A Scalable Test Suite for Continuous Dynamic Multiobjective Optimisation

Shouyong Jiang, Marcus Kaiser, *Senior Member, IEEE*, Shengxiang Yang, *Senior Member, IEEE*,
Stefanos Kollias, *Fellow, IEEE*, and Natalio Krasnogor

***Abstract*—Dynamic multiobjective optimisation has gained increasing attention in recent years. Test problems are of great importance in order to facilitate the development of advanced algorithms that can handle dynamic environments well. However, many of existing dynamic multiobjective test problems have not been rigorously constructed and analysed, which may induce some unexpected bias when they are used for algorithmic analysis. In this paper, some of these biases are identified after a review of widely used test problems. These include poor scalability of objectives and, more importantly, problematic overemphasis of static properties rather than dynamics making it difficult to draw accurate conclusion about the strengths and weaknesses of the algorithms studied. A diverse set of dynamics and features is then highlighted that a good test suite should have. We further develop a scalable continuous test suite, which includes a number of dynamics or features that have been rarely considered in literature but frequently occur in real life. It is demonstrated with empirical studies that the proposed test suite is more challenging to the dynamic multiobjective optimisation algorithms found in the literature. The test suite can also test algorithms in ways that existing test suites can not.**

***Index Terms*—Dynamic multiobjective optimisation, scalable test problems, dynamics, adversarial examples, Pareto front**

I. Introduction

Multiobjective optimisation problems (MOPs) involving dynamic features are reported in a variety of real applications [7], [10], [17], [48]. This kind of problems is known as dynamic multiobjective optimisation problems (DMOPs). Due to their dynamic nature, the optimisation of DMOPs is more challenging than that of static MOPs as it has to deal with not only conflicting objectives, but also changes in objective functions or constraints. In other words, evolutionary algorithms (EAs) for dynamic multiobjective optimisation must be capable of tracking the changing Pareto-optimal front (PF) and/or Pareto-optimal set (PS) to provide a set of diverse solutions that approximate each new PF or PS over time.

Dynamic multiobjective optimisation (DMO) has attracted increasing research interest in recent years. Significant contributions have been made mainly toward benchmarking [10], [20], algorithm design [5], [22], [34], [52], and real-life application [17], [48]. Despite these, there are still a number of open topics that need to be addressed in order to further advance the development of the DMO research. One important topic is how to benchmark DMO test environments properly for algorithm analysis.

Roughly speaking, DMOPs can be regarded as a sequence of static MOPs (static and stationary are interchangable in this paper, both meaning unchanged states) for a while. In this sense, DMOPs can be constructed as a dynamic version of MOPs used in static multiobjective optimisation (SMO) [6]. For example, an MOP can be made dynamic by adding time-dependent elements, as observed in some early DMO studies [10], [14]. While this benchmarking strategy is of great use to easily develop test problems, there are also some potential limitations which need to be borne in mind. One limitation is that the DMOPs created in this way may be highly similar to each other and also to their static counterparts. As a result, the DMOPs have simple and undiversified problem properties. For example, some FDA [10] and dMOP [14] problems derived from the static ZDT [57] test suite cannot capture (time-varying) mixed PFs which greatly challenge algorithms [20]. A lack of diverse problem properties in benchmarks may not provide a comprehensive analysis of algorithms.

Another limitation is that the base MOPs can be the dominating factor for the difficulty of the resulting DMOPs and therefore decrease the importance of dynamics. A hard-to-solve MOP may not be a good candidate for DMOPs [11]. This is because the critical factor that causes the failure of algorithms to solve the DMOP is not the underlying dynamics but rather the static properties (e.g. strong variable dependencies) of the problem. Some DMOPs like UDF [3] and ZJZ [52] were created based on the LZ test suite [30], but it remains unclear, of the dynamic and static features they involve, which is more challenging to algorithms, since LZ is already challenging in the SMO literature [21], [54].

Having realised the above limitations that exist in DMO test problems, Jiang and Yang [20] have proposed a new benchmark generator for constructing DMOPs. While focusing more on the effect of dynamic features, the proposed benchmark also captures a number of dynamic features that have not or rarely been considered, such as time-varying mixed PF, mixed type of change mode. In another recent work [11], dynamics were intentionally outweighed over static properties when existing MOPs are used to create dynamic scenarios regarding modality, PF tradeoff connectivity, and PF degeneration.

Despite much effort, DMO benchmarking is still far from satisfying. It remains unknown what characteristics are de-

Manuscript received September 11, 2018; revised November 7, 2018; accepted January 16, 2019. This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) of U.K. under Grant EP/N031962/1 and EP/K001310/1. (Corresponding author: Natalio Krasnogor).

Shouyong Jiang, Marcus Kaiser, and Natalio Krasnogor are with Interdisciplinary Computing and Complex Biosystems (ICOS) research group, School of Computing, Newcastle University, Newcastle upon Tyne NE4 5TG, U.K. (email: math4neu@gmail.com, {marcus.kaiser, natalio.krasnogor}@ncl.ac.uk).

Shengxiang Yang is with the Centre for Computational Intelligence (CCI), School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, U.K. (email: syang@dmu.ac.uk).

Stefanos Kollias is with the Machine Learning Group, School of Computer Science, University of Lincoln, Lincoln LN6 7TS, U.K. (email: SKollias@lincoln.ac.uk).

sirable for a good test suite. For example, one important characteristic is the scalability of objectives in DMOPs. It is observed that many real-life applications have more objectives, and the number of objectives can be time-varying [6]. The scalability of objectives, the nature of its dynamics, and the balance between them are poorly studied in existing DMOPs [4], [19]. Another important characteristic that a good test suite should embrace is the detectability of environmental changes. To the best knowledge of the authors, there is no test problem whose changes cannot be detected with one re-evaluation of a random population member. As a result, most algorithms simply assume all changes are knowable, leading to the neglect of this important factor in the existing literature.

In summary, although after more than a decade of research, DMO benchmarking still face a number of issues, which are highlighted as follows:

1) *Problems are not scalable or have limited scalability*. Most constructed DMOPs have a few objectives and the number of objectives is time independent [3], [20].
2) *Static properties outweigh dynamics*. For example, deceptive or strong variable-linkage characteristics obscure the importance of dynamics [3], [52].
3) *Problem properties are similar and not diversified*. DMO benchmarks have similar PF/PS properties and environmental changes, such as FDA [10] and dMOP [14] cases.
4) *Dynamic features are very limited.* Most existing benchmarks cover a small subset of dynamics from real-life applications, and leave unconsidered many other dynamic features, e.g. predictability and visibility [35].

Bearing these in mind, it is not trivial to rethink what characteristics contribute to a good DMO test suite. After enumerating and justifying a number of desirable characteristics that a test suite should have, we are motivated develop a new test suite for DMO. The test suite is aimed to present characteristics linking closely to real-world applications while addressing as many as possible the above-mentioned gaps, and to potentially serve as adversarial counter cases [12] for testing DMO solvers. It provides a systematic tool for algorithm designers and practitioners to comprehensively analyse and develop algorithms.

The rest of this paper is organised as follows. Section II introduces related work and presents desirable features and dynamics for DMOPs. The test suite construction process and the resulting test suite are described in Section III. Section IV presents experimental studies on the proposed test suite. Section V concludes the paper.

II. Related Work

A. Existing Dynamic Multiobjective Test Problems

One of the earliest test suite is FDA [10]. FDA suggested a benchmark design methodology which considers dynamics and multiobjectivity separately. Following this idea, five FDA problems were made by borrowing two important test suites, i.e. ZDT [57] and DTLZ [8], from the SMO literature, and adding dynamics of interest to them. Since establishment, the FDA design methodology has impacted greatly a number of DMO benchmarking studies [3], [11], [14], [16], [17]. Unlike FDA, Jin and Sendhoff [25] developed an open scheme of aggregating objective functions of static problems by dynamically changing weights to form a low-dimensional DMOP. However, no well-defined DMOPs were derived from this scheme. Guan *et al.* [15] studied DMOPs with objective replacement, where some objectives may be replaced with new objectives during the evolution. Mehnen *et al.* [33] 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, which is a generalised 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. Recently, Helbig and Engelbrecht [16] have made a sound investigation into the existing DMOPs used in the literature, and have highlighted the characteristics that an ideal DMO benchmark function suite should exhibit. In addition to highlighting shortcomings of existing DMOPs, they also provided several HE problems with complicated static characteristics borrowing from the SMO literature [18], [30]. As a result, main challenges that the HE problems pose to algorithms are static problem properties instead of dynamics. Similarly, the recent UDF problems [3] and Zhou *et al.*'s work [52] introduce complicated static variable linkages, which can obscure the importance of dynamics.

In the meanwhile, there are several studies concentrating on dynamic aspects of DMOPs. Huang *et al.* [17] proposed some DMOPs where the number of objectives or variables can change over time. Following this direction, Chen *et al.* [5] highlighted challenges in DMOPs with a changing number of objectives, e.g. severe diversity loss, and suggested more DMOPs of this type. The study in [20] proposed a new problem generator that can produce various time-dependent PF geometries for DMOPs. It also suggested some important dynamics, including time-varying variable linkages that cause unbalanced diversity, and mixed types of change modes. But, the study focused mainly on multiobjective cases and had scalability issues, which may hinder its wide application. Gee *et al.* [11] developed a number of GTA problems targeting dynamic modality, tradeoff connectivity, and PF degeneration. Despite great usefulness, the GTA problems limit performance assessment of EAs to the target dynamics.

