Benchmarks for Dynamic Multi-Objective Optimisation Algorithms

MARDÉ HELBIG, CSIR, Meraka Institute and University of Pretoria, Department of Computer Science
ANDRIES P. ENGELBRECHT, University of Pretoria, Department of Computer Science

Algorithms that solve Dynamic Multi-Objective Optimisation Problems (DMOOPs) should be tested on benchmark functions to determine whether the algorithm can overcome specific difficulties that can occur in real-world problems. However, for Dynamic Multi-Objective Optimisation (DMOO), no standard benchmark functions are used. A number of DMOOPs have been proposed in recent years. However, no comprehensive overview of DMOOPs exist in the literature. Therefore, choosing which benchmark functions to use is not a trivial task. This article seeks to address this gap in the DMOO literature by providing a comprehensive overview of proposed DMOOPs, and proposing characteristics that an ideal DMOO benchmark function suite should exhibit. In addition, DMOOPs are proposed for each characteristic. Shortcomings of current DMOOPs that do not address certain characteristics of an ideal benchmark suite are highlighted. These identified shortcomings are addressed by proposing new DMOO benchmark functions with complicated Pareto-Optimal Sets (POSs), and approaches to develop DMOOPs with either an isolated or deceptive Pareto-Optimal Front (POF). In addition, DMOO application areas and real-world DMOOPs are discussed.

Categories and Subject Descriptors: I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic search*

General Terms: Measurement, Performance

Additional Key Words and Phrases: Dynamic multi-objective optimisation, benchmark functions, ideal benchmark function suite, complex Pareto-optimal set, isolated Pareto-optimal front, deceptive Pareto-optimal front

ACM Reference Format:
Mardé Helbig and Andries P. Engelbrecht. 2014. Benchmarks for dynamic multi-objective optimisation algorithms. ACM Comput. Surv. 46, 3, Article 37 (January 2014), 39 pages.
DOI: http://dx.doi.org/10.1145/2517649

1. INTRODUCTION

Dynamic Multi-Objective Optimisation Problems (DMOOPS) are Multi-Objective Optimisation Problems (MOOPs) where either the objective functions or the constraints change over time. This article focuses on unconstrained DMOOPs with objectives that change over time and with static boundary constraints—that is, bounded constraint DMOOPs. Furthermore, it should be noted that this article does not focus on MOOPs with noise [Goh et al. 2010; Chia et al. 2012].

In order to determine whether an algorithm can solve DMOOPs efficiently, it should be evaluated on DMOOPs that test the ability of the algorithm to overcome certain difficulties, such as tracking a Pareto-Optimal Front (POF) that changes from convex

Authors' addresses: M. Helbig, Meraka Institute, CSIR, Brummeria, South Africa, and Computer Science Department, University of Pretoria, Pretoria, South Africa; email: mhelbig@csir.co.za; A. P. Engelbrecht, Computer Science Department, University of Pretoria, Pretoria, South Africa; email: engel@cs.up.ac.za.

to concave over time or finding a diverse set of solutions where the density of solutions changes over time. Such functions are referred to as benchmark functions.

The set of benchmark functions chosen for a comparative study influence the results and effectiveness of the study. Therefore, the benchmark functions should be chosen with care. However, one of the main problems in the field of DMOO is a lack of standard benchmark functions. Therefore, selecting which benchmark functions to use is not a trivial task. In addition, currently no comprehensive overview of DMOOPs is presented in the literature. This article seeks to address this problem by:

- Providing a comprehensive overview of DMOOPs that were suggested in the literature
- Suggesting an ideal set of DMOO benchmark functions

This will enable a uniform comparison of Dynamic Multi-Objective Optimisation Algorithms (DMOAs). In order to achieve these two main objectives, the following subobjectives were identified:

- Investigating the current DMOOPs presented in the literature to establish whether they are efficiently testing the performance of DMOO algorithms
- Identifying shortcomings of current DMOOPs
- Addressing the identified shortcomings of current DMOOPs by:
  - Introducing an approach to develop DMOOPs with an isolated POF
  - Introducing an approach to develop DMOOPs with a deceptive POF
  - Introducing new DMOOPs with complicated Pareto-Optimal Sets (POSs)

The rest of the article is outlined as follows. Formal definitions of concepts that are required as background for this article are provided in Section 2. Section 3 discusses characteristics proposed for an ideal set of static Multi-Objective Optimisation (MOO) and Dynamic Single-Objective Optimisation (DSOO) benchmark functions. In addition, the characterisation of DMOOPs are discussed. A comprehensive overview of DMOOPs proposed in the literature are provided in Section 4. Section 5 highlights shortcomings of current DMOOPs. To address the identified shortcomings, new DMOOPs are introduced. Section 6 highlights observations made in a study that compares the performance of five DMOAs on various DMOOPs. Characteristics that an ideal DMOO benchmark function suite should have are suggested in Section 7, taking into account new advancements in the MOO literature. Furthermore, a set of DMOOPs are suggested for each identified characteristic. Section 8 highlights real-world application areas of DMOO and discusses four real-world DMOOPs. Finally, the conclusions are discussed in Section 9.

2. DEFINITIONS

This section provides definitions that are required as background for the rest of the article. Definitions with regards to MOO and DMOO are provided in Sections 2.1 and 2.2, respectively.

2.1. Multi-Objective Optimisation

The various objectives of a MOOP are normally in conflict with one another—that is, improvement in one objective leads to a worse solution for at least one other objective. Therefore, the definition of optimality that is used for Single-Objective Optimisation Problems (SOOPs) has to be adjusted when solving MOOPs.

For MOOPs, when one decision vector dominates another, the dominating decision vector is considered as a better decision vector.

Let the n_x-dimensional search space (also referred to as the *decision space*) be represented by $S \subseteq \mathbb{R}^{n_x}$ and the feasible space represented by $F \subseteq S$, where $F = S$ for unconstrained optimisation problems. Let $\mathbf{x} = (x_1, x_2, \ldots, x_{n_x}) \in S$ represent a vector of the decision variables (i.e., the *decision vector*), and let a single objective function be defined as $f_k : \mathbb{R}^{n_x} \to \mathbb{R}$. Then, $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_{n_k}(\mathbf{x})) \in O \subseteq \mathbb{R}^{n_k}$ represents an *objective vector* containing n_k objective function evaluations, and O is the *objective space*.

Using the previous notation, and assuming minimisation, decision vector domination is defined as follows:

Definition 1. *Decision Vector Domination*: Let f_k be an objective function. Then, a decision vector $\mathbf{x}_1$ dominates another decision vector $\mathbf{x}_2$, denoted by $\mathbf{x}_1 \prec \mathbf{x}_2$, if and only if:

—$\mathbf{x}_1$ is at least as good as $\mathbf{x}_2$ for all the objectives (i.e., $f_k(\mathbf{x}_1) \leq f_k(\mathbf{x}_2),\ \forall k = 1, \ldots, n_k$); and
—$\mathbf{x}_1$ is strictly better than $\mathbf{x}_2$ for at least one objective (i.e., $\exists i = 1, \ldots, n_k : f_i(\mathbf{x_1}) < f_i(\mathbf{x_2})$).

The best decision vectors are called Pareto-optimal, defined as follows:

Definition 2. *Pareto-Optimal*: A decision vector $\mathbf{x}^*$ is Pareto-optimal if there does not exist a decision vector $\mathbf{x} \neq \mathbf{x}^* \in F$ that dominates it—that is, $\nexists k : f_k(\mathbf{x}) \prec f_k(\mathbf{x}^*)$. If $\mathbf{x}^*$ is Pareto-optimal, the objective vector, $\mathbf{f}(\mathbf{x}^*)$, is also Pareto-optimal.

The set of all Pareto-optimal decision vectors are referred to as the POS, defined as:

Definition 3. *Pareto-Optimal Set*: The POS, POS^*, is formed by the set of all Pareto-optimal decision vectors—that is:

$$POS^* = \{\mathbf{x}^* \in F \mid \nexists \mathbf{x} \in F : \mathbf{x} \prec \mathbf{x}^*\} \tag{1}$$

The POS contains the best trade-off solutions for the MOOP. The set of corresponding objective vectors are the POF or Pareto front, which is defined as follows:

Definition 4. *Pareto-Optimal Front*: For the objective vector $\mathbf{f}(\mathbf{x})$ and the POS POS^*, the POF, $POF^* \subseteq O$ is defined as:

$$POF^* = \{\mathbf{f} = (f_1(\mathbf{x}^*), f_2(\mathbf{x}^*), \ldots, f_{n_k m}(\mathbf{x}^*)) \mid \mathbf{x}^* \in POS^*\} \tag{2}$$

2.2. Dynamic Multi-Objective Optimisation

Using the notation defined in Section 2.1, an unconstrained DMOOP can be mathematically defined as:

$$\begin{aligned} &\textit{minimise}: && \mathbf{f}(\mathbf{x}, \mathbf{W}(t)) \\ &\textit{subject to}: && \mathbf{x} \in [\mathbf{x}_{min}, \mathbf{x}_{max}]^{n_x}, \end{aligned} \tag{3}$$

where $\mathbf{W}(t)$ is a matrix of time-dependent control parameters of an objective function at time t, $\mathbf{W}(t) = (\mathbf{w}_1(t), \ldots, \mathbf{w}_{n_m}(t))$, n_x is the number of decision variables, $\mathbf{x} = (x_1, \ldots, x_{n_x}) \in \mathbb{R}^{n_x}$, and $\mathbf{x} \in [\mathbf{x}_{min}, \mathbf{x}_{max}]^{n_x}$ refers to the boundary constraints.

In order to solve a DMOOP the goal of an algorithm is to track the POF over time (i.e., for each timestep) to find:

$$POF^*(t) = \{\mathbf{f}(t) = (f_1(\mathbf{x}^*, \mathbf{w}_1(t)), f_2(\mathbf{x}^*, \mathbf{w}_2(t)), \ldots, f_{n_k}(\mathbf{x}^*, \mathbf{w}_{n_k}(t))) \mid \mathbf{x}^* \in POS^*(t)\} \tag{4}$$

3. BACKGROUND

This section discusses characteristics that were proposed for an ideal set of static MOOPs and Dynamic Single-Objective Optimisation Problems (DSOOPs). Furthermore, the characterisation of DMOOPs are discussed.

3.1. Ideal MOO Benchmark Function Characteristics

The ZDT [Deb 1999; Zitzler et al. 2000] and DTLZ [Deb et al. 2002] MOOP suites were constructed by Deb et al. in such a way that the benchmark functions are [Deb et al. 2002]:

- —Easy to construct
- —Scalable in terms of both the number of decision variables and the number of objective functions
- —Producing a POF with a known shape and location and that is easy to understand
- —Hindering an algorithm to produce a good distribution of solutions and/or converging to the true POF

According to Deb [1999], an algorithm can be hindered to converge to the true POF when a benchmark function is multimodal, has an isolated optimum, is deceptive, or contains noise. Deceptive functions have at least two optima in the search space, with the search space favouring the deceptive optimum. For a DMOOP the deceptive optimum is a local POF and not the true global POF. A multimodal function has many POFs (local and global), and therefore a DMOO algorithm can become stuck in a local POF. An objective function where an open subset of decision variable values maps to a single value is referred to as an objective function with—that is, regions where small perturbations of the decision variable values do not change the objective function value. The flat regions' lack of gradient information may cause an algorithm to struggle to converge to the optima. An *isolated* POF occurs if the majority of the fitness landscape is fairly flat and no useful information is provided with regards to the location of the POF. Therefore, if a DMOOP has an isolated POF, a DMOA may struggle to converge towards the POF. It should be noted that if the majority of the fitness landscape is not fairly flat and therefore the POF is not completely isolated from the rest of the search space, an algorithm may still struggle to converge towards the POF if the density of solutions close to the POF is significantly less than in the rest of the search space.

An algorithm may struggle to find a diverse set of solutions if the true POF has the following properties: convexity or nonconvexity in the POF, a discontinuous POF, and nonuniform spacing of solutions in the POS or POF [Deb 1999, 2004]. When a POF is convex, it may be difficult to solve the DMOOP by algorithms that assign a solution's fitness based on Pareto ranking—that is, the number of solutions that the solution dominates [Deb 1999]. Pareto ranking may cause bias towards certain portions of the POF that contain intermediate solutions. This occurs since this fitness assignment favours intermediate solutions that perform reasonably well with regards to all objective functions more than solutions that perform very well with regards to one objective and not so well with regards to the other objectives. If the POF has a set of disconnected continuous subregions, referred to as a discontinous POF, an algorithm may struggle to find solutions in all regions of the POF. However, even though an algorithm may find solutions within each region, solutions from certain subregions may be outranked (or dominated) when the solutions compete amongst each other (for a rank or for storage in the archive) and may therefore disappear from the nondominated solution set. In addition, an algorithm may struggle to find a diverse set of nondominated solutions if the POS or POF is not uniformly spaced [Deb 2004].

3.2. Ideal Dynamic SOO Benchmark Function Characteristics

If a DSOOP completely changes over time without any connection to a previous environment, an algorithm implementing a restart after a change will perform the best

[Branke 1999]. Therefore, DSOO benchmark functions should change in such a way over time that the new environment has a connection to a previous environment [Branke 1999]. Furthermore, according to Morrison and Jong [1999], a DSOO benchmark function generator should enable easy:

—changes to the landscape complexity to develop benchmark functions that are representative of real-world problems (in terms of their complexity).
—specification of the morphological characteristics and changes of the landscape (e.g., the peak location, shape and height).
—specification of the type of changes of the environment (e.g., recurrent, chaotic, large or small changes).
—representation of the environment mechanisms to ensure that the environment can be defined in an unambiguous manner.

In addition, the benchmark function should have a reasonable computational complexity [Morrison and Jong 1999]. According to Branke [1999], ideal characteristics of benchmark functions are in general tunable parameters and simplicity—that is, the function is easy to describe and analyse. However, although benchmark functions should be simple enough to gain a better understanding of the performance of an algorithm that is solving the DSOOP, at the same time the benchmark functions should be complex enough to represent real-world problems.

3.3. Characterisation of DMOO Benchmark Functions

One the first categorisations of DMOOPs was proposed by Farina et al. [2004], who categorised DMOOPs into four types, namely:

—Type I DMOOPs, where the POS changes over time but the POF remains unchanged
—Type II DMOOPs, where both the POS and the POF change over time
—Type III DMOOPs, where the POF changes over time but the POS remains unchanged
—Type IV DMOOPs, where a change occurs in the environment but both the POS and POF remain unchanged

Goh and Tan [2009c] characterised DMOOPs according to spatial and temporal features. Spatial features were divided into two categories, namely physical attributes and nonphysical attributes. Physical attributes refer to physical aspects, such as the POF or POS. Nonphysical attributes refer to the manner in which the physical attributes change. The categorisation of spatial features are [Goh and Tan 2009c]:

(1) Physical attributes:
  —The whole POS moves to a new location.
  —The shape of the POF changes or a part of the POF disappears.
  —The fitness landscape changes without affecting the POS or POF.
  —Random changes to the POS, POF and/or landscape.
(2) Nonphysical attributes:
  —Random changes to physical attributes.
  —Changes to physical attributes follow a fixed pattern, where past physical topologies may or may not be revisited again.
  —Periodic changes to physical attributes, where changes within a period may or may not follow a fixed pattern.

According to Goh and Tan [2009c], the temporal features of DMOOPs are as follows:

—No change occurs.
—A change occurs randomly.

—A change occurs at fixed intervals.
—A change occurs according to a predetermined schedule.
—A change occurs after a predefined condition is satisfied.

The following challenges are unique to DMOOPs [Goh and Tan 2009c]:

—A DMOOP does not have a single solution at a specific time, but has a set of solutions. Therefore, an algorithm has to track the changing POF over time.
—After a change in the environment occurred, any solution within the set of solutions (or all solutions) can become obsolete or invalid.
—Changes can occur with regards to both the shape of the POF and the distribution of the solutions within the POF. Therefore, both the decision variable space and objective space have to be considered when dealing with DMOOPs (refer to the cat egorisation of DMOOPs by Farina et al. [2004]).

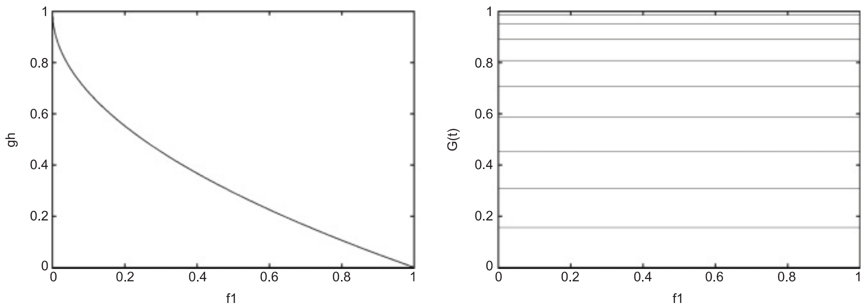
Furthermore, similar to algorithms solving DSOOPs, algorithms that solve DMOOPs have to be adapted to overcome diversity loss and outdated memory [Blackwell and Branke 2006]. Outdated memory occurs when the environment changes and the information that is currently stored is no longer valid and can even guide the search in the wrong direction. For evolutionary algorithms, this outdated information may include the individual's fitness and various solutions' ranks. For particle swarm optimisations (PSOs), this outdated information may include the particle's fitness: *pbest* solutions of the particles and the swarm's *gbest*. Diversity loss may occur when the algorithm is converging to a specific optimum. For example, with a PSO the *gbest* and the *pbest* of the particles will be close to the previous optimum and therefore the particles' velocities will be small. The smaller velocities may prevent the particles from tracking a changing optimum, especially if the new optimum is not in close proximity to the previous optimum. Therefore, the algorithm may get stuck in the previous optimum and be unable to search for new optima [Blackwell and Branke 2006].

4. DMOO BENCHMARK FUNCTIONS CURRENTLY USED

This section discusses benchmark functions that have been used in the DMOO literature to evaluate whether algorithms can efficiently solve DMOOPs. Due to space constraints, only POSs and POFs with different characteristics are illustrated in this section.

One of the first DMOOPs suggested in the literature was proposed by Tan et al. [2003] and is based on the DSOO Moving Peaks benchmark function generator [Branke 1999]. Guan et al. [2005] suggested creating DMOOPs by replacing objective functions with new objective functions over time. The advantage of Guan et al.'s approach is that the new objective function(s) can cause a severe change in the DMOOP, and by selecting the objective functions carefully, various types of changes can be incorporated into the DMOOP.

