Benchmark Problems for CEC2018 Competition on Dynamic Multiobjective Optimisation

Shouyong Jiang[1], Shengxiang Yang[2], Xin Yao[3], Kay Chen Tan[4], Marcus Kaiser[1], and Natalio Krasnogor[1]

[1]School of Computing, Newcastle University, Newcastle upon Tyne, NE4 5TG, U.K.
{math4neu@gmail.com, {marcus.kaiser, natalio.krasnogor}@ncl.ac.uk}
[2]School of Computer Science and Informatics, De Montfort University, Leicester, LE1 9BH, U.K.
{email: syang@dmu.ac.uk}
[3]School of Computer Science, University of Birmingham, Birmingham, B15 2TT, U.K.
{x.yao@cs.bham.ac.uk}
[4]Department of Computer Science, City University of Hong Kong, HK
{kaytan@cityu.edu.hk}

1 Introduction

The past decade has witnessed a growing amount of research interest in dynamic multiobjective optimisation, a challenging yet very important topic that deals with problems with multi-objective and time-dependent properties [3–7,10]. Due to the presence of dynamics, dynamic multiobjective problems (DMOPs) are more complex and challenging than static multiobjective problems. As a result, evolutionary algorithms (EAs) face great difficulties in solving them. Generally speaking, DMOPs pose at least three main challenges. First, environmental changes can exhibit any dynamics. A variety of dynamics pose different levels of difficulties to algorithms, and there is no single change reaction mechanism that can handle all dynamics. Second, diversity, the key driving force of population-based algorithms, is sensitive to dynamics and therefore difficult to be well maintained. Finally, often than not the response time for environmental changes is rather tight for algorithms. Time restriction on DMOPs requires algorithms to reach a good balance between diversity and convergence such that any environmental changes can be promptly handled in order to closely track time-varying Pareto fronts or sets. All these suggest there be a great need for new methodologies for tacking DMOPs.

Benchmark problems are of great importance to algorithm analysis, which helps algorithm designers and practitioners to better understand the strengths and weaknesses of evolutionary algorithms. In dynamic multiobjective optimisation, there exist several widely used test suites, including FDA [4] and dMOP [6]. However, these problem suites only represent one or several aspects of real-world scenarios. For example, the FDA and dMOP functions have no detection difficulty for algorithms. Environmental changes involved in these problems can be easily detected with one re-evaluation of a random population member. Real-life environmental changes should not be so simple. It has also been recognised that most existing DMOPs are a direct modification of popular static test suites, e.g. ZDT [17] and DTLZ [2]. As a result, the DMOPs are more or less the same regarding their problem properties, and therefore are of limited use when a comprehensive algorithm analysis is pursued. Furthermore, another worrying characteristic of most existing DMOPs is that static problem properties overweigh too much dynamics [1, 14]. A problem property (e.g. strong variable dependency) that is challenging for static multiobjective optimisation may not be a good candidate for dynamic multiobjective optimisation [5]. One reason for this is that a failure of algorithms for DMOPs is not due to the presence of dynamics, but rather the static property. It is therefore likely to get a misleading conclusion on the performance of algorithms when such DMOPs are used. In a nutshell, a set of diverse and unbiased benchmark test problems for a systematic study of evolutionary algorithms are greatly needed in the area.

In this competition, a total of 14 benchmark functions are introduced, covering diverse properties which nicely represent various real-world scenarios, such as time-dependent PF/PS geometries, irregular PF shapes, disconnectivity, knee, and so on. Through suggesting a set of benchmark functions with a good representation of various real-world scenarios, we aim to promote the research on evolutionary dynamic multiobjective optimisation. All the benchmark functions have been implemented in MATLAB code and C/C++ code.

2 Summary of 14 Test Problems

The proposed test suite (called DF in this competition) has 9 nine biobjective and 5 triobjective problems. The main dynamic characteristics that each problem involves are briefly tabulated in Table 1.

3 Problem Definitions

The following notations are widely used in each problem definition:

- M: the number of objectives
- n: the number of decision variables
- x_i: the i-th decision variable
- f_i: the i-th objective function
- τ: generation counter
- τ_t: frequency of change
- n_t: severity of change
- $t = \frac{1}{n_t}\lfloor\frac{\tau}{\tau_t}\rfloor$: time instant

Table 1: Main dynamic characteristics of the 13 test problems

Problem	#objectives	Dynamics	Remarks
DF1	2	mixed convexity-concavity, location of optima	dynamic PF and PS
DF2	2	switch of position-related variable, location of optima	static convex PF, dynamic PS, severe diversity loss
DF3	2	mixed convexity-concavity, variable-linkage, location of optima	dynamic PF and PS
DF4	2	variable-linkage, PF range, bounds of PS	dynamic PF and PS
DF5	2	number of knee regions, local of optima	dynamic PF and PS
DF6	2	mixed convexity-concavity, multimodality, location of optima	dynamic PF and PS
DF7	2	PF range, location of optima	convex PF, static PS centroid, dynamic PF and PS
DF8	2	mixed convexity-concavity, distribution of solutions, location of optima	static PS centroid, dynamic PF and PS, variable-linkage
DF9	2	number of disconnected PF segments, location of optima	dynamic PS and PF, variable-linkage
DF10	3	mixed convexity-concavity, location of optima	dynamic PS and PF, variable-linkage
DF11	3	size of PF region, PF range, location of optima	dynamic PS and PF, concave PF, variable-linkage
DF12	3	number of PF holes, location of optima	dynamic PS, static concave PF, variable-linkage
DF13	3	number of disconnected PF segments, location of optima	dynamic PS and PF, the PF can be a continuous convex or concave segment, or several disconnected segments
DF14	3	degenerate PF, number of knee regions, location of optima	dynamic PS and PF, variable-linkage

3.1 DF1 (dMOP2 [6])

$$min \begin{cases} f_1(x) = x_1 \\ f_2(x) = g(x)\left(1 - (\frac{x_1}{g(x)})^{H(t)}\right) \end{cases} \tag{1}$$

