Scalable Test Problems
for Evolutionary Multi-Objective Optimization

Kalyanmoy Deb
Kanpur Genetic Algorithms Laboratory
Indian Institute of Technology Kanpur
PIN 208 016, India
deb@iitk.ac.in
Lothar Thiele, Marco Laumanns and Eckart Zitzler
Computer Engineering and Networks Laboratory
ETH Zürich
CH-8092, Switzerland
{thiele,laumanns,zitzler}@tik.ee.ethz.ch

TIK-Technical Report No. 112
Institut für Technische Informatik und Kommunikationsnetze, ETH Zürich
Gloriastrasse 35., ETH-Zentrum, CH-8092, Zürich, Switzerland

July 17, 2001

Abstract

After adequately demonstrating the ability to solve different two-objective optimization problems, multi-objective evolutionary algorithms (MOEAs) must now show their efficacy in handling problems having more than two objectives. In this paper, we have suggested three different approaches for systematically designing test problems for this purpose. The simplicity of construction, scalability to any number of decision variables and objectives, knowledge of exact shape and location of the resulting Pareto-optimal front, and introduction of controlled difficulties in both converging to the true Pareto-optimal front and maintaining a widely distributed set of solutions are the main features of the suggested test problems. Because of the above features, they should be found useful in various research activities on MOEAs, such as testing the performance of a new MOEA, comparing different MOEAs, and better understanding of the working principles of MOEAs.

1 Introduction

Most earlier studies on multi-objective evolutionary algorithms (MOEAs) introduced test problems which were either simple or not scalable. Some test problems were too complicated to visualize the exact shape and location of the resulting Pareto-optimal front. Schaffer's (1984) study introduced two single-variable test problems (SCH1 and SCH2), which have been widely used as test problems. Kursawe's (1990) test problem KUR was scalable to any number of decision variables, but was not scalable in terms of the number of objectives. The same is true with Fonseca and Fleming's (1995) test problem FON. Poloni et al.'s (2000) test problem POL used only two decision variables. Although the mathematical formulation of the problem is non-linear, the resulting Pareto-optimal front corresponds an almost linear relationship among decision variables. Viennet's (1996) test problem VNT has a discrete set of Pareto-optimal fronts, but was designed for three objectives only. Similar simplicity prevails in the existing constrained test problems (Veldhuizen, 1999; Deb, 2001).

However, in 1999, the first author introduced a systematic procedure of designing test problems which are simple to construct and are scalable to the number of decision variables (Deb, 1999). In these problems, the exact shape and location of the Pareto-optimal solutions are also known. The basic construction used two functionals g and h^* with non-overlapping sets of decision variables to introduce difficulties towards the convergence to the true Pareto-optimal front and to introduce difficulties along the Pareto-optimal front for an MOEA to find a widely distributed set of solutions, respectively. Although these test problems are used by many researchers since then, they have been somewhat criticized for the relative independence feature of the functionals in achieving both the tasks. Such critics have grossly overlooked an important aspect of that study. The non-overlapping property of the two key functionals in the test problems was introduced for an ease of the construction procedure. That study also suggested the use of a procedure to map the variable vector (say $\mathbf{y}$) to a different decision variable vector (say $\mathbf{x}$). This way, although test problems are constructed for two non-overlapping sets from $\mathbf{y}$, each decision variable x_i (on which MOEA operators work) involves a correlation of all (or many) variables of $\mathbf{y}$. Such a mapping couples both aspects of convergence and maintenance of diversity and makes the problem harder to solve. However, Zitzler, Deb, and Thiele (2000) showed that the uncorrelated version of the test problems was even difficult to solve exactly using the then-known state-of-the-art MOEAs.

In the recent past, many MOEAs have adequately demonstrated their ability to solve two-objective optimization problems. With the suggestion of a number of such MOEAs, it is time that they must be investigated for their ability to solve problems with more than two objectives. In order to help achieve such studies, it is therefore necessary to develop scalable test problems for higher number of objectives. Besides testing an MOEA's ability to solve problems with a large number of objectives, the proposed test problems can also be used for systematically comparing two or more MOEAs. Since one such test problem can be used to test a particular aspect of multi-objective optimization, such as for convergence to the true Pareto-optimal front or maintenance of a good spread of solutions, etc., the test problems can be used to identify MOEAs which are better in terms of that particular aspect. For these reasons, these test problems may help provide a better understanding of the working principles of MOEAs, thereby allowing a user to develop better and more efficient MOEAs.

In the remainder of the paper, we first describe the essential features needed in a test problem and then suggest three approaches for systematically designing test problems for multi-objective optimization algorithms. Although most problems are illustrated for three objectives (for an ease of illustration), the test problems are generic and scalable to an arbitrary number of objectives.

2 Desired Features of Test Problems

Based on the above discussion, we suggest that the following features must be present in a test problem suite for adequately testing an MOEA:

1. Test problems should be easy to construct.
2. Test problems should be scalable to have any number of decision variables.
3. Test problems should be scalable to have any number of objectives.
4. The resulting Pareto-optimal front (continuous or discrete) must be easy to comprehend, and its exact shape and location should be exactly known. The corresponding decision variable values should also be easy to find.

5. Test problems should introduce controllable hindrance to converge to the true Pareto-optimal front and also to find a widely distributed set of Pareto-optimal solutions.

The popularity of two-objective test problems suggested earlier (Deb, 1999) in many research studies is partly because of the ease of constructing the test problems and the ease of illustrating the obtained set of Pareto-optimal solutions in two dimensions. Visually comparing the obtained set of solutions against the true Pareto-optimal front provides a clear idea of the performance of an MOEA. This can somewhat be achieved even for three objectives, with such a comparison shown in three dimensions. But for problems with more than three objectives, it becomes difficult to illustrate such a plot. Thus, for higher-objective test problems, it may be wise to have some regularity in the search space so that the Pareto-optimal surface is easily comprehensible. One of the ways to achieve this would be to have a Pareto-optimal surface symmetric along interesting hyper-planes, such as $f_1 = f_2 = \cdots = f_{M-1}$ (where M is the number of objectives). This only requires a user to comprehend the interaction between f_M and f_1, and the rest of the problem can be constructed by using symmetry. Another interesting approach would be to construct a problem for which the Pareto-optimal surface is a symmetric curve or at most a three-dimensional surface. Although M-dimensional, the obtained solutions can be easily illustrated parametrically in a two-dimensional plot in the case of a curve and in a three-dimensional plot in the case of the three-dimensional surface.

It is now well established that MOEAs have two tasks to achieve: converging to the Pareto-optimal front and finding a good distribution of solutions on the entire Pareto-optimal front. An MOEA can, therefore, must be tested for each of the two tasks. Thus, some test problems should test an MOEA's ability to negotiate artificial hurdles which hinders its progress towards converging to the true Pareto-optimal front. This can be achieved by placing some local Pareto-optimal attractors or biased density of solutions away from the Pareto-optimal front. Some test problems must also test an MOEA's ability to find a diverse set of solutions. This can be achieved by making the Pareto-optimal front nonconvex, discrete, and having variable density of solutions along the front. Although these features of test problems were also suggested for two-objective problems earlier (Deb, 1999), they are also applicable for a large number of objectives. Moreover, the increased dimensionality associated with a large number of objectives may cause an added difficulty to MOEAs.

In the following sections, we suggest different approaches of designing test problems for multi-objective optimization.

3 Different Methods of Test Problem Design

Based on the above principles, there exist a number of different ways to systematically design test problems for multi-objective optimization. Here we discuss three different methods:

1. Multiple single-objective functions approach,

2. Bottom-up approach,

3. Constraint surface approach.

The first approach is the most intuitive one and has been implicitly used by early MOEA researchers to construct test problems. In this approach, M different single-objective functions are used to construct a multi-objective test problem. To simplify the construction procedure, in many cases, different objective functions are simply used as different translations of a single objective function. For example, the problem SCH1 uses the following two single-objective functions for minimization (Schaffer, 1984):

$$f_1(x) = x^2, \quad f_2(x) = (x-2)^2.$$

Since the optimum $x^{*(1)}$ for f_1 is not the optimum for f_2 and vice versa, the Pareto-optimal set consists of more than one solution, including the individual minimum of each of the above functions. All other solutions which make trade-offs between the two objective functions with themselves and with the above two solutions become members of the Pareto-optimal set. In the above problem, all solutions $x^* \in [0, 2]$ become member of the Pareto-optimal set. Similarly, the problem FON shown below

$$\begin{cases} \text{Minimize} & f_1(\mathbf{x}) = 1 - \exp\left(-\sum_{i=1}^{n}(x_i - \frac{1}{\sqrt{n}})^2\right), \\ \text{Minimize} & f_2(\mathbf{x}) = 1 - \exp\left(-\sum_{i=1}^{n}(x_i + \frac{1}{\sqrt{n}})^2\right), \\ & -4 \leq x_i \leq 4 \quad i = 1, 2, \ldots, n. \end{cases} \tag{1}$$

has $x_i^* = -1/\sqrt{n}$ for all i as the minimum solution for f_1 and $x_i^* = 1/\sqrt{n}$ for all i as the minimum solution for f_2. The Pareto-optimal set is constituted with all solutions in $x_i^* \in [-1/\sqrt{n}, 1/\sqrt{n}]$ for all i. Veldhuizen (1999) lists a number of such test problems. It is interesting to note that such a construction procedure can be extended to higher-objective problems as well (Laumanns, Rudolph, and Schwefel, 2001). In a systematic procedure, each optimum may be assumed to lie on each of M (usually $< n$) coordinate directions. However, the Pareto-optimal set resulting from such a construction depends on the chosen objective functions, thereby making it difficult to comprehend the true nature of the Pareto-optimal front. Moreover, even in simple objective functions (such as in SCH2 (Schaffer, 1984)), the Pareto-optimal front may be a combination of disconnected fronts. Thus, a test problem constructed using this procedure must be carefully analyzed to find the true Pareto-optimal set of solutions.

The latter two approaches of test problem design mentioned above directly involve the Pareto-optimal front, thereby making them convenient to be used in practice. Since they require detailed discussions, we devote two separate sections for describing them.

4 Bottom-Up Approach

In this approach, a mathematical function describing the Pareto-optimal front is assumed in the objective space and an overall objective search space is constructed from this front to define the test problem. For two objectives, one such construction was briefly suggested earlier (Deb, 1999) and was extended for higher objectives elsewhere (Deb, 2001). But, here we make the idea more clear by illustrating the principle in a three-objective example test problem. Later, we suggest a generic procedure.

Let us assume that we would like to have a Pareto-optimal front where all objective functions take non-negative values and the desired front is the first quadrant of a sphere of radius one (as shown in Figure 1). With the help of spherical coordinates (θ, γ, and $r = 1$), the front can be described as follows:

$$\left.\begin{array}{rcl} f_1(\theta, \gamma) & = & \cos\theta \cos(\gamma + \pi/4), \\ f_2(\theta, \gamma) & = & \cos\theta \sin(\gamma + \pi/4), \\ f_3(\theta, \gamma) & = & \sin\theta, \\ \text{where} & & 0 \leq \theta \leq \pi/2, \\ & & -\pi/4 \leq \gamma \leq \pi/4. \end{array}\right\} \tag{2}$$

It is clear from the construction of the above surface that if all three objective functions are minimized, any two points on this surface are non-dominated to each other. Now, if the rest of the objective search space is constructed above this surface, we shall have a problem where the unit sphere constitutes the Pareto-optimal front. A simple way to construct the rest of the search space is to construct surfaces parallel to the above surface. This means constructing spherical surfaces with radius greater than one. This can be achieved by multiplying the above three functions with a term, which takes a value greater than or equal to one. Different values of the

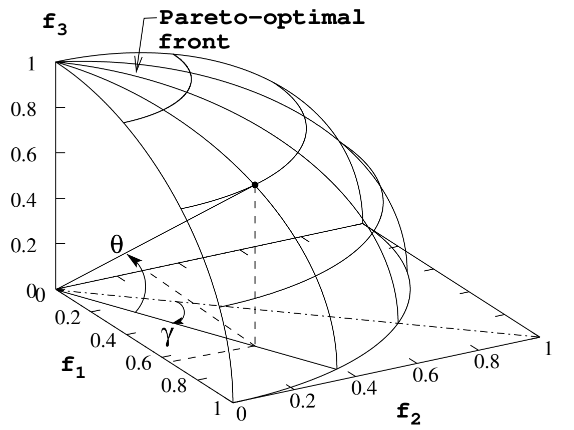


Figure 1: First quadrant of a unit sphere as a Pareto-optimal front.

