A Scalable Multi-objective Test Problem Toolkit

Simon Huband[1], Luigi Barone[2], Lyndon While[2], and Phil Hingston[1]

[1] Edith Cowan University, Mount Lawley WA 6050, Australia,
`{s.huband,p.hingston}@ecu.edu.au`
[2] The University of Western Australia, Crawley WA 6009, Australia
`{luigi,lyndon}@csse.uwa.edu.au`

Abstract. This paper presents a new toolkit for creating scalable multi-objective test problems. The WFG Toolkit is flexible, allowing characteristics such as bias, multi-modality, and non-separability to be incorporated and combined as desired. A wide variety of Pareto optimal geometries are also supported, including convex, concave, mixed convex/concave, linear, degenerate, and disconnected geometries.

All problems created by the WFG Toolkit are well defined, are scalable with respect to both the number of objectives and the number of parameters, and have known Pareto optimal sets. Nine benchmark multi-objective problems are suggested, including one that is both multi-modal and non-separable, an important combination of characteristics that is lacking among existing (scalable) multi-objective problems.

1 Introduction

There have been several attempts to define test suites and toolkits for testing multi-objective evolutionary algorithms (MOEAs) [1–4]. However, existing multi-objective test problems do not test a wide range of characteristics, and are often poorly designed. Typical defects include not being scalable and being susceptible to simple search strategies. Moreover, many problems are poorly constructed, with unknown Pareto optimal sets, or featuring parameters with poorly located optima.

As suggested for single-objective problems by Whitley et al. [5] and Bäck and Michalewicz [6], test suites should include scalable problems that are resistant to hill climbing strategies, are non-linear, non-separable[1], and multi-modal. Such requirements are also a good start for multi-objective test suites, but unfortunately are poorly represented in the literature.

Addressing this problem, this paper presents the Walking Fish Group (WFG) Toolkit, which places an emphasis on allowing test problem designers to construct scalable test problems with any number of objectives, where features such as modality and separability can be customised as required. Test problems in the WFG Toolkit are defined in terms of a simple underlying problem that defines

---

[1] Separable problems can be optimised by considering each parameter in turn, independently of one another. A non-separable problem is thus characterised by parameter dependencies, is more difficult, and is more representative of real world problems.

the fitness space and a series of composable, configurable transformations that allow the test problem designer to add arbitrarily levels of complexity to the test problem. Problems created by the WFG Toolkit are well defined, are scalable with respect to both the number of objectives and the number of parameters, and have known Pareto optimal sets.

The next section of the paper introduces the multi-objective terminology used throughout. Section 3 briefly examines previous multi-objective test suites, highlighting the deficiencies with them. Section 4 specifies our new WFG Toolkit, generalising the concepts introduced in these previous test suites to produce a configurable toolkit that allows for the construction of scalable, well-behaved test problems. Section 5 then describes how the WFG Toolkit can be used to construct an example test problem. Some experimental results are presented in Section 6. A suite of nine test problems are proposed in Section 7 that exceeds the functionality of previous test suites. Section 8 concludes the paper.

2 Terminology

Consider a multi-objective optimisation problem given in terms of a search space of allowed values of n parameters $x_1, \ldots, x_n$, and a vector of M objective functions $\{f_1, \ldots, f_M\}$ mapping parameter vectors into fitness space. The mapping from the search space to fitness space defines the *fitness landscape*.

In multi-objective optimisation, we aim to find the set of optimal trade-off solutions known as the *Pareto optimal set*. 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.

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. A *distance parameter* is one that when modified only ever results in a dominated, dominating, or equivalent parameter vector. A *position parameter* is one that when modified only ever results in an incomparable or equivalent parameter vector. All other parameters are *mixed parameters*.

When the projection of the Pareto optimal set onto the domain of a single parameter, the parameter optima, is a single value at the edge of the domain, then we call the parameter an *extremal parameter*. If instead the parameter optima cluster around the middle of the domain, then it is a *medial parameter*. Extremal parameters can be unduly favoured by truncation based mutation correction strategies, whereas medial parameters can be favoured by EAs that employ intermediate recombination [7].

3 Previous Multi-objective Test Problems

Deb's toolkit [1] for constructing two-objective problems is the only toolkit for multi-objective problems of which we are aware. Deb's toolkit segregates parameters into distance and position parameters — mixed parameters are atypical.

Three functionals are used that control the shape of and position on the tradeoff surface, and the distance to the Pareto optimal front. Deb's toolkit provides a number of functionals, including multi-modal and biased functions, most of which are scalable parameter-wise.