Recently, Wang and Li [2010] presented a DMOOP where the one subfunction of an objective function changes over time. When objective functions are changed over time, as in the approaches followed by Guan et al. [2005] and Wang and Li, the objective functions should be selected carefully to ensure that the resulting objective functions hinder the algorithm in finding the POF in various ways, as discussed in Section 3. Another approach was followed by Jin and Sendhoff [2004], where a two-objective DMOOP is constructed from a three-objective MOO function. The approach of Jin and Sendhoff has been used by various researchers [Li et al. 2007; Liu 2010; Liu and Wang 2006, 2007]. However, the adherence to the guidelines suggested by Deb [1999] by the benchmark functions suggested by Guan et al., Wang and Li, and Jin and Sendhoff will depend on the specific objective functions that are used.



(a) POF of FDA1 (b) POS of FDA1

Fig. 1. POF and POS of FDA1 with $n_t = 10$ and $\tau_t = 10$ for 1,000 iterations. POF remains static over time, but POS changes over time.

Based on the ZDT [Deb 1999; Zitzler et al. 2000] and DTLZ [Deb et al. 2002] functions, Farina et al. [2004] developed the first suite of DMOOPs, namely the FDA benchmark functions.

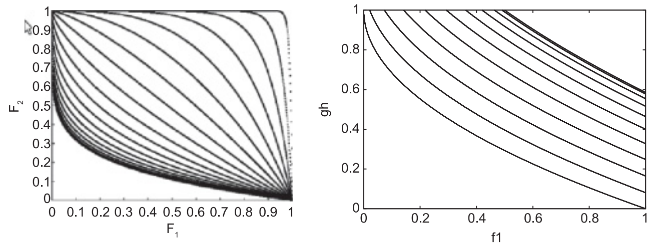
The DMOOPs of the FDA DMOOP suite are easy to construct, and the number of decision variables are easily scalable. FDA4 and FDA5 are constructed in such a way that they are easily scalable with regards to both the number of decision variables and the number of objective functions. The FDA benchmark functions are of Type I, II, and III DMOOPs, and the POF of these DMOOPs is either convex, nonconvex, or changes from convex to concave (or vice versa) over time. Therefore, the FDA DMOOP suite exhibits the characteristics that benchmark functions should have, as defined by Deb [1999].

The five FDA DMOOPs are defined as follows:

$$\text{FDA1} = \begin{cases} \textit{Minimise}: \mathbf{f}(\mathbf{x}, t) = (f_1(\mathbf{x}_{\mathsf{I}}), g(\mathbf{x}_{\mathsf{II}}, t) \cdot h(f_1(\mathbf{x}_{\mathsf{I}}), g(\mathbf{x}_{\mathsf{II}}, t))) \\ f_1(\mathbf{x}_{\mathsf{I}}) = x_1 \\ g(\mathbf{x}_{\mathsf{II}}, t) = 1 + \sum_{x_i \in \mathbf{x}_{\mathsf{II}}} (x_i - G(t))^2 \\ h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}} \\ \textit{where}: \\ G(t) = \sin(0.5\pi t), \quad t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_t} \right\rfloor \\ \mathsf{x}_{\mathsf{I}} \in [0, 1]; \quad \mathbf{x}_{\mathsf{II}} = (x_2, \ldots, x_n) \in [-1, 1]^{n-1}, \end{cases} \tag{5}$$

where τ, τ_t, and n_t refer to the current iteration, the frequency of change, and the severity of change, respectively. For FDA1, values in the decision variable space (POS) change over time, but the values in the objective space (POF) remain the same. Therefore, it is a Type I DMOOP. It has a convex POF with $POF = 1 - \sqrt{f_1}$, as illustrated in Figure 1(a). The POS is $x_i = G(t), \; \forall x_i \in \mathbf{x}_{\mathsf{II},}$ as illustrated in Figure 1(b). Appendix B discusses how to determine the POS and POF of a DMOOP.

$$\text{FDA2} = \begin{cases} \textit{Minimise}: f(\mathsf{x}, t) = (f_1(\mathbf{x}_{\mathsf{I}}), g(\mathbf{x}_{\mathsf{II}}) \cdot h(\mathbf{x}_{\mathsf{III}}, f_1(\mathbf{x}_{\mathsf{I}}), g(\mathbf{x}_{\mathsf{II}}), t)) \\ f_1(\mathbf{x}_{\mathsf{I}}) = x_1 \\ g(\mathbf{x}_{\mathsf{II}}) = 1 + \sum_{x_i \in \mathbf{x}_{\mathsf{II}}} x_i^2 \\ h(\mathbf{x}_{\mathsf{III}}, f_1, g, t) = 1 - \left(\frac{f_1}{g}\right)^{H_2(t)} \\ \textit{where}: \\ H(t) = 0.75 + 0.75\sin(0.5\pi t), \quad t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_t} \right\rfloor \\ H_2(t) = \left(H(t) + \sum_{x_i \in \mathbf{x}_{\mathsf{III}}} (x_i - H(t))^2\right)^{-1} \\ \mathbf{x}_{\mathsf{I}} \in [0, 1]; \quad \mathbf{x}_{\mathsf{II}_i}, \mathbf{x}_{\mathsf{III}_i} \in [-1, 1] \end{cases} \tag{6}$$



(a) POF of FDA2. ©2004 IEEE. Reprinted with permission from Farina et al. [2004].

(b) POF of FDA3

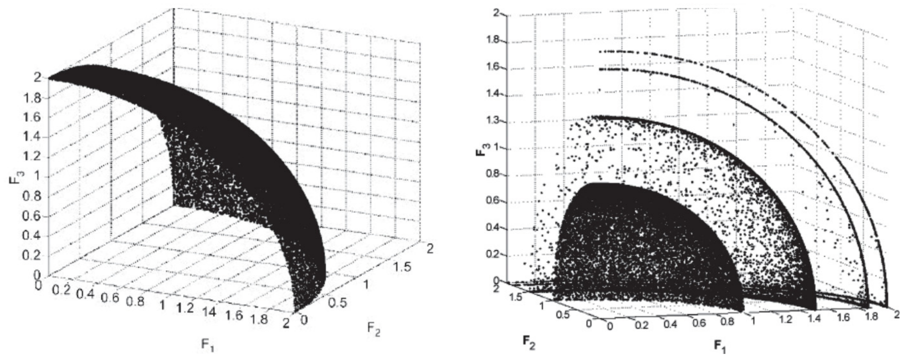
Fig. 2. POF of FDA2 and FDA3 with $n_t = 10$ and $\tau_t = 10$ for 1,000 iterations. POF of FDA2 changes in a cyclic manner over time, moving from the top line to bottom line for certain timesteps and from the bottom line to the top line for other timesteps. POF of FDA3 changes over time in a cyclic manner, moving either from the top line to the bottom line for certain timesteps or from the bottom line to the top line for the other timesteps.

FDA2 has a POF that changes from convex to concave and vice versa. It is a Type II DMOOP, as both the POS and POF change over time. For FDA2, $POF = 1 - f_1^{H(t)^{-1}}$, as illustrated in Figure 2(a). The POS of FDA2 is $x_i = 0,\ \forall x_i \in \mathbf{x}_{\mathsf{II}}$ and $x_i = H(t),\ \forall x_i \in \mathbf{x}_{\mathsf{III}}$. It should be noted that many researchers refer to FDA2 as a Type III DMOOP due to an error in the DMOOP definition in Farina et al. [2004]. However, before the definition of FDA2 in Farina et al., the explanation of the effect of the h function on the DMOOP states that the h function in FDA2 causes the POF to only change through a change in $\mathbf{x}_{III}$ and that FDA2 is therefore a Type II DMOOP.

$$\text{FDA3} = \begin{cases} \mathit{Minimise}\colon \mathbf{f}(\mathbf{x},t) = (f_1(\mathbf{x}_{\mathsf{I}},t), g(\mathbf{x}_{\mathsf{II}},t)\cdot h(f_1(\mathbf{x}_{\mathsf{I}}), g(\mathbf{x}_{\mathsf{II}},t))) \\ f_1(\mathbf{x}_{\mathsf{I}},t) = \sum_{x_i \in \mathbf{x}_{\mathbf{I}}} x_i^{F(t)} \\ g(\mathbf{x}_{\mathsf{II}},t) = 1 + G(t) + \sum_{x_i \in \mathbf{x}_{\mathsf{II}}} (x_i - G(t))^2 \\ h(f_1,g) = 1 - \sqrt{\frac{f_1}{g}} \\ \mathit{where}\colon \\ G(t) = |sin(0.5\pi t)|, \quad F(t) = 10^{2\sin(0.5\pi t)}, \quad t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor \\ \mathbf{x}_{\mathsf{I}_i} \in [0,1]; \quad \mathbf{x}_{\mathsf{II}_i} \in [-1,1] \end{cases} \tag{7}$$

FDA3 has a convex POF, and both values of the POS and POF change. Therefore, it is called a Type II DMOOP. For FDA3, $POF = (1 + G(t))(1 - \sqrt{\frac{f_1}{1+G(t)}})$, as illustrated in Figure 2(b). The POS is $x_i = G(t),\ \forall x_i \in \mathbf{x}_{\mathsf{II}}$, similar to the POS of FDA1 (refer to e 1). The f_1 function of the two-objective FDA DMOOPs regulate the spread of solutions in objective space. Therefore, when f_1 changes over time, as is the case with FDA3, the spread of solutions in the POF changes over time.

$$\text{FDA4} = \begin{cases} \mathit{Minimise}\colon \mathbf{f}(\mathbf{x},t) = (f_1(\mathbf{x}, g(\mathbf{x}_{\mathsf{II}},t)), \ldots, f_k(\mathbf{x}, g(\mathbf{x}_{\mathsf{II}},t))) \\ f_1(\mathbf{x},g,t) = \left(1 + g(\mathbf{x}_{\mathsf{II}},t)\right) \prod_{i=1}^{M-1} \cos\left(\frac{x_i\pi}{2}\right) \\ f_k(\mathbf{x},g,t) = \left(1 + g(\mathbf{x}_{\mathsf{II}},t)\right) \left(\prod_{i=1}^{M-1} \cos\left(\frac{x_i\pi}{2}\right)\right) \sin\left(\frac{y_{M-k+1}\pi}{2}\right), \forall k = 2,\ldots,M-1 \\ f_m(\mathbf{x},g,t) = \left(1 + g(\mathbf{x}_{\mathsf{II}},t)\right) \prod_{i=1}^{M-1} \sin\left(\frac{x_1\pi}{2}\right) \\ \mathit{where}\colon \\ g(\mathbf{x}_{\mathsf{II}},t) = \sum_{x_i \in \mathbf{x}_{II}} \left(x_i - G(t)\right)^2, \quad G(t) = |sin(0.5\pi t)|, \quad t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor \\ \mathbf{x}_{\mathsf{II}} = (x_M, \ldots, x_n); \quad x_i \in [0,1],\ \forall i = 1,\ldots,n \end{cases} \tag{8}$$



(a) POF of FDA4. ©2004 IEEE. Reprinted with permission from Farina et al. [2004].

(b) POF of FDA5 for four timesteps. ©2004 IEEE. Reprinted with permission from Farina et al. [2004].

Fig. 3. POF of FDA4 and FDA5 for three objective functions. The size of the sphere's radius of FDA5's POF changes in a cyclic manner as the value of G changes over time. The radius increases over time and then decreases to the value of 1.0.

For FDA4, values in the decision variable space (POS) change over time, but the values in the objective space (POF) remain the same. Therefore, it is a Type I DMOOP. It has a nonconvex POF with the true POF (POF) defined as $f_1^2 + f_2^2 + f_3^2 = 1$ for three objective functions, as illustrated in Figure 3(a). The POS of FDA4 is $x_i = G(t), \forall x_i \in \mathbf{x}_{\text{II}}$, similar to FDA1 (refer to Figure 1).

$$\text{FDA5} = \begin{cases} Minimise\text{: } \mathbf{f}(\mathbf{x}, t) = (f_1(\mathbf{x}, g(\mathbf{x}_{\text{II}}, t)), \ldots, f_k(\mathbf{x}, g(\mathbf{x}_{\text{II}}, t))) \\ f_1(\mathbf{x}, g, t) = \left(1 + g(\mathbf{x}_{\text{II}}, t)\right) \prod_{i=1}^{M-1} \cos\left(\frac{y_i \pi}{2}\right) \\ f_k(\mathbf{x}, g, t) = \left(1 + g(\mathbf{x}_{\text{II}}, t)\right) \left(\prod_{i=1}^{M-1} \cos\left(\frac{y_i \pi}{2}\right)\right) \sin\left(\frac{y_{M-k+1}\pi}{2}\right), \forall k = 2, \ldots, M-1 \\ f_m(\mathbf{x}, g, t) = \left(1 + g(\mathbf{x}_{\text{II}}, t)\right) \prod_{i=1}^{M-1} \sin\left(\frac{y_1 \pi}{2}\right) \\ where\text{:} \\ g(\mathbf{x}_{\text{II}}, t) = G(t) + \sum_{x_i \in \mathbf{x}_{II}} \left(x_i - G(t)\right)^2, \quad G(t) = |sin(0.5\pi t)|, \quad t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor \\ y_i = x_i^{F(t)}, \quad \forall i = 1, \ldots, (M-1) \\ F(t) = 1 + 100 \sin^4(0.5\pi t) \\ \mathbf{x}_{\text{II}} = (x_M, \ldots, x_n); \quad x_i \in [0, 1], \forall i = 1, \ldots, n \end{cases} \tag{9}$$

FDA5 has a nonconvex POF, where both the values in the decision variable space (POS) and the objective space (POF) change over time. Therefore, it is a Type II DMOOP. Furthermore, the spread of solutions in the POF changes over time. For FDA5 with three objective functions, the POF is $f_1^2 + f_2^2 + f_3^2 = \left(1 + G(t)\right)^2$, as illustrated in Figure 3(b). The POS of FDA5 is $x_i = G(t), \forall x_i \in \mathbf{x}_{\text{II}}$, similar to FDA1 (refer to Figure 1).

Many researchers have used the FDA DMOOPs over the years, as highlighted in Table I, where "Modified" indicates that the authors have used a modified version of the specific FDA DMOOP and "Other" indicates that the authors have used DMOOPs other than the FDA set. Table I shows that most researchers used the FDA1 DMOOP, which is of Type I, where the POS changes over time but the POF remains the same. Clearly, FDA1 is the easiest DMOOP of the FDA suite to solve. Therefore, using the FDA1 DMOOP alone to test whether an algorithm can solve DMOOPs is not sufficient.

Table I. Usage of FDA DMOOP to Test Algorithms' Performance

DMOOP	Version	Authors
FDA1	Original	[Farina et al. 2004; Amato and Farina 2005; Hatzakis and Wallace 2006; Mehnen et al. 2006; Zeng et al. 2006; Bingul 2007; Cámara et al. 2007a, 2007b; Zheng 2007; Zhou et al. 2007; Greeff and Engelbrecht 2008; Isaacs et al. 2008; Tan and Goh 2008; Wang and Dang 2008; Chen et al. 2009; Goh and Tan 2009b, 2009a; Isaacs et al. 2009; Ray et al. 2009; Lechuga 2009; Wang and Li 2009; Cámara et al. 2009, 2010; Cámara Sola 2010; Greeff and Engelbrecht 2010; Koo et al. 2010; Liu et al. 2010; Wang and Li 2010; Helbig and Engelbrecht 2011]
	Modified	[Zhou et al. 2007]
FDA2	Original	[Farina et al. 2004; Zeng et al. 2006; Cámara et al. 2007a, 2007b; Liu and Wang 2007; Wang and Dang 2008; Greeff and Engelbrecht 2010; Liu 2010; Wang and Li 2010; Helbig and Engelbrecht 2011]
	Modified	[Mehnen et al. 2006; Deb et al. 2007; Zheng 2007; Isaacs et al. 2008; Talukder and Khaled 2008; Khaled et al. 2008; Isaacs et al. 2009; Ray et al. 2009; Lechuga 2009; Cámara et al. 2009, 2010; Cámara Sola 2010; Liu et al. 2010]
FDA3	Original	[Farina et al. 2004; Shang et al. 2005; Zeng et al. 2006; Liu and Wang 2007; Wang and Dang 2008; Koo et al. 2010; Wang and Li 2010; Helbig and Engelbrecht 2011]
	Modified	[Zheng 2007; Talukder and Khaled 2008; Khaled et al. 2008; Cámara et al. 2009, 2010; Cámara Sola 2010]
FDA4	Original	[Farina et al. 2004; Mehnen et al. 2006; Zheng 2007; Greeff and Engelbrecht 2008; Cámara et al. 2009, 2010; Cámara Sola 2010; Greeff and Engelbrecht 2010]
FDA5	Original	[Farina et al. 2004; Shang et al. 2005; Zheng 2007; Greeff and Engelbrecht 2008; Chen et al. 2009; Cámara et al. 2009, 2010; Cámara Sola 2010; Greeff and Engelbrecht 2010]
	Modified	[Talukder and Khaled 2008; Khaled et al. 2008]
Other		[Mehnen et al. 2006; Liu and Wang 2007; Goh and Tan 2009b, 2009a; Wang and Li 2009; Koo et al. 2010; Liu et al. 2010; Liu 2010; Wang and Li 2010; Helbig and Engelbrecht 2011]

Several researchers have used the FDA2 DMOOP. However, the POF of FDA2 changes from a convex to a concave shape only for specific values of the decision variables [Mehnen et al. 2006; Deb et al. 2007], as can be seen, for example, in Helbig and Engelbrecht [2011, 2013b]. Therefore, even if an algorithm finds Pareto-optimal solutions, it may find a convex POF instead of a concave POF. To address this issue, several modifications to the h or g function of FDA2 have been suggested [Cámara et al. 2009, 2010; Deb et al. 2007; Isaacs et al. 2008; Lechuga 2009; Liu et al. 2010; Mehnen et al. 2006; Ray et al. 2009; Cámara Sola 2010; Zheng 2007]. Underlying problems with FDA3 also lead to several modifications to FDA3 being suggested [Cámara et al. 2010; Khaled et al. 2008; Talukder and Khaled 2008; Zheng 2007]. In order to test an algorithm's ability to solve Type III DMOOPs, Talukder and Khaled [2008] modified FDA5 to a Type III DMOO.