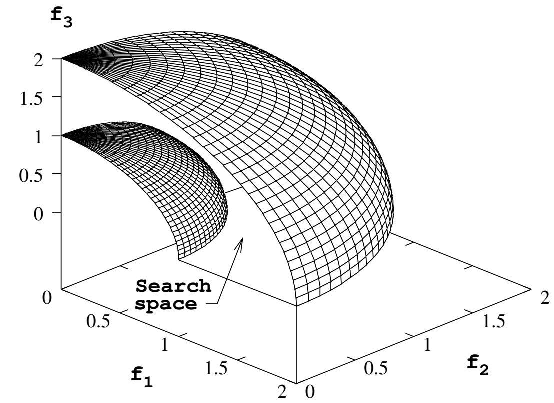


Figure 2: Overall search space is bounded by the two spheres.

third independent variable r (besides θ and γ) will construct different layers of spherical surfaces on top of the Pareto-optimal sphere. Thus, the overall problem with the above three variables is as follows:

$$\left.\begin{array}{ll}\text{Minimize} & f_1(\theta,\gamma,r) = (1+g(r))\cos\theta\cos(\gamma+\pi/4),\\ \text{Minimize} & f_2(\theta,\gamma,r) = (1+g(r))\cos\theta\sin(\gamma+\pi/4),\\ \text{Minimize} & f_3(\theta,\gamma,r) = (1+g(r))\sin\theta,\\ & 0\leq\theta\leq\pi/2,\\ & -\pi/4\leq\gamma\leq\pi/4,\\ & g(r)\geq 0.\end{array}\right\} \quad (3)$$

As described earlier, the Pareto-optimal solutions for the above problem are as follows:

$$0 \leq \theta^* \leq \pi/2, \quad -\pi/4 \leq \gamma^* \leq \pi/4, \quad g(r^*) = 0.$$

Figure 2 shows the overall objective search space with any function for $g(r)$ with $0 \leq g(r) \leq 1$. We shall discuss more about different $g(r)$ functions a little later. The above construction procedure illustrates how easily a multi-objective test problem can be constructed from an initial choice of a Pareto-optimal surface.

4.1 Construction of the decision space

Although the above three-objective problem requires three independent variables, the decision search space can be higher than three-dimensional. The three variables used above (θ, γ, and r) can all be considered as *meta-variables* and each of them can be considered as a function of n decision variables of the underlying problem:

$$\theta = \theta(x_1, x_2, \ldots, x_n), \quad (4)$$

$$\gamma = \gamma(x_1, x_2, \ldots, x_n), \quad (5)$$

$$r = r(x_1, x_2, \ldots, x_n). \quad (6)$$

The functions must be so chosen that they satisfy the lower and upper bounds of θ, γ and $g(r)$ mentioned in equation 3.

Although the above construction procedure is simple, it can be used to introduce different modes of difficulty described earlier. In the following, we describe a few such extensions of the above construction.

4.2 Difficulty in converging to the Pareto-optimal front

The difficulty of a search algorithm to progress towards the Pareto-optimal front from the interior of the objective search space can be introduced by simply using a difficult g function. It is clear that the Pareto-optimal surface corresponds to the minimum value of function g. A multi-modal g function with a global minimum at $g^* = 0$ and many local minima at $g^* = \nu_i$ value will introduce global and local Pareto-optimal fronts, where a multi-objective optimizer can get stuck to.

Moreover, even using a unimodal $g(r)$ function, variable density of solutions can be introduced in the search space. For example, if $g(r) = r^{10}$ is used, denser solutions exist away from the Pareto-optimal front. Figure 3 shows 15,000 solutions, which are randomly created in the decision variable space. On the objective space, they are shown to be biased away from the Pareto-optimal front. For a such a biased search space away from the Pareto-optimal front, multi-objective optimizers may have difficulties in converging quickly to the desired front.

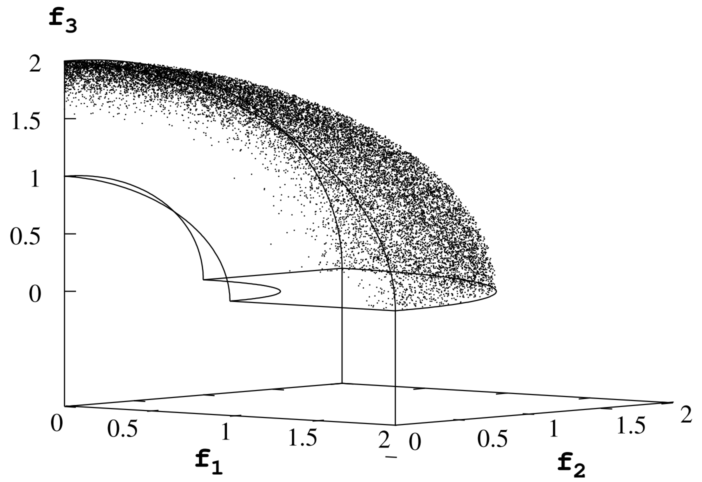


Figure 3: The effect of a non-linear g function.

4.3 Difficulties across the Pareto-optimal front

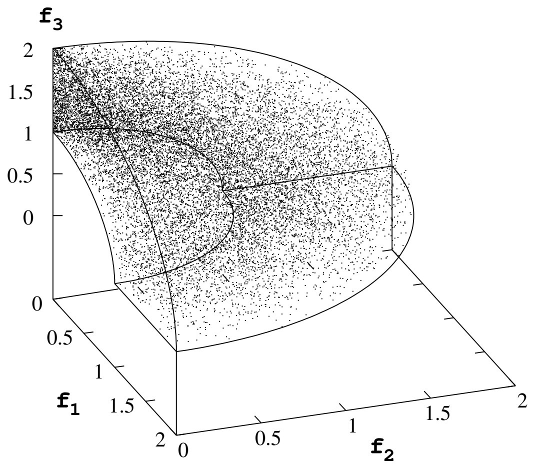
By using a non-linear mapping in equations 4 to 6, some portion of the search space can be made to have more dense solutions than the rest of the search space.

In order to create a variable density of solutions on the Pareto-optimal front, the θ and γ functions (as in equations 4 to 5) must be manipulated. For example, Figures 4 and 5 show the problem stated in equation 3 with

$$\begin{array}{lcl} \theta(x_1) = \frac{\pi}{2}x_1, & & \theta(x_1) = \frac{\pi}{2}0.5\left(1 + [2(x_1 - 0.5)]^{11}\right), \\ \gamma(x_2) = \frac{\pi}{2}x_2 - \frac{\pi}{4}. & \text{and} & \gamma(x_2) = \frac{\pi}{2}0.5\left(1 + [2(x_2 - 0.5)]^{11}\right) - \frac{\pi}{4}. \end{array}$$

respectively. In both cases, $g(r) = r = x_3$ is chosen. In order to satisfy the bounds in equation 3, we have chosen $0 \le x_1, x_2, x_3 \le 1$ and 15,000 randomly created points (in the decision space) are shown in each figure showing the objective space. The figures show the density of solutions in the search space gets affected by the choice of mapping of the meta-variables. In the second problem, there is a natural bias for an algorithm to find solutions in middle region of the search space[1]. In

[1]In three-objective knapsack problems, such a biased search space is observed elsewhere (Zitzler and Thiele, 1999).



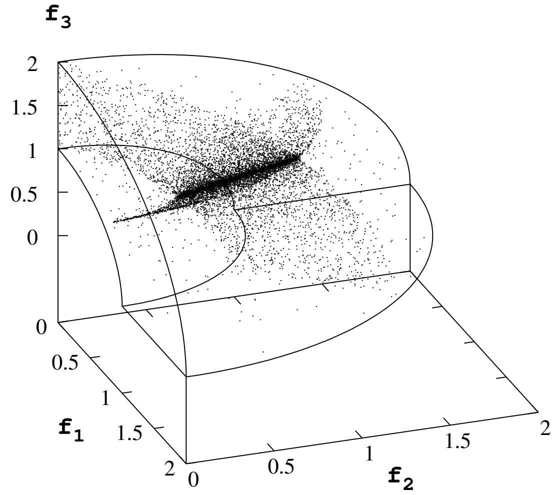


Figure 4: Linear mapping (15,000 points).

Figure 5: Non-Linear mapping introduces bias in the search space (15,000 points).

trying to solve such test problems, the task of an MOEA would be to find a widely distributed set of solutions on the entire Pareto-optimal front despite the natural bias of solutions in certain regions on the Pareto-optimal front.

4.4 Generic sphere problem

The following is a generic problem to that described in equation 3, having M objectives.

$$\left.\begin{array}{ll}
\text{Minimize} & f_1(\boldsymbol{\theta}, r) = (1+g(r))\cos\theta_1\cos(\theta_2)\cdots\cos(\theta_{M-2})\cos(\theta_{M-1}),\\
\text{Minimize} & f_2(\boldsymbol{\theta}, r) = (1+g(r))\cos\theta_1\cos(\theta_2)\cdots\cos(\theta_{M-2})\sin(\theta_{M-1}),\\
\text{Minimize} & f_3(\boldsymbol{\theta}, r) = (1+g(r))\cos\theta_1\cos(\theta_2)\cdots\sin(\theta_{M-2}),\\
\vdots & \vdots\\
\text{Minimize} & f_{M-1}(\boldsymbol{\theta}, r) = (1+g(r))\cos\theta_1\sin(\theta_2),\\
\text{Minimize} & f_M(\boldsymbol{\theta}, r) = (1+g(r))\sin\theta_1,\\
 & 0 \le \theta_i \le \pi/2, \quad \text{for } i = 1, 2, \ldots, (M-1),\\
 & g(r) \ge 0.
\end{array}\right\} \tag{7}$$

Note that the variables are mapped in a different manner here. The decision variables are mapped to the meta-variable vector $\boldsymbol{\theta}$ (of size $(M-1)$) as follows:

$$\theta_i = \frac{\pi}{2}x_i, \quad \text{for } i = 1, 2, \ldots, (M-1). \tag{8}$$

The above mapping and the condition on θ_i in equation 7 restrict each of the above x_i to lie within $[0, 1]$. To simplify the matter, the function $g(r) = r = x_M^2$ (where $x_M \in [-1, 1]$) can be used. The Pareto-optimal surface occurs for the minimum of the g function, or at $x_M^* = 0$ and the function values must satisfy the following condition:

$$\sum_{i=1}^{M}(f_i^*)^2 = 1. \tag{9}$$

As mentioned earlier, the difficulty of the above test problem can also be varied by using different functionals for f_i and g.

