On Test Functions for Evolutionary Multi-Objective Optimization

Tatsuya Okabe, Yaochu Jin, Markus Olhofer and Bernhard Sendhoff

Honda Research Institute Europe GmbH
Carl-Legien-Strasse 30, 63073 Offenbach/Main, Germany

Abstract. In order to evaluate the relative performance of optimization algorithms benchmark problems are frequently used. In the case of multi-objective optimization (MOO), we will show in this paper that most known benchmark problems belong to a constrained class of functions with piecewise linear Pareto fronts in the parameter space. We present a straightforward way to define benchmark problems with an arbitrary Pareto front both in the fitness and parameter spaces. Furthermore, we introduce a difficulty measure based on the mapping of probability density functions from parameter to fitness space. Finally, we evaluate two MOO algorithms for new benchmark problems.

1 Introduction

In the recent literature, several multi-objective evolutionary optimization algorithms have been proposed [1, 3]. In order to compare their performance and in order for practitioners to decide which algorithm to employ, benchmark problems (test functions) are extensively used. Therefore, the properties of test functions and their difficulty and representativeness are of utmost importance. Although some work on summarizing the properties of test functions for multi-objective optimization can be found in the literature [1], we will see in the next section that a majority of all test functions share the same property: their Pareto front in parameter space is piecewise linear. This restriction of benchmarks to a limited class of functions is dangerous since we cannot expect that most real-world problems will also have piecewise linear Pareto fronts in parameter space. It is the target of this paper to present an "easy to follow" recipe to construct test functions for multi-objective optimization problems with arbitrary, user-specified Pareto curves in the parameter space *and* fitness space. Furthermore, in Section 4, we will present a measure for the difficulty of benchmark problems based on the mapping between parameter space and fitness space[1]. In Section 5, examples of generated test functions will be given. We will compare two multi-objective evolutionary algorithms for these new test problems in Section 6 and summarize this paper in the last section.

[1] Note that the fitness space is the space spanned by the number of objectives in MOO. It should not be confused with the fitness landscape, which in single objective optimization is sometimes also referred to as fitness space.

2 Properties of Test Functions

Coello et al. [1] have summarized test functions for MOO according to their properties in the fitness space (FS) and in the parameter space (PS). However the geometry of the Pareto front in parameter space and in particular the mapping between the PS and the FS has not received much attention. We investigated the geometry of the Pareto front in the PS of the test functions found in [1, 3]. The results are shown in Table 1 for the two-dimensional case. It is noticed that the Pareto fronts of most test functions consists of piecewise linear curves and/or single points in the PS.

Table 1. Geometry of Pareto fronts in the parameter space. The properties of test functions with "*" are obtained empirically.

Geometry	Test Functions	Geometry	Test Functions
1 Point	Binh(3), Osyczka	4 Lines	Schaffer(2),ZDT3
31 Points	ZDT5	5 Lines	Osyczka(2)
1 Line	Binh(1),Binh(2),Fonseca,	1 Point and 3 Lines	Kursawe*
	Fonseca(2),Laumanns,Lis,	1 Point and 4 Lines	Poloni*,Viennet(3)*
	Murata,Rendon(2),Schaffer,	1 Curve	Binh(4),Obayashi
	Belegundu,Kita,Srinivas,	3 Curves	Tanaka
	ZDT1,ZDT2,ZDT4,ZDT6	1 Surface	Viennet*,Tamaki
2 Lines	Jimenez	1 Surface and 1 Line	Viennet(4)*
3 Lines	Rendon,Viennet(2)*	6 Parts of Surface	Quaglizrella

In [2, 3], Deb has proposed the following method to construct "tunable" test functions for MOO:

$$\min\left(f_1(\boldsymbol{x}), f_2(\boldsymbol{x})\right) = \left(f_1(x_1, x_2, ..., x_m), g(x_{m+1}, x_{m+2}, ..., x_n) \cdot h(f_1, g)\right). \quad (1)$$