Deb's toolkit has various limitations: it was designed for two-objective problems, no real-valued deceptive functions are suggested, the suggested functions do not facilitate the construction of problems with degenerate Pareto optimal front geometries[2], only one non-separable function is suggested (but it scales poorly and has but weak parameter dependencies), and position and distance parameters are always independent of one another[3].

Related by authorship to Deb's toolkit are the DTLZ test problems [4, 8], which, unlike the majority of multi-objective problems, are scalable objective-wise. This important characteristic has facilitated several recent investigations into what are commonly called "many" objective problems. Like Deb's toolkit, the DTLZ problems have distinct distance and position components, have known Pareto optimal fronts, and are simple to employ. The DTLZ problems also address a variety of problem characteristics, including multi-modality, bias, and several Pareto optimal front geometries.

However, the DTLZ test suite has serious limitations: none of its problems is deceptive, none of its problems is non-separable[4], and the number of position parameters is always fixed relative to the number of objectives. DTLZ5 and DTLZ6 also deserve special mention, as they are both meant to be problems with degenerate Pareto optimal fronts. However, we have found that this is untrue for instances with four or more objectives (due to space limitations, we omit the proof). As DTLZ5 and DTLZ6 do not behave as expected, their Pareto optimal fronts are unclear beyond three-objectives.

Whilst other test problems exist, including those employed by Van Veldhuizen [9], Zitzler et al. [10], and others [11, 12], they tend to be of limited scope, and are often ad hoc and consequently difficult to analyse (but by the same token, some also have unusual Pareto optimal geometries). Many are restricted to three or fewer parameters or objectives, some have poorly located parameter optima, few are non-separable (and even fewer are both non-separable and multi-modal), and those that are non-separable are either not scalable, or have unknown Pareto optimal fronts.

Despite the variety of existing test problems, there is clear need for additional work. At present there is no toolkit for creating problems with an arbitrary number of objectives, where desirable features can easily be incorporated or omitted as desired. We remedy this problem with our WFG Toolkit.

---

[2] A degenerate front is a front that is of lower dimension than the objective space in which it is embedded, less one.

[3] Deb does suggest a way of making position and distance parameters mutually non-separable. However, the suggested approach can lead to cyclical dependencies, potentially causing unwanted side effects on the fitness landscape.

[4] Technically speaking, the majority of the DTLZ problems are non-separable, but only marginally so.

4 The WFG Toolkit

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 multi-modality and non-separability. 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 labelled 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:

$$
\begin{array}{ll}
\text{Given} & \mathbf{z} = \{z_1, \ldots, z_k, z_{k+1}, \ldots, z_n\} \\
\text{Minimise} & f_{m=1:M}(\mathbf{x}) = x_M + S_m h_m(x_1, \ldots, x_{M-1}) \\
\text{where} & \mathbf{x} = \{x_1, \ldots, x_M\} = \{\max(t^p_M, A_1)(t^p_1 - 0.5) + 0.5, \ldots, \\
 & \qquad \max(t^p_M, A_{M-1})(t^p_{M-1} - 0.5) + 0.5, t^p_M\} \\
 & \mathbf{t}^p = \{t^p_1, \ldots, t^p_M\} \mapsfrom \mathbf{t}^{p-1} \mapsfrom \ldots \mapsfrom \mathbf{t}^1 \mapsfrom \mathbf{z}_{[0,1]} \\
 & \mathbf{z}_{[0,1]} = \{z_{1,[0,1]}, \ldots, z_{n,[0,1]}\} = \{z_1/z_{1,\max}, \ldots, z_n/z_{n,\max}\}
\end{array}
$$

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), $A_{1:M-1} \in \{0,1\}$ 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 "$\mapsfrom$" 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 normalise 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 normalise fitness values).

4.1 Shape Functions

Shape functions determine the nature of the Pareto optimal front, and map parameters with domain $[0, 1]$ onto the range $[0, 1]$. Each of $h_{1:M}$ must be associated with a shape function. For example, letting $h_1 = \text{linear}_1$, $h_{m=2:M-1} = \text{convex}_m$, and $h_M = \text{mixed}_M$ indicates that h_1 uses the linear shape function, h_M uses the mixed shape function, and all of $h_{2:M-1}$ use convex shape functions.

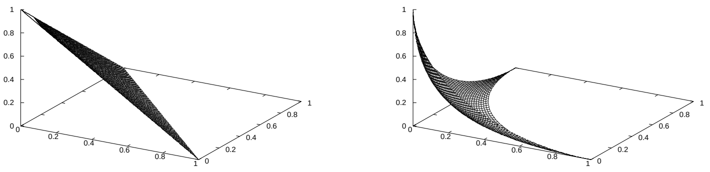
Table 1 presents five different types of shape functions. Example Pareto optimal fronts constructed using these shape functions are given in Fig. 1.

