

2006

# A Review of Multi-objective Test Problems and a Scalable Test Problem Toolkit

Simon Huband  
*Edith Cowan University*

Philip Hingston  
*Edith Cowan University*

Luigi Barone

Lyndon While

---

[10.1109/TEVC.2005.861417](https://doi.org/10.1109/TEVC.2005.861417) This article was originally published as: Huband, S., Hingston, P. F., Barone, L., & While, L. (2006). A Review of Multi-objective Test Problems and a Scalable Test Problem Toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5), 477-506. Original article available [here](#)

This Journal Article is posted at Research Online.

<https://ro.ecu.edu.au/ecuworks/2022>

# A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit

Simon Huband, *Member, IEEE*, Phil Hingston, *Member, IEEE*, Luigi Barone, *Member, IEEE*, and Lyndon While, *Senior Member, IEEE*

**Abstract**—When attempting to better understand the strengths and weaknesses of an algorithm, it is important to have a strong understanding of the problem at hand. This is true for the field of multiobjective evolutionary algorithms (EAs) as it is for any other field.

Many of the multiobjective test problems employed in the EA literature have not been rigorously analyzed, which makes it difficult to draw accurate conclusions about the strengths and weaknesses of the algorithms tested on them. In this paper, we systematically review and analyze many problems from the EA literature, each belonging to the important class of real-valued, unconstrained, multiobjective test problems. To support this, we first introduce a set of test problem criteria, which are in turn supported by a set of definitions.

Our analysis of test problems highlights a number of areas requiring attention. Not only are many test problems poorly constructed but also the important class of nonseparable problems, particularly nonseparable multimodal problems, is poorly represented. Motivated by these findings, we present a flexible toolkit for constructing well-designed test problems. We also present empirical results demonstrating how the toolkit can be used to test an optimizer in ways that existing test suites do not.

**Index Terms**—Evolutionary algorithms (EAs), multiobjective evolutionary algorithms, multiobjective optimization, multiobjective test problems.

## I. INTRODUCTION

EVOLUTIONARY algorithms (EAs) have proven to be very useful when solving complex multiobjective optimization problems, including many real-world problems [1]. Examples of real-world multiobjective optimization problems include stationary gas turbine combustion process optimization [2], rock crusher design [3], distributing products through oil pipeline networks [4], Yagi-Uda antenna design [5], nuclear fuel management [6], scheduling [7], the design of telecommunication networks [8], and defense applications [9]. The population-based nature of EAs lends itself well to multiobjective optimization, where the aim is to discover a range of solutions offering a variety of tradeoffs between the objectives. Indeed, past research has been so successful that new multiobjective EAs (MOEAs) are constantly being devised [10]–[13].

The development of such a large variety of EAs has likewise resulted in numerous comparisons, with the goal of demon-

strating the general superiority of one algorithm over its peers, the “no free lunch” theorem [14], [15] notwithstanding. This process of comparison is supported by two resources: a large set of easily implemented, artificially constructed, multiobjective test problems, and a wide range of measures with which results can be compared. Consider the typical scenario of EA comparison.

- 1) Select the EAs to compare.
- 2) Choose a set of existing (preferably benchmark) test problems or create new ones.
- 3) Choose a set of measures on which to compare the sets of results produced by EAs.
- 4) Obtain results for each EA on each test problem, whether from the Web or by implementation.
- 5) Generate measures for the results and compare the data.
- 6) Draw conclusions.

An alternative approach is to compare the attainment function derived from each set of results [16]. Although this method does not make use of a numerical measure as such, the key role of suitable test functions remains.

In order to draw accurate conclusions, it is imperative that the test problems employed be well understood, that the measures be appropriate, and that proper statistical methods be employed. While more work is needed, the latter two of these have received much attention in the literature. The quality of numerous measures has recently been rigorously analyzed by several authors [17]–[19] and there are useful statistical techniques available (for example, randomization testing [12], [20] or in the case of attainment functions [16], [21]). However, just how well understood are the test problems in the literature?

Artificially constructed test problems offer many advantages over real-world problems for the purpose of general performance testing. Test problems can be designed to be easy to describe, easy to understand and visualize, easy to implement, fast, and their optima are often known in advance. But what makes a test problem well designed for general use? What makes a test suite good? How do we define what a test problem does and does not test in an algorithm? Without having a clear understanding of the answers to these questions, how can we assert the veracity of our comparisons?

This paper contributes to the literature in four ways. First, we provide a set of definitions that facilitate the analysis of test problem characteristics. Second, we define a comprehensive set of test problem recommendations and features by which test problems can be categorized. Third, and importantly, we analyze a large range of test problems from the literature using these recommendations and features. Significantly, it is apparent from

Manuscript received June 14, 2005; revised September 11, 2005. This work was supported in part by the Australian Research Council.

S. Huband and P. Hingston are with the Edith Cowan University, Mount Lawley WA 6050, Australia (e-mail: p.hingston@ecu.edu.au).

L. Barone and L. While are with The University of Western Australia, Crawley WA 6009, Australia (e-mail: lyndon@csse.uwa.edu.au).

Digital Object Identifier 10.1109/TEVC.2005.861417

our broad analysis of the literature that several important features are rarely tested. Finally, in the light of this finding, we describe a toolkit for the creation of test problems, and use it to develop a suite of problems that meet our recommendations and exhibits these features.

This paper is organized as follows. Section II begins by identifying the paper focus. This is followed in Section III by a range of definitions that pertain to multiobjective optimization and test problem characteristics. Section IV goes on to provide a set of test problem recommendations, and a set of possible test problem features. The composition of test suites, as opposed to the makeup of individual test problems, is considered in Section V. The bulk of this paper follows in Section VI, wherein several popular test suites are reviewed in detail, and the properties of a number of other test problems from the literature are summarized. The limitations of the reviewed literature are then highlighted in Section VII. In Section VIII, we introduce our toolkit and describe a test suite built using the toolkit. We then present some experiments illustrating the effectiveness of this test suite in Section IX. We conclude in Section X.

## II. SCOPE

The variety and type of test problems addressed in the literature is enormous, and for practical reasons we focus our attention on one specific class—multiobjective problems that do not have side constraints, whose parameters are real valued, and whose objectives are well-defined mathematical functions. Such problems are the easiest to analyze and construct and are also a popular choice.

Some of the other important areas of research that are not covered include real-world problems, combinatorial optimization problems, discrete or integer-based problems, noisy problems, dynamic problems, and problems with side constraints. The area we do consider remains one of the most studied in the test problem literature.

## III. DEFINITIONS

In order to discuss and categorize test problems unambiguously, it is important to properly define all of the related concepts. Although the concept of Pareto optimality is well understood, this paper deals with the fine details of test problems, fitness landscapes, and Pareto optimal front geometries. For this, we draw on and extend the work of earlier authors, particularly that of Deb [22].

### A. The Basics

In multiobjective optimization, we aim to find the set of optimal tradeoff solutions known as the Pareto optimal set. For problems of large cardinality, including continuous problems, a representative subset will usually suffice.

Without loss of generality, consider a multiobjective optimization problem defined in terms of a search space of allowed values of  $n$  parameters  $x_1, \dots, x_n$ , and a vector of  $m$  objective functions  $\{f_1(x_1, \dots, x_n), \dots, f_m(x_1, \dots, x_n)\}$  mapping parameter vectors into fitness space. The mapping from the search space to fitness space defines the *fitness landscape*. A problem is *scalable parameter-wise* iff for any  $n_0 > 0$  variants can be

created for some  $n > n_0$ . Likewise, a problem can also be *scalable objective-wise*.

Pareto optimality is defined using the concept of domination. Given two parameter vectors  $\mathbf{a}$  and  $\mathbf{b}$ ,  $\mathbf{a}$  *dominates*  $\mathbf{b}$  iff  $\mathbf{a}$  is at least as good as  $\mathbf{b}$  in all objectives, and better in at least one. Similarly,  $\mathbf{a}$  is *equivalent to*  $\mathbf{b}$  iff  $\mathbf{a}$  and  $\mathbf{b}$  are identical to one another in all objectives. If  $\mathbf{a}$  either dominates or is equivalent to  $\mathbf{b}$ , then  $\mathbf{a}$  *covers*  $\mathbf{b}$ . Two parameter vectors are *incomparable* iff they are not equivalent, and neither dominates the other. A parameter vector  $\mathbf{a}$  is *nondominated* with respect to a set of vectors  $X$  iff there is no vector in  $X$  that dominates  $\mathbf{a}$ .  $X$  is a *nondominated set* iff all vectors in  $X$  are mutually nondominating. The set of corresponding objective vectors for a nondominated set is a *nondominated front*.

Given two sets of objective vectors  $Y$  and  $Z$ ,  $Y$  *dominates*  $Z$  iff every element of  $Z$  is dominated by some element of  $Y$ . Similar definitions relating  $Y$  and  $Z$  in terms of equivalence, coverage, and mutual nondominance can also be made.

A parameter vector  $\mathbf{a}$  is *Pareto optimal* iff  $\mathbf{a}$  is nondominated with respect to the set of all allowed parameter vectors. Pareto optimal vectors are characterized by the fact that improvement in any one objective means worsening at least one other objective. The *Pareto optimal set* is the set of all Pareto optimal parameter vectors, and the corresponding set of objective vectors is the *Pareto optimal front*. The Pareto optimal set is a subset of the search space, whereas the Pareto optimal front is a subset of the fitness space.

### B. Fitness Landscape

We are interested in both the nature of the fitness landscape, and the more specific relationship between the Pareto optimal set and the Pareto optimal front. The former identifies the types of difficulties encountered in the search space, whereas the latter influences our judgement of what is considered a “good” representative subset of the Pareto optimal set, which is important when it is impractical to identify the entire Pareto optimal set.

The fitness landscape can be one-to-one or many-to-one. The many-to-one case presents more difficulties to the optimizer, as choices must be made between two parameter vectors that evaluate to identical objective vectors. Likewise, the mapping between the Pareto optimal set and the Pareto optimal front may be one-to-one or many-to-one. In each case, we say that the problem is *Pareto one-to-one* or *Pareto many-to-one*, respectively.

A special instance of a many-to-one mapping occurs when a connected open subset of parameter space maps to a singleton. We refer to problems with this characteristic as problems with *flat regions*, that is, regions where small perturbations of the parameters do not change the objective values. Optimizers can have difficulty with flat regions due to a lack of gradient information. Should the majority of the fitness landscape be fairly flat, providing no useful information regarding the location of Pareto optima, then the Pareto optima are said to be *isolated optima* [22]. Problems with isolated optima are very difficult to solve.

Another characteristic of fitness landscapes is *modality*. An objective function is *multimodal* when it has multiple local optima. An objective function with only a single optimum is *unimodal*. A *multimodal problem* is one that has a multimodal objective.

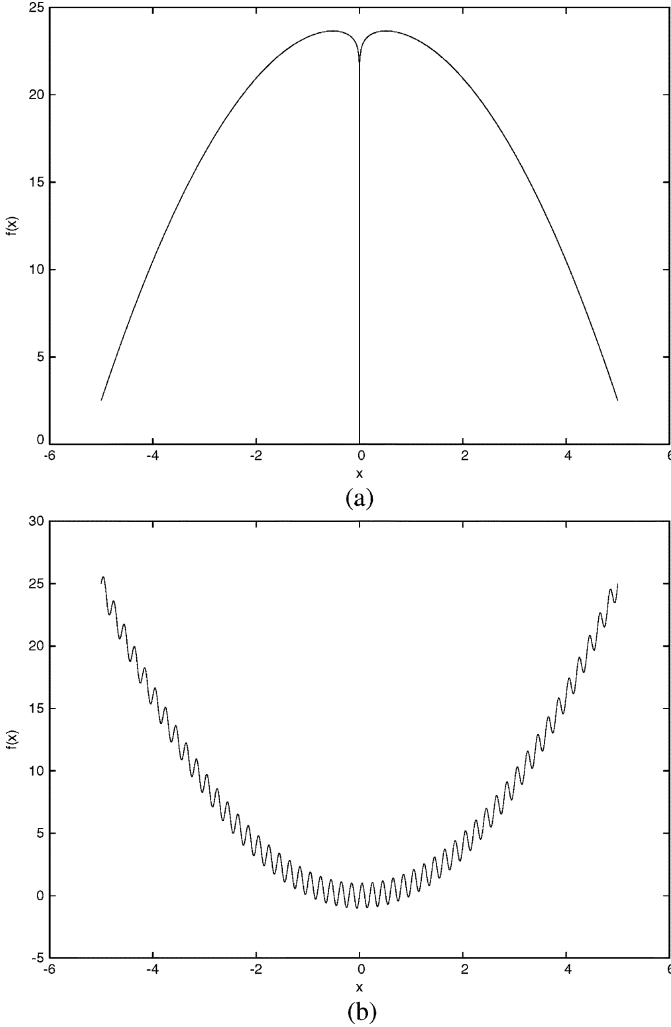


Fig. 1. (a) Example of a deceptive multimodal objective and (b) a nondeceptive multimodal objective. The deceptive multimodal objective has been exaggerated for clarity. (a) Deceptive: minimize  $f(x) = -0.9x^2 + (5|x|^{0.001}/5^{0.001})^2$ , where  $-5 \leq x \leq 5$ . (b) Multimodal: minimize  $f(x) = x^2 + \sin(10\pi x)$ , where  $-5 \leq x \leq 5$ .

A *deceptive* objective function has a special kind of multimodality. As defined by Deb [22], for an objective function to be deceptive it must have at least two optima—a true optimum and a deceptive optimum—but the majority of the search space must favor the deceptive optimum. A *deceptive problem* is one with a deceptive objective function. Multimodal problems are difficult because an optimizer can become stuck in local optima. Deceptive problems exacerbate this difficulty by placing the global optimum in an unlikely place. Two examples of multimodality, highlighting the difference between deceptiveness and nondeceptiveness, are plotted in Fig. 1.

Another characteristic of the fitness landscape is whether an evenly distributed sample of parameter vectors in the search space maps to an evenly distributed set of objective vectors in fitness space. We expect some variation in distribution, but are especially interested in significant variation, which is known as *bias*. Bias has a natural impact on the search process, particularly when the mapping from the Pareto optimal set to the Pareto optimal front is biased. If it is only feasible to identify a representative subset of the Pareto optimal set, and the mapping from

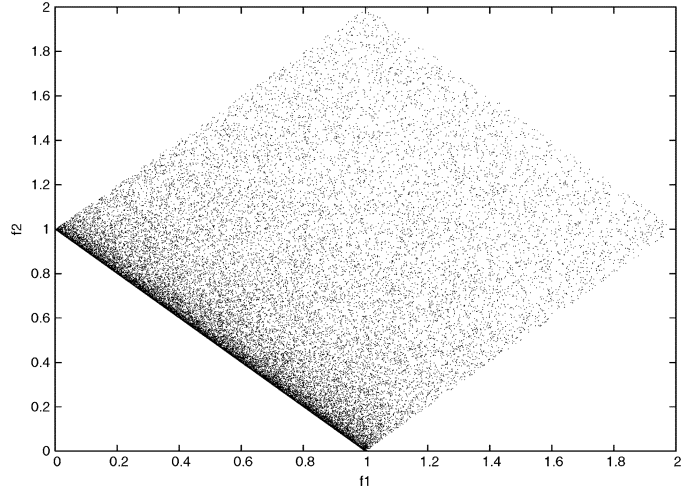


Fig. 2. The objective vectors that correspond to 40 000 randomly selected parameter vectors from the biased two objective problem  $f_1(x_1, x_2) = (x_1)^5 + x_2$ ,  $f_2(x_1, x_2) = (x_1)^5 + 1 - x_2$ , where  $x_1, x_2 \in [0, 1]$ . Note how, in this example, the objective vectors are denser toward the Pareto optimal front.

the Pareto optimal set to the Pareto optimal front is biased, then which is more important: to achieve an even distribution of solutions with respect to the search space or with respect to the fitness space? The answer to this question depends on the decision maker. Fig. 2 shows the effects of bias.

The judgment of whether a problem is biased is based on the density variation of solutions in fitness space, given an even spread of solutions in parameter space. While it is usually easy enough to agree whether a problem has bias, at the present time there is no agreed mathematical definition of bias (but see [16] for one possibility). Bias is perhaps best indicated by plotting solutions in fitness space.

For the purpose of this paper, we judge the bias of problems with respect to their most scaled-down instance (that is, with the minimum number of parameters); by changing the fitness landscape, scaling the number of parameters directly influences bias. We only qualify a problem as being biased when the density variation is significant, such as when bias is deliberately incorporated into a problem.

Parameter dependencies are an important aspect of a problem. Given a single objective  $O$ , a parameter vector  $\mathbf{x}$ , and an index  $i$ , we define a derived problem  $P_{O,\mathbf{x},i}$  as the problem of optimizing  $O$  by varying only  $x_i$ . This is a single objective problem with a single parameter. We also define  $P_{O,\mathbf{x},i}^*$  to be the set of global optima (in parameter space) for each subproblem. If  $P_{O,\mathbf{x},i}^*$  is the same for all values of  $\mathbf{x}$ , we say that  $x_i$  is *separable on  $O$* . Otherwise,  $x_i$  is *nonseparable on  $O$* .

Should every parameter of  $O$  be separable, then  $O$  is a *separable objective*. Otherwise,  $O$  is a *nonseparable objective*. Similarly, should every objective of a problem  $P$  be separable, then  $P$  is a *separable problem*. Otherwise,  $P$  is a *nonseparable problem*.

Separable objectives can be optimized by considering each parameter in turn, independently of one another, and the resultant set of globally optimal parameter vectors is the cross-product of the optimal sets for each individually optimized parameter. In the multiobjective sense, this means that the ideal points for separable objectives can be determined by

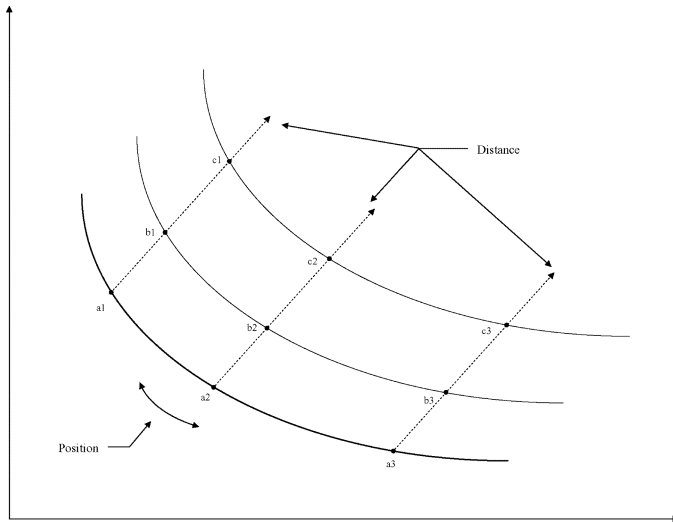


Fig. 3. The difference between distance and position. Objective vectors labeled with the same letter occur at a different position on the same non-dominated front, whereas objective vectors labeled with the same number occur at different distances to the Pareto optimal front (at the same position on different fronts).

considering only one parameter at a time. Finding at least some points on the Pareto optimal front for a separable problem tends to be simpler than for an otherwise equivalent nonseparable problem.

Individual parameters can also be categorized in terms of their relationship with the fitness landscape. The following types of relationships are useful because they allow us to separate the convergence and spread aspects of sets of solutions for a problem. The first type of parameter is called a *distance parameter*. A parameter  $x_i$  is a distance parameter iff for all parameter vectors  $\mathbf{a}$ , modifying  $x_i$  in  $\mathbf{a}$  results in a parameter vector that dominates  $\mathbf{a}$ , is equivalent to  $\mathbf{a}$ , or is dominated by  $\mathbf{a}$ . Modifying a distance parameter on its own never results in incomparable parameter vectors.

If, instead, modifying  $x_i$  in  $\mathbf{a}$  can only result in a vector that is incomparable or equivalent to  $\mathbf{a}$ , then  $x_i$  is a *position parameter*. Modifying a position parameter on its own never results in a dominating or dominated parameter vector. The difference between distance and position is highlighted in Fig. 3.

All parameters that are neither position nor distance parameters are *mixed parameters*. Modifying mixed parameters on their own can result in a change in position or distance.

The projection of the Pareto optimal set onto the domain of a single parameter can be a small subset of the domain. If that subset is a single value at the edge of the domain, then we call the parameter an *extremal parameter*. If instead the projection should cluster around the middle of the domain, then it is a *medial parameter*.

### C. Pareto Optimal Geometries

Unlike single objective problems, for which the Pareto optimal front is but a single point, Pareto optimal fronts for multi-objective problems can have a wide variety of geometries.

Recall that a set is *convex* iff it covers its convex hull. Conversely, it is *concave* iff it is covered by its convex hull. A set is *strictly convex* (respectively, *strictly concave*) if it is convex

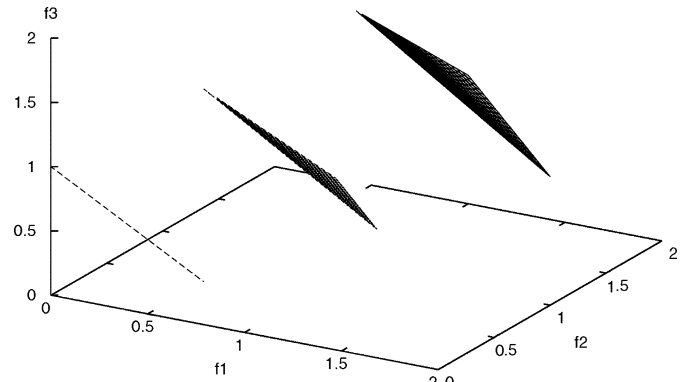


Fig. 4. Three different non-dominated fronts taken from a degenerate problem. The front closest to the origin is the Pareto optimal front and is a degenerate line.