Here, $f_i, i = 1, 2$ and $x_j, j = 1, \ldots, n$ are the i-th objective function, and the j-th component of the parameter vector with dimension n. Functions g and h are defined by the user. In Equation (1), function h determines the shape of the Pareto front in the fitness space, function g controls the difficulty for an algorithm to converge to the Pareto front, and function f_1 influences the diversity of the Pareto optimal solutions. By applying deceptive or multi-modal test functions from single objective optimization (SOO) to g, one can realize the same difficulty, i.e. deceptiveness or multi-modality, in MOO. Therefore, it is possible to determine the shape of the Pareto front in FS and to tune the difficulty with respect to "standard" measures of difficulty from single objective optimization. However, we can decompose the optimization problem resulting from such test functions in the following way: (1) minimize the g function; (2) generate proper f_1 values to get widely spread and well-distributed solutions. Since all values of f_1 locate on the boundary of the feasible region and might be Pareto optimal solutions, it is sufficient to generate well-distributed f_1 values to get well-distributed Pareto-optimal solutions. The mapping between the parameter and the fitness spaces, and the shape of the Pareto front in parameter space have not been taken into account in the framework outlined above. This might be the reason for the similarity of the class of test functions created using the framework.

Although many papers in the literature concentrate on bi-objective optimization and/or optimization without constraints, most of real-world problems have more than two objectives and a lot of constraints. Three different approaches for

generating test functions with more than two objectives in [4, 6] and the one for constrained test functions in [5] are proposed. These are frontier work for more practical test functions.

3 A New Method to Build Test Functions

This section proposes a method to construct different types of test functions with arbitrary, customized Pareto fronts in fitness *and* in parameter space. The basic idea is to start from a starting space between a parameter space and a fitness space[2], which we denote by $\mathcal{S}^2$, and to construct the parameter space and the fitness by applying appropriate functions to $\mathcal{S}^2$. Using the inverse of the generation operation, i.e. deformation, rotation and shift, for the parameter space, we arrive at the mapping from PS to FS. The basic procedure is outlined in Figure 1 and will be described in detail in the next sections.

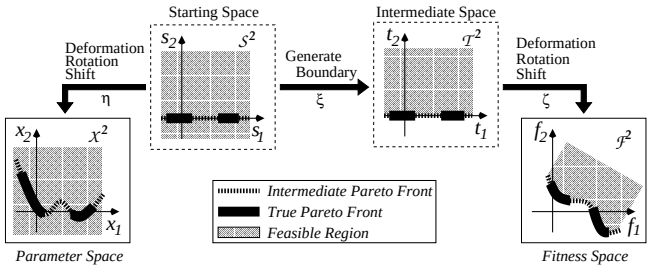


Fig. 1. Basic idea to generate test functions with arbitrary Pareto fronts in the PS and the FS. The space $\mathcal{S}^2$ is the starting space. The final result is given by the Pareto fronts in the PS $\mathcal{X}^2$ and the FS $\mathcal{F}^2$ and the mapping from PS to FS.

3.1 Determine Intermediate Pareto Front in the PS

The space $\mathcal{S}^2(s_1, s_2)$ with the Pareto front $s_2 = 0$ is used as the starting point. The mapping between $\mathcal{S}^2$ and the parameter space $\mathcal{X}^2(x_1, x_2)$ is given by

$$x_1 = \eta_1(s_1, s_2), \quad x_2 = \eta_2(s_1, s_2). \tag{2}$$

The following equations correspond to the intermediate Pareto front in PS

$$x_1 = \eta_1(s_1, s_2 = 0), \quad x_2 = \eta_2(s_1, s_2 = 0). \tag{3}$$

Since a true Pareto front will be determined later, we added the term *intermediate*. If the defining space for η_1^{-1} is denoted as $\mathcal{X}^2_{\eta_1^{-1}}$, the functional relation for the Pareto front in the parameter space is given by

$$x_2 = \eta_2(\eta_1^{-1}(x_1, s_2 = 0), s_2 = 0), \quad x_1 \in \mathcal{X}^2_{\eta_1^{-1}} \tag{4}$$

Therefore, functions η_1 and η_2 define the Pareto front in parameter space. Additionally, some constraints are introduced if $\mathcal{X}^2_{\eta_1^{-1}} \neq I\!R^2$, e.g. if $\eta_1 = s_1^2$, the constraint is given by $x_1 \geq 0$. However, besides defining the (intermediate) Pareto front in parameter space, η_1 also has a more hidden property. It relates

[2] Note that we will restrict the discussion in the following to two dimensions, however, the identical approach can be used to construct test functions in higher dimensions (both of a parameter space and a fitness space).

