n_T)	Statistic	DNSGA		FPS	PPS	SGEA	MOEA/D	Immune	MOEA
			-II-A	-II-B	-RM -MEDA	-RM -MEDA			GDE3	-OSD /SRS
F10	(20,10)	Mean	0.7061 †	0.7066 †	0.7057 †	0.7054 †	0.7090 ≈	0.7064 †	0.7068 †	0.7086
		Std	5.67E-04	3.21E-04	1.18E-05	8.30E-05	3.96E-05	1.42E-04	5.62E-04	2.17E-05
	(15,10)	Mean	0.7043 †	0.7041 †	0.7053 †	0.7046 †	0.7089 ≈	0.7048 †	0.7049 †	0.7082
		Std	1.38E-04	3.77E-04	8.95E-05	5.22E-04	2.08E-05	4.39E-04	3.17E-04	1.64E-04
	(10,10)	Mean	0.7021 †	0.7027 †	0.7034 †	0.7017 †	0.7084 §	0.7032 †	0.7038 †	0.7064
		Std	2.31E-04	3.05E-04	4.33E-04	4.06E-04	9.22E-05	3.29E-04	3.60E-04	0.0013
F11	(20,10)	Mean	0.0734 †	0.1444 †	0.4341 †	0.2560 †	0.5887 †	0.3369 †	0.4823 †	0.6009
		Std	0.0033	0.0229	0.0017	0.0053	0.0023	0.0116	0.0127	9.01E-04
	(15,10)	Mean	0.0639 †	0.0943 †	0.3046 †	0.1155 †	0.5166 †	0.3217 †	0.4092 †	0.5463
		Std	0.0132	0.0531	0.0130	0.0319	0.0122	0.0243	0.0099	0.0054
	(10,10)	Mean	0.0900 †	0.0413 †	0.0896 †	0.0445 †	0.2651 †	0.2673 †	0.1993 †	0.4142
		Std	9.56E-04	0.0076	0.0136	0.0266	0.0365	0.0133	0.0085	4.68E-04

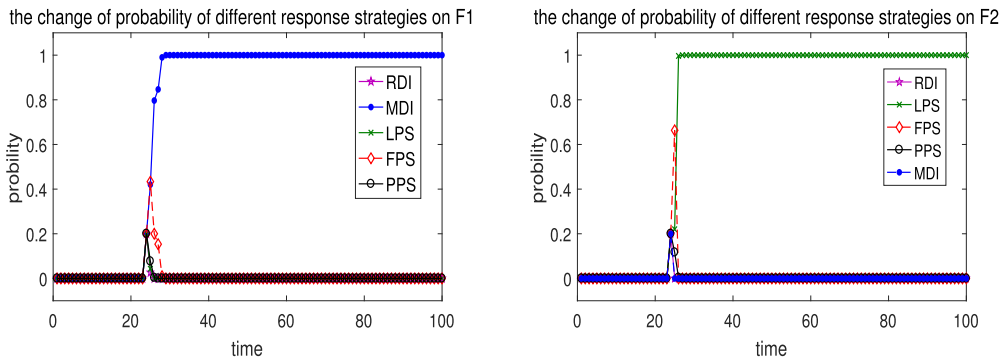


Figure 3: The change of probability of selecting different response strategies.

compared algorithms are presented in Section S5.3 of the Supplementary material, verifying that MOEA-OSD/SRS is the fastest algorithm among the eight algorithms.

Furthermore, it should be noted that the baseline multiobjective optimization algorithm is also an important component of a DMOA, which heavily affects the performance of DMOAs. Thus, we also make a comparison of MOEA-OSD with other six well-known multiobjective optimization algorithms, indicating that MOEA-OSD is the best-performing and computationally most efficient algorithm. The details are shown in Section S6 of the Supplementary material.

Last but not least, we also conduct a sensitivity analysis to investigate the influence of some parameters on the performance of MOEA-OSD/SRS, including the severity of the environmental changes, that is, n_T , the proportion of diversity introduction in RDI and MDI, differential crossover probability (CR), differential scale factor (F), and Gaussian mutation probability. The details are in Section S7 of the Supplementary material.

5 PID Controller Parameter Turning

To verify the performance of the proposed MOEA-OSD/SRS on real applications, we use it to tune the coefficients of the PID controller of a dynamic system.