Table 1. Shape functions. In all cases, $x_1, \ldots, x_{M-1} \in [0, 1]$. A, α, and β are constants.

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

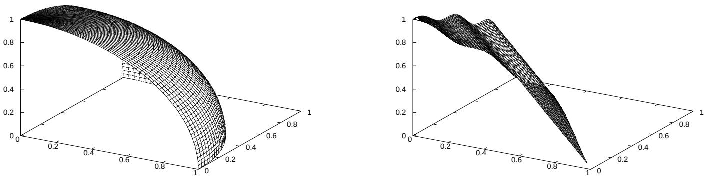
4.2 Transformation Functions

Transformation functions map input parameters with domain $[0, 1]$ onto the range $[0, 1]$. All transformation functions take a vector of parameters (called the primary parameters) and map them to a single value. Transformation functions may also employ constants and secondary parameters that further influence the mapping. Primary parameters allow us to qualify working parameters as being position- and distance-related.



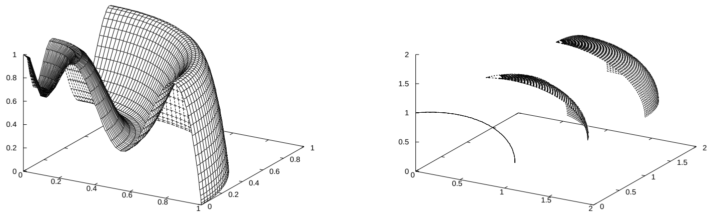
(a) $h_{m=1:3} = \text{linear}_m$.

(b) $h_{m=1:3} = \text{convex}_m$.



(c) $h_{m=1:3} = \text{concave}_m$.

(d) $h_{m=1:2} = \text{linear}_m$, $h_3 = \text{mixed}_3$ $(\alpha = 0.4,\ A = 3)$.



(e) $h_{m=1:2} = \text{concave}_m$, $h_3 = \text{disc}_3$ (dominated regions not removed, $\alpha = 0.4$, $\beta = 0.4$, $A = 3$).

(f) $h_{m=1:3} = \text{concave}_m$, degenerate on x_2, shown for distances 0, 0.5, and 1.

Fig. 1. Example three-objective Pareto optimal fronts.

There are three types of transformation functions: bias, shift, and reduction functions. Bias and shift functions only ever employ one primary parameter, whereas reduction functions can employ many.

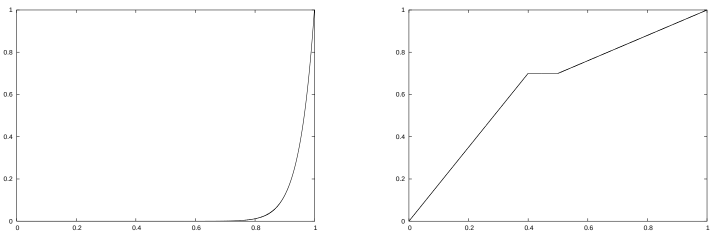
Bias transformations have a natural impact on the search process by biasing the fitness landscape. Shift transformations move the location of optima. In the absence of any shift, all distance-related parameters would be extremal parameters, with optimal value at zero. Shift transformations can be used 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 to at least one shift transformation.

The transformation functions are specified in Table 2 and plotted in Fig. 2. To ensure problems are well designed, some restrictions apply as given in Table 3. For brevity, we have omitted a weighted product reduction function (analogous to the weighted sum reduction function).

Table 2. Transformation functions. The primary parameters y and $y_1, \ldots, y_{\lvert\mathbf{y}\rvert}$ always have domain $[0,1]$. A, B, C, α, and β are constants. For b_param, $\mathbf{y}'$ is a vector of secondary parameters (of domain $[0,1]$), and u is a reduction function.

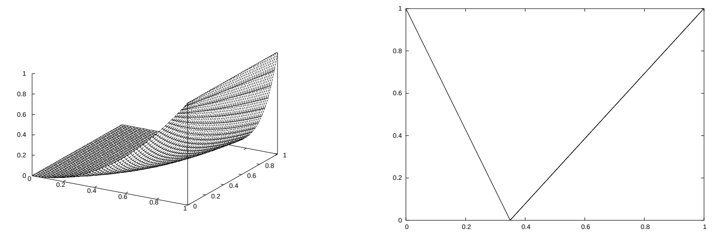
Bias: Polynomial ($\alpha > 0$, $\alpha \neq 1$) $\text{b\_poly}(y, \alpha) = y^{\alpha}$ When $\alpha > 1$ or when $\alpha < 1$, y is biased towards zero or towards one respectively.
Bias: Flat Region ($A, B, C \in [0,1]$, $B < C$, $B = 0 \Rightarrow A = 0 \wedge C \neq 1$, $C = 1 \Rightarrow A = 1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

By incorporating secondary parameters via a reduction function, b_param can create dependencies between distinct parameters, including position- and distance-related parameters. Moreover, when employed before any shift transformation, b_param can create objectives that are effectively non-separable — a separable optimisation approach would fail unless given multiple iterations, or a specific order of parameters to optimise.