the probability density (PD) in $\mathcal{S}^2$ space to the probability density in $\mathcal{X}^2$ space. This is an important aspect because the search process in multi-objective evolutionary optimization can be understood by mapping the search distribution in parameter space to the one in fitness space [8]. Two examples of the mapping of a uniform distribution on $s_2 = 0$ in $\mathcal{S}^2$ space to the Pareto front in parameter space are shown in Figure 2 with $\eta_2 = \sin(s_1)$. Here, the points are generated with $s_1 = 0.1i - 4.0$ $(i = 0, 1, \cdots, 80)$. We refer to this property of η_1 as *deformation.*

In order to analyze the effect of the functions η_1 and η_2 on the whole $\mathcal{X}^2$ space (not just the Pareto front), we decompose them into functions that depend on s_1 or s_2 only[3].

$$x_1 = \eta_1(s_1, s_2) \equiv g_1(s_1) + g_2(s_2), \quad x_2 = \eta_2(s_1, s_2) \equiv g_3(s_1) + g_4(s_2). \tag{5}$$

To simplify our discussion, $g_2(s_2) = 0$ will be used for the rest of this paper. If we assume $g_4(s_2) = 0, \forall s_2 = 0$, then our comments above on the role of η_1 equally apply to $g_3(s_1)$. Thus, $g_3(s_1)$ determines the relationship of the probability density on the Pareto front from the $\mathcal{S}^2$ space to the $\mathcal{X}^2$. The role of $g_4(s_2)$ is similar, however, it controls the mapping of the probability density function *approaching* the Pareto front. Again in Figure 3 two examples are shown. Here, the points are generated with $s_1 = 0.1i - 4.0$ $(i = 0, 1, \cdots, 80)$, $s_2 = 0.25j - 1.0$ $(j = 0, 1, \cdots, 8)$. Needless to say that the function $g_4(s_2)$ is very important as a measure for the difficulty of the optimization problem. In the extreme case, the Pareto front might be an "island" surrounded by areas of extremely low probability (under the assumption of a uniform distribution in $\mathcal{S}^2$ space).

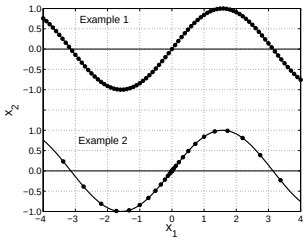


Fig. 2. The effect of η_1 function. Probability density of solutions along the Pareto front for $\eta_1(s_1, s_2 = 0) = s_1$ (top) and $\eta_1(s_1, s_2 = 0) = s_1^3$ (bottom). Data points (dots) are uniformly generated in $\mathcal{S}^2$ space.

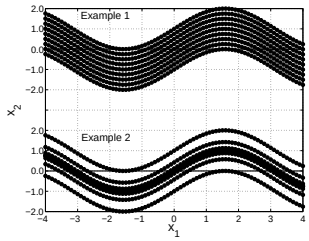


Fig. 3. The effect of g_4 function. Probability density for $g_4(s_2) = s_2$ (top) and $g_4(s_2) = s_2^3$ (bottom). Here, $\eta_1(s_1, s_2) = s_1$ and $g_3(s_1) = \sin(s_1)$ are assumed. Data points (dots) are uniformly generated in the $\mathcal{S}^2$ space.

In order to extend the possible degree of variation, the Pareto front in $\mathcal{X}^2$ space can additionally be *rotated* using a standard rotation matrix.

So far we have discussed the mapping from the starting space $\mathcal{S}^2$ to $\mathcal{X}^2$ space, because this is the direction for constructing the Pareto front. Of course the mapping direction during search is the opposite. The projected probability

[3] To simplify our discussion, we assume that η_1 and η_2 can be decomposed. However, this assumption is not necessary for generating test functions in practice.