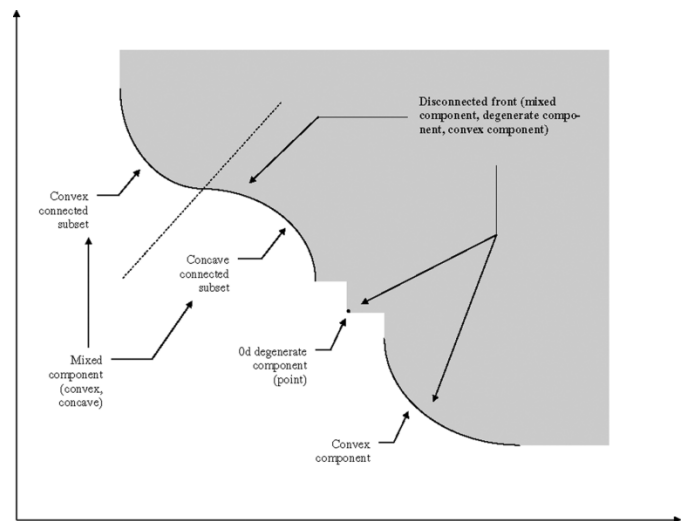


Fig. 5. Sample geometry of a disconnected, mixed front that consists of: a half-convex half-concave component, a degenerate zero dimensional point, and a convex component.

(respectively, concave) and not concave (respectively, convex). A *linear* set is one that is both concave and convex.

A *mixed* front is one with connected subsets that are each strictly convex, strictly concave, or linear, but not all of the same type.

A *degenerate* front is a front that is of lower dimension than the objective space in which it is embedded, less one. For example, a front that is a line segment in a three objective problem is degenerate. Conversely, a two-dimensional front in a three objective problem is not degenerate. Fig. 4 plots a degenerate problem.

Degenerate Pareto optimal fronts can cause problems for some algorithms. For example, methods employed to encourage an even spread of solutions across the Pareto optimal front might operate differently should the front effectively employ fewer dimensions than expected.

We are also interested in whether a front is a *connected* set. In the literature, a front that is a disconnected set is often referred to as *discontinuous*, which we feel is more appropriate when used to describe a function that identifies a front. Fig. 5 serves to clarify some of these geometries.

TABLE I  
LISTING OF DESIRABLE MULTIOBJECTIVE TEST PROBLEM RECOMMENDATIONS AND POSSIBLE TEST PROBLEM FEATURES

Recommendation (R) or Feature (F)	Comment
R1: No extremal parameters	Prevents exploitation by truncation based correction operators
R2: No medial parameters	Prevents exploitation by intermediate recombination
R3: Scalable number of parameters	Increases flexibility, demands scalability
R4: Scalable number of objectives	Increases flexibility, demands scalability
R5: Dissimilar parameter domains	Encourages EAs to scale mutation strengths appropriately
R6: Dissimilar tradeoff ranges	Encourages normalisation of objective values
R7: Pareto optima known	Facilitates the use of measures, analysis of results, in addition to other benefits
F1: Pareto optimal geometry	Convex, linear, concave, mixed, degenerate, disconnected, or some combination
F2: Parameter dependencies	Objectives can be separable or non-separable
F3: Bias	Substantially more solutions exist in some regions of fitness space than they do in others
F4: Many-to-one mappings	Pareto one-to-one/many-to-one, flat regions, isolated optima
F5: Modality	Uni-modal, or multi-modal (possibly deceptive multi-modality)

The geometry of the Pareto optimal set can also be described using various terms. We do not go into detail regarding search space geometries, except to say that Pareto optimal sets can also be *disconnected*. Although disconnected Pareto optimal sets usually map to disconnected Pareto optimal fronts, this is not always the case.

#### IV. MULTIOBJECTIVE TEST PROBLEM CHARACTERISTICS

Multiojective problems form an inherently rich domain, requiring a correspondingly rich set of criteria by which to judge them. Such matters have been discussed by other authors, in particular, in the pioneering work of Deb *et al.* [22], [23]. In this section, we draw on this work and enhance it, presenting a detailed and thorough formalization of multiojective problems.

Note that this section focuses on the properties of individual test problems. Section V makes recommendations regarding the construction of test suites.

Our formalization is divided into two levels: recommendations and features. Recommendations are so named as they are always beneficial, whereas features are properties that collectively identify the difficulties a test problem presents to an optimizer. Recommendations are either adhered to or not, whereas features are merely present in one form or another, or absent. Each of the following sections discusses and justifies each recommendation and feature in turn. A brief summary is presented in Table I.

The justification for several recommendations is best demonstrated by example. To facilitate this, the following simple EA is customized as needed.

- 1) Create an initial current population of  $n$  individuals by randomly initializing parameters uniformly on their domains.
- 2) Clone the current population to create a child population of  $n$  individuals.
- 3) For each individual in the child population do the following.
  - a) Mutate each parameter  $x_i$  of the current child with probability  $1/|\mathbf{x}|$  (where  $\mathbf{x}$  is the parameter vector and  $|\mathbf{x}|$  is the number of parameters) according to the

schedule  $x_i \leftarrow x_i + \sigma N(0, 1)$ , where  $\sigma$  is some step size and  $N(0, 1)$  is a normally distributed random value with expectation zero and standard deviation one.  $N(0, 1)$  is sampled anew for all parameters considered, and by default we let  $\sigma = 0.1$ .

- b) Correct any invalid parameter values by truncating them back to their closest valid value.
- 4) Perform crossover. Unique pairs of individuals in the child population randomly swap their parameters.
- 5) Add the current population to the child population to create a combined population of individuals.
- 6) The next generation consists of the  $n$  best individuals from the combined population, according to Goldberg's nondominated ranking procedure [24], resolving ties randomly.
- 7) If  $G$  generations have not yet elapsed, return to step 2). Otherwise, output the nondominated solutions of the current population as the solution.

##### A. Recommendation 1: No Extremal Parameters

No parameter of the test problem should be an extremal parameter.

1) *Justification:* Placing the optimum of a distance parameter at the edge of its domain is bad practice. Extremal parameters are easily optimized “by accident” when EAs employ mutation strategies that truncate invalid parameter values back to the edge of their domain. Conversely, EAs that correct invalid mutations by reflecting about the edge of the domain are less likely to succeed when dealing with extremal parameters.

This can be demonstrated with the following problem:

$$\begin{aligned} \text{Minimize } f_1(x) &= (x - A)^2 \\ \text{Minimize } f_2(x) &= 1 - \cos(12\pi(x - A)) \end{aligned}$$

where  $x, A \in [0, 1]$ . This problem has a degenerate Pareto optimal front, namely, the objective vector that occurs when  $x = A$ , which results in both fitness functions evaluating to zero.

We consider two instances of the above problem: when  $A = 0$  (which makes  $x$  an extremal parameter) and when  $A = 0.01$ .

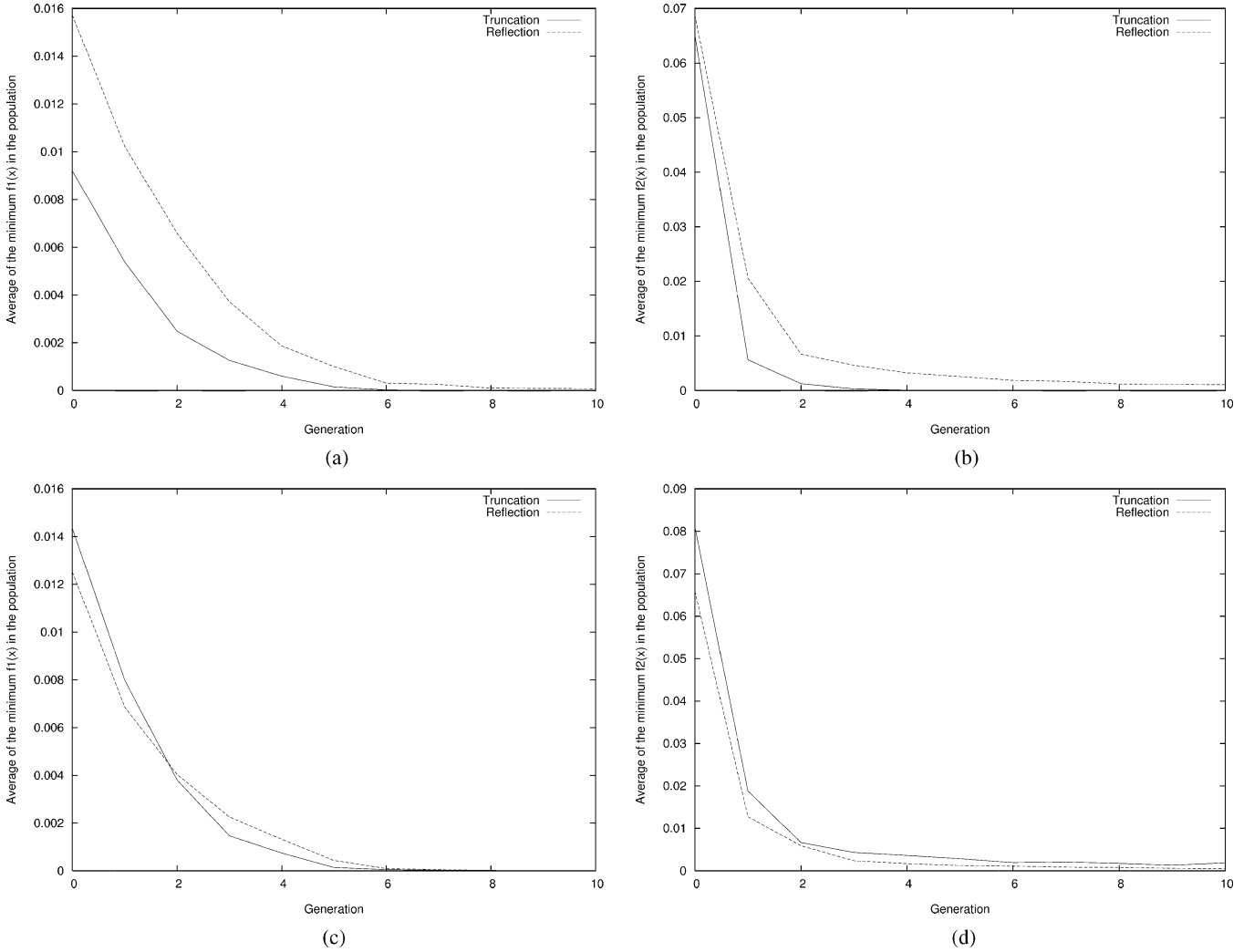


Fig. 6. Example of the relationship between extremal parameters and the use of truncation to correct invalid mutations. Each of the graphs plots the minimum of either the first or second fitness value from a population of ten individuals over ten generations of evolution, as averaged across 100 runs. (a) Performance relative to  $f_1(x)$ , with  $A = 0$ . (b) Performance relative to  $f_2(x)$ , with  $A = 0$ . (c) Performance relative to  $f_1(x)$ , with  $A = 0.01$ . (d) Performance relative to  $f_2(x)$ , with  $A = 0.01$ .

The results of attempting to optimize these two instances are shown in Fig. 6. Two versions of our simple EA are employed: the first as described above (with truncation) and the second modified to correct invalid parameter values by reflecting them about their closest invalid value.

As Fig. 6 shows, the performance of the EA with truncation is better when  $x$  is an extremal parameter ( $A = 0$ ). Indeed, it was only in this case that the Pareto optimal solution was identified, on average after only ten generations. When the optimum is moved marginally from  $x = 0$  to  $x = 0.01$  by setting  $A$  to 0.01, the performance of the EA with truncation is reduced to that of the EA with reflection. The performance of the EA with reflection is largely unchanged.

### B. Recommendation 2: No Medial Parameters

No parameter of the test problem should be a medial parameter.

1) *Justification:* Fogel and Beyer [25] show the following.

If initial trial solutions are uniformly distributed symmetrically about the optimum, the use of intermediate recombination followed by independent zero-mean Gaussian perturbation

generates offspring that are unbiased estimates of the optimum solution.

They also show that a similar theorem can be proved for the use of zero-mean Gaussian perturbations alone.

This means that EAs that employ an initial population of uniformly randomly initialized parameter vectors (uniformly with respect to the domain of each parameter), and that employ intermediate recombination, can be biased toward finding optimal solutions should the problem include medial parameters. Ensuring there are no medial parameters is not difficult.<sup>1</sup>

### C. Recommendation 3: Scalable Number of Parameters

The test problem should be scalable to have any number of parameters.

1) *Justification:* It is beneficial if the number of parameters can be changed in order to test different levels of difficulty. This

<sup>1</sup>Fogel and Beyer actually recommend that EAs should be tested on benchmark functions in various configurations that include initializing the population with large perturbations directed away from the optimum. As this paper focuses on the construction of test problems for general use, rather than the design and configuration of EAs, the recommendation regarding medial parameters instead aims to avoid such a favorable set of circumstances in the first place.

recommendation is identical to the second desired test feature proposed by Deb *et al.* [23].

#### D. Recommendation 4: Scalable Number of Objectives

The test problem should be scalable to have any number of objectives.

1) *Justification*: Much like Recommendation 3, the benefits of this recommendation are clear. The number of objectives influences a test problem's difficulty. For convenience in building test suites, it is desirable to allow the number of objectives to vary, whilst maintaining the same underlying properties of the test problem. This recommendation is identical to Deb *et al.*'s third desired feature of test problems.

#### E. Recommendation 5: Dissimilar Parameter Domains

The parameters of the test problem should have domains of dissimilar magnitude.

1) *Justification*: A test problem that employs parameters with domains of dissimilar magnitude enforces the need for mutation strengths that vary accordingly. Really, algorithms should always normalize parameter domains, as doing so is trivial and avoids this problem. However, in practice this is often not done, so we include this recommendation.

To demonstrate this, consider the following problem:

$$\begin{aligned} \text{Minimize } f_1(x) &= (x_1 - 5)^2 + (10x_2/A - 6)^2 \\ \text{Minimize } f_2(x) &= (x_1 - 7)^2 + (10x_2/A - 6)^2 \end{aligned}$$

where  $x_1 \in [-10, 10]$ ,  $x_2 \in [-A, A]$ , and  $A > 0$ . This problem has a convex Pareto optimal front. A parameter vector is Pareto optimal whenever  $x_1 \in [5, 7]$  and  $x_2 = 6A/10$ .

Two instances of this problem are considered: when  $A = 10$  (identical parameter domains) and when  $A = 100$  (dissimilar parameter domains). To optimize these two instances, we employ two variations of our simple EA. The first mutates parameters according to  $\sigma = 0.1 \times (x_{1,\max} - x_{1,\min} + x_{2,\max} - x_{2,\min}/2)$ , which is based on the average of the parameter domain magnitudes. The second mutates parameters according to  $\sigma = 0.1 \times (x_{i,\max} - x_{i,\min})$ , which effectively scales the mutation strength proportionally to the magnitude of each parameter's domain. This is equivalent to normalizing parameter domains.

Using attainment surfaces, Fig. 7 demonstrates the need for scaling mutation strengths relative to each parameter. An *attainment surface* is the boundary in objective space formed by the obtained front, which separates the region dominated by the obtained solutions from the region that is not dominated [26]. Multiple attainment surfaces can be superimposed and interpreted probabilistically. For example, the 50% attainment surface identifies the region of objective space that is dominated by half of the given attainment surfaces, whereas the 100% attainment surface identifies the region dominated by every given attainment surface.

#### F. Recommendation 6: Dissimilar Tradeoff Ranges

The magnitude of the tradeoff ranges in each objective of the test problem's Pareto optimal front should be dissimilar.

1) *Justification*: As it is usually infeasible to identify the entire Pareto optimal front, many EAs settle for attempting to

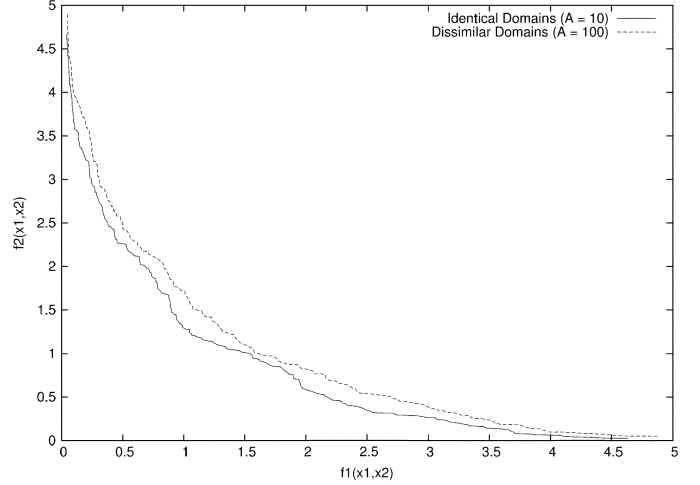


Fig. 7. The 50% attainment surfaces achieved by the EA when mutation strengths are not scaled individually for each parameter, as determined from 100 runs using a population size of ten, where each run lasted 50 generations. As can be seen, in this example the 50% attainment surface of the EA has degraded on the problem with dissimilar parameter domains ( $A = 100$ ). The attainment surfaces for the EA that scales mutation strengths accordingly are not shown, as they are identical to the better attainment surface shown above.

find a representative subset. This is commonly achieved using Euclidean distance based measures that attempt to maintain an even spread of solutions with respect to the fitness landscape. The purpose of this recommendation is to encourage algorithms to normalize objective values accordingly prior to employing any scaling dependent measures. In this way, equal emphasis can be placed on each objective, at least so far as the algorithm can determine. It is important to note that in the real world the true Pareto optimal tradeoff magnitudes are not always known *a priori*. Consequently, it is reasonable to expect that EAs should dynamically employ mechanisms to cope with dissimilar tradeoff ranges.

For example, real-world multiobjective problems might define a tradeoff between, say, quality and price, which could be defined on vastly different scales. It stands to reason that algorithms should be attempting to find an even spread of solutions irrespective of the scales employed. All objectives should be treated equally.

Recommending test problems employ dissimilar tradeoff ranges is more consistent with real-world expectations and should encourage EAs to normalize tradeoff ranges.

#### G. Recommendation 7: Pareto Optima Known

The test problem's Pareto optimal set and Pareto optimal front should be expressed in closed form, or at least in some other usable form.

For example, it might be possible to define an  $(M-1)$ -dimensional function (where  $M$  is the number of objectives) that covers and is covered by the Pareto optimal front—the function might map to some dominated solutions, but they can be readily identified.

1) *Justification*: Many performance measures require knowledge of the Pareto optimal front. More generally, knowing the location of the Pareto optimal front is necessary if we are to accurately and independently assess the overall performance of any given algorithm. In any case, it is simply good



practice for test problems to be well defined and understood. This recommendation is similar to Deb *et al.*'s fifth desired feature of test problems.

We also note that there are theorems relating to  $M$  objective problems with  $M$  parameters (in particular, two objective problems with two parameters) that can help identify Pareto optima [27].

#### H. Feature 1: Pareto Optimal Geometry

The geometry of the Pareto optimal front can be convex, linear, concave, mixed, degenerate, disconnected, or some combination of the former. We are also interested in whether the Pareto optimal set is connected.

1) *Importance:* The geometry of the Pareto optimal front can directly influence the performance of EAs. We have already provided a formalization of a wide variety of geometries (see Section III-C). Certain among them are of special interest.

For example, disconnected Pareto optimal fronts and their analogue, disconnected Pareto optimal sets, increase the likelihood that an EA will fail to find all regions of the Pareto optimal front and/or set.

A less obvious complication occurs with convex Pareto optimal fronts. Convex Pareto optimal fronts can cause difficulty for EAs that rank solutions for selection based on the number of other solutions that they dominate [22]. Assuming a fairly even spread of solutions in the fitness landscape, solutions around the middle of the convex Pareto optimal front will tend to dominate more solutions, giving them a better rank. This problem can also occur with other Pareto optimal geometries, provided the density distribution of solutions in the fitness landscape varies accordingly.

#### I. Feature 2: Parameter Dependencies

The objectives of a test problem can be separable or nonseparable.

1) *Importance:* Problems with parameter dependencies are of great importance, but have received little attention in the multiobjective EA literature. The importance of nonseparable problems is already recognized with respect to single objective problems [28], [29]. Multiobjective problems have the additional complication of potentially having multiple nonseparable objectives.

#### J. Feature 3: Bias

A test problem may or may not be biased.

1) *Importance:* It is useful to be able to identify the presence of bias in a test problem, as bias directly influences the convergence speed toward the Pareto optimal front of EAs. In combination with other features, bias can also be used as a means of increasing problem difficulty. For example, the fitness landscape could be biased such that the global optimum of a multimodal objective is more difficult to find. Deb [22] provides a more detailed discussion of the effects of bias.

#### K. Feature 4: Many-to-One Mappings

The fitness landscape may be one-to-one or many-to-one. Pareto many-to-one problems and problems with flat regions are of particular interest.

1) *Importance:* Pareto many-to-one problems affect an EA's ability to find multiple, otherwise equivalent optima. Finding these would allow the decision maker greater leeway when choosing between solutions.

Flat regions in the fitness landscape, including fitness landscapes with isolated optima, are of interest due to their effect on problem difficulty.

#### L. Feature 5: Modality

The objectives of a test problem are either unimodal or multimodal. The latter may also be the special case of deceptive multimodality.

1) *Importance:* It is well understood that multimodal problems are both more difficult than unimodal problems, and more representative of real-world problems. Bäck and Michalewicz [29] also identify a number of important design criteria for single objective multimodal test problems. First, the number of local optima should scale exponentially with respect to the number of associated parameters, and local optima should not be arranged in an extremely regular fashion. In addition, the problems should not (counterintuitively) become easier to solve as the number of parameters increases, which has been observed to happen when a global shape is imposed over a finer structure of local optima in such a fashion that increasing the number of parameters reduces the number and complexity of the local optima.