There are also studies on dynamic time-linkage problems (DTPs) where solutions found for previous environments determine the behaviour of problems for future environments and this type of problems is very common in real applications [35], [36]. Almost all the studies about DTPs focus mainly on single-objective scenarios except [17]. DTPs can exhibit various characteristics, e.g. time/prediction deception [36], and many characteristics are not fully known. Benchmarking DTPs can be very difficult in dynamic multiobjective scenarios and therefore is not considered in this paper.

B. Desirable Features under Dynamic Environments

This subsection presents a number of desirable features and dynamics that have been ignored or rarely discussed in the literature, although we recognise that other dynamic features such as time-dependent variable dependencies, nonseparability are also important under certain circumstances. Note that, the first two features are not dynamic but specially highlighted here due to importance in any scalable test problems.

1) Scalability: DMO test problems should be scalable to have any number of objectives and variables.

Importance: Scalable problems are beneficial for assessing the effect of variation in the number of objectives and variables. Similar to multiobjective benchmarks [18], this property should be considered when building DMO test problems.

2) Shape of PF boundaries: The boundary of a M-objective PF after max-min normalisation fully or partially lies on the hyperplanes defined by all combination of $(M-1)$ axes. For example, normalised PF boundaries can be similar to those of a unit simplex or sphere in the case of $M=3$, a regular shape commonly used for 3-objective test problems.

Importance: The shape of PF boundaries have not been well studied, but it has recently attracted increasing attention due to its significant effect on some algorithms [19]. The majority of existing scalable test problems are designed such that the PF boundary (with or without normalisation) is triangular. However, real-world problems [13], [38] often have more diverse PF shapes instead of triangles. Intuitively, irregular PF boundaries affect directly solution spread of algorithms in objective space, and therefore possibly affect objective normalisation which is very important for solution uniformity for certain problems. The shape of PF boundaries can also possibly influence the computation of corner solutions used in some objective reduction techniques when dealing with many objectives [45].

3) Dynamic concavity-convexity: The PFs of a DMOP are dynamically convex, linear, concave, or mixed in a series of changes.

Importance: Concavity-convexity of DMOPs has a direct impact on the performance of algorithms, as evidenced in their static counterparts [18], [21]. When a DMOP changes its concavity-convexity from one type to another due to an environmental change, it can cause the variation of uniformity of solutions along the PF and potentially lose diversity. As a consequence, EAs face difficulties in uniformly redistributing solutions along the PF or rescuing diversity loss, the latter of which is even a big issue in many-objective cases [23].

Time-varying concavity-convexity has been discussed in a number of biobjective DMOPs [10], [20]. However, this kind of characteristic have not been well understood in DMOPs with a larger or scalable number of objectives.

4) Dynamic PF connectivity: The PF has time-dependent connectivity. At any time the PF can be simply-connected, non-simply connected (e.g., a surface with holes), or have several disconnected PF segments.

Importance: The connectivity of the PF manifold has a great influence on the performance of EAs, as demonstrated in [21]. Time-varying PF connectivity has existed in some real-life problems [10]. In the DMO literature, most DMOPs are constructed to have a simply-connected PF manifold. Although there are several PF-discontinuous DMOPs, none of them is non-simply connected. Benchmarking DMOPs with time-dependent PF connectivity (i.e., time-varying number of PF segments or non-simply connected PF) can help to further our understanding of the capability of EAs.

5) Dynamic PF shapes: The overall PF shape is not fixed, but rather diverse over time.

Importance: Changes in PF shapes over time have been reported in many real applications. In control systems [4], [48], the size of the PF can increase or decrease, resulting in different PF shapes over time. This type of change is very likely to affect solution distribution on the PF. In another study [46], the PF varies partially subject to environmental changes. Partial PF variations may increase the difficulty of change detection for PF-based detection methods.

6) Dynamic modality: Objective functions can have different modal modes (either unimodal or multimodal) over time.

Importance: Multimodal problems are generally more challenging than unimodal ones, as shown in a number of studies of static multiobjective optimisation [8], [57]. They are an important type of optimisation problems that arise frequently from real-world applications. However, multimodal problems have not been well explored in DMO. Most existing DMO test problems are unimodal rather than multimodal, and a few multimodal problems have either a fixed number of local optima or a fixed location of global optima. Dynamic modality changes problem landscapes, making DMOPs more difficult to solve. It also helps to assess the change detection ability of EAs if the global optima is relocated in a new environment.

7) Dynamic PF/solution favourability: From decision makers' point of view, dynamic PF favourability relates to time-dependent preference of some solutions, e.g. knee points/regions [58]. From EAs' point of view, it relates to certain solutions that are easier to obtain than others [24], [31] in different environments.

Importance: Dynamic PF favourability regarding knee points is of high interest to the decision maker. This is because most often such points are more important and dynamically preferred. Likewise, the existence of PF favourability of solutions can create imbalanced difficulties of approximation to different PF regions. As a result, population members of EAs are very likely to be attracted toward easily obtainable PF regions, leaving harder PF regions under-explored or poorly approximated [31]. This dynamic feature has been observed in [28].

8) Time-dependent number of objectives/variables: In dynamic environments, DMOPs are allowed to add new or remove some existing objectives/variables. Note that, this is different from scalability for which the number of objectives remains unchanged over time.

Importance: DMOPs with a time-varying number of objectives/variables appear in many real-life scenarios [2], [44], but have been rarely studied in the DMO literature. The effect of changes in the number of objectives in regular pattern has been recently studied in [5]. Changes in the number of variables or objectives in irregular pattern can complicate DMOPs and create another scenario to study dynamics.

9) Time-dependent degeneration: Most often, the PF of M-objective problems is a $(M-1)$-dimensional manifold under Karush-Kuhn-Tucker conditions. A degenerate PF can be of lower dimension than $(M-1)$. For example, a 3-D problem may degenerate to have a PF that is a curve or line segment.

Importance: PF-degenerate problems can raise serious challenges to some algorithms. One challenge, for example, can be the need of effective diversity maintenance to guarantee a uniform spread of solutions. Degeneration has been reported in PID controller design problems [10].

10) Detectability: A DMOP can be easily or hardly detectable, depending on how much environmental changes are computationally visible to optimisers.

Importance: Detectability of DMOPs is of great importance, but it has received little attention in the field of EMO. Almost all existing DMO test problems are made easily detectable for every single environmental change, and a population-based algorithm therefore can exactly detect the change by simply re-evaluating any one solution of population [43]. The importance of detectability has been recognised in dynamic single-objective optimisation [41]. In DMO, failing to detect changes that have actually occurred not only results in ineffective tracking of the new PF, but also could mislead the search process because nondominated solutions obtained so far may be no longer nondominated in new environments.

11) Predictability: The location of PS can change in a predictable or unpredictable manner over time.

Importance: The importance of the predictability of optima has been already recognised in dynamic single-objective optimisation [47]. Existing DMO test problems were made to vary PS locations in a regular and predictable pattern, e.g., the popular FDA [10] and dMOP [14] test problems have a PS of $x_{i=M:n}$=sin$(0.5\pi t)$ (n is the number of variables) at time t. It has been found that the dynamics of PS are more likely unpredictable and sometimes even random in many real-life problems [37]. This gap should be addressed by investigating DMOPs with unpredictable or random changes of PS.

III. SDP: Scalable Dynamic Multiobjective Test Suite

A. Basic Framework

The proposed SDP test suite was constructed with component functions, a way that has been commonly used in building various popular multiobjective problems, such as DTLZ [8], WFG [18], and LZ [30] test suites. Mathematically, SDP can be described as:

$$\text{Minimise} \qquad (f_1(\mathbf{x},t),\dots,f_M(\mathbf{x},t)) \tag{1}$$

with

$$f_{i=1:M}(\mathbf{x},t) = (1+g(\mathbf{x},t))\mu_i(\mathbf{x},t)+\nu(t) \tag{2}$$

where

- $\mu(\mathbf{x},t)=(\mu_1,\dots,\mu_M)$ is a time-dependent function that describes PF properties, such as geometry, degeneration.
- $g(\mathbf{x},t)$ is a time-dependent function that defines PS properties, such as optima location, modality, variable dependency. The PF of (1) is obtained when $g(\mathbf{x},t)=0$.
- $\nu(t)$ is a time-dependent function that allows the PF to move away from or back to its initial location (i.e., the PF at $t=0$). In the work, $\nu(t)$ is either zero or $|\sin(0.5\pi t)|$.

$\mathbf{x}=(x_1,\dots,x_n)\in R^n$ is the decision variable vector and can be divided into two subvectors, i.e., the distance-related $\mathbf{x_I}=(x_1,\dots,x_{M-1})$ for $\mu(\mathbf{x},t)$ and position-related $\mathbf{x_{II}}=(x_M,\dots,x_n)$ mainly for $g(\mathbf{x},t)$, according to [18]. This work constrains the search space $\mathbf{x}$ to lie in $[0,1]^n$, unless otherwise stated. The search space can be also scaled to a bigger range. $t\in R$ is the discrete time instant. $f_i(\mathbf{x},t)$ is the i-th objective value of solution $\mathbf{x}$ at time t.

B. Building Component Functions

In what follows, component functions $g(\mathbf{x},t)$ and $\mu(\mathbf{x},t)$ of (2) are built to adhere to the desirable features in Section II-B. $\nu(t)$ is ignored as it is fairly easy to build. For example $\nu(t)$ can be assigned zero if disregarding the shift of PF or $|\sin(0.5\pi t)|$ if considering a time-varying shift.