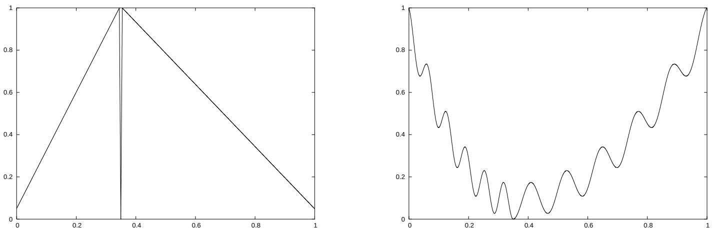
(a) b_poly ($\alpha = 20$).

(b) b_flat ($A = 0.7$, $B = 0.4$, $C = 0.5$).



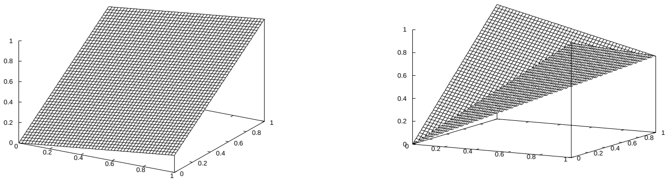
(c) b_param plotted against u ($A = 0.5$, $B = 2$, $C = 10$).

(d) s_linear ($A = 0.35$).



(e) s_decept ($A = 0.35$, $B = 0.005$, $C = 0.05$).

(f) s_multi ($A = 5$, $B = 10$, $C = 0.35$).



(g) r_sum for two parameters ($w_1 = 1$, $w_2 = 5$).

(h) r_nonsep for two parameters ($A = 2$).

Fig. 2. Example transformations. Each example plots the value of the input primary parameter(s) versus the result of the transformation.

Table 3. 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 b_param. Consider the following terminology: if a is a primary parameter of b_param, and b is one of the secondary parameters, then we say that a *depends* on b. If b likewise depends on c, then a also (indirectly) depends on c. When a depends on some parameter b, 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.
b_flat	When $A = 0$, b_flat should only belong to transition vectors that are further away from the underlying parameter vector than any shift or reduction transformation.

The deceptive and multi-modal shift transformations make the corresponding problem deceptive and multi-modal respectively[5]. When applied to position-related parameters, some regions of the Pareto optimal set can become difficult to find, and the mapping from the Pareto optimal set to the Pareto optimal front will be many-to-one (even when $k = M - 1$)[6]. When applied to distance-related parameters, finding any Pareto optimal solution becomes more difficult.

The flat region transformation can have a significant impact on the fitness landscape[7], and can also be used to create a stark many-to-one mapping from the Pareto optimal front to the Pareto optimal set.

5 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 meta-language could be used to help select and connect together the different components making up the test problem. With the use of sensible default 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 4 and expanded in Fig. 3.

This example problem is scalable both objective- and parameter-wise, where the number of distance- and position-related parameters can be scaled indepen-

---

[5] Multi-modal problems are difficult because an optimiser can become stuck in local optima. Deceptive problems (as defined by Deb [1]) exacerbate this difficulty by placing the global optimum in an unlikely place.

[6] Many-to-one mappings from the Pareto optimal set to the Pareto optimal front present difficulties to the optimiser, as choices must be made between two otherwise equivalent parameter vectors.

[7] Optimisers can have difficulty with flat regions due to a lack of gradient information

Table 4. 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 $\mathbf{y} = \mathbf{z}_{[0,1]} = \{z_1/2, \ldots, z_n/(2n)\}$.

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 trade-off 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.
$\mathbf{t}^1$	$t^1_{i=1:n-1} = \text{b\_param}(y_i, \text{r\_sum}(\{y_{i+1}, \ldots, y_n\}, \{1, \ldots, 1\}), \frac{0.98}{49.98}, 0.02, 50)$ $t^1_n = y_n$ 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.
$\mathbf{t}^2$	$t^2_{i=1:k} = \text{s\_decept}(y_i, 0.35, 0.001, 0.05)$ $t^2_{i=k+1:n} = \text{s\_multi}(y_i, 30, 95, 0.35)$ 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.
$\mathbf{t}^3$	$t^3_{i=1:M-1} = \text{r\_nonsep}(\{y_{(i-1)k/(M-1)}, \ldots, y_{ik/(M-1)}\}, k/(M-1))$ $t^3_M = \text{r\_nonsep}(\{y_{k+1}, \ldots, y_n\}, l)$ 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.

