2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Evolutionary Multiobjective Optimization in Dynamic Environments: A Set of Novel Benchmark Functions

Subhodip Biswas, Swagatam Das, Ponnuthurai N. Suganthan and Carlos A. Coello Coello

***Abstract*— Time varying nature of the constraints, objectives and parameters that characterize several practical optimization problems have led to the field of dynamic optimization with Evolutionary Algorithms. In recent past, very few researchers have concentrated their efforts on the study of Dynamic multi-objective Optimization Problems (DMOPs) where the dynamicity is attributed to multiple objectives of conflicting nature. Considering the lack of a somewhat diverse and challenging set of benchmark functions, in this article, we discuss some ways of designing DMOPs and propose some general techniques for introducing dynamicity in the Pareto Set and in the Pareto Front through shifting, shape variation, slope variation, phase variation, and several other types. We introduce 9 benchmark functions derived from the benchmark suite used for the 2009 IEEE Congress on Evolutionary Computation competition on bound-constrained and static MO optimization algorithms. Additionally a variant of multiobjective EA based on decomposition (MOEA/D) have been put forward and tested along with peer algorithms to evaluate the newly proposed benchmarks.**

I. Introduction

THE past few decades have witnessed an overwhelming growth in the field of Evolutionary Multiobjective Optimization (EMO) [1], [2]. Several efficient EAs have been proposed to overcome the difficulties in attaining the best compromise among two or more (conflicting) objectives that characterize a Multiobjective Optimization Problem (MOP). Most of such algorithms are, however, suitable for static MOPs, whose features do not change over time. In the recent past, benchmark suites of diverse levels of complexity were also developed for testing and ranking the real parameter multiobjective EAs that work under static conditions (e.g. [3], [4], [5]). However, despite the demands from several application domains, very few research works have so far been devoted to solving DMOPs by using evolutionary computing techniques or studying various difficulties associated with the DMOPs. In 2004, Farina *et al*. [5] took a bold step in this direction by suggesting a test-suite of five real-parameter dynamic MOPs. Although the authors in [5] urged EA-researchers to pursue the investigation of DMOPs to a large extent, such problems are yet to receive significant attention so as to match the attention paid to the fields of dynamic single-objective optimization problems [6] and static MOPs.

Dynamicity in a MOP can occur due to time variation of the search-variables, objective function landscapes, and/or the constraints involved [7]. As the main intricacies of MOPs lie in the Pareto-Front (PF) and Pareto-Set (PS) relationships, dynamicity can be imparted to the problem by applying time variation to them. The rest of this paper has been organized in the following way: in Section II we mention related works which have motivated us to perform this research. Section III presents the general idea and basic concepts of the dynamic multiobjective benchmark generator. Section IV proposes 9 new benchmark functions evolving from the bound-constrained multiobjective test bed developed by Zhang *et al*. [4] for the CEC 2009 competition on MOEAs. In Section V, we present the statistical analysis of the proposed as well as contemporary MOEAs on these benchmark functions. Finally, Section VI concludes the paper.

II. Related Works and Motivations

The earliest work in this direction started with the development of some benchmark problems proposed by Jin and Sendhoff [8], who aggregated different objectives of the existing stationary benchmark functions and varied the weights dynamically. Farina *et al*. [5] contributed to the dynamic multiobjective benchmarks by considering time variation of the objective functions through dynamicity in PF and PS. Their benchmarks were based on existing ZDT [9] and DTLZ [3] type of static MOPs. Mehnen *et al*. also presented a set of test functions in [10]. But the problems in this set are not sufficiently diverse, scalable, and challenging.

This article presents a more generalized benchmark generator considering different kinds of changes in the PF and the PS that have not been presented earlier in context of DMOPs. The benchmark generator is extended to include the bound-constrained multiobjective test-bed developed by Zhang *et al*. [4] for the competition of Multiobjective Evolutionary Algorithms (MOEAs) held at the 2009 IEEE Congress on Evolutionary Computation (CEC).

Note that the UFs (Unconstrained Functions as termed in [4]) that we use here are more exhaustive and diverse in terms of the nature of the PF and the PS than the ZDT and DTLZ series, which were used by Farina *et al*. [5]. Some UFs have a discrete PF, and so do some of our dynamic problems like UDF3 and UDF6 (UDF: Unconstrained Dynamic Functions and these are discussed in Section IV). For the ZDT and DTLZ test problems, the global optimum has the same parameter values for different variables/dimensions. But in

S. Biswas is with the Department of Electronics and Tele-communication Engineering, Jadavpur University, Kolkata, WB 700032 INDIA (e-mail: sub17was@gmail.com).

S. Das is with the Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata, WB 700108 INDIA (e-mail: swagatam.das@isical.ac.in).

P. N. Suganthan is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore (e-mail: epnsugan@ntu.edu.sg).

C. A. Coello Coello is with the Evolutionary Computation Group of the Department of Computer Science, CINVESTAV-IPN, Mexico. (e-mail: ccoello@cs.cinvestav.mx).

the UF problems, different dimensions are treated differently. We observed that the PS in various UFs is of two types in general– sinusoidal and polynomial. We include both in our test suite. In addition, Farina *et al*. [5] introduced only two types of changes: shift and shape (curvature) change of the PF, and shift of the PS along with the change in radius for a spherical PF. Besides retaining such changes, we incorporate dynamicity through a few other interesting changes like angular shift or slope change of the PF, shape change of both a polynomial PS (by changing the order) and a trigonometric PS, change of curvature of a spherical PF resulting in different ellipsoids, etc.

We apply various combinations of such changes to static MO problems that are themselves different, to keep the test-bed more diverse, exhaustive, and challenging.

III. Dynamic MOPs Definition & Formulations

A dynamic MOP with m objective functions can be formulated as

$$\begin{cases} \min F(\vec{x},t) = \{f_1(\vec{x},t), f_2(\vec{x},t), ..., f_m(\vec{x},t)\}, \\ \text{subject to}: g(\vec{x},t) \leq 0, h(\vec{x},t) = 0, \end{cases} \qquad (1)$$

where Ω is the decision space, $F : \Omega \rightarrow R^m$ consists of m real-valued objective functions and R^m is the objective space. If $\vec{x} \in R^n$, all the objectives are continuous and Ω is described by: $\Omega = \{\vec{x} \in R^n | l_j(\vec{x}) \leq 0, j = 1, ..., k\}$, l_js are continuous functions.

Definition 1 (Dominance): Let $\vec{x}_i, \vec{v} \in R^m$, $\vec{u}$ is said to dominate $\vec{v}$ if and only if $u_i \leq v_i$ for every $i \in 1, ..., m$ and $u_j < v_j$ for at least one index $j \in \{1, 2, ...m\}$.

Definition 2 (Pareto optimality): A solution $\vec{x}^* \in \Omega$ is Pareto optimal if there is no $\vec{x} \in \Omega$ such that $F(\vec{x})$ dominates $F(\vec{x}^*)$. $F(\vec{x}^*)$ is then called a Pareto optimal front. In other words, any improvement in a Pareto optimal solution in one objective leads to a deterioration of at least one other objective. The set of all the Pareto optimal solutions is called *Pareto Optimal Set* or Pareto Set (PS) and the set of all the Pareto optimal vectors is the *Pareto Optimal Front* or Pareto Front (PF).

Unlike the dynamic single-objective problems, where dynamicity with time is brought about by controlling the width, height and position of the peaks of the problem, for MOPs, variations in the PF and the PS are the keys to maintain dynamicity of the problem. As argued by Farina *et al*. [5], dynamicity with time in any MOP can occur through the following criteria:

i) The PS changes with time but the PF does not change.

ii) The PF changes with time but the PS does not change.

iii) Both the PF and the PS change with time.

iv) Both the PF and the PS remain unchanged with time but other changes in the problem definition induce dynamicity.