### V. TEST SUITE RECOMMENDATIONS

In principle, test suites should consist of a variety of test problems that collectively capture a wide variety of characteristics. However, this is not so easy to achieve with the multiobjective problem domain. For example, consider each of the test problem features listed in Section IV. A wide variety of Pareto optimal geometries exist, each of which could be associated with a variety of parameter dependencies, the presence or absence of bias, many-to-one mappings, and different types of modality. To expect a test suite to capture all possible combinations of features is impractical.

However, it is certainly possible to suggest a few baseline rules for multiobjective test suites. In order to do so, we first consider suggestions that have already been made in the context of single objective problems. Specifically, Whitley *et al.* [28] identify that single objective test suites should generally consist of problems that are resistant to hill-climbing strategies, should include scalable problems, and should also include non-linear, nonseparable problems. These requirements are further extended in [30] by Bäck and Michalewicz [29]. In addition to discussing noisy test problems and test problems with side constraints, Bäck and Michalewicz state that single objective test suites should preferably consist of scalable test problems, should include a few unimodal test problems to test convergence velocity, and should include several nonseparable multimodal test problems whose number of local optima grows exponentially with respect to the number of parameters. Bäck and Michalewicz also detail further design considerations for multimodal problems.

Given that multiobjective problems form a superset of single objective problems, it stands to reason that multiobjective test

suite guidelines should likewise be a superset of single objective guidelines. In addition to the recommendations presented in Section IV, which apply to all test problems, we propose the following guidelines for constructing multiobjective test suites.

- 1) There should be a few unimodal test problems to test convergence velocity relative to different Pareto optimal geometries and bias conditions.
- 2) The following three core types of Pareto optimal geometries should be covered by the test suite: degenerate Pareto optimal fronts, disconnected Pareto optimal fronts, and disconnected Pareto optimal sets.
- 3) The majority of test problems should be multimodal, and there should be a few deceptive problems.
- 4) The majority of problems should be nonseparable.
- 5) There should be problems that are both nonseparable and multimodal. Bäck and Michalewicz state that unimodal and separable problems are not representative of real-world problems.

If diversity in parameter space is also important to the decision maker, then the test suite should include Pareto many-to-one problems.

It is important to understand that no given test problem should be considered poorly designed if it is not sufficiently “complex,” or if it does not test some “important” feature. However, it is reasonable to criticize a test suite were none of its test problems difficult to optimize, or if it only tested a limited range or combination of features. The distinction is that while a test problem might embody only one of a variety of possible features, a test suite should contain test problems that collectively test a broad range of possible features.

## VI. LITERATURE REVIEW

Testing algorithms is important, so there have been several attempts to define test suites or toolkits for building test suites. However, existing multiobjective test problems do not test a wide range of characteristics, and often have design flaws. Typical defects include not being scalable or being susceptible to simple search strategies.

In this section, we employ the categorizations introduced in Section IV to provide a detailed review of numerous test problems from the literature. In doing so we present tables such as Table V and Table VII, each of which is formatted similarly. The symbols “✓” and “+” indicate whether a given recommendation is adhered to, whereas the symbols “+” and “−” indicate the presence of absence of some feature. In some cases a more descriptive entry is made in the form of numbers, text, or an abbreviation. The latter includes “NA” for not applicable, “S” for separable, “NS” for nonseparable, “U” for unimodal, “M” for multimodal, and “D” for deceptive. An asterisk (\*) indicates an entry is further commented on in the text.

The left side of each table pertains to recommendations and features that apply to each objective. Conversely, the right side of each table deals with recommendations and features that apply to the problem as a whole, not each objective. Objectives are analyzed individually with respect to whether they employ a scalable number of parameters, their separability, and their modality. For brevity, we have omitted Recommendation 4

$$\begin{array}{ll}
 \text{Given} & \mathbf{x} = \{x_1, \dots, x_n\} \\
 \text{Minimise} & f_1(\mathbf{y}) \\
 & f_2(\mathbf{y}, \mathbf{z}) = g(\mathbf{z})h(f_1(\mathbf{y}), g(\mathbf{z})) \\
 \text{where} & \mathbf{y} = \{x_1, \dots, x_j\} \\
 & \mathbf{z} = \{x_{j+1}, \dots, x_n\}
 \end{array}$$

Fig. 8. Deb’s toolkit’s high-level decomposition of two-objective problems. The three unspecified functionals  $f_1$ ,  $g$ , and  $h$  can be selected from a list of example functions provided by Deb. As described in the text,  $\mathbf{y}$  and  $\mathbf{z}$  are position and distance parameter vectors, respectively.

from each table, as the number of objectives employed by each problem is obvious from the text or from the table itself.

Our review is split into three sections. The first two sections provide a detailed review of several test suites. Specifically, Section VI-A analyzes three related works (by Deb [22], Zitzler *et al.* [31], and more recently Deb *et al.* [23]), and Section VI-B analyzes a suite of test problems employed by Van Veldhuizen [32]. Section VI-C, briefly categorizes a variety of other test problems.

### A. Three Related Test Suites

In this section, we review three related prominent works on multiobjective test problems: Deb’s two objective test problem toolkit [22], Zitzler *et al.*’s ZDT test suite [31], and Deb *et al.*’s DTLZ test suite [23].

Aside from being related by authorship, all three test suites are neatly constructed and share common characteristics. In particular, with a few exceptions, none of the test suites feature problems with mixed parameters—all employ problems whose parameters are either position or distance parameters.

To denote this, we divide the set  $\mathbf{x}$  of  $n$  parameters into two distinct sets as follows:

$$\begin{array}{ll}
 \text{Given} & \mathbf{x} = \{x_1, \dots, x_n\} = \{x_1, \dots, x_j, x_{j+1}, \dots, x_n\} \\
 \text{let} & \mathbf{y} = \{y_1, \dots, y_j\} = \{x_1, \dots, x_j\} \\
 & \mathbf{z} = \{z_1, \dots, z_k\} = \{x_{j+1}, \dots, x_n\}
 \end{array}$$

where  $\mathbf{y}$  is a set of  $j$  position parameters,  $\mathbf{z}$  is a set of  $k$  distance parameters, and the total number of parameters is  $n = j + k$ . This notation holds for each of the three test suites analyzed below.

1) *Deb’s Toolkit*: Deb’s toolkit for constructing two objective problems is the most well known of the very small number of toolkits for multiobjective problems of which we are aware. Creating a problem using Deb’s toolkit involves choosing three functions, a distribution function  $f_1$ , which tests an algorithm’s ability to diversify along the Pareto optimal front, a distance function  $g$ , which tests an algorithm’s ability to converge to the true Pareto optimal front, and a shape function  $h$ , which determines the shape of the Pareto optimal front. These three functions are related to one another according to the template shown in Fig. 8.

By decomposing problems into distinct functional units Deb has made it very easy for practitioners to construct problems with different characteristics. As a number of example functions are also provided, expert knowledge is not required.

TABLE II

FUNCTIONS PROVIDED BY DEB FOR  $f_1$ ,  $g$ , AND  $h$ . IF A FUNCTION IS COMPATIBLE WITH  $f_1$  OR  $g$ , THEN  $\mathbf{p}$  CAN BE SUBSTITUTED WITH  $\mathbf{y}$  OR  $\mathbf{z}$ , RESPECTIVELY. FOR BREVITY, F1 AND G1 HAVE BEEN OMITTED FROM THE TABLE, AS THEY ARE SUBSUMED BY F2 AND G2, RESPECTIVELY

Name	For	Function	Parameter Domains
F2	$f_1$	$\delta + \sum_{i=1}^{ \mathbf{p} } c_i p_i$ , where $\delta, c_1, \dots, c_{ \mathbf{p} } > 0$ .	Arbitrary
F3	$f_1$	$1 - \exp(-4r(\mathbf{p})) \sin^4(5\pi r(\mathbf{p}))$ , where $r(\mathbf{p}) = \sqrt{\sum_{i=1}^{ \mathbf{p} } p_i^2}$ .	$[0, 1]$
F4	$f_1, g$	$2.0 - \exp\left(-\left(\frac{p_1-0.2}{0.004}\right)^2\right) - 0.8 \exp\left(-\left(\frac{p_1-0.6}{0.4}\right)^2\right)$	$[0, 1]$
G2	$f_1, g$	$g_{\min} + (g_{\max} - g_{\min}) \left( \frac{\sum_{i=1}^{ \mathbf{p} } p_i - p_{i,\min}}{\sum_{i=1}^{ \mathbf{p} } p_{i,\max} - p_{i,\min}} \right)^\gamma$ , where $\gamma \neq 0$ .	Arbitrary
G3.i	$f_1, g$	Rastrigin: $1 + 10 \mathbf{p}  + \sum_{i=1}^{ \mathbf{p} } (p_i^2 - 10 \cos(2\pi p_i))$	$[-30, 30]$
G3.ii	$f_1, g$	Schwefel: $1 + (6.5\pi)^2  \mathbf{p}  - \sum_{i=1}^{ \mathbf{p} } p_i \sin(\sqrt{ p_i })$	$[-512, 511]$
G3.iii	$f_1, g$	Griewank: $2 + \sum_{i=1}^{ \mathbf{p} } p_i^2 / 4000 - \prod_{i=1}^{ \mathbf{p} } \cos(p_i / \sqrt{i})$	$[-512, 511]$
H1.i	$h$	$1/f_1$ , where $f_1 > 0$ .	NA
H1.ii	$h$	$\begin{cases} 1 - \left(\frac{f_1}{\beta g}\right)^\alpha, & f_1 \leq \beta g \\ 0, & f_1 > \beta g \end{cases}$ , where $f_1 \geq 0, g > 0$ , and $\alpha \leq 1$ .	NA
H2	$h$	As H1.ii, except $\alpha > 1$ .	NA
H3	$h$	As H1.ii, except $\alpha = 0.25 + 3.75 \frac{g - g_{\min}}{g_{\min}^* - g_{\min}}$ , where $g_{\min}^*$ is the weakest locally optimal fitness value of $g$ .	NA
H4	$h$	$1 - \left(\frac{f_1}{g}\right)^\alpha - \frac{f_1}{g} \sin(2\pi q f_1)$ , where $q$ is the number of disconnected regions in an unit interval of $f_1$ .	NA

Deb's toolkit is also the first to segregate parameters into distance and position parameters—mixed parameters are atypical. Moreover, the toolkit relates the three functional units such that whenever the distance function  $g$  is minimized, which is solely a function of the distance parameters, then the resultant solution is Pareto optimal. The shape of and position on the tradeoff surface is then determined by  $h$  and  $f_1$ , respectively. It is not possible for the position parameters to influence  $g$ , nor is it possible for the distance parameters to influence  $f_1$ , and although  $h$  can be a function of  $g$ , in most cases it is not.

The decomposition of problems used by Deb's toolkit both simplifies its design and analysis and greatly simplifies determining Pareto optimal fronts. In fact, substituting in the minimum value of  $g$  results in the parametric form of an equation that covers the Pareto optimal front. It is also relatively easy to construct the Pareto optimal set of problems constructed with Deb's toolkit.

Deb suggests a number of functions for  $f_1, g$ , and  $h$ , each of which is shown in Table II, with the exception of the binary encoded functions F5, G4, and G5. Using the requirements and features identified earlier, an analysis of these functions is given in Table III, where Recommendation 6 has been omitted in addition to Recommendation 4 (as commented earlier). We note that it should always be possible to enforce dissimilar Pareto optimal front tradeoff ranges using Deb's toolkit, where the exact tradeoff is function dependent.

Several of the functions suggested by Deb require further comment. First, the function H3 has the special property that the Pareto optimal front (when  $g$  is minimized) is convex, but the shape of suboptimal tradeoff surfaces (for example, when  $g$  is maximized) can be concave. In other words, the convexity of H3 changes as a function of  $g$ . Second, H4 creates a problem that is disconnected with respect to the Pareto optimal front (but not necessarily the Pareto optimal set), where the Pareto optimal

front generally consists of convex components, although it is possible to create mixed convex/concave components.

Care must also be used when employing G3.iii (Griewank's multimodal function) as the local optima counterintuitively decrease in number and complexity when the number of parameters is increased. In addition, although Griewank's function is nonseparable, the parameter dependencies are weak. Even in the worst case, optimizing parameters one-by-one will still result in a near optimal solution. The optimum of Griewank's function occurs when all of its parameters are zero, and while zero is not technically the middle of the domain ( $-0.5$ ), we consider it close enough to rate them as being medial parameters.

As we can see from Table II, the choice of  $f_1$  and  $g$  determines the number and the domains of the position and distance parameters, respectively. The choice of  $h$  (namely, H4) can result in there being mixed parameters instead of position parameters.

Depending on which functionals are multimodal, different effects occur. If  $f_1$  is multimodal, then irrespective of the number of position parameters, the problem will be Pareto many-to-one, and may also be biased (depending on the nature of the multimodality). Should  $g$  be multimodal, then the problem becomes "multifrontal," in that there exist distinct nondominated fronts that correspond to locally optimal  $g$  values. Lastly, should  $h$  be multimodal, then the Pareto optimal front could be disconnected, and  $\mathbf{y}$  will most likely be a vector of mixed parameters, not position parameters.

Although Deb's toolkit offers a number of advantages, it also has a number of significant limitations.

- It is limited to constructing two objective problems.
- No functions are suggested that facilitate the construction of problems with flat regions.
- No real valued deceptive functions are suggested.
- The suggested functions do not facilitate the construction of problems with degenerate or mixed Pareto optimal front geometries.

TABLE III  
ANALYSIS OF THE FUNCTIONS DESCRIBED FOR DEB'S TOOLKIT. DUE TO THE TOOLKIT NATURE, EACH FUNCTION IS ONLY ANALYZED WITH RESPECT TO WHAT IT IS COMPATIBLE WITH. FOR SIMPLICITY, AN NA IS STILL INDICATED IN SOME CASES WHERE THE FUNCTIONAL COULD (BUT IS NOT LIKELY TO) INFLUENCE THE OUTCOME

Name	For	R3: # Parameters	F2: Separability	F5: Modality	R1: No Extremal	R2: No Medial	R5: Diss. Domains	R7: Optima Known	F1: Geometry	F3: Bias	F4a: Pareto Many-to-one	F4b: Flat Regions
F2	$f_1$	✓	S	U	✓	✓	✓	✓	NA	—	+	—
F3	$f_1$	✓	S	M	✓	✓	×	✓	NA	+	+	—
F4	$f_1$	1	S	M	✓	✓	NA	✓	NA	—	+	—
G2	$f_1$	✓	S	U	✓	✓	✓	✓	NA	+	+	—
G3.i	$f_1$	✓	S	M	✓	✓	×	✓	NA	—	+	—
G3.ii	$f_1$	✓	S	M	✓	✓	×	✓	NA	—	+	—
G3.iii	$f_1$	✓	NS*	M*	✓	✓	×	✓	NA	—	+	—
F4	$g$	1	S	M	✓	✓	NA	✓	NA	—	—	—
G2	$g$	✓	S	U	×	✓	✓	✓	NA	+	—	—
G3.i	$g$	✓	S	M	✓	×	×	✓	NA	—	—	—
G3.ii	$g$	✓	S	M	✓	✓	×	✓	NA	—	—	—
G3.iii	$g$	✓	NS*	M*	✓	×	×	✓	NA	—	—	—
H1.i	$h$	NA	NA	U	NA	NA	NA	✓	convex	NA	NA	NA
H1.ii	$h$	NA	NA	U	NA	NA	NA	✓	convex or linear	NA	NA	NA
H2	$h$	NA	NA	U	NA	NA	NA	✓	concave	NA	NA	NA
H3	$h$	NA	NA	U	NA	NA	NA	✓	convex*	NA	NA	NA
H4	$h$	NA	NA	M	NA	NA	NA	✓	disconnected*	NA	NA	NA

- Only Griewank's function is nonseparable, and even then Griewank's function scales poorly and has but weak parameter dependencies.
- Position and distance parameters are always independent of one another. While this is beneficial in terms of constructing and analyzing test problems, it seems unlikely that it is representative of real-world problems.

Recognizing the importance of separability, Deb suggests a way of making position and distance parameters mutually nonseparable. Given  $\mathbf{R}$ , a random  $n \times n$  orthogonal matrix, the working parameter vector  $\mathbf{w}$  can be mapped to the actual parameter vector  $\mathbf{x}$  according to  $\mathbf{x} = \mathbf{R}\mathbf{w}$ . Objective functions employ the computed vector  $\mathbf{x}$ , and algorithms operate on the working vector  $\mathbf{w}$ .

Although this mechanism introduces dependencies, with the desired result being nonseparable position and distance parameters, it also leads to cyclical dependencies between parameters that limit the range of actual parameter vectors  $\mathbf{x}$  that can be created from the otherwise unrestricted working vector  $\mathbf{w}$ . As a result, it is quite possible that the Pareto optimal set will change in an unpredictable fashion, and one of the main benefits of employing Deb's toolkit, that of knowing the Pareto optimal set, will be lost.

2) *The ZDT Test Suite*: The suite of six test problems created by Zitzler *et al.* is perhaps the most widely employed suite of benchmark multiobjective problems in the EA literature. The ZDT problems can almost entirely be created using Deb's

toolkit, and as such share many of the same advantages and disadvantages.

The five real-valued ZDT problems are presented in Table IV, noting that ZDT5, the omitted problem, is binary encoded.<sup>2</sup> Table V provides an analysis of the ZDT problems. Elaborating on the table, ZDT3 is disconnected on both the Pareto optimal set and front, the latter of which consists of one mixed convex/concave component and several convex components. It should also be noted that ZDT4 only employs one parameter of dissimilar domain—namely, the single position parameter  $y_1$  has domain  $[0,1]$ , whereas all other parameters have domain  $[-5,5]$ . Given that the majority of parameters are of identical domain, we have listed ZDT4 as not conforming to Recommendation 5.

The ZDT problems share many of the characteristics already described for Deb's toolkit, including how multimodality can cause Pareto many-to-one problems (ZDT6), disconnected problems (ZDT3), and so-called multifrontal problems (ZDT4). Importantly, all of the ZDT problems employ only one position parameter, meaning  $f_1$  is a function of only one parameter.

The ZDT test suite offers two main advantages: the Pareto optimal fronts of its problems are well defined and test results from a variety of other research papers are commonly available, which facilitates comparisons with new algorithms. However,

<sup>2</sup>Incidentally, due to being binary encoded, ZDT5 has often been omitted from analysis elsewhere in the EA literature.

TABLE IV  
THE FIVE REAL-VALUED ZDT TWO OBJECTIVE PROBLEMS. SIMILAR TO DEB'S TOOLKIT, THE SECOND OBJECTIVE IS  $f_2(\mathbf{y}, \mathbf{z}) = g(\mathbf{z})h(f_1(\mathbf{y}), g(\mathbf{z}))$ , WHERE BOTH OBJECTIVES ARE TO BE MINIMIZED

Name	Problem	Parameter Domains
ZDT1	$f_1 = y_1$ $g = 1 + 9 \sum_{i=1}^k z_i/k$ $h = 1 - \sqrt{f_1/g}$	$[0, 1]$
ZDT2	as ZDT1, except $h = 1 - (f_1/g)^2$ .	$[0, 1]$
ZDT3	as ZDT1, except $h = 1 - \sqrt{f_1/g} - (f_1/g)\sin(10\pi f_1)$ .	$[0, 1]$
ZDT4	as ZDT1, except $g = 1 + 10k + \sum_{i=1}^k (z_i^2 - 10\cos(4\pi z_i))$	$y_1 \in [0, 1]$ $z_1, \dots, z_k \in [-5, 5]$
ZDT6	$f_1 = 1 - \exp(-4y_1)\sin^6(6\pi y_1)$ $g = 1 + 9(\sum_{i=1}^k z_i/k)^{0.25}$ $h = 1 - (f_1/g)^2$	$[0, 1]$

TABLE V  
ANALYSIS OF THE ZDT PROBLEMS

Name	Objective	R3: # Parameters	F2: Separability	F5: Modality	R1: No Extremal	R2: No Medial	R5: Diss. Domains	R6: Diss. Ranges	R7: Optima Known	F1: Geometry	F3: Bias	F4a: Pareto Many-to-one	F4b: Flat Regions
ZDT1	$f_1$	1	S	U	×	✓	×	×	✓	convex	—	—	—
	$f_2$	✓	S	U									
ZDT2	$f_1$	1	S	U	×	✓	×	×	✓	concave	—	—	—
	$f_2$	✓	S	U									
ZDT3	$f_1$	1	S	U	×	✓	×	✓	✓	disconnected*	—	—	—
	$f_2$	✓	S	M									
ZDT4	$f_1$	1	S	U	✓	×	×	×	✓	convex	—	—	—
	$f_2$	✓	S	M									
ZDT6	$f_1$	1	S	M	×	✓	×	✓	✓	concave	+	+	—
	$f_2$	✓	S	M									

despite being an immensely popular test suite, it has numerous shortcomings.

- It only has problems with two objectives.
- None of its problems has fitness landscapes with flat regions.
- Its only deceptive problem is binary encoded.
- None of its problems has a degenerate Pareto optimal front.
- None of its problems is nonseparable.
- Only the number of distance parameters is scalable.
- All except ZDT4's distance parameters are extremal parameters, and even those are medial parameters.

Although the ZDT test suite is popular, by itself it is far from comprehensive.

3) *The DTLZ Test Suite*: The DTLZ suite of benchmark problems, created by Deb *et al.*, is unlike the majority of multiobjective test problems in that the problems are scalable to any number of objectives. This is an important characteristic

that has facilitated several recent investigations into what are commonly called “many” objective problems.