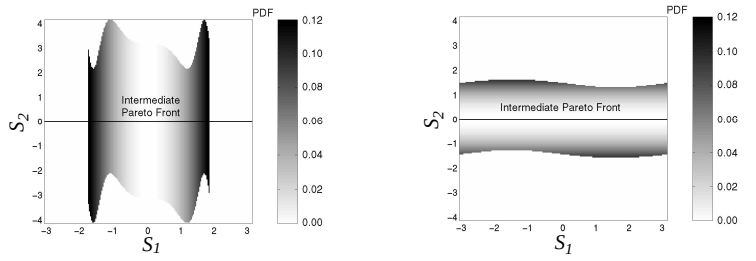
distributions for the mapping from $\mathcal{X}^2$ space to $\mathcal{S}^2$ space assuming uniform distribution of (x_1, x_2) in $[-\pi, \pi]^2$ is shown in Figure 4(a) for the example of

$$\eta_1(s_1, s_2) = s_1^3, \quad \eta_2(s_1, s_2) = \sin(s_1^3) + s_2, \tag{6}$$

and in Figure 4(b) for the example of

$$\eta_1(s_1, s_2) = s_1, \quad \eta_2(s_1, s_2) = \sin(s_1) + s_2^3. \tag{7}$$

In Figure 4(b), we can observe the "island effect" which we mentioned above.



(a) $(x_1, x_2) = (s_1^3, \sin(s_1^3) + s_2)$ (b) $(x_1, x_2) = (s_1, \sin(s_1) + s_2^3)$

Fig. 4. Projected distribution in the $\mathcal{S}^2$ space. The distribution in the PS (the $\mathcal{X}^2$ space), is assumed to be uniform in $(x_1, x_2) \in [-\pi, \pi]^2$.

3.2 Generate Boundary

The first step from the intermediate space $\mathcal{S}^2$ toward the fitness space is the generation of a boundary, since the Pareto front in fitness space usually consists of parts of a boundary.

There are two approaches to generating boundaries. First, the boundary can explicitly be defined by introducing appropriate constraints. Second, it can implicitly be defined by using functions that generate a boundary. We will concentrate on the second approach and define a second intermediate space $\mathcal{T}^2 = (t_1, t_2)$. The following functions will be used as "boundary generators":

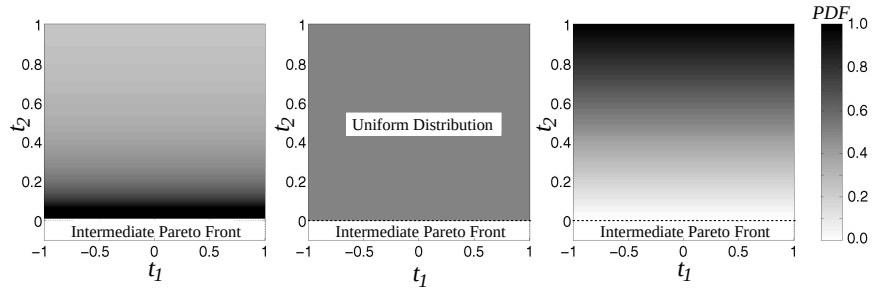
(1) $t_1 = \xi_1(s_1) = s_1$, $t_2 = \xi_2(s_2) = s_2^2$ (2) $t_1 = \xi_1(s_1) = s_1$, $t_2 = \xi_2(s_2) = |s_2|$ (3) $t_1 = \xi_1(s_1) = s_1$, $t_2 = \xi_2(s_2) = \sqrt{|s_2|}$.

Now, the distribution in the $\mathcal{T}^2$ space can be calculated theoretically [8]. The distribution in the $\mathcal{S}^2$ space is assumed to be uniform in $[-1, 1]^2$. The results are shown in Figure 5. As explained in [8], the sphere function results in the highest value of the probability density function (PDF) near the intermediate Pareto front. Theoretically, the PDF of case (1) at $t_2 = 0$ is infinity. For the absolute function, case (2), the distribution remains uniform. The square root function, case (3), generates difficult problems to reach the intermediate Pareto front.

3.3 Determine Intermediate and True Pareto Front in FS

To generate the shape of the intermediate Pareto front in the FS, the same procedure as in Section 3.1 can be used. In the $\mathcal{T}^2$ space, the boundary of $t_1 = 0$ is the candidate for the Pareto front in the FS. The relationship between the space $\mathcal{T}^2$ and the fitness space can be written as:

$$f_1 = \zeta_1(t_1, t_2), \quad f_2 = \zeta_2(t_1, t_2). \tag{8}$$



(a) $t_2 = s_2^2$ (b) $t_2 = |s_2|$ (c) $t_2 = \sqrt{|s_2|}$

Fig. 5. Projected Distribution in the $\mathcal{T}^2$ space. The distribution in the $\mathcal{S}^2$ space is assumed to be uniform in $[-1, 1]^2$.