In this paper, we mainly deal with the first three types of changes indicated above. Following the basic philosophy of the dynamic test problem construction, we introduce several kinds of changes possible on the PF and the PS. From the benchmark functions proposed for the CEC 2009 competition on MOEAs [4], we chose certain representative problems and extended them to dynamic multiobjective optimization. To discuss the constructions of dynamic test problems by varying different aspects of the PF and the PS, we will consider two sample functions and elaborate our discussions around them.

Sample 1:

$$\begin{aligned} f_1 &= x_1 + \tfrac{2}{|J_1|} \sum_{j \in J_1} [x_j - \sin(6\pi x_1 + j\tfrac{\pi}{n})]^2, \\ f_2 &= 1 - \sqrt{x_1} + \tfrac{2}{|J_2|} \sum_{j \in J_2} [x_j - \sin(6\pi x_1 + j\tfrac{\pi}{n})]^2, \end{aligned} \qquad (2)$$

where $J_1 = \{j | j$ is odd and $2 \leq j \leq n\}$, $J_2 = \{j | j$ is even and $2 \leq j \leq n\}$. Search space: $[0,1] \times [-1,1]^{n-1}$. Its optimal PF is $f_2 = 1 - \sqrt{f_1}$, and the corresponding PS is $x_j = \sin(6\pi x_1 + j\frac{\pi}{n}), j = 2, ..., n; 0 \leq x_1 \leq 1$, where n is the dimensionality of the search space.

Sample 2:

$$\begin{aligned} f_1 &= x_1 + \tfrac{2}{|J_1|} \sum_{j \in J_1} [x_j - x_1^{0.5(1+3\frac{j-2}{n-2})}]^2, \\ f_2 &= 1 - x_1^2 + \tfrac{2}{|J_2|} \sum_{j \in J_2} [x_j - x_1^{0.5(1+3\frac{j-2}{n-2})}]^2, \end{aligned} \qquad (3)$$

where $J_1 = \{j | j$ is odd and $2 \leq j \leq n\}$ and $J_2 = \{j | j$ is even and $2 \leq j \leq n\}$. The search space is $[0,1] \times [-1,1]^{n-1}$. Its optimal PF is $f_2 = 1 - f_1^2$, and the corresponding PS is $x_j = x_1^{0.5(1+3\frac{j-2}{n-2})}; j = 2, ..., n; 0 \leq x_1 \leq 1$, and as before n is the dimension of the search space. Sample 2 is constructed by the combination of different shapes of PF and PS as described in [4].

We can see that the basic difference between the two sample functions taken are that their PS variations are trigonometric and polynomial, respectively. The PF shape is taken concave and convex in two samples. Now we describe the possible changes that can be applied to them to bring in the dynamicity. These concepts will be later utilized for the benchmark generator proposed.

In general, for testing purposes, the basic time varying function that will be incorporated within the MOP is kept sinusoidal in nature to restrict the variation within a limit and to maintain periodicity in the function. The function is $G(t) = \sin(0.5\pi t), t = \lfloor \tau / T \rfloor / n_s$, where τ is the generation counter, n_s is the number of distinct steps and represents the severity of change, T is the window where the dynamic problem remains constant. Other types of variations, which are periodic in nature, can always be integrated along with the benchmarks that we develop.

A. Pareto Set Variation

Sample 1. Trigonometric PS

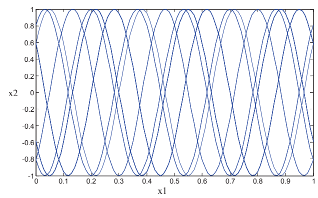
The decision variables $x_j, \forall j = 2, ..., n$ are independent among them and are only dependent on x_1 with sinusoidal variation. There are two possible ways of bringing dynamicity to the PS:

(a) Horizontal shift or Phase change of the x_j for different j with time variation occurs as per $x_j = \sin(6\pi x_1 + (j + K(t))\pi/n)$, $j = 2, ..., n$; $0 \leq x_1 \leq 1$, $K(t) = \lceil nG(t) \rceil$. Search space: $[0,1] \times [-1,1]^{n-1}$. The PS shape for this case is shown in Figure 1 (a).

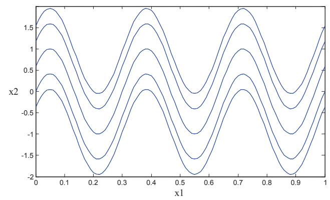
(b) Vertical shift of the x_j for different j with time variation occurs as per $x_j = \sin(6\pi x_1 + \frac{\pi}{n} j) + G(t)$,

$j = 2, ..., n;\ 0 \le x_1 \le 1$. Search space: $[0, 1] \times [-2, 2]^{n-1}$. The PS shape for this case is shown in Figure 1(b).

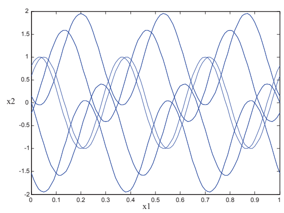
(c) Combining both horizontal and vertical shifts we can have a more challenging variation as shown in Figure 1(c). $x_j = \sin(6\pi x_1 + \frac{\pi}{n}(j + K(t))) + G(t),\ j = 2, ..., n;$ $0 \le x_1 \le 1$ and $K(t) = \lceil nG(t) \rceil$. Search space: $[0, 1] \times [-2, 2]^{n-1}$.



(a) Horizontal Shift of the PS in 2D



(b) Vertical Shift of the PS in 2D



(c) Combined Shift of the PS

Fig. 1: Pareto Set shape obtained through trigonometric variation

Sample 2: Polynomial PS

The decision variables $x_j, \forall j = 2, ..., n$ are independent among them and are only dependent on x_1 with polynomial variation of different orders. There are two possible ways of bringing dynamicity to the Pareto set:

(a) Polynomial order change of the x_j for different j with time variation occuring as per $x_j = x_1^{0.5(2+G(t)+3\frac{j-2}{n-2})}, j = 2, ..., n;\ 0 \le x_1 \le 1$. Search space: $[0, 1]^n$.

(b) Vertical shift of the x_j for different j with time variation occurs by $x_j = x_1^{0.5(1+3\frac{j-2}{n-2})} + G(t),\ j = 2, ..., n;$ $0 \le x_1 \le 1$. Search space: $[0, 1] \times [-1, 2]^{n-1}$.

(c) Combining both shifts we can have a more challenging version as $x_j = x_1^{0.5(2+G(t)+3\frac{j-2}{n-2})} + G(t),\ j = 2, ..., n;\ 0 \le x_1 \le 1$. Search space: $[0, 1] \times [-2, 2]^{n-1}$. The corresponding variations of the PS have been depicted in Figure 2.

In the above examples we present different ways of changing the PS. These are the general approaches and can be integrated with any type of PS. The time varying functional adjustment for vertical shift, horizontal shift and order change can be made in various other ways, too.

B. Pareto Front Variation

The above two samples have both convex and concave Pareto sets, respectively. Let us consider a general PF equation of order N : $f_2 = 1 - f_1^N$, which may be convex, concave, or straight and $0 \le f_1, f_2 \le 1$. Changes can be made in the PF in the following ways.

i) *Change in curvature or shape of the PF*: The order N plays an important role in deciding the nature of the curvature of the front in the following way: Convex for $N > 1$; Concave for $N < 1$; Straight for $N = 1$.

Here we vary the order of N with time to change the curvature. The change can be done in any desired way. Let us consider a case where we need to vary N between N_1 and N_2. The value of N_1 can be taken to be in the range of $1/5$ to $1/2$ and N_2 in the range of 1 to 5. The functional relationship can be designed as $H(t) = N_1 + (N_2 - N_1)|G(t)|$ and the final front equation as $f_2 = 1 - f_1^{H(t)}$. In Figure 3 we show a sample variation for $N_1 = 0.5$ and $N_2 = 1.5$.