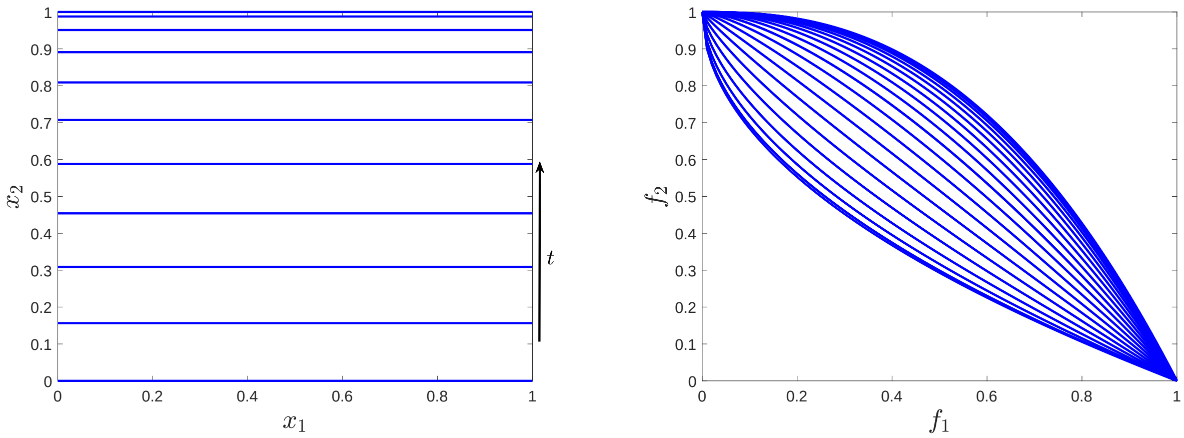

Figure 1: Illustration of the PS and PF of DF1.

with

$$g(x) = 1 + \sum_{i=2}^{n} (x_i - G(t))^2$$

where $H(t) = 0.75\sin(0.5\pi t) + 1.25, G(t) = |\sin(0.5\pi t)|$ and the search space is $[0, 1]^n$.

The PF and PS at time t can be described as:

$$\begin{aligned} &\text{PS}(t) : 0 \leq x_1 \leq 1, x_i = G(t), i = 2, \ldots, n \\ &\text{PF}(t) : f_2 = 1 - {f_1}^{H(t)}, 0 \leq f_1 \leq 1 \end{aligned}$$

Remark: DF1 has a simple dynamic on the PS, and its PF geometry changes from concavity to convexity, or vice versa. This problem is used to assess the tracking ability of concavity or convexity variations.

3.2 DF2 (modified dMOP3 [6])

$$min \begin{cases} f_1(x) = x_r \\ f_2(x) = g(x)(1 - \sqrt{f_1/g}) \end{cases} \tag{2}$$

with

$$g(x) = 1 + \sum_{i=\{1,\ldots,n\}/\{r\}} (x_i - G(t))^2$$

where $G(t) = |\sin(0.5\pi t)|$, $r = 1 + \lfloor (n-1)G(t) \rfloor$. The search space is $[0, 1]^n$.

The PF and PS at time t can be described as:

$$\begin{aligned} &\text{PS}(t) : 0 \leq x_r \leq 1, x_{i \neq r} = G(t), i = 1, \ldots, n \\ &\text{PF}(t) : f_2 = 1 - \sqrt{f_1}, 0 \leq f_1 \leq 1 \end{aligned}$$

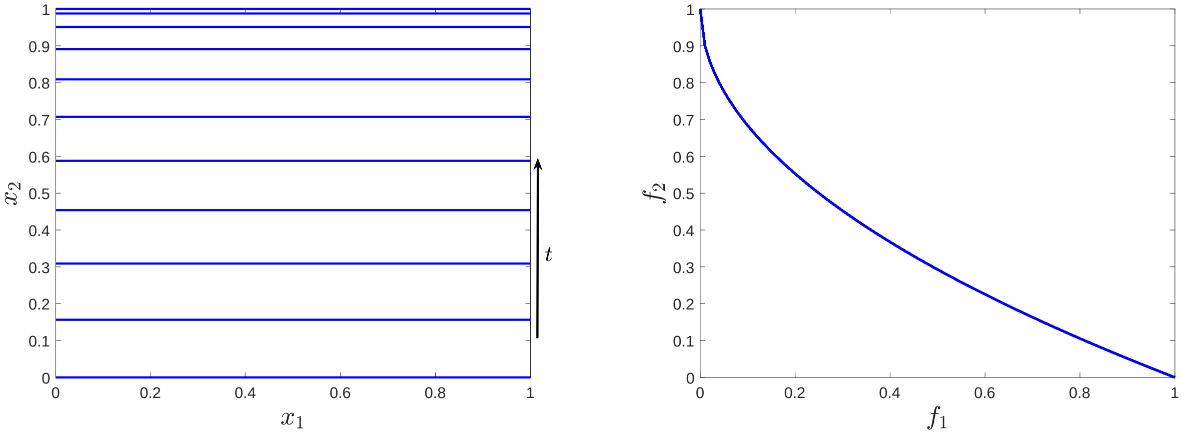

Figure 2: Illustration of the PS and PF of DF2.

Remark: DF2 has a simple dynamic on the PS, and its PF remains stationary over time. However, the switch of the position-related variable (x_r) is a challenging dynamic, as it can cause severe diversity loss to population. Hence, good diversity maintenance or increase techniques are required in order to solve this problem nicely.

3.3 DF3 (ZJZ [15])

$$min \begin{cases} f_1(x) = x_1 \\ f_2(x) = g(x)\left(1 - (\frac{x_1}{g(x)})^{H(t)}\right) \end{cases} \tag{3}$$

with

$$g(x) = 1 + \sum_{i=2}^{n} (x_i - G(t) - x_1^{H(t)})^2$$

where $G(t) = \sin(0.5\pi t)$, $H(t) = 1.5 + G(t)$. The search space is $[0,1] \times [-1,2]^{n-1}$.

The PF and PS at time t can be described as:

$$\text{PS}(t) : 0 \leq x_1 \leq 1, x_i = G(t) + x_1^{H(t)}, i = 2, \dots, n$$
$$\text{PF}(t) : f_2 = 1 - f_1^{H(t)}, 0 \leq f_1 \leq 1$$

Remark: The concavity-convexity of DF3 varies over time, and the variables are correlated. This problem is used to assess the tracking ability of concavity or convexity variations as well as time-varying variable linkages.

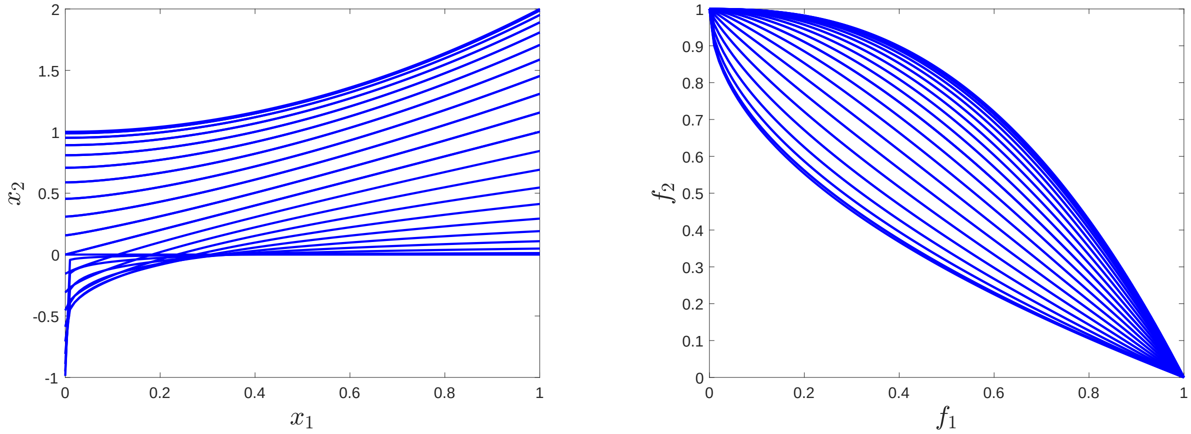

Figure 3: Illustration of the PS and PF of DF3.