A generalisation of the FDA functions was suggested by Mehnen et al. [2006]. In contrast to the FDA functions, this generalised DMOOP, DTF, is constructed in such a way that the number of disconnected continuous POF sections, the number of local POFs, the curvature of the POF, the spread of the solutions, and the optimal decision variable values that represent the POS can be easily specified. The DTF DMOOP is defined as:

$$\text{DTF} = \begin{cases} Minimise\colon \mathbf{f}(\mathbf{x}, t) = (f_1(\mathbf{x}_{\mathrm{I}}, t), g(\mathbf{x}_{\mathrm{II}}, t) h(f_1(\mathbf{x}_{\mathrm{I}}, t), g(\mathbf{x}_{\mathrm{II}}, t), t)) \\ f_1(\mathbf{x}_{\mathrm{I}}, t) = x_1^{\beta(t)} \\ g(\mathbf{x}_{\mathrm{II}}, t) = 1 + \sum_{x_i \in \mathbf{x}_{\mathrm{II}}} ((x_i - \gamma(t))^2 - \cos(\omega(t(\tau)))\pi(x_i - \gamma(t)) + 1) \\ h(f_1, g, t) = 2 - \left(\frac{f_1}{g}\right)^{\alpha(t)} - \left(\frac{f_1}{g}\right) \left|\sin(\psi(t)\pi f_1)\right|^{\alpha(t)} \\ where\colon \\ t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor \\ \mathbf{x}_{\mathrm{I}} \in [0, 1], \quad \mathbf{x}_{\mathrm{II}_i} \in [-1, 1], \end{cases} \tag{10}$$

where β represents the spread of solutions, α the curvature of the POF, γ the optimal decision variable values or POS, ψ the number of POF sections, and ω the number of local POFs. For example, a Type II DMOOP can be constructed from DTF by setting the following parameter values: $n = 20$, $\alpha(t) = 0.2 + 4.8t^2$, $\beta(t) = 10^{2\sin(0.5\pi t)}$, $\gamma(t) = \sin(0.5\pi t)$, $\psi(t) = ts$ with $s \in \mathbb{R}$, and $\omega(t) \propto \psi(t)$.

Tang et al. [2007] also suggested constructing DMOOPs based on the ZDT functions of Deb [1999]. Three objective functions are constructed similar to the DMOOPs of Farina et al. [2004] and provide an additional explanation of how the POF is calculated. For two objective DMOOPs, the following format is used:

$$\begin{cases} Minimise\colon \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}_{\mathrm{I}}),\ f_2(\mathbf{x}_{\mathrm{II}})) \\ f_1(\mathbf{x}_{\mathrm{I}}) = f_1(\mathbf{x}_{\mathrm{I}}) \\ f_2(\mathbf{x}_{\mathrm{II}}) = u(t) g(\mathbf{x}_{\mathrm{II}}) v(t) \left[h\left(f(\mathbf{x}_{\mathrm{I}}), g(\mathbf{x}_{\mathrm{II}}) v(t)\right)\right] \end{cases} \tag{11}$$

with $u(t)$ and $v(t)$ functions of time t. The selection of $u(t)$ and $v(t)$ lead to the construction of various types of DMOOPs:

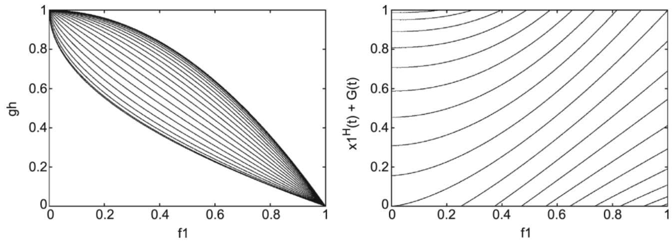
—$u(t) = 1$ and $v(t)$ that changes over time create a DMOOP of Type I.
—$v(t) = 1$ and $u(t)$ that changes over time create a DMOOP of Type III.
—$u(t)$ and $v(t)$ that change over time create a DMOOP of Type II.

The formulation of the DMOOP using Equation (11) can therefore lead to the creation of various types of DMOOPs by changing the values of $v(t)$ and $u(t)$. It is very similar to the FDA DMOOPs, but by formulating the DMOOP in this way, the required type of DMOOP can be easily created. Since these functions are based on the ZDT functions, they adhere to the characteristics of benchmark functions recommended by Deb [2004]. An example of Type III DMOOP using Equation (11) where $v(t) = 1$ and $u(t) = t^2$ is:

$$\begin{cases} Minimise\colon \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}_{\mathrm{I}}),\ f_2(\mathbf{x}_{\mathrm{II}})) \\ f_1(\mathbf{x}_{\mathrm{I}}) = 1 - exp(-4x_1)\sin^6(6\pi x_1) \\ f_2(\mathbf{x}_{\mathrm{II}}) = t^2 g\left(1 - \left(\frac{f_1}{g}\right)^2\right) \\ where\colon \\ g = 1 + 9\left(\frac{\sum_{i=2}^{n} x_i}{n-1}\right)^{0.25} \\ x_i \in [0, 1], \forall i = 1, 2, \ldots, 10 \end{cases} \tag{12}$$

Wang and Li [2009, 2010] recently also suggested new Type I DMOOPs that are created by adapting the ZDT functions.

Based on the construction guidelines of Farina et al. [2004] and Goh and Tan [2009b] presented three DMOOPs, namely dMOP1, dMOP2, and dMOP3. dMOP1 and dMOP2 have a POF that changes from convex to concave over time, with dMOP1 being a



(a) POF of dMOP1 (b) POS of ZJZ

Fig. 4. POF of dMOP1 and POS of ZJZ with $n_t = 10$ and $\tau_t = 10$ for 1,000 iterations. POF of dMOP1 changes in a cyclic manner over time, by moving either from the middle line to the top line for certain timesteps or from the bottom line to the middle line for the other timesteps.

Type III DMOOP and dMOP2 a Type II DMOOP. In the FDA DMOOP suite, FDA2 also has a POF that changes from convex to concave over time, and FDA2 is a Type II DMOOP. However, dMOP1 and dMOP2 do not suffer from the decision variable selection problem from which FDA2 suffers. dMOP1 tests whether a DMOO algorithm can solve problems where the POF changes from convex to concave but the POS remains the same over time, and dMOP2 adds the difficulty of solving this problem with a changing POS and POF. dMOP3 is very similar to FDA1; however, the variable that controls the spread of the POF solutions (x_1 in FDA1) changes over time. This may cause an algorithm to struggle to maintain a diverse set of solutions as the POS changes over time. The dMOP benchmark functions are defined as follows:

$$\text{dMOP1} = \begin{cases} \textit{Minimise}\colon\ f(\mathbf{x},t) = (f_1(\mathbf{x}_{\text{I}}), g(\mathbf{x}_{\text{II}}) \cdot h(f_1(\mathbf{x}_{\text{I}}), g(\mathbf{x}_{\text{II}}), t)) \\ f_1(\mathbf{x}_{\text{I}}) = x_1 \\ g(\mathbf{x}_{\text{II}}) = 1 + 9\sum_{x_i \in \mathbf{x}_{\text{II}}} (x_i)^2 \\ [0.1cm] h(f_1, g, t) = 1 - \left(\frac{f_1}{g}\right)^{H(t)} \\ \textit{where}\colon \\ H(t) = 0.75\sin(0.5\pi t) + 1.25, \quad t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor \\ x_i \in [0,1];\ \ \mathbf{x}_{\text{I}} = (x_1);\ \ \mathbf{x}_{\text{II}} = (x_2, \ldots, x_n) \end{cases} \tag{13}$$

The POF of dMOP1 changes from convex to concave over time, but the POF remains the same. Therefore, it is a Type III problem, with $POF = 1 - f_1^{H(t)}$, as illustrated in Figure 4(a). The POS of dMOP1 is $x_i = 0, \forall x_i \in \mathbf{x}_{\text{II}}$, similar to FDA2.

$$\text{dMOP2} = \begin{cases} \textit{Minimise}\colon\ f(\mathbf{x},t) = (f_1(\mathbf{x}_{\text{I}}), g(\mathbf{x}_{\text{II}}, t) \cdot h(f_1(\mathbf{x}_{\text{I}}), g(\mathbf{x}_{\text{II}}, t), t)) \\ f_1(\mathbf{x}_{\text{I}}) = x_1 \\ g(\mathbf{x}_{\text{II}}, t) = 1 + 9\sum_{x_i \in \mathbf{x}_{\text{II}}} (x_i - G(t))^2 \\ [0.1cm] h(f_1, g, t) = 1 - \left(\frac{f_1}{g}\right)^{H(t)} \\ \textit{where}\colon \\ H(t) = 0.75\sin(0.5\pi t) + 1.25, \quad G(t) = \sin(0.5\pi t), \quad t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor \\ x_i \in [0,1];\ \ \mathbf{x}_{\text{I}} = (x_1);\ \ \mathbf{x}_{\text{II}} = (x_2, \ldots, x_n) \end{cases} \tag{14}$$

dMOP2 has a POF that changes from convex to concave, where the values in both the POS and POF change. Therefore, dMOP2 is a Type II problem, with $POF = 1 - f_1^{H(t)}$, similar to dMOP1 (refer to Figure 4(a)). The POS of dMOP2 is $x_i = G(t), \forall x_i \in \mathbf{x}_{\text{II}}$, similar to FDA1 (refer to Figure 1).

$$\text{dMOP3} = \begin{cases} \textit{Minimise}: \ f(\mathbf{x}, t) = (f_1(\mathbf{x}_{\mathsf{I}}), g(\mathbf{x}_{\mathsf{II}}, t) \cdot h(f_1(\mathbf{x}_{\mathsf{I}}), g(\mathbf{x}_{\mathsf{II}}, t))) \\ f_1(\mathbf{x}_{\mathsf{I}}) = x_r \\ g(\mathbf{x}_{\mathsf{II}}, t) = 1 + 9 \sum_{x_i \in \mathbf{x}_{\mathsf{II}} \setminus \mathsf{x}_r} (x_i - G(t))^2 \\ h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}} \\ \textit{where}: \\ G(t) = \sin(0.5\pi t), \quad t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_t} \right\rfloor \\ x_i \in [0, 1]; \quad r = \bigcup(1, 2, \ldots, n) \end{cases} \tag{15}$$

dMOP3 has a convex POF, where the POS changes over time but the POF remains the same. dMOP3 is therefore a Type I DMOOP, and the spread of the *POF* solutions changes over time. Similar to FDA1, for dMOP3, $POF = 1 - \sqrt{f_1}$ (refer to Figure 1) and the POS is $x_i = G(t), \forall x_i \in \mathbf{x}_{\mathsf{II}}$ (refer to Figure 1(b)).

More recently, Li and Zhang [2006] and Deb et al. [2006] presented MOOPs with decision variable dependencies (or linkages). Zhou et al. [2007] modified FDA1 to incorporate dependencies between the decision variables. The modified FDA1 DMOOP is defined as follows:

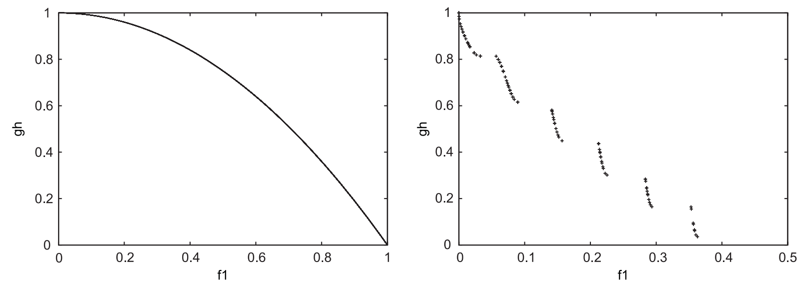
$$\text{ZJZ} = \begin{cases} \textit{Minimise}: \ f(\mathbf{x}, t) = (f_1(\mathbf{x}_{\mathsf{I}}), g(\mathbf{x}_{\mathsf{II}}, t) \cdot h(f_1(\mathbf{x}_{\mathsf{I}}), g(\mathbf{x}_{\mathsf{II}}, t))) \\ f_1(\mathbf{x}_{\mathsf{I}}) = x_1 \\ g(\mathbf{x}_{\mathsf{II}}, t) = 1 + \sum_{x_i \in \mathbf{x}_{\mathsf{II}}} \left(x_i - G(t) - x_1^{H(t)}\right)^2 \\ h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^{H(t)} \\ \textit{where}: \\ G(t) = \sin(0.5\pi t), \quad H(t) = 1.5 + G(t), \quad t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_t} \right\rfloor \\ \mathbf{x}_{\mathsf{I}} \in [0, 1]; \quad \mathbf{x}_{\mathsf{II}} = (x_2, \ldots, x_n) \in [-1, 2] \end{cases} \tag{16}$$

For ZJZ, the values of both the POS and POF change over time. Therefore, it is a Type II DMOOP. ZJZ's POF is similar to dMOP1 (refer to Figure 4(a)) and changes from convex to concave over time, with $POF = 1 - f_1^{H(t)}$. However, there are nonlinear dependencies between the decision variables that make the DMOOP more difficult to solve. The POS of ZJZ is $x_i = G(t) + x_1^{H(t)}, \forall x_i \in \mathbf{x}_{\mathsf{II}}$, as illustrated in Figure 4(b).

Another shortcoming of the FDA DMOOP suite is that all DMOOP objective functions consist of decision variables with the same rate of change over time. Koo et al. [2010] suggested two new benchmark functions where each decision variable has its own rate of change, except the variable x_1 that controls the spread of solutions. These two functions, DIMP1 and DIMP2, are defined as follows:

$$\text{DIMP1} = \begin{cases} \textit{Minimise}: \ f(\mathbf{x}, t) = (f_1(\mathbf{x}_{\mathsf{I}}), g(\mathbf{x}_{\mathsf{II}}, t) \cdot h(f_1(\mathbf{x}_{\mathsf{I}}), g(\mathbf{x}_{\mathsf{II}}, t))) \\ f_1(\mathbf{x}_{\mathsf{I}}) = x_1 \\ g(\mathbf{x}_{\mathsf{II}}, t) = 1 + \sum_{x_i \in \mathbf{x}_{\mathsf{II}}} (x_i - G_i(t))^2 \\ h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^2 \\ \textit{where}: \\ G_i(t) = \sin\left(0.5\pi t + 2\pi \left(\frac{i}{n+1}\right)\right)^2, \quad t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_t} \right\rfloor \\ \mathbf{x}_{\mathsf{I}} = (x_1) \in [0, 1]; \quad \mathbf{x}_{\mathsf{II}} = (x_2, x_3, \ldots, x_n) \in [-1, 1]^{n-1} \end{cases} \tag{17}$$

The POS of DIMP1 changes over time, but the POF remains the same. Therefore, DIMP1 is a Type I DMOOP, with $POF = 1 - f_1^2$ (as illustrated in Figure 5(a)), and the



(a) POF of DIMP1 (b) POF of HE1

Fig. 5. POF of DIMP1 with $n_t = 10$ and $\tau_t = 10$ for 1,000 iterations and POF of HE1 with $n_t = 10$, $\tau_t = 50$ and $\tau = 299$. POF of HE1 is the shape of a sine wave and therefore discontinuous. The sine wave's period changes over time.

POS is $x_i = G(t), \forall x_i \in \mathbf{x}_{II}$, similar to FDA1 (refer to Figure 1(b)).

$$\text{DIMP2} = \begin{cases} \textit{Minimise}: f(\mathbf{x}, t) = (f_1(\mathbf{x}_I), g(\mathbf{x}_{II}, t) \cdot h(f_1(\mathbf{x}_I), g(\mathbf{x}_{II}, t))) \\ f_1(\mathbf{x}_I) = x_1 \\ g(\mathbf{x}_{II}, t) = 1 + 2(n-1) + \sum_{x_i \in \mathbf{x}_{II}} [(x_i - G_i(t))^2 - 2\cos(3\pi(x_i - G_i(t)))] \\ h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}} \\ \textit{where}: \\ G_i(t) = \sin\left(0.5\pi t + 2\pi\left(\frac{i}{n+1}\right)\right)^2, \quad t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor \\ \mathbf{x}_I \in [0, 1]; \quad \mathbf{x}_{II} \in [-2, 2]^{n-1} \end{cases} \tag{18}$$

DIMP2 is a Type I problem, since its POS changes over time but its POF remains the same. Similar to FDA1, DIMP2's *POF* is $1 - \sqrt{f_1}$ (refer to Figure 1), and the POS is $x_i = G(t), \forall x_i \in \mathbf{x}_{II}$ (refer to Figure 1(b)).

The FDA and dMOP suites only contain DMOOPs with a continuous POF. Two discontinous functions, namely TP1_{mod} and TP2_{mod}, were presented by Greeff and Engelbrecht [2008]. However, these two functions do not allow easy scalability of the number of decision variables. Therefore, TP1_{mod} and TP2_{mod} do not adhere to the characteristics of benchmark functions that are recommended by Deb et al. Recently, Helbig and Engelbrecht [2011] presented two DMOOPs with a discontinuous POF, namely HE1 and HE2. These two functions are based on the ZDT3 [Zitzler et al. 2000] MOOP that developed in such a way that it adheres to the characteristics recommended by Deb et al. HE1 and HE2 were developed by adapting ZDT3 to be dynamic and therefore adhere to the benchmark function characteristics recommended by Deb et al. HE1 and HE2 are defined as:

$$\text{HE1} = \begin{cases} \textit{Minimise}: f(\mathbf{x}, t) = (f_1(\mathbf{x}_I), g(\mathbf{x}_{II}) \cdot h(f_1(\mathbf{x}_I), g(\mathbf{x}_{II}), t)) \\ f_1(\mathbf{x}_I) = x_1 \\ g(\mathbf{x}_{II}) = 1 + \frac{9}{n-1} \sum_{x_i \in \mathbf{x}_{II}} x_i \\ h(f_1, g, t) = 1 - \sqrt{\frac{f_1}{g}} - \frac{f_1}{g} \sin(10\pi t f_1) \\ \textit{where}: \\ t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor \\ x_i \in [0, 1]; \quad \mathbf{x}_I = (x_1); \quad \mathbf{x}_{II} = (x_2, \ldots, x_n) \end{cases} \tag{19}$$

$$
\text{HE2} = \begin{cases} Minimise\colon\ f(\mathbf{x},t) = (f_1(\mathbf{x}_{\mathrm{I}}), g(\mathbf{x}_{\mathrm{II}}) \cdot h(f_1(\mathbf{x}_{\mathrm{I}}), g(\mathbf{x}_{\mathrm{II}}), t)) \\ f_1(\mathbf{x}_{\mathrm{I}}) = x_i \\ g(\mathbf{x}_{\mathrm{II}}) = 1 + \frac{9}{n-1}\sum_{x_i \in \mathbf{x}_{\mathrm{II}}} x_i \\ h(f_1, g, t) = 1 - \left(\sqrt{\frac{f_1}{g}}\right)^{H(t)} - \left(\frac{f_1}{g}\right)^{H(t)} \sin(10\pi f_1) \\ where\colon \\ H(t) = 0.75\sin(0.5\pi t) + 1.25, \quad t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor \\ x_i \in [0,1];\ \ \mathbf{x}_{\mathrm{I}} = (x_1);\ \ \mathbf{x}_{\mathrm{II}} = (x_2, \ldots, x_n) \end{cases} \tag{20}
$$

Both HE1 and HE2 have a discontinuous POF, with various disconnected continuous subregions. Both are Type III DMOOPs, since their POFs change over time but their POSs remain the same. For HE1, $POF = 1 - \sqrt{f_1} - f_1 \sin(10\pi t f_1)$, as illustrated in Figure 5(b), and for HE2, $POF = 1 - (\sqrt{f1})^{H(t)} - f_1^{H(t)} \sin(0.5\pi f_1)$. The shape of HE2's POF is similar to HE1 (refer to Figure 5(b)). The POS for both HE1 and HE2 is $x_i = 0, \forall x_i \in \mathbf{x}_{\mathrm{II}}$, similar to FDA2.