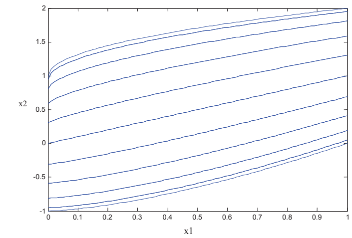


Fig. 2: Combined change of (a) and (b) in PS.

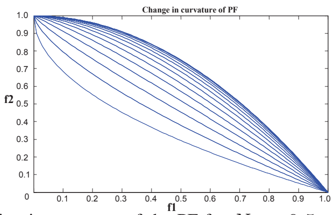


Fig. 3: Variation in curvature of the PF for $N_1 = 0.5$ and $N_2 = 1.5$.

ii) *Shift in the PF*: Another way of dealing with the change is that the PF shifts in different possible directions with time. The possibilities of such change are given below:

(a) Vertical shift: $f_2 = 1 - f_1^N + |G(t)|$ gives the vertical shift as shown in Figure 4(a).

(b) Horizontal shift: $f_2 = 1 - (f_1 - |G(t)|)^N$ gives the horizontal shift as shown in Figure 4(b).

(c) Diagonal shift: $f_2 = 1 - (f_1 - |G(t)|)^N + |G(t)|$ gives the diagonal shift as shown in Figure 4(c).

(d) Angular shift or rotation of the curvature of the PF: $f_2 = 1 - M(t) f_1^N$, where $M(t)$ controls the angular shift of the curvature as shown in Figure 4(d) and given as: $M(t) = M_1 + (M_2 - M_1)|G(t)|$, M_1 and M_2 being the extremities of the slope. The value M_1 can be taken in the range of $1/5$ to $1/2$ and M_2 in the range of 1 to 5.

In Figure 4(e), we present the combined shift effects due to diagonal and angular shift whose general equation is given as: $f_2 = 1 - M(t)(f_1 - |G(t)|)^N + |G(t)|$. Combining both the shifting change and shape change we form an ultimate dynamic PF presented in Figure 4(f) and given by $f_2 = 1 - M(t)(f_1 - |G(t)|)^{H(t)} + |G(t)|$. Higher complexity can be added if both vertical shifts and horizontal shifts are different with respect to time and can be given as: $f_2 = 1 - M(t)(f_1 - |G_1(t)|)^{H(t)} + |G_2(t)|$.

iii) The next type of possible changes can be made on the general DTLZ [3] type of functions where the PF equation remains as $f_1^2 + f_2^2 + ... + f_M^2 = 1$, where M is the number of objectives and the test problem is scalable. The possible changes that can be applied to these general types of problem are described below:

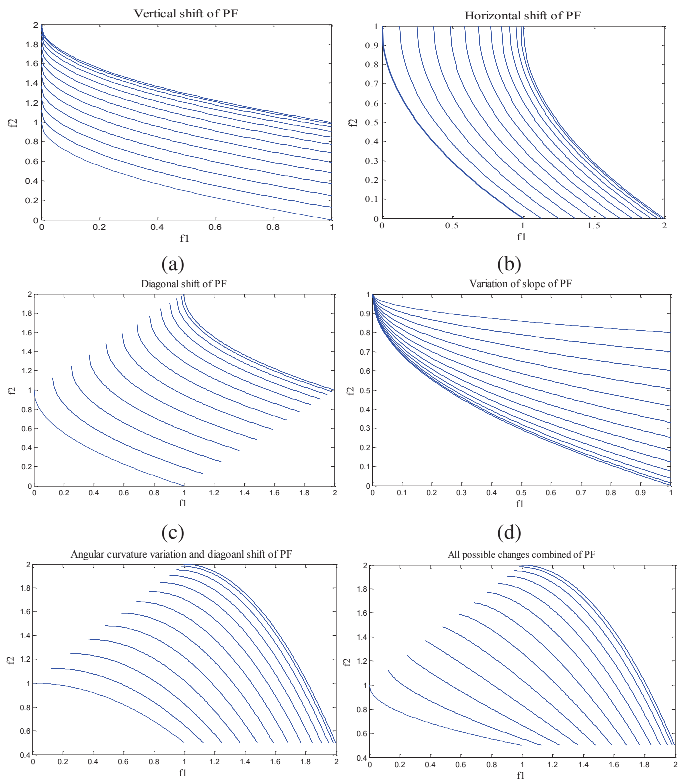
(a) Change in the radius: The PF is practically spherical in shape and the radius is unity for the constant case which can be varied with time as shown in Figure 5(a):

$$f_1^2 + f_2^2 + ... + f_M^2 = R(t)^2, R(t) = |1 + |G(t)||.$$

(b) Change in curvature: Consider the PF equation below

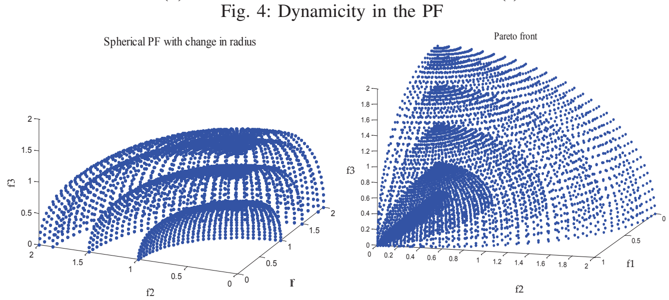
$$\frac{f_1^2}{R(t)^2}+\frac{f_2^2}{R(t)^2}+...+\frac{f_{M-1}^2}{R(t)^2}+f_M^2=1, \quad R(t)=|1+|G(t)||.$$
$$\text{or, } \frac{f_1^2}{R(t)^2}+f_2^2+...+f_M^2=1, \quad R(t)=|1+|G(t)||.$$

Any other combinations, where any number of objective functions among $\{f_1, f_2, ..., f_M\}$, but not all, are time varying, leads to a dynamic curvature of the PF, the PFs being ellipsoidal in general. The changes have been depicted in Figure 5 (b). UF12 [4] is one such test instance formulated using DTLZ type functions with 3 objectives.



(a) (b) (c) (d) (e) (f)

Fig. 4: Dynamicity in the PF



(a) Radius variation (b) Curvature variation

Fig. 5: Spherical PFs (3D) with radius and curvature variation

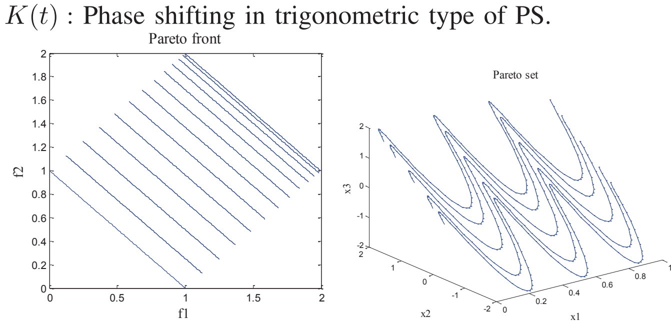
Any combination of the above changes can be applied to bring dynamicity in the problem. Certain types of changes may occur simultaneously at every time instant to produce a deterministic function or each may occur randomly at any instant generating a somewhat nondeterministic function of random nature. Test functions have been discussed in Section IV to elaborate those concepts. Besides the 9 test functions presented here, many more can be generated from the dynamic function generator concept discussed so far by considering several combinations of changes.