3.4 DF4

$$min \begin{cases} f_1(x) = g(x)|x_1 - a|^{H(t)} \\ f_2(x) = g(x)|x_1 - a - b|^{H(t)} \end{cases} \tag{4}$$

with

$$g(x) = 1 + \sum_{i=2}^{n} \left(x_i - \frac{a x_1^2}{i} \right)^2$$

where $a = \sin(0.5\pi t), b = 1 + |\cos(0.5\pi t)|$, and $H(t) = 1.5 + a$. The search space is $[-2, 2]^n$.

The PF and PS at time t can be described as:

$$\text{PS}(t) : a \leq x_1 \leq a + b, x_i = \frac{a x_1^2}{i}, i = 2, \ldots, n$$

$$\text{PF}(t) : f_2 = (b - f_1^{\frac{1}{H(t)}})^{H(t)}, 0 \leq f_1 \leq b^{H(t)}$$

Remark: DF4 has dynamics on both the PF and PS. As can be seen from Figure 4, the length and position of the PS changes over time. The length and curvature of the PF segment is also time-varying.

3.5 DF5 (modified JY2 [8])

$$min \begin{cases} f_1(x) = g(x)(x_1 + 0.02 \sin(w_t \pi x_1)) \\ f_2(x) = g(x)(1 - x_1 + 0.02 \sin(w_t \pi x_1)) \end{cases} \tag{5}$$

with

$$g(x) = 1 + \sum_{i=2}^{n} (x_i - G(t))^2$$

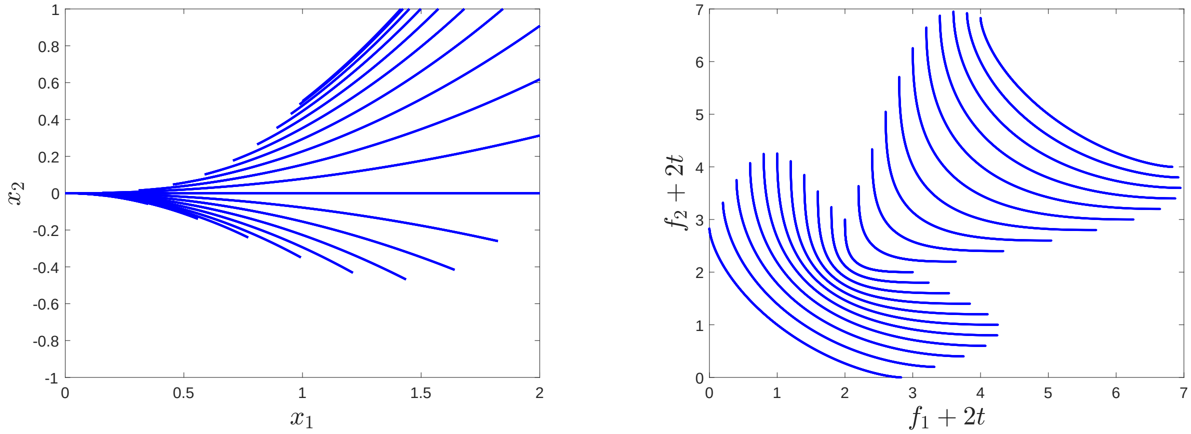

Figure 4: Illustration of the PS and PF of DF4.

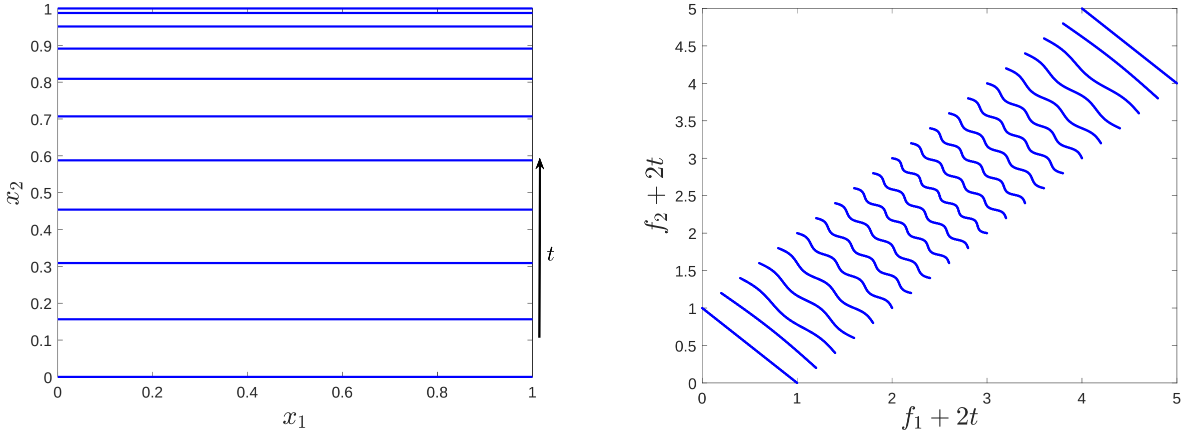

Figure 5: Illustration of the PS and PF of DF5.

where $G(t) = \sin(0.5\pi t)$, and $w_t = \lfloor 10G(t) \rfloor$. The search space is $[0, 1] \times [-1, 1]^{n-1}$.

The PF and PS at time t can be described as:

$$\begin{aligned} &\text{PS}(t) : 0 \leq x_1 \leq 1, x_i = G(t), i = 2, \ldots, n \\ &\text{PF}(t) : f_1 + f_2 = 1 + 0.04 \sin\left(w_t \pi \frac{f_1 - f_2 + 1}{2} \right), 0 \leq f_1 \leq 1 \end{aligned}$$

Remark: The PS of DF5 is rather simple, but the PF has time-varying geometries (see Figure 5). To be more specific, the PF is sometimes linear, and sometimes contains several locally concave/convex segments. Also, the number of local segments is time-dependent.