Nine test problems are included in the DTLZ test suite,<sup>3</sup> of which the first seven are shown in Table VI. DTLZ8 and DTLZ9 have side constraints, hence their omission from this paper. Table VII presents an analysis of the DTLZ problems.

Further to the content of Table VII, Deb *et al.* suggest a modification to DTLZ2 which involves a mapping that averages sets of adjacent working position parameters to arrive at the set of computed position parameters. Whilst such a mapping introduces some level of parameter dependencies, it does not change the analysis of DTLZ2.

<sup>3</sup>In fact, the number of DTLZ problems depends on which paper is being referred to. In this paper, we are using the nine DTLZ problems from the original technical report [23]. A more recent conference paper version of the technical report also exists [33], in which only seven of the original nine problems were reproduced (DTLZ5 and DTLZ9 from the original paper were dropped). As a result, DTLZ6 from the original technical report is DTLZ5 in the conference paper, DTLZ7 is DTLZ6 in the conference paper, and DTLZ8 is DTLZ7 in the conference paper.

TABLE VI  
SEVEN OF THE NINE DTLZ MANY OBJECTIVE PROBLEMS. ALL OBJECTIVES ARE TO BE MINIMIZED

Name	Problem	Parameter Domains
DTLZ1	$f_1 = (1+g)0.5 \prod_{i=1}^{M-1} y_i$ $f_{m=2:M-1} = (1+g)0.5 \left( \prod_{i=1}^{M-m} y_i \right) (1 - y_{M-m+1})$ $f_M = (1+g)0.5(1 - y_1)$ $g = 100 \left[ k + \sum_{i=1}^k ((z_i - 0.5)^2 - \cos(20\pi(z_i - 0.5))) \right]$	[0, 1]
DTLZ2	$f_1 = (1+g) \prod_{i=1}^{M-1} \cos(y_i \pi / 2)$ $f_{m=2:M-1} = (1+g) \left( \prod_{i=1}^{M-m} \cos(y_i \pi / 2) \right) \sin(y_{M-m+1} \pi / 2)$ $f_M = (1+g) \sin(y_1 \pi / 2)$ $g = \sum_{i=1}^k (z_i - 0.5)^2$	[0, 1]
DTLZ3	As DTLZ2, except the equation for $g$ is replaced by the one from DTLZ1.	[0, 1]
DTLZ4	As DTLZ2, except all $y_i \in \mathbf{y}$ are replaced by $y_i^\alpha$ , where $\alpha > 0$ .	[0, 1]
DTLZ5	As DTLZ2, except all $y_2, \dots, y_{M-1} \in \mathbf{y}$ are replaced by $\frac{1+2gy_i}{2(1+g)}$ .	[0, 1]
DTLZ6	As DTLZ5, except the equation for $g$ is replaced by $g = \sum_{i=1}^k z_i^{0.1}$ .	[0, 1]
DTLZ7	$f_{m=1:M-1} = y_m$ $f_M = (1+g) \left( M - \sum_{i=1}^{M-1} \left[ \frac{f_i}{1+g} (1 + \sin(3\pi f_i)) \right] \right)$ $g = 1 + 9 \sum_{i=1}^k z_i / k$	[0, 1]

TABLE VII  
ANALYSIS OF THE DTLZ PROBLEMS

Name	Objective	R3: # Parameters	F2: Separability	F5: Modality	R1: No Extremal	R2: No Medial	R5: Diss. Domains	R6: Diss. Ranges	R7: Optima Known	F1: Geometry	F3: Bias	F4a: Pareto Many-to-one	F4b: Flat Regions
DTLZ1	$f_{1:M}$	✓*	S*	M	✓	×	×	×	✓	linear	—	+	—
DTLZ2*	$f_{1:M}$	✓*	S*	U	✓	×	×	×	✓	concave	—	+	—
DTLZ3	$f_{1:M}$	✓*	S*	M	✓	×	×	×	✓	concave	—	+	—
DTLZ4	$f_{1:M}$	✓*	S*	U	✓	×	×	×	✓	concave	+	+	—
DTLZ5*	$f_{1:M}$	✓*	?	U	?	?	×	?	×	?	—	+	—
DTLZ6*	$f_{1:M}$	✓*	?	U	?	?	×	?	×	?	+	+	—
DTLZ7	$f_{1:M-1}$ $f_M$	1 ✓	NA S	U M	×	✓	×	×	✓	disconnected*	—	—	—

Additionally, all of DTLZ1–DTLZ6 are scalable with respect to the number of distance parameters, but have a fixed number of  $M-1$  position parameters, where  $M$  is the number of objectives. Note also that the objective functions of DTLZ1–DTLZ4 have multiple global optima since terms such as  $\cos(y_i \pi / 2)$  can evaluate to zero, thereby allowing flexibility in the selection of other parameter values. Technically speaking, these objectives are nonseparable, as attempting to optimize them one parameter at a time (in only one pass) will not identify all global optima. As this is a minor point, we classify the objectives of DTLZ1–DTLZ4 as being separable irrespective, as attempting to optimize them one parameter at a time will identify at least one global optima. Incidentally, there being multiple global optima is why many of the DTLZ problems are Pareto many-to-one.

DTLZ7 is disconnected in both the Pareto optimal set and the Pareto optimal front, where the Pareto optimal front gen-

erally consists of convex components, with some mixed convexity/concavity. We have also indicated that DTLZ7 does not satisfy Requirement 6, as the majority of objectives (all but objective  $M$ ) have identical Pareto front tradeoff magnitudes.

Both DTLZ5 and DTLZ6 also deserve special mention. DTLZ5 and DTLZ6 are both claimed to be problems with degenerate Pareto optimal fronts; the Pareto optimal fronts are meant to be an arc embedded in  $M$ -objective space. However, we have found that this is untrue for instances with four or more objectives. The problem arises from the expectation that the minimization of  $g$  (when  $g = 0$ ) results in a Pareto optimal solution. Unfortunately, this is not true with DTLZ5 or DTLZ6: it is possible to have Pareto optimal solutions which correspond to a nonzero  $g$  value.

By way of demonstration, consider a four objective version of DTLZ5. Let  $g = 10$  (which is possible when  $k \geq 40$ ),  $y_1 = 0$ ,  $y_2 = 0.95$ , and  $y_3 = 0$ . Evaluating the four objective

DTLZ5 with these values yields, for the second and last objective,  $f_2 = 11 \cos(0) \cos(20\pi/44) \sin(\pi/44) \approx 0.11168$  and  $f_4 = 11 \sin(0) = 0$ . On the other hand, evaluating for the same parameters, but with  $g = 0$  (the supposed optimum), yields  $f_2 = \cos(0) \cos(\pi/4) \sin(\pi/4) = 0.5$  and  $f_4 = \sin(0) = 0$ . In the case of  $g = 10$ ,  $f_2$  is better than in the case of  $g = 0$ , and in fact the two solutions are mutually nondominating. If the assertion that  $g = 0$  is required for a solution to be Pareto optimal, then there must exist a different parameter vector that dominates our example with  $g = 10$ . When  $g = 0$ , the only parameter that influences fitness values is  $y_1$  ( $y_2$  and  $y_3$  become irrelevant). Consequently, it should be possible to vary  $y_1$  in such a way that our  $g = 10$  case becomes dominated. However,  $y_1$  can only be increased from zero, and while doing so is required to improve  $f_2$ , doing so will also worsen  $f_4$ , where any worsening of  $f_4$  will once again result in mutually nondominating solutions. As such, it is not possible for a parameter vector that corresponds to  $g = 0$  to dominate our example with  $g = 10$ . A similar example can be constructed for DTLZ6.

As a result of this, the nature of DTLZ5 and DTLZ6's Pareto optimal fronts is unclear beyond three objectives, hence our incomplete analysis.

Aside from being able to employ many objectives, the DTLZ problems differ in a few key areas from Deb's toolkit and the ZDT problems. First, the distance functions ( $g$ s) are incorporated into all objective functions (barring those of DTLZ7), meaning objectives tend to be a function of both distance and position parameters, not just position parameters alone. Second, varying the position parameters alone never causes multimodal behavior (noting that DTLZ7 has mixed parameters, not position parameters). Such behavior was exhibited with multimodal  $f_1$  functions in Deb's toolkit and ZDT6.

The DTLZ problems represent a considerable step forward, as they allow researchers to investigate the properties of many objective problems in a controlled manner, with known problem characteristics and knowledge of the Pareto optimal front. However, as with Deb's toolkit and the ZDT problems, the DTLZ test suite has several limitations.

- None of its problems features fitness landscapes with flat regions.
- None of its problems is deceptive.
- None of its problems is (practically) nonseparable.
- The number of position parameters is always fixed relative to the number of objectives.

In some ways, the DTLZ problems are the inverse of Deb's toolkit: the former offers flexibility with respect to the number of objectives, whereas the latter offers flexibility with respect to problem features and construction. What is needed is an approach that offers both.

### B. Van Veldhuizen's Test Suite

There are many multiobjective test problems in the literature, and it is common for a subset of them to be collected together as a test suite. A sizeable example of such a test suite is the one employed by Van Veldhuizen [32]. The problems employed by Van Veldhuizen are largely representative of the types of problems employed in the literature prior to the publication of Deb's toolkit and the ZDT problems.

In addition to a number of problems with side constraints, Van Veldhuizen employs seven multiobjective test problems from the literature, as shown in Table VIII. The original authors of MOP1–MOP7 are as follows: MOP1 is due to Schaffer [34]; MOP2 is due to Fonseca and Fleming [35] (originally parameters had domain  $[-2,2]$ ); MOP3 is due to Poloni *et al.* [36]; MOP4 is based on Kursawe [37] (as indicated by Deb [27], the form employed by Van Veldhuizen, which is limited to three parameters, and uses the term  $\sin(x_i^3)$  instead of  $\sin^3(x_i)$ , proves more tractable to analysis); MOP5: due to Viennet *et al.* [38] (originally parameters had domain  $[-3,3]$ ); MOP6 is constructed using Deb's toolkit (F1, G1, H4); MOP7 is due to Viennet *et al.* [38] (originally parameters had domain  $[-4,4]$ ).

Many of the problems employed by Van Veldhuizen are less methodically constructed than those presented in Section VI-A, and thus tend to be more difficult to analyze. However, they also include properties not exercised by Deb's toolkit, the ZDT problems, or the DTLZ problems.

To see this, consider the analysis of Van Veldhuizen's test suite given in Table IX. Of particular interest is that several problems, namely, MOP3 and MOP5, are both nonseparable and multimodal, characteristics that are known to be more representative of real-world problems. Also of interest is the variety of unusual Pareto optimal geometries exhibited by Van Veldhuizen's test suite, including problems with disconnected, and sometimes degenerate, fronts.

Our analysis of MOP3, MOP4, and MOP5 is based on existing work by Deb [27]. Deb comments that it is difficult to know the Pareto optimal set for MOP3, and it appears from Deb's analysis of MOP4 that determining MOP4's Pareto optimal set was nontrivial.

MOP3–MOP6 are disconnected with respect to their Pareto optimal sets and fronts. MOP3's Pareto optimal front consists of convex components. MOP4's front includes a degenerate point and several mixed convex/concave components. A degenerate convex line and a degenerate mixed convex/concave line comprise MOP5's Pareto optimal front, and MOP6 has a front with convex and mixed convex/concave components.

MOP7 is connected, and appears to have a convex Pareto optimal front, with some regions tending toward degenerate lines.

Another way in which Van Veldhuizen's test suite distinguishes itself from Deb's toolkit, the ZDT problems, and the DTLZ problems, is that most parameters are mixed parameters. There are relatively few position and distance parameters.

Van Veldhuizen's test suite is not without limitations.

- Most have only two or three parameters.
- None of the nonseparable problems is scalable parameter-wise.
- None of the problems is scalable objective-wise.
- The ad hoc nature of many of the test problems makes them difficult to analyze.
- Only MOP3, MOP4, and MOP5 have neither extremal nor medial parameters.
- None of the problems is deceptive.
- None of the problems has flat regions.
- None of the problems is Pareto many-to-one.

TABLE VIII  
VAN VALEDHUIZEN'S TEST SUITE (NOT INCLUDING PROBLEMS WITH SIDE CONSTRAINTS).  
ALL OBJECTIVES OTHER THAN THOSE OF MOP3 ARE TO BE MINIMIZED

Name	Problem	Parameter Domains
MOP1	$f_1(x) = x^2$ $f_2(x) = (x-2)^2$	$[-10^5, 10^5]$
MOP2	$f_1(x_1, \dots, x_n) = 1 - \exp(-\sum_{i=1}^n (x_i - 1/\sqrt{n})^2)$ $f_2(x_1, \dots, x_n) = 1 - \exp(-\sum_{i=1}^n (x_i + 1/\sqrt{n})^2)$	$[-4, 4]$
MOP3	$\text{Max. } f_1(x_1, x_2) = -1 - (A_1 - B_1)^2 - (A_2 - B_2)^2$ $\text{Max. } f_2(x_1, x_2) = -(x_1 + 3)^2 - (x_2 + 1)^2$ $A_1 = 0.5 \sin 1 - 2 \cos 1 + \sin 2 - 1.5 \cos 2$ $A_2 = 1.5 \sin 1 - \cos 1 + 2 \sin 2 - 0.5 \cos 2$ $B_1 = 0.5 \sin x_1 - 2 \cos x_1 + \sin x_2 - 1.5 \cos x_2$ $B_2 = 1.5 \sin x_1 - 1 \cos x_1 + 2 \sin x_2 - 0.5 \cos x_2$	$[-\pi, \pi]$
MOP4	$f_1(x_1, x_2, x_3) = \sum_{i=1}^2 -10 \exp^{-0.2\sqrt{x_i^2 + x_{i+1}^2}}$ $f_2(x_1, x_2, x_3) = \sum_{i=1}^3  x_i ^{0.8} + 5 \sin(x_i^3)$	$[-5, 5]$
MOP5	$f_1(x_1, x_2) = 0.5(x_1^2 + x_2^2) + \sin(x_1^2 + x_2^2)$ $f_2(x_1, x_2) = \frac{(3x_1 - 2x_2 + 4)^2}{8} + \frac{(x_1 - x_2 + 1)^2}{27} + 15$ $f_3(x_1, x_2) = \frac{1}{x_1^2 + x_2^2 + 1} - 1.1 \exp(-x_1^2 - x_2^2)$	$[-30, 30]$
MOP6	$f_1(x_1) = x_1$ $f_2(x_1, x_2) = (1 + 10x_2) \left( 1 - \left( \frac{x_1}{1 + 10x_2} \right)^2 - \frac{x_1}{1 + 10x_2} \sin(8\pi x_1) \right)$	$[0, 1]$
MOP7	$f_1(x_1, x_2) = \frac{(x_1 - 2)^2}{2} + \frac{(x_2 + 1)^2}{13} + 3$ $f_2(x_1, x_2) = \frac{(x_1 + x_2 - 3)^2}{36} + \frac{(-x_1 + x_2 + 2)^2}{8} - 17$ $f_3(x_1, x_2) = \frac{(x_1 + 2x_2 - 1)^2}{175} + \frac{(-x_1 + 12x_2)^2}{17} - 13$	$[-400, 400]$

TABLE IX  
ANALYSIS OF VAN VALEDHUIZEN’S TEST SUITE

[illegible]



### C. Other Test Problems in Brief

Many multiobjective test problems have been employed in the EA literature. A listing of many of these is presented in Table XVI, and an analysis of each is given in Table XVII (both tables are in the Appendix). To be thorough, even “toy” problems, designed to test specific concepts, have been analyzed. Note that scalable (parameter-wise) problem extensions are sometimes possible. For example, an  $n$  parameter form of LE1 is not difficult to derive. Despite this, for the sake of simplicity, we strictly restrict our analysis to published material.

As with the test problems employed by Van Veldhuizen, the ad hoc nature of some of these test problems complicates their analysis. Indeed, not all authors describe (or even claim to know) the Pareto optimal geometries of the test problems they themselves have created.

It is difficult to collectively summarize such a large number of test problems. Even so, there are some common limitations.

- None of the problems is deceptive, nor do any problems have flat regions in their fitness landscape.
- Many of the test problems are defined with respect to only one or two parameters, a number that Bäck and Michalewicz [29] specifically state insufficiently represents real-world problems.
- ZLT1 is the only test problem that is scalable objective-wise, but unfortunately it is neither multimodal nor nonseparable.
- There are few problems with both multimodal and nonseparable objectives, none of which is both scalable parameter-wise and has known Pareto optima.

In addition to Van Veldhuizen’s test suite, other authors have also collected subsets of the problems into distinct test suites. Examples include Bentley and Wakefield’s test suite [39], which consists of a single objective problem, MOP1, Sch1, and FF1; Knowles and Corne’s test suite [40], which extends Bentley and Wakefield’s test suite to include an integer valued nonseparable multiobjective problem with flat regions and isolated optima, and a real-world problem; and a test suite by Zitzler *et al.* [11], which includes ZLT1, ZDT6, QV1, Kur1, and a binary encoded knapsack problem.

A recent addition to the literature is a method due to Okabe for constructing test problems, with a high degree of control over bias and the shape of the Pareto front [41], [42]. The focus of Okabe’s work is a novel type of estimation of distribution algorithm (EDA), based on Voronoi tessellation. Since it functions using an estimate of the distribution of good solutions, the performance of the new algorithm would obviously be expected to depend on the nature of this distribution in both parameter- and objective-space, especially near the Pareto front. Therefore, test problems with varying difficult distributions are required.

The method does not attempt to control other problem features, such as modality or separability—these are not its focus. However, it is an interesting example of a toolkit approach, and has the advantage that the Pareto fronts can be simply located. In [41], the toolkit is presented, and is then used to construct a test suite. The author also makes the interesting observation that

most existing problems have Pareto fronts that are mostly piecewise linear, and in that sense are unlikely to be representative of real problems.

### VII. LIMITATIONS OF EXISTING WORK

In Section V, we identified several guidelines for the composition of test suites. From the literature review in the previous section, it is apparent that some of these guidelines are inadequately supported by existing test problems.

One of the primary issues is the number of problems that do not adhere to the recommendations of Section IV. Of the 54 problems considered (Deb’s toolkit aside), none satisfies all of the recommendations.

Secondly, how well do existing multiobjective problems support the recommended composition of a test suite? Although unimodal and multimodal test problems and problems with a variety of different Pareto optimal front geometries are well represented, nonseparable problems are seriously underrepresented, and there are no suitable problems that are both nonseparable and employ multimodal objective functions. Although this combination of features is reflected in the examined literature, in each case at least one of the more important recommendations is not satisfied, typically Recommendations 3 (scalable parameters) or 7 (Pareto optima known).

Considering the importance of multimodal and nonseparable problems, that they are not commonplace is a little surprising. One possible explanation is the difficulty of conceiving scalable, multimodal, nonseparable problems that adhere to all of the recommendations. To demonstrate that such a problem exists, consider the following, where all objectives are to be minimized:

$$\begin{aligned}
 f_1(\mathbf{y}, \mathbf{z}) &= g(\mathbf{z}) + 1 - \prod_{i=1}^{M-1} \cos(u_i \pi / 2) \\
 f_{m=2:M-1}(\mathbf{y}, \mathbf{z}) &= g(\mathbf{z}) + 1 - \left( \prod_{i=1}^{M-m} \cos(u_i \pi / 2) \right) \\
 &\quad \times \sin(u_{M-m+1} \pi / 2) \\
 f_M(\mathbf{y}, \mathbf{z}) &= g(\mathbf{z}) + 1 - \sin(u_1 \pi / 2) \\
 g(\mathbf{z}) &= \sum_{i=1}^k \left( v_i / k + 4 \sum_{j=i+1}^k |v_i - v_j| / k^2 \right) \\
 u_i &= \frac{M-1}{l} \sum_{j=1+(i-1)\frac{l}{M-1}}^{i\frac{l}{M-1}} y_j \\
 v_i &= (|z_i - 0.3| + 1 + \cos(10\pi z_i)) / 2.7
 \end{aligned}$$

where  $\mathbf{y}$  and  $\mathbf{z}$  are vectors of  $l$  position and  $k$  distance parameters, respectively,  $l \geq M-1$ ,  $l \bmod M-1 = 0$ ,  $k \bmod 2 = 0$ , and all parameters have domain  $[0, 1]$ . A parameter vector is Pareto optimal so long as  $g$  is minimized, which is achieved by setting all  $z_1, \dots, z_k = 0.3$ .

All of the objectives of the above problem are both multimodal and nonseparable. We assume that the problem has been normalized in parameter space and in fitness space. The problem thus adheres to all of our recommendations. Fig. 9 shows the

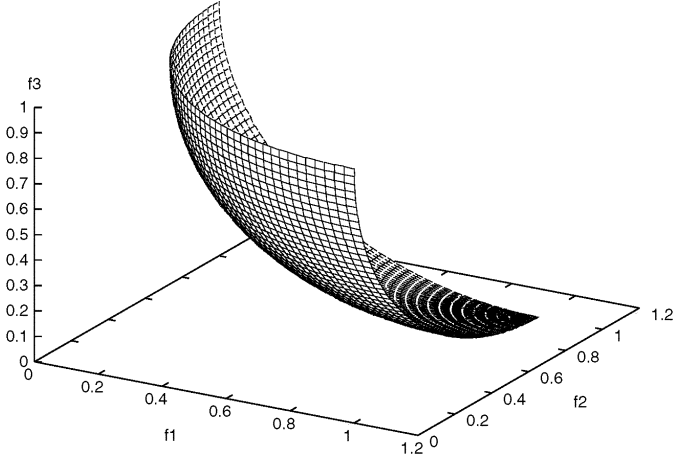


Fig. 9. The three objective Pareto optimal front for our nonseparable, multimodal, many objective problem.