IV. Benchmark Functions

From the above general notion of the types of changes that can be applied to a multiobjective function to produce its dynamic nature we are now attempting to formulate a set of diversified benchmark functions. Below we describe each benchmark function mathematically and discuss the unique properties associated with it. As before, we assume $G(t)=\sin(0.5\pi t), t=\lfloor \tau/T \rfloor/n_s, M(t)=0.5+|G(t)|, H(t)=0.5+|G(t)|, R(t)=1+|G(t)|$ and $K(t)=\lceil nG(t) \rceil$.

N.B.: In all the benchmark functions we have used some common variables. They denote some typical kind of changes in PS or PF as discussed below:

$G(t)$: Vertical or Horizontal shift in PF or PS.
$M(t)$: Angular shift in PF.
$H(t)$: Curvature variation in PF.
$R(t)$: Radius variation in three dimensional PF.
$K(t)$: Phase shifting in trigonometric type of PS.



(a) (b)

Fig. 6: PS and PF for UDF1.

A. UDF 1: The nature of the PS is trigonometric and we have incorporated a vertical shifting with time in the PS. The PF is linear and continuous in nature and associated with diagonal shifting with time.

$$f_1=x_1+\frac{2}{|J_1|}\sum_{j\in J_1}[x_j-\sin(6\pi x_1+j\tfrac{\pi}{n})-G(t)]^2+|G(t)|$$
$$f_2=1-x_1+|G(t)|+\frac{2}{|J_2|}\sum_{j\in J_2}[x_j-\sin(6\pi x_1+j\tfrac{\pi}{n})-G(t)]^2 \tag{4}$$

where $J_1=\{j|j$ is odd and $2\le j\le n\}$ and $J_2=\{j|j$ is even and $2\le j\le n\}$.

Search space: $[0,1]\times[-2,2]^{n-1}$, where n is the number of dimensions of the search space.

Optimal PF: $f_2=1-(f_1-|G(t)|)+|G(t)|\,; 0+|G(t)|\le f_1\le 1+|G(t)|$.

Optimal PS: $x_j=\sin(6\pi x_1+j\frac{\pi}{n})+G(t)$; $j=2,...,n$; $0\le x_1\le 1$.

B. UDF2: The PS is polynomial in nature and associated with a vertical shifting and shift in the order of the polynomial function with time. The PF is linear and continuous in nature associated with vertical shifting with time.

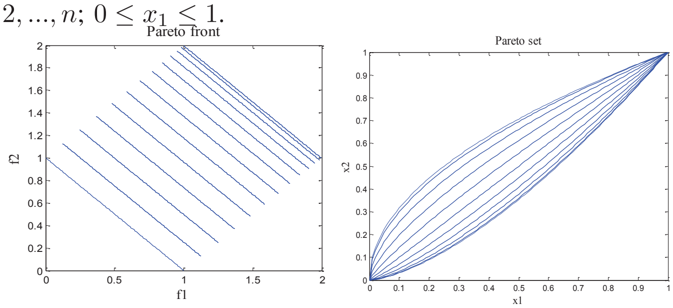
$$f_1=x_1+\frac{2}{|J_1|}\sum_{j\in J_1}[x_j-{x_1}^{0.5(2+\frac{3(j-2)}{n-2}+G(t))}-G(t)]^2+|G(t)|$$
$$f_2=1-x_1+\frac{2}{|J_2|}\sum_{j\in J_2}[x_1{}^{0.5(2+\frac{3(j-2)}{n-2}+G(t))}-G(t)]^2+|G(t)|, \tag{5}$$

where $J_1 = \{j|j$ is odd and $2 \le j \le n\}$ and $J_2 = \{j|j$ is even and $2 \le j \le n\}$.

Search space: $[0,1] \times [-1,2]^{n-1}$, where n is the number of dimensions of the search space.

Optimal PF: $f_2 = 1 - (f_1 - |G(t)|) + |G(t)|, 0 + |G(t)| \le f_1 \le 1 + |G(t)|$.

Optimal PS: $x_j = x_1^{0.5(2+\frac{3(j-2)}{n-2}+G(t))} + G(t);\ j = 2, ..., n;\ 0 \le x_1 \le 1$.



(a) Fig. 7: PS and PF for UDF2. (b)

C. UDF3: The PS is trigonometric but more challenging than UDF1 and UDF2. There is no time variation in PS. The Pareto front is linear but discontinuous. The PF is associated with diagonal shifting. Due to the discontinuous nature of the Pareto front, the problem itself is little harder.

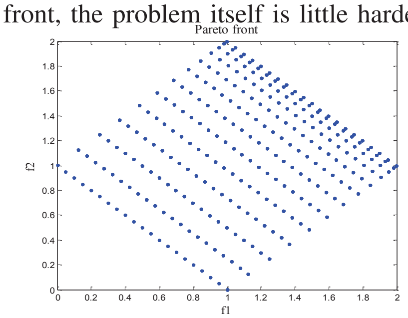


Fig. 8: PF for UDF3.

$$\begin{aligned} f_1 &= x_1 + \max\{0, (\tfrac{1}{2N} + \varepsilon)[\sin(2N\pi x_1) - 2N\,|G(t)|]\} \\ &+ \tfrac{2}{|J_1|}[4\sum_{j\in J_1} 2y_j^2 - 2\prod_{j\in J_1}\cos(\tfrac{20\pi y_j}{\sqrt{j}}) + 2]^2, \\ f_2 &= 1 - x_1 + \tfrac{2}{|J_2|}[4\sum_{j\in J_2} 2y_j^2 - 2\prod_{j\in J_2}\cos(\tfrac{20\pi y_j}{\sqrt{j}}) + 2]^2 \\ &+ \max\{0, (\tfrac{1}{2N} + \varepsilon)[\sin(2N\pi x_1) - 2N\,|G(t)|]\}, \end{aligned} \tag{6}$$

where $y_j = x_j - \sin(6\pi x_1 + j\frac{\pi}{n});\ j = 2, ..., n,\ J_1 = \{j|j$ is odd and $2 \le j \le n\}$, $J_2 = \{j|j$ is even and $2 \le j \le n\}$, $\varepsilon = 0.1$ and $N = 10$.

Search space: $[0,1] \times [-1,1]^{n-1}$, where n is the number of dimensions of the search space.

Optimal PF: One disconnected point $(0,1)$. N disconnected parts: $\bigcup_{i=1}^{N} [\frac{2i-1}{2N} + |G(t)|, \frac{2i}{2N} + |G(t)|]$.

Optimal PS: $x_j = \sin(6\pi x_1 + j\frac{\pi}{n}); j = 2, ..., n; 0 \le x_1 \le 1$.

D. UDF4: The PS is simple trigonometric and horizontal shifting with time is applied to it. The PF is continuous and associated with curvature change from convex to concave and angular shift with time.

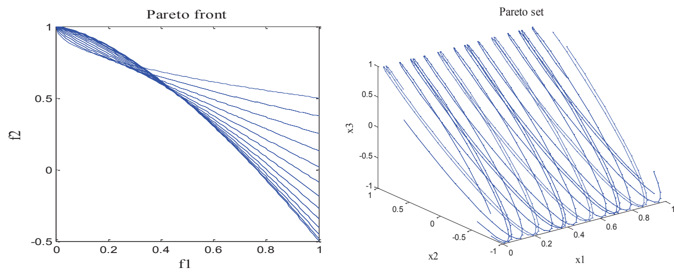
$$\begin{aligned} f_1 &= x_1 + \tfrac{2}{|J_1|}\sum_{j\in J_1}[x_j - \sin(6\pi x_1 + (j + K(t))\tfrac{\pi}{n})]^2, \\ f_2 &= 1 - M(t)(x_1)^{H(t)} + \\ &\tfrac{2}{|J_2|}\sum_{j\in J_2}[x_j - \sin(6\pi x_1 + (j + K(t))\tfrac{\pi}{n})]^2, \end{aligned} \tag{7}$$

where $J_1 = \{j|j$ is odd and $2 \le j \le n\}$ and $J_2 = \{j|j$ is even and $2 \le j \le n\}$.