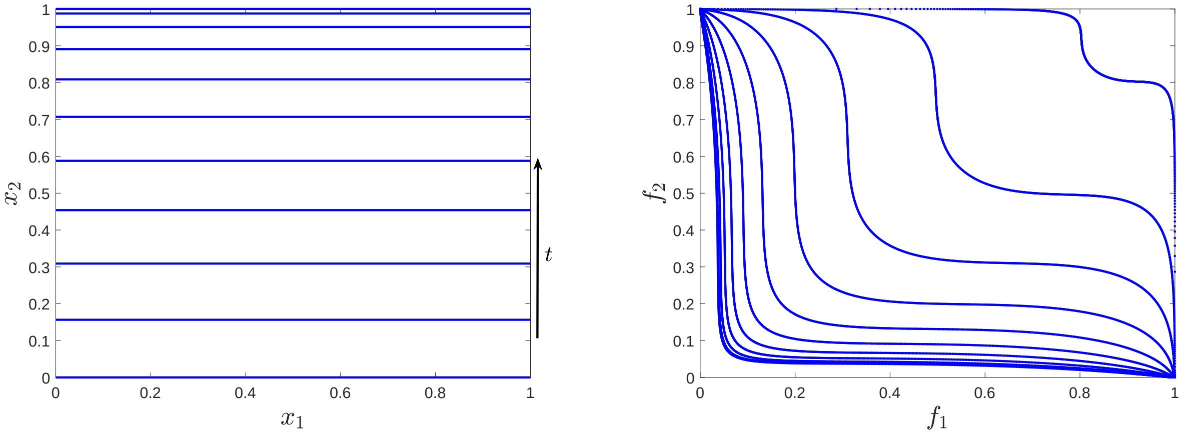

Figure 6: Illustration of the PS and PF of DF6.

3.6 DF6 (modified JY7 [8])

$$min \begin{cases} f_1(x) = g(x)(x_1 + 0.1\sin(3\pi x_1))^{\alpha_t} \\ f_2(x) = g(x)(1 - x_1 + 0.1\sin(3\pi x_1))^{\alpha_t} \end{cases} \tag{6}$$

with

$$g(x) = 1 + \sum_{i=2}^{n}(|G(t)|y_i^2 - 10\cos(2\pi y_i) + 10)$$

where $y_i = x_i - G(t), G(t) = \sin(0.5\pi t)$, and $\alpha_t = 0.2 + 2.8|G(t)|$. The search space is $[0,1] \times [-1,1]^{n-1}$.

The PF and PS at time t can be described as:

$$\text{PS}(t) : 0 \leq x_1 \leq 1, x_i = G(t), i = 2, \dots, n$$

$$\text{PF}(t) : f_1^{\frac{1}{\alpha_t}} + f_2^{\frac{1}{\alpha_t}} = 1 + 0.2\sin\left(3\pi\frac{f_1^{\frac{1}{\alpha_t}} - f_2^{\frac{1}{\alpha_t}} + 1}{2}\right), 0 \leq f_1 \leq 1$$

Remark: The PF geometry of DF6 is time-changing. Interestingly, the PF can have knee regions/points and long tails, which have already been recognised as a challenging property in some recent studies [9, 11]. This problem is used to see what the performance is when algorithms are confronted with a dynamic version of these properties.

3.7 DF7

$$min \begin{cases} f_1(x) = g(x)\frac{1+t}{x_1} \\ f_2(x) = g(x)\frac{x_1}{1+t} \end{cases} \tag{7}$$

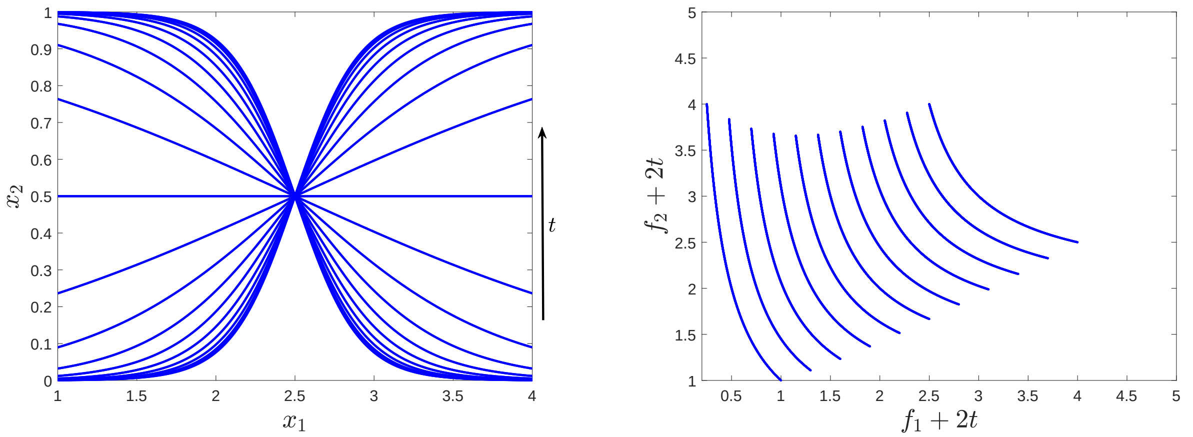

Figure 7: Illustration of the PS and PF of DF7.

with

$$g(x) = 1 + \sum_{i=2}^{n} \left(x_i - \frac{1}{1 + e^{\alpha_t (x_1 - 2.5)}} \right)^2$$

where $\alpha_t = 5\cos(0.5\pi t)$. The search space is $[1, 4] \times [0, 1]^{n-1}$.

The PF and PS at time t can be described as:

$$\begin{aligned} &\text{PS}(t) : 0 \leq x_1 \leq 1, x_i = \frac{1}{1 + e^{\alpha_t (x_1 - 0.5)}}, i = 2, \ldots, n \\ &\text{PF}(t) : f_2 = \frac{1}{f_1}, \frac{1+t}{4} \leq f_1 \leq (1+t) \end{aligned}$$

Remark: The PF range of DF7 is dissimilarly scaled and changes over time. The PS is dynamic, but its centroid remains unchanged. Such property can be difficult for centroid-based prediction methods [10, 14, 16, 18].

3.8 DF8

$$min \begin{cases} f_1(x) = g(x)(x_1 + 0.1\sin(3\pi x_1)) \\ f_2(x) = g(x)(1 - x_1 + 0.1\sin(3\pi x_1))^{\alpha_t} \end{cases} \tag{8}$$

with

$$g(x) = 1 + \sum_{i=2}^{n} \left(x_i - \frac{G(t)\sin(4\pi x_1^{\beta_t})}{1 + |G(t)|} \right)^2$$

where $\alpha_t = 2.25 + 2\cos(2\pi t)$, $\beta_t = 100G^2(t)$, and $G(t) = \sin(0.5\pi t)$. The search space is $[0, 1] \times [-1, 1]^{n-1}$.