Avdagić et al. [2009] introduced an adaptation of the DTLZ problems to develop the following types of benchmark functions: Type I DMOOP, where the POS changes coherently over time but the POF remains the same; Type II DMOOP, where the shape of the POS continuously changes and the POF also changes over time; and a Type II DMOOP, where the number of objective functions change over time [Avdagić et al. 2009]. These benchmark functions are developed from the following general equation:

$$
DTLZ_{Av} = \begin{cases} Minimise: q(\mathbf{x}) = (q_1(\mathbf{x}), \ldots, q_m(\mathbf{x})) \\ q_1(\mathbf{x}) = a_1 x_1^{c_1} x_2^{c_1} \ldots x_{m-1}^{c_1} (1 - x_m)^{c_1} g_1(\mathbf{x}) + b_1 \\ q_2(\mathbf{x}) = a_2 x_1^{c_2} x_2^{c_2} \ldots (1 - x_{m-1})^{c_2} (1 - x_m)^{c_2} g_2(\mathbf{x}) + b_2 \\ \vdots \\ q_{m-1}(\mathbf{x}) = a_{m-1} x_1^{c_{m-1}} (1 - x_2)^{c_{m-1}} \ldots (1 - x_{m-1})^{c_{m-1}} (1 - x_m)^{c_{m-1}} \\ \qquad\qquad g_{m-1}(\mathbf{x}) + b_{m-1} \\ q_m(\mathbf{x}) = a_m (1 - x_1)^{c_m} (1 - x_2)^{c_m} \ldots (1 - x_{m-1})^{c_m} (1 - x_m)^{c_m} g_m(\mathbf{x}) + b_m \\ where: \\ g_i = 1 - d_i \cos(20\pi x_i), \quad a_i, b_i, c_i, d_i \in \mathbb{R} \end{cases} \tag{21}
$$

A Type I DMOOP with a continuously changing POS is created by using Equation (21) and setting the following parameter values: $a_i = 1$, $d_i = 0$, and $b_i = b_i k$, where k represents the iteration and $c_i = 1$ or $c_i = 2$. Similarly, a Type II DMOOP with continuously changing POS and POF are developed by setting the following parameter values: $a_i = 1$, $b_i = b_i(k)$, $c_i(k) = 5b_i k$, and $d_i = 0$. To develop a Type II DMOOP with a changing number of objectives, the same parameters are used as those specified for the Type II DMOOP, with two objective functions being used for a certain number of iterations and then using three objective functions for the other iterations. These additional types of DMOOPs, which are not part of the FDA benchmark function set, may become important if these kind of changes occur in a real-world problem.

Recently, Huang et al. [2011] pointed out that all DMOOPs assume that the current found POS does not affect the future POS or POF. To the best knowledge of the authors of this article, none of the suggested DMOOPs have a POS or POF that depends on the previous POS or POF. Furthermore, most DMOOPs consist of a static number of decision variables and objective functions. Therefore, Huang et al. [2011] introduced four DMOOPs that incorporate these scenarios, defined as follows:

$$
\mathsf{T1} = \begin{cases}
Minimise\colon\ f(\mathbf{x},t) = \big(f_1(\mathbf{x},t),\ f_2(\mathbf{x},t)\big) \\
f_1(\mathbf{x},t) = \sum_{i=1}^{d_1(t)} \left(x_i^2 - 10\cos(2\pi x_i) + 10\right) \\
f_2(\mathbf{x},t) = (x_1-1)^2 + \sum_{i=2}^{d_2(t)} \left(x_i^2 - x_{i-1}\right)^2 \\
where\colon \\
d_1(t) = \lfloor n|\sin(t)|\rfloor \\
d_2(t) = \lfloor n|\cos^3(2t)|\rfloor \\
t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor,
\end{cases} \tag{22}
$$

with d_1 and d_2 varying the number of decision variables over time. The minimum for f_1 is 0, and the POS for f_1 is $x_i = 0, \forall i = 1, \dots, d_1(t)$. The minimum for f_2 is 0 with the POS $x_i = 1, \forall i = 1, \dots, d_2(t)$. Both the POF and POS remains static, but the number of decision variables change over time. Therefore, T1 is a Type IV DMOOP.

$$
\mathsf{T2} = \begin{cases}
Minimise\colon\ f(\mathbf{x},t) = \big(f_1(\mathbf{x},t), \dots,\ f_m(\mathbf{x},t)\big) \\
f_1(\mathbf{x},t) = (1+g(\mathbf{x}_{\mathsf{II}}))\prod_{i=1}^{m(t)-1} \cos\left(\frac{\pi x_i}{2}\right) \\
f_k(\mathbf{x},t) = (1+g(\mathbf{x}_{\mathsf{II}}))\prod_{i=1}^{m(t)-k} \cos\left(\frac{\pi x_i}{2}\right) \sin\left(\frac{\pi x_{m(t)-k+1}}{2}\right), \forall k = 2, \dots, m(t)-1 \\
f_m(\mathbf{x},t) = (1+g(\mathbf{x}_{\mathsf{II}}))\prod_{i=1}^{m(t)-1} \sin\left(\frac{\pi x_1}{2}\right) \\
where\colon \\
g(\mathbf{x}_{\mathsf{II}}) = \sum_{i=1}^{m(t)} (x_i - 0.5)^2 \\
m(t) = \lfloor M|\sin(0.5\pi t)|\rfloor, \quad t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor \\
\mathbf{x}_i \in [0,1],
\end{cases} \tag{23}
$$

with M representing the maximum number of objective functions, and m varying the number of objective functions over time. T2 is a Type III DMOOP, since its POF changes over time but its POS remains the same. The POS of T2 is $x_i = 0.5, \forall i = 1, \dots, m(t)$, and the POF is $\sum_i^{m(t)} \mathbf{f}_i^2 = 1$.

$$
\mathsf{T3} = \begin{cases}
Minimise\colon\ f(\mathbf{x},t) = \big(f_1(\mathbf{x},t),\ f_2(\mathbf{x},t)\big) \\
f_1(\mathbf{x},t) = R(\mathbf{x},t)\cos\left(\frac{\pi x_1}{2}\right) \\
f_2(\mathbf{x},t) = R(\mathbf{x},t)\sin\left(\frac{\pi x_1}{2}\right) \\
where\colon \\
R(\mathbf{x},t) = \bar{R}(\mathbf{x}, \mathsf{t}-1, t) + G(\mathbf{x},t) \\
\bar{R}(\mathbf{x},t) = \frac{1}{P}\sum_j^P R_j(\mathbf{x}, t-1) \\
\bar{R}(\mathbf{x},-1) = 1 \\
G(\mathbf{x},t) = \sum_{i=2}^{n} \left(x_i - \bar{R}(\mathbf{x}, t-1)\right)^2 \\
t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor \\
x_1 \in [0,1]; \quad x_i \in [\bar{R}(\mathbf{x},t) - 100, \bar{R}(\mathbf{x},t) + 100], \quad \forall i = 2, \dots, n,
\end{cases} \tag{24}
$$

with the value of $R(\mathbf{x}, t)$ depending on previous values of R. Therefore, if a slight error occurs with regards to the found value of R at time t, this error will increase over time, influencing the algorithm's ability to find the solutions at the next timesteps. Both the POS and POF remain static. Therefore, T3 is a Type IV DMOOP. The POS is $x_i = \bar{R}(\mathbf{x}, t-1), \forall i = 2, \dots, n$. The POF is $f_1^2 + f_2^2 = 1$. Similar to T1, T4 is a Type IV

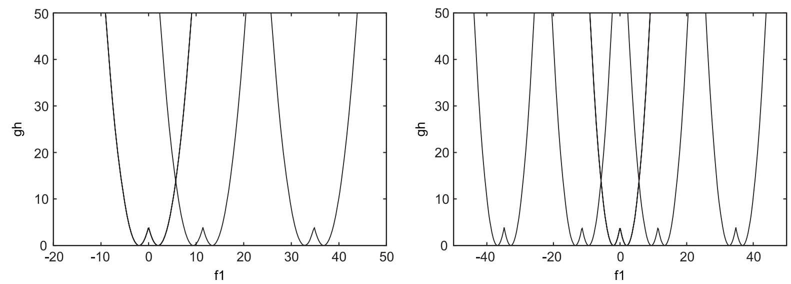


Fig. 6. POF of DSW1 and DSW2 with $n = 10$, $s = 6$, $n_t = 10$, and $\tau_t = 10$ for 1,000 iterations. POF of DSW1 changes in a cyclic manner, moving from left to right and then returning to left. POF of DSW2 moves from the middle to both the left and right at the same time, creating a mirror image—that is, to the left of zero, the POF moves to the left,and to the right of zero, the POF moves to the right.

DMOOP, defined as:

$$\text{T4} = \begin{cases} \textit{Minimise}\colon\ f(\mathbf{x}, t) = \big(f_1(\mathbf{x}, t),\ f_2(\mathbf{x}, t)\big) \\ f_1(\mathbf{x}, t) = \sum_{i=1}^{n}\big(x_i^2 - 10\cos(2\pi x_i) + 10\big) \\ f_2(\mathbf{x}, t) = (x_1 - r(t))^2 + \sum_{i=2}^{n}\big(x_i^2 - x_{i-1}\big)^2 \\ \textit{where}\colon \\ r(\mathbf{x}, t) = \frac{1}{n}\sum_{x_i \in \mathbf{x}}(x_i - 0) \\ t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor, \end{cases} \tag{25}$$

with r representing the average error of the decision variables of the selected POS (POS^*). Since the POS of T4 is $x_i = 0, \forall i = 1, 2, \ldots, n$, the average error of the decision variables of POS^* is $r(\mathbf{x}, t) = \frac{1}{n}\sum_{x_i \in \mathbf{x}}(x_i - 0)$. The selected trade-off solution set, POS^*, is derived from the current POS by a decision-making mechanism used by the decision maker. Therefore, for T4, the POF depends on the decision-making mechanism used at previous timesteps.

Mehnen et al. [2006] suggested that simpler benchmark functions are required to analyse the effect of different dynamic properties in a more isolated manner. For this reason, they presented the DSW DMOOP generator that is based on the static MOOP of Schaffer [1985]. The DSW DMOOPs are parabolic and are similar to the sphere function that are typically used to test whether an algorithm can solve DSOOPs. The DSW benchmark generator is defined as:

$$\text{DSW} = \begin{cases} \textit{Minimise}\colon\ f(\mathbf{x}, t) = (f_1(\mathbf{x}, t),\ f_2(\mathbf{x}, t)) \\ f_1(\mathbf{x}, t) = (a_{11}x_1 + a_{12}|x_1| - b_1 G(t))^2 + \sum_{i=2}^{n} x_i^2 \\ f_2(\mathbf{x}, t) = (a_{21}x_1 + a_{22}|x_1| - b_2 G(t) - 2)^2 + \sum_{i=2}^{n} x_i^2 \\ \textit{where}\colon \\ G(t) = t(\tau)s, \quad t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor, \end{cases} \tag{26}$$

with s representing the severity of change. Using Equation (26), the following three benchmark functions are created:

$$DSW1\colon \begin{cases} x \in [-50, 50]^n,\ a_{11} = 1,\ a_{12} = 0,\ a_{21} = 1, \\ a_{22} = 0,\ b_1 = 1,\ b_2 = 1 \end{cases} \tag{27}$$

DSW1 has a dynamic POF and POS, and is therefore a Type II DMOOP. The POS of DSW1 is $x_1 \in [G(t), G(t) + 2]$ and $x_i = 0, \forall i = 2, 3, \ldots, n$. The POF is $POF = (\sqrt{f_1} - 2)^2$ with $f_1 = (x_1 - G(t))^2$, as illustrated in Figure 6(a). DSW1 is similar to the spherical

SOOP function, where the center of the sphere is linearly shifted over time.

$$DSW2: \begin{cases} x \in [-50, 50]^n,\ a_{11} = 0,\ a_{12} = 1,\ a_{21} = 0, \\ a_{22} = 1,\ b_1 = 1,\ b_2 = 1 \end{cases} \tag{28}$$

Both the POS and POF of DSW2 change over time. Therefore, DSW2 is a Type II DMOOP. DSW2 has a disconnected POS, with $x_1 \in [-G(t) - 2, -G(t)] \cup [G(t), G(t) + 2]$, and $x_i = 0, \forall i = 2, 3, \ldots, n$. If a periodical $G(t)$ is used, the POSs will join and depart periodically. The POF of DSW2 is similar to that of DSW1, namely $POF = (\sqrt{f_1} - 2)^2$, but with $f_1 = (|x_1| - G(t))^2$.

$$DSW3: \begin{cases} x \in [-50, 50]^n,\ a_{11} = 1,\ a_{12} = 0,\ a_{21} = 1, \\ a_{22} = 0,\ b_1 = 0,\ b_2 = 1 \end{cases} \tag{29}$$

DSW3 has a changing POF and POS, and is therefore a Type II DMOOP. For DSW3, the POS is $x_1 \in [0, G(t) + 2]$, and the POF is $POF = (\sqrt{f_1} - G(t) - 2)^2$ with $f_1 = x_1^2$. Setting $b_1 = 0$ causes one border of the POS interval for x_1, namely $G(t) + 2$, to change over time, whereas the other border, 0, remains static.

5. ISSUES WITH CURRENT DMOOPS

From Section 4, the following shortcomings of current DMOOPs are identified [Helbig and Engelbrecht 2013a]. None of the DMOOPs have:

—An isolated POF (refer to Section 3.1)
—A deceptive POF (refer to Section 3.1)
—A POF that is defined by nonlinear curves in the decision space (i.e., a complex POS.

When a DMOOP has an isolated POF, the lack of gradient information may cause a DMOA difficulty converging towards the POF. In addition, since the majority of the search space is fairly flat, no useful information is provided with regards to the location of the POF. Therefore, DMOOPs with an isolated POF are difficult to solve [Huband et al. 2006; Deb 2004]. A DMOOP with a deceptive POF is a multimodal problem, since there exist more than one optima and the search space favours the deceptive optimum, which is a local POF and not the global POF. Multimodal problems are difficult to solve, as a DMOA can get stuck in a local POF. DMOOPs with a deceptive POF are even more difficult to solve than multimodal DMOOPs, since the global POF is in an unlikely place in the search space [Huband et al. 2006; Deb 2004]. Although many benchmark functions have a nonlinear POF, the POS is defined by a linear function. However, when a DMOOP has a POS that is defined by a nonlinear function, the DMOOP will be more difficult to solve.

It should be noted that although these shortcomings do not occur in benchmark functions, they may occur in real-world DMOOPs. Therefore, the three shortcomings of DMOOPs listed earlier are addressed in this section. Section 5.1 presents an approach to adapt current DMOOPs' POF to an isolated POF. A similar approach to change a DMOOPs' POF to a deceptive POF is presented in Section 5.2. In addition, new DMOOPs with complex POSs are introduced in Section 5.3.

5.1. DMOOPs with an Isolated POF

Flat regions occur when an open subset of decision variable values maps to a single objective function value. When a DMOOP has objective functions with flat regions, its POF is referred to as an isolated POF. No DMOOPs with an isolated POF have been proposed in the DMOO literature. Therefore, this section presents an approach that can be used to develop DMOOPs with an isolated POF [Helbig and Engelbrecht 2013a].

The WFG MOOP benchmark function suite was introduced by Huband et al. [2006] to address shortcomings of other MOO test suites. One of the shortcomings that the WFG suite addresses is the development of MOO benchmark functions with isolated POFs. This approach is adapted so that it can be applied to DMOOPs currently used to evaluate DMOAs.

Decision variables are mapped to new values to create flat regions with the following equation [Huband et al. 2006]:

$$y_i(x_i, A, B, C) = A + \min(0, \lfloor x_i - B \rfloor)\frac{A(B - x_i)}{B} - \min(0, \lfloor C - y \rfloor)\frac{(1 - A)(x_i - C)}{1 - C}, \quad (30)$$

where $A, B, C \in [0, 1]$, $B < C$, $B = 0 \Rightarrow A = 0 \wedge C \neq 0$, and $C = 1 \Rightarrow A = 1 \wedge B \neq 0$. All values of x_i between B and C are mapped to the value of A to create a flat region between B and C.