Search space: $[0,1] \times [-1,1]^{n-1}$, where n is the number of dimensions of the search space.

Optimal PF: $f_2 = 1 - M(t){f_1}^{H(t)};\ 0 \le f_1 \le 1$.

Optimal PS: $x_j = \sin(6\pi x_1 + (j + K(t))\frac{\pi}{n}),\ j = 2, .., n;\ 0 \le x_1 \le 1$.



(a) Fig. 9: PS and PF for UDF4. (b)

E. UDF5: The PS is simple polynomial and associated with change in the order of the polynomial and vertical shifting. The PF is continuous and associated with curvature change from convex to concave and angular shift with time.

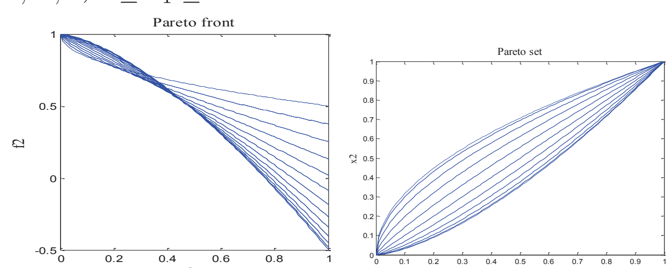
$$\begin{aligned} f_1 &= x_1 + \tfrac{2}{|J_1|}\sum_{j\in J_1}[x_j - x_1^{0.5(2+\frac{3(j-2)}{n-2}+G(t))} - G(t)]^2, \\ f_2 &= 1 - M(t){x_1}^{H(t)} + \tfrac{2}{|J_2|}\sum_{j\in J_2}[x_1^{0.5(2+\frac{3(j-2)}{n-2}+G(t))} - G(t)]^2, \end{aligned} \tag{8}$$

where $J_1 = \{j|j$ is odd and $2 \le j \le n\}$ and $J_2 = \{j|j$ is even and $2 \le j \le n\}$.

Search space: $[0,1] \times [-1,2]^{n-1}$, where n is the number of dimensions of the search space.

Optimal PF: $f_2 = 1 - M(t){f_1}^{H(t)};\ 0 \le f_1 \le 1$.

Optimal PS: $x_j = x_1^{0.5(2+\frac{3(j-2)}{n-2}+G(t))} + G(t),\ j = 2, ..., n;\ 0 \le x_1 \le 1$.



(a) Fig. 10: PS and PF for UDF5. (b)

F. UDF6: The PS is trigonometric but more challenging than UDF1 and UDF2. There is no time variation in PS. The PF is linear but discontinuous. The PF is associated with diagonal shifting and angular shifting. Due to the discontinuity of the PF, the problem itself is little harder.

$$\begin{aligned} f_1 &= x_1 + (\tfrac{1}{2N} + \varepsilon)\,|\sin(2N\pi x_1) - 2N\,|G(t)|| \\ &+ \tfrac{2}{|J_1|}\sum_{j\in J_1}[2y_j^2 - \cos(4\pi y_j) + 1]^2, \\ f_2 &= 1 - M(t)x_1 + (\tfrac{1}{2N} + \varepsilon)\,|\sin(2N\pi x_1) - 2N\,|G(t)|| \\ &+ \tfrac{2}{|J_2|}\sum_{j\in J_2}[2y_j^2 - \cos(4\pi y_j) + 1]^2, \end{aligned} \tag{9}$$

where $J_1 = \{j|j$ is odd and $2 \le j \le n\}$, $J_2 = \{j|j$ is even and $2 \le j \le n\}$, $\varepsilon = 0.1$, $N = 10$ and $y_j = x_j - \sin(6\pi x_1 + j\frac{\pi}{n});\ j = 2, ..., n$.

Search space: $[0,1] \times [-1,1]^{n-1}$, where n is the number of dimensions of the search space.

Optimal PF: $2N+1$ distinct points: $(\frac{i}{2N}+|G(t)|, 1-\frac{i}{2N}M(t)+|G(t)|), i=0,1,...,2N$.
Optimal PS: $x_j = \sin(6\pi x_1 + j\frac{\pi}{n}), j=2,...,n;\ 0 \le x_1 \le 1$.

G. UDF7: The PS is trigonometric and there is no dynamicity associated with it. The PF is concave and 3D in nature. There is shifting of center and radius of the concave front with time.

$$\begin{aligned} f_1 &= R(t)\cos(0.5\pi x_1)\cos(0.5\pi x_2) + G(t) \\ &+ \tfrac{2}{|J_1|}\sum_{j \in J_1}[x_j - 2x_2\sin(2\pi x_1 + j\tfrac{\pi}{n})]^2 \\ f_2 &= R(t)\cos(0.5\pi x_1)\sin(0.5\pi x_2) + G(t) \\ &+ \tfrac{2}{|J_2|}\sum_{j \in J_2}[x_j - 2x_2\sin(2\pi x_1 + j\tfrac{\pi}{n})]^2 \\ f_3 &= R(t)\sin(0.5\pi x_1) + G(t) + \\ &\tfrac{2}{|J_3|}\sum_{j \in J_3}[x_j - 2x_2\sin(2\pi x_1 + j\tfrac{\pi}{n})]^2 \end{aligned} \tag{10}$$

where $J_1 = \{j|3 \le j \le n, \text{and } j-1 \text{ is a multiple of } 3\}$, $J_2 = \{j|3 \le j \le n, \text{ and } j-2 \text{ is a multiple of } 3\}$ and $J_3 = \{j|3 \le j \le n, \text{ and } j \text{ is a multiple of } 3\}$.

Search space: $[0,1]^2 \times [-2,2]^{n-2}$, where n is the number of dimensions of the search space.

Optimal PF: $(f_1 - G(t))^2 + (f_2 - G(t))^2 + (f_3 - G(t))^2 = [R(t)]^2;\ 0 \le f_1, f_2, f_3 \le 1$.

Optimal PS: $x_j = 2x_2\sin(2\pi x_1 + j\frac{\pi}{n}),\ j = 3, ...n$.

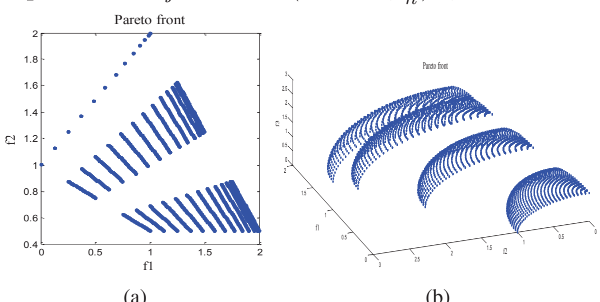


Fig. 11: Optimal PF for (a) UDF6 and (b) UDF7.

The above seven test functions are all deterministic in nature. The changes are all periodic and cyclic with time. One may question the validity of these types of changes. But it is important to ascertain that the changes had to be within a bound, for both theoretical test functions and for practical cases. Now we introduce two new benchmark functions where changes are not at all cyclic and periodic as before. The changes here are random from a pool of changes and simulated using Algorithm 1. This randomness makes the test functions hard to optimize.