4.5 Curve problem

Instead of having a complete M-dimensional surface as the Pareto-optimal surface, an M-dimensional curve can also be chosen as the Pareto-optimal front to the above problem. We realize that in this case there would be only one independent variable describing the Pareto-optimal front. A simple way to achieve this would be to use the following mapping of variables:

$$\theta_i = \frac{\pi}{4(1+g(r))}\left(1 + 2g(r)x_i\right), \quad \text{for } i = 2, 3, \ldots, (M-1), \tag{10}$$

The above mapping ensures that the curve is the only non-dominated region in the entire search space. Since $g(r) = 0$ corresponds to the Pareto-optimal front, $\theta_i = \pi/4$ for all but the first variable. The advantage of this problem over the generic sphere problem as a test problem is that a two-dimensional plot of Pareto-optimal points with f_M and any other f_i will mark a curve (circular or elliptical). A plot with any two objective functions (other than f_M) will show a straight line. Figure 6 shows a sketch of the search space and the resulting Pareto-optimal curve for a three-objective version of the above problem. One drawback with this formulation is that

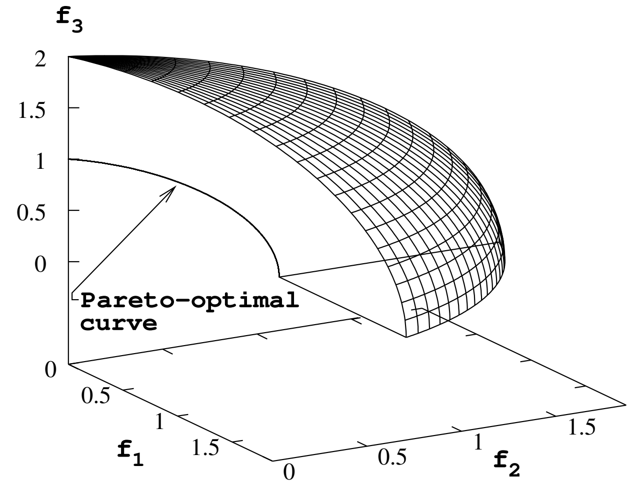


Figure 6: The search space and the Pareto-optimal curve.

the density of solutions closer to the Pareto-optimal curve is more than anywhere else in the search space. In order to make the problem more difficult, a non-linear $g(r)$ function with a higher density of solutions away from $g(r) = 0$ (such as $g(r) = 1/r^\alpha$, where $\alpha \gg 1$) can be used. Using a multi-modal $g(r)$ function will also cause multiple local Pareto-optimal surfaces to exist. Interestingly, this drawback of the problem can be used to create a hard maximization problem. If the all objectives are maximized in the above problem, the top surface becomes the desired Pareto-optimal front. Since there exists less dense solutions on this surface, this problem may be a difficult maximization test problem.

4.6 Comet problem

To illustrate the concept of the bottom-up approach of test problem design further, we design one more problem which has a comet-like Pareto-optimal front. Starting from a widely spread region, the Pareto-optimal front continuously reduces to thinner region. Finding a wide variety of solutions in both broad and thin portions of the Pareto-optimal region simultaneously will be a challenging task for any MOEA:

$$\left.\begin{array}{ll}\text{Minimize} & f_1(\mathbf{x}) = (1+g(x_3))(x_1^3x_2^2 - 10x_1 - 4x_2),\\ \text{Minimize} & f_2(\mathbf{x}) = (1+g(x_3))(x_1^3x_2^2 - 10x_1 + 4x_2),\\ \text{Minimize} & f_3(\mathbf{x}) = 3(1+g(x_3))x_1^2,\\ & 1 \le x_1 \le 3.5,\\ & -2 \le x_2 \le 2,\\ & g(x_3) \ge 0.\end{array}\right\} \quad (11)$$

Here, we have chosen $g(x_3) = x_3$ and $0 \le x_3 \le 1$. The Pareto-optimal surface corresponds to $x_3^* = 0$ and for $-2 \le x_1^{*3} x_2^* \le 2$ with $1 \le x_1^* \le 3.5$. Figure 7 shows the Pareto-optimal front on the $x_3 = 0$ surface. For a better illustration purpose, the figure is plotted with negative of

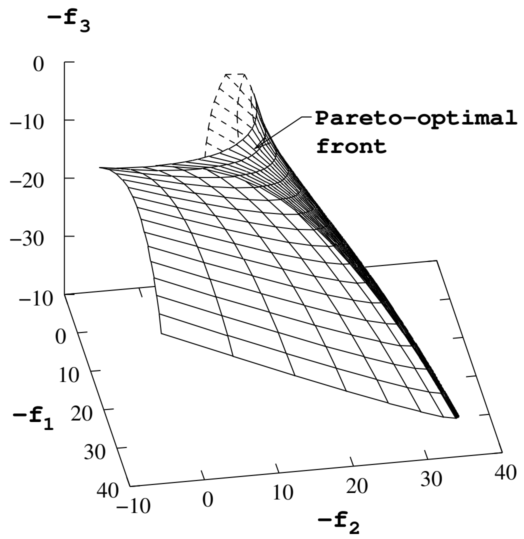


Figure 7: The comet problem.

all f_i values. This problem illustrates that the entire $g = g^*$ surface need not correspond to the Pareto-optimal front. Only the region which dominates the rest of the $g = g^*$ surface belongs to the Pareto-optimal front.

We have designed this function and the curve function for a special purpose. Because of the narrow Pareto-optimal region in both problems, we argue that the classical generating methods will require a large computational overhead in solving the above problems. Figure 8 shows the projection of the Pareto-optimal region in the f_1-f_2 space of the comet problem. For the use of the ϵ-constraint method as a method of generating Pareto-optimal solutions (usually recommended for its convergence properties), the resulting single-objective optimization problem, which has to be solved for different combination of ϵ_1 and ϵ_2, is as follows:

$$\left.\begin{array}{ll}\text{Minimize} & f_3(\mathbf{x}),\\ \text{subject to} & f_2(\mathbf{x}) \le \epsilon_2,\\ & f_1(\mathbf{x}) \le \epsilon_1,\\ & \mathbf{x} \in \mathcal{D},\end{array}\right\} \quad (12)$$

where $\mathcal{D}$ is the feasible decision space. It is well known that the minimum solution for the above problem for any ϵ_1 and ϵ_2 (≥ 0) is either a Pareto-optimal solution or is infeasible (Mietinnen,

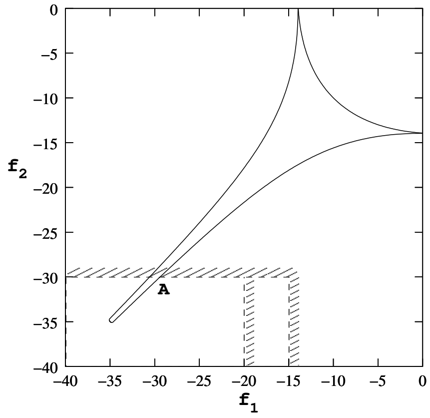


Figure 8: Classical generating method with the ϵ-constraint method will produce redundant single-objective optimization problems.

1997). The figure illustrates a scenario with $\epsilon_2 = -30$. It can be seen from the figure that solution of the above single-objective optimization problem for $\epsilon_1 = -15$ is not going to produce any new Pareto-optimal solution other than that obtained for $\epsilon_1 = -20$ (for example) or for ϵ_1 set to get the Pareto-optimal solution at A. Thus, the above generating method will resort to solving many redundant single-objective optimization problems. By calculating the area of the projected Pareto-optimal region, it is estimated that about 88% of single-objective optimization problems are redundant in the above three-objective optimization problem, if a uniform set of $\boldsymbol{\epsilon}$ vectors are chosen in the generating method. Compared to classical generating methods, MOEAs may show superior performance in these problems in terms of overall computational effort needed in finding multiple and well distributed Pareto-optimal solutions. This is mainly because of their implicit parallel processing which enables them to quickly settle to feasible regions of interest and due to their population approach which allow them to find a wide variety of solutions simultaneously with the action of a niche-preserving operator.

4.7 Test Problem Generator

The earlier study (Deb, 2001) suggested a generic multi-objective test problem generator, which belongs to this bottom-up approach. For M objective functions, with a complete decision variable vector partitioned into M non-overlapping groups

$$x \equiv (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{M-1}, \mathbf{x}_M)^T,$$

the following function was suggested:

$$\left.\begin{array}{rl}
\text{Minimize} & f_1(\mathbf{x}_1), \\
\text{Minimize} & f_2(\mathbf{x}_2), \\
\vdots & \vdots \\
\text{Minimize} & f_{M-1}(\mathbf{x}_{M-1}), \\
\text{Minimize} & f_M(\mathbf{x}) = g(\mathbf{x}_M) h\left(f_1(\mathbf{x}_1), f_2(\mathbf{x}_2), \ldots, f_{M-1}(\mathbf{x}_{M-1}), g(\mathbf{x}_M)\right), \\
\text{subject to} & \mathbf{x}_i \in \mathbb{R}^{|\mathbf{x}_i|}, \quad \text{for } i = 1, 2, \ldots, M.
\end{array}\right\} \tag{13}$$

Here, the Pareto-optimal front is described by solutions which are global minimum of $g(\mathbf{x}_M)$ (with g^*). Thus, the Pareto-optimal front is described as

$$f_M = g^* h(f_1, f_2, \ldots, f_{M-1}). \tag{14}$$

Since g^* is a constant number, the h function (with a fixed $g = g^*$) describes the Pareto-optimal surface. In the bottom-up approach of test problem design, the user can first choose an h function in terms of the objective function values, without caring about the decision variables at first. For example, for constructing a problem with a non-convex Pareto-optimal front, a non-convex h function can be chosen, such as the following:

$$h(f_1, f_2, \ldots, f_{M-1}) = 1 - \left(\frac{\sum_{i=1}^{M-1} f_i}{\beta} \right)^{\alpha}, \tag{15}$$

where $\alpha > 1$. Figure 9 shows a non-convex Pareto-optimal front with $\alpha = 2$ for $M = 3$ and $\beta = 0.5$.

A disjoint set of Pareto-optimal front can be constructed by simply choosing a multi-modal h function as done in the case of two-objective test problem design (Deb, 1999). Figure 10 illustrates a disconnected set of Pareto-optimal surfaces (for three-objectives), which can be generated from the following generic h function:

$$h(f_1, f_2, \ldots, f_{M-1}) = 2M - \sum_{i=1}^{M-1} \left(2f_i + \sin(3\pi f_i)\right). \tag{16}$$

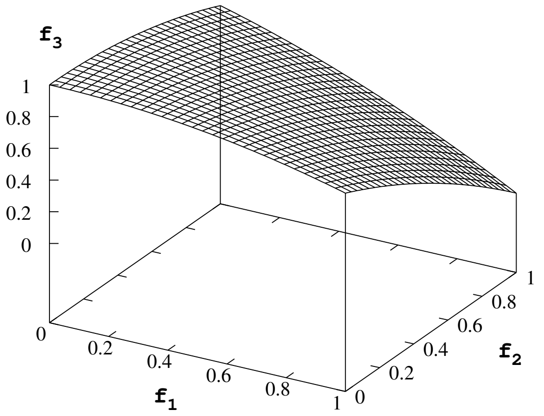


Figure 9: A non-convex Pareto-optimal front.

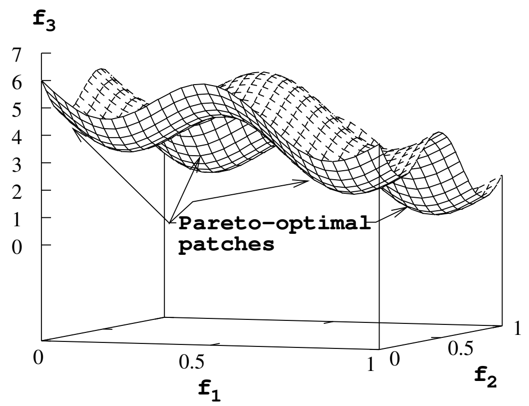


Figure 10: A disjoint set of Pareto-optimal regions.

Once the h function is chosen, a g function can be chosen to construct the entire objective search space. It is important to note that the g function is defined over a set of decision variables. Recall that the Pareto-optimal front corresponds to the global minimum value of the g function. Any other values of g will represent a surface parallel to the Pareto-optimal surface. All points in this parallel surface will be dominated by some solutions in the Pareto-optimal surface. As mentioned earlier, the g function can be used to introduce complexities in approaching towards the Pareto-optimal region. For example, if g has a local minimum, a local Pareto-optimal front exists on the corresponding parallel surface.