The PID controller is very widely used to obtain the control according to the deviation between the desired input value and the actual output value of the system. In fact, the parameters of the system may change due to the aging of the equipment or the

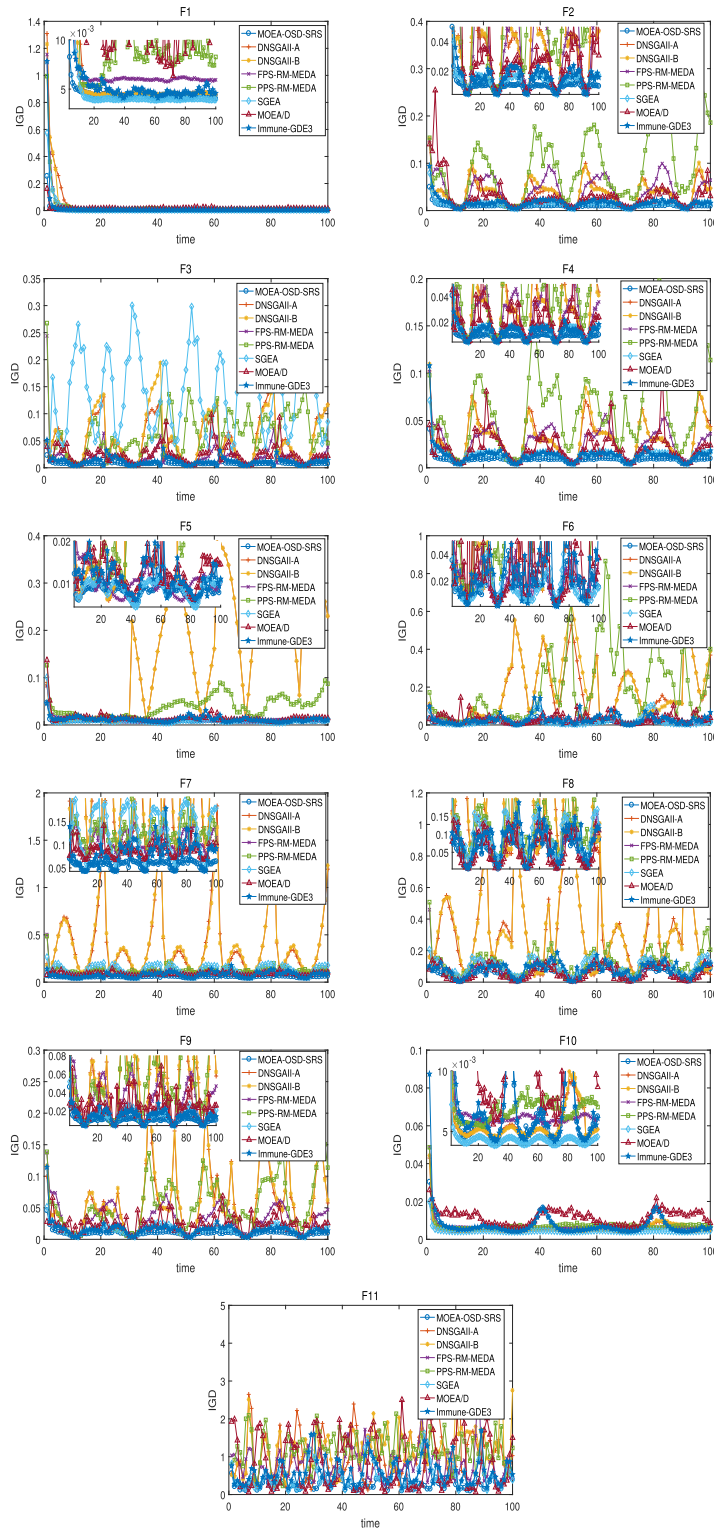


Figure 4: The average IGD over 20 runs versus the time.

Table 8: The statistical results of \overline{IGD} , \overline{S} , and \overline{HV} for different algorithms.