dently. For a solution to be Pareto optimal, it is required that all of:

$$z_{i=k+1:n} = \begin{cases} 0.35^{\left(0.02+1.96\text{r\_sum}(\{z_{i+1},\ldots,z_n\},\{1,\ldots,1\})\right)^{0.5}}, & i \neq n \\ 0.35, & i = n \end{cases}$$

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 non-separable, deceptive, and multi-modal, 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 non-separable and multi-modal — we are not aware of any problem in the literature with comparable characteristics.

$$
\begin{aligned}
\text{Given} \qquad & \mathbf{z} = \{z_1, \ldots, z_k, z_{k+1}, \ldots, z_n\} \\
\text{Minimise} \qquad & f_1(\mathbf{x}) = x_M + 2 \textstyle\prod_{i=1}^{M-1} \sin(x_i \pi/2) \\
& f_{m=2:M-1}(\mathbf{x}) = x_M + 2m \left(\textstyle\prod_{i=1}^{M-m} \sin(x_i \pi/2)\right) \cos(x_{M-m+1}\pi/2) \\
& f_M(\mathbf{x}) = x_M + 2M \cos(x_1 \pi/2) \\
\text{where} \qquad & x_{i=1:M-1} = \text{r\_nonsep}(\{y_{(i-1)k/(M-1)}, \ldots, y_{ik/(M-1)}\}, k/(M-1)) \\
& x_M = \text{r\_nonsep}(\{y_{k+1}, \ldots, y_n\}, l) \\
& y_{i=1:k} = \text{s\_decept}(y'_i, 0.35, 0.001, 0.05) \\
& y_{i=k+1:n} = \text{s\_multi}(y'_i/, 30, 95, 0.35) \\
& y'_{i=1:k} = \text{b\_param}\left(z_i/(2i), \textstyle\sum_{j=i+1}^{n} z_j/(2n-2i), 0.5, 0.02, 50\right) \\
& y'_{i=k+1:n} = \text{b\_param}\left(z_i/(2i), \textstyle\sum_{j=1}^{i-1} z_j/(2i-2), 0.5, 0.02, 50\right)
\end{aligned}
$$

Fig. 3. The expanded form of the problem defined in Table 4. $|\mathbf{z}| = n = k + l$, $k \in \{M-1, 2(M-1), 3(M-1), \ldots\}$, $l \in \{1, 2, \ldots\}$, and the domain of all $z_i \in \mathbf{z}$ is $[0, 2i]$.

6 Implications

In this section, we consider the performance of NSGA-II [13] on five related problem instances with varying levels of complexity (see Table 5). I1 is separable and hence the simplest problem. I2 adds complexity by making position-related parameters depend on distance-related parameters (and other position-related parameters), whereas I3 makes distance-related parameters depend on position-related parameters (and other distance-related parameters). I4 instead employs a non-separable reduction. I5 combines the difficulties of both I3 and I4. I1–I5 all have concave Pareto optimal front, and are all uni-modal.

Table 5. Test problems I1–I5. To enhance readability, for any transition vector $\mathbf{t}^i$, we let $\mathbf{y} = \mathbf{t}^{i-1}$. For $\mathbf{t}^1$, let $\mathbf{y} = \mathbf{z}_{[0,1]} = \{z_1/2, \ldots, z_n/(2n)\}$.

Problem	Type	Setting
I1	Constants	$S_{1:M} = 1$
		$A_{1:M-1} = 1$
	Domains	$z_{i=1:n,\max} = 1$
	Shape	$h_{m=1:M} = \text{concave}_m$
	$\mathbf{t}^1$	$t^1_{i=1:n} = y_i$
	$\mathbf{t}^2$	$t^2_{i=1:k} = y_i$
		$t^2_{i=k+1:n} = \text{s\_linear}(y_i, 0.35)$
	$\mathbf{t}^3$	$t^3_{i=1:M-1} = \text{r\_sum}(\{y_{(i-1)k/(M-1)+1}, \ldots, y_{ik/(M-1)}\}, \{1, \ldots, 1\})$
		$t^3_M = \text{r\_sum}(\{y_{k+1}, \ldots, y_n\}, \{1, \ldots, 1\})$
I2	As I1, except the following replaces $\mathbf{t}^1$:	
	$\mathbf{t}^1$	$t^1_{i=1:n-1} = \text{b\_param}(y_i, \text{r\_sum}(\{y_{i+1}, \ldots, y_n\}, \{1, \ldots, 1\}), \frac{0.98}{49.98}, 0.02, 50)$
		$t^1_n = y_n$
I3	As I1, except the following replaces $\mathbf{t}^1$:	
	$\mathbf{t}^1$	$t^1_1 = y_1$
		$t^1_{i=2:n} = \text{b\_param}(y_i, \text{r\_sum}(\{y_1, \ldots, y_{i-1}\}, \{1, \ldots, 1\}), \frac{0.98}{49.98}, 0.02, 50)$
I4	As I1, except the following replaces $\mathbf{t}^3$:	
	$\mathbf{t}^3$	$t^3_{i=1:M-1} = \text{r\_nonsep}(\{y_{(i-1)k/(M-1)+1}, \ldots, y_{ik/(M-1)}\}, k/(M-1))$
		$t^3_M = \text{r\_nonsep}(\{y_{k+1}, \ldots, y_n\}, l)$
I5	As I1, except use $\mathbf{t}^1$ from I3, and $\mathbf{t}^3$ from I4.	