Once appropriate h and g functions are chosen, f_1 to f_{M-1} can be chosen as functions of different non-overlapping sets of decision variables. Using a non-linear objective function introduces a variable density of solutions along that objective. The non-linearity in these functions will test an MOEA's ability to find a good distribution of solutions, despite the natural non-uniform density in solutions in the search space. Another way to make the density of solutions non-uniform is to use an overlapping set of decision variables for objective functions. To construct more difficult test problems, the procedure of mapping the decision variables to an intermediate variable vector, as suggested earlier (Deb, 1999) can also be used here.

4.8 Advantages and disadvantages of the bottom-up approach

The advantage of using the above bottom-up approach is that the exact form of the Pareto-optimal surface can be controlled by the developer. The number of objectives and the variability in density of solutions can all be controlled by choosing proper functions.

Since the Pareto-optimal front must be expressed mathematically, some complicated Pareto-optimal fronts can be difficult to write mathematically. Nevertheless, the ability to control different features of the problem is the main strength of this approach.

5 Constraint Surface Approach

Unlike starting from a pre-defined Pareto-optimal surface in the bottom-up approach, the constraint surface approach begins by a predefining the overall search space. Here, a simple geometry such as a rectangular hyper-box is assumed. Each objective function value is restricted to lie within a predefined lower and a upper bound. The resulting problem is as follows:

$$\left.\begin{array}{ll}\text{Minimize} & f_1(\mathbf{x}),\\ \text{Minimize} & f_2(\mathbf{x}),\\ \vdots & \vdots\\ \text{Minimize} & f_M(\mathbf{x}),\\ \text{Subject to} & f_i^{(L)} \leq f_i(\mathbf{x}) \leq f_i^{(U)} \quad \text{for } i = 1, 2, \ldots, M.\end{array}\right\} \tag{17}$$

It is intuitive that the Pareto-optimal set of the above problem has only one solution (the solution with the lower bound of each objective $(f_1^{(L)}, f_2^{(L)}, \ldots, f_1^{(L)})^T$. Figure 11 shows this problem for three objectives (with $f_i^{(L)} = 0$ and $f_i^{(U)} = 1$) and the resulting singleton Pareto-optimal solution $\mathbf{f} = (0, 0, 0)^T$) is also marked.

The problem is now made more interesting by adding a number of constraints (linear or non-linear):

$$g_j(f_1, f_2, \ldots, f_M) \geq 0, \quad \text{for } j = 1, 2, \ldots, J. \tag{18}$$

This is done to chop off portions of the original rectangular region systematically. Figure 12 shows the resulting feasible region after adding the following two linear constraints:

$$\begin{aligned} g_1 &\equiv f_1 + f_3 - 0.5 \geq 0,\\ g_2 &\equiv f_1 + f_2 + f_3 - 0.8 \geq 0. \end{aligned}$$

What remains is the feasible search space. The objective of the above problem is to find the non-dominated portion of the boundary of the feasible search space. Figure 12 also marks the Pareto-optimal surface of the above problem. For simplicity and easier comprehension, each constraint involving at most two objectives (similar to the first constraint above) can be used.

In addition to the complicated shape of the Pareto-optimal front, further difficulties can be introduced by using varying density of solutions in the search space. This can be easily achieved by using non-linear functionals for f_i with the decision variables. Interestingly, there exist two-variable and three-variable constrained test problems TNK (Tanaka, 1995) and Tamaki (1996) in the literature using the above concept. In this problem, only two objectives (with $f_i = x_i$) and two constraints were used. The use of $f_i = x_i$ made a uniform density of solutions in the search

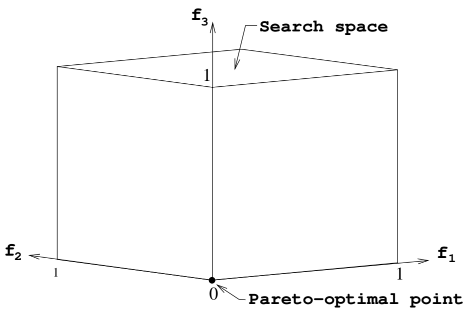


Figure 11: Entire cube is the search space. The origin is the sole Pareto-optimal solution.

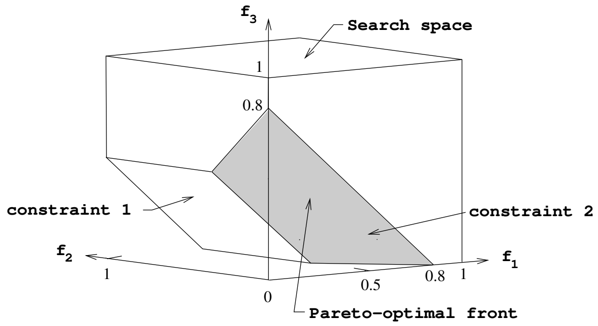


Figure 12: Two constraints are added to eliminate a portion of the cube, thereby resulting in a more interesting Pareto-optimal front.

space. As an illustration of further difficulties through non-linear f_i, we construct the following three-objective problem with a bias in the search space:

$$\left.\begin{array}{ll}\text{Minimize} & f_1(x_1) = 1 + (x_1 - 1)^5, \\ \text{Minimize} & f_2(x_2) = x_2, \\ \text{Minimize} & f_3(x_3) = x_3, \\ \text{Subject to} & g_1 \equiv f_3^2 + f_1^2 - 0.5 \geq 0, \\ & g_2 \equiv f_3^2 + f_2^2 - 0.5 \geq 0, \\ & 0 \leq x_1 \leq 2, \\ & 0 \leq x_2, x_3 \leq 1. \end{array}\right\} \tag{19}$$

Figure 13 shows 25,000 feasible solutions randomly generated in the decision variable space. The

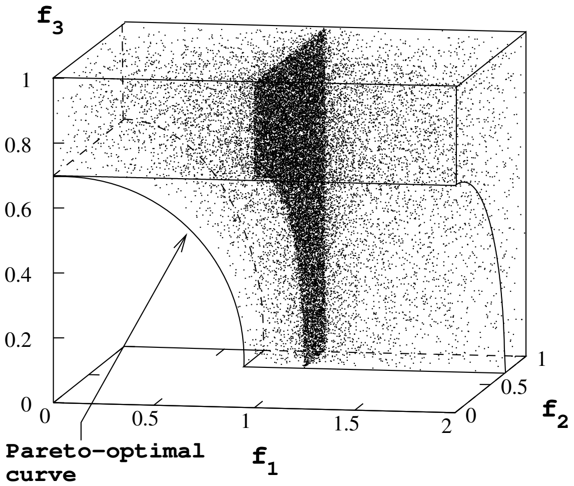


Figure 13: Non-Linearity in functionals produces non-uniform density of solutions.

Pareto-optimal curve and the feasible region are shown in the figure. Because of the non-linearity in the functional f_1 with x_1, the search space results in a variable density of solutions along the f_1

axis. Solutions are more dense near $f_1 = 1$ than any other region in the search space. Although this apparent plane ($f_1 = 1$) is not a local Pareto-optimal front, an MOEA may get attracted here simply because of sheer density of solutions near it.

It is clear that by choosing complicated functions of f_1 to f_M, more complicated search spaces can be created. Since the task of an MOEA is to reach the Pareto-optimal region (which is located at one end of the feasible search space), interesting hurdles in the search space can be placed to provide difficulties to an MOEA to reach the desired Pareto-optimal front.

5.1 Advantages and Disadvantages

The construction process here is much simpler compared to the bottom-up approach. Simple geometric constraints can be used to construct the feasible search space. Using this procedure, different shapes (convex, nonconvex, or discrete) of the Pareto-optimal region can be generated. Unlike the bottom-up approach, here the feasible search space is not constructed by layer-wise construction from the Pareto-optimal front. Since no particular structure is used, the feasible objective space can be derived with any non-linear mapping of the decision variable space.

However, the resulting Pareto-optimal front will, in general, be hard to express mathematically. Moreover, although the constraint surfaces can be simple, the shape and continuity of the resulting Pareto-optimal front may not be easy to visualize. Another difficulty is that since the Pareto-optimal front will lie on one or more constraint boundaries, a good constraint-handling strategy must be used with an MOEA. Thus, this approach may be ideal for testing MOEAs for their ability to handle constraints.

6 Difficulties with Existing MOEAs

Most MOEA studies up until now have considered two objectives, except a few application studies where more than two objectives are used. This is not to say that the existing MOEAs cannot be applied to problems having more than two objectives. Developers of the state-of-the-art MOEAs (such as PAES (Knowles and Corne, 1999), SPEA (Zitzler and Thiele, 1999), NSGA-II (Deb et al., 2000) and others) have all considered the scalability aspect while developing their algorithms. The domination principle, non-dominated sorting algorithms, elite-preserving and other EA operators can all be extended for handling more than two objectives. Although the niching operator can also be applied in most cases, their computational issues and ability in maintaining a good distribution of solutions are needed to be investigated in higher-objective problems. For example, a niching operator may attempt to maintain a good distribution of solutions by replacing a crowded solution with a less crowded one and the crowding of a solution may be determined by the distance from its neighbors. For two objectives, the definition of a neighbor along the Pareto-optimal curve is clear and involves only two solutions (left and right solutions). However, for more than two objectives, when the Pareto-optimal front is a higher-dimensional surface, it is not clear which (and how many) solutions are neighbors of a solution. Even if a definition can be made, finding a distance metric from all distances of its neighbors gets computationally expensive because of the added dimensionality. Because of the higher dimensionality of the Pareto-optimal front, more computationally effective distribution metrics may be needed. Although a widely distributed set of solutions can be found, as using NSGA-II or SPEA2 (a modified version of SPEA suggested in (Zitzler, Laumanns, and Thiele, 2001)) shown in the next section, the obtained distribution can be far from being a uniformly distributed set of 100 points on the Pareto-optimal surface. The niching method are usually designed to attain a uniform distribution (provided that the EA operators are able to generate the needed solutions), the overall process may be much slower in problems with large number of objectives. The test problems suggested in this paper will certainly enable researchers to make such a complexity study for the state-of-the-art MOEAs.

Although the distribution of solutions is a matter to test for problems with large number of objectives, the convergence to the true Pareto-optimal front is also important to keep track of. Because of sheer increase in the dimensionality in the objective space, interactions among decision variables may produce difficulties in terms of having local Pareto-optimal fronts and variable density of solutions. An increase in dimensionality of the objective space also causes a large proportion of a random initial population to be non-dominated to each other (Deb, 2001), thereby reducing the effect of selection operator in an MOEA. Thus, it is also important to test if the existing domination-based MOEAs can reach the true Pareto-optimal front in such test problems. Since the desired front will be known in test problems, a convergence metric (such as average distance to the front) can be used to track the convergence of an algorithm. We discuss more about performance metrics in the next section.

7 Performance Metrics

A number of performance metrics for MOEA studies have been discussed in Veldhuizen (1999) and Deb (2001). The latter study has classified the metrics according to the aspect measures by them and suggested three different categories:

1. Metrics that evaluate closeness to the Pareto-optimal front,
2. Metrics that evaluate diversity in obtained solutions, and
3. Metrics that evaluate both the above.

Most of the existing metrics require a pre-specified set of Pareto-optimal (reference) solutions P^*. The obtained set Q is compared against P^*.

In the first category, almost all metrics suggested in the context of two-objective problems can be applied to problems having more than two objectives. The error ratio metric, which counts the number of solutions common to P^* and Q can be used in higher dimensions. The set coverage metric which calculates the proportion of solutions in Q which are dominated by P^* can also be used in higher dimensions. The generational distance metric, which calculates the geometric mean of the nearest Euclidean distances of solutions Q from members of P^*, can also be used in higher dimensions.

