

# A Benchmark Generator for Dynamic Multi-objective Optimization Problems

Shouyong Jiang

Centre for Computational Intelligence (CCI)  
School of Computer Science and Informatics,  
De Montfort University  
Leicester LE1 9BH, UK  
Email: shouyong.jiang@email.dmu.ac.uk

Shengxiang Yang

Centre for Computational Intelligence (CCI)  
School of Computer Science and Informatics,  
De Montfort University  
Leicester LE1 9BH, UK  
Email: syang@dmu.ac.uk

**Abstract**—Many real-world optimization problems appear to not only have multiple objectives that conflict each other but also change over time. They are dynamic multi-objective optimization problems (DMOPs) and the corresponding field is called dynamic multi-objective optimization (DMO), which has gained growing attention in recent years. However, one main issue in the field of DMO is that there is no standard test suite to determine whether an algorithm is capable of solving them. This paper presents a new benchmark generator for DMOPs that can generate several complicated characteristics, including mixed Pareto-optimal front (convexity-concavity), strong dependencies between variables, and a mixed type of change, which are rarely tested in the literature. Experiments are conducted to compare the performance of five state-of-the-art DMO algorithms on several typical test functions derived from the proposed generator, which gives a better understanding of the strengths and weaknesses of these tested algorithms for DMOPs.

## I. INTRODUCTION

Many real-world multi-objective optimization problems (MOPs) appear to change over time in a dynamic environment, such as planning [20], scheduling [3], and control [22]. Dynamic multi-objective optimization problems (DMOPs) have gained increasing attention over the past several years. Due to the dynamics of these problems, the optimization of DMOPs is much more challenging than that of static MOPs as it has to deal with not only the conflicting objectives, but also the changes in objective functions or constraints. In other words, dynamic multi-objective optimization algorithms (DMOAs) must be capable of tracking the changing Pareto-optimal front (POF) to provide a diverse set of solutions that approximates the new POF over time.

Artificial benchmark problems have played a fundamental role in determining whether a DMOA has the ability to solve DMOPs. Furthermore, benchmark problems contribute to analysing and identifying the strength and weakness of a DMOA in order to modify it and improve its performance. However, one of the main problems in the field of dynamic multi-objective optimization (DMO) is that there are no standard test functions. Therefore, developing a new set of test functions to compare the performance of DMOAs becomes meaningful and essential.

In this paper, we first briefly review the DMOP test problems that are used in the literature and discuss their

limitations. Based on the understanding of the limitations of current DMOP test problems, we present a new benchmark generator for DMOPs that can generate DMOPs with several complex characteristics which are rarely tested in the literature, including POF of mixed types of convexity and concavity, strong dependencies between variables, and a mixed type of change. From this generator, a number of DMOP test instances are derived and experiments are conducted to compare the performance of five state-of-the-art DMO algorithms on them. The results give us a better understanding of the strengths and weaknesses of these tested algorithms for DMOPs.

## II. RELATED WORK

There are many dynamic characteristics involved in DMOPs, and different problems may have different mathematical definitions. This paper focuses on the DMOPs defined as follows:

$$\begin{aligned} \min \quad & F(x, t) = (f_1(x, t), \dots, f_M(x, t))^T \\ \text{s.t.} \quad & \begin{cases} h_i(x, t) = 0, & i = 1, \dots, n_h \\ g_i(x, t) \geq 0, & i = 1, \dots, n_g \\ x \in \Omega_x, \quad t \in \Omega_t \end{cases} \end{aligned} \quad (1)$$

where  $\Omega_x \subseteq R^n$  is the decision space and  $t$  is the discrete time instance defined as  $t = \frac{1}{n_t} \lfloor \frac{\tau}{\tau_t} \rfloor$ , where  $n_t$  represents the severity of change,  $\tau$  is the iteration counter and  $\tau_t$  represents the frequency of change and  $\Omega_t \subseteq R$  is the time space.  $F(x, t): \Omega_x \times \Omega_t \rightarrow R^M$  is the objective function vector that evaluates the solution  $x$  at time  $t$ .

Most artificial DMOPs developed by researchers in the literature conform to Eq. (1). A distinct characteristic of Eq. (1) is that the POF and Pareto-optimal set (POS) are susceptible to change, which challenges the tracking ability of DMOAs. Farina *et al.* [5] classified DMOPs into four types according to the dynamics of POF and POS.

- Type I: the POS changes over time while the POF remains stationary;
- Type II: both the POF and POS change over time;
- Type III: the POF changes over time while the POS remains stationary;
- Type IV: both the POF and POS remain stationary, although the objective function or constraints may change over time.

Farina *et al.* also created some FDA DMOPs by adapting the static problems from the ZDT [28] and DTLZ [4] test suites. One distinct Type-III problem in the FDA test suite is FDA2, which is defined as follows:

$$FDA2 : \begin{cases} f_1(\mathbf{x}, t) = x_1 \\ f_2(\mathbf{x}, t) = g(\mathbf{x}_{II})h(\mathbf{x}_{III}, f_1(\mathbf{x}), g(\mathbf{x}_{II}), t) \\ g(\mathbf{x}_{II}, t) = 1 + \sum_{x_i \in \mathbf{x}_{II}} x_i^2 \\ h(\mathbf{x}_{III}, f_1, g, t) = 1 - \left(\frac{f_1}{g}\right)^{H_2(t)} \\ H(t) = 0.75 + 0.7 \sin(0.5\pi t), t = \frac{1}{n_t} \lfloor \frac{\tau}{n_t} \rfloor \\ H_2(t) = (H(t) + \sum_{x_i \in \mathbf{x}_{III}} (x_i - H(t))^2)^{-1} \\ \mathbf{x}_I = (x_1) \in [0, 1], \mathbf{x}_{II}, \mathbf{x}_{III} \in [-1, 1]^{n-1} \end{cases} \quad (2)$$

where the POF is  $f_2 = 1 - \sqrt[n_t]{f_1}$ , and the POS is  $x_i = 0, \forall x_i \in \mathbf{x}_{II}$  and  $x_i = -1, \forall x_i \in \mathbf{x}_{III}$ .

FDA2 has been widely used and further modified by researchers to test algorithms' performance. It should be noted that there is a misunderstanding of this test problem. Helbig and Engelbrecht [9], [10] studied the performance of several algorithms on FDA2, finding that all the tested algorithms lose track of the changing POF. However, the phenomenon of losing track of the changing POF they observed does not exist in FDA2 because the approximated POF would never be better than the true Pareto-optimal front. Actually, their misleading results may come from the assumption that the POS of FDA2 is  $x_i = 0, \forall x_i \in \mathbf{x}_{II}$  and  $x_i = H(t), \forall x_i \in \mathbf{x}_{III}$ , which gives a misleading POF as is illustrated in [9], [10].

Jin and Sendhoff [14] suggested a method for constructing dynamic multi-objective test problems by dynamically changing the weights that aggregate the different objectives of static MOPs. But, their method did not provide clear defined problems. Guan *et al.* [8] studied DMOPs with objective replacement, where some objectives may be replaced with new objectives during the evolution. Mehnen *et al.* [18] argued that the DTLZ and ZDT test suites are already challenging in their static version, and simpler test functions are needed to analyse the effect of dynamics in DMOPs. Hence, they suggested the DSW functions for DMOPs. Furthermore, they proposed a new generic scheme DTF that is a generalized FDA function and allows a variable scaling of the complexity of the dynamic properties. They also added scalable and dynamic constraints to DMOPs by moving circular obstacles in the objective space.

The ZJZ problem defined by Zhou *et al.* [24] is the first DMOP with variable linkages, which is a modified version of FDA1 but more challenging. In [23], the authors further argued that most DMOPs derived from the FDA test suite are too simple, and the correlation between decision variables should be enhanced. Thus, they gave four new DMOP test instances that have nonlinear correlation between the decision variables. Helbig and Engelbrecht [11] made a sound investigation into the current DMOPs used in the literature, and identified their shortcomings that none of them had deceptive and isolated features in POF. Then, they developed DMOPs with either an isolated or deceptive POF that follows the concept of the static WFG [13] test suite. In addition, they proposed some HE problems that have complicated POSs based on the MOPs of Li and Zhang [17].

Most existing DMOPs that have been discussed in the literature are included in the first three types of change

mentioned earlier, but none of them is a Type-IV problem. Recently, Huang *et al.* [12] created several Type-IV problems by implying that the current found POS may affect the further POS or POF. Furthermore, they introduced two DMOPs where the number of decision variables or objective functions changes over time.