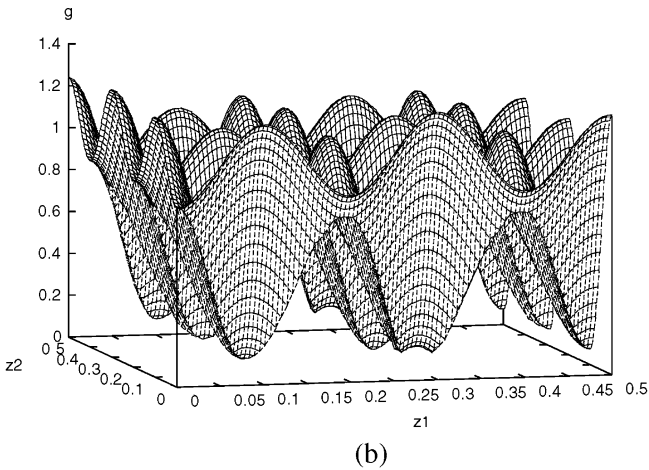
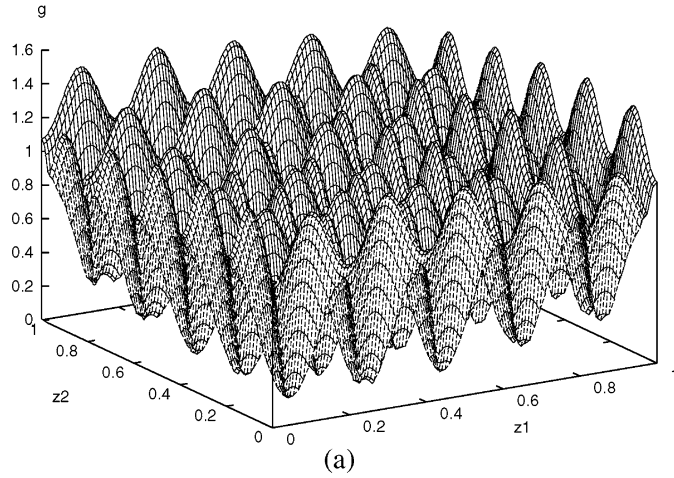


Fig. 10. The fitness landscape for the two parameter instance of our multimodal, nonseparable  $g$  function. The first instance shows the entire landscape of  $g$ , whereas the second instance shows the region about which  $g$  is optimal. (a) Fitness landscape of  $g$ . (b) Fitness landscape of  $g$ , where the domain of  $z_1$  and  $z_2$  is restricted.

Pareto optimal front for this problem. The  $g$  function is plotted separately in Fig. 10.

Less conspicuously, some other features are not represented by the reviewed test problems. There were no real-valued deceptive problems, or problems with flat regions. An example of

a real-valued deceptive problem was given in Fig. 1(a), and a real-valued test problem with flat regions can trivially be constructed using piecewise functions.

## VIII. A TEST FUNCTION TOOLKIT AND A SUGGESTED TEST SUITE

In this section, we describe the WFG Toolkit, first introduced in [43], a toolkit that can be used to design test problems meeting our recommendations, and exhibiting a desired set of features. We hope that this toolkit will ease the task of researchers wishing to improve the rigor and quality of their MOEA testing.<sup>4</sup> To illustrate the toolkit, we use it to construct a test suite that consists of nine scalable, multiobjective test problems (WFG1–WFG9) focussing on some of the more pertinent problem characteristics. Table XIV specifies WFG1–WFG9, the properties of which are summarized in Table XV.

The WFG Toolkit defines a problem in terms of an underlying vector of parameters  $\mathbf{x}$ . The vector  $\mathbf{x}$  is always associated with a simple underlying problem that defines the fitness space. The vector  $\mathbf{x}$  is derived, via a series of transition vectors, from a vector of working parameters  $\mathbf{z}$ . Each transition vector adds complexity to the underlying problem, such as multimodality and nonseparability. The EA directly manipulates  $\mathbf{z}$ , through which  $\mathbf{x}$  is indirectly manipulated.

Unlike previous test suites in which complexity is “hard-wired” in an ad hoc manner, the WFG Toolkit allows a test problem designer to control, via a series of composable transformations, which features will be present in the test problem. To create a problem, the test problem designer selects several shape functions to determine the geometry of the fitness space, and employs a number of transformation functions that facilitate the creation of transition vectors. Transformation functions must be designed carefully such that the underlying fitness space (and Pareto optimal front) remains intact with a relatively easy to determine Pareto optimal set. The WFG Toolkit provides a variety of predefined shape and transformation functions to help ensure this is the case.

For convenience, working parameters are labeled as either distance- or position-related parameters (even if they are actually mixed parameters), depending on the type of the underlying parameter being mapped to.

All problems created by the WFG Toolkit conform to the following format:

$$\text{Given } \mathbf{z} = \{z_1, \dots, z_k, z_{k+1}, \dots, z_n\}$$

$$\text{Minimize } f_{m=1:M}(\mathbf{x}) = Dx_M + S_m h_m(x_1, \dots, x_{M-1})$$

$$\text{where } \mathbf{x} = \{x_1, \dots, x_M\}$$

$$= \{\max(t_M^p, A_1)(t_1^p - 0.5) + 0.5, \dots,$$

$$\max(t_M^p, A_{M-1})(t_{M-1}^p - 0.5) + 0.5, t_M^p\}$$

$$\mathbf{t}^p = \{t_1^p, \dots, t_M^p\} \leftarrow \mathbf{t}^{p-1} \leftarrow \dots \leftarrow \mathbf{t}^1 \leftarrow \mathbf{z}_{[0,1]}$$

$$\mathbf{z}_{[0,1]} = \{z_{1,[0,1]}, \dots, z_{n,[0,1]}\}$$

$$= \{z_1/z_{1,\max}, \dots, z_n/z_{n,\max}\}$$

<sup>4</sup>Source files in C++ supporting the toolkit are available from <http://www.wfg.csse.uwa.edu.au/>.

TABLE X  
SHAPE FUNCTIONS. IN ALL CASES,  $x_1, \dots, x_{M-1} \in [0, 1]$ .  $A, \alpha$ , AND  $\beta$  ARE CONSTANTS

<b>Linear</b>	
$\text{linear}_1(x_1, \dots, x_{M-1})$	$= \prod_{i=1}^{M-1} x_i$
$\text{linear}_{m=2:M-1}(x_1, \dots, x_{M-1})$	$= \left( \prod_{i=1}^{M-m} x_i \right) (1 - x_{M-m+1})$
$\text{linear}_M(x_1, \dots, x_{M-1})$	$= 1 - x_1$
When $h_{m=1:M} = \text{linear}_m$ , the Pareto optimal front is a linear hyperplane, where $\sum_{m=1}^M h_m = 1$ .	
<b>Convex</b>	
$\text{convex}_1(x_1, \dots, x_{M-1})$	$= \prod_{i=1}^{M-1} (1 - \cos(x_i \pi / 2))$
$\text{convex}_{m=2:M-1}(x_1, \dots, x_{M-1})$	$= \left( \prod_{i=1}^{M-m} (1 - \cos(x_i \pi / 2)) \right) (1 - \sin(x_{M-m+1} \pi / 2))$
$\text{convex}_M(x_1, \dots, x_{M-1})$	$= 1 - \sin(x_1 \pi / 2)$
When $h_{m=1:M} = \text{convex}_m$ , the Pareto optimal front is purely convex.	
<b>Concave</b>	
$\text{concave}_1(x_1, \dots, x_{M-1})$	$= \prod_{i=1}^{M-1} \sin(x_i \pi / 2)$
$\text{concave}_{m=2:M-1}(x_1, \dots, x_{M-1})$	$= \left( \prod_{i=1}^{M-m} \sin(x_i \pi / 2) \right) \cos(x_{M-m+1} \pi / 2)$
$\text{concave}_M(x_1, \dots, x_{M-1})$	$= \cos(x_1 \pi / 2)$
When $h_{m=1:M} = \text{concave}_m$ , the Pareto optimal front is purely concave, and a region of the hyper-sphere of radius one centred at the origin, where $\sum_{m=1}^M h_m^2 = 1$ .	
<b>Mixed convex/concave</b> ( $\alpha > 0, A \in \{1, 2, \dots\}$ )	
$\text{mixed}_M(x_1, \dots, x_{M-1})$	$= \left( 1 - x_1 - \frac{\cos(2A\pi x_1 + \pi/2)}{2A\pi} \right)^\alpha$
Causes the Pareto optimal front to contain both convex and concave segments, the number of which is controlled by $A$ . The overall shape is controlled by $\alpha$ : when $\alpha > 1$ or when $\alpha < 1$ , the overall shape is convex or concave respectively. When $\alpha = 1$ , the overall shape is linear.	
<b>Disconnected</b> ( $\alpha, \beta > 0, A \in \{1, 2, \dots\}$ )	
$\text{disc}_M(x_1, \dots, x_{M-1})$	$= 1 - (x_1)^\alpha \cos^2(A(x_1)^\beta \pi)$
Causes the Pareto optimal front to have disconnected regions, the number of which is controlled by $A$ . The overall shape is controlled by $\alpha$ (when $\alpha > 1$ or when $\alpha < 1$ , the overall shape is concave or convex respectively, and when $\alpha = 1$ , the overall shape is linear), and $\beta$ influences the location of the disconnected regions (larger values push the location of disconnected regions towards larger values of $x_1$ , and vice versa).	

where  $M$  is the number of objectives,  $\mathbf{x}$  is a set of  $M$  underlying parameters (where  $x_M$  is an underlying distance parameter and  $x_{1:M-1}$  are underlying position parameters),  $\mathbf{z}$  is a set of  $k + l = n \geq M$  working parameters (the first  $k$  and the last  $l$  working parameters are position- and distance-related parameters, respectively),  $D > 0$  is a distance scaling constant (equal to one in [43]),  $A_{1:M-1} \in \{0, 1\}$  are degeneracy constants (for each  $A_i = 0$ , the dimensionality of the Pareto optimal front is reduced by one),  $h_{1:M}$  are shape functions,  $S_{1:M} > 0$  are scaling constants, and  $\mathbf{t}^{1:p}$  are transition vectors, where “ $\leftarrow$ ” indicates that each transition vector is created from another vector via transformation functions. The domain of all  $z_i \in \mathbf{z}$  is  $[0, z_{i,\max}]$  (the lower bound is always zero for convenience), where all  $z_{i,\max} > 0$ . Note that all  $x_i \in \mathbf{x}$  will have domain  $[0, 1]$ .

Some observations can be made about the above formalism: substituting in  $x_M = 0$  and disregarding all transition vectors provides a parametric equation that covers and is covered by the Pareto optimal front of the actual problem, working parameters can have dissimilar domains (which would encourage EAs to normalize parameter domains), and employing dissimilar scaling constants results in dissimilar Pareto optimal front tradeoff ranges (this is more representative of real-world problems and encourages EAs to normalize fitness values).

*Shape functions* determine the nature of the Pareto optimal front, and map parameters with domain  $[0, 1]$  onto the range  $[0, 1]$ . Table X presents five different types of shape functions. Each of  $h_{1:M}$  must be associated with a shape function. For example, if all of  $h_{1:M}$  are linear, the Pareto optimal front is a hyperplane; if they are all convex, it is a convex hypersurface; if they are all concave, it is concave; if mixed, it is a controlled mixture of convex and concave segments; while disconnected causes it to have disconnected regions, in a controlled manner.

*Transformation functions* map input parameters with domain  $[0, 1]$  onto the range  $[0, 1]$ . The transformation functions are specified in Table XI. To ensure problems are well designed, some restrictions apply as given in Table XII. For brevity, we have omitted a weighted product reduction function (analogous to the weighted sum reduction function).

Bias transformations impact the search process by biasing the fitness landscape. Shift transformations move the location of optima. In the absence of any shift, all distance-related parameters would be extremal parameters, with optimal value at zero. Shift transformations can be used to set the location of parameter optima (subject to skewing by bias transformations), which is useful if medial and extremal parameters are to be avoided. We recommend that all distance-related parameters be subjected

TABLE XI  
TRANSFORMATION FUNCTIONS. THE PRIMARY PARAMETERS  $y$  AND  $y_1, \dots, y_{|y|}$  ALWAYS HAVE DOMAIN  $[0,1]$ .  $A, B, C, \alpha$ , AND  $\beta$  ARE CONSTANTS. FOR  $\mathbf{b\_param}$ ,  $\mathbf{y'}$  IS A VECTOR OF SECONDARY PARAMETERS (OF DOMAIN  $[0,1]$ ), AND  $u$  IS A REDUCTION FUNCTION

<p><b>Bias: Polynomial</b> (<math>\alpha &gt; 0, \alpha \neq 1</math>)</p> $\mathbf{b\_poly}(y, \alpha) = y^\alpha$ <p>When <math>\alpha &gt; 1</math> or when <math>\alpha &lt; 1</math>, <math>y</math> is biased towards zero or towards one respectively.</p>
<p><b>Bias: Flat Region</b> (<math>A, B, C \in [0, 1], B &lt; C, B = 0 \Rightarrow A = 0 \wedge C \neq 1, C = 1 \Rightarrow A = 1 \wedge B \neq 0</math>)</p> $\mathbf{b\_flat}(y, A, B, C) = A + \min(0, \lfloor y - B \rfloor) \frac{A(B-y)}{B} - \min(0, \lfloor C - y \rfloor) \frac{(1-A)(y-C)}{1-C}$ <p>Values of <math>y</math> between <math>B</math> and <math>C</math> (the area of the flat region) are all mapped to the value <math>A</math>.</p>
<p><b>Bias: Parameter Dependent</b> (<math>A \in (0, 1), 0 &lt; B &lt; C</math>)</p> $\mathbf{b\_param}(y, \mathbf{y'}, A, B, C) = y^{B+(C-B)v(u(\mathbf{y'}))}$ $v(u(\mathbf{y'})) = A - (1 - 2u(\mathbf{y'})) \lfloor 0.5 - u(\mathbf{y'}) \rfloor + A$ <p><math>A, B, C</math>, and the secondary parameter vector <math>\mathbf{y'}</math> together determine the degree to which <math>y</math> is biased by being raised to an associated power: values of <math>u(\mathbf{y'}) \in [0, 0.5]</math> are mapped linearly onto <math>[B, B + (C - B)A]</math>, and values of <math>u(\mathbf{y'}) \in [0.5, 1]</math> are mapped linearly onto <math>[B + (C - B)A, C]</math>.</p>
<p><b>Shift: Linear</b> (<math>A \in (0, 1)</math>)</p> $\mathbf{s\_linear}(y, A) = \frac{ y-A }{\lfloor \lfloor A-y \rfloor + A \rfloor}$ <p><math>A</math> is the value for which <math>y</math> is mapped to zero.</p>
<p><b>Shift: Deceptive</b> (<math>A \in (0, 1), 0 &lt; B \ll 1, 0 &lt; C \ll 1, A - B &gt; 0, A + B &lt; 1</math>)</p> $\mathbf{s\_decept}(y, A, B, C) = 1 + ( y - A  - B) \times \left( \frac{\lfloor y - A + B \rfloor (1 - C + \frac{A-B}{B})}{A-B} + \frac{\lfloor A + B - y \rfloor (1 - C + \frac{1-A-B}{B})}{1-A-B} + \frac{1}{B} \right)$ <p><math>A</math> is the value at which <math>y</math> is mapped to zero, and the global minimum of the transformation. <math>B</math> is the “aperture” size of the well/basin leading to the global minimum at <math>A</math>, and <math>C</math> is the value of the deceptive minima (there are always two deceptive minima).</p>
<p><b>Shift: Multi-modal</b> (<math>A \in \{1, 2, \dots\}, B \geq 0, (4A + 2)\pi \geq 4B, C \in (0, 1)</math>)</p> $\mathbf{s\_multi}(y, A, B, C) = \frac{1 + \cos \left[ (4A + 2)\pi \left( 0.5 - \frac{ y-C }{2(\lfloor C-y \rfloor + C)} \right) \right]}{B+2} + 4B \left( \frac{ y-C }{2(\lfloor C-y \rfloor + C)} \right)^2$ <p><math>A</math> controls the number of minima, <math>B</math> controls the magnitude of the “hill sizes” of the multi-modality, and <math>C</math> is the value for which <math>y</math> is mapped to zero. When <math>B = 0</math>, <math>2A + 1</math> values of <math>y</math> (one at <math>C</math>) are mapped to zero, and when <math>B \neq 0</math>, there are <math>2A</math> local minima, and one global minimum at <math>C</math>. Larger values of <math>A</math> and smaller values of <math>B</math> create more difficult problems.</p>
<p><b>Reduction: Weighted Sum</b> (<math> \mathbf{w}  =  \mathbf{y} , w_1, \dots, w_{ \mathbf{y} } &gt; 0</math>)</p> $\mathbf{r\_sum}(\mathbf{y}, \mathbf{w}) = \left( \sum_{i=1}^{ \mathbf{y} } w_i y_i \right) / \sum_{i=1}^{ \mathbf{y} } w_i$ <p>By varying the constants of the weight vector <math>\mathbf{w}</math>, EAs can be forced to treat parameters differently.</p>
<p><b>Reduction: Non-separable</b> (<math>A \in \{1, \dots,  \mathbf{y} \},  \mathbf{y}  \bmod A = 0</math>)</p> $\mathbf{r\_nonsep}(\mathbf{y}, A) = \frac{\sum_{j=1}^{ \mathbf{y} } (y_j + \sum_{k=0}^{A-2}  y_j - y_{1+(j+k) \bmod  \mathbf{y} } )}{\frac{ \mathbf{y} }{A} \lceil A/2 \rceil (1+2A-2\lceil A/2 \rceil)}$ <p><math>A</math> controls the degree of non-separability (noting that <math>\mathbf{r\_nonsep}(\mathbf{y}, 1) = \mathbf{r\_sum}(\mathbf{y}, \{1, \dots, 1\})</math>).</p>

to at least one shift transformation. By incorporating secondary parameters via a reduction transformation,  $\mathbf{b\_param}$  can create dependencies between distinct parameters, including position- and distance-related parameters. Moreover, when employed before any shift transformation,  $\mathbf{b\_param}$  can create objectives that are effectively nonseparable—a separable optimization approach would fail unless given multiple iterations, or a specific order of parameters to optimize. The deceptive and multimodal shift transformations make the corresponding problem deceptive and multimodal, respectively. The flat region transformation can have a significant impact on the fitness landscape and can also be used to create a stark many-to-one mapping from the Pareto optimal front to the Pareto optimal set.

Using these shape functions and transformations, we now show how to construct a well-designed, scalable test problem, that is both separable and multimodal (originally described in [43]).

#### A. Building an Example Test Problem

Creating problems with the WFG Toolkit involves three main steps: specifying values for the underlying formalism (including scaling constants and parameter domains), specifying the shape functions, and specifying transition vectors. To aid in construction, a computer-aided design tool or metalanguage could be used to help select and connect together the different components making up the test problem. With the use of sensible de-

TABLE XII  
TRANSFORMATION FUNCTION RESTRICTIONS

Restriction	Comment
Constants	Must be fixed (not tied to the value of any parameters).
Primary parameters	For any given transition vector, all parameters of the originating transition vector must be employed exactly once as a primary parameter (counting parameters that appear independently as primary parameters), and in the same order in which they appear in the originating transition vector.
Secondary parameters	Care must be taken to avoid cyclical dependencies in <i>b_param</i> . Consider the following terminology: if <i>a</i> is a primary parameter of <i>b_param</i> , and <i>b</i> is one of the secondary parameters, then we say that <i>a</i> depends on <i>b</i> . If <i>b</i> likewise depends on <i>c</i> , then <i>a</i> also (indirectly) depends on <i>c</i> . When <i>a</i> depends on some parameter <i>b</i> , then there is an associated dependency between the corresponding working parameters. To prevent cyclical dependencies, no two working parameters should be dependent on one another. In addition, a parameter should not depend on itself.
Shifts	Parameters should not be subjected to more than one shift transformation.
Reductions	Reduction transformations should belong to transition vectors that are closer to the underlying parameter vector than any shift transformation.
<i>b_flat</i>	When <i>A</i> = 0, <i>b_flat</i> should only belong to transition vectors that are further away from the underlying parameter vector than any shift or reduction transformation.

fault values, the test problem designer then need only specify which features of interest they desire in the test problem. An example scalable test problem is specified in Table XIII and expanded in Fig. 11.

This example problem is scalable both objective- and parameter-wise, where the number of distance- and position-related parameters can be scaled independently. For a solution to be Pareto optimal, it is required that all of

$$z_{i=k+1:n} = 2i \times \begin{cases} 0.35^{(0.02+1.96u)^{-1}}, & i \neq n \\ 0.35, & i = n \end{cases}$$

$$u = \text{r.sum}(\{z_{i+1}, \dots, z_n\}, \{1, \dots, 1\})$$

which can be found by first determining  $z_n$ , then  $z_{n-1}$ , and so on, until the required value for  $z_{k+1}$  is determined. Once the optimal values for  $z_{k+1:n}$  are determined, the position-related parameters can be varied arbitrarily to obtain different Pareto optimal solutions.

The example problem has a distinct many-to-one mapping from the Pareto optimal set to the Pareto optimal front due to the deceptive transformation of the position-related parameters. All objectives are nonseparable, deceptive, and multimodal, the latter with respect to the distance component. The problem is also biased in a parameter dependent manner.

This example constitutes a well-designed scalable problem that is both nonseparable and multimodal—we are not aware of any problem in the literature with comparable characteristics.