However, most of the metrics suggested for measuring the spread in two-objective problems may not be possible to use in higher-objective problems so easily. For example, the spacing metric, which calculates the standard deviation in the distances between consecutive solutions, would be difficult to use in higher dimensions. The concept of consecutive solutions in higher dimensions does not simply exist. Although Schott's (1995) suggestion of using the minimum objective-wise distance to any other solution can be used in higher-objective problems, a Q with a good spacing metric does not necessarily mean a good distribution of solutions in the entire Pareto-optimal front. To ensure a good distribution in the entire Pareto-optimal front, the obtained set Q must be compared with the known Pareto-optimal front P^*, so that the obtained set is evaluated for its spread with the extreme solutions of the front. Although the spread metric, which uses consecutive distances, is a way to consider the extreme solutions in the metric evaluation, it cannot be used in problems having more than two objectives. The maximum spread metric, which measures the Euclidean distance between extreme solutions, does not reveal the true distribution of intermediate solutions anyway. However, the chi-square-like deviation measure can be used with a user-defined neighborhood parameter σ. In this metric, for each solution of P^* the number of points in Q which are within a σ away can be counted. Such a count vector can then be compared with a pre-defined (in most cases, a uniform) distribution and the deviation measure can be calculated. Alternatively, the Pareto-optimal front can represented into a number of small hyper-boxes. Thereafter, the number of solutions (of Q) present in each hyper-box is counted.

Using these counts, either the chi-square-like deviation measure (from a uniform distribution of solutions in all hyper-boxes) or simply the number of occupied hyper-boxes can be used as a distribution metric.

In the third category, the hyper-volume metric, measuring the overall hyper-volume covered by the set Q, in combination with the relative set coverage metric were designed with respect to an arbitrary number of objectives and have been used already on 2, 3, and 4-objective knapsack problems (Zitzler and Thiele, 1999). The mutual non-covered hyper-volume metric (Zitzler, 1999), which is similar to the set coverage metric, is applicable in higher-objective problems as well. Nevertheless, the convergence to coverage issue may not become clear from one such metric. Furthermore, the attainment surface metric is in another method independent of the number of objectives, but repeatedly finding the intersection of cross-lines with higher-dimensional attainment surfaces may be computationally expensive.

From the above discussion, it is clear that some of the existing performance metrics for evaluating the distribution of solutions can be used in higher-objective problems but they must be evaluated with caution. Further research in this direction is necessary to develop faster and better performance metrics.

8 Test Problem Suite

Using the latter two approaches of test problem design discussed in this paper, we suggest here a representative set of test problems. However, other more interesting and useful test problems can also be designed using the techniques of this paper.

8.1 Test Problem DTLZ1

As a simple test problem, we construct an M-objective problem with a linear Pareto-optimal front:

$$\left.\begin{array}{rl}\text{Minimize} & f_1(\mathbf{x}) = \frac{1}{2}x_1x_2\cdots x_{M-1}(1+g(\mathbf{x}_M)),\\ \text{Minimize} & f_2(\mathbf{x}) = \frac{1}{2}x_1x_2\cdots(1-x_{M-1})(1+g(\mathbf{x}_M)),\\ \vdots & \vdots\\ \text{Minimize} & f_{M-1}(\mathbf{x}) = \frac{1}{2}x_1(1-x_2)(1+g(\mathbf{x}_M)),\\ \text{Minimize} & f_M(\mathbf{x}) = \frac{1}{2}(1-x_1)(1+g(\mathbf{x}_M)),\\ \text{subject to} & 0 \le x_i \le 1, \quad \text{for } i = 1,2,\ldots,n.\end{array}\right\} \tag{20}$$

The functional $g(\mathbf{x}_M)$ requires $|\mathbf{x}_M| = k$ variables and must take any function with $g \ge 0$. We suggest the following:

$$g(\mathbf{x}_M) = 100\left[|\mathbf{x}_M| + \sum_{x_i\in\mathbf{x}_M}(x_i-0.5)^2 - \cos(20\pi(x_i-0.5))\right]. \tag{21}$$

The Pareto-optimal solution corresponds to $\mathbf{x}_M = \mathbf{0}$ and the objective function values lie on the linear hyper-plane: $\sum_{m=1}^{M} f_m = 0.5$. A value of $k = 5$ is suggested here. In the above problem, the total number of variables is $n = M + k - 1$. The difficulty in this problem is to converge to the hyper-plane. The search space contains $(11^k - 1)$ local Pareto-optimal fronts, each of which can attract an MOEA. NSGA-II nd SPEA2 with a population size of 100 is run for 300 generations using a real-parameter SBX crossover operator ($\eta_c = 15$) and a variable-wise polynomial mutation operator ($\eta_m = 20$). The crossover probability of 1.0 and mutation probability of $1/n$ are used. After a variable is crossed with the SBX probability distribution, they are exchanged with a probability 0.5. The performances of NSGA-II and SPEA2 are shown in Figures 14 and 15, respectively. The figure shows that both NSGA-II and SPEA2 come close to the Pareto-optimal front and the distributions of solutions over the Pareto-optimal front are also

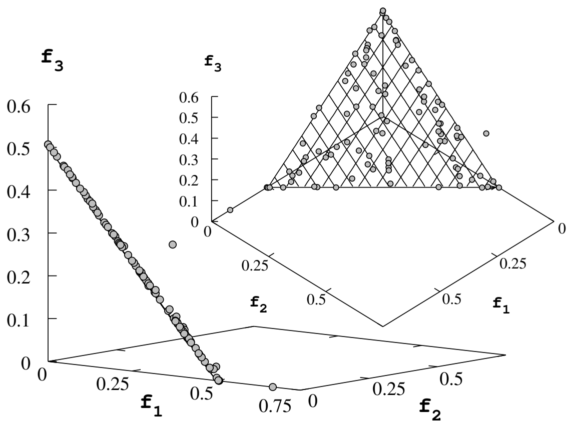


Figure 14: The NSGA-II population on test problem DTLZ1.

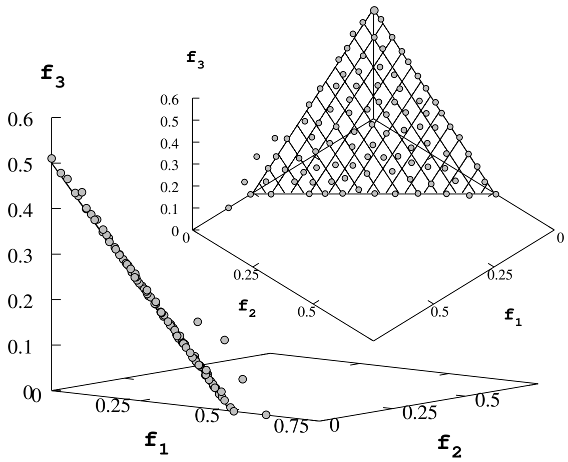


Figure 15: The SPEA2 population on test problem DTLZ1.

not bad. In this problem and in most of the latter problems, we observed a better distribution of solutions with SPEA2 compared to NSGA-II. However, the better distributing ability of SPEA2 comes with a larger computational complexity in its selection/truncation approach compared to that in the objective-wise crowding approach of NSGA-II.

The problem can be made more difficult by using other difficult multi-modal g functions (using a larger k) and/or replacing x_i by non-linear mapping $x_i = N_i(y_i)$ and treating y_i as decision variables.

For a scale-up study, we suggest testing an MOEA with different values of M, may be in the range $M \in [2, 10]$. It is interesting to note that for $M > 3$ cases all Pareto-optimal solutions on a three-dimensional plot involving f_M and any two other objectives will lie on or below the above hyper-plane.

8.2 Test Problem DTLZ2

This test problem is identical to the problem described in subsection 4.4:

$$\left.\begin{array}{ll}
\text{Minimize} & f_1(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(x_1\pi/2)\cos(x_2\pi/2)\cdots\cos(x_{M-2}\pi/2)\cos(x_{M-1}\pi/2), \\
\text{Minimize} & f_2(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(x_1\pi/2)\cos(x_2\pi/2)\cdots\cos(x_{M-2}\pi/2)\sin(x_{M-1}\pi/2), \\
\text{Minimize} & f_3(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(x_1\pi/2)\cos(x_2\pi/2)\cdots\sin(x_{M-2}\pi/2), \\
\vdots & \vdots \\
\text{Minimize} & f_{M-1}(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(x_1\pi/2)\sin(x_2\pi/2), \\
\text{Minimize} & f_M(\mathbf{x}) = (1+g(\mathbf{x}_M))\sin(x_1\pi/2), \\
& 0 \le x_i \le 1, \quad \text{for } i = 1, 2, \ldots, n, \\
\text{where} & g(\mathbf{x}_M) = \sum_{x_i \in \mathbf{x}_M}(x_i - 0.5)^2.
\end{array}\right\} \tag{22}$$

The Pareto-optimal solutions corresponds to $x_M^* = 0.5$ and all objective function values must satisfy the equation 9. Here, we recommend $k = |\mathbf{x}_M| = 10$. The total number of variables is $n = M+k-1$. NSGA-II and SPEA2 with identical parameter setting as in DTLZ1 simulation run finds Pareto-optimal solutions very close to the true Pareto-optimal front, as shown in Figures 16 and 17, respectively.

This function can also be used to investigate an MOEA's ability to scale up its performance

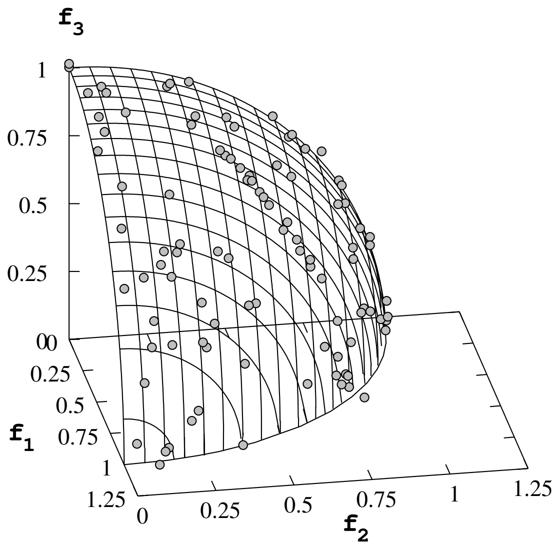


Figure 16: The NSGA-II population on test problem DTLZ2.

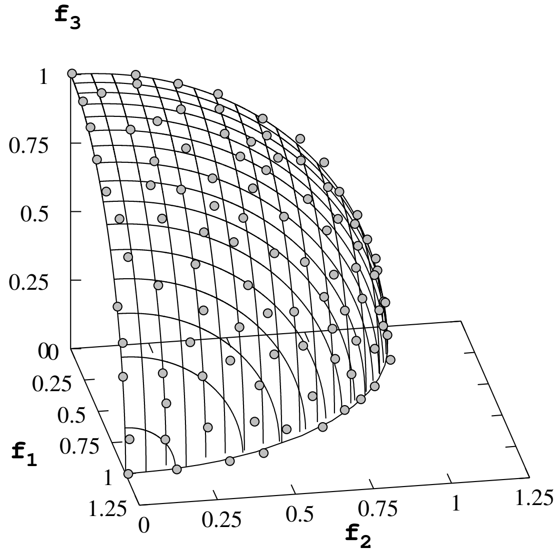


Figure 17: The SPEA2 population on test problem DTLZ2.

in large number of objectives. Like in DTLZ1, for $M > 3$, the Pareto-optimal solutions must lie inside the first quadrant of the unit sphere in a three-objective plot with f_M as one of the axes. To make the problem difficult, each variable x_i (for $i = 1$ to $(M-1)$) can be replaced by the mean value of p variables: $x_i = \frac{1}{p}\sum_{k=(i-1)p+1}^{ip} x_k$.