The current commonly used DMOPs are the variants of the FDA test problems. None of them have the following characteristics:

- 1) Mixed POFs (convexity and concavity) that changes over time
- 2) A complicated diversity-resistant scheme that hinders a set of diverse solutions
- 3) A problem that can change between different types during the evolution
- 4) A strong correlation between variables instead of the simple dependency between  $x_i$  and  $x_1$ .

### III. PROPOSED BENCHMARK GENERATOR AND TEST INSTANCES

#### A. The Proposed Benchmark Generator

Considering the following DMOP:

$$JY : \begin{cases} f_1(\mathbf{x}, t) = H(\mathbf{x}_{II}, t)(h(\mathbf{x}_I) + A(t)\sin(W(t)\pi h(\mathbf{x}_I)))^{\gamma_1} \\ f_2(\mathbf{x}, t) = H(\mathbf{x}_{II}, t)(1 - h(\mathbf{x}_I) + A(t)\sin(W(t)\pi h(\mathbf{x}_I)))^{\gamma_2} \\ H(\mathbf{x}_{II}, t) = 1 + g(\mathbf{x}_{II}, t) \end{cases} \quad (3)$$

where  $0 \leq h(\mathbf{x}_I) \leq 1$ , and  $\mathbf{x}_I$  and  $\mathbf{x}_{II}$  are sub-vectors of the decision vector  $\mathbf{x}$ .  $A(t)$  and  $W(t)$  are two parameters to control the local shape of the Pareto front, with  $A(t)$  adjusting the curvature and  $W(t)$  controlling the number of mixed convex and concave segments on the Pareto front. A large value of  $W(t)$  causes the Pareto front to have disconnected regions, while a small value produces a continuous Pareto front. Here,  $W(t)$  is recommended to be an integer.  $\gamma_1$  and  $\gamma_2$  ( $\gamma_1 > 0, \gamma_2 > 0$ ) are parameters that control the overall shape of the Pareto front: when  $\gamma_1 > 1$  and  $\gamma_2 > 1$  or  $\gamma_1 < 1$  and  $\gamma_2 < 1$ , the overall shape is convex or concave respectively; when  $\gamma_1 = \gamma_2 = 1$ , the overall shape is linear; otherwise, the overall shape is mixed.  $g(\mathbf{x}_{II}, t)$  is a non-negative function, hindering algorithms from converging towards the true Pareto front. The minimum of  $g(\mathbf{x}_{II}, t)$  is zero. Thus, Eq. (3) can produce various Pareto-front geometries by properly configuring relevant parameters. Generally, the mathematical description of the continuous Pareto front for Eq. (3) is as follows:

$$f_1^{\frac{1}{\gamma_1}} + f_2^{\frac{1}{\gamma_2}} = 1 + 2A(t)\sin\left(W(t)\pi \frac{f_1^{\frac{1}{\gamma_1}} - f_2^{\frac{1}{\gamma_2}} + 1}{2}\right) \quad (4)$$

where the values of  $A(t)$  and  $W(t)$  must enable Eq. (3) to be a continuous Pareto front. To have a better understanding of the proposed generator, we denote  $F_1 = f_1^{\frac{1}{\gamma_1}} - f_2^{\frac{1}{\gamma_2}}$  and  $F_2 = f_1^{\frac{1}{\gamma_1}} + f_2^{\frac{1}{\gamma_2}}$ . This means that a clockwise rotation with an angle  $\pi/4$  is made from the current coordinate axis. Then, Eq. (4) can be rewritten as:

$$F_2 = 1 + 2A(t)\sin\left(W(t)\pi \frac{F_1 + 1}{2}\right) \quad (5)$$

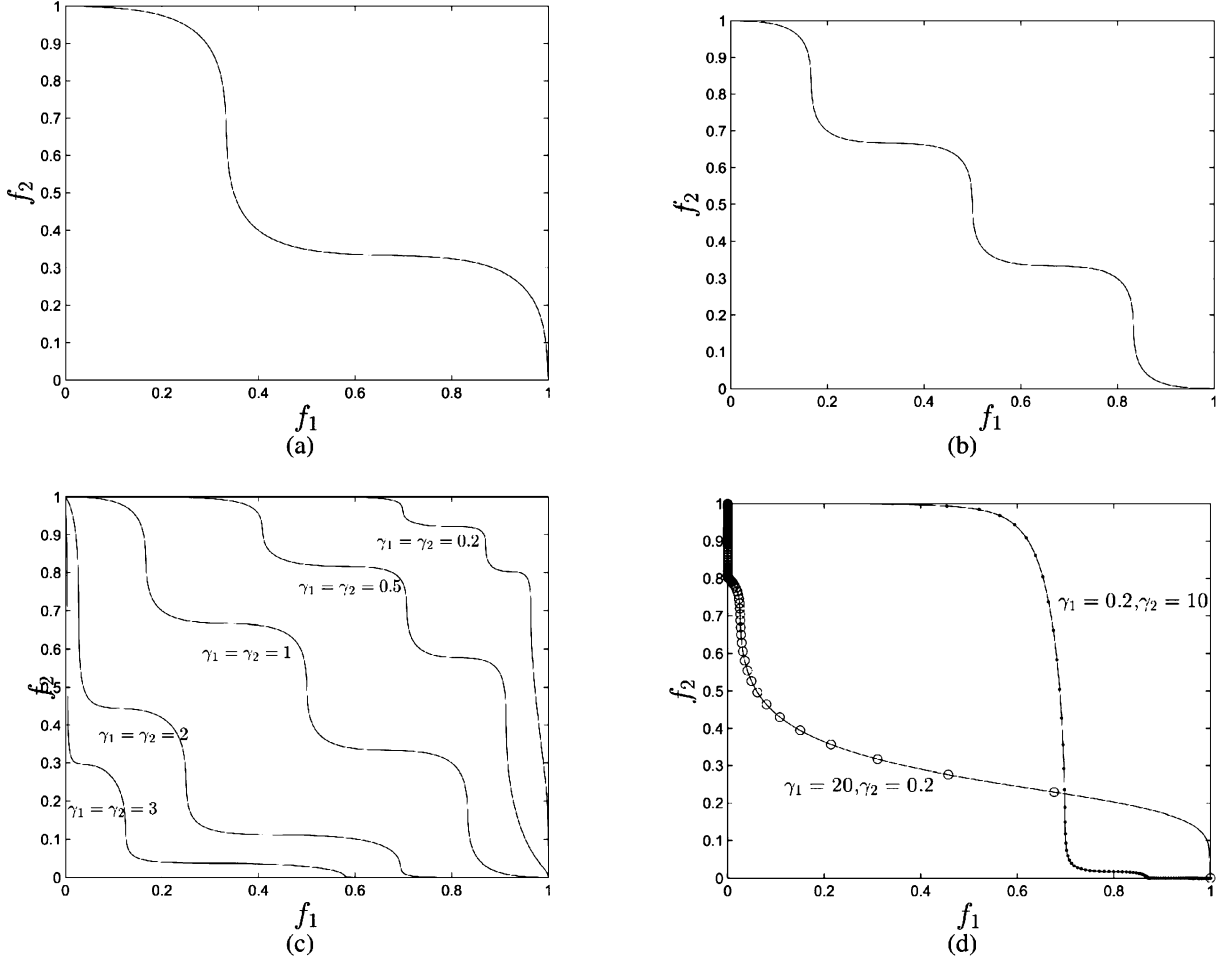


Fig. 1. POFs of  $JY$  with different overall shapes: (a)  $\gamma_1 = \gamma_2 = 1$ ,  $A(t) = 0.1$ ,  $W(t) = 3$ ; (b)  $\gamma_1 = \gamma_2 = 1$ ,  $A(t) = 0.05$ ,  $W(t) = 6$ ; (c) convex or concave overall shapes with  $A(t) = 0.05$ ,  $W(t) = 6$ ; (d) mixed overall shapes with  $A(t) = 0.05$ ,  $W(t) = 6$ .

where a sin wave is described if  $W(t) \neq 0$  and  $A(t) \neq 0$ . Thus, the proposed generator has a wave-like geometry, containing both concave and convex regions. Figure 1 illustrates examples of POFs of  $JY$  with linear and non-linear overall shapes.