1) Scalability and Continuous PF Geometry

These two are determined by $\mu(\mathbf{x},t)$ and often coupled, and here PF convexity-concavity and boundaries shapes are the focus of discussion. In the SMO literature, $\mu(\mathbf{x},t)$ was built to be a scalable linear hypersimplex [6], [18] defined as

$$\mu_{linear,i}{=}\begin{cases}(1-x_i)\prod_{j=1}^{i-1}x_j & \text{if } i=1\\ x_1\dots x_{M-2}x_{M-1} & \text{if } 2{\le}i{<}M\end{cases} \tag{3}$$

or spherical hypersurface (the first hyperorthant) defined as

$$\mu_{sphere,i}{=}\begin{cases}\sin(y_i)\prod_{j=1}^{i-1}\cos(y_j) & \text{if } 1{\le}i{<}M\\ \cos(y_1)\dots\cos(y_{M-1}) & \text{if } i=M,\end{cases} \tag{4}$$

where $y_i = 0.5\pi x_i$. Both of them have non-concave geometries. Such plain PF shapes are well defined and easy to understand, but insufficient to help draw a comprehensive conclusion on the overall performance of any assessed algorithms. The limitation of hypersimplex or hypersphere based PFs has been identified recently [19]. It is also noticed that real-world applications have more complicated and diverse PF geometries [13], [38]. In what follows, we introduce several selected novel PF geometries with interesting properties.

a) Product-form PF: The product of all objective values is one. It means each objective is inversely proportional to other objectives. The PF is defined as

$$\mu_{\prod,i} = \frac{x_i}{\sqrt[M-1]{\prod_{j=1,j\neq i}^{M} x_j}} \tag{5}$$

It is noticed that solutions are not uniformly distributed on the PF (Fig. S1(a), supplementary material), implying boundary solutions be much difficult to be found than intermediate solutions. Note that, the overall curvature of this PF can be adjusted by a mapping: $\mu \to \mu^c$, where c can be a constant (i.e, 2) or a time-varying function (e.g., $|\sin(t)|$). The mapping can further complicate the PF geometry and is applicable to any scalable PF mentioned in this paper.

b) Sum-of-reciprocals PF: The sum of reciprocals of f_i+1 equals one. $\mu(\mathbf{x},t)$ is defined as

$$\mu_{\sum,i} = \begin{cases}\frac{1+t+\sum_{j=1,j\neq i}^{M-1}x_j}{x_i} & \text{if } 1\le i<M\\ \sum_{i=1}^{M-1}\frac{x_i}{1+t} & \text{if } i=M,\end{cases} \tag{6}$$

One important feature of this PF is that corner PF points are not located on the axes (Fig. S1(b), supplementary material), and algorithms relying on them for objective normalisation can be affected. Also, the irregular shape of the boundary poses challenges to decomposition-based algorithms [49] that employ weight vectors, as not all weight vectors sampled from a unit simplex pass through the PF.

c) Mixed/knee PF: The PF has both local convex and concave regions, and sometime produce some knee points that may be hard to approximate. A mathematical description can be

$$\mu_{knee,i}=\begin{cases}(1-x_i+0.05\sin(w\pi x_i))\times \\ \quad\prod_{j=1}^{i-1}(x_j+0.05\sin(w\pi x_j)) & \text{if } 1\leq i<M \\ \prod_{i=1}^{M-1}(x_i+0.05\sin(w\pi x_i)) & \text{if } i=M,\end{cases} \tag{7}$$

where w is a parameter (either static or dynamic) controlling the number of knee regions on the PF (see the illustrative PF in Fig. S1(c), supplementary material). $p_i(\mathbf{x},t)$ of (7) can have a slight adaptation

$$\mu_{mix,i}=\begin{cases}x_i & \text{if } 1\leq i<M-1 \\ (s+0.05\sin(w\pi s)) & \text{if } i=M-1 \\ (1-s+0.05\sin(w\pi s)) & \text{if } i=M,\end{cases} \tag{8}$$

where $s=\sum_{i=1}^{M-1}x_i/(M-1)$ and $w=\operatorname{sgn}(rnd_t-0.5)\lfloor 6|\sin(0.5\pi t)|\rfloor$ ($sgn(\cdot)$ is the sign function and rnd_t is a random value in [0,1]) is recommendation of use. This change results in a very different PF shape, and the PF has mixed regions and its boundary is irregularly shaped (Fig. S2, supplementary material), which can be challenging for some algorithms [19].

2) PF Connectivity

The PF can be simply connected (sometimes called continuous), non-simply connected, or disconnected. Here, we focus on the construction of non-simply connected and disconnected $\mu(\mathbf{x},t)$.

a) Non-simply connected PF: To the best of the authors' knowledge, there is no problem in the multiobjective literature that has a non-simply connected PF. To bridge this gap, we put forward a way to construct a non-simply connected PF. It works as follows. Let $\mu(\mathbf{x},t)$ describes a simply connected PF, it is multiplied by $g(\mathbf{x},t)$, a function that has a value larger than one for some $\mathbf{x}$ components and equals one for others. $g(\mathbf{x},t)$ deforms some regions of the PF of $\mu(\mathbf{x},t)$ such that points on these regions are dominated by the other regions. For example, the spherical PF $\mu_{sphere,i}$ can be multiplied by

$$g_{nsc}=1+\left|\prod_{j=1}^{M-1}\sin(\lfloor k_t(2x_j-r)\rfloor\pi/2)\right|, \tag{9}$$

where $r=1-\bmod(k_t,2)$ and $k_t=\lfloor 10\sin(0.5\pi t)\rfloor$ is used to produce a changing number of holes on the PF generated by $g(\mathbf{x},t)\mu(\mathbf{x},t)$ (Fig. S3, supplementary material).

b) Disconnected PF: Unlike non-simply connected PFs, a disconnected PF consists of multiple disconnected PF components. This type of PF can be generated from

$$\mu_{disc1,i}=\begin{cases}\cos^2(0.5\pi x_i) & \text{if } 1\leq i<M \\ \sum_{j=1}^{M-1}[\sin^2(0.5\pi x_j)+ \\ \quad\sum_{j=1}^{M-1}\sin(0.5\pi x_j)\cos^2(k_t\pi x_j)] & \text{if } i=M,\end{cases} \tag{10}$$

where k_t ($k_t=\lfloor 6\sin(0.5\pi t)\rfloor$ is suggested) renders a changing number of disconnected components (Fig. S4, supplementary material). Alternatively, we can also construct a disconnected PF as follows:

$$\mu_{disc2,i}=\begin{cases}x_i & \text{if } 1\leq i<M \\ (2-s-\sqrt{s}(-\sin(2.5\pi s))^{r_t}) & \text{if } i=M,\end{cases} \tag{11}$$

where $s=\sum_{i=1}^{M-1}x_i^2/(M-1)$ and r_t is a time-varying parameter impacting the PF shape, which is suggested to be $\lfloor 10|\sin(0.5\pi t)|\rfloor$. Different r_t settings result in various PF characteristics (Fig. S5, supplementary material).

3) Detectability

Environmental changes of existing DMOPs are made easily detectable within one re-evaluation of any solutions. Real-world dynamic changes should not be so simple.

There are two ways to increase the difficulty of change detection. One way is to change the global basin of attraction of the previous environment into a local one of the new environment. This can be done by defining $g(\mathbf{x},t)$ as

$$g_{multi}=\frac{1}{n-M+1}\sum_{i=M}^{n}\min_{k=1,\dots,5}\{h_k+10(10x_i-y_k)^2\}, \tag{12}$$

where h_k ($k=1,\dots,5$) defines 5 local minima, one of which is global. y_k defines the location of the local minima. A recommendation is $h_k=k$ and $y_k=2(k-1)$. The global optima is set to $h_{k_t}=0.5$ with $k_t=\lceil 5rnd_t\rceil$, at time t. Fig. S6 in the supplementary material shows deceptive changes, which will leave undetected if detectors are in the unchanged basins of attraction (this happens if not all population members reach the global basin of attractions).

Another way to create a less detectable scenario is to have a time-varying search subspace that causes objective variation. As a result, only detectors falling within this subspace can detect environmental changes. Let $\mu_{detect,i}=\mu_{sphere,i}$ for $i=2,\dots,M$ and its first objective $\mu_1(\mathbf{x},t)$ be

$$\mu_{detect,1}=\begin{cases}|k_t(\cos(0.5\pi x_1)-\cos(0.5\pi\alpha_t)) \\ \quad+\sin(0.5\pi\alpha_t)| & \text{if } x_1\leq\alpha_t \\ \sin(0.5\pi x_1), & \text{otherwise}\end{cases} \tag{13}$$

where $\alpha_t\in(0,1)$ defines a subspace $[0,\alpha_t]$, where x_1 renders dynamic variation of the M-th objective, and k_t is a parameter controlling the severity of change regarding the PF. Environmental changes on the PF can be detected only when detectors have its x_1 smaller than α_t. We recommend the use of $\alpha_t=0.5|\sin(\pi t)|$ and $k_t=10\cos(2.5\pi t)$. Fig. S7 in the supplementary material illustrates the change of the PF derived from (13).

4) PS randomness

Existing DMOPs favour prediction-based EAs, because the PS changes in regular pattern in dynamic environments. Here, we present a PS (distance-related variables x_i) that changes in a random manner:

$$\bar{x}_i^t=\begin{cases}\bar{x}_i^{t-1}+5(rnd_t-0.5)\sin(0.5\pi t) & \text{if } 0\leq\bar{x}_{i,t}\leq 1 \\ rnd_t & \text{otherwise}\end{cases} \tag{14}$$

where rnt_t is a random value from [0,1) and $\bar{x}_i^t$ is the optimal value of $x_i\in[0,1]$ at time t and $\bar{x}_i^0=i/n$ is recommended. Note that (14) can be scaled to any range restriction of x_i.