#### B. An Example Test Suite: WFG1-WFG9

In [43], we used the toolkit to construct a suite of test problems WFG1–WFG9 that satisfies all the requirements of Section V, including well-designed problems with a variety of characteristics, providing a thorough test for MOEAs in our target class. This suite is displayed in Table XIV. Note that WFG9 is the problem constructed in Section VIII-A.

We make the following additional observations: WFG1 skews the relative significance of different parameters by employing

dissimilar weights in its weighted sum reduction, only WFG1 and WFG7 are both separable and unimodal, the nonseparable reduction of WFG6 and WFG9 is more difficult than that of WFG2 and WFG3, the multimodality of WFG4 has larger “hill sizes” (and is thus more difficult) than that of WFG9, the deceptiveness of WFG5 is more difficult than that of WFG9 (WFG9 is only deceptive on its position parameters), the position-related parameters of WFG7 are dependent on its distance-related parameters (and other position-related parameters)—WFG9 employs a similar type of dependency, but distance-related parameters also depend on other distance-related parameters, the distance-related parameters of WFG8 are dependent on its position-related parameters (and other distance-related parameters) and as a consequence the problem is nonseparable. The predominance of concave Pareto optimal fronts facilitates the use of performance measures that require knowledge of the distance to the Pareto optimal front.

For WFG1-WFG7, a solution is Pareto optimal iff all  $z_{i=k+1:n} = 2i \times 0.35$ , noting that WFG2 is disconnected. For WFG8, it is required that all of

$$z_{i=k+1:n} = 2i \times 0.35^{(0.02+49.98(\frac{0.98}{49.98}-(1-2u)|[0.5-u]+\frac{0.98}{49.98}))^{-1}}$$

$$u = \text{r.sum}(\{z_1, \dots, z_{i-1}\}, \{1, \dots, 1\}).$$

To obtain a Pareto optimal solution, the position should first be determined by setting  $z_{1:k}$  appropriately. The required distance-related parameter values can then be calculated by first determining  $z_{k+1}$  (which is trivial given  $z_{1:k}$  have been set), then  $z_{k+2}$ , and so on, until  $z_n$  has been calculated. Unlike the other WFG problems, different Pareto optimal solutions will have different distance-related parameter values, making WFG8 a difficult problem. Optimality conditions for WFG9 are given in Section VIII-A.

The WFG test suite exceeds the functionality of previous existing test suites. In particular, it includes a number of problems that exhibit properties not evident in the commonly-used

TABLE XIII

AN EXAMPLE TEST PROBLEM. THE NUMBER OF POSITION-RELATED PARAMETERS,  $k$ , MUST BE DIVISIBLE BY THE NUMBER OF UNDERLYING POSITION PARAMETERS,  $M-1$  (THIS SIMPLIFIES  $\mathbf{t}^3$ ). THE NUMBER OF DISTANCE-RELATED PARAMETERS,  $l$ , CAN BE SET TO ANY POSITIVE INTEGER. TO ENHANCE READABILITY, FOR ANY TRANSITION VECTOR  $\mathbf{t}^i$ , WE LET  $\mathbf{y} = \mathbf{t}^{i-1}$ . FOR  $\mathbf{t}^1$ , LET  $\mathbf{y} = \mathbf{z}_{[0,1]} = \{z_1/2, \dots, z_n/(2n)\}$

Type	Setting
Constants	$S_{m=1:M} = 2m$ $A_{1:M-1} = 1$ The settings for $S_{1:M}$ ensures the Pareto optimal front will have dissimilar tradeoff magnitudes, and the settings for $A_{1:M-1}$ ensures the Pareto optimal front is not degenerate.
Domains	$z_{i=1:n,\max} = 2i$ The working parameters have domains of dissimilar magnitude.
Shape	$h_{m=1:M} = \text{concave}_m$ The purely concave Pareto optimal front facilitates the use of some performance metrics, where the distance of a solution to the nearest point on the Pareto optimal front must be determined.
$\mathbf{t}^1$	$t_{i=1:n-1}^1 = \text{b\_param}(y_i, \text{r\_sum}(\{y_{i+1}, \dots, y_n\}, \{1, \dots, 1\}), \frac{0.98}{49.98}, 0.02, 50)$ $t_n^1 = y_n$ By employing the parameter dependent bias transformation, this transition vector ensures that distance- and position-related working parameters are inter-dependent and somewhat non-separable.
$\mathbf{t}^2$	$t_{i=1:k}^2 = \text{s\_decept}(y_i, 0.35, 0.001, 0.05)$ $t_{i=k+1:n}^2 = \text{s\_multi}(y_i, 30, 95, 0.35)$ This transition vector makes some parts of the Pareto optimal front more difficult to determine (due to the deceptive transformation), and also makes it more difficult to converge to the Pareto optimal front (due to the multi-modal transformation). The multi-modality is similar to Rastrigin's function, with many local optima ( $61^l - 1$ ), and one global optimum, where the "hill size" between adjacent local optima is relatively small.
$\mathbf{t}^3$	$t_{i=1:M-1}^3 = \text{r\_nonsep}(\{y_{(i-1)k/(M-1)}, \dots, y_{ik/(M-1)}\}, k/(M-1))$ $t_M^3 = \text{r\_nonsep}(\{y_{k+1}, \dots, y_n\}, l)$ This transition vector ensures that all objectives are non-separable, and also reduces the number of parameters down to $M$ , as required by the framework.

$$\begin{aligned}
 \text{Given } \mathbf{z} &= \{z_1, \dots, z_k, z_{k+1}, \dots, z_n\} \\
 \text{Minimise } f_1(\mathbf{x}) &= x_M + 2 \prod_{i=1}^{M-1} \sin(x_i \pi / 2) \\
 f_{m=2:M-1}(\mathbf{x}) &= x_M + 2m \left( \prod_{i=1}^{M-m} \sin(x_i \pi / 2) \right) \cos(x_{M-m+1} \pi / 2) \\
 f_M(\mathbf{x}) &= x_M + 2M \cos(x_1 \pi / 2) \\
 \text{where } x_{i=1:M-1} &= \text{r\_nonsep}(\{y_{(i-1)k/(M-1)+1}, \dots, y_{ik/(M-1)}\}, k/(M-1)) \\
 x_M &= \text{r\_nonsep}(\{y_{k+1}, \dots, y_n\}, l) \\
 y_{i=1:k} &= \text{s\_decept}(y'_i, 0.35, 0.001, 0.05) \\
 y_{i=k+1:n} &= \text{s\_multi}(y'_i, 30, 95, 0.35) \\
 y'_{i=1:n-1} &= \text{b\_param}\left(z_i/(2i), \sum_{j=i+1}^n z_j/(2j(n-i)), 0.98/49.98, 0.02, 50\right) \\
 y'_n &= z_n/(2n)
 \end{aligned}$$

Fig. 11. The expanded form of the problem defined in Table XIII.  $|\mathbf{z}| = n = k + l$ ,  $k \in \{M-1, 2(M-1), 3(M-1), \dots\}$ ,  $l \in \{1, 2, \dots\}$ , and the domain of all  $z_i \in \mathbf{z}$  is  $[0, 2i]$ .

DTLZ test suite. These include: nonseparable problems, deceptive problems, a truly degenerate problem, a mixed shape Pareto front problem, problems scalable in the number of position-related parameters,<sup>5</sup> and problems with dependencies between position- and distance-related parameters. The WFG test suite provides a truer means of assessing the performance of optimization algorithms on a wide range of different problems.

<sup>5</sup>The DTLZ test suite uses a fixed (relative to the number of objectives) number of position parameters.

## IX. ILLUSTRATION OF USE OF THE WFG TOOLKIT

In this section, we present some experiments comparing the performance of a well known MOEA, NSGA-II [44], on the WFG test suite, and on the DTLZ test suite. The results demonstrate that the WFG suite presents a more comprehensive set of challenges.

Note that we chose to use four position related parameters and 20 distance related parameters (a total of 24) for each WFG

TABLE XIV

THE WFG TEST SUITE. THE NUMBER OF POSITION-RELATED PARAMETERS  $k$  MUST BE DIVISIBLE BY THE NUMBER OF UNDERLYING POSITION PARAMETERS,  $M-1$  (THIS SIMPLIFIES REDUCTIONS). THE NUMBER OF DISTANCE-RELATED PARAMETERS  $l$  CAN BE SET TO ANY POSITIVE INTEGER, EXCEPT FOR WFG2 AND WFG3, FOR WHICH  $l$  MUST BE A MULTIPLE OF TWO (DUE TO THE NATURE OF THEIR NONSEPARABLE REDUCTIONS). TO ENHANCE READABILITY, FOR ANY TRANSITION VECTOR  $\mathbf{t}^i$ , WE LET  $\mathbf{y} = \mathbf{t}^{i-1}$ . FOR  $\mathbf{t}^1$ , LET  $\mathbf{y} = \mathbf{z}_{[0,1]} = \{z_1/2, \dots, z_n/(2n)\}$

Problem	Type	Setting
All	Constants	$S_{m=1:M} = 2m$ $D = 1$ $A_1 = 1$ $A_{2:M-1} = \begin{cases} 0, & \text{for WFG3} \\ 1, & \text{otherwise} \end{cases}$ The settings for $S_{1:M}$ ensures the Pareto optimal fronts have dissimilar tradeoff magnitudes, and the settings for $A_{1:M-1}$ ensures the Pareto optimal fronts are not degenerate, except in the case of WFG3, which has a one dimensional Pareto optimal front.
All	Domains	$z_{i=1:n, \max} = 2i$ The working parameters have domains of dissimilar magnitude.
WFG1	Shape $\mathbf{t}^1$ $\mathbf{t}^2$ $\mathbf{t}^3$ $\mathbf{t}^4$	$h_{m=1:M-1} = \text{convex}_m$ $h_M = \text{mixed}_M$ (with $\alpha = 1$ and $A = 5$ ) $t_{i=1:k}^1 = y_i$ $t_{i=k+1:n}^1 = \text{s\_linear}(y_i, 0.35)$ $t_{i=1:k}^2 = y_i$ $t_{i=k+1:n}^2 = \text{b\_flat}(y_i, 0.8, 0.75, 0.85)$ $t_{i=1:n}^3 = \text{b\_poly}(y_i, 0.02)$ $t_{i=1:M-1}^4 = \text{r\_sum}(\{y_{(i-1)k/(M-1)+1}, \dots, y_{ik/(M-1)}\}, \{2((i-1)k/(M-1)+1), \dots, 2ik/(M-1)\})$ $t_M^4 = \text{r\_sum}(\{y_{k+1}, \dots, y_n\}, \{2(k+1), \dots, 2n\})$
WFG2	Shape $\mathbf{t}^1$ $\mathbf{t}^2$ $\mathbf{t}^3$	$h_{m=1:M-1} = \text{convex}_m$ $h_M = \text{disc}_M$ (with $\alpha = \beta = 1$ and $A = 5$ ) As $\mathbf{t}^1$ from WFG1. (Linear shift.) $t_{i=1:k}^2 = y_i$ $t_{i=k+1:k+l/2}^2 = \text{r\_nonsep}(\{y_{k+2(i-k)-1}, y_{k+2(i-k)}\}, 2)$ $t_{i=1:M-1}^3 = \text{r\_sum}(\{y_{(i-1)k/(M-1)+1}, \dots, y_{ik/(M-1)}\}, \{1, \dots, 1\})$ $t_M^3 = \text{r\_sum}(\{y_{k+1}, \dots, y_{k+l/2}\}, \{1, \dots, 1\})$
WFG3	Shape $\mathbf{t}^{1:3}$	$h_{m=1:M} = \text{linear}_m$ (degenerate) As $\mathbf{t}^{1:3}$ from WFG2. (Linear shift, non-separable reduction, and weighted sum reduction.)
WFG4	Shape $\mathbf{t}^1$ $\mathbf{t}^2$	$h_{m=1:M} = \text{concave}_m$ $t_{i=1:n}^1 = \text{s\_multi}(y_i, 30, 10, 0.35)$ $t_{i=1:M-1}^2 = \text{r\_sum}(\{y_{(i-1)k/(M-1)+1}, \dots, y_{ik/(M-1)}\}, \{1, \dots, 1\})$ $t_M^2 = \text{r\_sum}(\{y_{k+1}, \dots, y_n\}, \{1, \dots, 1\})$
WFG5	Shape $\mathbf{t}^1$ $\mathbf{t}^2$	$h_{m=1:M} = \text{concave}_m$ $t_{i=1:n}^1 = \text{s\_decept}(y_i, 0.35, 0.001, 0.05)$ As $\mathbf{t}^2$ from WFG4. (Weighted sum reduction.)
WFG6	Shape $\mathbf{t}^1$ $\mathbf{t}^2$	$h_{m=1:M} = \text{concave}_m$ As $\mathbf{t}^1$ from WFG1. (Linear shift.) $t_{i=1:M-1}^2 = \text{r\_nonsep}(\{y_{(i-1)k/(M-1)+1}, \dots, y_{ik/(M-1)}\}, k/(M-1))$ $t_M^2 = \text{r\_nonsep}(\{y_{k+1}, \dots, y_n\}, l)$
WFG7	Shape $\mathbf{t}^1$ $\mathbf{t}^2$ $\mathbf{t}^3$	$h_{m=1:M} = \text{concave}_m$ $t_{i=1:k}^1 = \text{b\_param}(y_i, \text{r\_sum}(\{y_{i+1}, \dots, y_n\}, \{1, \dots, 1\}), \frac{0.98}{49.98}, 0.02, 50)$ $t_{i=k+1:n}^1 = y_i$ As $\mathbf{t}^1$ from WFG1. (Linear shift.) As $\mathbf{t}^2$ from WFG4. (Weighted sum reduction.)
WFG8	Shape $\mathbf{t}^1$ $\mathbf{t}^2$ $\mathbf{t}^3$	$h_{m=1:M} = \text{concave}_m$ $t_{i=1:k}^1 = y_i$ $t_{i=k+1:n}^1 = \text{b\_param}(y_i, \text{r\_sum}(\{y_1, \dots, y_{i-1}\}, \{1, \dots, 1\}), \frac{0.98}{49.98}, 0.02, 50)$ As $\mathbf{t}^1$ from WFG1. (Linear shift.) As $\mathbf{t}^2$ from WFG4. (Weighted sum reduction.)
WFG9	Shape $\mathbf{t}^1$ $\mathbf{t}^2$ $\mathbf{t}^3$	$h_{m=1:M} = \text{concave}_m$ $t_{i=1:n-1}^1 = \text{b\_param}(y_i, \text{r\_sum}(\{y_{i+1}, \dots, y_n\}, \{1, \dots, 1\}), \frac{0.98}{49.98}, 0.02, 50)$ $t_n^1 = y_n$ $t_{i=1:k}^2 = \text{s\_decept}(y_i, 0.35, 0.001, 0.05)$ $t_{i=k+1:n}^2 = \text{s\_multi}(y_i, 30, 95, 0.35)$ As $\mathbf{t}^2$ from WFG6. (Non-separable reduction.)

problem, and one position parameter and 23 distance parameters (also a total of 24) for DTLZ. In order to facilitate analysis, we use only two objectives. NSGA-II was run with real-coded parameters, a mutation probability of  $1/24$ , a crossover probability of 0.9, a crossover distribution index of 10.0, and a mutation distribution index of 50.0.

The procedure we used is as follows: for each problem in the WFG test suite and the DTLZ test suite, we executed 35 runs of NSGA-II, with a population size of 100, for 25 000 generations. We saved the nondominated front of the populations in each case after 250, 2500, and 25 000 generations. In the literature, a population size of 100 for 250 generations is a common

TABLE XV

PROPERTIES OF THE WFG PROBLEMS. ALL WFG PROBLEMS ARE SCALABLE, HAVE NO EXTREMAL NOR MEDIAL PARAMETERS, HAVE DISSIMILAR PARAMETER DOMAINS AND PARETO OPTIMAL TRADEOFF MAGNITUDES, HAVE KNOWN PARETO OPTIMAL SETS, AND CAN BE MADE TO HAVE A DISTINCT MANY-TO-ONE MAPPING FROM THE PARETO OPTIMAL SET TO THE PARETO OPTIMAL FRONT BY SCALING THE NUMBER OF POSITION PARAMETERS

Problem	Obj.	Separability	Modality	Bias	Geometry
WFG1	$f_{1:M}$	S	U	polynomial, flat	convex, mixed
WFG2	$f_{1:M-1}$	NS	U	—	convex, disconnected
	$f_M$	NS	M		
WFG3	$f_{1:M}$	NS	U	—	linear, degenerate
WFG4	$f_{1:M}$	S	M	—	concave
WFG5	$f_{1:M}$	S	D	—	concave
WFG6	$f_{1:M}$	NS	U	—	concave
WFG7	$f_{1:M}$	S	U	parameter dependent	concave
WFG8	$f_{1:M}$	NS	U	parameter dependent	concave
WFG9	$f_{1:M}$	NS	M,D	parameter dependent	concave

choice, so 25 000 generations should be more than ample to ensure convergence.

For each problem, from the 35 fronts at 250 generations, we computed the 50% attainment surface. We did this also for the 35 fronts at generation 2500 and at generation 25 000. In Figs. 12. and 13, for each problem, we have plotted the Pareto optimal front, plus the 50% attainment surface after 250, 2500, and 25 000 generations. Note that some of the plots feature a magnified view of part of the front, to aid interpretation.

Visually, it seems clear that NSGA-II has solved some of the problems easily, quickly converging on the Pareto optimal front. We would put WFG2, WFG3, WFG4, and WFG7 into this category, along with all the DTLZ problems except DTLZ6. Although convergence of DTLZ1 and DTLZ3 appears qualitatively different from that on the other DTLZ problems, this is explained by the large values of the  $g$  function for these problems, which has the effect of stretching out the distances between successive fronts (we can achieve the same effect with the toolkit by using a large value of  $D$ ). DTLZ6 is the only one of the DTLZ problems where NSGA-II appears to fail to converge [see Fig. 13(f)]. We hypothesise that this may be due to strong distance-dependent bias.

In contrast, five of the nine WFG problems posed considerable difficulties for NSGA-II. Note that we do not claim, from this, that NSGA-II is a poor optimizer—the WFG problems are challenging, and we expect that other optimizers would experience similar difficulties, perhaps moreso. However, analysis of the causes of these difficulties should lead to improved algorithms for problems having similar characteristics.

On WFG1, for example, NSGA-II shows poor coverage of the front at 250 generations. By 25 000 generations, coverage has improved, but convergence is still poor. This behavior might be explained in terms of the algorithm having difficulty coping with bias. WFG8 and WFG9 also feature significant bias. For WFG8, distance related parameters are dependent on position related parameters, meaning that an optimizer cannot simply find a good set of distance parameters, and then use that set to spread out along the front. Crossover is much less likely to yield good results. WFG9 has position related param-

eters dependent on distance related parameters. This type of dependency is not as difficult as that seen in WFG8, however, WFG9 is also multimodal, and has a troublesome kind of non-separable reduction. This kind of reduction, with all distance related parameters being confounded, is also seen with WFG6. That WFG5 causes difficulties is to be expected: WFG5 is highly deceptive.

A tentative research question that arises from these observations: how common is bias of different types in real-world problems, and how could an MOEA be modified to cope with these types of bias? (It would be interesting to see how an Estimation of Distribution Algorithm would fare, for example.) Likewise, how common are different kinds of nonseparability, and how could an MOEA be modified to cope with them?

## X. CONCLUSION

Test problems of the type reviewed in this paper are regularly employed to support the testing and comparison of EAs, both new and old. Unfortunately, the characteristics of many test problems were previously not well understood, and whilst this paper helps to address this lack of understanding, additional issues have been identified. For example, the suitability of many of the reviewed test problems, at least in the context of general algorithm testing, is cast into doubt in view of our recommendations. More significantly, although this paper has also shown that there are a number of well designed test problems, the important class of nonseparable problems, particularly in combination with multimodal problems, is poorly represented.