It is worth noticing that the mapping direction from $\mathcal{T}^2$ to $\mathcal{F}^2$ is the same as for the optimization, i.e. from the PS to the FS. Thus, the discussion of the properties of ζ_1 and ζ_2 carry directly over to the discussion of the PS to FS mapping for the optimization.

After generating the intermediate Pareto front, the true Pareto front in the FS will be determined. According to the definition of Pareto optimality [1], only parts of the generated boundary will be the Pareto front. Here, we can also generate a disconnected Pareto front by selecting proper functions of ζ_1 and ζ_2. To know which parts are the Pareto front in the PS, the boundary information can be mapped back to the PS using the functions and their inverses outlined in the previous sections.

4 Distribution Indicator

In this section, we will discuss the difficulty of test functions from the viewpoint of the transformation of a given probability density under the mapping from the parameter space to the fitness space. For this purpose, the *Distribution Indicator* will be proposed.

Assume the distribution before projection to be $\phi(x_1, x_2)$ in the parameter space $\mathcal{X}^2$. Using the following equation, the projected distribution, $\psi(f_1, f_2)$, in the fitness space $\mathcal{F}^2$ can be calculated with the Jacobian matrix J as follows [8]:

$$\psi(f_1, f_2) = \frac{1}{|J|}\phi(x_1, x_2), \quad |J| = \frac{\partial f_1}{\partial x_1}\frac{\partial f_2}{\partial x_2} - \frac{\partial f_1}{\partial x_2}\frac{\partial f_2}{\partial x_1}. \tag{9}$$

Again equation (9) shows only the two-dimensional case with a one-to-one mapping between both spaces, however, both conditions can easily be relaxed. The necessary extension can be found in [8].

We are now in a position to define the distribution indicator as an additional measure for the difficulty of test problems for multi-objective optimization as follows:

$$\mathcal{D} = (\det J)^{-1} = |J|^{-1}. \tag{10}$$

The distribution indicator measures the amount of distortion the probability density in parameter space suffers under the mapping from PS to FS.

In the previous section, we constructed test functions using the intermediate spaces $\mathcal{S}^2$ and $\mathcal{T}^2$ and the additional rotation operation. The rotation results in no distortion and will therefore be neglected in the following. The remaining distribution indicators can simply be multiplied to yield the overall indicator:

$$\mathcal{D} \equiv \mathcal{D}_{x\to f} = \mathcal{D}_{x\to s} \times \mathcal{D}_{s\to t} \times \mathcal{D}_{t\to f} \tag{11}$$

The *Distribution Indicators* $\mathcal{D}_{x\to s}$ for the functions that we used in the previous sections are given as follows

$$\begin{aligned}
&(1)\ \ (x_1, x_2) = (s_1, \sin(s_1) + s_2) \ \rightarrow\ \mathcal{D}_{x\to s} = 1\\
&(2)\ \ (x_1, x_2) = (s_1^3, \sin(s_1^3) + s_2) \ \rightarrow\ \mathcal{D}_{x\to s} = 3s_1^2\\
&(3)\ \ (x_1, x_2) = (s_1, \sin(s_1) + s_2^3) \ \rightarrow\ \mathcal{D}_{x\to s} = 3s_2^2
\end{aligned}$$

Case 1 results in no changes in the probability density. For the second case, the distribution is uniform in the direction of s_2 but not in s_1. Since the *Distribution Indicator* becomes low near $s_1 = 0$, the probability density in this area becomes sparse whereas it is high for $s_1 >> 0$ and $s_1 << 0$. Case 3 is similar to case 2, however, the role of s_1 and s_2 is exchanged. These results which show good agreement with Figure 4 demonstrate that difficult MOO problems can be generated with a smaller *Distribution Indicator* close to the Pareto front.