5) Dynamic number of objectives/variables

a) Dynamic number of variables: Let n_t be the number of variables at time t, the simplest way to make n_t change over time is probably to pick n_t randomly from a range $[n_l, n_u]$:

$$n_t = n_l + \lfloor rnd_t(n_u - n_l) \rfloor. \tag{15}$$

Then the $n_t - M + 1$ distance-related variables are used to build $g(\mathbf{x}, t)$, e.g., $g(\mathbf{x}, t) = \sum_{i=M}^{n_t} x_i^2$.

b) Dynamic number of objectives: Let M_t be the number of objectives at time t, it can can be defined similarly to n_t:

$$M_t = M_l + \lfloor rnd_t(M_u - M_l) \rfloor, \tag{16}$$

which means M_t is randomly generated from a range $[M_l, M_u]$. Correspondingly, the M_t-dimensional PF can be constructed as

$$\mu_{dobj,i} = \sin(y_i) \prod_{k=1}^{i-1} \cos(y_k), \tag{17}$$

where $y_k = \pi(x_k + 1)/6$ for $k = 1, \ldots, M_t$. Thus, the PF always satisfies $\sum_{i=1}^{M_t - 1} \mu_i^2 + 4\mu_{M_t}^2 = 1$ for any M_t.

6) Degeneration

Time-varying PF degeneration can be achieved by specifying first a degeneration level d_t at time t, and then constructing the d_t-D PF. The following presents two degenerate examples.

a) Linear degenerate PF: This is based on the simplex-shape PF of (3). It is defined as

$$\mu_{ldeg,i} = \begin{cases} (1 + g - y_i) \prod_{j=1}^{i-1} y_j & \text{if } 1 \leq i < M, \\ y_1 \ldots y_{M-2} y_{M-1} & \text{if } i = M. \end{cases} \tag{18}$$

with

$$y_i = \begin{cases} x_i, & \text{for } 1 \leq i \leq d_t \\ 0.5 + x_i |\sin(0.5\pi t)| g, & \text{for } d_t < i \leq M - 1 \end{cases}$$

where the g function is already defined in (1). It indicates that the last $M - d_t$ objectives are positively correlated. Fig. S8(a) (supplementary material) shows a linear degenerate PF of $M = 3$ and $d_t = 1$.

b) Spherical degenerate PF: This requires the sphere-shape PF. It is defined as

$$\mu_{sdeg,i} = \begin{cases} \sin(y_i) \prod_{j=1}^{i-1} \cos(y_j) & \text{if } 1 \leq i < M, \\ \prod_{j=1}^{M-1} \cos(y_i) & \text{if } i = M. \end{cases} \tag{19}$$

with

$$y_k = \begin{cases} 0.5\pi x_k, & \text{for } 1 \leq i \leq d_t \\ \arccos\left(\frac{0.5\sqrt{2}}{1 + x_k |\sin(0.5\pi t)| g}\right), & \text{for } d_t < i \leq M - 1 \end{cases}$$

where $k = (p_t + i - 1) \bmod (M - 1) + 1$ with p_t randomly chosen from $[1, M - 1]$. This also results in $M - d_t$ correlated objectives whose indices are determined by p_t. Thus, the spherical degenerate PF is more difficult than the linear one. Fig. S8(b) (supplementary material) shows a spherical degenerate PF of $M = 3$ ($d_t = 1$ and $p_t = 2$).

7) PF shrinkage/expansion

Another scenario we would like to consider is the shrinkage or expansion of the PF over time. Here, we create this scenario based on the spherical PF of (19), in which y_i is defined as

$$y_i = \frac{\pi}{6} G_t + \left(\frac{\pi}{2} - \frac{\pi}{3} G_t\right) x_i, \tag{20}$$

where $G_t = |\sin(0.5\pi t)|$. Fig. S9 (supplementary material) shows the scenario of PF shrinkage over time.

8) Search favourability

Often than not, EAs favour PF regions that are easy to approximate. It is interesting to create a dynamic scenario where favoured PF regions change over time and investigate its effect on EAs. Consider the following g function

$$g_{favour} = \begin{cases} \sum_{i=M}^{n} (-0.9 p_i^2 + |p_i|^{0.6}), & x \in \Phi(\mathbf{x}, t), \\ p_i^2, & \text{otherwise.} \end{cases} \tag{21}$$

where $p_i = x_i - |\sin(0.5\pi t)|$ and $\Phi(\mathbf{x}, t)$ is a time-varying subspace for which barriers exist around the global optima. This means the PF region corresponding to $\Phi(\mathbf{x}, t)$ is much unfavoured than other regions. Here, we recommend $\Phi(\mathbf{x}, t) = \{x \in [0, 1]^n \mid 3t - \lfloor 3t \rfloor \leq \sum_{i=1}^{M-1} x_i \leq 3t + 0.2 - \lfloor 3t + 0.2 \rfloor\}$.

C. Test Suite

A total of 15 scalable dynamic problems (SDP) are instanced using the above-mentioned component functions to imitate a number of representative and well-known real-world features. These are briefly tabulated in Table I, and a detailed description of SDP can be found in Table S1 of the supplementary material.

Note that, SDP uses different g functions, capturing a variety of PS shapes. For example, there are several problems (e.g. SDP2) for which the PS centroid and some PS regions move in distinct directions in the event of changes (most existing problems do not have this feature, see Fig. S10 for details). This may be a new challenge for recently popular centroid-based prediction algorithms [52], [55]. Prediction algorithms may encounter difficulties when solving problems like SDP1, since the PS varies in a random manner. The PS shapes for all the problems could be made as complicated as LZ problems [30], but such complexity is not, and also should not be the main focus of DMOPs, as its existence shadows the importance of landscape dynamics [11].

Table II provides a comparison between SDP and existing DMOPs. For each property, '✓' ('✗') represents the presence (absence) of this property for the corresponding test suite. In other words, environmental changes in those problems can be detected with one re-evaluation of a random population member. As can be seen, multimodality and variable linkages have gained most attention, and SDP embraces a variety of rarely studied but important characteristics. The source code of SDP is available at http://homepages.cs.ncl.ac.uk/shouyong.jiang/.

D. Links between SDP and Applications

SDP captures a number of real-world features. Examples include but not limited to the following applications: Electromagnetic micromirrors [38], crop growth [40], railway rescheduling [9], dynamic subset sum [42], speed reducer model [51], pressure vessel model [51], greenhouse control [48], predictive control [4], weapon selection & planning [46], PID control [10], unstable plant control [17], self-paced learning [28], spacecraft equipment layout [27], distance minimisation [56], job shop scheduling [44], engine calibration [32], and water distribution systems [39].

Fig. 1 briefly shows the links between the above applications and SDP features. Please refer to Section VIII of the supplementary material for more details on how they are linked to each other.

TABLE I
DEFINITION OF SDP PROBLEMS. PF AND VARIABLE RANGES CAN BE ADJUSTED BASED ON [18]. ALSO, $x_i \in \mathbf{x_I}$ IS REPLACED BY $x_i^{K(t)}$ WHERE $K(t) = \exp(10\sin(0.5\pi t))$ IF DYNAMICS ON DIVERSITY ARE CONCERNED