### B. Test Instances

When the overall shape of the POF of  $JY$  is non-linear, DMOAs may have difficulties in achieving desirable results, such as a set of diverse solutions. For an easier understanding and analysis of the proposed generator, the POF of  $JY$  is set to have a linear overall shape, namely  $\gamma_1 = \gamma_2 = 1$ . Apart from this, we concentrate on  $h(\mathbf{x}_I) = x_1$ , although we recognize that movement across the POF can be achieved by adjusting a number of variables, i.e., the use of rotation matrices for  $h(\mathbf{x}_I)$  and the normalization of  $h(\mathbf{x}_I)$ . Below, we provide some benchmark instances with detailed information for the first three types of change in the environment. So, a problem with mixed types is generated.

$JY1$ , as shown in Eq. (6), is a Type-I problem, where the POS changes over time in a regular pattern, with  $x_i = G(t)$ ,  $\forall x_i \in \mathbf{x}_{II}$ . It mainly tests the convergence speed and reactivity of an algorithm.

$$JY1: \begin{cases} f_1(\mathbf{x}, t) = (1 + g(\mathbf{x}_{II}, t))(x_1 + A(t)\sin(W(t)\pi x_1)) \\ f_2(\mathbf{x}, t) = (1 + g(\mathbf{x}_{II}, t))(1 - x_1 + A(t)\sin(W(t)\pi x_1)) \\ g(\mathbf{x}_{II}, t) = \sum_{x_i \in \mathbf{x}_{II}} (x_i - G(t))^2 \\ G(t) = \sin(0.5\pi t), \quad t = \frac{1}{n_t} \lfloor \frac{\tau}{\tau_t} \rfloor \\ A(t) = 0.05, \quad W(t) = 6 \\ \mathbf{x}_I = (x_1) \in [0, 1], \quad \mathbf{x}_{II} = (x_2, \dots, x_n) \in [-1, 1]^{n-1} \end{cases} \quad (6)$$

$JY2$ , as shown in Eq. (7), is a Type-II problem with dynamic POFs and POSSs. The change of POSSs is simple, but the change of the POF, as illustrated in Fig. 2, is quite complicated because the convexity-concavity of any regions along the POF may swap after a change (refer to Eq. (4)).

$$JY2: \begin{cases} f_1(\mathbf{x}, t) = (1 + g(\mathbf{x}_{II}, t))(x_1 + A(t)\sin(W(t)\pi x_1)) \\ f_2(\mathbf{x}, t) = (1 + g(\mathbf{x}_{II}, t))(1 - x_1 + A(t)\sin(W(t)\pi x_1)) \\ g(\mathbf{x}_{II}, t) = \sum_{x_i \in \mathbf{x}_{II}} (x_i - G(t))^2 \\ G(t) = \sin(0.5\pi t), \quad t = \frac{1}{n_t} \lfloor \frac{\tau}{\tau_t} \rfloor \\ A(t) = 0.05, \quad W(t) = \lfloor 6\sin(0.5\pi(t-1)) \rfloor \\ \mathbf{x}_I = (x_1) \in [0, 1], \quad \mathbf{x}_{II} = (x_2, \dots, x_n) \in [-1, 1]^{n-1} \end{cases} \quad (7)$$

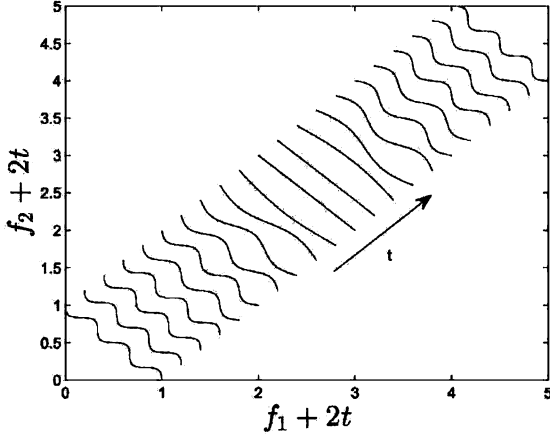


Fig. 2. POF of  $JY2$  with 21 time windows varying from 0 to 2. For a better visualization,  $f_1 + 2t$  and  $f_2 + 2t$  are shown on the  $x$  and  $y$  axes, respectively.

$JY3$ , as shown in Eq. (8), introduces different dependencies between any two decision variables, which is closer to real-world problems. Therefore, the POF of  $JY3$  is similar to that of  $JY2$ , and the POS is  $y_1 = |x_1 \sin((2\alpha + 0.5)\pi x_1)|$ ,  $\alpha = \lfloor 100 \sin^2(0.5\pi t) \rfloor$ ,  $y_i = \sqrt{y_{i-1}}$ ,  $i = 2, \dots, n$ . The POS for variables  $x_1$  and  $x_2$  is shown in Fig. 3, where the dependency between these two variables is increasingly complicated as the time increases. Furthermore, the increasing time leads to the change of the density of solutions. Therefore,  $JY3$  not only assesses the effect of variable-linkage but also tests the diversity performance of an algorithm in a dynamic environment.

$$JY3: \begin{cases} f_1(\mathbf{x}, t) = (1 + g(\mathbf{x}_{II}, t))(y_1 + A(t) \sin(W(t)\pi y_1)) \\ f_2(\mathbf{x}, t) = (1 + g(\mathbf{x}_{II}, t))(1 - y_1 + A(t) \sin(W(t)\pi y_1)) \\ g(\mathbf{x}_{II}, t) = \sum_{x_i \in \mathbf{x}_{II}} (y_i^2 - y_{i-1})^2 \\ A(t) = 0.05, \quad W(t) = \lfloor 6 \sin(0.5\pi(t-1)) \rfloor \\ \alpha = \lfloor 100 \sin^2(0.5\pi t) \rfloor, \quad t = \frac{1}{n_t} \lfloor \frac{t}{\tau_t} \rfloor \\ y_1 = |x_1 \sin((2\alpha + 0.5)\pi x_1)|, y_i = x_i, i = 2, \dots, n \\ \mathbf{x}_I = (x_1) \in [0, 1], \quad \mathbf{x}_{II} = (x_2, \dots, x_n) \in [-1, 1]^{n-1} \end{cases} \quad (8)$$

$JY4$ , as shown in Eq. (9), is constructed to have a number of disconnected POF segments. The number of disconnected POF segments and the spread of each segment change over time. As illustrated in Fig. 4, the POF of  $JY4$  is subjected to the definition of Eq. (4), but has a number of disconnected segments. The POS is  $x_i = G(t)$ ,  $\forall x_i \in \mathbf{x}_{II}$ .

$$JY4: \begin{cases} f_1(\mathbf{x}, t) = (1 + g(\mathbf{x}_{II}, t))(x_1 + A(t) \sin(W(t)\pi x_1)) \\ f_2(\mathbf{x}, t) = (1 + g(\mathbf{x}_{II}, t))(1 - x_1 + A(t) \sin(W(t)\pi x_1)) \\ g(\mathbf{x}_{II}, t) = \sum_{x_i \in \mathbf{x}_{II}} (x_i - G(t))^2 \\ G(t) = \sin(0.5\pi t), \quad t = \frac{1}{n_t} \lfloor \frac{t}{\tau_t} \rfloor \\ A(t) = 0.05, \quad W(t) = 10^{\lfloor 1 + |G(t)| \rfloor} \\ \mathbf{x}_I = (x_1) \in [0, 1], \quad \mathbf{x}_{II} = (x_2, \dots, x_n) \in [-1, 1]^{n-1} \end{cases} \quad (9)$$

Contrary to the above problems,  $JY5$  (as shown in Eq. (10)) does not have a mixed POF and is a Type-III problem.

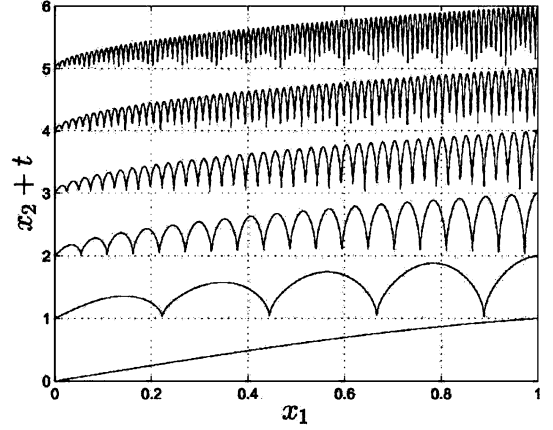


Fig. 3. An example of POS of  $JY3$  for the first two variables with 6 time windows varying from 0 to 0.5. For a better visualization,  $x_1$  and  $x_2 + t$  are shown on the  $x$  and  $y$  axes, respectively.