This mapping can be applied to existing DMOOPs, of which two examples are provided next, namely the adjustment of the three-objective FDA5 DMOOP [Helbig and Engelbrecht 2013a] (refer to Equation (9)) and the two-objective dMOP2 DMOOP (refer to Equation (14)):

$$\text{FDA5}_{iso} = \begin{cases} \textit{Minimise}: \mathbf{f}(\mathbf{x}, t) = (f_1(\mathbf{x}, g(\mathbf{x_{II}}, t)), \ldots, f_k(\mathbf{x}, g(\mathbf{x_{II}}, t))) \\ f_1(\mathbf{x}, g, t) = \left(1 + g(\mathbf{x_{II}}, t)\right) \prod_{i=1}^{M-1} \cos\left(\frac{y_i \pi}{2}\right) \\ f_k(\mathbf{x}, g, t) = \left(1 + g(\mathbf{x_{II}}, t)\right) \left(\prod_{i=1}^{M-1} \cos\left(\frac{y_i \pi}{2}\right)\right) \sin\left(\frac{y_{M-k+1} \pi}{2}\right), \forall k = 2, \ldots, M-1 \\ f_m(\mathbf{x}, g, t) = \left(1 + g(\mathbf{x_{II}}, t)\right) \prod_{i=1}^{M-1} \sin\left(\frac{y_1 \pi}{2}\right) \\ \textit{where}: \\ g(\mathbf{x_{II}}, t) = \sum_{x_j \in \mathbf{x_{II}}} \left(y_j - G(t)\right)^2; \quad G(t) = |sin(0.5\pi t)|, \quad t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor \\ F(t) = 1 + 100 \sin^4(0.5\pi t) \\ y_i = x_i^{F(t)}, \quad \forall i = 1, \ldots, (M-1); \quad y_j = y_j(x_j, A, B, C), \quad \forall x_j \in \mathbf{x_{II}} \\ \mathbf{x_{II}} = (x_M, \ldots, x_n), \quad x_i \in [0, 1], \forall i = 1, \ldots, n, \end{cases} \quad (31)$$

where y_j is calculated using Equation (30). Example values for A, B, and C are $G(t)$, 0.001, and 0.05, respectively. Similar to FDA5 (refer to Equation (9)), the POF of FDA5$_{iso}$ is $f_1^2 + f_2^2 + f_3^2 = (1 + G(t))^2$ (as illustrated in Figure 3(b)). The POS of FDA5$_{iso}$ is $x_i = G(t)$, $\forall x_i \in \mathbf{x_{II}}$, similar to FDA1 (refer to Figure 1).

$$\text{dMOP2}_{iso} = \begin{cases} \textit{Minimise}: f(\mathbf{x}, t) = (f_1(\mathbf{x_I}), g(\mathbf{x_{II}}, t) \cdot h(f_1(\mathbf{x_I}), g(\mathbf{x_{II}}, t), t)) \\ f_1(\mathbf{x_I}) = x_1 \\ g(\mathbf{x_{II}}, t) = 1 + 9 \sum_{x_i \in \mathbf{x_{II}}} (y_i - G(t))^2 \\ h(f_1, g, t) = 1 - \left(\frac{f_1}{g}\right)^{H(t)} \\ \textit{where}: \\ y_i = y_i(x_i, A, B, C), \quad \forall x_i \in \mathbf{x_{II}} \\ H(t) = 0.75 \sin(0.5\pi t) + 1.25 \\ G(t) = \sin(0.5\pi t), \quad t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor \\ x_i \in [0, 1]; \quad \mathbf{x_I} = (x_1), \quad \mathbf{x_{II}} = (x_2, \ldots, x_n), \end{cases} \quad (32)$$

where y_i is calculated using Equation (30). Example values for A, B, and C are $G(t)$, 0.001, and 0.05, respectively.

5.2. DMOOPs with a Deceptive POF

When a DMOOP has a deceptive POF, the DMOOP has at least two optima, with the search space favouring the deceptive POF. Some of the benchmark functions discussed in Section 4 are multimodal (e.g., FDA3 in Equation (7)). However, none of the DMOOPs discussed in Section 4 has a deceptive optimum. This section presents an approach to adjust existing DMOOPs in such a way that the DMOOPs have a deceptive POF [Helbig and Engelbrecht 2013a].

Huband et al. [2006] also introduced an approach to develop MOOPs with a deceptive POF. Similar to the approach to develop MOOPs with isolated POFs (refer to Section 5.1), the following transformation function is used:

$$y_i(x_i, A, B, C) = \left(\frac{\lfloor y - A + B \rfloor \left(1 - C + \frac{A-B}{B}\right)}{A - B} + \frac{1}{B} + \frac{\lfloor A + B - y \rfloor \left(1 - C + \frac{1-A-B}{B}\right)}{1 - A - B} \right) (|y - A| - B) + 1, \tag{33}$$

where $A \in (0, 1)$, $0 < B \ll 1$, $0 < C \ll 1$, $A - B > 0$, and $A + B < 1$. A represents the value at which x_i is mapped to zero—that is, the global minimum of the transformation function. The size of the basin leading to A is represented by B, and the value of the deceptive optimum is represented by C.

Therefore, DMOOPs with a deceptive POF can be developed by applying this transformation (or mapping) function to existing DMOOPs. For example, calculating y_j in Equation (31) and y_i in Equation (32) using Equation (33) will transform FDA5$_{iso}$ and dMOP2$_{iso}$ into DMOOPs with deceptive POFs. A, B, and C in Equation (33) can, for example, be selected as 0.35, 0.001, and 0.05, respectively.

5.3. DMOOPs with Complex POSs

Another shortcoming of MOOPs is that the POS is defined by a simple function—for example, $x_i = sin(0.5\pi t)$ [Li and Zhang 2009]. Therefore, Li and Zhang [2009] presented MOOPs with complicated POSs, where the POS is defined by nonlinear curves in the decision space—for example, $x_j = \sin(6\pi x_1 + \frac{j\pi}{n}),\ \forall j = 2, 3, \ldots, n$. This shortcoming is also true for DMOOPs [Helbig and Engelbrecht 2013a]. Recently, Helbig and Engelbrecht [2013a] proposed three new DMOOPs with complex POSs. This section presents these DMOOPs and introduces four new DMOOPs with complicated POSs, based on the MOOPs of Li and Zhang [2009].

The first DMOOP, HE3, has a POF that changes over time but the POS remains the same. Therefore, HE3 is a Type III DMOOP. The POS and POF of HE3 are:

$$POS : x_j = x_1^{0.5\left(\frac{3(j-2)}{n-2}\right)},\ \forall j = 2, 3, \ldots, n.$$
$$POF : y = \left(2 - \sqrt{x_1}\right) \left[1 - \left(\frac{x_1}{2 - \sqrt{x_1}} \right)^{H(t)} \right]$$

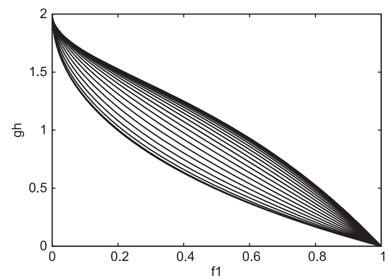
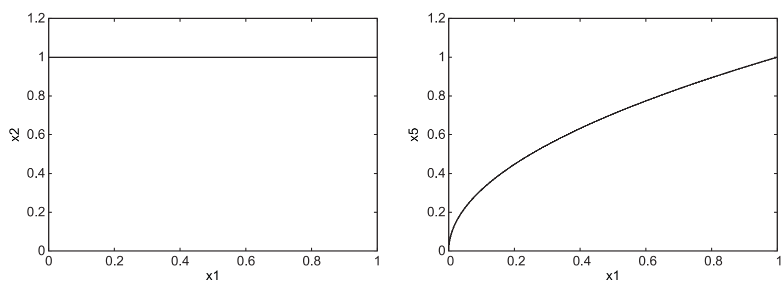


Fig. 7. POF of HE3 with $n_t = 10$ and $\tau_t = 10$ for 1,000 iterations. POF changes in a cyclic manner, moving from the middle to the top, then from the top to the middle, then from the middle to the bottom, then from the bottom to the middle. This whole process is then repeated. ©2013 IEEE. Reprinted with permission from [Helbig and Engelbrecht 2013a].



(a) POS of x_2 with $n_t = 10$ and $\tau_t = 10$ for 1,000 iterations

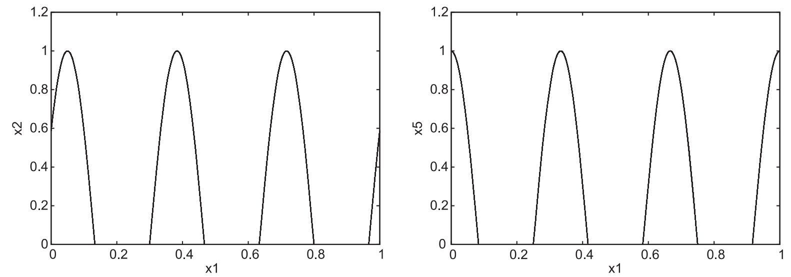
(b) POS of x_5 with $n_t = 10$ and $\tau_t = 10$ for 1,000 iterations

Fig. 8. POS of HE3 for two decision variables: x_2 and x_5.

HE3 is defined as:

$$\text{HE3} = \begin{cases} \textit{Minimise}: \mathbf{f}(\mathbf{x}, t) = (f_1(\mathbf{x}), g(\mathbf{x}, t) \cdot h(f_1(\mathbf{x}), g(\mathbf{x}, t))) \\ f_1(\mathbf{x}) = x_1 + \frac{2}{|J_1|} \sum_{j \in J_1} \left(x_j - x_1^{0.5\left(1.0 + \frac{3(j-2)}{n-2}\right)} \right)^2 \\ g(\mathbf{x}) = 2 - \sqrt{x_1} \frac{2}{|J_2|} \sum_{j \in J_2} \left(x_j - x_1^{0.5\left(1.0 + \frac{3(j-2)}{n-2}\right)} \right)^2 \\ h(f_1, g) = 1 - \left(\frac{f_1}{g} \right)^{H(t)} \\ \textit{where}: \\ H(t) = 0.75 \sin(0.5\pi t) + 1.25, \quad t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_t} \right\rfloor \\ J_1 = \{ j | \text{ j is odd and } 2 \le j \le n \} \\ J_2 = \{ j | \text{ j is even and } 2 \le j \le n \} \\ x_i \in [0, 1] \end{cases} \quad (34)$$

The POF and POS of HE3 are illustrated in Figures 7 and 8, respectively. It is important to note that unlike most of the other DMOOPs, the POS of HE3 to HE10 are different



(a) POS of x_2 with $n_t = 10$ and $\tau_t = 10$ for 1,000 iterations

(b) POS of x_5 with $n_t = 10$ and $\tau_t = 10$ for 1,000 iterations

Fig. 9. POS of HE4 for two decision variables: x_2 and x_5.

for each decision variable.

$$\text{HE4} = \begin{cases} \mathit{Minimise}: \mathbf{f}(\mathbf{x}, t) = (f_1(\mathbf{x}), g(\mathbf{x}, t) \cdot h(f_1(\mathbf{x}), g(\mathbf{x}, t))) \\ f_1(\mathbf{x}) = x_1 + \frac{2}{|J_1|} \sum_{j \in J_1} \left(x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right)^2 \\ g(\mathbf{x}) = 2 - \sqrt{x_1} + \frac{2}{|J_2|} \sum_{j \in J_2} \left(x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right)^2 \\ h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^{H(t)} \\ \mathit{where}: \\ H(t) = 0.75 \sin(0.5\pi t) + 1.25, \quad t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_t} \right\rfloor \\ J_1 = \{j |\ \text{j is odd and } 2 \leq j \leq n\} \\ J_2 = \{j |\ \text{j is even and } 2 \leq j \leq n\} \\ x_1 \in [0, 1], \quad x_i \in [-1, 1], \quad \forall i = 2, 3, \ldots, n \end{cases} \tag{35}$$

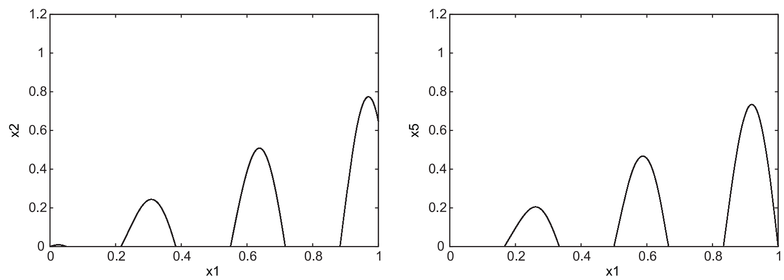
The POF of HE4 changes over time but the POS remains the same. Therefore, HE4 is a Type III DMOOP. The POS and POF of HE4 are:

$$POS: x_j = \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), \quad \forall j = 2, 3, \ldots, n.$$

$$POF: y = \left(2 - \sqrt{x_1}\right)\left[1 - \left(\frac{x_1}{2 - \sqrt{x_1}}\right)^{H(t)}\right]$$

The POS of HE4 is illustrated in Figure 9. The POF is similar to the POF of HE3 (refer to Figure 7).

$$\text{HE5} = \begin{cases} \mathit{Minimise}: \mathbf{f}(\mathbf{x}, t) = (f_1(\mathbf{x}), g(\mathbf{x}, t) \cdot h(f_1(\mathbf{x}), g(\mathbf{x}, t))) \\ f_1(\mathbf{x}) = x_1 + \frac{2}{|J_1|} \sum_{j \in J_1} \left(x_j - 0.8 x_1 \cos\left(6\pi x_1 + \frac{j\pi}{n}\right)\right)^2 \\ g(\mathbf{x}) = 2 - \sqrt{x_1} + \frac{2}{|J_2|} \sum_{j \in J_2} \left(x_j - 0.8 \cos\left(6\pi x_1 + \frac{j\pi}{n}\right)\right)^2 \\ h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^{H(t)} \\ \mathit{where}: \\ H(t) = 0.75 \sin(0.5\pi t) + 1.25, \quad t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_t} \right\rfloor \\ J_1 = \{j |\ \text{j is odd and } 2 \leq j \leq n\} \\ J_2 = \{j |\ \text{j is even and } 2 \leq j \leq n\} \\ x_1 \in [0, 1], \quad x_i \in [-1, 1], \quad \forall i = 2, 3, \ldots, n \end{cases} \tag{36}$$



(a) POS of x_2 with $n_t = 10$ and $\tau_t = 10$ for 1,000 iterations (b) POS of x_5 with $n_t = 10$ and $\tau_t = 10$ for 1,000 iterations

Fig. 10. POS of HE5 for two decision variables: x_2 and x_5.

HE5 is a Type III DMOOP, since the POF changes over time but the POS remains the same. The POS and POF of HE5 are:

$$POS : x_j = \begin{cases} 0.8x_1 \cos\left(6\pi x_1 + \frac{j\pi}{n}\right), & j \in J_1 \\ 0.8x_1 \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), & j \in J_2 \end{cases}$$

$$POF : y = \left(2 - \sqrt{x_1}\right)\left[1 - \left(\frac{x_1}{2 - \sqrt{x_1}}\right)^{H(t)}\right]$$

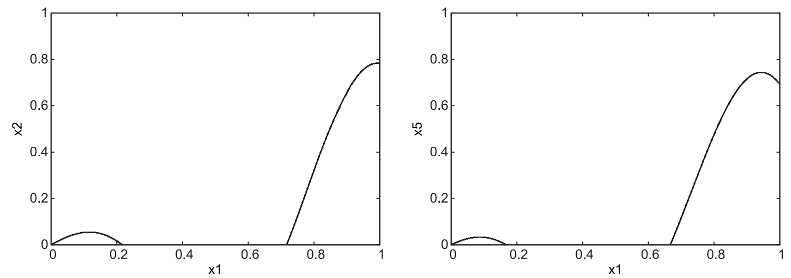
The POS of HE5 is illustrated in Figure 10. The POF is similar to the POF of HE3, illustrated in Figure 7.

$$\text{HE6} = \begin{cases} \textit{Minimise}: \mathbf{f}(\mathbf{x}, t) = (f_1(\mathbf{x}), g(\mathbf{x}, t) \cdot h(f_1(\mathbf{x}), g(\mathbf{x}, t))) \\ f_1(\mathbf{x}) = x_1 + \frac{2}{|J_1|} \sum_{j \in J_1} \left(x_j - 0.8x_1 \cos\left(\frac{6\pi x_1 + \frac{j\pi}{n}}{3}\right)\right)^2 \\ g(\mathbf{x}) = 2 - \sqrt{x_1} + \frac{2}{|J_2|} \sum_{j \in J_2} \left(x_j - 0.8 \cos\left(6\pi x_1 + \frac{j\pi}{n}\right)\right)^2 \\ h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^{H(t)} \\ \textit{where}: \\ H(t) = 0.75 \sin(0.5\pi t) + 1.25, \quad t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_t} \right\rfloor \\ J_1 = \{j | \text{ j is odd and } 2 \leq j \leq n\} \\ J_2 = \{j | \text{ j is even and } 2 \leq j \leq n\} \\ x_1 \in [0, 1], \quad x_i \in [-1, 1], \quad \forall i = 2, 3, \ldots, n \end{cases} \tag{37}$$

For HE6, the POF changes over time but the POS remains the same. Therefore, HE6 is a Type III DMOOP. The POS and POF of HE6 are:

$$POS : x_j = \begin{cases} 0.8x_1 \cos\left(\frac{6\pi x_1 + \frac{j\pi}{n}}{3}\right), & j \in J_1 \\ 0.8x_1 \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), & j \in J_2 \end{cases}$$

$$POF : y = \left(2 - \sqrt{x_1}\right)\left[1 - \left(\frac{x_1}{2 - \sqrt{x_1}}\right)^{H(t)}\right]$$



(a) POS of x_2 with $n_t = 10$ and $\tau_t = 10$ for 1,000 iterations

(b) POS of x_5 with $n_t = 10$ and $\tau_t = 10$ for 1,000 iterations

Fig. 11. POS of HE6 for two decision variables: x_2 and x_5. ©2013 IEEE. Reprinted with permission from Helbig and Engelbrecht [2013a].

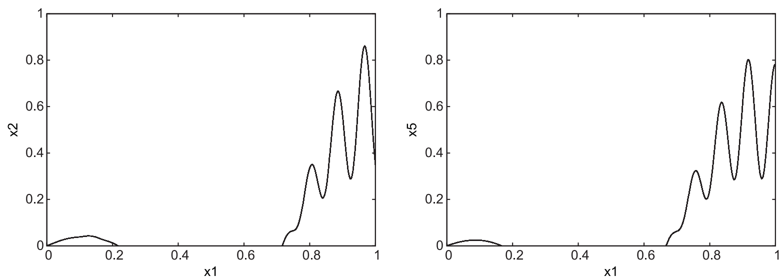
The POF of HE6 is similar to the POF of HE3 (refer to Figure 7). The POS of HE6 is illustrated in Figure 11.

$$
\text{HE7} = \begin{cases}
\textit{Minimise}: \mathbf{f}(\mathbf{x}, t) = (f_1(\mathbf{x}), g(\mathbf{x}, t) \cdot h(f_1(\mathbf{x}), g(\mathbf{x}, t))) \\
f_1(\mathbf{x}) = x_1 + \frac{2}{|J_1|} \sum_{j \in J_1} \left(x_j - [0.3x_1^2 \cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) \right. \\
\qquad \left. + \ 0.6x_1] \cos\left(6\pi x_1 + \frac{j\pi}{n}\right)\right)^2 \\
g(\mathbf{x}) = 2 - \sqrt{x_1} + \frac{2}{|J_2|} \sum_{j \in J_2} \left(x_j - [0.3x_1^2 \cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) \right. \\
\qquad \left. +0.6x_1] \sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right)^2 \\
h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^{H(t)} \\
\textit{where}: \\
H(t) = 0.75 \sin(0.5\pi t) + 1.25, \quad t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_t} \right\rfloor \\
J_1 = \{j | \text{ j is odd and } 2 \le j \le n\} \\
J_2 = \{j | \text{ j is even and } 2 \le j \le n\} \\
x_1 \in [0, 1], \quad x_i \in [-1, 1], \quad \forall i = 2, 3, \ldots, n
\end{cases} \tag{38}
$$

HE7 is a Type III DMOOP, since the POF changes over time but the POS remains the same. The POS and POF of HE7 are:

$$
POS : x_j = \begin{cases}
a \cos\left(\frac{6\pi x_1 + \frac{j\pi}{n}}{3}\right), \quad j \in J_1 \\
a \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), \quad j \in J_2 \\
\text{with:} \\
a = \left[0.3x_1^2 \cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1\right]
\end{cases}
$$

$$
POF : y = (2 - \sqrt{x_1}) \left[1 - \left(\frac{x_1}{2 - \sqrt{x_1}}\right)^{H(t)}\right]
$$



(a) POS of x_2 with $n_t = 10$ and $\tau_t = 10$ for 1,000 iterations (b) POS of x_5 with $n_t = 10$ and $\tau_t = 10$ for 1,000 iterations

Fig. 12. POS of HE7 for two decision variables: x_2 and x_5. ©2013 IEEE. Reprinted with permission from Helbig and Engelbrecht [2013a].