Metrics	Statistic	DNSGA	DNSGA	FPS	PPS	SGEA	MOEA/D	Immune	MOEA
		-II-A	-II-B	-RM -MEDA	-RM -MEDA			GDE3	-OSD /SRS
\overline{IGD}	Mean	4.9289+	4.8479+	3.7097+	3.6284+	3.9999+	4.0811+	4.6840+	3.4995
	Std	0.0579	0.1245	0.1196	0.0373	0.0815	0.1001	0.0968	0.0128
\overline{S}	Mean	1.8673+	1.6659+	0.1303+	3.6284+	0.0417§	0.0639+	0.0491+	0.0451
	Std	0.3479	0.5166	0.1156	0.0373	0.0019	0.0108	0.0030	0.0028
\overline{HV}	Mean	2.3793+	2.3823+	2.4038+	2.4050+	2.4005+	2.3891+	2.4089+	2.6319
	Std	0.0093	0.0164	0.0033	0.0011	0.0030	0.0032	0.0125	1.15E-04

interference of the environmental noise (Farina et al., 2004). So in order to obtain the satisfactory control effect, the parameters of PID controller should be adjusted adaptively with the change of the system parameters.

Firstly, the transfer function of the dynamic system is given as follows (Huang et al., 2011):

$$G(s) = \frac{400}{a_1(t)s^2 + a_2(t)s}, \tag{9}$$

where $a_1(t)$ and $a_2(t)$ are time-varying, which simulates the aging of the equipment or the interference of the environmental noise.

$$\begin{aligned} a_1(t) &= 3 + 30 \sin\left(\frac{\pi t}{50}\right) \\ a_2(t) &= 43 + 30 \sin\left(\frac{\pi t}{50}\right). \end{aligned} \tag{10}$$

The rising time t_u , maximum overshooting $ey(t)$, and control deviation $e(t)$ of the system are usually adopted to evaluate the performance of the PID controller. The dynamic multiobjective optimization problem solved in this section is formulated as:

$$\begin{cases} \min J_1 = \int_0^\infty (|e(t)| + |ey(t)|)dt \\ \min J_2 = t_u. \end{cases} \tag{11}$$

We assume that the range of the controller parameters are known; that is, $K_p \in [0, 20]$, K_i , and $K_d \in [0, 1]$.

To verify the performance of MOEA-OSD/SRS for tuning the PID parameters of the dynamic system, we compare MOEA-OSD/SRS with seven DMOAs as described in Section 4.4.1.

Table 8 shows the statistical results of \overline{IGD} , \overline{S} and \overline{HV} obtained by MOEA-OSD/SRS and other seven compared algorithms. Note, the PF corresponding to the nondominant solutions in the PS obtained by all eight algorithms makes up the real PF of each environment. It is obvious that MOEA-OSD/SRS is of better performance, that is to say, the control effect of the PID controller optimized by MOEA-OSD/SRS is much better. Therefore, MOEA-OSD/SRS is promising to be applied to tune the parameters of the PID controller of the dynamic system.

6 Conclusion

In this article, we propose a self-adaptive response strategy together with a decomposition-based evolutionary algorithm, MOEA-OSD/SRS for short, for optimization of DMOPs. MOEA-OSD aims to find the Pareto optimal solution for each

subobjective space, making it easier to maintain the diversity of the obtained solution set. MOEA-OSD adopts the maxi-min fitness function to compute the fitness value of each individual in each subobjective space, which is able to take into account both diversity and convergence properties of the solutions.

The proposed self-adaptive response strategy (SRS) integrates five popular response strategies developed for dynamic optimization and determines the probability of selecting each of the response strategies based on their previous performance. As a result, SRS is able to adaptively select the most effective response strategy for different dynamic problems, thereby being able to more effectively cope with unknown environmental changes than a single prespecified response strategy. It is worth noting that SRS can be embedded in any high-performance multiobjective optimization algorithm to solve DMOPs. Extensive empirical studies are conducted to demonstrate the effectiveness of the proposal self-adaptive response strategy. We first compare SRS with six state-of-the-art response strategies taken from the literature. Our results show that SRS is able to adaptively select the best response strategy for an unknown environment. Then, our comparisons of MOEA-OSD/SRS with seven state-of-the-art dynamic evolutionary multiobjective algorithms demonstrate its competitiveness for solving DMOPs. Finally, MOEA-OSD, as the base optimizer, is compared with six popular MOEAs to verify that MOEA-OSD is well suited for dynamic multiobjective optimization. In addition, a sensitivity analysis of the parameters in MOEA-OSD/SRS has been performed. Finally, an application of the proposed algorithm to the PID controller parameter tuning problem demonstrates the competitiveness of the proposed algorithm.