Algorithm 1: Pseudocode to simulate UDF8 and UDF9

```
begin
    Initialize t1 = t2 = t3 = t4 = t5 = 0. ;
    Let us define ts as a random variable belonging to the set
    {t1, t2, t3, t4, t5}, where probability of each of the variables being
    selected is equal i.e. P(ts = t1) = P(ts = t2) = P(ts = t3) =
    P(ts = t4) = P(ts = t5) = 1/5. ;
    for each iteration τ = 1 : max_gen do
        if t(τ) ≠ t(τ − 1) then
            Select ts randomly from the set {t1, t2, t3, t4.t5} ;
            Increase the selected variable ts by a step of 1 ;
            Update f1 and f2 as per the current values of t1,t2,t3,t4,t5 ;
        end
    end
end
```

H. UDF8: The PS is trigonometric in nature with random vertical or horizontal shift with time. The PF is continuous with random changes between diagonal shifting, curvature changes and angular shifting. There are five different types of changes occurring at different instance of time.

$$\begin{aligned} f_1 &= x_1 + |G(t_3)| + \\ &\tfrac{2}{|J_1|}\sum_{j \in J_1}[x_j - \sin(6\pi x_1 + (j + K(t_1))\tfrac{\pi}{n}) - G(t_2)]^2, \\ f_2 &= 1 - H(t_4){x_1}^{H(t_5)} + |G(t_3)| + \\ &\tfrac{2}{|J_2|}\sum_{j \in J_2}[x_j - \sin(6\pi x_1 + (j + K(t_1))\tfrac{\pi}{n}) - G(t_2)]^2, \end{aligned} \tag{11}$$

where $J_1 = \{j|j \text{ is odd and } 2 \le j \le n\}$ and $J_2 = \{j|j \text{ is even and } 2 \le j \le n\}$.

Search space: $[0,1] \times [-1,1]^{n-1}$, where n is the dimensionality of the search space.

Optimal PF: $f_2 = 1 - H(t_4)(f_1 - |G(t_3)|)^{H(t_5)} + |G(t_3)|; 0 + |G(t_3)| \le f_1 \le 1 + |G(t_3)|$.

Optimal PS: $x_j = \sin(6\pi x_1 + (j + K(t_1))\frac{\pi}{n}) + G(t_2)$, $j = 2, ..., n;\ 0 \le x_1 \le 1$.

I. UDF9: The PS is polynomial in nature with random vertical shift or variation in the order of the polynomial with time. The PF is continuous with random changes between diagonal shifting, curvature changes and angular shifting. There are five different types of changes occurring at different instance of time.

$$\begin{aligned} f_1 &= x_1 + |G(t_3)| \\ &+ \tfrac{2}{|J_1|}\sum_{j \in J_1}[x_j - {x_1}^{0.5(2+\frac{3(j-2)}{n-2}+G(t_1))} - G(t_2)]^2, \\ f_2 &= 1 - H(t_4){x_1}^{H(t_5)} + |G(t_3)| + \\ &\tfrac{2}{|J_2|}\sum_{j \in J_2}[{x_1}^{0.5(2+\frac{3(j-2)}{n-2}+G(t_1))} - G(t_2)]^2, \end{aligned} \tag{12}$$

where $J_1 = \{j|j \text{ is odd and } 2 \le j \le n\}$ and $J_2 = \{j|j \text{ is even and } 2 \le j \le n\}$.

Search space: $[0,1] \times [-2,2]^{n-1}$, where n is the number of dimensions of the search space.

Optimal PF: $f_2 = 1 - H(t_4)(f_1 - |G(t_3)|)^{H(t_5)} + |G(t_3)|; 0 + |G(t_3)| \le f_1 \le 1 + |G(t_3)|$.

Optimal PS: $x_j = {x_1}^{0.5(2+\frac{3(j-2)}{n-2}+G(t_1))} + G(t_2)$, $j = 2, ..., n$, $0 \le x_1 \le 1$.

Note that in UDF8 and UDF9, several types of dynamic variation of PS and PF have been incorporated to form a pool of changes and at a time only one kind of change is employed. The variables t_1 to t_5 correspond to a single type of variation each. At a particular point of time, only one of the variables is selected randomly and gets increased by 1. The objective functions are obtained according to updated values of the variables.

TABLE I: Summary of proposed benchmark set: Unconstrained Dynamic Functions

Func.	Number of objectives	Pareto Set		Pareto Front		
		Nature	Variation	Nature	Shift	Angular shift / Curvature variation
UDF1	2	Trigonometric	Vertical shift	Continuous	Diagonal	No change
UDF2	2	Polynomial	Curvature change +Vertical shift	Continuous	Diagonal	No change
UDF3	2	Trigonometric	No change	Discrete	Diagonal	No change
UDF4	2	Trigonometric	Horizontal shift	Continuous	No change	Angular shift + Curvature variation
UDF5	2	Polynomial	Curvature change +Vertical shift	Continuous	No change	Angular shift + Curvature variation
UDF6	2	Trigonometric	No change	Discrete	Diagonal	Angular shift only
UDF7	3	Trigonometric	No change	Continuous	Radial + shift of the center	No change
UDF8	2	Trigonometric	Random Vertical or Horizontal shift	Continuous	Random Diagonal Shift	Random Angular Shift or Curvature variation
UDF9	2	Polynomial	Random Curvature change or Vertical shift	Continuous	Random Diagonal Shift	Random Angular Shift or Curvature variation

In the functions UDF3, UDF6 and UDF7 due to the already challenging nature of PFs (which are discontinuous and three dimensional) to avoid a higher complexity in the problem PS variation is not considered in those cases. So the PS diagrams of the static cases are not shown separately. Also for UDF8 and UDF9, due to the stochastic nature of change it is not possible to draw the actual PF or PS variation with time. It must be mentioned that these 9 benchmark functions proposed here are not exhaustive but rather a small subset of all the possible functions that consider all types of possible changes at least once and can be generated by using the generalized benchmark function generator as discussed in Section III. Table I summarizes their properties.

Algorithm 2: Pseudocode of the MOEA/CER algorithm

input : i) MOP(1) ii) a stopping criterion iii) N the number of the subproblems considered in MOEA/DFD [17] iv) a uniform spread of N weight vectors: $\lambda_1, , \lambda_N$ v) T : the number of the weight vectors in the neighborhood of each λ_i
output : $\{\vec{x}^1, \ldots.., \vec{x}^N\}$ and $\{F(\vec{x}^1), \ldots.., F(\vec{x}^N)\}$

Compute the Euclidean distances between any two weight vectors and then find the T closest weight vectors to each weight vector to find $B(i)$
Generate an initial population $\{\vec{x}^1, \ldots.., \vec{x}^N\}$
Initialize $\vec{z}$ by setting
$\vec{z} = \min\{f_i(\vec{x}^1), f_i(\vec{x}^2), \ldots\ldots f_i(\vec{x}^N)\}$
Set $gen = 0 \ \ \forall i = \{1, 2, \ldots N\}$
Set Q_t, Q_{t-1}, Q_{t-2}, as the initial positions of the individuals
if *mod(gen, T)=0* **then**
  Call *Reinitialize*(); \\ Module for prediction based re-initialization;
  Set $P = Q_{t+1}$
end
while *termination criteria not met* **do**
  for $i = 1, 2, \ldots.N$ **do**
    Selection of mating/update range;
    Reproduce the offspring $\vec{u}_i$ corresponding to parent $\vec{x}_i$ by DE/rand/1/bin scheme. For j^{th} component of the i^{th} vector:
    $u_{i,j} =$
    $$\left\{ \begin{array}{ll} u_{i,j} = x_{r_1^i,j} + F.(x_{r_2^i,j} - x_{r_3^i,j}), & \text{if rand}(0,1)\, Cr \\ x_{i,j}, & \text{otherwise} \end{array} \right\}.$$
    F being the standard scale factor for DE ;
    If an element of $\vec{y}$ is out of the boundary reset its value;
    For each $j = 1, 2, .., m$, if $z_j > f_j(\vec{y})$, then set $z_j = f_j(\vec{y})$;
    Calculate fuzzy dominance level and update the solutions as per [17];
    Update $Q_{t-1}, Q_{t-2}, ..$ and $gen = gen + 1$;
  end
end