8.3 Test Problem DTLZ3

In order to investigate an MOEA's ability to converge to the global Pareto-optimal front, we suggest using the above problem with the g function given in equation 21:

$$\left.\begin{array}{ll}
\text{Minimize} & f_1(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(x_1\pi/2)\cos(x_2\pi/2)\cdots\cos(x_{M-2}\pi/2)\cos(x_{M-1}\pi/2), \\
\text{Minimize} & f_2(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(x_1\pi/2)\cos(x_2\pi/2)\cdots\cos(x_{M-2}\pi/2)\sin(x_{M-1}\pi/2), \\
\text{Minimize} & f_3(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(x_1\pi/2)\cos(x_2\pi/2)\cdots\sin(x_{M-2}\pi/2), \\
\vdots & \vdots \\
\text{Minimize} & f_{M-1}(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(x_1\pi/2)\sin(x_2\pi/2), \\
\text{Minimize} & f_M(\mathbf{x}) = (1+g(\mathbf{x}_M))\sin(x_1\pi/2), \\
 & 0 \le x_i \le 1, \quad \text{for } i = 1,2,\ldots,n, \\
\text{where} & g(\mathbf{x}_M) = 100\left[|\mathbf{x}_M| + \sum_{x_i\in\mathbf{x}_M}(x_i-0.5)^2 - \cos(20\pi(x_i-0.5))\right].
\end{array}\right\} \tag{23}$$

Here, we suggest $k = |\mathbf{x}_M| = 10$. There are a total of $n = M + k - 1$ decision variables in this problem. The above g function introduces $(3^k - 1)$ local Pareto-optimal fronts, and one global Pareto-optimal front. All local Pareto-optimal fronts are parallel to the global Pareto-optimal front and an MOEA can get stuck at any of these local Pareto-optimal fronts, before converging to the global Pareto-optimal front (at $g^* = 0$). The global Pareto-optimal front corresponds to $\mathbf{x}_M = (0.5, \ldots, 0.5)^T$. The next local Pareto-optimal front is at $g^* = 1$. NSGA-II and SPEA2 populations after 500 generations are shown in the true Pareto-optimal fronts in Figures 18 and 19. It is seen that both algorithms could not quite converge on to the true front, however both algorithms have maintained a good diversity of solutions on the true front. The problem can be

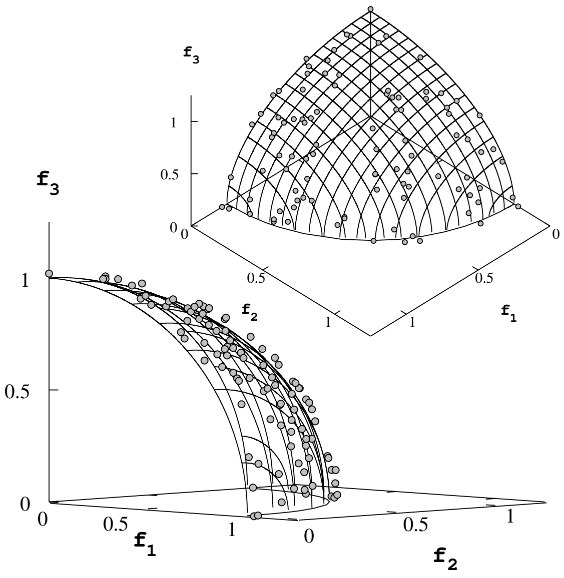


Figure 18: The NSGA-II population on test problem DTLZ3.

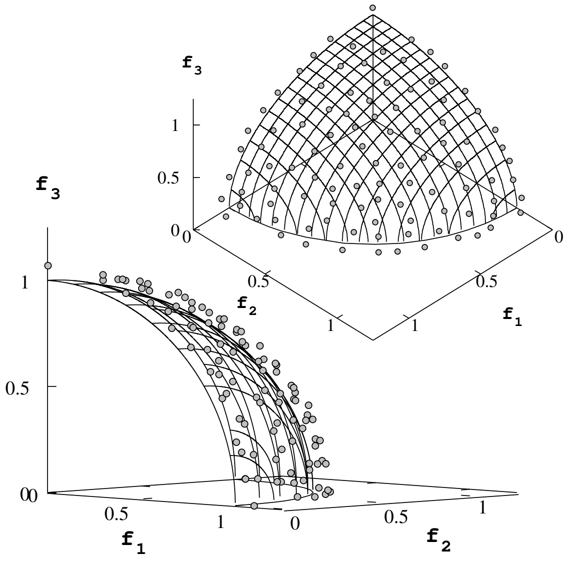


Figure 19: The SPEA2 population on test problem DTLZ3.

made more difficult by using a large k or a higher-frequency cosine function.

8.4 Test Problem DTLZ4

In order to investigate an MOEA's ability to maintain a good distribution of solutions, we modify problem DTLZ2 with a different meta-variable mapping:

$$\left.\begin{array}{ll}
\text{Minimize} & f_1(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(x_1^\alpha\pi/2)\cos(x_2^\alpha\pi/2)\cdots\cos(x_{M-2}^\alpha\pi/2)\cos(x_{M-1}^\alpha\pi/2), \\
\text{Minimize} & f_2(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(x_1^\alpha\pi/2)\cos(x_2^\alpha\pi/2)\cdots\cos(x_{M-2}^\alpha\pi/2)\sin(x_{M-1}^\alpha\pi/2), \\
\text{Minimize} & f_3(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(x_1^\alpha\pi/2)\cos(x_2^\alpha\pi/2)\cdots\sin(x_{M-2}^\alpha\pi/2), \\
\vdots & \vdots \\
\text{Minimize} & f_{M-1}(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(x_1^\alpha\pi/2)\sin(x_2^\alpha\pi/2), \\
\text{Minimize} & f_M(\mathbf{x}) = (1+g(\mathbf{x}_M))\sin(x_1^\alpha\pi/2), \\
 & 0 \le x_i \le 1, \quad \text{for } i = 1,2,\ldots,n, \\
\text{where} & g(\mathbf{x}_M) = \sum_{x_i\in\mathbf{x}_M}(x_i-0.5)^2.
\end{array}\right\} \tag{24}$$

The parameter $\alpha = 100$ is suggested here. Here, too, all variables x_1 to x_{M-1} are varied in $[0,1]$. We also suggest $k = 10$ here. There are $n = M + k - 1$ decision variables in the problem. This modification allows a dense set of solutions to exist near the f_M-f_1 plane (as in Figure 5). NSGA-II and SPEA2 populations at the end of 200 generations are shown in Figures 20 and 21, respectively. For this problem, the final population is dependent on the initial population. But in both methods, we have obtained three different outcomes: (i) all solutions are in the f_3-f_1 plane, (ii) all solutions are in the f_1-f_2 plane and (iii) solutions are on the entire Pareto-optimal surface. Since the problem has more dense solutions near the f_3-f_1 and f_1-f_2 planes, some simulation runs of both NSGA-II and SPEA2 get attracted to these planes. Problems with a biased density of solutions at other regions in the search space may also be created using the mapping suggested in subsection 4.3. It is interesting to note that although the search space has a variable density of solutions, the classical weighted-sum approaches or other directional methods may not have any

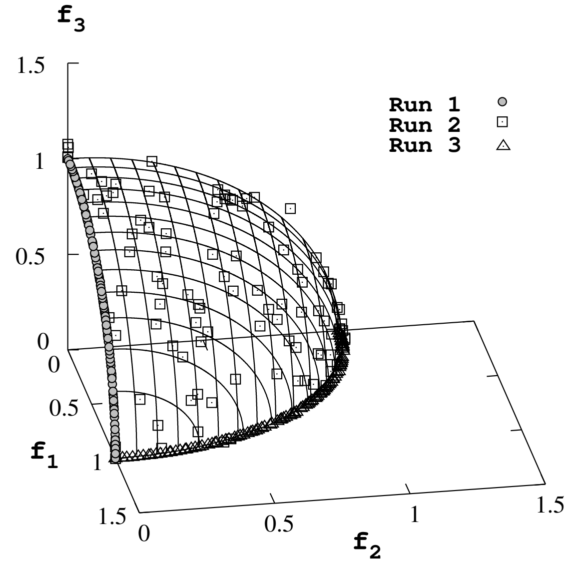


Figure 20: The NSGA-II population on test problem DTLZ4. Three different simulation runs are shown.

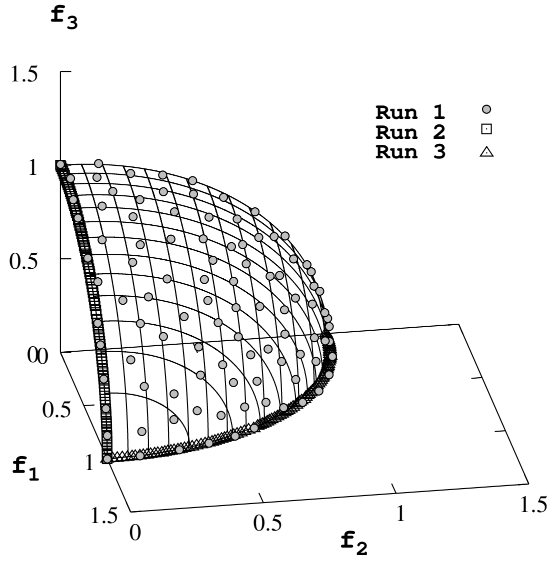


Figure 21: The SPEA2 population on test problem DTLZ4. Three different simulation runs are shown.

added difficulty in solving these problems compared to DTLZ2. Since MOEAs attempt to find multiple and well-distributed Pareto-optimal solutions in one simulation run, these problems may hinder MOEAs to achieve a well-distributed set of solutions.

8.5 Test Problem DTLZ5

The mapping of θ_i in the test problem DTLZ2 can be replaced with that given in equation 10:

$$\left.\begin{array}{ll}
\text{Minimize} & f_1(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(\theta_1\pi/2)\cos(\theta_2\pi/2)\cdots\cos(\theta_{M-2}\pi/2)\cos(\theta_{M-1}\pi/2),\\
\text{Minimize} & f_2(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(\theta_1\pi/2)\cos(\theta_2\pi/2)\cdots\cos(\theta_{M-2}\pi/2)\sin(\theta_{M-1}\pi/2),\\
\text{Minimize} & f_3(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(\theta_1\pi/2)\cos(\theta_2\pi/2)\cdots\sin(\theta_{M-2}\pi/2),\\
\vdots & \vdots\\
\text{Minimize} & f_{M-1}(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(\theta_1\pi/2)\sin(\theta_2\pi/2),\\
\text{Minimize} & f_M(\mathbf{x}) = (1+g(\mathbf{x}_M))\sin(\theta_1\pi/2),\\
\text{where} & \theta_i = \frac{\pi}{4(1+g(r))}\left(1+2g(r)x_i\right), \quad \text{for } i=2,3,\ldots,(M-1),\\
 & g(\mathbf{x}_M) = \sum_{x_i\in\mathbf{x}_M}(x_i-0.5)^2,\\
 & 0\le x_i\le 1, \quad \text{for } i=1,2,\ldots,n.
\end{array}\right\} \quad (25)$$

The g function with $k = |\mathbf{x}_M| = 10$ variables is suggested. As before, there are $n = M + k - 1$ decision variables in this problem. This problem will test an MOEA's ability to converge to a curve and will also allow an easier way to visually demonstrate (just by plotting f_M with any other objective function) the performance of an MOEA. Since there is a natural bias for solutions close to this Pareto-optimal curve, this problem may be easy for an algorithm to solve, as shown in Figure 22 and 23, obtained using NSGA-II and SPEA2 after 200 generations and with other parameter setting as before. Because of its simplicity and ease of representing the Pareto-optimal front, we recommend that a higher-objective ($M \in [5, 10]$) version of this problem be used to study the computational time complexity of an MOEA.