To facilitate analysis, I1–I5 are tested only for two objectives, with $k = 1$ and $l = 10$. The following (un-optimised) NSGA-II settings were used: population

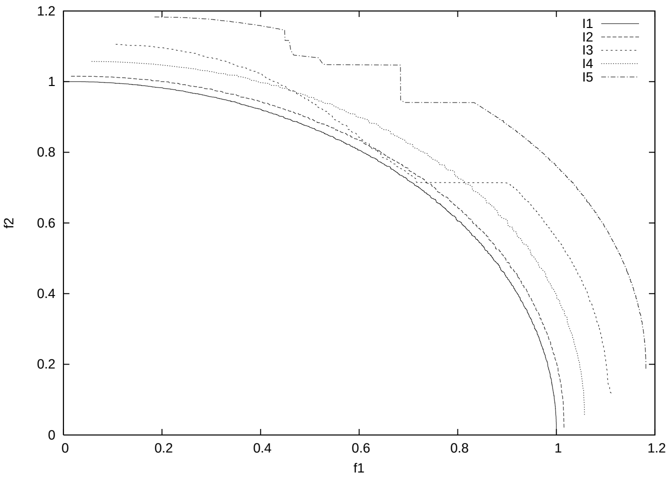


Fig. 4. The 50% attainment surfaces obtained by NSGA-II on I1–I5.

size 100, 250 generations, crossover probability 0.9, real parameter mutation probability $\frac{1}{n}$, SBX parameter 10, and mutation parameter 50. NSGA-II was run 35 times on each of I1–I5. The 50% attainment surfaces [14] (the "median" non-dominated fronts) are plotted in Fig. 4.

These are the first results we know of for NSGA-II on scalable, non-separable problems with known Pareto optimal sets, and clearly show the effects different types of problem complexity can have. For I1, NSGA-II effectively finds the Pareto optimal front, whereas having distance parameters depend on position parameters, as with I3 and I5, causes much difficulty. Moreover, I1–I5 are all uni-modal, and as the tests are only in two-objectives, more challenging problem instances are easily envisaged — NSGA-II can clearly be challenged by a variety of problem characteristics.

7 A Suggested Test Suite

In this section, we propose a test suite that consists of nine scalable, multi-objective test problems (WFG1–WFG9) that focuses on some of the more pertinent problem characteristics. Table 6 specifies WFG1–WFG9, the properties of which are summarised in Table 7.

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 uni-modal, the non-separable reduction of WFG6 and WFG9 is more difficult than that of WFG2 and WFG3, the multi-modality 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

Table 6. 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 non-separable reductions). To enhance readability, for any transition vector $\mathbf{t}^i$, we let $\mathbf{y} = \mathbf{t}^{i-1}$. For $\mathbf{t}^1$, let $\mathbf{y} = \mathbf{z}_{[0,1]} = \{z_1/2, \ldots, z_n/(2n)\}$.