Prob.	Definition	Prob.	Definition
SDP1	$f_{i=1:M}(\mathbf{x},t) = (1+g)\mu_{\prod,i}$ $g = \sum_{i=M+1}^{n}(x_i - \bar{x}_{i,t})^2$ $\triangleright$ $\bar{x}_{i,t}$ updated by (14) search space: $1 \leq x_{i=1:M} \leq 4, 0 \leq x_{i=M+1:n} \leq 1$	SDP9	$f_{i=1:M}(\mathbf{x},t) = (1+g\cdot\max(0,i-M-1))\mu_{disc1,i} + G(t)$ $g = \sum_{i=M}^{n}(x_i - \frac{1}{\pi}\lvert\arctan(\cot(3\pi t^2))\rvert)^2$ $G_t = \lvert\sin(0.5\pi t)\rvert,$ search space: $0 \leq x_{i=1:n} \leq 1$
SDP2	$f_{i=1:M-1}(\mathbf{x},t) = (1+g)\mu_{\sum,i}$ $g = \sin(\frac{\pi}{8}x_1)\sum_{i=M}^{n}(x_i - \cos(t+2x_1))^2$ search space: $1 \leq x_{i=1:M-1} \leq 4, -1 \leq x_{i=M:n} \leq 1$	SDP10	$f_{i=1:M}(\mathbf{x},i) = (1+g\cdot\max(0,i-M-1))\mu_{disc2,i}$ $g = \sum_{i=M}^{n}(x_i - \sin(x_1+0.5\pi t))^2$ search space: $0 \leq x_{i=1:M-1} \leq 1, -1 \leq x_{i=M:n} \leq 1$
SDP3	$f_{i=1:M}(\mathbf{x},t) = (1+g)\mu_{knee,i}$ $y_i = x_i - \cos(t)$ $k_t = \lfloor 5\lvert\sin(\pi t)\rvert\rfloor,$ $g = \sum_{i=M}^{n}(4y_i^2 - \cos(2k_t\pi y_i)+1)$ search space: $0 \leq x_{i=1:M-1} \leq 1, -1 \leq x_{i=M:n} \leq 1$	SDP11	$f_{i=1:M}(\mathbf{x},t) = (1+g_{favour})\mu_{sphere,i}$ search space: $0 \leq x_{i=1:n} \leq 1$
SDP4	$f_{i=1:M}(\mathbf{x},t) = (1+g\cdot\max(0,i-M-1))\mu_{mix,i}$ $g = \sum_{i=M}^{n}(x_i - \cos(t+x_1+x_{i-1}))^2$ search space: $0 \leq x_{i=1:M-1} \leq 1, -1 \leq x_{i=M:n} \leq 1$	SDP12	$f_{i=1:M-1}(\mathbf{x},t) = (1+g)(1-x_i)\mu_{linear,i}$ $g = \sum_{i=M}^{n_t}(x_i - \sin(n_t t)\sin(2\pi x_1))^2$ $\triangleright$ n_t from (15) search space: $0 \leq x_{i=1:M-1} \leq 1, -1 \leq x_{i=M:n_t} \leq 1$
SDP5	$f_{i=1:M}(\mathbf{x},t) = (1+g)\mu_{sphere,M-i+1}$ $\triangleright$ y_i from (20) $g = G_t + \sum_{i=M}^{n}(x_i - 0.5G_t x_1)^2, G_t = \lvert\sin(0.5\pi t)\rvert$ search space: $0 \leq x_{i=1:n} \leq 1$	SDP13	$f_{i=1:M_t}(\mathbf{x},t) = (1+g)\mu_{dobj,i}$ $\triangleright M_t$ from (16) $g = \sum_{i=M+1}^{i=n}(x_i - \frac{it}{M+it})^2$ search space: $0 \leq x_{i=1:n} \leq 1$
SDP6	$f_{i=1:M}(\mathbf{x},t) = (1+g)\mu_{detect,M-i+1}$ $g = \sum_{i=M}^{n}(x_i - 0.5)^2(1+\lvert\cos(8\pi x_i)\rvert)$ search space: $0 \leq x_{i=1:n} \leq 1$	SDP14	$f_{i=1:M-1}(\mathbf{x},t) = (1+g)\mu_{ldeg,i}$ $d_t = 1 + \lfloor\lvert(M-2)\cos(0.5\pi t)\rvert\rfloor$, $g = \sum_{i=M}^{n}(x_i - 0.5)^2$ search space: $0 \leq x_{i=1:n} \leq 1$
SDP7	$f_{i=1:M}(\mathbf{x},t) = (0.5+g_{multi})\mu_{linear,i}$ search space: $0 \leq x_{i=1:n} \leq 1$	SDP15	$f_{i=1:M-1}(\mathbf{x},t) = (1+g)^i\mu_{sdeg,i}$ $d_t = rndi_t^1(1,M-1)$, $p_t = rndi_t^2(1,M-1)$ $\triangleright$ p_t for (19) $g = \sum_{i=M}^{n}(x_i - \frac{d_t}{M-1})^2$ search space: $0 \leq x_{i=1:n} \leq 1$
SDP8	$f_{i=1:M}(\mathbf{x},t) = (1+g)\mu_{sphere,M-i+1},$ $g = \sum_{i=M}^{n}(x_i - \sin(tx_1))^2 + g_{nsc},$ search space: $0 \leq x_{i=1:M} \leq 1, -1 \leq x_{i=M:n} \leq 1$		

The time t is $\frac{1}{n_t}\lfloor\frac{\tau}{\tau_t}\rfloor$, where n_t, τ_t, τ represent the severity of change, frequency of change, and iteration counter, respectively.
$rndi_t(a,b)$ generates a random integer from $[a,b]$ at time t. sgn(·) denotes the sign function. $\triangleright$ starts a comment.

TABLE II
COMPARISON BETWEEN SDP AND EXISTING DMO PROBLEMS

Test Suite	Scalability	Multi-modality	Variable linkage	Irregular PF boundary	PS unpre-dictability	Multi-knee	Search fa-vourability	Degen-eration	PF con-nectivity	dynamic M/n	Detectability of change	Origination
SDP	✓	✓	✓	✓	✓	✓	✓	✓	s,ns,d	M,n	✓	new
FDA [10]	✗	✗	✗	✗	✗	✗	✗	✗	s	✗	✗	ZDT/DTLZ
DIMP [26]	✗	✗	✗	✗	✗	✗	✗	✗	s,d	✗	✗	ZDT
DSW [33]	✗	✗	✗	✗	✗	✗	✗	✗	s	✗	✗	
dMOP [14]	✗	✗	✗	✗	✗	✗	✗	✗	s	✗	✗	ZDT
T [17]	✓	✓	✗	✗	✗	✗	✗	✗	s	M,n	✗	DTLZ
ZJZ [52]	✗	✗	✓	✗	✗	✗	✗	✗	s	✗	✗	FDA/LZ
HE [16]	✗	✓	✓	✗	✗	✗	✗	✗	s,d	✗	✗	FDA/LZ/WFG
UDF [3]	✗	✓	✗	✗	✓	✗	✗	✗	s,d	✗	✗	ZDT
JY [20]	✗	✓	✓	✗	✗	✓	✗	✗	s,d	✗	✗	
GTA [11]	✗	✓	✓	✗	✗	✗	✗	✓	s,d	✗	✗	ZDT/DTLZ
CLY [5]	✓	✓	✗	✗	✗	✗	✗	✗	s	M	✗	T/DTLZ

‘s’, ‘ns’, or ‘s’ in the column of PF connectivity denote DMOPs exist having a simply-connected, non-simply connected, or disconnected PF, respectively.
‘M’ or ‘n’ in column of ‘dynamic M/n’ means DMOPs exist having a time-varying number of objectives or variables, respectively.

The benchmarks have a few limitations due to the fact that real applications are highly complex and diverse, and there are still a lot features that are not well characterised. Thus, it is difficult to develop a perfect test suite that can model and reflect well all possible real-world features. Despite that, constructing some test scenarios based on well characterised features is of great importance for the analysis of EAs. The proposed SDP test suite simulates a number of real-world features in a simplified manner from the literature. The simplification somehow reduces realisticity but is still helpful for understanding of strengths and weaknesses of algorithms. Also, some of the test cases may serve as adversarial examples for DMOP solvers.

IV. EXPERIMENTAL STUDIES

A. DMO Algorithms

Five different DMO algorithms are employed for studying the proposed benchmarks, each representing a type of EAs. They are 1) the Pareto dominace based DNSGA-II [7]; the decomposition-based MOEA/D [49]; the multipopulation based dCOEA [14]; the population prediction based PPS [52]; (5) and the steady-generational SGEA [22]. In this paper, PPS with regularity model [50] (PPS+RM2) and nondominated sorting (PPS+NS) are both analysed. All the algorithms are allowed to use 10% of the population as change detectors in every generation. Each algorithm has a way to respond to the detected environmental changes. To be more specific, DNSGA-II hypermutates a portion of population members. MOEA/D re-evaluates the population. PPS records history search information and predicts the new location of the PF/PS. dCOEA increases population diversity by retrieving its memory population whereas SGEA relocates a portion of population that have high density. More detailed information can be referred to the original papers.

In the experiment, 2, 3 and 5 objectives were tested for SDP1-12. The number of objectives varied from 2 to 5 for

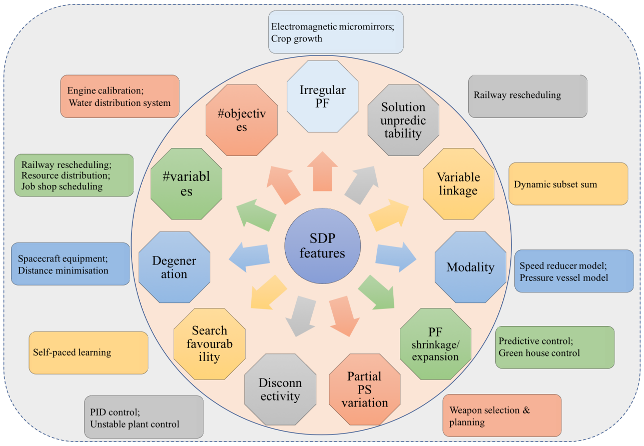


Fig. 1. Links between SDP features and real-world applications.

SDP13, and was fixed to 5 for SDP14-15 to simulate PF degeneration. Since all the algorithms except MOEA/D were originally designed for multiobjective cases, their performance may degrade vastly for the 5-objective case, a well-recognised issue in many-objective optimisation [23]. Here, such algorithms were incorporated with the shift-based density estimator (SDE) [29] to alleviate this scalability issue. The number of variables (n) for all SDP instances except SDP12 was 10. For SDP12, the lower bound $n_l$and upper bound n_u were 10 and 20, respectively. The severity and frequency of change was set to n_t=10 and τ_t=10, respectively. All the algorithms with a population size of 100 (note that, the same number of weight vectors was generated for MOEA/D) were executed 30 runs for each SDP instance. Each execution was terminated after $30\tau_t+50$ generations, where the first environmental change occurs after 50 generations.

B. DMO Performance Measures

The paper uses the following three popular performance measures for algorithm analysis.

1) Mean inverted Generational Distance (MIGD): MIGD [20] is adapted from IGD [50], a static performance indicator that measures both the convergence and diversity of solutions found by an algorithm. The reference set consists of around 1000 uniformly sampled points from the PF.

2) Mean Hypervolume Difference (MHVD): The MHVD [22] is a modification of the static measure HVD [53] that computes the gap between the hypervolume of the obtained PF and that of the true PF. The reference point for the computation of M-dimensional 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.

3) Mean Detection Timeliness (MDT): MDT is proposed to measure how timely an environmental change is detected, which is defined as:

$$MDT = \frac{1}{T}\sum_{i=1}^{T} DT_t \tag{22}$$

with

$$DT_t = \begin{cases} \frac{Eval_t - 1}{n_d\tau_t - 1} & \text{if change detected,} \\ 1 & \text{otherwise,} \end{cases} \tag{23}$$

where $Eval_t$ is the number fitness re-evaluations to detect a change. n_d is the number of candidate detectors in every generation. The smaller DT_t, the earlier a change detected. The earlier the change detected, the more time left for algorithms to react to the change. This new measure is expected to facilitate detectability analysis of some SDP problems, i.e. SDP6-7.