The POS of HE7 is illustrated in Figure 12. The POF is similar to the POF of HE3, as illustrated in Figure 7.

$$\text{HE8} = \begin{cases} \textit{Minimise}\colon \mathbf{f}(\mathbf{x}, t) = (f_1(\mathbf{x}), g(\mathbf{x}, t) \cdot h(f_1(\mathbf{x}), g(\mathbf{x}, t))) \\ f_1(\mathbf{x}) = x_1 + \frac{2}{|J_1|} \sum_{j \in J_1} \left(4y_j^2 - \cos(8y_i\pi) + 1.0\right) \\ g(\mathbf{x}) = 2 - \sqrt{x_1} + \frac{2}{|J_2|} \sum_{j \in J_2} \left(4y_j^2 - \cos(8y_i\pi) + 1.0\right) \\ h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^{H(t)} \\ \textit{where}\colon \\ H(t) = 0.75\sin(0.5\pi t) + 1.25, \quad t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor \\ J_1 = \{j |\ \text{j is odd and } 2 \le j \le n\} \\ J_2 = \{j |\ \text{j is even and } 2 \le j \le n\} \\ y_j = x_j - x_1^{\left(0.5\left(1.0 + \frac{3(j-2)}{n-2}\right)\right)}, \quad \forall j = 2, 3, \ldots, n \\ x_i \in [0, 1], \quad \forall i = 1, 2, \ldots, n \end{cases} \tag{39}$$

$$\text{HE9} = \begin{cases} \textit{Minimise}\colon \mathbf{f}(\mathbf{x}, t) = (f_1(\mathbf{x}), g(\mathbf{x}, t) \cdot h(f_1(\mathbf{x}), g(\mathbf{x}, t))) \\ f_1(\mathbf{x}) = x_1 + \frac{2}{|J_1|} \sum_{j \in J_1} \left(4\sum_{j \in J_1} y_j^2 - 2\prod_{j \in J_1} \cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 2.0\right) \\ g(\mathbf{x}) = 2 - \sqrt{x_1} + \frac{2}{|J_2|} \sum_{j \in J_2} \left(4\sum_{j \in J_2} y_j^2 - 2\prod_{j \in J_2} \cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 2.0\right) \\ h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^{H(t)} \\ \textit{where}\colon \\ H(t) = 0.75\sin(0.5\pi t) + 1.25, \quad t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor \\ J_1 = \{j |\ \text{j is odd and } 2 \le j \le n\} \\ J_2 = \{j |\ \text{j is even and } 2 \le j \le n\} \\ y_j = x_j - x_1^{\left(0.5\left(1.0 + \frac{3(j-2)}{n-2}\right)\right)}, \quad \forall j = 2, 3, \ldots, n \\ x_i \in [0, 1] \ \ \forall i = 1, 2, \ldots, n \end{cases} \tag{40}$$

The POF of HE8 changes over time but the POS remains the same. Therefore, HE8 is a Type III DMOOP. The POS (refer to Figure 8) and POF (refer to Figure 7) of

HE8 are:

$$POS: x_j = x_1^{0.5\left(\frac{3(j-2)}{n-2}\right)}, \quad \forall j = 2, 3, \ldots, n.$$

$$POF: y = \left(2 - \sqrt{x_1}\right)\left[1 - \left(\frac{x_1}{2 - \sqrt{x_1}}\right)^{H(t)}\right]$$

For HE9, the POF changes over time but the POS remains the same. Therefore, HE9 is a Type III DMOOP. The POS (refer to Figure 8) and POF (refer to Figure 7) of HE9 are:

$$POS: x_j = x_1^{0.5\left(\frac{3(j-2)}{n-2}\right)}, \quad \forall j = 2, 3, \ldots, n.$$

$$POF: y = \left(2 - \sqrt{x_1}\right)\left[1 - \left(\frac{x_1}{2 - \sqrt{x_1}}\right)^{H(t)}\right]$$

$$\text{HE10} = \begin{cases} \textit{Minimise}: \mathbf{f}(\mathbf{x}, t) = (f_1(\mathbf{x}), g(\mathbf{x}, t) \cdot h(f_1(\mathbf{x}), g(\mathbf{x}, t))) \\ f_1(\mathbf{x}) = x_1 + \frac{2}{|J_1|}\sum_{j \in J_1}\left(x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right)^2 \\ g(\mathbf{x}) = 2 - x_1^2 + \frac{2}{|J_2|}\sum_{j \in J_2}\left(x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right)^2 \\ h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^{H(t)} \\ \textit{where}: \\ H(t) = 0.75\sin(0.5\pi t) + 1.25, \quad t = \frac{1}{n_t}\left\lfloor\frac{\tau}{\tau_t}\right\rfloor \\ J_1 = \{j|\ \text{j is odd and } 2 \le j \le n\} \\ J_2 = \{j|\ \text{j is even and } 2 \le j \le n\} \\ x_i \in [0, 1]\ \ \forall i = 1, 2, \ldots, n \end{cases} \tag{41}$$

The POF of HE10 changes over time but the POS remains the same. Therefore, HE10 is a type I DMOOP. The POS (refer to Figure 9) and POF (refer to Figure 7) of HE10 are:

$$POS: x_j = sin\left(6\pi x_1 + \frac{j\pi}{n}\right), \quad \forall j = 2, 3, \ldots, n.$$

$$POF: y = \left(2 - \sqrt{x_1}\right)\left[1 - \left(\frac{x_1}{2 - \sqrt{x_1}}\right)^{H(t)}\right]$$

The HE3 to HE10 DMOOPs can be changed from Type III DMOOPs to Type II DMOOPs by changing the h function in Equations (34) through (41) as follows:

$$h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^{H_2(t)}$$

with:

$$H_2(t) = H(t) + \sum_{x_i \in \mathbf{x}_{||}}\left(x_i - H(t)\right)^2, \quad \mathbf{x}_{||} \subset \mathbf{x} \tag{42}$$

This new h function will cause the POS to change over time.

6. EVALUATION OF DMOO ALGORITHMS

This section highlights findings of a study that compares the performance of DMOAs on DMOOPs with various characteristics [Helbig 2012]. Five DMOAs were used for the experiments, namely the Dynamic Nondominated Sorting Genetic Algorithm II

(DNSGA-II)-A [Deb et al. 2007], DNSGA-II-B [Deb et al. 2007], the dynamic cooperative competitive Evolutionary Algorithm (dCOEA) [Goh and Tan 2009b], the Dynamic Multi-objective Particle Swarm Optimisation (DMOPSO) algorithm [Lechuga 2009], and the Dynamic Vector Evaluated Particle Swarm Optimisation (DVEPSO) algorithm [Helbig and Engelbrecht 2013b]. All DMOAs were evaluated on a modified version of DIMP2 with a concave POF, ZJZ (Equation (16)), FDA2 (Equation (6)), FDA2$_{Camara}$ [Cámara et al. 2010], FDA3 (Equation (7)), FDA3$_{Camara}$ [Cámara et al. 2010], FDA5 (Equation (9)), FDA5$_{iso}$ (Equation (31)), FDA5$_{dec}$ (refer to Section 5.2), dMOP2 (Equation (14)), dMOP3 (Equation (15)), dMOP2$_{iso}$ (Equation (32)), dMOP2$_{dec}$ (refer to Section 5.2), HE1 (Equation (19)), HE2 (Equation (20)), HE6 (Equation (37)), HE7 (Equation (38)), and HE9 (Equation (40)). For all benchmark functions, the severity of change frequency of change combination ($n_t - \tau_t$) was set to 1–10, 10–10, 10–25, 10–50, and 20–10. For each DMOOP, the DMOA was executed for 30 runs, with each run consisting of 1,000 iterations. Three performance measures were used, namely the number of nondominated solutions found, accuracy [Cámara Sola 2010] (low value indicates good performance), and stability [Cámara Sola 2010].

The following observations were made:

—DMOAs solving a DMOOP with a discontinuous POF will struggle to find a diverse set of solutions for each of the continuous sections of the POF. Therefore, a DMOA may require a longer period to find a diverse set of solutions. However, the time available depends on the frequency of changes in the environment. Only DMOPSO performed really well on DMOOPs with a discontinuous POF. DVEPSO struggled to converge towards a discontinuous POF. However, the other algorithms managed to find solutions that were relatively close to the true POF. DMOOPs with a discontinuous POF are selected as a characteristic of an ideal benchmark function suite (refer to Table II, item 2).

—Only DVEPSO and dCOEA could solve DIMP2, where each decision variable has its own rate of change. In addition, DVEPSO outperformed dCOEA. In a fast-changing environment, both DMOAs obtained very high accuracy and stability values. Therefore, the found solutions were far from the true POF, and the performance of both DMOAs was severely affected by changes in the environment. DMOOPs with decision variables that change at different rates are selected as a characteristic that an ideal benchmark function suite should exhibit (refer to Table III, item 6).

—The lack of gradient information may cause a DMOA to converge slower to an isolated POF. All DMOAs, except DMOPSO, obtained a better performance for dMOP2$_{iso}$ than the original dMOP2 DMOOP. However, in contrast to dMOP2, when solving FDA5$_{iso,}$ all DMOAs obtained a worse performance than with the original FDA5 DMOOP. Furthermore, dCOEA struggled to converge towards the POF of FDA5$_{iso}$. In a fast-changing environment with severe changes, all DMOAs obtained much larger accuracy and stability values. Therefore, they struggled to find solutions close to the true POF in the available time, and their performance was severely affected by the changes the environment.

—Since the search space favours the local POF, DMOAs take longer to converge to the true POF if the POF is deceptive. Therefore, when the changes in the environment are gradual and occur only occasionally, the DMOAs obtain reasonable accuracy and stability values. However, when the environment changes frequently, the DMOAs' performance degrade, leading to very large accuracy and stability values. On dMOP2$_{dec,}$ all DMOAs except DMOPSO performed much worse than on the original dMOP2. For FDA5$_{dec}$, all DMOAs performed much worse than for the original FDA5 function. In addition, dCOEA struggled to find solutions for FDA5$_{dec}$, even in slow-changing environments. DMOOPs with an isolated or deceptive POF are

Table II. Set of DMOO Benchmark Functions for Each Identified Characteristic for MOOPs in General

Characteristic	DMOOP Type: Suggested DMOOPs
1. DMOOPs that cause difficulties to converge towards the POF:	
— Multimodal DMOOPs	Type I: DMZDT4 [Wang and Li 2009]
— DMOOPs with an isolated optimum	Various: DMOOPs developed according to Section 5.1
— DMOOPs with a deceptive optimum	Various: DMOOPs developed according to Section 5.2
2. DMOOPs that cause difficulties to find a diverse set of solutions:	
— DMOOP with a convex POF	—Type I: FDA1 (Equation (5)), DMZDT1 [Wang and Li 2009] —Type II: Modified FDA3 functions [Zheng 2007; Talukder and Khaled 2008; Khaled et al. 2008; Cámara et al. 2010] —Type III: dMOP1 (Equation (13))
— DMOOPs with a nonconvex POF	— Type I: DMZDT2 [Wang and Li 2009], FDA4 (Equation (8)), DMOP3 [Liu et al. 2010] — Type II: FDA5 (Equation (9)) —Type III: Modified FDA5 [Talukder and Khaled 2008]
— DMOOPs with a discontinuous POF	—Type I: DMZDT3 [Wang and Li 2009] —Type III: HE1 (Equation (19)), HE2 (Equation (20))
— DMOOPs with a nonuniform spread of solutions	—Type I: dMOP3 (Equation (15)) —Type II: FDA5 (Equation (9)), modified FDA3 functions [Zheng 2007; Talukder and Khaled 2008; Khaled et al. 2008; Cámara et al. 2010] —Type III: Modified FDA5 [Talukder and Khaled 2008]
3. DMOOPs with various types or shapes of POSs	— Types I, II: DTLZ$_{A_v}$ (Equation (21)) — Type II: ZJZ (Equation (16)), DSW2 (Equation (28)), DSW3 (Equation (29)) — Type III: HE3 to HE10 (Equations (34) through (41)) —Types II, III: Modified FDA2 [Mehnen et al. 2006; Deb et al. 2007; Liu et al. 2010; Zheng 2007; Isaacs et al. 2008; Ray et al. 2009; Lechuga 2009; Cámara et al. 2009, 2010; Cámara Sola 2010]
4. DMOOPs with dependencies between the decision variables	—Type II: ZJZ (Equation (16))

identified as characteristics of an ideal benchmark function suite (refer to Table II, item 1).

—DMOAs solving a DMOOP with a complex POS will require more time to converge than when they are solving a DMOOP with a simple POS—that is, when all decision variables have the same POS, with the POS being defined by a linear function. When solving HE6, HE7, and HE9 with complicated POSs, the DMOAs struggled to converge to the true POF. For HE9, three of the five DMOAs obtained very high accuracy values, and two DMOAs obtained high accuracy values, indicating a poor performance. In addition, when solving HE6 and HE7, all DMOAs, except dCOEA, obtained high accuracy values. DMOOPs where the POS is a nonlinear function have been identified as a characteristic that an ideal benchmark function suite should exhibit (refer to Table II, item 5).

It should be noted that similar to a lack of standard DMOO benchmark functions, there are no standard DMOO performance measures. Selecting which performance

Table III. Set of DMOO Benchmark Functions for Each Identified Characteristic for DMOOPs

Characteristic	DMOOP
1. DMOOPs where the distribution of solutions in the POF changes over time	— Type I: dMOP3 (Equation (15)) — Type II: FDA5 (Equation (9)), modified FDA3 functions [Zheng 2007; Talukder and Khaled 2008; Khaled et al. 2008; Cámara et al. 2010] — Type III: Modified FDA5 [Talukder and Khaled 2008]
2. DMOOPs where the POF changes from convex to nonconvex and/or vice versa over time	—Type II: dMOP2 (Equation (14)), ZJZ (Equation (16)) — Type III: dMOP1 (Equation (13)) — Types II, III: Modified FDA2 functions [Mehnen et al. 2006; Deb et al. 2007; Liu et al. 2010; Zheng 2007; Isaacs et al. 2008; Ray et al. 2009; Lechuga 2009; Cámara et al. 2009, 2010; Cámara Sola 2010]
3. DMOOPs where the shape of POS changes over time	— Various types: $DTLZ_{Av}$ (refer to Equation (21))
4. DMOOPs with a disconnected POS that changes over time	— Type II: DSW2 (Equation (28))
5. DMOOPs where each decision variable has a different POS that changes over time	— Type III: HE3 to HE10 (Equations (34) through (41))
6. DMOOPs with decision variables that change with different rates over time	— Type I: DIMP1 (Equation (17)), DIMP2 (Equation (18))
7. DMOOPs where the current POF depends on the previous POF or POS	— Type IV: T3 (Equation (24)), T4 (Equation (25))
8. DMOOPs where the number of decision variables vary over time	— Type IV: T1 (Equation (22))
9. DMOOPs where the number of objective functions vary over time	— Types I, II: $DTLZ_{Av}$ (Equation (21)) —Type III: T2 (Equation (23))
10. Real-world DMOOPs	— Refer to Section 8

measures to use to evaluate DMOAs is no trivial task. However, the reader is referred to Helbig [2012] and Helbig and Engelbrecht [2013c], which provide a comprehensive overview of DMOO performance measures and highlight issues with performance measures that are currently used to evaluate DMOAs.

7. IDEAL SET OF DMOO BENCHMARK FUNCTIONS

Taking into consideration the benchmark functions currently being used for DMOO (discussed in Section 4) and the ideal characteristics of benchmark functions (discussed in Section 3), it becomes clear that many different types of DMOOPs have been suggested to be used as benchmark functions. Therefore, when a new DMOO algorithm has been developed, the selection of benchmark functions to test the algorithm's ability to solve DMOOPs in comparison with other algorithms is a daunting task. This section presents the characteristics of an ideal benchmark function set and suggests DMOOPs that can be used to sufficiently test an algorithm's ability to solve DMOOPs.

From Sections 3 and 4, the following characteristics were identified that an ideal MOO (static or dynamic) set of benchmark functions should have [Helbig and Engelbrecht 2013a]:

(1) The set of benchmark functions should test whether an algorithm can converge towards a POF with the following characteristics:

—Multimodality
—Isolated optimum
—Deceptive optimum

(2) The set of benchmark functions should test whether an algorithm can obtain a diverse set of solutions when a POF has the following characteristics:
—Convexity and/or nonconvexity in the POF
—Discontinuous POF (i.e., a POF with disconnected continuous subregions)
—Nonuniform distribution of solutions in the POF

(3) The benchmark functions should have various types (Type I to IV of Farina et al. [2004]) or shapes of POSs, including POSs with nonlinear curves.
(4) The benchmark functions should have decision variables with dependencies (linkages).

In addition, the following characteristics were identified that an ideal DMOO benchmark function suite should have [Helbig and Engelbrecht 2013a]:

(1) The benchmark functions should have a nonuniform distribution of solutions in the POF and/or the distribution of solutions should change over time.
(2) The POFs' shape should change over time from convex to nonconvex and/or vice versa.
(3) The POSs' shape should change over time.
(4) The POS should be disconnected and change over time.
(5) Each decision variable should have a different POS that changes over time.
(6) The benchmark functions should have decision variables that change with different rates over time.
(7) The benchmark functions should include cases where the POF depends on values of POSs or POFs of previous environments.
(8) The benchmark functions should enable varying the number of decision variables over time.
(9) The benchmark functions should enable varying the number of objective functions over time.
(10) A real-world DMOOP

For each characteristic, a set of DMOOPs was identified from Sections 4, 5.1, and 5.2. Tables II and III present the proposed ideal benchmark functions suite from which DMOOPs can be selected to evaluate the performance of DMOAs.

Selection of DMOOPs for a study should be done in such a way that various types of DMOOPs are selected for each characteristic. The reason for this is to ensure that an algorithm can overcome a certain difficulty in various types of DMOO environments.