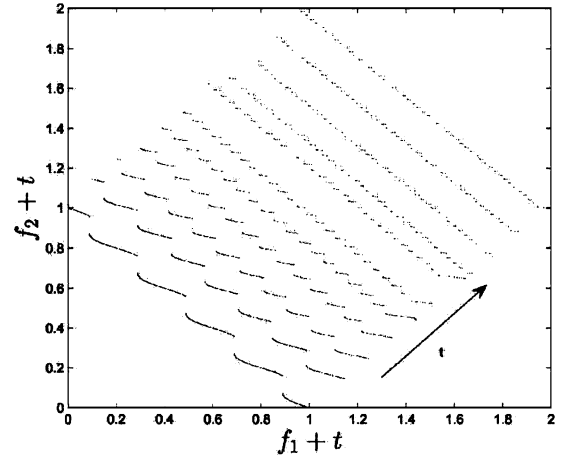


Fig. 4. POF of  $JY4$  with 11 time windows varying from 0 to 2. For a better visualization,  $f_1 + t$  and  $f_2 + t$  are shown on the  $x$  and  $y$  axes, respectively.

Its POF is very simple and changes from convex to concave. The POF is defined in Eq. (4) and illustrated in Fig. 5.

$$JY5: \begin{cases} f_1(\mathbf{x}, t) = (1 + g(\mathbf{x}_{II}, t))(x_1 + A(t) \sin(W(t)\pi x_1)) \\ f_2(\mathbf{x}, t) = (1 + g(\mathbf{x}_{II}, t))(1 - x_1 + A(t) \sin(W(t)\pi x_1)) \\ g(\mathbf{x}_{II}, t) = \sum_{x_i \in \mathbf{x}_{II}} x_i^2 \quad t = \frac{1}{n_t} \lfloor \frac{t}{\tau_t} \rfloor \\ A(t) = 0.3 \sin(0.5\pi(t-1)), \quad W(t) = 1 \\ \mathbf{x}_I = (x_1) \in [0, 1], \quad \mathbf{x}_{II} = (x_2, \dots, x_n) \in [-1, 1]^{n-1} \end{cases} \quad (10)$$

In practice, more complicated Type-I to Type-III benchmark problems can be generated by further varying parameters  $A(t)$ ,  $W(t)$ ,  $\gamma_1$ , or  $\gamma_2$  over time.

The first three types of change can be easily realized when constructing test functions, and they have been commonly reported in the literature. However, many real-world optimization problems with dynamic characteristics, i.e., the launch of contingency plan for dealing with bad weather in air traffic scheduling, may jump between types. To the best



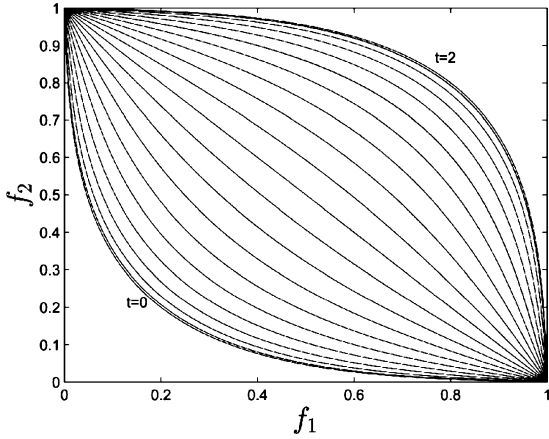


Fig. 5. POE of JY5 with 21 time windows varying from 0 to 2.

of our knowledge, none of this kind of test function has been introduced into the family of DMOPs. In this paper, we propose such a problem that cyclically switches from Type I to Type II, then to Type III. Technically, this kind of problem is macroscopically a Type-II problem from the perspective of the whole period of changes. Despite that, we would refer to this kind of change as the Mixed Type from a microscopic angle, which can help us to analyse the performance of algorithms on a problem with changing types. This type of problem can be formulated as follows:

$$\begin{cases}
 f_1(\mathbf{x}, t) = (1 + g(\mathbf{x}_{\text{II}}, t))(x_1 + A(t)\sin(W(t)\pi x_1)) \\
 f_2(\mathbf{x}, t) = (1 + g(\mathbf{x}_{\text{II}}, t))(1 - x_1 + A(t)\sin(W(t)\pi x_1)) \\
 g(\mathbf{x}_{\text{II}}, t) = \sum_{x_i \in \mathbf{x}_{\text{II}}} (x_i + \sigma - G(t))^2 \\
 G(t) = |\sin(0.5\pi t)|, \quad t = \lfloor \frac{t}{\tau_t} \rfloor \lfloor \frac{\tau}{\tau_t} \rfloor \\
 A(t) = 0.05, \quad W(t) = \lfloor 6\sin^\sigma(0.5\pi(t-1)) \rfloor \\
 \sigma \equiv \lfloor \frac{\tau}{\tau_t \rho_t} \rfloor \pmod{3} \\
 \mathbf{x}_{\text{I}} = (x_1) \in [0, 1], \quad \mathbf{x}_{\text{II}} = (x_2, \dots, x_n) \in [-1, 1]^{n-1}
 \end{cases}
 \quad (11)$$

where  $\rho_t$  represents the frequency of type change, and is suggested as  $\rho_t = 5$ , meaning that the current type lasts 5 time windows. If  $\sigma = 0$ , JY6 is a Type-I problem, and the POS is  $x_i = G(t)$ ,  $\forall x_i \in \mathbf{x}_{\text{II}}$ , the POE is referred to Eq. (4) and similar to Fig. 1(b). If  $\sigma = 1$ , JY6 belongs to Type II, where the POS is  $x_i = G(t) - 1$ ,  $\forall x_i \in \mathbf{x}_{\text{II}}$ , and the POE is referred to Eq. (4) and illustrated in Fig. 6(a). If  $\sigma = 2$ , JY6 is a Type-III problem with the POS being  $x_i = -1$ ,  $\forall x_i \in \mathbf{x}_{\text{II}}$ , and the POE being not Eq. (4) since  $g(\mathbf{x}_{\text{II}}, t) \neq 0$ . In this case, the minimum of  $g(\mathbf{x}_{\text{II}}, t)$  is  $g^*(t) = 1 + \sum_{i=2}^n (1 - G(t))^2$ . Thus, the POE is:

$$f_1 + f_2 = (1 + g^*(t))(1 + 2A(t)\sin(W(t)\pi(\frac{f_1 - f_2}{2(1 + g^*(t))} + \frac{1}{2}))) \quad (12)$$

where the POE is illustrated in Fig. 6(b).

#### IV. EXPERIMENTAL STUDY

##### A. Compared Algorithms and Parameter Settings

Four different kinds of algorithms: SPEA2 [27], dNSGA-II (including dNSGA-II-A and dNSGA-II-B) [2], DMOPSO [16] and RM-MEDA+PPS [23], were evaluated on the test instances JY1–JY6. The parameter settings for all the tested

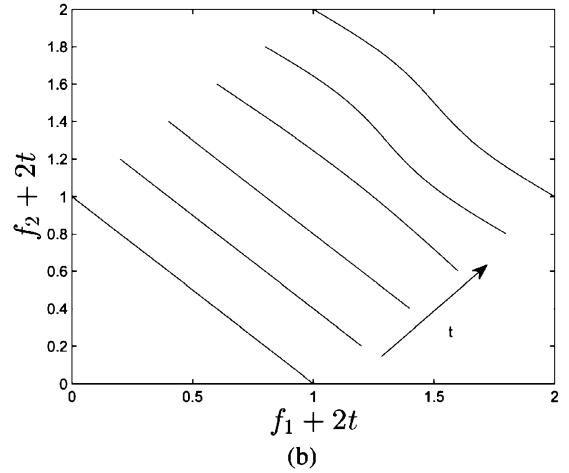
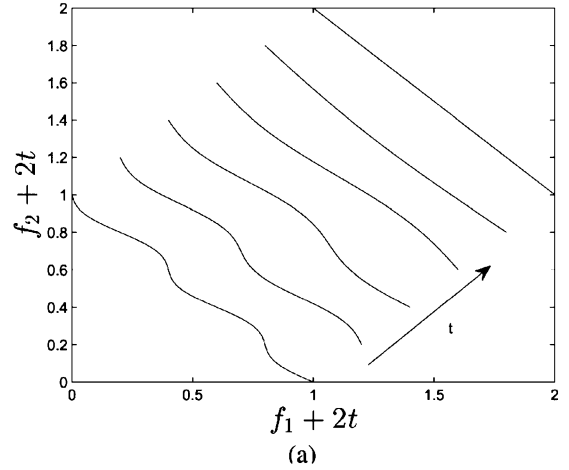


Fig. 6. POE of JY6 with 12 time windows varying from: (a) 0.5 to 1; (b) 1 to 1.5. For a better visualization,  $f_1 + 2t$  and  $f_2 + 2t$  are shown on the  $x$  and  $y$  axes, respectively.

algorithms were inherited from the referenced papers. All the chosen algorithms have strategies to deal with a change except SPEA2. In this paper, SPEA2 was set as follows: 10% randomly selected population members was reevaluated for change detection, and the restart scheme was employed for change responses. For each problem, each tested algorithm, with a population size of 100, was executed 30 runs, and the average results were recorded. The change severity was  $n_t = 10$  for each problem.