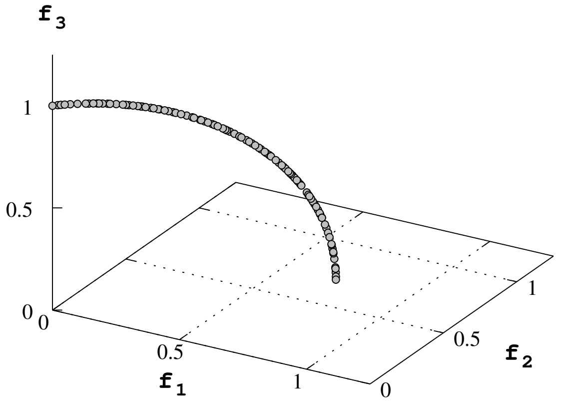


Figure 22: The NSGA-II population on test problem DTLZ5.

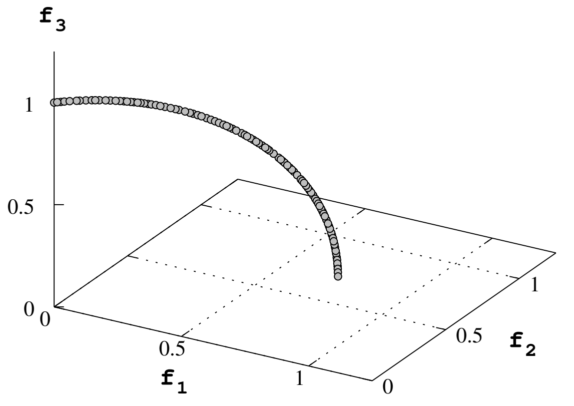


Figure 23: The SPEA2 population on test problem DTLZ5.

8.6 Test Problem DTLZ6

The above test problem can be made harder by doing similar modification to the g function in DTLZ5 as done in DTLZ3. However, in DTLZ6, we use a different g function:

$$g(\mathbf{x}_M) = \sum_{x_i \in \mathbf{x}_M} x_i^{0.1}. \tag{26}$$

The size of $\mathbf{x}_M$ vector is chosen as 10 and the total number of variables is identical as in DTLZ5. The above change in the problem makes NSGA-II and SPEA2 difficult to converge to the true Pareto-optimal front as in DTLZ5. The population after 500 generations of both algorithms are shown in Figures 24 and 25, respectively. The Pareto-optimal curve is also marked on the plots. It is clear from the figures that both NSGA-II and SPEA2 do not quite converge to the true

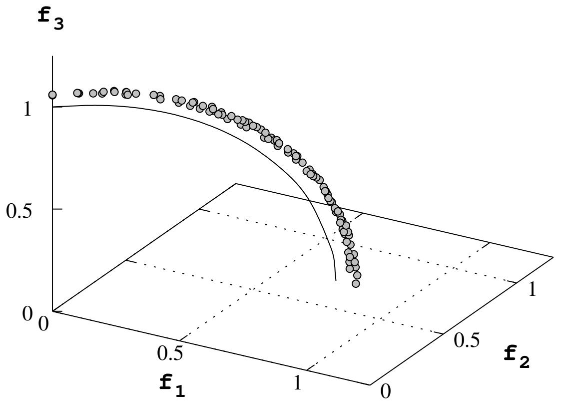


Figure 24: The NSGA-II population on test problem DTLZ6.

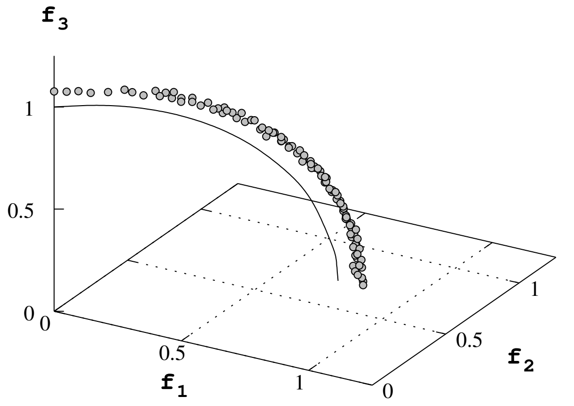


Figure 25: The SPEA2 population on test problem DTLZ6.

Pareto-optimal curve. The lack of convergence to the true front in this problem causes these MOEAs to find a dominated surface as the obtained front, whereas the true Pareto-optimal front is a curve. In real-world problems, this aspect may provide misleading information about the properties of the Pareto-optimal front, a matter which we discuss more in subsection 8.9 in this paper.

8.7 Test Problem DTLZ7

This test problem is constructed using the problem stated in equation 13. This problem has a disconnected set of Pareto-optimal regions:

$$\left.\begin{array}{rl}
\text{Minimize} & f_1(\mathbf{x}_1) = x_1, \\
\text{Minimize} & f_2(\mathbf{x}_2) = x_2, \\
\vdots & \vdots \\
\text{Minimize} & f_{M-1}(\mathbf{x}_{M-1}) = x_{M-1}, \\
\text{Minimize} & f_M(\mathbf{x}) = (1+g(\mathbf{x}_M))h(f_1, f_2, \ldots, f_{M-1}, g), \\
\text{where} & g(\mathbf{x}_M) = 1 + \frac{9}{|\mathbf{x}_M|}\sum_{x_i \in \mathbf{x}_M} x_i, \\
 & h(f_1, f_2, \ldots, f_{M-1}, g) = M - \sum_{i=1}^{M-1}\left[\frac{f_i}{1+g}\left(1+\sin(3\pi f_i)\right)\right], \\
\text{subject to} & 0 \le x_i \le 1, \quad \text{for } i = 1, 2, \ldots, n.
\end{array}\right\} \tag{27}$$

This test problem has 2^{M-1} disconnected Pareto-optimal regions in the search space. The functional g requires $k = |\mathbf{x}_M|$ decision variables and the total number of variables is $n = M + k - 1$. We suggest $k = 20$. The Pareto-optimal solutions corresponds to $\mathbf{x}_M = \mathbf{0}$. This problem will test an algorithm's ability to maintain subpopulation in different Pareto-optimal regions. Figures 26 and 27 show the NSGA-II and SPEA2 populations after 200 generations. It is clear that both

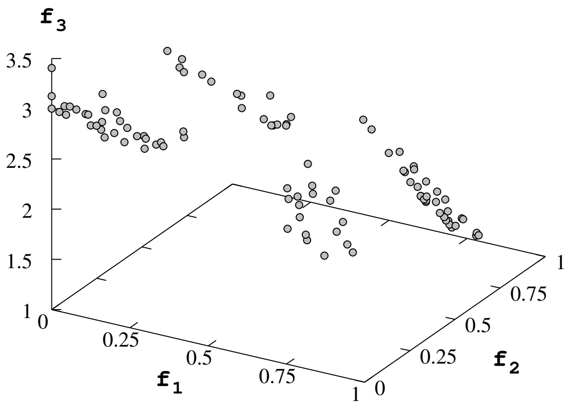


Figure 26: The NSGA-II population on test problem DTLZ7.

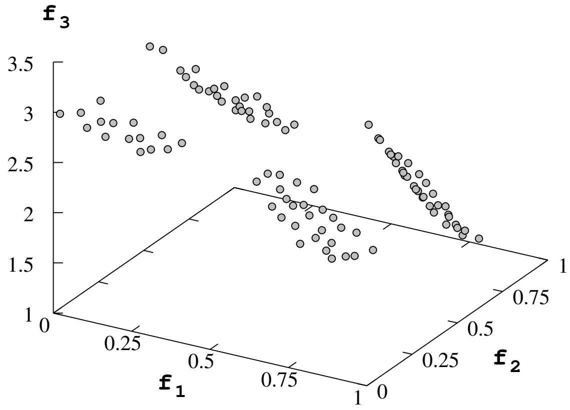


Figure 27: The SPEA2 population on test problem DTLZ7.

algorithms are able to find and maintain stable and distributed subpopulations in all four disconnected Pareto-optimal regions. The problem can be made harder by using a higher-frequency sine function or using a multi-modal g function as that described in equation 21.

8.8 Test Problem DTLZ8

Here, we use the constraint surface approach to construct the following test problem:

$$\left.\begin{array}{ll} \text{Minimize} & f_j(\mathbf{x}) = \frac{1}{\lfloor \frac{n}{M} \rfloor} \sum_{i=\lfloor (j-1)\frac{n}{M} \rfloor}^{\lfloor j\frac{n}{M} \rfloor} x_i, \quad j = 1, 2, \ldots, M, \\ \text{Subject to} & g_j(\mathbf{x}) = f_M(\mathbf{x}) + 4f_j(\mathbf{x}) - 1 \geq 0, \quad \text{for } j = 1, 2, \ldots, (M-1), \\ & g_M(\mathbf{x}) = 2f_M(\mathbf{x}) + \min_{\substack{i,j=1 \\ i\neq j}}^{M-1} [f_i(\mathbf{x}) + f_j(\mathbf{x})] - 1 \geq 0, \\ & 0 \leq x_i \leq 1, \quad \text{for } i = 1, 2, \ldots, n. \end{array}\right\} \quad (28)$$

Here, the number of variables is considered to be larger than the number of objectives or $n > M$. We suggest $n = 10M$. In this problem, there are a total of M constraints. The Pareto-optimal front is a combination of a straight line and a hyper-plane. The straight line is the intersection of the first $(M-1)$ constraints (with $f_1 = f_2 = \cdots = f_{M-1}$) and the hyper-plane is represented by the constraint g_M. MOEAs may find difficulty in finding solutions in both the regions in this problem and also in maintaining a good distribution of solutions on the hyper-plane. Figures 28 and 29 show NSGA-II ands SPEA2 populations after 500 generations. The Pareto-optimal region

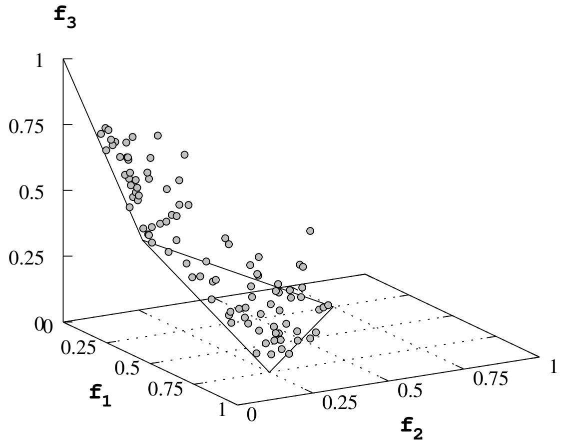


Figure 28: The NSGA-II population of non-dominated solutions on test problem DTLZ8.

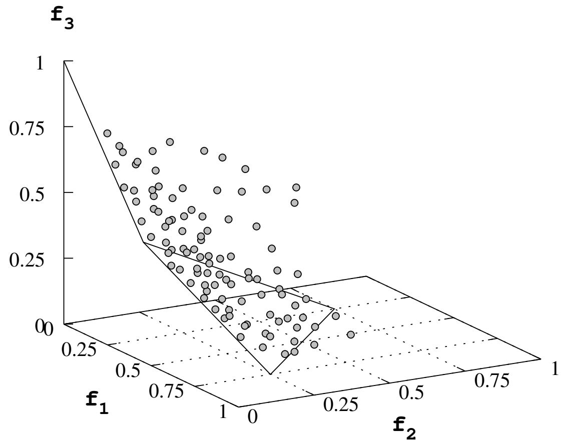


Figure 29: The SPEA2 population of non-dominated solutions on test problem DTLZ8.

(a straight line and a triangular plane) is also marked in the plots. Although some solutions on the true Pareto-optimal front are found, there exist many other non-dominated solutions in the final population. These redundant solutions lie on the adjoining surfaces to the Pareto-optimal front. Their presence in the final non-dominated set is difficult to eradicate in real-parameter MOEAs, a matter which we discuss in the next subsection.