In order to evaluate whether an algorithm can solve DMOOPs with various change frequencies (τ_t) and change severities (n_t), the following parameter values are suggested: $\tau_t = \{5, 10, 25, 50, 100\}$ and $n_t = \{1, 10, 20\}$, where various combinations of τ_t and n_t values should be used. These parameter values will enable researchers to analyse the performance of the algorithms for specific type of environments—that is, whether a specific algorithm performs well in slow-changing environments, fast-changing environments, or both; gradually changing environments, severely changing environments, or both; and a combination of these listed environment types.

In addition to the benchmark functions listed in Table II, generic benchmark function generators can be used to create DMOOPs of various types with specific characteristics as outlined in this section—for example, DTF (refer to Equation (10)), DTLZ$_{Av}$ (refer to Equation (21)), DSW (refer to Equation (26)), and the DMOOP of Tang et al. (refer to Equation (11)).

8. REAL-WORLD DMOO PROBLEMS

Normally, the degree of difficulty of a real-world problem is unknown beforehand. Furthermore, in many cases, the true POF of a real-world problem is unknown. Therefore, artificial problems or benchmark functions are used to evaluate the performance of an algorithm. The benchmark functions exhibit certain characteristic and therefore test whether an algorithm can overcome specific difficulties.

Numerous real-world DMOOP application areas exist, of which some are hydrothermal power scheduling [Deb et al. 2007], machining of gradient material [Roy and Mehnen 2008], controller design for a time-varying unstable plant [Farina et al. 2004; Huang et al. 2011], war resource allocation [Palaniappan et al. 2001], route optimisation according to real-time traffic [Wahle et al. 2001], design optimisation of wind turbine structures [Maalawi 2011], supply chain networks [Chen and Lee 2004; Selim et al. 2008], and energy-efficient routing optimisation in mobile ad hoc networks [Constantinou 2011].

In this section, four real-world DMOOPs of various application areas are discussed, namely the regulation of a lake-river system, the optimisation of a heating system, the control of a greenhouse, and the management of hospital resources.

8.1. Regulation of a Lake-River System

Hämäläinen and Mäntysaari [2001] proposed a mathematical model to regulate a lake-river system that consists of four lakes and a river that connects the lakes to the sea. The DSOOP in Hämäläinen and Mäntysaari [2001] is adapted to a DMOOP as follows:

$$\begin{aligned} \min\ \ & \mathbf{f} = (f_1,\ f_2) \\ f_1 &= \sum_{k\in K} c_g g_k + \sum_{i=1}^{n} p_i \\ f_2 &= \sum_{k\in K} c_I I_k + \sum_{i=1}^{n} p_i \end{aligned}$$

with:

$$\begin{aligned} g_k &= \left(x_k^{goal} - x_k\right)^2 \\ I_k &= \begin{cases} (I_k - x_k)^2, & \text{if } \ x_k < I_k \\ (x_k - u_k)^2, & \text{if } \ x_k > I_k \\ 0, & \text{otherwise} \end{cases} \\ p_i &= \begin{cases} c_1\left(|q_i - q_{i-1}| - \Delta q_{max}\right)^2 + c_2\left(q_i - 1_{i-1}\right)^2, & \text{if } \ |q_i - q_{i-1}| > \Delta q_{max} \\ c_2\left(q_i - q_{i-1}\right)^2, & \text{otherwise} \end{cases} \end{aligned}$$

where K is the set of goal observation indexes of the planning period; c_1 and c_2 are adjustable parameters; g_k is the deviation from the goal point; x_k^{goal} is the goal; x_k is the true water level; I_k is the deviation from the goal set; l_k and u_k are the lower and upper bounds of the goal x_k, respectively; p_i is a penalty function; q_i is the outflow from Lake Päijänne; Δq_{max} is the upper limit of the change in flow rate; and i refers to the discretized time interval. The following parameter values are suggested [Hämäläinen and Mäntysaari 2001]: $c_g = 10L/m^2$, $c_I = 100L/m^2$, $c_1 = 100s^2/m^6$, and $c_2 = 0.00001s^2/m^6$.

8.2. Heating Optimisation

Hämäläinen and Mäntysaari [2002] proposed a DMOOP to optimise indoor heating. The DMOOP is defined as:

$$\begin{aligned} &\min \mathbf{f} = (f_1,\ f_2,\ f_3) \\ &f_1 = \sum_{i=0}^{n-1} p_i q_i \\ &f_2 = \sum_{i=0}^{n-1} q_i \\ &f_3 = \sum_{i=0}^{n-1} \left| T_i - T_i^{ideal} \right| \end{aligned} \tag{44}$$

with:

$$\begin{aligned} &T_0 = T_n \qquad (45) \\ &l_i \leq T_i \leq u_i,\ \ \forall i = 0, \ldots, n-1 \\ &0 \leq q_i \leq q, \quad\ \forall i = 0, \ldots, n-1, \end{aligned}$$

where f_1 represents heating costs; f_2 represents heating energy; f_3 represents deviation from the ideal temperature; T_i represents indoor temperature that is a state variable; q_i represents the heating power at time i and is a decision variable; T_0 is the initial indoor temperature; l_i and u_i are the lower and upper bounds of T_i, respectively; q is the maximum heating capacity of the heating system; p_i is the hourly price of electricity at time i; and T_i^{ideal} is the hourly ideal indoor temperature specified by the decision maker.

The constraint in Equation (45) specifies that the indoor temperature of the first hour of the day has to be the same on the following day. This constraint can be managed by either only accepting solutions that adhere to this constraint or by converting the constraint to a penalty function.

8.3. Control of a Greenhouse

Ursem et al. [2002] proposed a mathematical model to describe the state transformation of a greenhouse for crops as a DSOOP. Zhang [2007] proposed a DMOOP to optimisation of the control of a greenhouse sytem based on the model proposed by Ursem et al. [2002]:

$$\begin{aligned} &\min_{\mathbf{U}(k) \in \mathbf{U}_{ad}} \mathbf{f}(\mathbf{U}(k)) = (-f_1,\ f_2,\ f_3) \\ &f_1(\mathbf{U}(k)) = \frac{1}{l} \sum_{j=k}^{k+l} v_{pcrop}(j) \Delta x(j) \\ &f_2(\mathbf{U}(k)) = \frac{1}{l} \sum_{j=k}^{k+l} v_{pheat}(j) u_{heat}(j) \\ &f_3(\mathbf{U}(k)) = \frac{1}{l} \sum_{j=k}^{k+l} v_{CO_2}(j) u_{pCO_2}(j) \end{aligned} \tag{46}$$

subject to:

$$
\begin{aligned}
x_{temp}(k+1) &= x_{temp}(k) + \Delta x_{temp}(k), x_{CO_2}(k+1) = x_{CO_2}(k) + \Delta x_{CO_2}(k)\\
x_{crop}(k+1) &= x_{crop}(k) + \Delta x_{crop}(k),\ y_{out}(k+1) = \Delta x_{crop}(k)\\
x_{temp}(0) &= 18;\ \ x_{CO_2}(0) = 1;\ \ x_{crop}(0) = 4;\ \ 16 \le x_{temp}(k) \le 35, \forall k \qquad (47)\\
\Delta x_{temp}(k) &= u_{heat}(k-1) + t_1 + u_{vent}(k-1)\big[v_{temp}(k-1) - x_{temp}(k-1)\big] + k_2 v_{sun}(k-1)\\
\Delta x_{crop}(k) &= \min(\max(k_5 - |x_{temp}(k-1) - k_6|, 0), \min(x_{CO_2}, k_7), \min(v_{sun}(k-1), t_8))\\
&\quad - k_9 \min(k_5 - |x_{temp}(k-1) - k_6|, 0)
\end{aligned}
$$

with:

$$
\begin{aligned}
&\mathbf{U}_{ad} = \{\mathbf{U}(k) = \big(\mathbf{u}_0(k), \mathbf{u}_1(k), \ldots, \mathbf{u}_{l-1}(k)\big)\, |\mathbf{u}_j(k) = \big(u_{heat}(k+j), u_{vent}(k+j), u_{CO_2}(k+j)\big)\,,\\
&\qquad \mathbf{u}_j(k) \in [0,5]\text{x}[0,1]\text{x}[0,4];\ \ 0 \le j \le l-1\}\\
&v_{temp}(k) = 10 + v_{temp,p}(k) + v_{temp,st}(k),\ v_{sun}(k) = 1.0 + v_{sun,p}(k) + v_{sun,st}(k)\\
&v_{pcrop}(k) = 22.0 + v_{pcrop,p}(k) + v_{pcrop,st}(k),\ v_{pheat}(k) = 2.5 + v_{pheat,st}(k),\ v_{pCO_2}(k)\\
&\qquad = 2.5 + v_{pCO_2,st}(k),\ t_0 = -10\\
&v_{temp,p}(k) = 7\cos(2\pi 10^{-2} t_k) + 9\cos(2\pi 10^{-3} t_k)\\
&v_{sum,p}(k) = 4\cos(2\pi 10^{-2} t_k) + 9\cos(2\pi 10^{-3} t_k)\\
&v_{pcrop,p}(k) = -3cos(2\pi 10^{-3} t_k)\\
&v_{temp,st}(k) = \min(\max(v_{temp,st}(k-1) + U(-0.5, 0.5), -4), 4)\\
&v_{sun,st}(k) = \min(\max(v_{sun,st}(k-1) + U(-0.25, 0.25), -1, 1)\\
&v_{pcrop,st}(k) = \min(\max(v_{pcrop,st}(k-1) + U(-10^{-2}, 10^{-2}), -5), 5)\\
&v_{pheat,st}(k) = \min(\max(v_{pheat,st}(k-1) + U(-10^{-3}, 10^{-3}), -0.5), 0.5)\\
&v_{pCO_2,st}(k) = \min(\max(v_{pCO_2,st}(k-1) + U(-10^{-3}, 10^{-3}), -0.5), 0.5)\\
&v_{temp,st}(0) = 0; v_{sun,st}(0) = 0; v_{pheat,st}(0) = 0; v_{pCO_2,st}(0) = 0; v_{pcrop,st}(0) = 0\\
&u_{heat} \in [0,5]; u_{vent} \in [0,1]; u_{CO_2} \in [0,4]\\
&v_{temp} \in [-20, 40]; v_{sun} \in [0, 8]; v_{pcrop} \in [0, 30]; v_{pheat}, v_{pCO_2} \in [0, 3]\\
&x_{temp} \in [-20, 50]; x_{CO_2} \in [0, 10]; x_{crop} \in [0, \infty),
\end{aligned}
$$

where t_k is the time when the greenhouse is in the k-th step, $U(a, b)$ is a stochastic variable with an uniform distribution over [a,b], l is the prediction timestep size, k_1 is the smallest coefficient of heat transformation, k_2 is the sun absorption rate of the greenhouse, k_3 is the increment rate at which the crop consumes CO_2, k_4 is the density of CO_2 outdoors, k_5 is the maximum crop output, k_6 is moderate temperature that results in the best crop growth, k_7 is the maximum quantity of CO_2 that the crop consumes, k_8 is the maximum intensity of the sun that results in crop growth, and k_9 is the loss rate that results in severe temperatures. The controller consists of three variables, namely heat (u_{heat}), ventilation (u_{vent}), and CO_2 (u_{CO_2}). Five variables are considered for the environmental system, namely environmental temperature (v_{temp}), intensity of the sun ($v_s un$), prices of the crop (v_{pcrop}), heat (v_{pheat}), and CO_2 (v_{pCO_2}). In addition, the greenhouse has three indoor state variables, namely temperature (x_{temp}), density of CO_2 (x_{CO_2}), and crop quantity (x_{crop}).

The constraint, $16 \le x_{temp} \le 35$ (refer to Eq. (47)), is transformed into a subobjective function:

$$f_4(\mathbf{U}(k)) = \sum_{j=k}^{k+l} \left\{[\max(16 - x_{temp}(j), 0)]^2 + [\min(35 - x_{temp}(j), 0]^2\right\} \tag{48}$$

Therefore, the DMOOP of Equation (46) is converted to the following four-objective DMOOP:

$$\min_{\mathbf{U}(k)\in\mathbf{U}_{ad}(k)} \left(-f_1(\mathbf{U}(k)),\ f_2(\mathbf{U}(k)),\ f_3(\mathbf{U}(k)),\ f_4(\mathbf{U}(k))\right) \tag{49}$$

The following parameter values are suggested in Zhang [2007]: $k_1 = 0.1$, $k_2 = 0.2$, $k_3 = 1$, $k_4 = 4$, $k_5 = 8$, $k_6 = 26$, $k_7 = 8$, $k_8 = 7$, and $k_9 = 0.1$.

8.4. Hospital Resource Management

Hutzschenreuter et al. [2009] proposed a DMOOP to model the management of hospital resources, defined as follows:

$$\begin{aligned} &\min \mathbf{f}(\pi) = (-f_1(\pi),\ f_2(\pi),\ f_3(\pi)) \\ &f_2(\pi) = \sum\sum c_u \pi_u(t_i, s_u) + c_{CTS-OR} uc_{CTS-OR}(\pi) \end{aligned} \tag{50}$$

with:

$$\begin{aligned} &s_u(t_i) = \frac{\text{utilised capacity at unit } u \text{ at start of day } t_i}{r_u(t_i^-)} \\ &p_i = \begin{cases} \max\ \left\{r_u^{min}, r_u(t_i^-) - r_u^{decr}\right\}, & \text{if } \ s_u(t_i) < UT_u^{decr} \\ r_u(t_i^-), & \text{if } \ s_u(t_i) \in \left[UT_u^{decr}, UT_u^{incr}\right] \\ \min\ \left\{r_u^{max}, r_u(t_i^-) + r_u^{incr}\right\}, & \text{otherwise} \end{cases} \\ &\pi_u(t_0, s_u) = r_u^{base} \end{aligned}$$

$$\begin{aligned} &r_u^{base} \in \mathbb{N} \bigcap \left| r_u^{min}, r_u^{max} \right|, \quad \forall u \in U \\ &s_u(t_i) \in \mathbb{R}_0^+, \quad \forall u \in U, \quad \forall t_i \in T' \\ &r_u^{decr}, r_u^{incr} \in [0, 5], \quad \forall u \in U \\ &UT_u^{decr} \in [0, 1], UT_u^{incr} \in \left[UT_u^{decr}, UT_u^{decr} + 1\right], \quad \forall u \in U, \end{aligned}$$

where T is the time horizon with discrete time units t and n equidistant decision moments denoted by $t_i \in T'$ with $t_{i-1} < t_i \forall i = 1, \ldots, n-1$ (typically t will be in steps of hours and t_i will be in steps of days), π is a resource allocation policy, $f_1(\pi)$ is the mean total throughput of patients under π defined as the number of patients discharged from the hospital after treatment, $f_2(\pi)$ is the mean total resource cost, $f_3(\pi)$ is the mean total weighted backup capacity usage under allocation π, uc_{CTS-OR} is the unused Cardio-Thoracic Surgery Operating Room (CTS-OR) capacity due to cancelled surgeries resulting from unavailable postoperative care beds give π, $s_u(t_i)$ is the state at unit u at decision moment i, $r_u(t_i^-)$ is the resource capacity of unit u at time t_i^- that is just before the adjustment at time t_i, r_u^{base} is the base resource allocation, r_u^{decr} and r_u^{incr} are resource adjustments, UT_u^{decr} and UT_u^{incr} are utilisation adjustments with $UT_u^{decr} \le UT_u^{incr}$, and $\pi_u(t_i, s_i) \in [r_i^{min}, r_i^{max}] \forall t_i \in T', u \in U$.

9. CONCLUSION

In recent years, many DMOOPs have been proposed in the DMOO literature. In addition, no standard benchmark functions exist to evaluate the performance of DMOAs. However, no comprehensive overview of the proposed DMOOPs exists. Therefore, it is a daunting task to select DMOAs for empirical studies.

This article seeked to address this gap in the literature by providing a comprehensive overview of the benchmark functions that have been used in the DMOO literature. In addition, characteristics that an ideal DMOO benchmark function suite should exibit were proposed, and DMOOPs were suggested for each of these characteristics. The suggested ideal benchmark function suite should enable a uniform comparison of DMOAs.

The investigation of the DMOOPs presented in the literature highlighted the following shortcomings of DMOOPs: no DMOOPs have a deceptive or isolated POF, for most DMOOPs the POS is the same for each decision variable and the POS is a simple function (such as $x_i = |\sin(0.5\pi t)|$).

To address these shortcomings, this article presented an approach to adapt existing DMOOPs in such a way that the DMOOPs have either a deceptive or an isolated POF. Furthermore, new DMOOPs were proposed where the POS is a nonlinear function and the POS varies for each decision variable.

In addition, DMOO application areas were highlighted, and four real-world DMOOPs were discussed in more detail, namely the regulation of a lake-river system, the optimisation of a heating system, the control of a greenhouse, and the management of hospital resources.

APPENDIX

B. CALCULATING THE TRUE POS AND POF

This section discusses how *POS* and *POF* are determined for DMOOPs. One example is provided, namely FDA2 modified by Cámara et al. [2009, 2010] and Cámara Sola [2010] referred to in this section as $\text{FDA2}_{Cámara}$.