The *Distribution Indicators* $\mathcal{D}_{s\to t}$ are given by

$$\begin{aligned}
&(1)\ \ (t_1, t_2) = (s_1, s_2^2) \ \rightarrow\ \mathcal{D}_{s\to t} = (2\sqrt{t_2})^{-1}\\
&(2)\ \ (t_1, t_2) = (s_1, |s_2|) \ \rightarrow\ \mathcal{D}_{s\to t} = 1\\
&(3)\ \ (t_1, t_2) = (s_1, \sqrt{|s_2|}) \ \rightarrow\ \mathcal{D}_{s\to t} = 2t_2
\end{aligned}$$

For case 1 the PDF becomes infinite on the intermediate Pareto front. Additionally, the area near the intermediate Pareto front has higher values of the PDF. This situation is the same as for SCH1 test function in [9]. Case 2 results in a uniform distribution due to $\mathcal{D} = 1$. In case 3, the distribution depends on $2t_2$. Thus, the area far from the intermediate Pareto front has high value. The minimum PDF is on the intermediate Pareto front.

The *Distribution Indicators* $\mathcal{D}_{t\to f}$ can be calculated in the same way with $\mathcal{D}_{x\to s}$ due to the same generation operation.

5 Illustrative Examples

With the framework presented in this paper, a variety of test functions can be generated. Due to the space limit, only two examples are shown here.

Test Function OKA1

Suppose that the desired Pareto front in the PS is defined by $x_2 = 3\cos(x_1) + 3$ with 15-degree rotation clockwise and the one in the FS by $f_2 = \sqrt{2\pi} - \sqrt{f_1}$. To control the hardness of the test function, the distribution in the f_1 direction is assumed to be uniform (see Example 1 in Figure 2) and the one in the f_2 direction is assumed to become more sparse towards the Pareto front, see Figure 4 (b). For this purpose, we can use the following mapping functions:

$$\begin{aligned}
&\eta_1(s_1, s_2) = \cos(\pi/12)s_1 + \sin(\pi/12)\left(3\cos(s_1) + 3 + s_2^3\right),\\
&\eta_2(s_1, s_2) = -\sin(\pi/12)s_1 + \cos(\pi/12)\left(3\cos(s_1) + 3 + s_2^3\right)\\
&\xi_1(s_1) = s_1, \quad \xi_2(s_2) = |s_2|, \quad \zeta_1(t_1, t_2) = t_1, \quad \zeta_2(t_1, t_2) = \sqrt{2\pi} - \sqrt{|t_1|} + 2t_2. \qquad (12)
\end{aligned}$$

Here, η_1 and η_2 define the Pareto front in the PS. To generate the boundary, the simple mapping functions $\xi_1 = s_1$ and $\xi_2 = |s_2|$ is used. Finally, mappings ζ_1 and ζ_2 determine the Pareto front in the the FS. In this way, the following MOO test function is created:

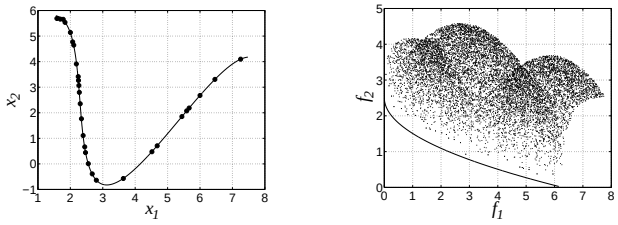
$$
\begin{aligned}
& f_1 = x_1', \quad f_2 = \sqrt{2\pi} - \sqrt{|x_1'|} + 2\left|x_2' - 3\cos(x_1') - 3\right|^{\frac{1}{3}}, \\
& x_1' = \cos(\pi/12)x_1 - \sin(\pi/12)x_2, \quad x_2' = \sin(\pi/12)x_1 + \cos(\pi/12)x_2, \\
& x_1 \in [6\sin(\pi/12), 6\sin(\pi/12) + 2\pi\cos(\pi/12)], \quad x_2 \in [-2\pi\sin(\pi/12), 6\cos(\pi/12)] \\
& \text{Pareto Front in the PS} \quad x_2' = 3\cos(x_1') + 3 \quad (x_1' \in [0, 2\pi]) \\
& \text{Pareto Front in the FS} \quad f_2 = \sqrt{2\pi} - \sqrt{f_1} \quad (f_1 \in [0, 2\pi]) \\
& \text{Distribution Indicator} \quad \mathcal{D}_{x \to f} = \frac{3}{2}\left|x_2' - 3\cos(x_1') - 3\right|^{\frac{2}{3}}.
\end{aligned} \tag{13}
$$