While awareness of the limitations of commonly used test problems should, of itself, contribute to improved practice in evaluating MOEAs, we have also offered a practical solution in the form of the WFG Toolkit, a flexible toolkit for creating multiobjective real valued, unconstrained problems. We showed how the toolkit may be used to construct a test suite that includes problems with characteristics largely missing from existing test suites. We demonstrated this with a set of experiments comparing the performance of an MOEA on this test suite, to its performance on another commonly used test



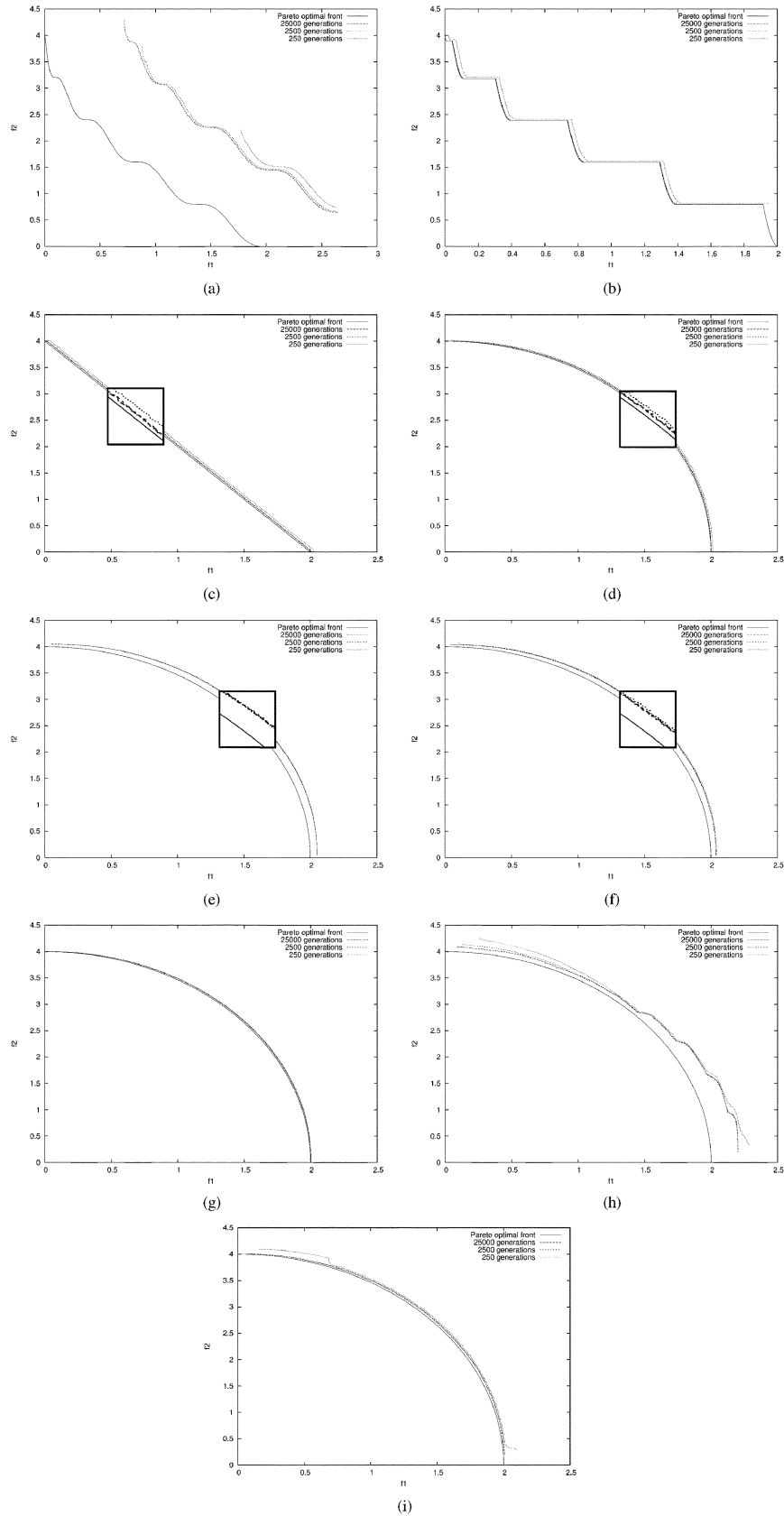


Fig. 12. Pareto optimal front and 50% attainment surfaces for NSGA-II after 250, 2500, and 25 000 generations on the WFG test suite problems. In (c), (d), (e) and (f), a portion of the fronts has been magnified for clarity. (a) WFG1. (b) WFG2. (c) WFG3. (d) WFG4. (e) WFG5. (f) WFG6. (g) WFG7. (h) WFG8. (i) WFG9.

suite. Considering that specialised and enhanced versions of MOEAs (as well as completely new ones) will continue to be

developed, and the range of multiobjective problems that are attempted will continue to expand, merely creating a number

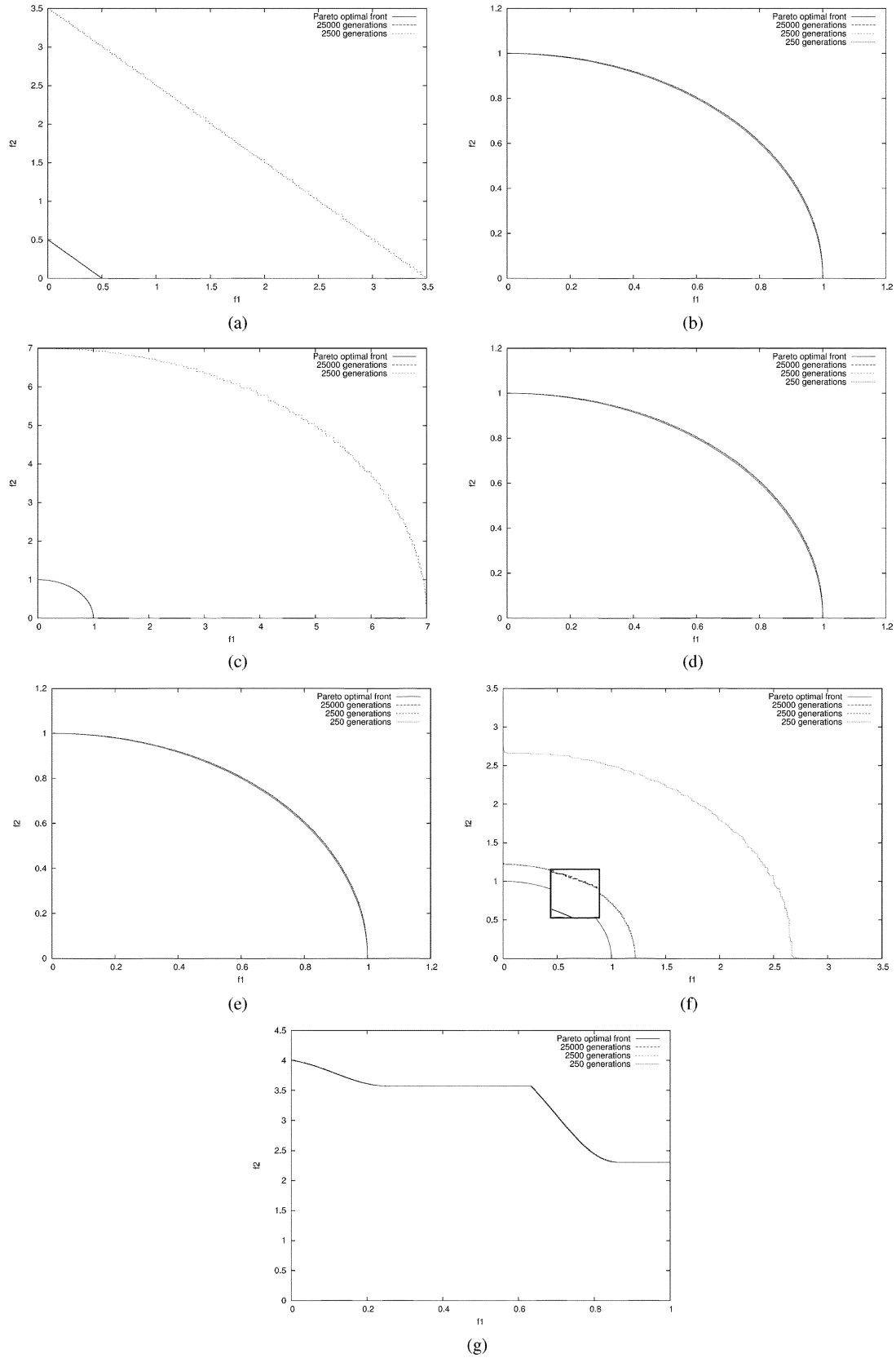


Fig. 13. Pareto optimal front and 50% attainment surfaces for NSGA-II after 250, 2500, and 25 000 generations on the DTLZ test suite problems. In (a) and (c), the 250 generations attainment surface has been omitted as it is outside the range of the plot. In (f), a portion of the fronts has been magnified for clarity. (a) DTLZ1. (b) DTLZ2. (c) DTLZ3. (d) DTLZ4. (e) DTLZ5. (f) DTLZ6. (g) DTLZ7.

of benchmark problems would be self-limiting. Instead, the toolkit lets us include exactly the features we want in a test

problem to suit the situation, whilst simultaneously conforming to our recommendations.

TABLE XVI  
OTHER MULTI-OBJECTIVE TEST PROBLEMS. UNLESS OTHERWISE STATED, ALL OBJECTIVES ARE TO BE MINIMIZED

Name	Reference	Problem	Parameter Domains
BK1	[45]	$f_1(x_1, x_2) = x_1^2 + x_2^2$ $f_2(x_1, x_2) = (x_1 - 5)^2 + (x_2 - 5)^2$	$[-5, 10]$
DPAM1	[13]	$f_1(\mathbf{y}) = y_1$ $f_2(\mathbf{y}) = g \exp(-y_1/g)$ $g = 1 + 10(n-1) + \sum_{i=2}^n [y_i^2 - 10 \cos(4\pi y_i)]$ $\mathbf{y} = \mathfrak{R}\mathbf{x}$ where $\mathfrak{R}$ is a fixed rotation matrix, $\mathbf{y}$ is an intermediate parameter vector, and $\mathbf{x}$ is the working parameter vector on which algorithms operate.	$[-0.3, 0.3]$
DGO1	[46]	$f_1(x) = \sin(x)$ $f_2(x) = \sin(x + 0.7)$	$[-10, 13]$
DGO2	[46]	$f_1(x) = x^2$ $f_2(x) = 9 - \sqrt{81 - x^2}$	$[-9, 9]$
FA1	[47]	$f_1(x_1) = (1 - \exp(-4x_1))/(1 - \exp(-4))$ $f_2(x_1, x_2, x_3) = (x_2 + 1)(1 - (f_1(x_1)/(x_2 + 1))^{0.5})$ $f_3(x_1, x_2, x_3) = (x_3 + 1)(1 - (f_1(x_1)/(x_3 + 1))^{0.1})$	$[0, 1]$
Far1	[48]	$f_1(x_1, x_2) = -2 \exp(15(-(x_1 - 0.1)^2 - x_2^2))$ $- \exp(20(-(x_1 - 0.6)^2 - (x_2 - 0.6)^2))$ $+ \exp(20(-(x_1 + 0.6)^2 - (x_2 - 0.6)^2))$ $+ \exp(20(-(x_1 - 0.6)^2 - (x_2 + 0.6)^2))$ $+ \exp(20(-(x_1 + 0.6)^2 - (x_2 + 0.6)^2))$ $f_2(x_1, x_2) = 2 \exp(20(-x_1^2 - x_2^2))$ $+ \exp(20(-(x_1 - 0.4)^2 - (x_2 - 0.6)^2))$ $- \exp(20(-(x_1 + 0.5)^2 - (x_2 - 0.7)^2))$ $- \exp(20(-(x_1 - 0.5)^2 - (x_2 + 0.7)^2))$ $+ \exp(20(-(x_1 + 0.4)^2 - (x_2 + 0.8)^2))$	$[-1, 1]$
FES1	[49]	$f_1(x_1, \dots, x_n) = \sum_{i=1}^n  x_i - \exp((i/n)^2)/3 ^{0.5}$ $f_2(x_1, \dots, x_n) = \sum_{i=1}^n (x_i - 0.5 \cos(10\pi i/n) - 0.5)^2$	$[0, 1]$
FES2	[49]	$f_1(x_1, \dots, x_n) = \sum_{i=1}^n (x_i - 0.5 \cos(10\pi i/n) - 0.5)^2$ $f_2(x_1, \dots, x_n) = \sum_{i=1}^n  x_i - \sin^2(i-1) \cos^2(i-1) ^{0.5}$ $f_3(x_1, \dots, x_n) = \sum_{i=1}^n  x_i - 0.25 \cos(i-1) \cos(2i-2) - 0.5 ^{0.5}$	$[0, 1]$
FES3	[49]	$f_1(x_1, \dots, x_n) = \sum_{i=1}^n  x_i - \exp((i/n)^2)/3 ^{0.5}$ $f_2(x_1, \dots, x_n) = \sum_{i=1}^n  x_i - \sin^2(i-1) \cos^2(i-1) ^{0.5}$ $f_3(x_1, \dots, x_n) = \sum_{i=1}^n  x_i - 0.25 \cos(i-1) \cos(2i-2) - 0.5 ^{0.5}$ $f_4(x_1, \dots, x_n) = \sum_{i=1}^n (x_i - 0.5 \sin(1000\pi i/n) - 0.5)^2$	$[0, 1]$
FF1	[50]	$f_1(x_1, x_2) = 1 - \exp(-(x_1 - 1)^2 - (x_2 + 1)^2)$ $f_2(x_1, x_2) = 1 - \exp(-(x_1 + 1)^2 - (x_2 - 1)^2)$	?
IKK1	[51]	$f_1(x_1) = x_1^4$ $f_2(x_1) = (x_1 - 20)^2$ $f_3(x_2) = x_2^2$	$[-50, 50]$
IM1	[52]	$f_1(x_1) = 2\sqrt{x_1}$ $f_2(x_1, x_2) = x_1(1 - x_2) + 5$	$x_1 \in [1, 4]$ $x_2 \in [1, 2]$
JOS1	[53]	$f_1(x_1, \dots, x_n) = \sum_{i=1}^n x_i^2/n$ $f_2(x_1, \dots, x_n) = \sum_{i=1}^n (x_i - 2)^2/n$	?
JOS2	[53]	$f_1(x_1) = x_1$ $f_2(x_1, \dots, x_n) = g(1 - (f_1(x_1)/g)^{0.25} - (f_1(x_1)/g)^4)$ $g = 1 + 9 \sum_{i=2}^n x_i/(n-1)$	$[0, 1]$
Kur1	[37]	$f_1(x_1, \dots, x_n) = \sum_{i=1}^{n-1} -10 \exp^{-0.2\sqrt{x_i^2 + x_{i+1}^2}}$ $f_2(x_1, \dots, x_n) = \sum_{i=1}^{n-1}  x_i ^{0.8} + 5 \sin^8 x_i$	?
LRS1	[54]	$f_1(x_1, x_2) = x_1^4 + x_2^2$ $f_2(x_1, x_2) = (x_1 + 2)^2 + x_2^2$	$[-50, 50]$
LTDZ1	[55]	Max. $f_1(x_1, x_2, x_3) = 3 - (1 + x_3) \cos(x_1\pi/2) \cos(x_2\pi/2)$ Max. $f_2(x_1, x_2, x_3) = 3 - (1 + x_3) \cos(x_1\pi/2) \sin(x_2\pi/2)$ Max. $f_3(x_1, x_2, x_3) = 3 - (1 + x_3) \sin(x_1\pi/2)$	$[0, 1]$
LE1	[56]	$f_1(x_1, x_2) = (x_1^2 + x_2^2)^{0.125}$ $f_2(x_1, x_2) = ((x_1 - 0.5)^2 + (x_2 - 0.5)^2)^{0.25}$	$[-5, 10]$
MHHM1	[57]	$f_1(x) = (x - 0.8)^2$ $f_2(x) = (x - 0.85)^2$ $f_3(x) = (x - 0.9)^2$	$[0, 1]$
MHHM2	[57]	$f_1(x_1, x_2) = (x_1 - 0.8)^2 + (x_2 - 0.6)^2$ $f_2(x_1, x_2) = (x_1 - 0.85)^2 + (x_2 - 0.7)^2$ $f_3(x_1, x_2) = (x_1 - 0.9)^2 + (x_2 - 0.6)^2$	$[0, 1]$
MLF1	[58]	$f_1(x) = (1 + x/20) \sin x$ $f_2(x) = (1 + x/20) \cos x$	$[0, 20]$
MLF2	[58]	Max. $f_1(x_1, x_2) = 5 - \frac{(x_1^2 + x_2^2 - 11)^2 + (x_1 + x_2^2 - 7)^2}{200}$ Max. $f_2(x_1, x_2) = f_1(2x_1, 2x_2)$	?
QV1	[59]	$f_1(x_1, \dots, x_n) = \left[ \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10) / n \right]^{0.25}$ $f_2(x_1, \dots, x_n) = \left[ \sum_{i=1}^n ((x_i - 1.5)^2 - 10 \cos(2\pi(x_i - 1.5)) + 10) / n \right]^{0.25}$	$[-5.12, 5.12]$
Sch1	[34], [39]	$f_1(x) = \begin{cases} -x, & x \leq 1 \\ -2 + x, & 1 < x \leq 3 \\ 4 - x, & 3 < x \leq 4 \\ -4 + x, & 4 < x \end{cases}$ $f_2(x) = (x - 5)^2$	?
SP1	[60]	$f_1(x_1, x_2) = (x_1 - 1)^2 + (x_1 - x_2)^2$ $f_2(x_1, x_2) = (x_2 - 3)^2 + (x_1 - x_2)^2$	?
SSFYY1	[61]	$f_1(x_1, x_2) = x_1^2 + x_2^2$ $f_2(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 2)^2$	$[-100, 100]$
SSFYY2	[61]	$f_1(x) = 10 + x^2 - 10 \cos(x\pi/2)$ $f_2(x) = (x - 4)^2$	$[-100, 100]$
SK1	[62]	Max. $f_1(x) = -x^4 - 3x^3 + 10x^2 + 10x + 10$ Max. $f_2(x) = -0.5x^4 + 2x^3 + 10x^2 - 10x + 5$	?
SK2	[62]	Max. $f_1(x_1, \dots, x_4) = -(x_1 - 2)^2 - (x_2 + 3)^2 - (x_3 - 5)^2 - (x_4 - 4)^2 + 5$ Max. $f_2(x_1, \dots, x_4) = \frac{\sin x_1 + \sin x_2 + \sin x_3 + \sin x_4}{1 + (x_1^2 + x_2^2 + x_3^2 + x_4^2)/100}$	?
TKLY1	[63]	$f_1(x_1) = x_1$ $f_2(x_1, \dots, x_4) = \frac{1}{x_1} \prod_{i=2}^4 \left[ 2.0 - \exp\left(-\left(\frac{x_i - 0.1}{0.004}\right)^2\right) - 0.8 \exp\left(-\left(\frac{x_i - 0.9}{0.4}\right)^2\right) \right]$	$x_1 \in [0.1, 1]$ $x_2, \dots, x_4 \in [0, 1]$
VU1	[64]	$f_1(x_1, x_2) = \frac{x_1^2 + x_2^2 + 1}{x_1^2 + 3x_2^2 + 1}$ $f_2(x_1, x_2) = x_1^2 + x_2^2 + 1$	$[-3, 3]$
VU2	[64]	$f_1(x_1, x_2) = x_1^2 + x_2^2 + 1$ $f_2(x_1, x_2) = x_1^2 + 2x_2 - 1$	$[-3, 3]$
VFM1	[38]	$f_1(x_1, x_2) = x_1^2 + (x_2 - 1)^2$ $f_2(x_1, x_2) = x_1^2 + (x_2 + 1)^2 + 1$ $f_3(x_1, x_2) = (x_1 - 1)^2 + x_2^2 + 2$	$[-2, 2]$
ZLT1	[11]	$f_{m=1:M}(x_1, \dots, x_n) = (x_m - 1)^2 + \sum_{1 \leq i \leq n, i \neq m} x_i^2$	$[-1000, 1000]$

TABLE XVII

ANALYSIS OF OTHER REAL VALUED MULTIOBJECTIVE TEST PROBLEMS FROM THE LITERATURE. DEGENERATE GEOMETRIES ARE INDICATED BY BRACKETED TERMS; FOR EXAMPLE “(1d)” WITH PROBLEM FA1 INDICATES A ONE-DIMENSIONAL TRADEOFF SURFACE

Name	Objective	R3: # Parameters	F2: Separability	F5: Modality	R1: No Extremal	R2: No Medial	R5: Diss. Domains	R6: Diss. Ranges	R7: Optima Known	F1: Geometry	F3: Bias	F4a: Pareto Many-to-one	F4b: Flat Regions
BK1	$f_1$	2	S	U	✓	×	×	×	✓	convex	—	—	—
	$f_2$	2	S	U									
DPAM1	$f_1$	✓	NS	U	?	?	×	?	×	?	—	?	—
	$f_2$	✓	NS	M									
DGO1	$f_1$	1	NA	M	✓	✓	NA	×	✓	convex	—	+	—
	$f_2$	1	NA	M									
DGO2	$f_1$	1	NA	U	✓	×	NA	×	✓	point (0d)	—	—	—
	$f_2$	1	NA	U									
FA1	$f_1$	1	NA	U	×	✓	×	×	✓	convex (1d)	—	—	—
	$f_2$	3	S	U									
	$f_3$	3	S	U									
Far1	$f_1$	2	NS	M	✓	✓	×	×	×	mixed*	—	—	—
	$f_2$	2	NS	M									
FES1	$f_1$	✓	S	U	✓	✓	×	✓	×	*	—	—	—
	$f_2$	✓	S	U									
FES2	$f_1$	✓	S	U	✓	✓	×	✓	×	*	—	—	—
	$f_2$	✓	S	U									
	$f_3$	✓	S	U									
FES3	$f_1$	✓	S	U	✓	✓	×	✓	×	*	—	—	—
	$f_2$	✓	S	U									
	$f_3$	✓	S	U									
	$f_4$	✓	S	U									
FF1	$f_1$	2	S	U	?	?	?	×	✓	concave	—	—	—
	$f_2$	2	S	U									
IKK1	$f_1$	1	NA	U	✓	×	×	✓	✓	convex (1d)	—	—	—
	$f_2$	1	NA	U									
	$f_3$	1	NA	U									
IM1	$f_1$	1	NA	U	×	✓	✓	✓	✓	concave	—	—	—
	$f_2$	2	S*	U									
JOS1	$f_1$	✓	S	U	?	?	?	×	✓	convex	—	—	—
	$f_2$	✓	S	U									
JOS2	$f_1$	1	NA	U	×	✓	×	✓	✓	mixed*	—	—	—
	$f_2$	✓	S	U									
Kur1	$f_1$	✓	S	U	?	?	?	✓	×	disconnected*	—	—	—
	$f_2$	✓	S	M									
LRS1	$f_1$	2	S	U	✓	×	×	×	✓	convex	—	—	—
	$f_2$	2	S	U									
LTDZ1	$f_1$	3	S*	U	×	✓	×	×	✓	concave	—	+	—
	$f_2$	3	S*	U									
	$f_3$	2	S	U									
LE1	$f_1$	2	S	U	✓	✓	×	✓	✓	concave	+	—	—
	$f_2$	2	S	U									
MHHM1	$f_1$	1	NA	U	✓	✓	NA	✓	✓	convex (1d)	—	—	—
	$f_2$	1	NA	U									
	$f_3$	1	NA	U									
MHHM2	$f_1$	2	S	U	✓	✓	×	✓	✓	convex	—	—	—
	$f_2$	2	S	U									
	$f_3$	2	S	U									
MLF1	$f_1$	1	NA	M	✓	✓	NA	✓	✓	convex	—	—	—
	$f_2$	1	NA	M									
MLF2	$f_1$	2	NS	M	?	?	?	✓	×	convex	—	—	—
	$f_2$	2	NS	M									
QV1	$f_1$	✓	S	M	✓	×	×	×	✓*	concave	—	—	—
	$f_2$	✓	S	M									
Sch1	$f_1$	1	NA	M	?	?	NA	✓	✓	disconnected*	—	—	—
	$f_2$	1	NA	U									
SP1	$f_1$	2	NS	U	?	?	?	×	✓*	convex	—	—	—
	$f_2$	2	NS	U									
SSFYY1	$f_1$	2	S	U	✓	×	×	×	✓	convex	—	—	—
	$f_2$	2	S	U									
SSFYY2	$f_1$	1	NA	M	✓	×	NA	×	✓	disconnected*	—	—	—
	$f_2$	1	NA	U									
SK1	$f_1$	1	S	M	?	?	NA	✓	✓	disconnected*	—	—	—
	$f_2$	1	S	M									
SK2	$f_1$	4	S	U	?	?	?	✓	×	disconnected*	—	—	—
	$f_2$	4	NS	M									
TKLY1	$f_1$	1	NA	U	✓	✓	×	✓	✓	convex	—	—	—
	$f_2$	4	S	M									
VU1	$f_1$	2	S	U	✓	✓	×	✓	✓*	convex	—	+	—
	$f_2$	2	S	U									
VU2	$f_1$	2	S	U	×	×	×	✓	✓	convex	—	—	—
	$f_2$	2	S	U									
VFM1	$f_1$	2	S	U	✓	×	×	×	✓	convex	—	—	—
	$f_2$	2	S	U									
	$f_3$	2	S	U									
ZLT1	$f_{1:A}$	✓	S	U	✓	×	×	×	✓	convex	—	—	—

This paper could be extended by performing a rigorous review of other types of test problems, including problems with side constraints, noisy problems, and problems whose parameters are not restricted to the domain of real numbers. The features and recommendations introduced in this paper could be extended to support the requirements of additional problem classes.