Although MOEA-OSD/SRS has provided encouraging performance on the DMOPs considered in this article, it still has much room for improvement. Our future work will be dedicated to design more efficient response strategies to integrate into SRS to solve challenging DMOPs. It is also of interest to extend the proposed MOEA-OSD/SRS to solve constrained DMOPs.

Acknowledgments

This work was supported by the Provincial Natural Science Foundation of Shaanxi of China (No. 2019JZ-26) and the National Natural Science Foundation of China (Nos. 61876141 and 61373111).

References

- Avdagić, Z., Konjicija, S., and Omanović, S. (2009). Evolutionary approach to solving non-stationary dynamic multi-objective problems. In *Foundations of Computational Intelligence*, Vol. 3, pp. 267–289.
- Balling, R. (2003). The maximin fitness function; multi-objective city and regional planning. In *International Conference on Evolutionary Multi-Criterion Optimization*, pp. 1–15.
- Bui, L. T., Abbass, H. A., and Branke, J. (2005). Multiobjective optimization for dynamic environments. In *IEEE Congress on Evolutionary Computation*, Vol. 3, pp. 2349–2356.
- Carlisle, A., and Dozier, G. (2000). Adapting particle swarm optimization to dynamic environments. In *International conference on artificial intelligence*, Vol. 1, pp. 429–434.
- Chang, P. C., Chen, S. H., Zhang, Q., and Lin, J. L. (2008). MOEA/D for flowshop scheduling problems. In *IEEE Congress on Evolutionary Computation*, pp. 1433–1438.

- Cheng, R., Jin, Y., Olhofer, M., and Sendhoff, B. (2016). A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 20(5):773–791.
- Cobb, H. G. (1990). *An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-dependent nonstationary environments*. Technical Report. Naval Research Lab, Washington, DC.
- Coello Coello, C. A. (2002). MOPSO: A proposal for multiple objective particle swarm optimization. In *IEEE Congress on Evolutionary Computation*, Vol. 2, pp. 1051–1056.
- Cruz, C., González, J. R., and Pelta, D. A. (2011). Optimization in dynamic environments: A survey on problems, methods and measures. *Soft Computing*, 15(7):1427–1448.
- Deb, K., Karthik, S., et al. (2007). Dynamic multi-objective optimization and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling. In *International Conference on Evolutionary Multi-Criterion Optimization*, pp. 803–817.
- Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.
- Derrac, J., Salvador Garca, D. M., and Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1):3–18.
- Farina, M., Deb, K., and Amato, P. (2004). Dynamic multiobjective optimization problems: Test cases, approximations, and applications. *IEEE Transactions on Evolutionary Computation*, 8(5):425–442.
- Fonseca, C. M., and Fleming, P. J. (1993). Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *International Conference on Genetic Algorithms*, pp. 416–423.
- Goh, C.-K., and Tan, K. C. (2009a). A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 13(1):103–127.
- Goh, C.-K., and Tan, K. C. (2009b). Evolutionary multi-objective optimization in uncertain environments. *Studies in Computational Intelligence*, 186:5–18.
- Greeff, M., and Engelbrecht, A. P. (2008). Solving dynamic multi-objective problems with vector evaluated particle swarm optimisation. In *IEEE Congress on Evolutionary Computation*, pp. 2922–2929.
- Greeff, M., and Engelbrecht, A. P. (2010). Dynamic multi-objective optimisation using PSO. *Studies in Computational Intelligence*, 261:105–123.
- Grefenstette, J. J. et al. (1992). Genetic algorithms for changing environments. In *Parallel Problem Solving from Nature*, Vol. 2, pp. 137–144.
- Hatzakis, I., and Wallace, D. (2006). 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.
- Helbig, M., and Engelbrecht, A. P. (2011). Archive management for dynamic multi-objective optimisation problems using vector evaluated particle swarm optimisation. In *IEEE Congress on Evolutionary Computation*, pp. 2047–2054.
- Helbig, M., and Engelbrecht, A. P. (2012). Analyses of guide update approaches for vector evaluated particle swarm optimisation on dynamic multi-objective optimisation problems. In *IEEE Congress on Evolutionary Computation*, pp. 1–8.