The $\text{FDA2}_{Cámara}$ DMOOP has two objective functions (refer to Section 4) and is defined as:

$$\begin{cases} Minimise\colon f(\mathsf{x}, t) = (f_1(\mathbf{x}_{\mathsf{I}}), g(\mathbf{x}_{\mathsf{II}}) \cdot h\left(\mathbf{x}_{\mathsf{III}},\ f_1(\mathbf{x}_{\mathsf{I}}), g\left(\mathbf{x}_{\mathsf{II}}\right), t\right)) \\ f_1(\mathbf{x}_{\mathsf{I}}) = x_1 \\ g(\mathbf{x}_{\mathsf{II}}) = 1 + \sum_{x_i \in \mathbf{x}_{\mathsf{II}}} x_i^2 \\ h(\mathbf{x}_{\mathsf{III}},\ f_1, g, t) = 1 - \left(\frac{f_1}{g}\right)^{H_2(t)} \\ where\colon \\ H(t) = z^{-\cos(\pi t/4)},\quad t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor \\ H_2(t) = H(t) + \sum_{x_i \in \mathbf{x}_{\mathsf{III}}} (x_i - H(t)/2)^2 \\ \mathbf{x}_{\mathsf{I}} \in [0, 1];\quad \mathbf{x}_{\mathsf{II}_i}, \mathbf{x}_{\mathsf{III}_i} \in [-1, 1] \end{cases}$$

The goal when solving $\text{FDA2}_{Cámara}$ is to minimise the two objective functions, namely f_1 and $f_2 = gh$. Since f_1 only depends on x_1, the true POF depends on f_2. In order to minimise gh, both g and h have to be minimised. h will be minimised if the term $\frac{f_1}{g}^{H_2(t)}$ is maximised (since this term is subtracted from 1). The term $\frac{f_1}{g}^{H_2(t)}$ is maximised if g is minimised (since f_1 is divided by g). g is minimised if the term $\sum_{x_i \in \mathbf{x}_{\mathsf{II}}} x_i^2$ is minimised—that is, if $\sum_{x_i \in \mathbf{x}_{\mathsf{II}}} x_i^2$ is zero. Therefore, the optimal values for $x_i \in \mathbf{x}_{\mathsf{II}}$ is

$x_i = 0$. If $\sum_{x_i \in \mathbf{x}_{\text{II}}} x_i^2 = 0$, $g = 1$. Replacing $g = 1$ into $f_2 = gh$ results in $f_2^* = 1 - f_1^{H_2(t)}$. In order to minimise f_2^*, $H_2(t)$ has to be minimised. $H_2(t)$ is minimised if the term $\sum_{x_i \in \mathbf{x}_{\text{III}}} (x_i - H(t)/2)^2$ is minimised, which results in $H_2^*(t) = H(t)$. Therefore, the optimal values of $x_i \in \mathbf{x}_{\text{III}}$ is $x_i = \frac{H(t)}{2}$. Replacing H_2 in f_2^* with H_2^* results in $f_2 = 1 - f_1^{H(t)}$.

Therefore, *POF* is $1 - f_1^{H(t)}$. The decision variable values that lead to *POF* is *POS*, namely $x_i = 0, \forall x_i \in \mathbf{x}_{\text{II}}$ and $x_i = \frac{H(t)}{2}, \forall x_i \in \mathbf{x}_{\text{III}}$.

REFERENCES

P. Amato and M. Farina. 2005. An ALife-Inspired evolutionary algorithm for dynamic multiobjective optimization problems. In *Soft Computing: Methodologies and Applications*, F. Hoffmann, M. Köppen, F. Klawonn, and R. Roy (Eds.). Advances in Soft Computing, Vol. 32. Springer, Berlin, 113–125.

Z. Avdagić, S. Konjicija, and S. Omanović. 2009. Evolutionary approach to solving non-stationary dynamic multi-objective problems. In *Foundations of Computational Intelligence Volume 3*, A. Abraham, A.-E. Hassanien, P. Siarry, and A. Engelbrecht (Eds.). Studies in Computational Intelligence, Vol. 203. Springer, Berlin, 267–289.

Z. Bingul. 2007. Adaptive genetic algorithms applied to dynamic multiobjective problems. *Applied Soft Computing* 7 (2007), 791–799.

T. Blackwell and J. Branke. 2006. Multiswarms, exclusion, and anti-convergence in Dynamic environments. *IEEE Transactions on Evolutionary Computation* 10, 4 (aug 2006), 459–472.

J. Branke. 1999. Memory enhanced evolutionary algorithms for changing optimization problems. In *Proceedings of the Congress on Evolutionary Computation*, Vol. 3. IEEE, Washington DC, 1875–1882.

M. Cámara, J. Ortega, and F. J. de Toro. 2007a. Parallel processing for multi-objective optimization in dynamic environments. *International Parallel and Distributed Processing Symposium* 0 (2007), 243–250.

M. Cámara, J. Ortega, and F. de Toro. 2007b. The parallel single front genetic algorithm (PSFGA) in dynamic multi-objective optimization. In *Computational and Ambient Intelligence*, F. Sandoval, A. Prieto, J. Cabestany, and M. Gran̈a (Eds.). Lecture Notes in Computer Science, Vol. 4507. Springer, Berlin, 300–307.

M. Cámara, J. Ortega, and F. de Toro. 2009. A single front genetic algorithm for parallel multi-objective optimization in dynamic environments. *Neurocomputing* 72, 16–18 (2009), 3570–3579.

M. Cámara, J. Ortega, and F. de Toro. 2010. Approaching dynamic multi-objective optimization problems by using parallel evolutionary algorithms. In *Advances in Multi-Objective Nature Inspired Computing*, C. Coello Coello, C. Dhaenens, and L. Jourdan (Eds.). Studies in Computational Intelligence, Vol. 272. Springer, Berlin, 63–86.

M. Cámara Sola. 2010. *Parallel Processing for Dynamic Multi-objective Optimization*. PhD Dissertation. Dept. of Computer Architecture and Computer Technology, Universidad de Granada, Spain.

C.-L. Chen and W.-C. Lee. 2004. Multi-objective optimization of multi-echolon supply chain networks with uncertain product demands and prices. *Computers and Chemical Engineering* 28 (2004), 1131–1144.

H. Chen, M. Li, and X. Chen. 2009. Using diversity as an additional-objective in dynamic multi-objective optimization algorithms. *Electronic Commerce and Security, International Symposium* 1 (2009), 484–487.

J. Y. Chia, C. K. Goh, V. A. Shim, and K. C. Tan. 2012. A data mining approach to evolutionary optimization of noisy multi-objective problems. *International Journal of System Science* 43, 7 (Jul 2012), 1217–1247.

D. Constantinou. 2011. *Ant Colony Optimisation Algorithms for Solving Multi-objective Power Aware Metrics for Mobile Ad Hoc Networks*. PhD Dissertation. Department of Computer Science, University of Pretoria, South Africa.

K. Deb. 1999. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation* 7, 3 (1999), 205–230.

K. Deb. 2004. *Multi-objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Ltd.

K. Deb, U. B. Rao N., and S. Karthik. 2007. Dynamic multi-Objective optimization and decision-making using modied NSGA-II: A case study on hydro-thermal power scheduling. In *Proceedings of International Conference on Evolutionary Multi-criterion optimization*. Lecture Notes in Computer Science, Vol. 4403. Springer, Berlin, 803–817.

K. Deb, A. Sinha, and S. Kukkonen. 2006. Multi-objective test problems, linkages, and evolutionary methodologies. In *Proceedings of the Conference on Genetic and Evolutionary Computation*. ACM, New York, NY, 1141–1148.

K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. 2002. Scalable multi-objective optimization test problems. In *Proceedings of Congress on Evolutionary Computation*. IEEE, Honolulu, 825–830.

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

C. K. Goh, K. C. Tan, C. Y. Cheong, and Y. S. Ong. 2010. An investigation on noise-induced features in robust evolutionary multi-objective optimization. *Expert Systems with Applications* 37, 8 (2010), 5960–5980.

C.-K. Goh and K. C. Tan. 2009a. A coevolutionary paradigm for dynamic multi-objective optimization. In *Evolutionary Multi-objective Optimization in Uncertain Environments*. Studies in Computational Intelligence, Vol. 186. Springer, Berlin, 153–185.

C.-K. Goh and K. C. Tan. 2009b. A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization. *IEEE Transactions on Evolutionary Computation* 13, 1 (Feb 2009), 103–127.

C.-K. Goh and K. C. Tan. 2009c. *Evolutionary Multi-objective Optimization in Uncertain Environments: Issues and Algorithms*. Studies in Computational Intelligence, Vol. 186. Springer-Verlag, Berlin.

M. Greeff and A. P. Engelbrecht. 2008. Solving dynamic multi-objective problems with vector evaluated particle swarm optimisation. In *Proceedings of World Congress on Computational Intelligence: Congress on Evolutionary Computation*. IEEE, 2917–2924.

M. Greeff and A. Engelbrecht. 2010. Dynamic multi-objective optimisation using PSO. In *Multi-Objective Swarm Intelligent Systems*, Nadia Nedjah, Leandro dos Santos Coelho, and Luiza de Macedo Mourelle (Eds.). Studies in Computational Intelligence, Vol. 261. Springer, Berlin, 105–123.

S.-U. Guan, Q. Chen, and W. Mo. 2005. Evolving dynamic multi-objective optimization problems with objective replacement. *Artificial Intelligence Review* 23, 3 (2005), 267–293.

R. P. Hämäläinen and J. Mäntysaari. 2001. A dynamic interval goal programming approach to the regulation of a lake-river system. *Journal of Multi-criteria Decision Analysis* 10, 2 (2001), 75–86.

R. P. Hämäläinen and J. Mäntysaari. 2002. Dynamic multi-objective heating optimization. *European Journal of Operational Research* 142 (2002), 1–15.

I. Hatzakis and D. Wallace. 2006. Dynamic multi-objective optimization with evolutionary algorithms: A forward-looking approach. In *Proceedings of the Conference on Genetic and Evolutionary Computation*. ACM, New York, NY, 1201–1208.

M. Helbig. 2012. *Solving Dynamic Multi-objective Optimisation Problems Using Vector Evaluated Particle swarm Optimisation*. PhD Dissertation. Department of Computer Science, University of Pretoria, South Africa.

M. Helbig and A. P. Engelbrecht. 2011. Archive management for dynamic multi-objective optimisation problems using vector evaluated particle swarm optimisation. In *Proceedings of Congress on Evolutionary Computation*. IEEE, 2047–2054.

M. Helbig and A. P. Engelbrecht. 2013a. Benchmarks for dynamic multi-objective optimisation. In *Proceedings of IEEE Symposium Series on Computational Intelligence*. IEEE, Singapore, 84–91.

M. Helbig and A. P. Engelbrecht. 2013b. *Metaheuristics for Dynamic Optimization*. Springer-Verlag, Berlin, Chapter Dynamic multi-objective optimization using PSO, 147–188. Retrieved December 5, 2012 from http://link.springer.com/chapter/10.1007/978-3-642-30665-5_8.

M. Helbig and A. P. Engelbrecht. 2013c. Performance measures for dynamic multi-objective optimisation algorithms. *Information Sciences* 250 (2013), 61–81.

L. Huang, I. H. Suh, and A. Abraham. 2011. Dynamic multi-objective optimization based on membrane computing for control of time-varying unstable plants. *Information Sciences* 181, 11 (2011), 2370–2391.

S. Huband, P. Hingston, L. Barone, and L. While. 2006. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation* 10, 5 (Oct 2006), 477–506.

A. K. Hutzschenreuter, P. A. N. Bosman, and H. L. Poutré. 2009. Evolutionary multiobjective optimization for dynamic hospital resource management. In *Proceedings of the Conference on Evolutionary Multicriterion Optimization*. Lecture Notes in Computer Science, Vol. 5467. Springer, Berlin, 320–334.

A. Isaacs, V. Puttige, T. Ray, W. Smith, and S. Anavatti. 2008. Development of a memetic algorithm for dynamic multi-objective optimization and its applications for online neural network modeling of UAVs. In *Proceedings of World Congress on Computational Intelligence: International Joint Conference on Neural Networks*. IEEE, 548–554.

A. Isaacs, T. Ray, and W. Smith. 2009. Memetic algorithm for dynamic bi-objective optimization problems. In *Proceedings of Congress on Evolutionary Computation*. IEEE, 1707–1713.

Y. Jin and B. Sendhoff. 2004. Constructing dynamic optimization test problems using the multi-objective optimization concept. In *Applications of Evolutionary Computing*, G. Raidl et al. (Eds.). Lecture Notes in Computer Science, Vol. 3005. Springer, Berlin, 525–536.

A. K. M. Khaled, A. Talukder, and M. Kirley. 2008. A Pareto following variation operator for fast-converging multiobjective evolutionary algorithms. In *Proceedings of World Congress on Computational Intelligence: Congress on Evolutionary Computation*. IEEE, 2270–2277.

W. Koo, C. Goh, and K. Tan. 2010. A predictive gradient strategy for multiobjective evolutionary algorithms in a fast changing environment. *Memetic Computing* 2, 2 (2010), 87–110.

M. S. Lechuga. 2009. *Multi-objective Optimisation Using Sharing in Swarm Optimisation Algorithms*. PhD Dissertation. University of Birmingham. Retrieved from http://etheses.bham.ac.uk/303/.

H. Li and Q. Zhang. 2006. A multiobjective differential evolution based on decomposition for multiobjective optimization with variable linkages. In *Parallel Problem Solving from Nature - PPSN IX*, T. Runarsson, H.-G. Beyer, E. Burke, J. Merelo-Guervós, L. Whitley, and X. Yao (Eds.). Lecture Notes in Computer Science, Vol. 4193. Springer, Berlin, 583–592.

H. Li and Q. Zhang. 2009. Multiobjective optimization problems with complicated Pareto Sets, MOEA/D and NSGA-II. *IEEE Transactions on Evoluationary Computation* 13, 2 (2009), 284–302.

X. Li, J. Branke, and M. Kirley. 2007. On performance metrics and particle swarm methods for dynamic multiobjective optimization problems. In *Proceedings of Congress on Evolutionary Computation*. IEEE, 576–583.

C.-A. Liu. 2010. New dynamic multiobjective evolutionary algorithm with core estimation of distribution. In *Proceedings of the International Conference on Electrical and Control Engineering.* 1345–1348.

C.-A. Liu and Y. Wang. 2006. New evolutionary algorithm for dynamic multiobjective optimization problems. In *Advances in Natural Computation*, L. Jiao, L. Wang, X.-B. Gao, J. Liu, and F. Wu (Eds.). Lecture Notes in Computer Science, Vol. 4221. Springer, Berlin, 889–892.

C.-A. Liu and Y. Wang. 2007. Dynamic multi-objective optimization evolutionary algorithm. In *Proceedings of the International Conference on Natural Computation* 4 (2007), 456–459.

R. Liu, W. Zhang, L. Jiao, F. Liu, and J. Ma. 2010. A sphere-dominance based preference immune-inspired algorithm for dynamic multi-objective optimization. In *Proceedings of the Conference on Genetic and Evolutionary Computation*. ACM, New York, NY, 423–430.

K. Maalawi. 2011. *Wind Turbines*. InTech, Special issue on design optimization of wind turbine structures, 3–24.

J. Mehnen, G. Rudolph, and T. Wagner. 2006. *Evolutionary Optimization of Dynamic Multiobjective Functions*. Technical Report CI-204/06. Universität Dortmund, Germany.

R. W. Morrison and K. A. De Jong. 1999. A test problem generator for non-stationary environments. In *Proceedings of the Congress on Evolutionary Computation*, Vol. 3. IEEE, Washington DC, 2047–2053.

S. Palaniappan, S. Zein-Sabatto, and A. Sekmen. 2001. Dynamic multiobjective optimization of war resource allocation using adaptive genetic algorithms. In *Proceedings of IEEE SoutheastCon*. IEEE, 160–165.

T. Ray, A. Isaacs, and W. Smith. 2009. A memetic algorithm for dynamic multiobjective optimization. In *Multi-Objective Memetic Algorithms*, C.-K. Goh, Y.-S. Ong, and K. Tan (Eds.). Studies in Computational Intelligence, Vol. 171. Springer, Berlin, 353–367.

R. Roy and J. Mehnen. 2008. Dynamic multi-objective optimisation for machining gradient materials. *CIRP Annals - Manufacturing Technology* 57, 1 (2008), 429–432.

J. D. Schaffer. 1985. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the International Conference on Genetic Algorithms*. L. Erlbaum Associates Inc., Hillsdale, NJ, 93–100.

H. Selim, C. Araz, and I. Ozkarahan. 2008. Collaborative production-distribution planning in supply chain: a fuzzy goal programming approach. *Transportation Research Part E* 44, 3 (2008), 396–419.

R. Shang, L. Jiao, M. Gong, and B. Lu. 2005. Clonal selection algorithm for dynamic multiobjective optimization. In *Computational Intelligence and Security*, Y. Hao, J. Liu, Y. Wang, Y.-M. Cheung, H. Yin, L. Jiao, J. Ma, and Y.-C. Jiao (Eds.). Lecture Notes in Computer Science, Vol. 3801. Springer, Berlin, 846–851.

A. K. M. Talukder and A. Khaled. 2008. Towards high speed multiobjective evolutionary optimizers. In *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, New York, NY, 1791–1794.

K. Tan and C. Goh. 2008. Handling uncertainties in evolutionary multi-objective optimization. In *Computational Intelligence: Research Frontiers*, Jacek Zurada, Gary Yen, and Jun Wang (Eds.). Lecture Notes in Computer Science, Vol. 5050. Springer, Berlin, 262–292.

K. C. Tan, T. H. Lee, Y. H. Chew, and L. H. Lee. 2003. A multiobjective evoluationary algorithm for solving truck and trailer vehicle routing problems. In *Proceedings of the Congress on Evolutionary Computation*. IEEE, 2134–2141.

M. Tang, Z. Huang, and G. Chen. 2007. The construction of dynamic multi-objective optimization test functions. In *Proceedings of International Conference on Advances in Computation and Intelligence*. Springer-Verlag, Berlin, 72–79.

R. K. Ursem, T. Krink, and B. Filipic. 2002. *A numerical simulator of a crop-producing greenhouse*. Technical Report EVALife 2002-01. EVALife, Department of Computer Science, University of Aarhus, Ny Munkegade, Aarhus, Denmark.

J. Wahle, O. Annen, Ch. Schuster, L. Neubert, and M. Schreckenberg. 2001. A dynamic route guidance system based on real traffic data. *European Journal of Operational Research* 131 (2001), 302–308.

Y. Wang and C. Dang. 2008. An evolutionary algorithm for dynamic multi-objective optimization. *Appl. Math. Comput.* 25 (2008), 6–18.

Y. Wang and B. Li. 2009. Investigation of memory-based multi-objective optimization evolutionary algorithm in dynamic environment. In *Proceedings of Congress on Evolutionary Computation*. IEEE, 630–637.

Y. Wang and B. Li. 2010. Multi-strategy ensemble evolutionary algorithm for dynamic multi-objective optimization. *Memetic Computing* 2, 1 (2010), 3–24.

S.-Y. Zeng, G. Chen, L. Zheng, H. Shi, H. de Garis, L. Ding, and L. Kang. 2006. A dynamic multi-objective evolutionary algorithm based on an orthogonal design. In *Proceedings of Congress on Evolutionary Computation*. IEEE, 573–580.

Z. Zhang. 2007. Multiobjective optimization immune algorithm in dynamic environments and its application to greenhouse control. *Applied Soft Computing* 8 (2007), 959–971.

B. Zheng. 2007. A new dynamic multi-objective optimization evolutionary algorithm. In *Proceedings of International Conference on Natural Computation*, Vol. 5. IEEE, 565–570.

A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang. 2007. Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization. In *Evolutionary Multi-Criterion Optimization*, Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata (Eds.). Lecture Notes in Computer Science, Vol. 4403. Springer, Berlin, 832–846.

E. Zitzler, K. Deb, and L. Thiele. 2000. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation* 8, 2 (2000), 173–195.