#### APPENDIX DETAILS OF OTHER TEST PROBLEMS

Table XVI presents a number of other problems from the literature; for convenience, we have associated our own name with each problem. Table XVII presents the analysis of these problems. A “?” is used to indicate problems for which the domain of parameters has not been explicitly stated. We note that for various reasons it is not uncommon for other authors to change parameter domains, although for brevity we only consider each problem in its original form. Far1, Kur1, LTDZ1, MFL1, and SK1 all appear to have typographic errors in their original papers (as can be determined by comparison to figures provided, etc.); each has been corrected in the table as appropriate. Note also that the rotation matrix was not specified for DPAM1.

The analysis of some of the problems requires further comment:

- Technically, IM1  $f_2$  is nonseparable, but as this is only evident under very limited circumstances, we have listed it as being separable. Similar reasoning applies to  $f_1$  and  $f_2$  of LTDZ1.
- Far1 has a connected, mixed Pareto optimal front that consists of a concave component, and several not smoothly connected convex components. Far1 has a disconnected Pareto optimal set. We also consider the tradeoff magnitudes of Far1 to be close enough to count as being similar.
- JOS2 has a mixed convex/concave tradeoff geometry.
- The Pareto geometry of Kur1 varies depending on the number of parameters. With two parameters, both the Pareto optimal set and front are disconnected, where the front includes a degenerate point and a mixed convex/concave component that is not smoothly connected to a convex component.
- Sch1, SK1, and SSFY2 all have disconnected Pareto optimal fronts and sets, where each front consists of two convex components.
- The disconnected Pareto optimal front of SK2 consists of two convex components.
- The Pareto optimal geometries of FES1, FES2, and FES3 all vary with the number of parameters. With one parameter, FES1 has a mixed convex/concave Pareto optimal front. The Pareto optimal fronts of FES2 and FES3 are more convoluted, but appear to include aspects of mixed convexity/concavity.
- The task of identifying the Pareto optimal set for each of QV1, SP1, and VU1 appears to be tractable. Conversely, the task of identifying the Pareto optimal set for each of FES1, FES2, FES3, and MFL2 does not immediately appear to be tractable.

#### ACKNOWLEDGMENT

The authors thank the anonymous reviewers for their comments and suggestions.

#### REFERENCES

- [1] Y. Collette and P. Siarry, *Multiobjective Optimization. Principles and Case Studies*. Berlin, Germany: Springer-Verlag, 2003.
- [2] D. Büche, P. Stoll, R. Dornberger, and P. Koumoutsakos, “Multiobjective evolutionary algorithm for the optimization of noisy combustion processes,” *IEEE Trans. Syst., Man, and Cybern.—Part C: Applications and Reviews*, vol. 32, no. 4, pp. 460–473, Nov. 2002.
- [3] L. Barone, L. While, and P. Hingston, “Designing crushers with a multi-objective evolutionary algorithm,” in *Proc. Genetic and Evol. Comput. Conf.*, W. B. Langdon, E. Cantú-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M. A. Potter, A. C. Schultz, J. F. Miller, E. Burke, and N. Jonoska, Eds., Jul. 2002, pp. 995–1002.
- [4] J. M. de la Cruz García, J. L. Risco Martín, A. Herrán González, and P. Fernández Blanco, “Hybrid heuristic and mathematical programming in oil pipeline networks,” in *Proc. Congr. Evol. Comput.*, vol. 2, June 2004, pp. 1479–1486.
- [5] N. V. Venkatarayalu and T. Ray, “Single and multi-objective design of Yagi-Uda antennas using computational intelligence,” in *Proc. Congr. Evol. Comput.*, vol. 2, R. R. Ruhul Sarker, H. Abbass, K. C. Tan, B. McKay, D. Essam, and T. Gedeon, Eds., Dec. 2003, pp. 1237–1242.
- [6] P. Engrand, “A multi-objective optimization approach based on simulated annealing and its application to nuclear fuel management,” in *Proc. 5th Int. Conf. Nuclear Eng.*, 1997, pp. 416–423.
- [7] K. Shaw, A. L. Nortcliffe, M. Thompson, J. Love, P. J. Fleming, and C. M. Fonseca, “Assessing the performance of multiobjective genetic algorithms for optimization of a batch process scheduling problem,” in *Proc. Congr. Evol. Comput.*, vol. 1, P. J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and A. Zalzala, Eds., July 1999, pp. 37–45.
- [8] S. Watanabe, T. Hiroyasu, and M. Miki, “Parallel evolutionary multi-criterion optimization for mobile telecommunication networks optimization,” in *EUROGEN 2001—Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, K. C. Giannakoglou, D. T. Tsahalis, J. Périaux, K. D. Papailiou, and T. Fogarty, Eds. Barcelona, Spain: EdsInternational Center for Numerical Methods in Engineering (CIMNE), Sept. 2001, pp. 167–172.
- [9] E. J. Hughes, “Swarm guidance using a multi-objective co-evolutionary on-line evolutionary algorithm,” in *Proc. Congr. Evol. Comput.*, vol. 2, June 2004, pp. 2357–2363.
- [10] S. Huband, P. Hingston, L. While, and L. Barone, “An evolution strategy with probabilistic mutation for multiobjective optimization,” in *Proc. Congr. Evol. Comput.*, vol. 4, R. R. Ruhul Sarker, H. Abbass, K. C. Tan, B. McKay, D. Essam, and T. Gedeon, Eds., Dec. 2003, pp. 2284–2291.
- [11] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization,” in *EUROGEN 2001—Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, K. C. Giannakoglou, D. T. Tsahalis, J. Périaux, K. D. Papailiou, and T. Fogarty, Eds. Barcelona, Spain: International Center for Numerical Methods in Engineering (CIMNE), Sep. 2001, pp. 95–100.
- [12] R. C. Purshouse and P. J. Fleming, “Why use elitism and sharing in a multi-objective genetic algorithm?,” in *Proc. Genetic and Evol. Comput. Conf.*, W. B. Langdon, E. Cantú-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M. A. Potter, A. C. Schultz, J. F. Miller, E. Burke, and N. Jonoska, Eds., Jul. 2002, pp. 520–527.
- [13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
- [14] D. H. Wolpert and W. G. Macready, “No free lunch theorem for search,” Santa Fe Institute, Res. Rep. SFI-TR-95-02-010, 1995.
- [15] ———, “No free lunch for optimization,” *IEEE Trans. Evol. Comput.*, vol. 1, pp. 67–82, Apr. 1997.
- [16] C. M. F. V. Grunert da Fonseca and A. O. Hall, “Inferential performance assessment of stochastic optimizers and the attainment function,” in *Lecture Notes in Computer Science*. Berlin, Germany: Springer-Verlag, 2001, vol. 1993, Proc. 1st Int. Conf. Evol. Multi-Criterion Optimization, pp. 213–225.
- [17] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, “Performance assessment of multiobjective optimizers: An analysis and review,” *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.
- [18] T. Okabe, Y. Jin, and B. Sendhoff, “A critical survey of performance indices for multi-objective optimization,” in *Proc. Congr. Evol. Comput.*, vol. 2, R. R. Ruhul Sarker, H. Abbass, K. C. Tan, B. McKay, D. Essam, and T. Gedeon, Eds., Dec. 2003, pp. 878–885.
- [19] J. Knowles and D. Corne, “On metrics for comparing nondominated sets,” in *Proc. Congr. Evol. Comput.*, vol. 1, D. B. Fogel, M. A. El-Sharkawi, X. Yao, G. Greenwood, H. Iba, P. Marrow, and M. Shackleton, Eds., May 2002, pp. 711–716.

- [20] B. F. J. Manly, *Randomization, Bootstrap and Monte Carlo Methods in Biology*. London, U.K.: Chapman & Hall, 1991.
- [21] P. Good, *Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses*, 2nd ed. Berlin, Germany: Springer-Verlag, 2000. ser. Springer Series in Statistics.
- [22] K. Deb, "Multi-objective genetic algorithms: Problem difficulties and construction of test problems," *Evol. Comput.*, vol. 7, no. 3, pp. 205–230, Fall 1999.
- [23] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, *Scalable Test Problems for Evolutionary Multi-Objective Optimization*. Kanpur, India: Kanpur Genetic Algorithms Lab. (KanGAL), Indian Inst. Technol., 2001. KanGAL Report 2001/001.
- [24] D. E. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning*. Reading, MA: Addison-Wesley, 1989.
- [25] D. B. Fogel and H.-G. Beyer, "A note on the empirical evaluation of intermediate recombination," *Evol. Comput.*, vol. 3, no. 4, pp. 491–495, Winter 1995.
- [26] C. M. Fonseca and P. J. Fleming, "On the performance assessment and comparison of stochastic multiobjective optimizers," in *Lecture Notes in Computer Science*, H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, Eds. Berlin, Germany: Springer-Verlag, Sep. 1996, vol. 1141, Proc. 4th Int. Conf. Parallel Problem Solving from Nature—PPSN IV, pp. 584–593. ser. .
- [27] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York: Wiley, 2001.
- [28] D. Whitley, K. Mathias, S. Rana, and J. Dzubera, "Building better test functions," in *Proc. 6th Int. Conf. Genetic Algorithms*, L. J. Eshelman, Ed., July 1995, pp. 239–246.
- [29] *Handbook of Evolutionary Computation*, pt. B2.7, T. Bäck, D. Fogel, and Z. Michalewicz, Eds., Oxford Univ. Press, Oxford, U.K., 1997, pp. 14–20. T. Bäck, Z. Michalewicz, "Test landscapes".
- [30] *Handbook of Evolutionary Computation*, Oxford Univ. Press, Oxford, U.K., 1997. T. Bäck, D. Fogel, Z. Michalewicz.
- [31] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, Summer 2000.
- [32] OHD. A. Van Veldhuizen, "Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations," Ph.D. Dissertation, Air Force Institute of Technology, Wright-Patterson AFB, Jun. 1999.
- [33] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proc. Congr. Evol. Comput.*, D. B. Fogel, M. A. El-Sharkawi, X. Yao, G. Greenwood, H. Iba, P. Marrow, and M. Shackleton, Eds., May 2002, vol. 1, pp. 825–830.
- [34] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proc. 1st Int. Conf. Genetic Algorithms and Their Applications*, J. J. Grenfenstett, Ed., 1985, pp. 93–100.
- [35] C. M. Fonseca and P. J. Fleming, "Multiobjective genetic algorithms made easy: Selection, sharing and mating restriction," *Genetic Algorithms in Engineering Systems: Innovations and Applications*, pp. 45–52, Sep. 1995. IEE.
- [36] C. Poloni, G. Mosetti, and S. Contessi, "Multi objective optimization by GAs: Application to system and component design," in *Proc. Comput. Methods in Applied Sciences '96: Invited Lectures and Special Technological Sessions of the 3rd ECCOMAS Comput. Fluid Dynamics Conf. and the 2nd ECCOMAS Conf. Numerical Methods in Engineering*, Sep. 1996, pp. 258–264.
- [37] F. Kursawe, "A variant of evolution strategies for vector optimization," in *Lecture Notes in Computer Science*, H.-P. Schwefel and R. Männer, Eds. Berlin, Germany: Springer-Verlag, 1991, vol. 496, Proc. Parallel Problem Solving From Nature. 1st Workshop, PPSN I, pp. 193–197.
- [38] R. Viennet, C. Fonteix, and I. Marc, "Multicriteria optimization using a genetic algorithm for determining a Pareto set," *Int. J. Syst. Sci.*, vol. 27, no. 2, pp. 255–260, 1996.
- [39] P. J. Bentley and J. P. Wakefield, "Finding acceptable solutions in the pareto-optimal range using multiobjective genetic algorithms," in *Soft Computing in Engineering Design and Manufacturing*, P. K. Chawdhry, R. Roy, and R. K. Pant, Eds. Berlin, Germany: Springer-Verlag, June 1998, pp. 231–240.
- [40] J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the Pareto archived evolution strategy," *Evol. Comput.*, vol. 8, no. 2, pp. 149–172, Summer 2000.
- [41] T. Okabe, "Evolutionary multi-objective optimization," Ph.D. dissertation, Honda Research Institute, Europe, 2004.
- [42] T. Okabe, Y. Jin, M. Olhofer, and B. Sendhoff, "On test functions for evolutionary multi-objective optimization," in *Lecture Notes in Computer Science*. Berlin, Germany: Springer-Verlag, 2004, vol. 3242, Proc. Parallel Problem Solving from Nature (PPSN VIII), p. 802.
- [43] S. Huband, L. Barone, L. While, and P. Hingston, "A scalable multi-objective test problem toolkit," in *Lecture Notes in Computer Science*. Berlin, Germany: Springer-Verlag, Mar. 2005, vol. 3410, Proc. Evolutionary Multi-Criterion Optimization: 3rd Int. Conf., pp. 280–294.
- [44] S. A. K. Deb, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.
- [45] T. T. Binh and U. Korn, "An evolution strategy for the multiobjective optimization," in *Pro. 2nd Int. Conf. Genetic Algorithms*, Jun. 1996, pp. 23–28.
- [46] D. Dumitrescu, C. Groşan, and M. Oltean, "A new evolutionary approach for multiobjective optimization," *Studia Universitatis Babeş-Bolyai, Informatica*, vol. XLV, no. 1, pp. 51–68, 2000.
- [47] A. Farhang-Mehr and S. Azarm, "Diversity assessment of pareto optimal solution sets: An entropy approach," in *Proc. Congr. Evol. Comput.*, vol. 1, D. B. Fogel, M. A. El-Sharkawi, X. Yao, G. Greenwood, H. Iba, P. Marrow, and M. Shackleton, Eds., May 2002, pp. 723–728.
- [48] M. Farina, "A neural network based generalized response surface multiobjective evolutionary algorithm," in *Proc. Congr. Evol. Comput.*, vol. 1, D. B. Fogel, M. A. El-Sharkawi, X. Yao, G. Greenwood, H. Iba, P. Marrow, and M. Shackleton, Eds., May 2002, pp. 956–961.
- [49] J. E. Fieldsend, R. M. Everson, and S. Singh, "Using unconstrained elite archives for multi-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 7, no. 3, pp. 305–323, June 2003.
- [50] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," in *Evol. Comput.*: Spring, 1995, vol. 3, pp. 1–16.
- [51] K. Ikeda, H. Kita, and S. Kobayashi, "Failure of Pareto-based MOEA's : Does nondominated really mean near to optimal?," in *Proc. Congr. Evol. Comput.*, vol. 2, May 2001, pp. 957–962.
- [52] H. Ishibuchi and T. Murata, "A multi-objective genetic local search algorithm and its application to flowshop scheduling," *IEEE Trans. Syst., Man, and Cyberne.—Part C: Applications and Reviews*, vol. 28, no. 3, pp. 392–403, Aug. 1998.
- [53] Y. Jin, T. Okabe, and B. Sendhoff, "Dynamic weighted aggregation for evolutionary multi-objective optimization: Why does it work and how?," in *Proc. Genetic and Evol. Comput. Conf.*, L. Spector, E. D. Goodman, A. Wu, W. B. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshek, M. H. Garzon, and E. Burke, Eds., Jul. 2001, pp. 1042–1049.
- [54] M. Laumanns, G. Rudolph, and H.-P. Schwefel, "A spatial predator-prey approach to multi-objective optimization: A preliminary study," in *Lecture Notes in Computer Science*, A. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, Eds. Berlin, Germany: Springer-Verlag, Sep. 1998, vol. 1498, Proc. Parallel Problem Solving from Nature—PPSN V, pp. 241–249.
- [55] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multi-objective optimization," *Evol. Comput.*, vol. 10, no. 3, pp. 263–282, Fall 2002.
- [56] J. Lis and A. E. Eiben, "Multi-sexual genetic algorithm for multiobjective optimization," in *Proc. IEEE Int. Conf. Evol. Comput.*, Apr. 1997, pp. 59–64.
- [57] J. Mao, K. Hirasawa, J. Hu, and J. Murata, "Genetic symbiosis algorithm for multiobjective optimization problem," in *Proc. IEEE Int. Workshop on Robot and Human Interactive Commun.*, Sep. 2000, pp. 137–142.
- [58] A. K. Molyneaux, G. B. Leyland, and D. Favrat, "A new, clustering evolutionary multi-objective optimization technique," in *Proc. 3rd Int. Symp. Adaptive Systems—Evol. Comput. and Probabilistic Graphical Models*, Mar. 2001, pp. 41–47.
- [59] D. Quagliarella and A. Vicini, "Sub-population policies for a parallel multiobjective genetic algorithm with applications to wing design," in *Proc. IEEE Int. Conf. Syst., Man, and Cybern.*, vol. 4, Oct. 1998, pp. 3142–3147.
- [60] M. Sefrioui and J. Periaux, "Nash genetic algorithms: Examples and applications," in *Proc. Congr. Evol. Comput.*, vol. 1, July 2000, pp. 509–516.
- [61] M.-B. Shim, M.-W. Suh, T. Furukawa, G. Yagawa, and S. Yoshimura, "Pareto-based continuous evolutionary algorithms for multiobjective optimization," *Eng. Comput.*, vol. 19, no. 1, pp. 22–48, 2002.
- [62] K. Socha and M. Kisiel-Dorohinicki, "Agent-based evolutionary multiobjective optimization," in *Proc. Congr. Evol. Comput.*, vol. 1, D. B. Fogel, M. A. El-Sharkawi, X. Yao, G. Greenwood, H. Iba, P. Marrow, and M. Shackleton, Eds., May 2002, pp. 109–114.
- [63] K. C. Tan, E. F. Khor, T. H. Lee, and Y. J. Yang, "A tabu-based exploratory evolutionary algorithm for multiobjective optimization," *Artif. Intell. Rev.*, vol. 19, no. 3, pp. 231–260, May 2003.
- [64] M. Valenzuela-Rendón and E. Uresti-Charre, "A nongenerational genetic algorithm for multiobjective optimization," *Proc. 7th Int. Conf. Genetic Algorithms*, pp. 658–665, Jul. 1997.



**Simon Huband** (M'04) received the B.Sc. and Ph.D. degrees from the University of Western Australia, Crawley, in 1997 and 2003, respectively.

He is currently a Research Fellow at the School of Computer and Information Science, Edith Cowan University, Mount Lawley, Australia. His research interests include parallel programming, the design of test problems, and the optimization of industrial systems using evolutionary algorithms.



**Luigi Barone** (M'04) received the B.Sc. and Ph.D. degrees from the University of Western Australia, Crawley, in 1994 and 2004, respectively.

He is currently an Associate Lecturer in the School of Computer Science and Software Engineering, University of Western Australia. His research interests include evolutionary algorithms and their use for optimization and opponent modeling, and the modeling of biological systems.



**Phil Hingston** (M'00) received the B.Sc. degree from the University of Western Australia, Crawley, in 1978, and the Ph.D. degree from Monash University, Melbourne, Australia, in 1984.

He is currently a Senior Lecturer in the School of Computer and Information Science, Edith Cowan University, Mount Lawley, Australia. His research interests include artificial intelligence and its application to industrial design tasks, and the modeling of social and natural systems.



**Lyndon While** (M'01–SM'03) received the B.Sc.(Eng.) and Ph.D. degrees from the Imperial College of Science and Technology, London, U.K., in 1985 and 1988, respectively.

He is currently a Senior Lecturer in the School of Computer Science and Software Engineering, University of Western Australia, Crawley. His research interests include evolutionary algorithms, multiobjective optimization, and the semantics and implementation of functional programming languages.