TABLE III
PERFORMANCE RANKING BASED ON THE MIGD MEASURE FOR SDP1-15

Alg.	SDP															Avg.
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	rank
SGEA	3	5	2	2	2	2	2	2	1	2	4	3	2	1	3	1
MOEA/D	6	6	3	5	4	5	4	6	3	5	6	4	6	6	2	6
DNSGA-II	5	3	6	3	3	1	3	5	3	4	5	6	3	1	5	4
dCOEA	1	4	1	6	5	3	1	1	6	6	1	1	1	3	1	2
PPS+RM2	2	1	5	1	1	6	6	2	2	1	3	5	4	5	3	3
PPS+NS	4	2	4	4	6	4	5	2	5	3	2	2	5	4	6	5

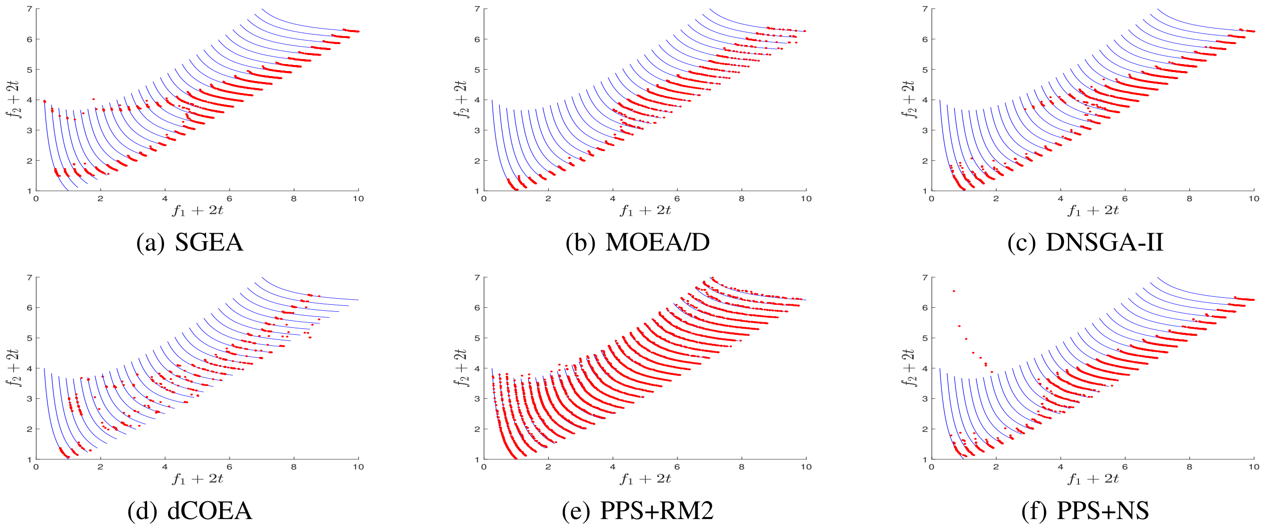
C. Algorithm Comparison on SDP Test Suite

Six algorithms are compared to study their strengths and weaknesses on the proposed SDP test suite. Both MIGD and MHVD values of these algorithms are detailed in Table S2 and Table S3 of the supplementary material, which are roughly consistent with each other. For simplicity, we only present in Table III the ranking of six algorithms for each SDP problem based on the MIGD measure. The ranking is calculated as follows. For every SDP problem with M ($M \in \{2,3,5\}$) objectives, each algorithm is compared with the rest, according to MIGD, using the Wilcoxon rank-sum test at the 0.05 significance level with the standard Bonferroni correction [1]. The algorithms are then ranked according to the total number of wins over different M values. Algorithms are allowed to have the same rank if there is a tie. This procedure applies to all the SDP problems. Finally, the six algorithms are also assigned a unique rank (the last column in Table III) for their performance on average on the SDP test suite, which is done by sorting the sum of individual ranks for each SDP problem.

The ranking table shows that SGEA, dCOEA, and PPS+RM2 are the top three best algorithms for SDP. It suggests that SGEA is most robust although it wins less than dCOEA over 15 SDP problems. dCOEA is in the first place for 8 problems but it obtains poor ranks for the other problms, making it the second in the average ranking.

dCOEA has the best performance on SDP1 and SDP3, showing it has good ability to track unpredictable PS moves (SDP1) and handle time-varying multimodality (SDP3). For SDP2, the time-dependent stretch of the PF range and variable dependencies present a big challenge to all the algorithm except PPS+RM2, as evidenced in Fig. 2. The outperformance of PPS+RM2 can be probably attributed to the regularity model (RM2), which has an advantage of cracking variable linkages. The suspicion is also supported by the performance difference between PPS+RM2 and PPS+NS on other SDP cases, e.g., SDP4-5. This means RM2 sometimes may play a more important role in the PPS+RM2 algorithm than PPS for handling dynamics.

In addition to variable-linkage, SDP4 and SDP5 also have dynamics on the number of local knee regions and the size of the PF manifold, respectively. dCOEA seems more sensitive to these dynamics and thus not an ideal solver for these problems compared to PPS+RM2 or SGEA.



(a) SGEA

(b) MOEA/D

(c) DNSGA-II

(d) dCOEA

(e) PPS+RM2

(f) PPS+NS

Fig. 2. PF approximations of six algorithms for SDP2.

SDP6-7 take into account the detectability of environmental changes, in addition to multimodality. DNSGA-II and dCOEA are top solvers for these two problems, respectively. It can be also noticed that PPS+RM2 is least suited to SDP7. We use MDT values, presented in table IV, to help us analyse the key challenge in SDP6-7. It is clear that all the six algorithms require multiple reevaluations for change detection. Despite slow change detection performance, SGEA has good MIGD values for both of the problems, implying that SGEA benefits greatly from its change reaction mechanism. PPS+RM2 also has relatively large MDT values, which can be an explanation for its poor performance on SDP6. One the other hand, PPS+RM2 detects changes promptly for SDP7, as evidenced by very small MDT values. However, its large MIGD values imply that PPS+RM2 struggles to track the moving global optima in a multimodal landscape. It is worth mentioning that SDP6-7 are specialised in testing algorithms' detectability of changes, a characteristic that existing test suites do not have. Such speciality is demonstrated by the visualization of PF approximations of SDP6-7 in Fig. S16 and Fig. S16 of the supplementary material, from which it is clear to see that a failure to detect changes leads to the loss of PF tracking.

The time-varying and non-simply connected PF of SDP8 influences MOEA/D most. This is due to the waste of weight vectors in MOEA/D that pass through PF holes. Thus, SDP8 is a good adversarial example for decomposition-based DMOP solvers.

Disconnected DMOPs are challenging. This applies especially to dCOEA for SDP9-10. For disconnected DMOPs, algorithms may have difficulty in finding the whole disconnected PF segments. Taking SDP10 as an example, we plot the PF approximations obtained by SGEA in Fig. 3, for the first four time steps. As seen, SGEA struggles to approximate the leftmost PF segment. There is a piece of PF segment for $f_1 \in [0.8, 1]$ at $t = 0$, but it disappears at t=0.1 and 0.2. Then, it is present at $t = 0.3$ and SGEA fails to identify its appearance. The dynamics of this problem cause SGEA to lose diversity when certain PF regions disappear suddenly and appear after several environmental changes. Other optimisers experience similar difficulties.

SDP11 has a time-varying PF segment that is difficult to approximate. This dynamic affects considerably the performance of MOEA/D and DNSGA-II. dCOEA is more robust than the other algorithms for this problem. The variable-based multipopulation in dCOEA renders a good performance of tracking the movement of the search-unfavoured PF segment. The multipopulation strategy in dCOEA is further shown to be a big advantage for solving SDP12 with a changing number of variables. PPS-based algorithms also have good performance for these two problems.

SDP13-15 have very simple PSs and satisfy the condition of PPS that PSs should be similar in two consecutive environments. However, the two PPS-based algorithms show poor performance on these problems, demonstrating that dynamics in the number of objectives and degeneration affect greatly PPS-based algorithms. In contrast, dCOEA and SGEA are more suited to these kind of dynamics.

Unsurprisingly, MOEA/D performs poorly on the SDP problems. This means that its fast convergence is no longer an advantage in handling dynamic scenarios like SDP. While fast convergence is desirable, it is more important to have a sound reaction response strategy when handling DMOPs.

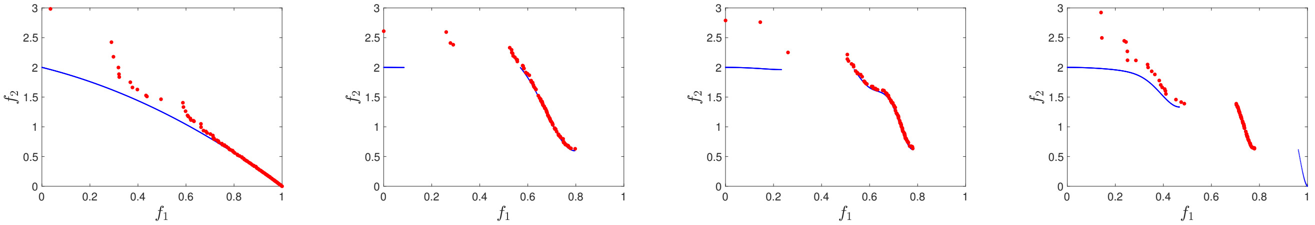
We display the averaged IGD evolution curves of the algorithms for some selected 2-objective SDP problems in Fig. 4 to illustrate the difficulty of the underlying dynamics. The IGD curves and PF approximations for other SDP instances can be found in Section IX, supplementary material. Clearly, the plots demonstrate that the SDP problems are able to identify the strengths and weaknesses of the algorithms.