Problem	Type	Setting
All	Constants	$S_{m=1:M} = 2m$
		$A_1 = 1$
		$A_{2:M-1} = \begin{cases} 0, \text{ for WFG3} \\ 1, \text{ otherwise} \end{cases}$
		The settings for $S_{1:M}$ ensures the Pareto optimal fronts have dissimilar trade-off 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	$h_{m=1:M-1} = \mathrm{convex}_m$
		$h_M = \mathrm{mixed}_M$ (with $\alpha = 1$ and $A = 5$)
	$\mathbf{t}^1$	$t^1_{i=1:k} = y_i$
		$t^1_{i=k+1:n} = \mathrm{s\_linear}(y_i, 0.35)$
	$\mathbf{t}^2$	$t^2_{i=1:k} = y_i$
		$t^2_{i=k+1:n} = \mathrm{b\_flat}(y_i, 0.8, 0.75, 0.85)$
	$\mathbf{t}^3$	$t^3_{i=1:n} = \mathrm{b\_poly}(y_i, 0.02)$
	$\mathbf{t}^4$	$t^4_{i=1:M-1} = \mathrm{r\_sum}(\{y_{(i-1)k/(M-1)+1}, \ldots, y_{ik/(M-1)}\}, \{2(i-1)k/(M-1)+1, \ldots, 2ik/(M-1)\})$
		$t^4_M = \mathrm{r\_sum}(\{y_{k+1}, \ldots, y_n\}, \{2(k+1), \ldots, 2n\})$
WFG2	Shape	$h_{m=1:M-1} = \mathrm{convex}_m$
		$h_M = \mathrm{disc}_M$ (with $\alpha = \beta = 1$ and $A = 5$)
	$\mathbf{t}^1$	As $\mathbf{t}^1$ from WFG1. (Linear shift.)
	$\mathbf{t}^2$	$t^2_{i=1:k} = y_i$
		$t^2_{i=k+1:k+l/2} = \mathrm{r\_nonsep}(\{y_{k+2(i-k)-1}, y_{k+2(i-k)}\}, 2)$
	$\mathbf{t}^3$	$t^3_{i=1:M-1} = \mathrm{r\_sum}(\{y_{(i-1)k/(M-1)+1}, \ldots, y_{ik/(M-1)}\}, \{1, \ldots, 1\})$
		$t^3_M = \mathrm{r\_sum}(\{y_{k+1}, \ldots, y_{k+l/2}\}, \{1, \ldots, 1\})$
WFG3	Shape	$h_{m=1:M} = \mathrm{linear}_m$ (degenerate)
	$\mathbf{t}^{1:3}$	As $\mathbf{t}^{1:3}$ from WFG2. (Linear shift, non-separable reduction, and weighted sum reduction.)
WFG4	Shape	$h_{m=1:M} = \mathrm{concave}_m$
	$\mathbf{t}^1$	$t^1_{i=1:n} = \mathrm{s\_multi}(y_i, 30, 10, 0.35)$
	$\mathbf{t}^2$	$t^2_{i=1:M-1} = \mathrm{r\_sum}(\{y_{(i-1)k/(M-1)+1}, \ldots, y_{ik/(M-1)}\}, \{1, \ldots, 1\})$
		$t^2_M = \mathrm{r\_sum}(\{y_{k+1}, \ldots, y_n\}, \{1, \ldots, 1\})$
WFG5	Shape	$h_{m=1:M} = \mathrm{concave}_m$
	$\mathbf{t}^1$	$t^1_{i=1:n} = \mathrm{s\_decept}(y_i, 0.35, 0.001, 0.05)$
	$\mathbf{t}^2$	As $\mathbf{t}^2$ from WFG4. (Weighted sum reduction.)
WFG6	Shape	$h_{m=1:M} = \mathrm{concave}_m$
	$\mathbf{t}^1$	As $\mathbf{t}^1$ from WFG1. (Linear shift.)
	$\mathbf{t}^2$	$t^2_{i=1:M-1} = \mathrm{r\_nonsep}(\{y_{(i-1)k/(M-1)+1}, \ldots, y_{ik/(M-1)}\}, k/(M-1))$
		$t^2_M = \mathrm{r\_nonsep}(\{y_{k+1}, \ldots, y_n\}, l)$
WFG7	Shape	$h_{m=1:M} = \mathrm{concave}_m$
	$\mathbf{t}^1$	$t^1_{i=1:k} = \mathrm{b\_param}(y_i, \mathrm{r\_sum}(\{y_{i+1}, \ldots, y_n\}, \{1, \ldots, 1\}), \frac{0.98}{49.98}, 0.02, 50)$
		$t^1_{i=k+1:n} = y_i$
	$\mathbf{t}^2$	As $\mathbf{t}^1$ from WFG1. (Linear shift.)
	$\mathbf{t}^3$	As $\mathbf{t}^2$ from WFG4. (Weighted sum reduction.)
WFG8	Shape	$h_{m=1:M} = \mathrm{concave}_m$
	$\mathbf{t}^1$	$t^1_{i=1:k} = y_i$
		$t^1_{i=k+1:n} = \mathrm{b\_param}(y_i, \mathrm{r\_sum}(\{y_1, \ldots, y_{i-1}\}, \{1, \ldots, 1\}), \frac{0.98}{49.98}, 0.02, 50)$
	$\mathbf{t}^2$	As $\mathbf{t}^1$ from WFG1. (Linear shift.)
	$\mathbf{t}^3$	As $\mathbf{t}^2$ from WFG4. (Weighted sum reduction.)
WFG9	As the example in Section 5.	