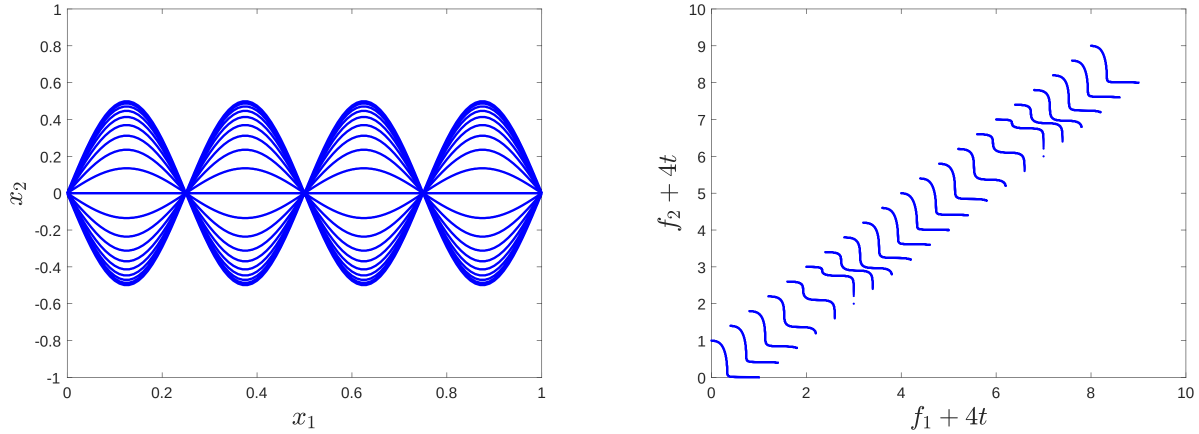

Figure 8: Illustration of the PS and PF of DF8.

The PF and PS at time t can be described as:

$$\text{PS}(t) : 0 \leq x_1 \leq 1, x_i = \frac{G(t)\sin(4\pi x_1^{\beta_t})}{1+|G(t)|}, i = 2, \dots, n$$

$$\text{PF}(t) : f_1 + f_2^{\frac{1}{\alpha_t}} = 1 + 0.2\sin\left(3\pi\frac{f_1 - f_2^{\frac{1}{\alpha_t}} + 1}{2}\right), 0 \leq f_1 \leq 1$$

Remark: DF8 has a stationary PS centroid, although the PS varies over time. The PS is harder to be approximated compared with that of DF7. The overall PF geometry switches between concavity and convexity, and contains knee regions.

3.9 DF9

$$min \begin{cases} f_1(x) = g(x)(x_1 + \max\{0, (\frac{1}{2N_t} + 0.1)\sin(2N_t\pi x_1)\}) \\ f_2(x) = g(x)(1 - x_1 + \max\{0, (\frac{1}{2N_t} + 0.1)\sin(2N_t\pi x_1)\}) \end{cases} \tag{9}$$

with

$$g(x) = 1 + \sum_{i=2}^{n}(x_i - \cos(4t + x_1 + x_{i-1}))^2$$

where $N_t = 1 + \lfloor 10|\sin(0.5\pi t)|\rfloor$. The search space is $[0, 1] \times [-1, 1]^{n-1}$.

The PF and PS at time t can be described as:

$$\text{PS}(t) : 0 \leq x_1 \leq 1, x_i = \cos(4t + x_1 + x_{i-1}), i = 2, \dots, n$$

$$\text{PF}(t) : f_2 = 1 - f_1, f_1 \in \bigcup_{i=1}^{N_t}[\frac{2i-1}{2N_t}, \frac{i}{N_t}]\bigcup\{0\}$$

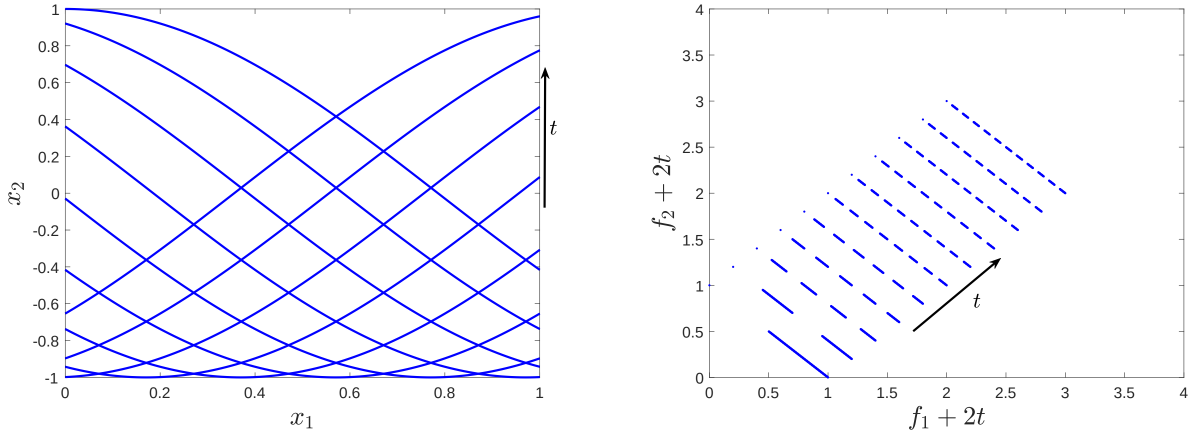

Figure 9: Illustration of the PS and PF of DF9.

Remark: DF9 has dependencies between variables. Its PF has a time-varying number of disconnected segments.

3.10 DF10

$$min \begin{cases} f_1(x) = g(x)[\sin(0.5\pi x_1)]^{H(t)} \\ f_2(x) = g(x)[\sin(0.5\pi x_2)\cos(0.5\pi x_1)]^{H(t)} \\ f_3(x) = g(x)[\cos(0.5\pi x_2)\cos(0.5\pi x_1)]^{H(t)} \end{cases} \tag{10}$$

with

$$g(x) = 1 + \sum_{i=3}^{n} \left(x_i - \frac{\sin(2\pi(x_1 + x_2))}{1 + |G(t)|} \right)^2$$

where $H(t) = 2.25 + 2\cos(0.5\pi t)$, and $G(t) = \sin(0.5\pi t)$. The search space is $[0, 1]^2 \times [-1, 1]^{n-2}$.

The PF and PS at time t can be described as:

$$\text{PS}(t) : 0 \leq x_{i=1,2} \leq 1, x_i = \frac{\sin(2\pi(x_1 + x_2)}{1 + |G(t)|}, i = 3, \ldots, n$$

$$\text{PF}(t) : \sum_{i=1}^{M} f_i^{\frac{2}{H(t)}} = 1, 0 \leq f_{i=1:M} \leq 1$$

Remark: DF10 has a stationary PS centroid in spite of the variation of the PS position. Its PF geometry changes from convexity to concavity, and vice versa. One of challenges that this problem poses to algorithm is how to maintain uniformity of solutions on the badly shaped PF at some time steps, i.e. t=0 and 2 (see Figure 10).