D. Advantages of SDP Features

We empirically compare DMOPs with and without SDP features in order to understand how advantageous SDP is over existing test problems for testing algorithms. To do so, we created another SDP variant, named vSDP, by deactivating important SDP features and adding existing popular features to the original SDP1-14 (changes detailed in Table S4 of the supplementary material). vSDP1-14 was compared against

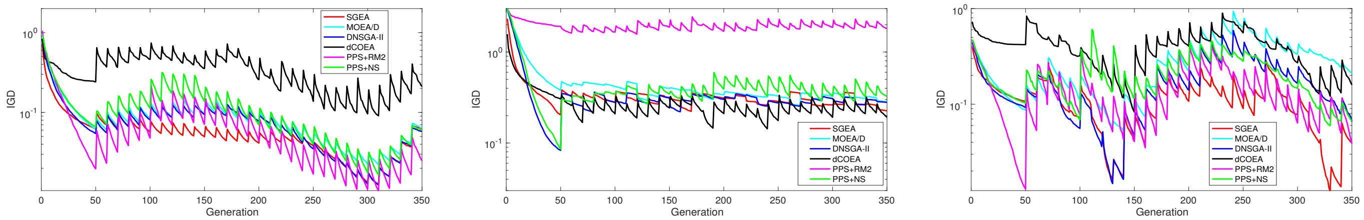
TABLE IV
MEAN AND STANDARD DEVIATION VALUES OF MDT OBTAINED BY SIX ALGORITHMS FOR SDP6-7 (BEST VALUES ARE HIGHLIGHTED IN BOLDFACE)

Prob.	M	SGEA	MOEA/D	DNSGA-II	dCOEA	PPS+RM2	PPS+NS
SDP6	2	9.8350E-2(4.6869E-2)	6.6128E-2(2.6956E-2)	7.9680E-2(2.5718E-2)	9.6768E-2(2.8321E-2)	8.9680E-2(1.7084E-2)	**5.7929E-2(2.0722E-2)**
	3	8.9175E-2(1.8792E-2)	5.7239E-2(1.2390E-2)	5.3316E-2(1.7277E-2)	**5.2054E-2(1.1462E-2)**	5.4579E-2(1.8596E-2)	5.4697E-2(1.4627E-2)
	5	8.5354E-2(2.9489E-2)	3.8939E-2(1.3097E-2)	**3.1279E-2(1.1426E-2)**	3.7071E-2(1.6682E-2)	7.5471E-2(1.7673E-2)	3.3872E-2(8.5727E-3)
SDP7	2	3.0690E-2(2.0408E-2)	2.2896E-2(2.1634E-2)	1.2795E-2(8.5305E-3)	2.2222E-3(4.1459E-3)	**1.8350E-3(4.9067E-3)**	1.5825E-2(1.7273E-2)
	3	3.0976E-2(2.1158E-2)	3.0741E-2(2.0845E-2)	3.9815E-2(2.5565E-2)	7.6936E-3(9.9103E-3)	**2.9125E-3(2.2587E-4)**	5.6869E-2(2.5704E-2)
	5	3.5354E-2(2.2739E-2)	2.6364E-2(2.0363E-2)	**8.9226E-4(2.2100E-3)**	4.7643E-3(5.7931E-3)	2.9293E-3(7.3574E-3)	1.0943E-3(9.1315E-4)



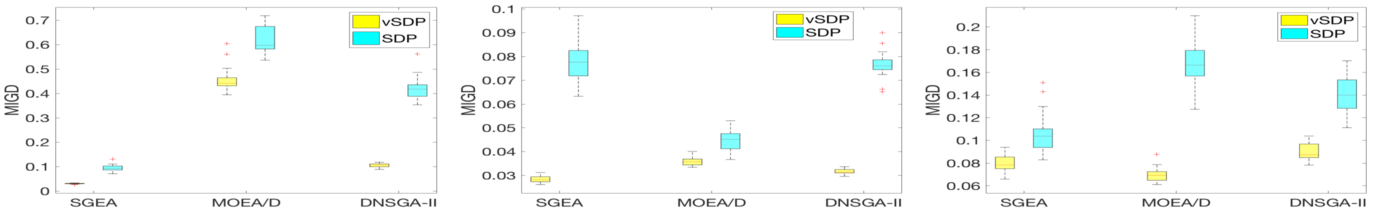
(a) $t = 0$ (b) $t = 0.1$ (c) $t = 0.2$ (d) $t = 0.3$

Fig. 3. PF approximations of SGEA for SDP10 from $t = 0$ to $t = 0.3$.



(a) SDP4 (b) SDP7 (c) SDP10

Fig. 4. Evolution curves of averaged IGD for selected problems.



(a) vSDP1 vs SDP1 (b) vSDP5 vs SDP5 (c) vSDP9 vs SDP9

Fig. 5. Comparison between problems with and without SDP features.

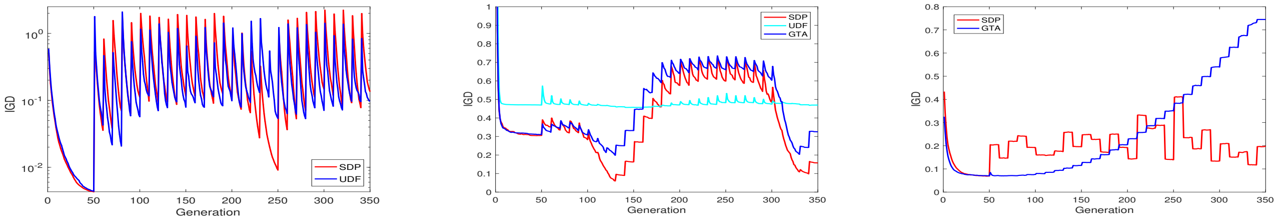
SDP1-14 based on their MIGD values obtained by three algorithms.

Fig. 5 shows the boxplots of MIGD values for the comparison between some selected SDP variants (please refer to Fig. S24 of the supplementary material for more comparisons). As seen, the SDP features are more challenging than existing features and they help to discriminate algorithms. For example, the three algorithms perform similarly on existing features in vSDP5 and vSDP9. In contrast, they obtain significantly different results on the SDP features for the corresponding problems. In addition, some SDP features can reveal the sensitivity of algorithms. It is shown that DNSGA-II is highly sensitive to unpredictability in SDP1 whereas SGEA and MOEA/D not. MOEA/D is more robust than SGEA and DNSG-II to PF/expansion/shrinkage captured by SDP5. Thus, SDP has a great advantage in discriminating algorithms and understanding the sensitivity of algorithms when used.

E. Comparison with Other Test Problems

Here, we would like to make a comparison between SDP problems and some widely-used test problems. Three SDP problems are chosen to compare with three FDA [10] problems and two dMOP [14] problems, for which the number of variables was set to the same as SDP. The comparison includes both 2-objective and 3-objective scenarios (the MHVD metric is reported in Table S5, supplementary material). It shows that the SDP problems are more challenging than the FDA and dMOP ones. The MHVD values for SDP problems are over an order of magnitude larger than most FDA or dMOP problems. Some algorithms that work well on existing problems experience performance deterioration when tested on SDP problems.

We noticed that some existing test suites, i.e., UDF [3] and GTA [11], also recognise the importance of several SDP features, including variable linkage, degeneration, and unpredictability. Thus, we compared SDP against the test suites with these features (settings for comparison are detailed in Section XII of the supplementary material), and the comparison is illustrated in Fig. 6. It is observed from the



(a) unpredictability

(b) variable-linkage

(c) degeneration

Fig. 6. Average IGD evolution curves obtained by SGEA for different predictability, variable-linkage, and degeneration schemes.

figure that the unpredictability of SDP presents more intense changes to the algorithm SGEA than that of UDF for most of time steps (when spikes appear). The difference is due to that variables in SDP have different magnitudes of random variations when an environmental change occurs, which is not the case with UDF. In Fig. 6(b), variable linkages in SDP and GTA induce similar impacts on SGEA. In contrast, variable linkages in UDF seem to overwhelmingly outweigh dynamics and therefore make it difficult to analyse the effect of the underlying dynamics. In Fig. 6(c), SDP-based degeneration shows more diverse dynamics variations compared to GTA. The IGD values for GTA-based degeneration keeps increasing after the first environmental change. A further analysis on problem formalism reveals that degeneration in GTA occurs very rarely, only when the PF surface becomes a curve. It is the changing size of the PF in GTA that causes the deterioration of SGEA over time. This GTA feature is different from degeneration in SDP and thus is not effective to be used for studying dynamic degeneration.

We also studied other important characteristics, i.e., spread of solutions, time-changing concavity-convexity, and disconnected PF components. Comparison of SDP and existing benchmark functions (mainly FDA [10], dMOP [14], HE [16]) is detailed in Section XII, supplementary material.

F. SDP as Adversarial Problems

SDP has another use for adversarial examples to optimisation solvers. It is shown in previous sections that the existing solvers encounter vast difficulties in a number of real-life dynamic features captured by SDP. We highlight these features that are poorly understood in the literature: time-dependent multimodality (SDP2-3), time-dependent knees (SDP3-4), deceptive changes (SDP6-7), dynamics in disconnectivity (SDP8-10), dynamics in objective and/or variable scalability (SDP12-13), and dynamic degeneracy (SDP14-15). These SDP adversarial examples facilitate a tool not only to better understand the robustness of solvers and make the right decision on selection of solvers for problems in question, but also to develop advanced techniques against them.