Table 7. 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	$f_{1:M}$	separable	uni	polynomial,flat	convex, mixed
WFG2	$f_{1:M-1}$	non-separable	uni	–	convex, disconnected
	f_M	non-separable	multi		
WFG3	$f_{1:M}$	non-separable	uni	–	linear, degenerate
WFG4	$f_{1:M}$	separable	multi	–	concave
WFG5	$f_{1:M}$	separable	deceptive	–	concave
WFG6	$f_{1:M}$	non-separable	uni	–	concave
WFG7	$f_{1:M}$	separable	uni	parameter dependent	concave
WFG8	$f_{1:M}$	non-separable	uni	parameter dependent	concave
WFG9	$f_{1:M}$	non-separable	multi,deceptive	parameter dependent	concave

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 non-separable, and the predominance of concave Pareto optimal fronts facilitates the use of performance metrics that require knowledge of the distance to the Pareto optimal front.

For WFG1–WFG7, a solution is Pareto optimal iff $z_{k+1} = \ldots = z_n = 0.35$, noting WFG2 is disconnected. For WFG8, it is required that all of:

$$
\begin{aligned}
z_{i=k+1:n} &= 0.35^{\left(0.02+49.98\left(\frac{0.98}{49.98}-(1-2u)\left|\lfloor 0.5-u\rfloor+\frac{0.98}{49.98}\right|\right)\right)^{0.5}} \\
u &= \text{r\_sum}(\{z_1,\ldots,z_{i-1}\},\{1,\ldots,1\})
\end{aligned}
$$

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.

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 DTLZ test suite. These include: non-separable problems, deceptive problems, a truly degenerative problem, a mixed shape Pareto front problem, problems scalable in the number of position-related parameters[8], and problems with dependencies between position- and distance-related parameters. The WFG test suite provides a fairer means of assessing the true performance of optimisation algorithms on a wider range of different problems.

8 Conclusions

The WFG Toolkit offers a substantial range of features. Test problem designers can construct problems with a diverse range of Pareto optimal geometries and

---

[8] The DTLZ test suite uses a fixed (relative to the number of objectives) number of position parameters.

can incorporate a variety of important features in the manner of their choosing. A suite of nine test problems is presented that exceeds the functionality of existing test suites. Significantly, the WFG Toolkit allows for the construction of scalable problems that are both non-separable and multi-modal. Given the relevance of both characteristics to real world problems and the corresponding lack of such problems in the literature, the WFG Toolkit offers an important contribution in assessing the quality of optimisation algorithms on these types of problems.

Acknowledgments

This work was partly supported by an Australian Research Council linkage grant.

References

1. Deb, K.: Multi-objective genetic algorithms: Problem difficulties and construction of test problems. Evolutionary Computation **7** (1999) 205–230
2. Van Veldhuizen, D.A.: Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations. PhD thesis, Air Force Institute of Technology, Wright-Patterson AFB, Ohio (1999)
3. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation **8** (2000) 173–195
4. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multi-objective optimization. KanGAL Report 2001001, Kanpur Genetic Algorithms Laboratory, Indian Institute of Technology, Kanpur, India (2001)
5. Whitley, D., Mathias, K., Rana, S., Dzubera, J.: Building better test functions. 6th International Conference on Genetic Algorithms, Morgan Kaufmann Publishers (1995) 239–246
6. Bäck, T., Michalewicz, Z.: Test landscapes. Handbook of Evolutionary Computation. Institute of Physics Publishing (1997) B2.7 14–20
7. Fogel, D.B., Beyer, H.G.: A note on the empirical evaluation of intermediate recombination. Evolutionary Computation **3** (1995) 491–495
8. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. CEC'02. Volume 1., IEEE (2002) 825–830
9. Van Veldhuizen, D.A., Lamont, G.B.: Multiobjective evolutionary algorithm test suites. 1999 ACM Symposium on Applied Computing, ACM (1999) 351–357
10. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. EUROGEN 2001, CIMNE, Barcelona, Spain (2001) 95–100
11. Bentley, P.J., Wakefield, J.P.: Finding acceptable solutions in the pareto-optimal range using multiobjective genetic algorithms. Soft Computing in Engineering Design and Manufacturing, Springer-Verlag (1998) 231–240
12. Knowles, J.D., Corne, D.W.: Approximating the nondominated front using the Pareto archived evolution strategy. Evolutionary Computation **8** (2000) 149–172
13. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation **6** (2002) 182–197
14. Fonseca, C.M., Fleming, P.J.: On the performance assessment and comparison of stochastic multiobjective optimizers. Parallel Problem Solving from Nature — PPSN IV, Springer-Verlag (1996) 584–593