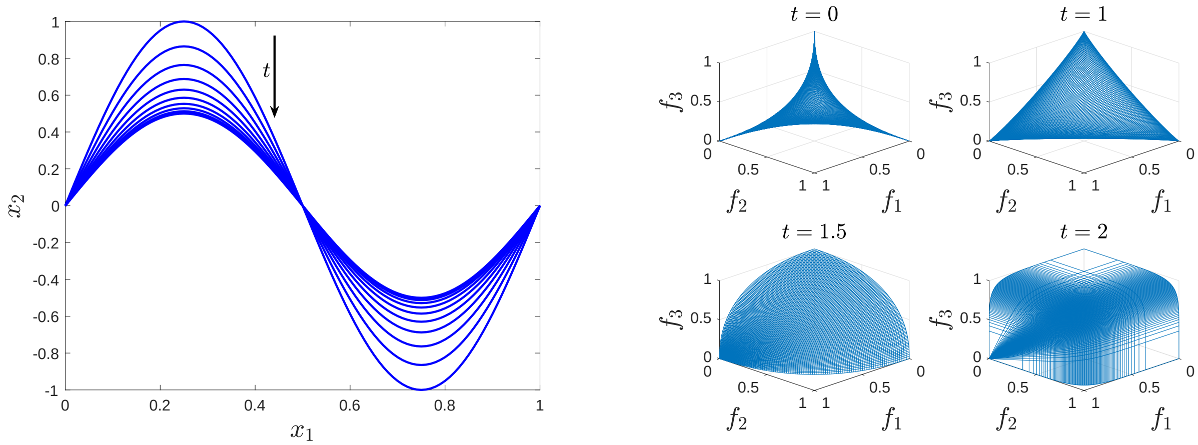

Figure 10: Illustration of the PS and PF of DF10.

3.11 DF11

$$min \begin{cases} f_1(x) = g(x)\sin(y_1) \\ f_2(x) = g(x)\sin(y_2)\cos(y_1) \\ f_3(x) = g(x)\cos(y_2)\cos(y_1) \end{cases} \tag{11}$$

with

$$y_{i=1:2} = \frac{\pi}{6}G_t + (\frac{\pi}{2} - \frac{\pi}{3}G_t)x_i,$$

and

$$g(x) = 1 + G(t) + \sum_{i=3}^{n}(x_i - 0.5G(t)x_1)^2$$

where $G(t) = |\sin(0.5\pi t)|$. The search space is $[0, 1]^n$.

The PF and PS at time t can be described as:

$$\text{PS}(t) : 0 \leq x_{i=1,2} \leq 1, x_i = 0.5G(t)x_1, i = 3, \ldots, n$$

$$\text{PF}(t) : \text{a part of } \sum_{i=1}^{M} f_i^2 = 1, 0 \leq f_{i=1:M} \leq 1$$

Remark: DF11 features the time-varying shrinkage/ expansion of the PF segment. Besides, the PF moves over time away from and close to the origin.

3.12 DF12

$$min \begin{cases} f_1(x) = g(x)\cos(0.5\pi x_1)\cos(0.5\pi x_2) \\ f_2(x) = g(x)\cos(0.5\pi x_1)\sin(0.5\pi x_2) \\ f_3(x) = g(x)\sin(0.5\pi x_1) \end{cases} \tag{12}$$

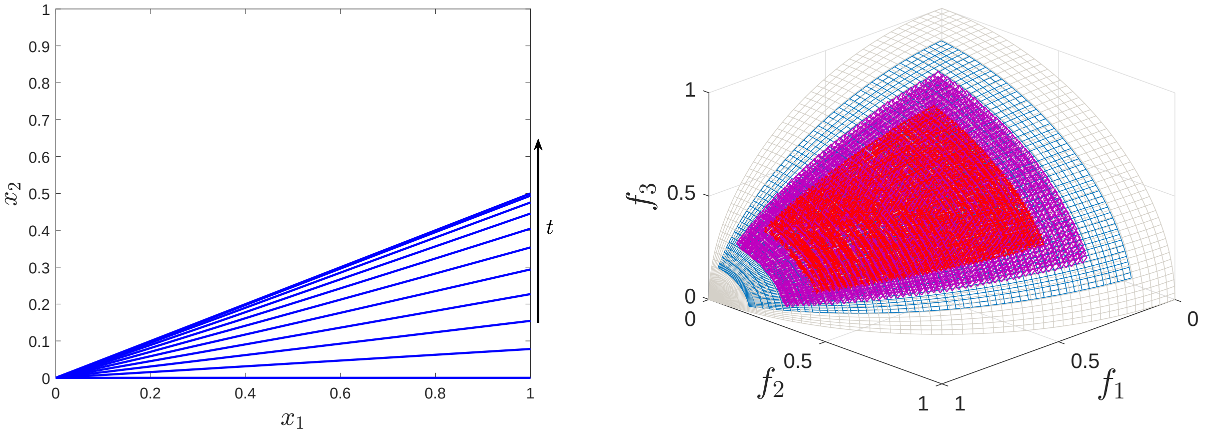

Figure 11: Illustration of the PS and PF of DF11.

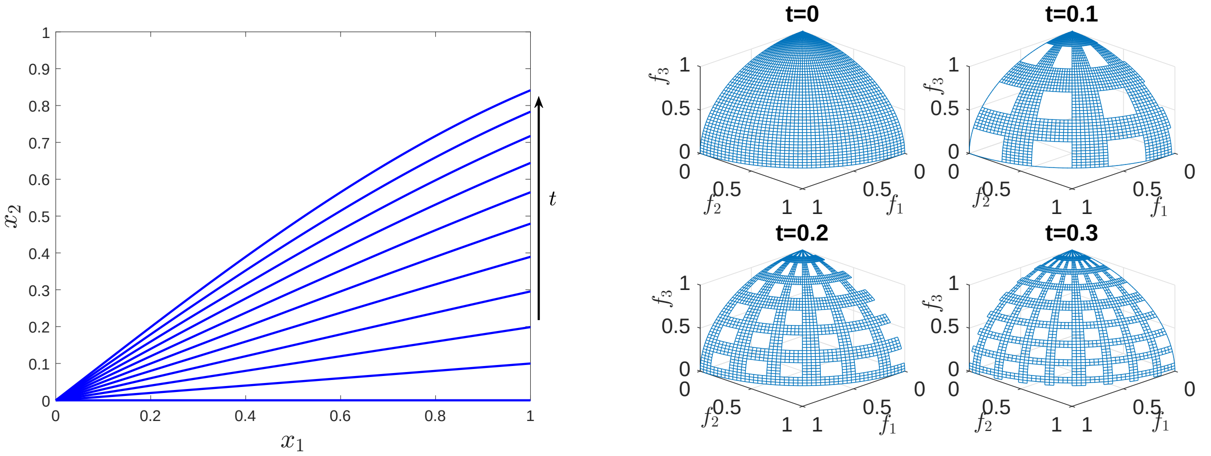

Figure 12: Illustration of the PS and PF of DF12.

with

$$g(x) = 1 + \sum_{i=3}^{n} (x_i - \sin(tx_1))^2 + \left|\prod_{j=1}^{2} \sin(\lfloor k_t(2x_j - 1)\rfloor \pi/2)\right|$$

where $k_t = 10\sin(\pi t)$. The search space is $[0,1]^2 \times [-1,1]^{n-2}$.