V. Conclusion

Dynamic multiojective optimisation has received growing attention over the last 10 years, yet the corresponding test environments facilitating algorithm analysis have not been well understood. While this paper helps to provide a better understanding of DMOPs, additional issues are identified after a wide review of existing test problems. These issues limit the assessment of EAs in dynamic environments.

Having realised the limitations of existing DMOPs, we highlighted a diverse set of dynamics and features that a good test suite should have. We also developed a SDP test suite to include desirable dynamics that have been rarely considered in the DMO literature but occur frequently in real life. Our experiments showed that the SDP test suite presents different degrees of difficulty to the algorithms tested and is able to identify the strengths and weaknesses of these algorithms. Furthermore, it was demonstrated that the SDP problems can test EAs in ways that commonly used DMOPs can not. SDP can be also used as adversarial examples for solvers.

It is expected that more advanced algorithms will be available for DMO, and a wider range of DMOPs as well. The SDP test suite could be adjusted or modified to create appropriate scenarios for other types or test environments, e.g. constrained and/or noisy DMOPs.

References

[1] H. Abdi, “Bonferroni and Šidák corrections for multiple comparison,” In: *Encyclopedia of Measurement and Statistics*, N. J. Salkind, Eds. Thousand Oaks, CA: Sage, 2007.

[2] M. B. Abello and Z. Michalewicz, “Multiobjective resource-constrained project scheduling with a time-varying number of tasks,” *The Scientific World Journal*, vol. 2014, Article ID 420101, pp. 1–35, 2014.

[3] 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.

[4] J. Butans, *Addressing Real-time Control Problems in Complex Environments Using Dynamic Multi-objective Evolutionary Approaches*, PhD thesis, School of Applied Sciences, Cranfield University, 2011.

[5] R. Chen, K. Li, and X. Yao, “Dynamic multi-objectives optimization with a changing number of objectives,” *IEEE Trans. Evolut. Comput.*, 2017, in press. DOI: 10.1109/TEVC.2017.2669638

[6] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester: John Wiley and Sons, 2001.

[7] 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.

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

[9] J. Eaton, S. Yang, and M. Gongora, “Ant colony optimization for simulated dynamic multi-objective railway junction rescheduling,” *IEEE Trans. Intelligent Transportation Systems*, vol. 18, no. 11, pp. 2980–2992, 2017.

[10] 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.

[11] 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.

[12] S. Ghosh, et al., “Verifying controllers against adversarial examples with Bayesian optimization,” https://arxiv.org/pdf/1802.08678.pdf

[13] M. Giuliani, J. D. Herman, A. Castelletti, and P. Reed, “Many-objective reservoir policy identification and refinement to reduce policy inertia and myopia in water management,” *Water Resour. Res.*, vol. 50, pp. 3355–3377, 2014.

[14] 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.

[15] S. Guan, Q. Chen, and W. Mo, “Evolving dynamic multi-objective optimization problems with objective replacement,” *Artif. Intell. Rev.*, vol. 23, no. 3, pp. 267–293, 2005.

[16] 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.

[17] L. Huang, I. H. Sub, and 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.

[18] S. Huband, P. Hingston, L. Barone, and 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.

[19] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, “Performance of decomposition-based many-objective algorithms strongly depends on Pareto front shapes,” *IEEE Trans. Evol. Comput.*, 2017, vol. 21, no. 2, pp. 169–190.

[20] 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.

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

[22] 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.

[23] S. Jiang and S. Yang, “A strength Pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimisation,” *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 329–346, 2017.

[24] S. Jiang, S. Yang, Y. Wang, and X. Liu, “Scalarizing functions in decomposition-based multiobjective evolutionary algorithms,” *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 296–313, 2017.

[25] Y. Jin and B. Sendhoff, “Constructing dynamic optimization test problems using the multi-objective optimization concept,” in *EvoWorkshops 2004: Appl. Evol. Comput.*, 2004, pp. 525–536.

[26] W. Koo, C. Goh, and K. Tan, “A predictive gradient strategy for multiobjective evolutionary algorithms in a fast changing environment,” *Memetic Comput.*, vol. 2, no. 2, pp. 87–110, 2010.

[27] V. Lau, et al, “A multidisciplinary design optimization tool for spacecraft equipment layout conception. *J. Aerosp. Technol. Manag.*, vol. 6, no. 4, pp. 431–446, 2014.

[28] H. Li, M. Gong, D. Meng, and Q. Miao, “Multi-objective self-paced learning,” in *Proc. 13th AAAI Conf. Art. Intell.*, 2016, pp. 1802–1808.

[29] M. Li, S. Yang, X. Liu, “Shift-based density estimation for pareto-based algorithms in many-objective optimization,” *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 348–365, Jun. 2014.

[30] H. Li and 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.

[31] H. Liu, L. Chen, K. Deb, and E. D. Goodman, “Investigating the effect of imbalance between convergence and diversity in evolutionary multiobjective algorithms,” *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 408–425, 2017.

[32] R. J. Lygoe, M. Cary, P. J. Fleming, “A real-world application of a many-objective optimisation complexity reduction process,” in *Int. Conf. Evolutionary Multi-Criterion Optimization (EMO)*, 2013, pp. 641–655.

[33] J. Mehnen, G. Rudolph, and T. Wagner, “Evolutionary optimization of dynamic multiobjective functions,” Universitat Dortmund, Dortmund, Germany, Tech. Rep. FI-204/06, 2006.

[34] A. Muruganantham, K. C. Tan, and P. Vadakkepat, “Evolutionary dynamic multiobjective optimization via Kalman filter prediction,” *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2862–2873, 2016.

[35] T. T. Nguyen, S. Yang, and Juergen Branke, “Evolutionary dynamic optimization: A survey of the state of the art,” *Swarm Evol. Comput.*, vol. 6, pp. 1–24, 2012.

[36] T. T. Nguyen, Z. Yang, and S. Bonsall, “Dynamic time-linkage problems – the challenges,” in *2012 IEEE RIVF Int. Conf. Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF)*, 2012, pp. 1–6.

[37] P. Nimmegeers, D. Telen, F. Logist, and J. V. Impe, “Dynamic optimization of biological networks under parametric uncertainty,” *BMC Systems Biology*, vol. 10, no. 86, pp. 1–20, 2016.

[38] F. Pieri and A. Cilea, “A fast multiobjective optimization strategy for single-axis electromagnetic MOEMS micromirrors,” *Micromachines*, vol. 9, no. 1. pp. 1-14, 2018.

[39] F. Rahmani, K. Behzadian, and A. Ardeshir, “Rehabilitation of a water distribution system using sequential multi-objective optimization models,” *J. Water Resources Planning and Management*, vol. 142, no. 5, article No. C4015003, 2015.

[40] A. Ramirez-Arias, F. Rodriguez, J. L. Guzman, and M. Berenguel, “Multiobjective hierarchical control architecture for greenhouse crop growth,” *Automatica*, vol. 48, no. 3, pp. 490–498, 2012.

[41] H. Richter, “Change detection in dynamic fitness landscapes: An immunological approach, ” in *2009 World Congress on Nature & Biologically Inspired Computing*, 2009, pp. 719–924.

[42] P. Rohlfshagen and X. Yao, “Dynamic combinatorial optimisation problems: an analysis of the subset sum problem,” *Soft Comput.*, vol. 15, no. 9, pp. 1723–1734, 2015.

[43] S. Sahmoud and H. R. Topcuoglu, “Sensor-based change detection schemes for dynamic multi-objective optimization problems,” in *2016 IEEE Symposium Series on Computational Intelligence (SSCI2016)*, 2017. DOI: 10.1109/SSCI.2016.7849963

[44] X. Shen and X. Yao, “Mathematical modeling and multi-objective evolutionary algorithms applied to dynamic flexible job shop scheduling problems,” *Inform. Sci.*, vol. 298, pp. 198–224, 2015.

[45] H. K. Singh, A. Isaacs, and T. Ray, “A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems,” *IEEE Trans. Evol. Comput.*, vol. 15, no. 4, pp. 539–556, 2011.

[46] J. Xiong, Z. Zhou, K. Tian, T. Liao, and J. Shi, “A multi-objective approach for weapon selection and planning problems in dynamic environments.” *Journal of Industrial & Management Optimization*, vol. 13, no. 3, pp. 1189–1211, 2017.

[47] S. Yang and C. Li, “A clustering particle swarm optimizer for dynamic optimization,” *IEEE Trans. Evol. Comput.*, vol. 14, no. 6, pp. 959–974, 2010.

[48] 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.

[49] 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.

[50] 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.

[51] Z. Zhang, M. Liao, and L. Wang, “Immune optimization approach for dynamic constrained multi-objective multimodal optimization problems,” *American Journal of Operations Research*, vol. 2, no. 2, pp. 193–202, 2012.

[52] 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.

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

[54] A. Zhou and Q. Zhang, “Are all the subproblems equally important? Resource allocation in decomposition-based multiobjective evolutionary algorithms,” *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 52–64, 2016.

[55] 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.

[56] H. Zille and S. Mostaghim, “Using ϵ-dominance for hidden and degenerated Pareto-fronts,” in *2015 IEEE Symposium Series on Computational Intelligence*, 2015, pp. 845–852.

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

[58] 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,” *Appl. Soft Comput.*, vol. 61, pp. 806–818, 2017.