To study the effect of the frequency of change ( $\tau_t$ ) on each problem,  $\tau_t$  was set to  $\tau_t = 5, 10, 20, 30, 50$ , respectively. To guarantee the fairness for all the tested algorithms, the total number of changes was set to 40 during the evolution, which is adequate to cover all potential changes in problems JY1–JY6. Besides, 100 more generations were given to each algorithm before the first change to minimize the potential effect of static optimization. Thus, the total generations for running an algorithm were  $100 + 40\tau_t$ .

##### B. Performance Metrics

To compare the performance of algorithms on dynamic benchmark problems, three performance metrics: mean in-

TABLE I. PERFORMANCE OF THE FIVE ALGORITHMS FOR  $JY1$ 

Metric	$\tau_t$	DMOPSO	dNSGA-II-A	dNSGA-II-B	RM-MEDA+PPS	SPEA2
$mIGD$	5	1.7505e-02(2.1832e-03)	3.0635e-02(4.3578e-03)	2.9667e-02(4.0957e-03)	<b>1.1887e-02</b> (6.4518e-03)	2.8827e-02(1.6109e-03)
	10	9.8322e-03(2.0619e-03)	1.3609e-02(1.9756e-03)	1.3112e-02(1.2369e-03)	<b>3.4334e-03</b> (2.1168e-03)	1.0612e-03(5.819e-04)
	20	4.1589e-03(4.0025e-04)	6.0708e-03(1.1323e-03)	5.7493e-03(9.7568e-04)	<b>1.4207e-03</b> (1.2211e-03)	2.5223e-03(1.4854e-04)
	30	2.5544e-03(1.8068e-04)	4.1342e-03(1.8694e-03)	3.7978e-03(1.6643e-03)	<b>3.9195e-04</b> (3.0416e-04)	1.1318e-03(7.9679e-05)
	50	1.5415e-03(7.5273e-05)	2.2128e-03(2.0444e-03)	1.8951e-03(1.0595e-03)	<b>2.4643e-04</b> (2.5436e-06)	3.5671e-04(1.3338e-05)
$S$	5	2.6556e-01(3.0333e-02)	3.3587e-01(6.5558e-02)	3.1612e-01(5.8218e-02)	<b>1.6402e-01</b> (5.6937e-02)	5.1872e-01(5.8583e-02)
	10	1.6482e-01(2.7931e-02)	1.4006e-01(3.1718e-02)	1.3629e-01(2.9348e-02)	<b>5.1175e-02</b> (2.6364e-02)	1.6277e-01(1.8001e-02)
	20	7.5798e-02(5.8709e-03)	5.0508e-02(1.8038e-02)	4.3022e-02(8.1048e-03)	<b>1.5625e-02</b> (1.0586e-02)	3.7293e-02(3.9097e-03)
	30	4.9653e-02(3.4691e-03)	2.6325e-02(8.6496e-03)	2.5487e-02(6.9684e-03)	<b>5.1341e-03</b> (2.0435e-03)	1.6962e-02(1.0997e-03)
	50	3.0114e-02(1.4108e-03)	2.3059e-02(4.5275e-03)	1.3802e-02(3.5865e-03)	<b>3.4159e-03</b> (6.8685e-05)	5.6034e-03(2.4551e-04)
$Acc\_alt$	5	<b>8.2926e-01</b> (1.5507e-01)	2.3394(3.8127e-01)	2.2555(3.8564e-01)	8.4673e-01(6.6102e-01)	2.1043(1.1853e-01)
	10	4.0317e-01(1.1826e-01)	7.7276e-01(1.2143e-01)	7.4433e-01(8.3087e-02)	<b>1.7091e-01</b> (1.5467e-01)	5.0056e-01(3.3712e-02)
	20	1.4001e-01(1.6677e-02)	2.9238e-01(6.2256e-02)	2.7648e-01(6.1546e-02)	<b>6.2348e-02</b> (7.1365e-02)	7.7304e-03(6.9241e-03)
	30	7.5731e-02(6.9719e-03)	1.8906e-01(1.2255e-01)	1.7036e-01(1.0135e-01)	<b>9.4076e-03</b> (1.4717e-03)	2.7575e-02(2.8931e-03)
	50	3.9579e-02(2.7225e-03)	8.3860e-02(8.2915e-02)	7.2788e-02(5.6941e-02)	<b>3.6218e-03</b> (7.909e-05)	6.4359e-03(3.0431e-04)

TABLE II. PERFORMANCE OF THE FIVE ALGORITHMS FOR  $JY2$ 

Metric	$\tau_t$	DMOPSO	dNSGA-II-A	dNSGA-II-B	RM-MEDA+PPS	SPEA2
$mIGD$	5	1.7023e-02(2.6104e-03)	3.1053e-02(3.6117e-03)	3.0857e-02(5.0833e-03)	<b>1.0574e-02</b> (4.9453e-03)	2.9061e-02(1.7009e-03)
	10	9.6873e-03(1.3589e-03)	1.3604e-02(1.3956e-03)	1.3439e-02(1.2181e-03)	<b>4.1500e-03</b> (2.2232e-03)	1.0962e-02(5.7798e-04)
	20	4.3342e-03(3.7999e-04)	6.4412e-03(1.6320e-03)	5.5738e-03(7.8922e-04)	<b>2.1881e-03</b> (5.5361e-04)	2.3901e-03(1.3463e-04)
	30	2.6505e-03(1.6112e-03)	3.7606e-03(1.2307e-03)	3.6413e-03(1.4149e-03)	1.9705e-03(6.3609e-04)	<b>1.0012e-03</b> (4.1347e-05)
	50	1.5136e-03(9.9003e-05)	2.1255e-03(1.1132e-03)	2.0777e-03(9.7886e-04)	1.7861e-03(2.3348e-06)	<b>2.9945e-04</b> (9.4734e-06)
$S$	5	2.6163e-01(4.0358e-02)	3.2274e-01(7.2115e-02)	2.9664e-01(6.5859e-02)	<b>1.5227e-01</b> (4.5242e-02)	4.9946e-01(5.0929e-02)
	10	1.5257e-01(1.9416e-02)	1.3429e-01(3.0646e-02)	1.3054e-01(3.3843e-02)	<b>4.4123e-02</b> (2.5313e-02)	1.6225e-01(1.4215e-02)
	20	7.3297e-02(6.9027e-03)	5.2618e-02(2.2927e-02)	4.8377e-02(1.785e-02)	<b>9.9628e-03</b> (7.2578e-03)	3.1382e-02(2.0637e-03)
	30	4.7107e-02(2.8012e-03)	2.6585e-02(7.7333e-03)	2.6537e-02(7.6141e-03)	<b>5.4329e-03</b> (3.6475e-03)	2.0552e-02(2.2704e-03)
	50	2.6597e-02(1.8521e-03)	1.4426e-02(5.2395e-03)	1.4761e-02(4.4598e-03)	<b>3.3358e-03</b> (8.9749e-05)	4.4667e-03(1.4429e-04)
$Acc\_alt$	5	8.1131e-01(1.8373e-01)	2.3648(3.0883e-01)	2.3641(4.6456e-01)	<b>6.6279e-01</b> (4.9213e-01)	2.0952(1.6479e-01)
	10	4.0087e-01(7.0847e-02)	7.9113e-01(8.9873e-02)	7.8707e-01(8.7395e-02)	<b>1.7276e-01</b> (1.7289e-01)	5.1393e-01(3.5427e-02)
	20	1.5882e-01(1.6297e-02)	3.4223e-01(1.6277e-01)	2.7795e-01(5.3495e-02)	<b>4.3763e-02</b> (3.2323e-02)	8.6175e-02(5.9986e-03)
	30	9.0053e-02(6.1636e-03)	1.7225e-01(7.4565e-02)	1.6743e-01(8.7188e-02)	3.2918e-02(4.5534e-02)	<b>3.1506e-02</b> (1.7452e-03)
	50	4.8103e-02(3.5654e-03)	9.1056e-02(6.0851e-02)	8.5083e-02(5.2819e-02)	2.1217e-02(1.2671e-04)	<b>8.3443e-03</b> (3.8368e-04)