The PF and PS at time t can be described as:

$$\text{PS}(t) : 0 \leq x_{i=1,2} \leq 1, x_i = \sin(tx_1), i = 3, \ldots, n$$

$$\text{PF}(t) : \text{a part of } \sum_{i=1}^{M} f_i = 1, 0 \leq f_{i=1:M} \leq 1$$

Remark: DF12 has a time-varying number of PF holes, which might be difficult for decomposition-based algorithms that employ weight vectors [12]. This is because weight vectors are wasted if they happen to pass through the holes.

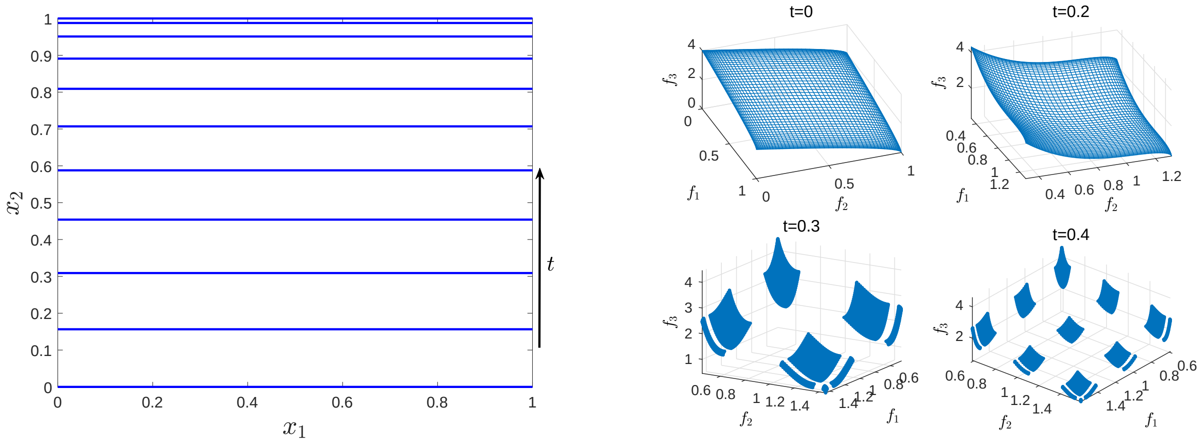

Figure 13: Illustration of the PS and PF of DF13.

3.13 DF13

$$min \begin{cases} f_1(x) = g(x)\cos^2(0.5\pi x_1) \\ f_2(x) = g(x)\cos^2(0.5\pi x_2) \\ f_3(x) = g(x)\sum_{j=1}^{2}\left[\sin^2(0.5\pi x_j) + \sin(0.5\pi x_j)\cos^2(p_t\pi x_j)\right] \end{cases} \tag{13}$$

with

$$g(x) = 1 + \sum_{i=3}^{n}(x_i - G(t))^2$$

where $p_t = \lfloor 6G_t \rfloor$, and $G(t) = \sin(0.5\pi t)$. The search space is $[0,1]^2 \times [-1,1]^{n-2}$.

The PF at time t can be continuous or disconnected as a result of environmental changes, and the PS is described as:

$$\mathrm{PS}(t) : 0 \leq x_1 \leq 1, x_i = G(t), i = 3, \ldots, n$$

Remark: DF13 generates both continuous and disconnected PF geometries. The number of disconnected PF segments varies over time. This problem is helpful for a better understanding of the impact of disconnectivity on algorithms.

3.14 DF14

$$min \begin{cases} f_1(x) = g(x)(1 - y_1 + 0.05\sin(6\pi y_1)) \\ f_2(x) = g(x)(1 - x_2 + 0.05\sin(6\pi x_2))(y_1 + 0.05\sin(6\pi x y_1)) \\ f_3(x) = g(x)(x_2 + 0.05\sin(6\pi x_2))(y_1 + 0.05\sin(6\pi y_1)) \end{cases} \tag{14}$$

with

$$y_1 = 0.5 + G(t)(x_1 - 0.5)$$

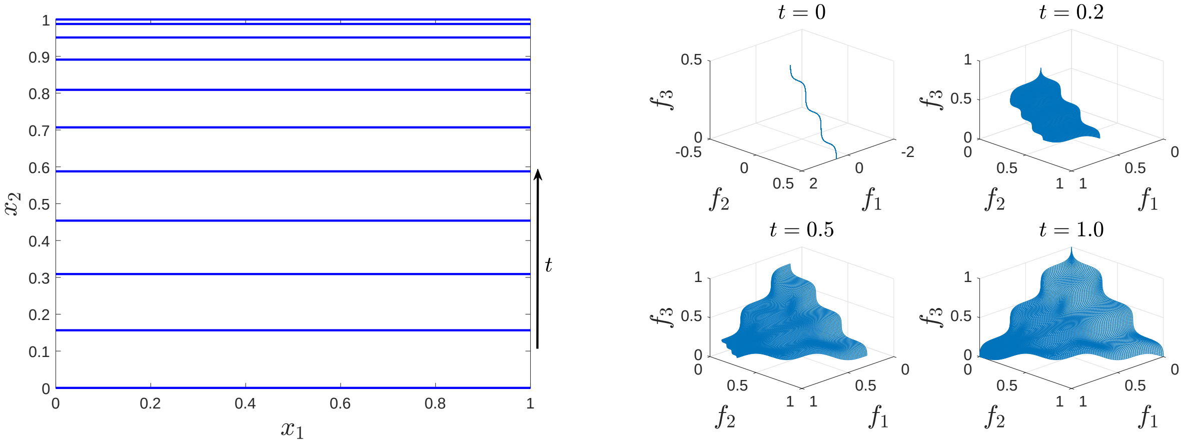

Figure 14: Illustration of the PS and PF of DF14.

and

$$g(x) = 1 + \sum_{i=3}^{n} (x_i - G(t))^2$$

where $G(t) = \sin(0.5\pi t)$. The search space is $[0, 1]^2 \times [-1, 1]^{n-2}$.