8.9 Test Problem DTLZ9

This test problem is also created using the constraint surface approach:

$$\left.\begin{array}{ll} \text{Minimize} & f_j(\mathbf{x}) = \sum_{i=\lfloor (j-1)\frac{n}{M} \rfloor}^{\lfloor j\frac{n}{M} \rfloor} x_i^{0.1}, \quad j = 1, 2, \ldots, M, \\ \text{Subject to} & g_j(\mathbf{x}) = f_M^2(\mathbf{x}) + f_j^2(\mathbf{x}) - 1 \geq 0, \quad \text{for } j = 1, 2, \ldots, (M-1), \\ & 0 \leq x_i \leq 1, \quad \text{for } i = 1, 2, \ldots, n. \end{array}\right\} \quad (29)$$

Here too, the number of variables is considered to be larger than the number of objectives. For this problem, we suggest $n = 10M$. The Pareto-optimal front is a curve with $f_1 = f_2 = \cdots = f_{M-1}$, similar to that in DTLZ5. However, the density of solutions gets thinner towards the Pareto-optimal region. The Pareto-optimal curve lies on the intersection of all $(M-1)$ constraints. This feature of this problem may cause MOEAs difficulty in solving this problem. However, the symmetry of the Pareto-optimal curve in terms of $(M-1)$ objectives allows an easier way to illustrate the obtained solutions. A two-dimensional plot of the Pareto-optimal front with f_M and any other objective function should represent a circular arc of radius one. A plot with any two objective functions except f_M should show a 45^o straight line. Figures 30 and 31 show NSGA-II and SPEA2 populations after 500 generations on a f_3-f_1 plot of the 30-variable, three-objective DTLZ9 problem. The Pareto-optimal circle is also shown in the plots. It is clear that both

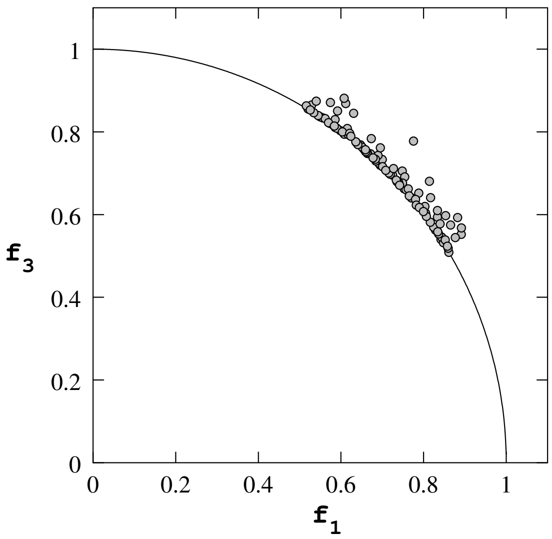


Figure 30: The NSGA-II population on test problem DTLZ9.

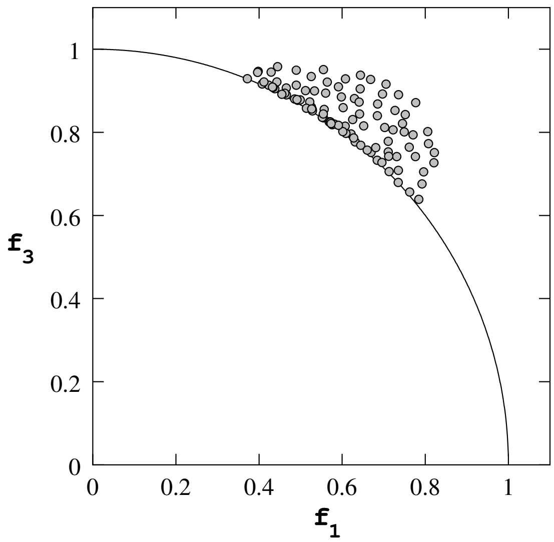


Figure 31: The SPEA2 population on test problem DTLZ9.

algorithms could not cover the entire range of the circle and there exist many non-dominated solutions away from the Pareto-optimal front.

Many of the above test problems (such as DTLZ6, DTLZ8, and DTLZ9) introduce another aspect of difficulty to large-objective real-parameter optimization techniques. In these problems, the Pareto-optimal curve or line (whatever may be the case) is weakly non-dominated with the adjoining surfaces (whose intersections give rise to the Pareto-optimal front). If a good representative set of solutions is not found on the true Pareto-optimal curve or line, a real-parameter optimization algorithm (such as an MOEA) can find a set of non-dominated solutions all of which may not be on the true Pareto-optimal curve or line. Figure 32 demonstrates this matter. With respect to two Pareto-optimal solutions A and B in the figure, any other solution in the shaded region is non-dominated to both A and B. The figure clearly demonstrates the fact the although a solution may not be on the true Pareto-optimal front (the straight line in the figure), it can exist in a set of non-dominated solutions obtained using an MOEA. In such problems, the obtained set of solutions may wrongly find a higher-dimensional surface as the Pareto-optimal front, although the true Pareto-optimal front may be of smaller dimension. It is interesting to note that this difficulty can also occur in problems having an M-dimensional Pareto-optimal front, as long as the Pareto-optimal surface is weakly non-dominated with adjoining surfaces. Another study (Kokolo, Kita, and Kobayashi, 2001) has also recognized that this feature of problems can cause MOEAs difficulty in finding the true Pareto-optimal solutions. However, in handling such problems, MOEAs with the newly-suggested ϵ-dominance concept (Laumanns et al., 2001) intro-

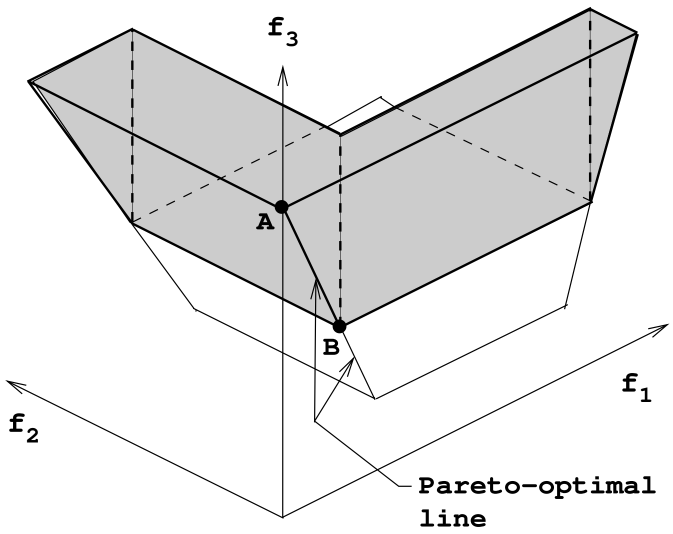


Figure 32: The shaded region is non-dominated with Pareto-optimal solutions A and B.

duced by the authors may be found useful. However, it is worth highlighting here that with the increase in the dimensionality of the objective space, the probabilities of occurrence of such difficulties is more. We term this additional difficulty as the problem of 'redundancy' in the obtained non-dominated solutions.

9 Conclusions

In this paper, we have suggested three approaches for systematically designing test problems for multi-objective optimization. The first approach simply uses a different, mostly-translated single-objective function as an objective. Although the construction procedure is simple, the resulting Pareto-optimal front may be difficult to comprehend. The second approach (we called a bottom-up approach) begins the construction procedure by assuming a mathematical formulation of the Pareto-optimal front. Such a function is then embedded in the overall test problem design so that two different types of difficulties of converging to the Pareto-optimal front and maintaining a diverse set of solutions can also be introduced. The third approach (we called the constraint surface approach) begins the construction process by assuming the overall search space to be a rectangular hyper-box. Thereafter, a number of linear or non-linear constraint surfaces are added one by one to eliminate some portion of the original hyper-box. The remaining enclosed region becomes the feasible search space. A few three-objective test problems are constructed illustrating the latter two approaches to demonstrate their relative advantages and disadvantages. Finally, a number of test problems have been suggested and attempted to solve using two popular state-of-the-art MOEAs (NSGA-II and SPEA2) for their systematic use in practice.

In this paper, we have not suggested any explicit constrained test problem, although problems constructed using the constraint surface approach can be treated as constrained test problems. However, other difficulties pertinent to the constrained optimization suggested in two-objective constrained test problem design elsewhere (Deb et al., 2000) can also be used with the proposed procedures for constrained test problem design.

Acknowledgments

Authors acknowledge the support from Swiss National Science Foundation under the ArOMA project 2100-057156.99/1.

References

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms.* Chichester: Wiley.

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

Deb, K., Pratap, A., and Meyarivan, T. (2001). Constrained test problems for multi-objective evolutionary optimization. *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO-2001)*, (pp. 284–298).

Deb, K., Agrawal, S., Pratap, A., Meyarivan, T. (2000). A Fast Elitist Non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *Proceedings of the Parallel Problem Solving from Nature VI Conference*, (pp. 849–858).

Fonseca, C. M. and Fleming, P. J. (1995). An overview of evolutionary algorithms in multi-objective optimization. *Evolutionary Computation Journal 3*(1), 1–16.

Knowles, J. D. and Corne, D. W. (1999). The Pareto archived evolution strategy: A new baseline algorithm for multiobjective optimization. *Proceedings of the 1999 Congress on Evolutionary Computation*, (pp. 98–105).

Kokolo, I., Kita, H., and Kobayashi, S. (2001). Failure of Pareto-based MOEAs: Does non-dominated really mean near to optimal? *Proceedings of the Congress on Evolutionary Computation 2001*, pp. 957–962.

Kursawe, F. (1990). A variant of evolution strategies for vector optimization. In *Parallel Problem Solving from Nature I (PPSN-I)*, pp. 193–197.

Laumanns, M., Rudolph, G. and Schwefel, H.-P. (2001). Mutation control and convergence in evolutionary multi-objective optimization. *Proceedings of the Seventh International Mendel Conference on Soft Computing (MENDEL 2001).*

Laumanns, M., Thiele, L., Deb, K., and Zitzler, E. (2001). On the convergence and diversity-preservation properties of multi-objective evolutionary algorithms. *TIK-Report No. 108*, Zurich, Switzerland: Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich.

Miettinen, K. (1999). *Nonlinear Multiobjective Optimization.* Boston: Kluwer.

Poloni, C., Giurgevich, A., Onesti, L. and Pediroda, V. (2000). Hybridization of a multiobjective genetic algorithm, a neural network and a classical optimizer for complex design problem in fluid dynamics. *Computer Methods in Applied Mechanics and Engineering 186*(2–4), 403–420.

Schaffer, J. D. (1984). *Some experiments in machine learning using vector evaluated genetic algorithms.* Ph. D. Thesis, Nashville, TN: Vanderbilt University.

Schott, J. R. (1995). Fault tolerant design using single and multi-criteria genetic algorithms. Master's Thesis, Boston, MA: Department of Aeronautics and Astronautics, Massachusetts Institute of Technology.

Tamaki, H. (1996). Multi-objective optimization by genetic algorithms: A review. *Proceedings of the Third IEEE Conference on Evolutionary Computation*, pp. 517–522.

Tanaka, M. (1995). GA-based decision support system for multi-criteria optimization. In *Proceedings of the International Conference on Systems, Man and Cybernetics*, Volume 2, pp. 1556–1561.

Veldhuizen, D. V. (1999). *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. Ph. D. Thesis, Dayton, OH: Air Force Institute of Technology. Technical Report No. AFIT/DS/ENG/99-01.

Viennet, R. (1996). Multicriteria optimization using a genetic algorithm for determining the Pareto set. *International Journal of Systems Science 27*(2), 255–260.

Zitzler, E. (1999). Evolutionary algorithms for multiobjective optimization: Methods and applications. Ph. D. Thesis, Zürich, Switzerland: Swiss Federal Institute of Technology (ETH) (Dissertation ETH No. 13398).

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

Zitzler, E., Laumanns, M. and Thiele, L. (2001). SPEA2: Improving the strength Pareto evolutionary algorithm. *TIK-Report No. 103*. Zurich, Switzerland: Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich.

Zitzler, E. and Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation, 3*(4), 257–271.