- Helbig, M., and Engelbrecht, A. P. (2013a). Analysing the performance of dynamic multi-objective optimisation algorithms. In *IEEE Congress on Evolutionary Computation*, pp. 1531–1539.
- Helbig, M., and Engelbrecht, A. P. (2013b). Dynamic multi-objective optimization using PSO. In *Metaheuristics for Dynamic Optimization*, pp. 147–188.
- Helbig, M., and Engelbrecht, A. P. (2014). Population-based metaheuristics for continuous boundary-constrained dynamic multi-objective optimisation problems. *Swarm and Evolutionary Computation*, 14:31–47.
- Higashi, N., and Iba, H. (2003). Particle swarm optimization with Gaussian mutation. In *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE*, pp. 72–79. IEEE.
- Huang, L., Suh, I. H., and Abraham, A. (2011). Dynamic multi-objective optimization based on membrane computing for control of time-varying unstable plants. *Information Ences*, 181(11):2370–2391.
- Ishibuchi, H., Sakane, Y., Tsukamoto, N., and Nojima, Y. (2009). Evolutionary many-objective optimization by NSGA-II and MOEA/D with large populations. In *Systems, Man and Cybernetics*, pp. 1758–1763.
- Jiang, S., and Yang, S. (2017a). Evolutionary dynamic multiobjective optimization: Benchmarks and algorithm comparisons. *IEEE Transactions on Cybernetics*, 47(1):198–211.
- Jiang, S., and Yang, S. (2017b). A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 21(1):65–82.
- Jin, Y., and Branke, J. (2005). Evolutionary optimization in uncertain environments—A survey. *IEEE Transactions on Evolutionary Computation*, 9(3):303–317.
- Jin, Y., and Sendhoff, B. (2002). Fitness approximation in evolutionary computation—A survey. In *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*, pp. 1105–1112.
- Jin, Y., Wang, H., Chugh, T., Guo, D., and Miettinen, K. (2018). Data-driven evolutionary optimization: An overview and case studies. *IEEE Transactions on Evolutionary Computation*, 23(3):442–458.
- Li, H., and Zhang, Q. (2009). Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*, 13(2):284–302.
- Liu, C.-a. (2010). New dynamic multiobjective evolutionary algorithm with core estimation of distribution. In *International Conference on Electrical and Control Engineering*, pp. 1345–1348.
- Liu, M., Zheng, J., Wang, J., Liu, Y., and Jiang, L. (2014). An adaptive diversity introduction method for dynamic evolutionary multiobjective optimization. In *IEEE Congress on Evolutionary Computation*, pp. 3160–3167.
- Liu, R., Chen, Y., Ma, W., Mu, C., and Jiao, L. (2014). A novel cooperative coevolutionary dynamic multi-objective optimization algorithm using a new predictive model. *Soft Computing*, 18(10):1913–1929.
- Liu, R., Li, J., Mu, C., Jiao, L., et al. (2017). A coevolutionary technique based on multi-swarm particle swarm optimization for dynamic multi-objective optimization. *European Journal of Operational Research*, 261(3):1028–1051.
- Liu, Y., and Niu, B. (2013). A multi-objective particle swarm optimization based on decomposition. In *International Conference on Intelligent Computing*, pp. 200–205.
- Luce, R. D., and Raiffa, H. (2012). *Games and decisions: Introduction and critical survey*. North Chelmsford, MA: Courier Corporation.