Algorithm 3: Reinitialize/CE ()

input : Q_t, Q_{t-1}, Q_{t-2}: Current population $(gen = t)$ and last population $(gen = t - 1)$
output : Q_{t+1} : Re-initialized population based on prediction model

for $i = 1, 2, \ldots.N$ **do**
  Find the nearest solution in PF of last generation corresponding to each solution $\vec{x}_{t-1} = \arg\min_{\vec{y} \in Q_{t-1}} \|f(\vec{y}) - f(\vec{x}_t)\|_2$;
  Calculate $d_1^i = (x_t^i - x_{t-1}^i), d_2^i = (x_{t-1}^i - x_{t-2}^i), \ \forall i = 1 : n$;
  Calculate $k^i = |d_1^i| - |d_2^i|$;
  if $rand_i(0, 1) < 0.5$ **then**
    $m^i = 1 + \tanh(k^i)$
  else
    $m^i = \frac{k^i}{|k^i|} N(1, |k^i|)$
  end
  Predict the next position of that solution:
  $x_{t+1}^i = \psi(x_t^i, x_{t-1}^i) = x_t^i + m^i.(x_t^i - x_{t-1}^i), \forall i = 1, 2, \ldots, n$;
  Return Q_{t+1};
end

V. Statistical verification of benchmarks

We consider five Dynamic MO algorithms from the literature and run them over our test-beds to compare their performances and to assess the nature and difficulty of these benchmark functions. Four well-known dynamic MOEAs – DEMO (direction based method) [5], DMEA/PRI [11], DNSGA-II [12] and DQMOO/AR [13] are considered in addition to a variant of MOEA/D [14], MOEA/D-BR (MOEA/D + PS-based nearest distance + Basic Re-initialization Strategy) developed using the re-initialization proposed in [11]. Additionally we have proposed another variant called MOEA/CER that relies on Controlled Extrapolation and PF based nearest distance approach [11]. A re-initialization strategy has been incorporated with the MOEA/D-DE framework to form a complete algorithm outlined in 2 and 3. To make a proper comparison of our benchmark functions with that of FDA given by Farina *et. al.* [5], we also ran the above mentioned algorithms on this test-bed.

For all the test functions, we take $T = 5$ and $n_s = 5$. The lower the values of these two parameters, the more difficult the problems are as far as their dynamic behavior is concerned. Since the UDF series of functions are more challenging, a larger population size is required to achieve satisfactory performance. For UDF1-UDF9, except for UDF7, the population size is kept at 300; for UDF7 (which has three objective functions), the population size is kept at 500. The time variation of the Pareto set is governed by the function: $G(t) = \sin(0.5\pi t)$,, where $t = \frac{1}{n_s} \left\lfloor \frac{\tau}{T} \right\rfloor$ (τ is the current iteration number) with a period of 4 in terms of the time variable t i.e. $4 \times T_s \times n_s = 4 \times 5 \times 5 = 100$ iterations. Here 300 iterations have been taken for all functions. It implies that 3 complete cycles of $G(t)$ have been considered. However, in case of PF variation, $|G(t)|$ is the time dependent factor with a period equal to half of the period of $G(t)$. So 300 iterations mean 6 cycles as far as the variation of Pareto front is concerned.

TABLE II: Mean IGD metric values, standard deviations and individual ranks (within parentheses) for benchmarks UDF1–UDF9. The best results are marked in boldface.

Algo. / Func.	DEMO	DMEA/PRI	DNSGA-II	DQMOO/AR	MOEA/D-BR	MOEA/CER
FDA1	0.1134 ± 0.0098 (4)	0.1228 ± 0.0035 (5)	0.1253 ± 0.0243 (6)	0.1037 ± 0.0042 (3)	0.0932 ± 0.0082 (2)	**0.0583 ± 0.0086 (1)**
FDA2	0.0255 ± 0.0032 (4)	0.0205 ± 0.0018 (3)	0.0298 ± 0.0061 (5)	0.0299 ± 0.0015 (6)	**0.0143 ± 8.54e-4 (1)**	0.0177 ± 0.0015 (2)
FDA3	0.1643 ± 0.0877 (3)	0.1786 ± 0.0564 (5)	0.1853 ± 0.8432 (6)	0.1769 ± 0.0221 (4)	0.1578 ± 0.0483 (2)	**0.1405 ± 0.0516 (1)**
UDF1	0.2240 ± 0.0022 (6)	0.1558 ± 0.0136 (3)	0.2153 ± 0.0558 (5)	0.1989 ± 0.0018 (4)	0.1399 ± 0.0268 (2)	**0.1322 ± 0.0029 (1)**
UDF2	0.0527 ± 0.0028 (4)	0.0455 ± 0.0017 (3)	0.0584 ± 0.0025 (6)	0.0575 ± 8.36e-4 (5)	**0.0351 ± 9.64e-4 (1)**	0.0358 ± 0.0017 (2)
UDF3	0.5843 ± 0.0983 (6)	0.5856 ± 0.0701 (5)	0.5251 ± 0.0141 (3)	0.4775 ± 0.0387 (2)	0.6923 ± 0.1217 (6)	**0.4308 ± 0.0275 (1)**
UDF4	0.4533 ± 0.0905 (6)	0.3092 ± 0.0236 (4)	0.3456 ± 0.0609 (3)	**0.1985 ± 0.0903 (1)**	0.4056 ± 0.0730 (5)	0.2269 ± 0.0614 (2)
UDF5	0.0456 ± 0.0079 (6)	0.0433 ± 0.0024 (5)	0.0379 ± 0.0098 (4)	0.0276 ± 7.78e-4 (2)	0.0327 ± 0.0028 (3)	**0.0235 ± 8.35e-4 (1)**
UDF6	1.7439 ± 0.0229 (6)	1.4639 ± 0.0308 (4)	1.5648 ± 0.0965 (5)	1.2318 ± 0.0530 (2)	1.2587 ± 0.0226 (3)	**1.0080 ± 0.0406 (1)**
UDF7	0.5742 ± 0.0473 (5)	0.4836 ± 0.0948 (4)	0.6846 ± 0.0404 (6)	0.3936 ± 0.0372 (3)	**0.2317 ± 0.0073 (1)**	0.2483 ± 0.0087 (2)
UDF8	0.5328 ± 0.1251 (5)	0.4910 ± 0.0728 (4)	0.5625 ± 0.0825 (6)	0.4414 ± 0.1283 (3)	0.4032 ± 0.1793 (2)	**0.3843 ± 0.0826 (1)**
UDF9	0.1628 ± 0.0281 (5)	0.1496 ± 0.0826 (4)	0.1972 ± 0.0638 (6)	0.1418 ± 0.0273 (3)	0.1376 ± 0.0317 (2)	**0.1149 ± 0.0376 (1)**
Final Rank	5	4	6	3	2	1

For dynamic multi-objective optimization, we generally use an average of the IGD values over all the iterations performed [16]. In Table II, we present the average values of the mean IGD values and their standard deviations over 25 independent runs for the benchmarks UDF1– UDF9 and FDA1–FDA3. Alongside we also indicate the rank of each algorithm for that problem in brackets. The last row presents the average rank of all the algorithms over the entire test-bed. A non-parametric statistical test called Wilcoxons rank sum test for independent samples [16] was conducted at the 5% significance level in order to judge whether the results obtained with the best performing algorithm differed from the final results of the rest of the competitors in a statistically significant way. P-values obtained through the rank sum test over all the benchmark functions are presented in Table III. In these tables, NA stands for *Not Applicable* and occurs for the best performing algorithm itself in each case. If the P-values are less than 0.05 (5% significance level), there is strong evidence against the null hypothesis, indicating that the better final objective function values achieved by the best algorithm in each case is statistically significant and has not occurred by chance.