TABLE III. PERFORMANCE OF THE FIVE ALGORITHMS FOR  $JY3$ 

Metric	$\tau_t$	DMOPSO	dNSGA-II-A	dNSGA-II-B	RM-MEDA+PPS	SPEA2
$mIGD$	5	2.3599e-02(6.2932e-03)	9.7288e-03(7.8936e-04)	<b>9.2812e-03</b> (1.6771e-03)	1.4777e-02(1.0017e-03)	1.9657e-02(4.0775e-04)
	10	1.6048e-02(5.6784e-03)	<b>9.1379e-03</b> (1.5465e-03)	9.3032e-03(4.3768e-04)	1.0383e-02(2.4557e-03)	1.2961e-02(1.5897e-04)
	20	1.1015e-02(6.8761e-03)	9.1724e-03(9.3572e-05)	9.1776e-03(6.4867e-05)	<b>6.9615e-03</b> (3.0059e-03)	1.0056e-02(6.5843e-05)
	30	8.6411e-03(5.4505e-03)	7.7523e-03(2.8595e-03)	8.3376e-03(2.4002e-03)	<b>6.7161e-03</b> (3.4349e-03)	9.4788e-03(4.9057e-05)
	50	7.8713e-03(5.6219e-03)	8.6505e-03(1.6988e-03)	8.3908e-03(2.0593e-03)	<b>5.1046e-03</b> (3.6133e-03)	9.1983e-03(4.5233e-05)
$S$	5	3.7776e-01(7.5253e-02)	<b>4.0884e-02</b> (1.5454e-02)	5.1484e-02(2.8655e-02)	2.3346e-01(2.3182e-02)	2.4299e-01(2.6076e-02)
	10	2.5831e-01(7.1518e-02)	<b>2.7418e-02</b> (1.2895e-02)	2.6633e-02(1.0049e-02)	1.2005e-01(4.4034e-02)	9.2738e-02(7.9343e-03)
	20	1.4092e-01(4.2812e-02)	<b>2.1187e-02</b> (1.0407e-02)	1.9624e-02(7.8504e-03)	5.6613e-02(4.5213e-02)	4.0906e-02(2.8493e-03)
	30	1.0815e-01(4.2515e-02)	<b>1.5595e-02</b> (5.4233e-03)	1.6945e-02(8.0503e-03)	2.4563e-02(3.2373e-02)	2.7527e-02(2.2372e-03)
	50	8.1833e-02(3.5264e-02)	<b>1.3713e-02</b> (1.9704e-03)	1.4625e-02(3.9032e-03)	8.7016e-03(3.4709e-03)	1.8547e-02(1.9695e-03)
$Acc\_alt$	5	1.2071(4.7153e-01)	<b>4.4001e-01</b> (7.3597e-02)	4.4006e-01(1.2819e-01)	8.9858e-01(7.4953e-02)	1.1656(2.6675e-02)
	10	7.5635e-01(3.6404e-01)	<b>3.7994e-01</b> (6.8552e-02)	3.8691e-01(2.2704e-02)	5.1669e-01(1.4451e-01)	6.5509e-01(9.0429e-03)
	20	5.1237e-01(4.0428e-01)	<b>3.6867e-01</b> (9.8961e-03)	3.6971e-01(6.3597e-03)	2.7865e-01(1.5029e-01)	4.4761e-01(4.2332e-03)
	30	3.8971e-01(3.0469e-01)	3.0592e-01(1.1904e-01)	3.3188e-01(1.0032e-02)	<b>2.5598e-01</b> (1.6424e-01)	4.0132e-01(2.6642e-03)
	50	3.5029e-01(2.9485e-01)	3.4264e-01(6.9677e-02)	3.3013e-01(8.7437e-02)	<b>1.7519e-01</b> (1.6926e-01)	3.7629e-01(1.9096e-03)

verted generational distance ( $mIGD$ ) [23], Schott's spacing metric ( $S$ ) [19], and alternative accuracy metric ( $Acc\_alt$ ) [1], found in the literature, were used to measure the performance of these algorithms.  $mIGD$  and  $S$  can evaluate the convergence and diversity performance of algorithms, respectively; while  $Acc\_alt$  can measure the approximation quality in convergence and diversity. All the metrics were calculated just before a change occurs.

### C. Experimental Results

The experimental results reported in Tables I to VI are the average values and standard errors of 30 independent runs, where the best results are highlighted in bold.

For the  $JY1$  problem, RM-MEAD+PPS performs better than the other compared algorithms regarding the three metrics for the tested change frequencies except for  $\tau_t = 5$ . In the case

of  $\tau_t = 5$ , RM-MEAD+PPS achieves the best results on the convergence and diversity performance, but is slightly worse than DMOPSO for  $Acc\_alt$ . dNSGA-II-A and dNSGA-II-B have similar performance on the three metrics. SPEA2 appears to be better than the dNSGA-II variants on those metrics.

Table II shows that, for  $JY2$ , RM-MEAD+PPS and SPEA2 achieve better performance on the metrics than DMOPSO and dNSGA-II. While RM-MEAD+PPS performs the best on the spacing metric, SPEA2 outperforms the others on  $mIGD$  and  $Acc\_alt$  for a slow frequency change. Besides, DMOPSO provides better results than dNSGA-II on  $mIGD$  and  $Acc\_alt$ , but worse results than dNSGA-II on the spacing metric. RM-MEAD+PPS gives the best results on  $mIGD$  and  $Acc\_alt$  when the frequency of change is quite low.

The results presented in Table III show that, the dNSGA-II variants work well on the metrics compared for  $JY3$ , and

TABLE IV. PERFORMANCE OF THE FIVE ALGORITHMS FOR  $JY4$ 

Metric	$\tau_t$	DMOPSO	dNSGA-II-A	dNSGA-II-B	RM-MEDA+PPS	SPEA2
$mIGD$	5	1.1531e-01(1.4708e-02)	1.1235e-01(1.3122e-02)	1.0864e-01(1.4759e-02)	<b>3.5004e-02</b> (1.5155e-02)	1.0815e-01(5.0667e-03)
	10	1.0915e-01(1.8752e-02)	5.4806e-02(4.4931e-03)	5.4508e-02(4.6836e-03)	<b>1.8793e-02</b> (8.0378e-03)	8.3751e-02(6.5767e-03)
	20	9.4708e-02(1.3054e-02)	3.0211e-02(2.9775e-03)	2.9543e-02(3.3004e-03)	<b>8.5521e-03</b> (1.3386e-03)	6.5489e-02(3.8453e-03)
	30	9.2561e-02(1.1793e-02)	2.3554e-02(3.5654e-03)	2.2571e-02(3.3538e-03)	<b>6.4392e-03</b> (1.0062e-03)	5.3773e-02(3.0794e-03)
	50	1.3863e-01(8.6714e-03)	1.9102e-02(3.6081e-03)	1.8206e-02(3.1381e-03)	<b>5.4079e-03</b> (6.1524e-04)	4.2854e-02(2.9763e-03)
$S$	5	1.3346(2.5285e-01)	2.8939e-01(7.4421e-02)	2.5686e-01(8.9607e-02)	<b>1.5886e-01</b> (5.0897e-02)	3.4684e-01(2.7206e-02)
	10	1.4748(2.2473e-01)	1.1873e-01(2.3083e-02)	1.2274e-01(3.5226e-02)	<b>1.0384e-01</b> (3.0214e-02)	1.9028e-01(1.1818e-02)
	20	1.2989(1.9391e-01)	5.2033e-02(1.0515e-02)	<b>5.1543e-02</b> (1.1842e-02)	8.2754e-02(3.3868e-02)	1.4351e-01(1.0248e-02)
	30	1.0563(1.5122e-01)	3.4739e-02(8.4508e-03)	<b>3.3261e-02</b> (6.8050e-03)	9.1682e-02(3.5839e-02)	1.2026e-01(9.7428e-03)
	50	8.5579e-01(4.9143e-02)	3.3013e-02(5.5601e-03)	<b>2.9316e-02</b> (5.2914e-03)	1.1716e-01(2.5033e-02)	1.0029e-01(6.9197e-03)
$Acc\_alt$	5	3.2494(6.4043e-01)	2.9156(5.2251e-01)	2.728(4.5239e-01)	<b>6.6395e-01</b> (4.8712e-01)	1.3589(1.7063e-01)
	10	3.1391(8.4615e-01)	1.0506(1.1639e-01)	1.0456(1.354e-01)	<b>2.6967e-01</b> (2.0685e-01)	2.3564(2.9584e-01)
	20	2.8937(5.5202e-01)	6.4533e-01(9.7037e-02)	6.2032e-01(1.3149e-01)	<b>1.5148e-01</b> (1.3066e-01)	4.9593(3.5368e-01)
	30	3.3866(8.5625e-01)	6.3378e-01(1.3218e-01)	6.0391e-01(1.2209e-01)	<b>2.9695e-01</b> (1.8668e-01)	7.1978(4.2115e-01)
	50	10.4612(48.491e-01)	7.9814e-01(1.9514e-01)	7.3339e-01(1.7811e-01)	<b>6.1189e-01</b> (2.6682e-01)	9.8753(4.2644e-01)