Test Function OKA2

Similarly, we can construct the following three-dimensional test function:

$$
\begin{aligned}
& f_1 = x_1, \quad f_2 = 1 - \frac{1}{4\pi^2}(x_1 + \pi)^2 + |x_2 - 5\cos(x_1)|^{\frac{1}{3}} + |x_3 - 5\sin(x_1)|^{\frac{1}{3}} \\
& x_1 \in [-\pi, \pi], \quad x_2, x_3 \in [-5, 5] \\
& \text{Pareto Front in the PS} \quad (x_1, x_2, x_3) = (\xi, 5\cos(\xi), 5\sin(\xi)) \quad (x_1 \in [-\pi, \pi]) \\
& \text{Pareto Front in the FS} \quad f_2 = 1 - \frac{1}{4\pi^2}(f_1 + \pi)^2 \quad (f_1 \in [-\pi, \pi]) \\
& \text{Distribution Indicator} \quad \mathcal{D}_{x \to f} = 9\,|x_2 - 5\cos(x_1)|^{\frac{2}{3}}\,|x_3 - 5\sin(x_1)|^{\frac{2}{3}}.
\end{aligned} \tag{14}
$$

To show the Pareto front, 10000 data points are generated randomly in the PS and projected into the FS, see Figure 6 and 7. In the FS, all data points are plotted to show the entire fitness landscape. In the PS, only non-dominated points are shown. The true Pareto front is denoted by the solid curves. It can be seen from Figure 6 and 7 that the designed test functions address our concerns discussed in Section 1 and 2.

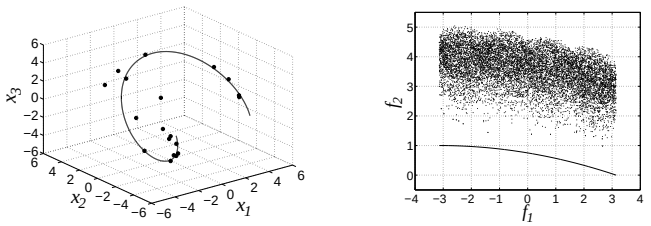


(a) Pareto Front in PS (b) Pareto Front in FS

Fig. 6. Pareto front (OKA1) in the parameter space (PS) and in the fitness space (FS).

6 Performance Comparison

In this section, the performance of NSGA-II (GA version) [6] and Hybrid Representation (HR) [9] will be compared on the two given test functions OKA1 and OKA2. Each algorithm was executed 30 times to reduce the randomness. The parameters used are summarized in Table 2 and the results are shown in Figure 8 (a)-(b) for NSGA-II, and (d)-(e) for HR, respectively. It can be seen that the performance of both optimizers on OKA1 and OKA2 is not sufficient.



(a) Pareto Front in PS (b) Pareto Front in FS

Fig. 7. Pareto front (OKA2) in the parameter space (PS) and in the fitness space (FS).

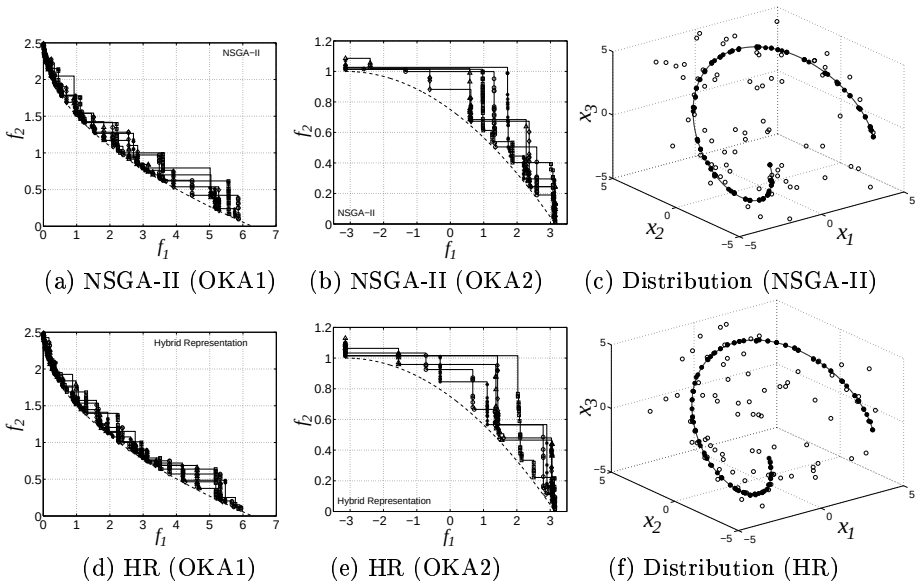
The main reason is that for both test functions, the closer the population gets to the Pareto front, the more sparse the probability density becomes. To illustrate this, we show the parent (denoted by dots) and offspring (denoted by circles) individuals at the final generation in Figure 8 (c) of NSGA-II and (f) of HR. Clearly, both optimizers cannot generate promising offspring even if parents have converged to the Pareto front (denoted by the curve). Using the new test functions, we show that although both optimizers can solve existing test functions of a low dimension perfectly, neither of them is able to solve the test functions proposed in this work satisfactorily even if the dimensionality is very low.