- Martínez-Peñaloza, M., and Mezura-Montes, E. (2018). Immune generalized differential evolution for dynamic multi-objective environments: An empirical study. *Knowledge Based Systems*, 142:192–219.
- Michalewicz, Z., Schmidt, M., Michalewicz, M., and Chiriach, C. (2007). Adaptive business intelligence: Three case studies. In *Evolutionary Computation in Dynamic and Uncertain Environments*, pp. 179–196.
- Morrison, R. W. (2002). *Designing evolutionary algorithms for dynamic environments*. Fairfax, VA: George Mason University.
- Ng, K. P., and Wong, K. C. (1995). A new diploid scheme and dominance change mechanism for non-stationary function optimization. In *International Conference on Genetic Algorithms*, pp. 159–166.
- Nguyen, T. T., Yang, S., and Branke, J. (2012). Evolutionary dynamic optimization: A survey of the state of the art. *Swarm and Evolutionary Computation*, 6:1–24.
- Pelosi, G., and Selleri, S. (2014). To Celigny, in the footprints of Vilfredo Pareto’s “optimum” [Historical Corner]. *IEEE Antennas and Propagation Magazine*, 56(3):249–254.
- Price, K., Storn, R. M., and Lampinen, J. A. (2006). *Differential evolution: A practical approach to global optimization*. Berlin: Springer Science & Business Media.
- Raquel, C., and Yao, X. (2013). Dynamic multi-objective optimization: A survey of the state-of-the-art. In *Evolutionary Computation for Dynamic Optimization Problems*, pp. 85–106.
- Rawls, J. (2009). *A theory of justice*. Cambridge, MA: Harvard University Press.
- Rong, M., Gong, D., Zhang, Y., Jin, Y., and Pedrycz, W. (2018). Multidirectional prediction approach for dynamic multiobjective optimization problems. *IEEE Transactions on Cybernetics*, 49(9):3362–3374.
- Salazar Lechuga, M. (2009). Multi-objective optimisation using sharing in swarm optimisation algorithms. PhD thesis, University of Birmingham.
- Schott, J. R. (1995). Fault tolerant design using single and multicriteria genetic algorithm optimization. *Cellular Immunology*, 37(1): 113.
- Shang, R., Jiao, L., Gong, M., and Lu, B. (2005). Clonal selection algorithm for dynamic multi-objective optimization. In *International Conference on Computational and Information Science*, pp. 846–851.
- Shang, R., Jiao, L., Ren, Y., Li, L., and Wang, L. (2014). Quantum immune clonal coevolutionary algorithm for dynamic multiobjective optimization. *Soft Computing*, 18(4):743–756.
- Tezuka, M., Munetomo, M., and Akama, K. (2007). Genetic algorithm to optimize fitness function with sampling error and its application to financial optimization problem. In *Evolutionary Computation in Dynamic and Uncertain Environments*, pp. 417–434.
- Tinós, R., and Yang, S. (2007). Genetic algorithms with self-organizing behaviour in dynamic environments. In *Evolutionary Computation in Dynamic and Uncertain Environments*, pp. 105–127.
- Vavak, F., Juke, K., and Fogarty, T. C. (1997). Adaptive combustion balancing in multiple burner boiler using a genetic algorithm with variable range of local search. In *International Conference on Genetic Algorithms*, pp. 719–726.
- Wu, Y., Jin, Y., and Liu, X. (2015). A directed search strategy for evolutionary dynamic multiobjective optimization. *Soft Computing*, 19(11):3221–3235.
- Yang, S. (2006). Associative memory scheme for genetic algorithms in dynamic environments. In *Workshops on Applications of Evolutionary Computation*, pp. 788–799.

- Yuen, T. J., and Ramli, R. (2010). Comparison of computational efficiency of MOEA/D and NSGA-II for passive vehicle suspension optimization. *ECMS*, 2010:219–225.
- Zeng, S.-Y., Chen, G., Zheng, L., Shi, H., de Garis, H., Ding, L., and Kang, L. (2006). A dynamic multi-objective evolutionary algorithm based on an orthogonal design. In *IEEE Congress on Evolutionary Computation*, pp. 573–580.
- Zhang, Q., and Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731.
- Zhang, Q., Zhou, A., and Jin, Y. (2008). RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm. *IEEE Transactions on Evolutionary Computation*, 12(1):41–63.
- Zhang, Z., and Qian, S. (2011). Artificial immune system in dynamic environments solving time-varying non-linear constrained multi-objective problems. *Soft Computing*, 15(7):1333–1349.
- Zhou, A., Jin, Y., and Zhang, Q. (2014). A population prediction strategy for evolutionary dynamic multiobjective optimization. *IEEE Transactions on Cybernetics*, 44(1):40–53.
- Zhou, A., Jin, Y., Zhang, Q., Sendhoff, B., and Tsang, E. (2007). Prediction-based population reinitialization for evolutionary dynamic multi-objective optimization. In *International Conference on Evolutionary Multi-Criterion Optimization*, pp. 832–846.
- Zitzler, E., Laumanns, M., and Thiele, L. (2002). SPEA2: Improving the strength Pareto evolutionary algorithm. In *Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, pp. 95–100.
- Zitzler, E., and Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271.