TABLE V. PERFORMANCE OF THE FIVE ALGORITHMS FOR  $JY5$ 

Metric	$\tau_t$	DMOPSO	dNSGA-II-A	dNSGA-II-B	RM-MEDA+PPS	SPEA2
$mIGD$	5	5.6935e-03(1.1383e-03)	<b>4.2372e-04</b> (1.4213e-05)	4.2689e-04(1.2964e-05)	1.1192e-02(7.7954e-04)	1.5369e-02(4.4615e-04)
	10	<b>3.1828e-03</b> (6.8099e-04)	7.9445e-03(4.0095e-06)	7.9442e-03(5.6882e-06)	3.4588e-03(1.6137e-03)	6.6841e-03(2.6297e-04)
	20	1.7495e-03(2.2214e-04)	<b>3.3667e-04</b> (2.9926e-06)	3.3799e-04(2.5178e-06)	1.3457e-03(1.5505e-04)	1.7564e-03(8.152e-05)
	30	1.2123e-03(1.2334e-04)	<b>3.3336e-04</b> (2.4069e-06)	3.3422e-04(3.4417e-06)	1.3125e-03(2.3341e-05)	7.0424e-04(4.3609e-05)
	50	8.2692e-04(4.9069e-05)	3.3149e-04(2.1079e-06)	3.3209e-04(2.1056e-06)	1.3234e-03(2.1125e-06)	<b>2.9344e-04</b> (9.4517e-06)
$S$	5	1.4267e-01(2.7347e-02)	5.6398e-02(1.0967e-02)	<b>1.5931e-02</b> (8.4356e-03)	1.8564e-01(1.7383e-02)	2.1955e-01(2.0356e-02)
	10	7.3482e-02(1.8245e-02)	<b>1.1493e-02</b> (2.6193e-03)	1.1558e-02(3.9014e-03)	5.3415e-02(3.0468e-02)	9.7312e-02(1.0995e-02)
	20	3.6158e-02(6.3071e-03)	<b>1.0423e-02</b> (1.1672e-03)	1.1077e-02(2.7653e-03)	5.5762e-03(4.8871e-03)	2.6076e-02(2.0936e-03)
	30	2.2031e-02(2.9194e-03)	1.0802e-02(1.7855e-03)	1.0155e-02(3.0517e-04)	<b>3.3434e-03</b> (3.6308e-04)	1.0246e-02(7.3913e-04)
	50	1.2774e-02(1.3045e-03)	1.0382e-02(1.1476e-03)	1.0097e-02(8.9283e-05)	<b>3.2048e-03</b> (3.9494e-05)	4.4926e-03(1.8419e-04)
$Acc\_alt$	5	1.6824e-01(3.8889e-02)	<b>1.0347e-02</b> (2.1637e-03)	1.0456e-02(1.8487e-03)	5.8922e-01(5.3415e-02)	7.8318e-01(2.8725e-02)
	10	<b>8.9992e-02</b> (2.2999e-02)	2.4557e-01(3.2555e-04)	2.4556e-01(2.8224e-04)	1.3747e-01(8.0553e-02)	2.5758e-01(1.3093e-02)
	20	4.9232e-02(6.4676e-03)	<b>7.0435e-03</b> (1.3899e-04)	7.0903e-03(1.9179e-04)	3.9703e-02(7.8549e-03)	5.6895e-02(3.0382e-03)
	30	3.3817e-02(4.3162e-03)	<b>6.8422e-03</b> (1.0683e-04)	6.8974e-03(1.1344e-04)	3.7733e-02(9.4809e-04)	2.1641e-02(1.4409e-03)
	50	2.3115e-02(1.7119e-03)	<b>6.7283e-03</b> (7.1826e-05)	6.7458e-03(8.3493e-05)	3.8104e-02(7.6103e-05)	7.9656e-03(3.0688e-04)

TABLE VI. PERFORMANCE OF THE FIVE ALGORITHMS FOR  $JY6$ 

Metric	$\tau_t$	DMOPSO	dNSGA-II-A	dNSGA-II-B	RM-MEDA+PPS	SPEA2
$mIGD$	5	<b>6.9106e-02</b> (6.0525e-03)	2.3105e-01(1.5253e-02)	2.3143e-01(1.4686e-02)	7.2139e-02(4.3618e-03)	2.3185e-01(5.4408e-03)
	10	5.9989e-02(8.5235e-03)	2.0888e-01(1.1397e-02)	2.0874e-01(9.7265e-03)	<b>5.7314e-02</b> (3.3356e-03)	2.1644e-01(2.8528e-03)
	20	<b>4.5686e-02</b> (2.3581e-03)	1.5452e-01(2.2481e-03)	1.5507e-01(2.4479e-03)	4.7269e-02(2.2734e-03)	1.6797e-01(1.3366e-03)
	30	<b>2.8953e-02</b> (2.4439e-03)	1.5047e-01(1.5099e-03)	1.5015e-01(1.1122e-03)	3.9275e-02(1.2415e-03)	1.4921e-01(7.1232e-04)
	50	<b>2.0264e-02</b> (1.7566e-03)	1.4442e-01(6.6098e-04)	1.4409e-01(8.2908e-04)	2.7779e-02(5.2546e-04)	1.4399e-01(3.3372e-04)
$S$	5	<b>3.0612e-01</b> (3.1451e-01)	5.5178e-01(9.4507e-02)	5.5683e-01(9.0453e-02)	4.7742e-01(4.9865e-02)	1.0746(1.4076e-01)
	10	3.5514e-01(2.0841e-01)	4.1926e-01(1.0313e-01)	4.1641e-01(8.2472e-02)	<b>3.1512e-01</b> (4.0486e-02)	4.7768e-01(4.2067e-02)
	20	2.8144e-01(4.1248e-02)	2.9431e-01(7.2912e-02)	2.8448e-01(6.3551e-02)	<b>1.4857e-01</b> (1.7165e-02)	1.5735e-01(1.1527e-02)
	30	1.8302e-01(2.7034e-02)	2.0476e-01(6.9789e-02)	2.1338e-01(7.1462e-02)	7.8132e-02(9.3671e-03)	<b>7.7071e-02</b> (5.7147e-03)
	50	1.0997e-01(2.1603e-02)	1.1227e-01(5.1807e-02)	9.8389e-02(3.9049e-02)	<b>2.6506e-02</b> (4.7213e-03)	3.0797e-02(1.4013e-03)
$Acc\_alt$	5	<b>5.5753(1.3689)</b>	156.2234(16.7679)	155.4011(14.0357)	10.7839(0.80255)	171.9286(5.4616)
	10	<b>4.9454(5.6518e-01)</b>	97.4117(8.4376)	97.6732(7.2611)	6.7528(0.53477)	119.6014(2.9582)
	20	<b>4.3684(2.0666e-01)</b>	74.3456(2.9994e-01)	74.1743(3.2581e-01)	5.3015(2.8883e-01)	88.3464(1.4432)
	30	<b>2.2289(1.6516e-01)</b>	72.7139(1.9639e-01)	72.7308(1.3627e-01)	3.9388(1.1924e-01)	78.7827(7.5836e-01)
	50	<b>1.5567(9.7405e-02)</b>	72.0976(8.3328e-02)	72.0691(6.6741e-02)	2.8378(4.3354e-02)	74.0895(4.0174e-01)

dNSGA-II-A is much better than dNSGA-II-B to obtain well-diversified solutions. The results of DMOPSO are similar to those of SPEA2 except on the spacing metric where SPEA2 offers better results than DMOPSO.