The PF at time t can be degenerate, and the PS is described as:

$$\text{PS}(t) : 0 \leq x_1 \leq 1, x_i = G(t), i = 3, \ldots, n$$

Remark: The dynamics that DF14 has is the changing size and dimension of the PF. The PF can be degenerated into an 1-D manifold. When the PF is not degenerate, the size of the 2-D PF manifold changes over time, and the number of knee regions changes accordingly.

4 Experimental Settings

The following experimental settings are encouraged to use when conducting empirical studies on the proposed test suite.

4.1 General Settings

- population size: 100 or a similar number for both 2 and 3 objectives.
- Number of variables: 10
- frequency of change (τ_t): 10 (fast changing environments), 30 (slow changing environments).
- severity of change (n_t): 10
- number of changes: 30

- stopping criterion: a maximum number of 100(30τ_t+50) fitness evaluations, where 500 fitness evaluations are given before the first environmental change occurs.
- Number of independent runs: 20

4.2 Performance Measures

MIGD [8] is adapted from IGD [13], a static performance indicator that measures both the convergence and diversity of solutions found by an algorithm. Let P_t be a set of uniformly distributed points in the true PF, and P_t^* be an approximation of the PF, at time t. The MIGD is calculated as follows:

$$MIGD = \frac{1}{T}\sum_{i=1}^{T} IGD(P_t^*, P_t) = \frac{1}{T}\sum_{i=1}^{T}\sum_{i=1}^{n_{P_t}} \frac{d_t^i}{n_{P_t}}, \tag{15}$$

where $n_{P_t} = |P_t|$, d_t^i is the Euclidean distance between the i-th member in P_t and its nearest member in P_t^*. A set of around 1000 points uniformly sampled from the true PF is expected to use for the calculation of MIGD.

4.2.1 Mean Hypervolume(MHV

The MHV [10] is a modification of the static measure HV [15] that computes the hypervolume of the area dominated by the obtained P_t^*:

$$MHV = \frac{1}{T}\sum\nolimits_{i=1}^{T} HV_t(P_t^*), \tag{16}$$

where $HV(S)$ is the hypervolume of a set S. The reference point for the computation of hypervolume is $(z_1 + 0.5, z_2 + 0.5, \cdots, z_M + 0.5)$, where z_j is the maximum value of the j-th objective of the true PF at time t and M is the number of objectives.

5 Result Submission

It is expected that competition results can be submitted in tables in a format exemplified in Table 2. However, other ways of result presentation are also acceptable. Please do make sure your result is of high readability for submission.

References

[1] S. Biswas, S. Das, P. N. Suganthan, and C. A. C. Coello, “Evolutionary multiobjective optimization in dynamic environments: A set of novel benchmark functions,” in *Proc. 2014 IEEE Congr. Evol. Comput.*, 2014, pp. 3192–3199.

[2] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scable test problems for evolutionary multiobjective optimization,” Kanpur Genetic Algorithms Lab (KanGAL), Indian Inst. Technol., KanGAL Rep. 2001001, 2001.

Table 2: **Your Algorithm**

Problem	τ_t	MIGD (mean(std.))	MHV(mean(std.))
DF1	10	1.1234E-2(2.1234E-3)	1.1234E-1(2.1234E-2)
	30	1.1234E-3(2.1234E-4)	1.1234E-2(2.1234E-3)
DF2	10		
	30		
DF3	10		
	30		
DF4	10		
	30		
DF5	10		
	30		
DF6	10		
	30		
DF7	10		
	30		
DF8	10		
	30		
DF9	10		
	30		
DF10	10		
	30		
DF11	10		
	30		
DF12	10		
	30		
DF13	10		
	30		
DF14	10		
	30		

[3] K. Deb, N. Rao U. B., and S. Karthik, "Dynamic multi-objective optimization and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling," in *Proc. 4th Int. Conf. Evol. Multi-criterion Optimization (EMO 2007)*, 2007, pp. 803–917.

[4] M. Farina, K. Deb, and P. Amato, "Dynamic multiobjective optimization problems: Test cases, approximations, and applications," *IEEE Trans. Evol. Comput.*, vol. 8, no. 5, pp. 425–442, 2004.

[5] S. B. Gee, K. C. Tan, and H. A. Abbass, "A benchmark test suite for dynamic evolutionary multiobjective optimization," *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 461–472, 2017.

[6] C. Goh and 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. Helbig and A. P. Engelbrecht, “Benchmarks for dynamic multi-objective optimisation algorithms,” *ACM Comput. Surv. (CSUR)*, vol. 46, no. 3, Article No. 37, 2014.

[8] S. Jiang and S. Yang, “Evolutionary dynamic multi-objective optimization: Benchmarks and algorithm comparisons,” *IEEE Trans. Cybern.*, vol. 47, no. 1, pp. 198–211, 2017.

[9] S. Jiang and S. Yang, “An improved multi-objective optimization evolutionary algorithm based on decomposition for complex Pareto fronts,” *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 421–437, 2016.

[10] S. Jiang and S. Yang, “A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization,” *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 65–82, 2017.

[11] T. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, “MOEA/D with adaptive weight adjustment,” *Evol. Comput.*, vol. 22, no. 2, pp. 231–264, 2014.

[12] Q. Zhang and H. Li, “MOEA/D: A multiobjective evolutionary algorithm based on decomposition,” *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, 2007.

[13] Q. Zhang, A. Zhou, and Y. Jin, “RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm,” *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 41–63, 2008.

[14] A. Zhou, Y. Jin, and Q. Zhang, “A population prediction strategy for evolutionary dynamic multiobjective optimization,” *IEEE Trans. Cybern.*, vol. 44, no. 1, pp. 40–53, 2014.

[15] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, “Prediction-based population reinitialization for evolutionary dynamic multi-objective optimization,” in *Proc. 4th Int. Conf. Evol. Multi-Criterion Optim. (EMO 2007)*, vol. 4403, 2007, pp. 832-846.

[16] P. Zhou, J. Zheng, J. Zou, M. Liu, “Novel prediction and memory strategies for dynamic multiobjective optimization,” *Soft Computing*, vol. 19, no. 9, pp. 2633–2653, 2015.

[17] E. Zitzler, K. Deb, and L. Thiele, “Comparison of multiobjective evolutionary algorithms: Empirical results,” *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.

[18] J. Zou, Q. Li, S. Yang, H. Bai, and J. Zheng, “A prediction strategy based on center points and knee points for evolutionary dynamic multi-objective optimization,” *Applied Soft Computing*, vol. 61, pp. 806–818, 2017.