TABLE III: *P*-values calculated for Wilcoxon‘s rank sum test.

Algo. / Func.	DEMO	DMEA/PRI	DNSGA-II	DQMOO/AR	MOEA/D-BR	MOEA/CER
FDA1	0.0008	0.0038	0.0003	0.0026	0.1173	NA
FDA2	0.0093	0.0482	0.0036	0.0163	NA	0.2163
FDA3	0.0018	0.0128	0.0063	0.0035	0.1234	NA
UDF1	0.0026	0.0027	0.0273	0.0017	0.2192	NA
UDF2	0.0281	0.0027	0.0002	0.0021	NA	0.1963
UDF3	0.0008	0.0284	0.0003	0.1173	0.0620	NA
UDF4	0.0129	0.0182	0.0018	NA	0.0096	0.1973
UDF5	0.0283	0.0092	0.0047	0.1218	0.1273	NA
UDF6	0.0046	0.0138	0.0003	0.0918	0.0936	NA
UDF7	0.0036	0.0928	0.0059	0.0073	NA	0.1836
UDF8	0.0381	0.0483	0.0031	0.0389	0.1730	NA
UDF9	0.0384	0.0083	0.0027	0.0217	0.1129	NA

Test problems such as UDF3, UDF4, UDF6, UDF7 and UDF8 have higher associated values of the average IGD measure than the other benchmark functions. This correctly points out the difficulty associated with these functions. A closer look reveals that, in general, a trigonometric Pareto set variation is much more difficult than a polynomial one. Also, the curvature variation of the Pareto front makes the problem more challenging than the shifting. Other than this, in general, a discrete front and a 3-dimensional front is always a tough situation to deal with. The non-cyclic changes or rather the non-deterministic changes associated with UDF8 and UDF9 make the problem a little bit more challenging. The average IGD values that we obtained while simulating these algorithms over our benchmark test-bed is generally much higher than that obtained in the FDA test-bed. Also, in the FDA test-bed, the average IGD values of each algorithm differ in a smaller proportion with the others compared to our UDF test-bed. That is, the UDF test-bed gives a more diversified result compared to the FDA test-bed, which allows us to clearly distinguish the performance of each algorithm through proper statistical testing.

VI. Conclusions

In this paper, we have proposed a new and more challenging benchmark function generation scheme for DMOPs. Apart from the existing types of changes in PF and PS, we incorporated several other types of changes. Non-continuous PF, and challenging PS have also been dealt with in this paper. It is possible to construct other complex benchmarks by combining the changes in PF and PS proposed here, such that the changes may occur randomly in parallel or in a sequential manner. To support our dynamic MO benchmark generator we have developed a nine benchmark test-bed and have simulated four of the best known state-of-the-art dynamic MOEAs along with two modified MOEA/D variants incorporated with re-initialization strategies already proposed in literature over this test-bed. From the results obtained, we opine that our test-bed is more diverse and challenging than those already present in the current literature.

Dynamic multiobjective optimization still remains as a relatively new and promising research field, which has been studied scarcely so far. Thus, there are plenty of research opportunities within this area. We claim here that the most important issue to be considered when designing new benchmark MOPs is the definition of an appropriate dynamicity process. Besides the current notion of PF and PS variation, one can think of variation in other domains, too.

Finally, there is also a remarkable lack of real-world applications of dynamic MOEAs, in spite of the fact that such problems do exist in different domains of science and engineering such as optimal process control, sensor scheduling in real time for maximizing coverage and minimizing lifetime, etc. Thus, assessing the potential of dynamic MOEAs in the context of practical problems is an important research topic that will be explored by EA-researchers in the near future.

Acknowledgment

We would like to convey our gratitude to Mr. M. Nasir and Mr. S. Sengupta for their contribution towards this work.

References

[1] C. A. Coello Coello, G. B. Lamont, D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving multiobjective Problems*, Springer, 2007.

[2] A. Zhou, B-Y. Qu, H. Li, S-Z. Zhao, P. N. Suganthan and Q. Zhang, ”Multiobjective evolutionary algorithms: a survey of the state-of-the-art”, *Swarm and Evolutionary Computation*, 1(1): 32–49, 2011.

[3] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable test problems for evolutionary multiobjective optimization“, In A. Abraham, L. Jain, and R. Goldberg, editors, *Evolutionary Multiobjective Optimization*, pp. 105–145, London: Springer-Verlag, 2005.

[4] Q. Zhang, A. Zhou, S. Z. Zhao, P. N. Suganthan, W. Liu and S. Tiwari, “Multiobjective Optimization Test Instances for the CEC 2009 Special Session and Competition“, *Technical Report CES-887*, University of Essex and Nanyang Technological University, 2008.

[5] M. Farina, K. Deb, and P. Amato, “Dynamic multiobjective optimization problems: test cases, approximations, and applications“, *IEEE Transactions on Evolutionary Computation*, 8(5): 425–442, Oct. 2004.

[6] Y. Jin and J. Branke, “Evolutionary optimization in uncertain environmentsa survey“, *IEEE Trans. Evol. Comput.*, 9(3): 303–317, 2005.

[7] L. Wen-Fung and G. G. Yen, “PSO-based multiobjective optimization with dynamic population size and adaptive local archives“, *IEEE Trans. Sys., Man & Cyb. (Part B)*, 38(5):1270–1293, Oct, 2008.

[8] Y. Jin and B. Sendhoff, “Constructing dynamic test problems using the multiobjective optimization concept“, In *Applications of Evolutionary Computing*, Vol. 3005 of Lecture Notes in Computer Science, pp. 525–536, Coimbra, Portugal, April 2004, Springer.

[9] E. Zitzler, K. Deb, and L. Thiele, “Comparison of multiobjective evolutionary algorithms: empirical results,“ *Evol. Comput. J.*, 8(2): 125–148, 2000.

[10] J. Mehnen, T. Wagner, and G. Rudolph, “Evolutionary optimization of dynamic multiobjective test functions“, In *Proc. of 2nd Italian Workshop on Evolutionary Computation and 3rd Italian Workshop on Artificial Life*, 2006

[11] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, “Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization“, 4th Int. Conference on Evolutionary Multi-Criteria Optimization, LNCS, Springer, 2007.

[12] K. Deb, U. Bhaskara Rao, and S. Karthik, “Dynamic multi-objective optimization and decision-making using modified NSGA-II: a case study on hydro-thermal power scheduling bi-objective optimization problems“, KanGAL Report No. 2006008, Sept. 2006.

[13] I. Hatzakis and D. Wallace, “Dynamic multi-objective optimization with evolutionary algorithms: a forward-looking approach“, In Proc. of Genetic and Evolutionary Computation Conference (GECCO 2006), pp. 12011208, Seattle, Washington, USA, July 2006.

[14] Q. Zhang and H. Li, “MOEA/D: A multi-objective evolutionary algorithm based on decomposition,“ *IEEE Trans. on Evolutionary Computation*, vol. 11, no. 6, pp. 712731, 2007.

[15] X. Li, J. Branke and M. Kirley, “On performance metrics and particle swarm methods for dynamic multiobjective optimization problems“, in *Proc. of Cong. Evol. Comput.* (CEC ‘07), pp. 1635 - 1643.

[16] F. Wilcoxon, “Individual comparisons by ranking methods“, *Biometrics*, 1, pp. 80-83, 1945.

[17] S. Sengupta, S. Das, M. Nasir, A. V. Vasilakos, and W. Pedrycz, “An Evolutionary Multiobjective sleep-scheduling scheme for differentiated coverage in wireless sensor networks,“ *IEEE Trans. Sys., Man & Cyb. (Part C)*, 42(6):1093–1102, Nov., 2012.