For  $JY4$ , RM-MEDA+PPS, again, works better than the other algorithms on  $mIGD$  and  $Acc\_alt$ . However, when the frequencies of change are 20 and over, RM-MEDA+PPS fails to achieve a set of diverse solutions; on the contrary, dNSGA-II-B offers the best results on this metric. It is also clear from Table IV that the performance of each tested algorithm on  $Acc\_alt$  becomes worse when the frequency of change is higher. A possible reason for this is that a slow frequency of change enables the algorithms to achieve better convergence and diversity performance on each disconnected Pareto front of  $JY4$ , but meanwhile it may lead to the distance between two consecutive disconnected Pareto fronts approximated moving

away from each other. For this reason, DMOPSO and SPEA2 fail to give good results for  $JY4$ .

For  $JY5$ , the results are somehow divergent. In the cases of  $\tau_t = 5, 20,$  and  $30$ , dNSGA-II-A successfully provides the best results on the compared metrics. For a higher frequency of change, RM-MEDA+PPS achieves a set of well-distributed solutions. DMOPSO gives the best  $mIGD$  and  $Acc\_alt$  results when the frequency of change is 10. SPEA2 converges very well towards the Pareto front for a very slow change frequency.

The Mixed Type of change imposes some challenges to these algorithms. For  $JY6$ , DMOPSO has no difficulties in achieving well-converged solutions, while RM-MEDA+PPS gives the best results regarding the distribution of the approximated solutions for  $\tau_t = 10, 20,$  and  $50$ . It is also clear from Table VI that, DMOPSO provides the best results for  $Acc\_alt$ . The dNSGA-II variants and SPEA2 give poor performance

for *Acc\_alt*. This is because the *mIGD* results imply that they cannot track the landscape change when the type of the problem changes over time.

## V. CONCLUSIONS AND FUTURE WORK

An extensive study of the current DMOPs used to assess the performance of algorithms showed that there is lack of test functions that can occur in real-world life for dynamic multi-objective optimization. The commonly used DMOPs are so simple that some characteristics are excluded, such as a mixed (convexity and concavity) POF, a more powerful diversity-resistant structure in the problem, a stronger correlation between variables and a mixed type of problem that can jump between different types.

To address the above shortcomings, this paper presents a generic scheme to generate desired benchmark functions that can compare the performance of different DMOAs. Furthermore, experiments were carried out on some test instances generated by the benchmark generator. The results show that these test instances are effective and can be used for comparing the general performance of algorithms.

Despite that our proposed benchmark generator can produce a series of features that are rarely tested in the literature, further research is needed regarding how to extend the generator to many-objective problems where the number of objectives is easy to scale up. Besides, a standard test suite for DMOPs is urgently required to promote the research on dynamic multi-objective optimization.

## ACKNOWLEDGMENT

The authors would like to thank Dr. Aimin Zhou, Dr. Mardé Helbig, and Prof. Andries P. Engelbrecht for providing their kind help and source codes for the experiments. This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) of UK under Grant EP/K001310/1.

## REFERENCES

- [1] M. Camara, J. Ortega, F. J. Toro, "Approaching dynamic multi-objective optimization problems by using parallel evolutionary algorithms," *Advances in Multi-objective Nature Inspired Computing, Studies in Computational Intelligence*, vol. 272, 2010, pp. 63–86.
- [2] K. Deb, S. Agarwal, A. Pratap, T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *Kanpur, India, Tech. Report 200001*, 2000.
- [3] K. Deb, N. Rao U.B., S. Karthik, "Dynamic multi-objective optimization and decision-making using modified NSGA-II: a case study on hydro-thermal power scheduling," in *Proc. Int. Conf. Evol. Multi-criterion Optimization*, 2007, pp. 803–917.
- [4] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, "Scable test problems for evolutionary multi-objective optimization," *Kanpur Genetic Algorithms Lab (KanGAL), Indian Inst. Technol., KanGAL Report 2001001*, 2001.
- [5] M. Farina, K. Deb, P. Amato, "Dynamic multiobjective optimization problems: test cases, approximations, and applications," *IEEE Trans. Evol. Comput.*, vol. 8, no. 5, pp. 425–442, 2004.
- [6] C. Goh, K. C. Tan, "A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 1, pp. 103–127, 2009.
- [7] M. Greeff, A. P. Engelbrecht, "Solving dynamic multi-objective problems with vector evaluated particle swarm optimisation," in *Proc. World Congr. Comput. Intell.*, Hong Kong, 2008, pp. 2917–2924.
- [8] S. Guan, Q. Chen, W. Mo, "Evolving dynamic multi-objective optimization problems with objective replacement," *Artif. Intell. Rev.*, vol. 23, pp. 267–293, 2005.
- [9] M. Helbig, A. P. Engelbrecht, "Issues with performance measures for dynamic multi-objective optimisation," in *Proc. IEEE Symp. Comput. Intell. Dynamic and Uncertain Environments*, 2013, pp. 84–91.
- [10] M. Helbig, A. P. Engelbrecht, "Performance measures for dynamic multi-objective optimisation algorithms," *Inform. Sci.*, vol. 250, pp. 61–81, 2013.
- [11] M. Helbig, A. P. Engelbrecht, "Benchmarks for dynamic multi-objective optimisation," in *Proc. 2013 IEEE Symp. on Comput. Intell. in Dynamic and Uncertain Environments*, 2013, pp. 84–91.
- [12] L. Huang, I.H. Suh, A. Abraham, "Dynamic multi-objective optimization based on membrane computing for control of time-varying unstable plants," *Inform. Sci.*, vol. 181, no. 11, pp. 2370–2391, 2011.
- [13] S. Huband, P. Hingston, L. Barone, L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 2, pp. 477–506, 2006.
- [14] Y. Jin, B. Sendhoff, "Constructing dynamic optimization test problems using the multi-objective optimization concept," in *EvoWorkshops 2004: Appl. Evol. Comput.*, 2004, pp. 525–536.
- [15] J. D. Knowles, L. Thiele, E. Zitzler, "A tutorial on the performance assessment of stochastic multiobjective optimizers," *Computer Engineering and Networks Laboratory, ETH Zurich, Switzerland, Tech. Report 214*, 2006.
- [16] M. S. Lechuga, "Multi-objective optimisation using sharing in swarm optimisation algorithms," *Ph.D Dissertation, University of Birmingham, Birmingham, UK*, 2009.
- [17] H. Li, Q. Zhang, "Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no.2, pp. 284–302, 2009.
- [18] J. Mehnen, G. Rudolph, T. Wagner, "Evolutionary optimization of dynamic multiobjective functions," *Universitat Dortmund, Dortmund, Germany, Tech. Report FI-204/06*, 2006.
- [19] J. R. Schott, "Fault tolerance design using single and multi-criteria optimization," *M.S. thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology*, 1995.
- [20] P. P. Wu, D. Campbell, T. Merz, "Multiobjective four-dimensional vehicle motion planning in large dynamic environments," *IEEE Trans. Syst., Man, Cybern. Part B: Cybern.*, vol. 41, no. 3, pp. 621–634, 2011.
- [21] Q. Zhang, A. Zhou, Y. Jin, "RM-MEDA: a regular model based multiobjective estimation of distribution algorithm," *IEEE Trans. Evol. Comput.* vol. 12, no. 1, pp. 41–63, 2008.
- [22] Z. Zhang, "Multiobjective optimization immune algorithm in dynamic environments and its application to greenhouse control," *Appl. Soft Comput.*, vol. 8, no. 2, pp. 959–971, 2008.
- [23] A. Zhou, Y. Jin, Q. Zhang, "A population prediction strategy for evolutionary dynamic multiobjective optimization," *IEEE Trans. Cybern.*, vol. 44, no. 1, pp. 40–53, 2014.
- [24] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, "Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization," *Proc. Int. Conf. Evol. Multi-criterion Optimization, LNCS*, vol. 4403, 2007, pp. 54–73.
- [25] E. Zitzler, "Evolutionary algorithms for multiobjective optimization: methods and applications," *Ph.D. Dissertation, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland*, 1999.
- [26] E. Zitzler, L. Thiele, "Multiobjective optimization using evolutionary algorithms—a comparative case study," in *Proc. 10th Int. Conf. Parallel Problem Solving from Nature, LNCS* vol. 1498, 1998, pp. 292–301.
- [27] E. Zitzler, M. Laumanns, L. Thiele, "SPEA2: improving the strength pareto evolutionary algorithm," *Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Zurich, Switzerland, TIK Report 103*, 2001.
- [28] E. Zitzler, K. Deb, L. Thiele, "Comparison of multiobjective evolutionary algorithms: empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.