Table 2. Parameters used in the experiments.

	NSGA-II	HR
Population size	100	100
Maximum iterations	500	500
Coding	Gray coding	Gray coding
Crossover	One-point crossover	One-point crossover
Crossover rate	0.90	0.90
Mutation (GA)	Bit flip	Bit flip
Mutation rate (GA)	0.025	0.025
Number of bits per one design parameter	20	20
Initial step size (ES)	N/A	$\sigma_i = [0.0, 0.1]$
Lower bound	N/A	$\sigma_i \geq 0.004 \times \lvert x_i \rvert$
Minimum number of individuals	N/A	5%
Mutation rate (Switching)	N/A	0.01
Selection	Crowded Tournament	Crowded Tournament

7 Summary

Most test functions have piecewise linear Pareto fronts in parameter space. In order to circumvent this undesirable restriction, we proposed in this paper a method to construct benchmark problems for multi-objective optimization problems with arbitrary, customized Pareto fronts in fitness *and* parameter space. As a "by-product" of this method we suggested an additional measure for the difficulty of test problems which is based on the mapping of probability density functions from parameter to fitness space. We termed this measure *distribution indicator*. If the value of this indicator is small, the corresponding region is difficult to search for the optimizer. Finally, we gave examples of test functions with a higher order Pareto curve in parameter space and compared the performance of two evolutionary algorithms.



(a) NSGA-II (OKA1) (b) NSGA-II (OKA2) (c) Distribution (NSGA-II)

(d) HR (OKA1) (e) HR (OKA2) (f) Distribution (HR)

Fig. 8. Solutions obtained by NSGA-II (a)-(b) and HR (d)-(e) on OKA1 and OKA2. Only 5 results are shown to avoid complicated figures. Parents and offspring at the final generation of NSGA-II (c) and HR (f) on OKA2.

Acknowledgment

The authors would like to thank E. Körner and A. Richter for their kind support.

References

[1] Coello, C. A. C., Van Veldhuizen, D. A. and Lamont, G. B., *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer Academic Publishers, 2001.

[2] Deb, K., Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems, *Evolutionary Computation*, 7(3), pages 205–230, 1999.

[3] Deb, K., *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, LTD., 2001.

[4] Deb, K., Thiele, L., Laumanns, M. and Zitzler, E., Scalable Test Problems for Evolutionary Multi-Objective Optimization, KanGAL Report 2001001, 2001.

[5] Deb, K., Pratap, A. and Meyarivan, T., Constrained Test Problems for Multi-Objective Evolutionary Optimization, In *Evolutionary Multi-Criterion Optimization (EMO-2001)*, pages 284–298, 2001.

[6] Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T., A Fast and Elitist Multi-Objective Genetic Algorithms: NSGA-II, *IEEE Transactions on Evolutionary Computation*, 6(2), pages 182–197, 2002.

[7] Deb, K., Thiele, L., Laumanns, M. and Zitzler, E., Scalable Multi-Objective Optimization Test Problems, In *Proceedings of Congress on Evolutionary Computation (CEC-2002)*, pages 825–830, 2002.

[8] Okabe, T., Jin, Y. and Sendhoff, B., On the Dynamics of Evolutionary Multi-Objective Optimisation, *Proceedings of Genetic and Evolutionary Computation Conference (GECCO-2002)*, pages 247–255, 2002.

[9] Okabe, T., Jin, Y. and Sendhoff, B., Evolutionary Multi-Objective Optimisation with a Hybrid Representation, In *Proceedings of Congress on Evolutionary Computation (CEC-2003)*, pages 2262-2269, 2003.